Investigating Data Standardisation and Modelling Challenges to Enable Advanced Power Systems Analysis

A thesis submitted for the degree of Doctor of Philosophy

by

Corinne Margaret Shand

Supervisor: Prof. Gareth Taylor

Department of Electronic and Computer Engineering

College of Engineering, Design and Physical Sciences

Brunel University London

December 2017

Abstract

As the power industry moves towards more active distribution networks there is an increased requirement for greater analysis and observability of the current state of the network. There are a number of challenges for utilities in realising this including the quality and accuracy of their network models; the lack of integration between network models and the large quantities of sensor data being collected; the security and communication challenges posed when installing large numbers of sophisticated sensors across distribution networks; and the exponential increase in computing power required to fully analyse modern network configurations. This thesis will look at these challenges and how cloud computing can be used to provide novel solutions by providing secure platforms on which to deploy complex data collection and network analysis applications.

One of the main research contributions is the use of remote data collection from Micro Phasor Measurement Units (µPMUs), which collect synchronised information about the state of the distribution network. Impedance equations are applied to network data recorded from µPMUs and the results are compared to network models. This identifies areas of the distribution network as requiring resurveying or upgrading, potentially impacting planning for installation of generation or load. Triggers can be used to reduce the bandwidth of data being sent by a µPMU; these were tested with real world data to highlight how a combination of local intelligence and cloud-based analysis can be used to reduce bandwidth requirements while supporting the use of detailed measurement data for cloud-based analysis in a fault detection system.

Power flow analysis is an important tool for both operations and planning engineers, and as computing power has increased the time required to run individual power flow analysis cases has decreased rapidly. However there has also been a corresponding increase in the complexity of the data as utilities seek to model and analyse distributed energy resources attached on the medium and low voltage networks. This has made network models more complex, exponentially increasing the number of contingencies that need to be analysed in an emergency situation.

Another main research contribution is a demonstration of the challenges faced when using a commercial cloud platform to inexpensively solve computationally intensive power flow problems and the time, costs and feasibility of performing N-1 and N-2 analysis on a 21,000-bus network. It includes a full analysis and comparison of execution times and costs for different commercial cloud system configurations as well as the extrapolated costs required to run a full N-2 analysis of over 420 million contingencies in under 10 minutes. This includes a demonstration of a cloud client and server application developed as part of this research that leverages a commercial power flow engine.

Finally, this thesis will summarise how each of these research outputs can be combined to provide utilities with a commercial, open, standards-based cloud platform for continuous, automated contingency analysis using real-time sensor data based on current network conditions. This would better inform control engineers about areas of vulnerability and help them identify and counter these in real-time.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Gareth Taylor for his guidance and advice throughout this research.

I would like to thank Open Grid Systems for sponsoring this project; and especially Dr. Alan McMorran for his support, expertise and encouragement over the last four years.

Special acknowledgements must also go to Dr. Narsi Vempati, Dr. Herminio Pinto, Dr. Brian Stott and Guanji Hou at Nexant and to Dr. Emma Stewart, without whom this research would not have been possible.

Finally I need to thank my family and friends, in particular Marilyn and Anthony, for their endless support and patience while I completed this journey.

Declaration of Authorship

The work described in this thesis has not been previously submitted for a degree in this or any other university, and unless otherwise referenced it is the author’s work.

Table of Contents

Table of Figures

Table of Tables

Abbreviations

ACAM	Angle Constraint Active Management
ACO	Ant Colony Optimisation
ADN	Active Distribution Network
AMI	Advanced Metering infrastructure
AMI	Amazon Machine Image
ANM	Active Network Management
ARPA-E	Advanced Research Projects Agency - Energy
AWS	Amazon Web Services
BCSE	Branch Current based State Estimation
BETTA	British Electricity Trading Transmission Arrangements
BOX	Bus Observability Index
BPSO	Binary Particle Swarm Optimisation
CCAPI	Control Center Application Programming Interface
CDF	Common Data Format
CGMES	Common Grid Model Exchange Standard
CIEE	Council on International Educational Exchange
CIM	Common Information Model
CPU	Central Processing Unit
DACF	Day-Ahead Congestion Forecasting
DEF	Data Exchange Format
DER	Distributed Energy Resources
DKF	Discrete Kalman Filter
DNO	Distribution Network Operator
DNP3	Distributed Network Protocol
DOE	Department of Energy
DSM	Demand Side Management
DSO	Distribution System Operator
DSSE	Distribution System State Estimation
EBS	Elastic Block Storage

EC2	Elastic Compute Cloud
EMS	Energy Management System
ENTSO-E	European Network of Transmission System Operators for Electricity
EPFL	École Polytechnique Fédérale de Lausanne
EPRI	Electric Power Research Institute
GB	Great Britain
GOOSE	Generic Object Orientated Substation Events
GPS	Global Positioning System
HIL	Hardware-in-the-Loop
HPC	High Performance Computing
HTTPS	Hyper Text Transfer Protocol Secure
IBPSO	Improved Binary Particle Swarm Optimisation
ICCP	Inter-Control Centre Communications Protocol
IEC	International Electrotechnical Commission
IED	Intelligent Electronic Device
IEEE	Institute of Electrical and Electronics Engineers
IGA	Immunity Genetic Algorithm
ILP	Integer Linear Programming
LBNL	Lawrence Berkeley National Laboratory
MMS	Market Management System
NERC	North American Electric Reliability Corporation
NG	National Grid
NIC	Network Innovation Competition
NICTA	National ICT Australia
NYPA	New York Power Authority
Ofgem	Office of gas and electricity markets
OLS	Ordinary Least Squares
PDC	Phasor Data Concentrator
PMU	Phasor Measurement Unit
PNNL	Pacific Northwest National Laboratory
PSAT	Power System Analysis Toolbox

PSL	Power Standards Laboratory
PSLF	Positive Sequence Load Flow
PSS/E	Power System Simulator for Engineering
PV	Photovoltaics
QUB	Queen's University Belfast
RTSE	Real-Time State Estimator
RTU	Remote Terminal Unit
S3	Simple Storage Service
SCADA	Supervisory Control And Data Acquisition
SCL	Substation Configuration Language
SCOPF/ SCOPE	Security Constrained Optimal Power Flow
SHE	Scottish Hydro Electric
SHEPD	Scottish Hydro Electric Power Distribution
SHETL	Scottish Hydro Electric Transmission Limited
SMU	Synchronous Measurement Unit
SO	System Owner
SORI	Summation of Redundancy Index
SPEN	Scottish Power Energy Networks
SSEN	Scottish and Southern Electricity Network
SSO	Sub Synchronous Oscillation
TLS	Transport Layer Security
TO	Transmission Owner
TSO	Transmission System Operator
UCTE	Union for the Co-ordination of Transmission of Electricity
US	United States
VPN	Virtual Private Networks
WAMS	Wide Area Monitoring Systems
WLS	Weighted Least Squares
WMU	Waveform Measurement Unit
µPMU	Micro Phasor Measurement Unit

Chapter 1 Introduction

1.1 Changes in the Grid

Power systems in Great Britain (GB), and worldwide, are facing challenges associated with the evolution to a low carbon and high reliability future. These include:

- Fast changes in generation mix
- Slowly evolving network components
- Relatively high cost of upgrades per capita
- Environmental impact of adding equipment
- The increase in frequency/impact of severe weather events [1]

Managing the effect of evolving system dynamics requires enhanced monitoring at both transmission and distribution voltage levels. Improved monitoring also provides vital network services and safely directs power from lower voltage levels to the areas where it is required via the interconnected transmission network.

Real-time monitoring must be expanded beyond its current view of substations. It will need to provide an increased visibility of strategic locations, service providers and distributed energy resource (DER) installations. This will give an improved ability to observe and control the network. It will allow action to be taken in response to wider system events, and using enhanced monitoring solutions will benefit the detection of issues that will have a direct impact on system stability and network performance. This will also ensure that Distribution Network Operators (DNO) maintain a safe and reliable network. The development of new sensors, advanced communications equipment and improved information technologies will expand the efficiency of the distribution system. Improved reliability and power quality will support the increase of DER penetration. It will also aid consumers by improving their quality of service and providing an increase in customer choice.

A progressively more complex GB distribution network will require advanced levels of monitoring. Data must be recorded at an increased number of points, observing further afield than the current substation boundaries. This complexity is heightened by the inclusion of additional meshed networks and load points in the ageing distribution network: these will require more complex monitoring. Data needs to be collected at optimal network locations. The data is required to be at a higher frequency and granularity, for use in real-time system operations and longer term network planning [2].

The stability of the grid is being affected by the addition of DER at a low voltage level. This is creating situations where feeder loads can no longer be considered predictable, as they do not follow daily, weekly or seasonal patterns. The unpredictability makes it more challenging to balance supply and demand on the network. Utilities therefore require increased analysis of the impact that installed and planned generation will have on their grid for short, medium and long-term planning, as well as daily operation. To achieve this the data needs to be analysed with respect to the low voltage electrical network models, but the quality of this data is often unreliable, if it exists at all. It is impractical and costly to resurvey the networks to obtain the necessary network models, resulting in an underutilisation of the collected data [3].

1.2 Current Metering Options

1.2.1 PMUs

Phasor Measurement Units (PMU) are high frequency devices that sample analogue voltage and current data in synchronisation with a Global Positioning System (GPS) clock [4]. Data is recorded at a rate of 48 samples per cycle and sent to a Phasor Data Concentrator (PDC). The PDC validates the samples and sorts them using their GPS time-stamp, which ensures that all PMU samples recorded at the same time across a network are processed together. Samples that arrive late or fail to arrive can be identified using their time stamps and marked as problematic. An example of the voltage and current data collected from two PMUs on a single network over the course of an hour is shown in Figure 1.1.

A phasor is a complex equivalent of a simple-frequency cosine wave quantity such that the complex modulus is the cosine wave amplitude and the complex angle (in polar form) is the cosine wave phase angle [4]. Phasors are typically used to represent the power frequency signal in AC power, which is approximately 50 Hz or 60 Hz.

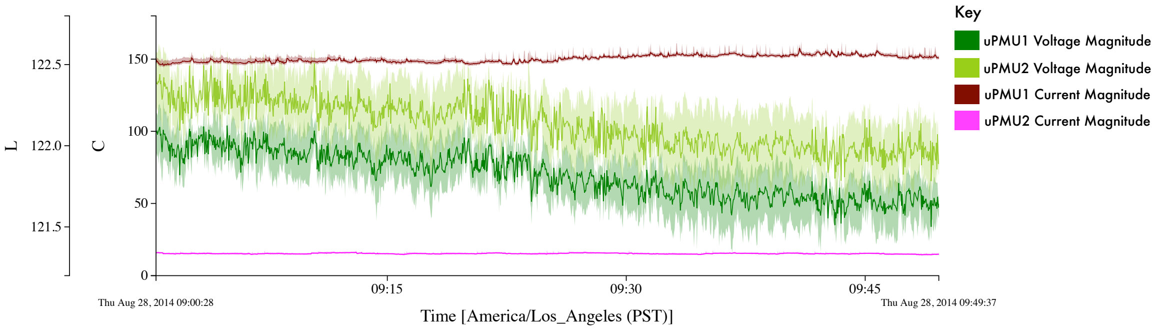


Figure 1.1: PMU Voltage and Current Outputs over 1 Hour

1.2.2 µPMUs

Transmission level PMUs connect to high voltage and are therefore higher cost than is considered reasonable for distribution expenditure [5]. The price of a distribution level µPMU is significantly lower [6] and in a range that is more realistic for data collection on the distribution network. µPMUs are therefore becoming a more common method of data collection. Distribution grids have much smaller and faster changing angle differences that cannot be recorded efficiently with a transmission level PMU, but µPMUs can capture portions of the operating state of the system to provide actionable intelligence in real-time. Other advantages of a µPMU include the low installation cost and a measurement resolution that is orders of magnitude better than that of a standard PMU [7].

1.2.3 Smart Meters

Smart meters are commonly installed on low voltage distribution grids at domestic locations, where they collect and send data detailing the energy usage of consumers at that property throughout the day. Smart meters provide increased insight into these areas of the network, but they are only useful if network operators have access to the data in close to real-time. This access is not always available in the GB system, or some parts of the North American system [8], as a liberalised electricity market separates meter readings and network operations.

In markets where the DNO is also the retail provider it is common for smart meters to be integrated with network and geographical data. This setup allows for a direct correlation between a grid supply point on the network model and the smart meter. In some cases, for example in the GB system, the DNO may have no direct visibility of a smart meter and the data it is collecting. This makes it harder to correlate a meter with a consumption point on the network.

1.3 Cloud Computing

Cloud computing is a flexible method of delivering server infrastructure and allowing the resources to scale according to demand. It allows for relatively cheap access to vast amounts of processing power without the requirement to invest in dedicated computing facilities, which is ideal for the analysis of a high volume of data. This means that when users require additional computing resources, additional hardware is employed to meet their demand. The additional demand could be from a new customer, with each customer using a single server instance, or it could be in terms of providing Central Processing Unit (CPU) cycles to meet a computationally intensive task. Cloud computing resources can quickly be scaled up to deal with a large influx of data, and can be reduced when demand returns to normal levels.

Cloud computing should not be confused with making resources directly available on the World Wide Web, nor does it imply free access to the instances running in the cloud. To set up additional resources on demand in a cloud environment, either automatically or through a manual process, a base system is required. A running instance of this base system with its own IP address, access controls and data stores can be created. This running instance may communicate with cloud databases or other instances, and it may run servers that allow direct connection by users.

1.3.1 Cloud Security

Unless otherwise specified, the cloud platform used in this research is Amazon Web Services (AWS) [9]. AWS is a cloud service that can be adjusted in real time depending on the requirements. Different parts of AWS can work together to provide the best solutions for the current needs. It includes storage, computing power, databases and management tools, which are all available on a ‘pay what you use' basis. An AWS Amazon Machine Image (AMI) is an encrypted machine image stored in Amazon Elastic Block Storage (EBS) or Simple Storage Service (S3). An AMI is a template of a root drive; it contains the operating system and any software and data required, such as database servers, middleware and web servers. AMIs are used to create instances, which are virtual servers in the cloud. One AMI can create many instances at the same time [10].

The following aspects must be considered in any cloud computing project:

- Securing the account(s) that will be used to administrate the cloud environment
- Securing the machine image files that will be started for each task
- Securing the running instances by ensuring logins for the instances are managed appropriately
- Ensuring only interactions that are required can take place
- Ensuring application-level security

1.3.2 Trusted Cloud

Trusted clouds are a development of private clouds as they only allow access to approved people or organisations. Users must also verify that the image used to start a machine instance has not been modified since it was created, providing a further layer of security. This ensures a more dependable and secure environment for any applications or data that are deployed in the cloud. Different access levels can be implemented; for example an administrator could be allowed access to the underlying infrastructure while end users are given a read-only account [11], [12].

1.3.3 Trusted Cloud & µPMUs

A trusted cloud allows the integrity of a complete cloud environment to be measured. This includes the hardware platform, firmware, and all operating system components responsible for booting. Integrating trusted cloud implementations into power systems would allow cloud administrators - ideally the System Owners (SO) - to measure and verify infrastructure integrity. Additional arrangements will be required to ensure the integrity of end-users. This is likely to be a secure channel to allow access to the cloud platform [13] combined with two-factor authentication. A challenge to overcome during implementation is the additional software that will be required to enable the connecting µPMUs to automatically authenticate, integrate and communicate with a trusted cloud platform. The cloud-based platform must meet the utility's requirements for security, including verification that any connecting µPMUs are trustworthy before they are connected to the network. µPMUs must also be able to verify that they are communicating with a known and trusted platform [14].

A cloud-based system will allow multiple µPMUs to automatically connect, authenticate, integrate and send their data to the back-end system [3]. A Trusted Cloud [11] platform can help to ensure that the connection and data is secure. If implemented, the integrity of the complete GB power system would rely on the ability of DNOs/Transmission System Operators (TSO) to independently update and maintain the segments of data that are related to their own equipment and network.

1.4 Power Flow

Successful power system operation under normal balanced three-phase steady-state conditions requires the following [15]:

- Generation supplies the demand (load) plus losses
- Bus voltage magnitudes remain close to rated values
- Generators operate within specified real and reactive power limits
- Transmission lines and transformers are not overloaded

The calculations that investigate these requirements are known as power flow or load flow analysis. These calculations are necessary for system planning and during operation, allowing operators to review the effectiveness of the network providing a supply to the connected and predicted load in a range of situations. The results enable the optimisation of the power system with minimum operating costs.

The power flow analysis problem is the computation of voltage magnitude and phase angle at each bus in a power system under balanced three-phase steady-state conditions [15]. Other information that can be gained from the calculations are the real and reactive power flows in equipment and also system losses. The input for power flow analysis includes bus, transmission line, and transformer data.

A balanced three-phase steady-state network is usually represented as a single line diagram, and the per-unit system is used. This simplifies the calculations required and still produces an acceptable and useable solution. In the per-unit system voltages, currents, impedances and powers are all expressed as a percentage or fraction of pre-defined base quantities [16]. The benefits of this are the simplicity of number representation and the ease of dealing with transformer ratios.

There are various iterative methods that can be used to solve a power flow. The Gauss-Seidel method starts with the network in matrix form, with an initial guess of the solution, substituting in the values obtained each time a new iteration begins. The Newton-Raphson method has a similar beginning, but after the initial guess a Taylor Series expansion occurs to form a set of linear equations that can be solved to find the value for the next iteration. Neither of these methods is perfect; the Gauss-Seidel is more economical, but not fast, while the Newton-Raphson method converges more often but is limited when dealing with smaller networks. The Fast Decoupled Load Flow method simplifies the previous methods to vastly speed up the power flow process and make it more reliable [17]. Convergence can be reached in seconds, which is ideal for SOs who need to know the results of a contingency analysis on their networks in close to real time.

1.5 Objectives

This research will investigate the design and development of novel, cutting-edge solutions for integrating, storing, sharing, visualising and analysing complex power systems data. It will explore and demonstrate the use of fast, scalable, secure and robust frameworks and architectures. These will cover the requirements of utilities both within the UK and worldwide to exploit their data and the use of cloud computing infrastructures. It will enable utilities to fully utilise the data by proposing and developing solutions for the automated invocation of advanced analysis applications on large, real-world networks. The main objectives of this research are:

- To investigate communication and data processing standards within the power industry
- To present a methodology for addressing the problems faced in deploying µPMUs at the distribution level
- To develop a novel approach that enables fast, detailed analysis of all possible contingencies on a network by using a highly parallelisable power flow based within a cloud environment
- To select and use appropriate methodology for the completion of research fulfilling the above objectives

1.6 Principal Contributions to Research

The main contributions to knowledge, as presented in this thesis, can be summarised as follows:

- Use of remote data collection from µPMUs to support improvements to network models for analysis, and the extension of wide area monitoring into distribution networks. Impedance equations are applied to real-world µPMU network data to prove the viability of using it to identify areas of the network where the analytical models differ from the as-built models, and so may require resurveying or upgrading to support the installation of localised generation or advanced control schemes.
- The design of a cloud-based architecture for deploying µPMUs in geographically remote locations with unpredictable communication bandwidth and low levels of physical security. This cloud architecture, combined with localised triggers implemented at the device level, is used to demonstrate a significant reduction in the amount of data being sent by µPMUs while still supporting real-time notification of abnormal system events.
- A highly parallelisable power flow analysis is run on a commercial cloud computing platform. A cloud client and server application was developed using a commercial power flow engine to execute a highly parallelised power flow with minimal human interaction and a standard, repeatable start-up procedure and configuration.
- A demonstration of the challenges faced when using a commercial cloud platform to inexpensively solve computationally intensive power flow problems. These include the restriction of system resources, the allocation of volume storage space and the presence of results that are outwith the expected performance of a machine.
- Investigation of the optimal settings for running an N-1 contingency analysis for a real-world 21,000-bus transmission model in the cloud environment to optimise both execution time and the financial costs. Tests were run on different machine families with variations in the number of cores and

number of instances being used. Contingency analysis of approximately 30,000 different contingencies was run on one processor, on multiple processors (dependent on the number of cores available for each machine time), and as multiple lists splitting the contingencies among instances.

- The linear scalability of running N-2 contingency analysis was examined using the optimal machine settings proposed from the N-1 research. This included the use of Monte-Carlo simulations on a random selection of N-2 contingencies to provide the extrapolated costs required to run a full N-2 analysis of over 420 million contingencies in under 10 minutes.
- Finally an architecture is proposed that would allow commercial power system analysis vendors to offer a standards-based, on-demand analysis service to utilities using either public or private cloud platforms. This would allow utilities to integrate the real-time data coming from µPMUs with the network models to perform complex, computationally intensive analysis automatically. It would provide greater visibility of the system status and pro-actively respond to network disturbances.

1.7 Publications

The research described in this thesis has resulted in a number of conference publications, and one paper that is being prepared for consideration for a journal publication.

1.7.1 Conference Publications

C. M. Shand, K. W. Brady, E. M. Stewart, C. M. Roberts, A. W. McMorran, P. Mohapatra and G. A. Taylor, "Improving Actionable Observability of Large Distribution Networks for Transmission Operators to Support Improved System Control, Fault Detection and Mitigation," *24th International Conference & Exhibition on Electricity Distribution (CIRED)*, Glasgow, 2017

C. M. Roberts, C. M. Shand, K. W. Brady, E. M. Stewart, A. W. McMorran and G. A. Taylor, "Improving Distribution Network Model Accuracy using Impedance Estimation from Micro-Synchrophasor Data," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, 2016, pp. 1-5.

C. M. Shand, A. W. McMorran, E. M. Stewart and G. A. Taylor, "Exploiting Massive PMU Data Analysis for LV Distribution Network Model Validation," *2015 50th International Universities Power Engineering Conference (UPEC)*, Stoke on Trent, 2015, pp. 1-4.

C. M. Shand, A. W. McMorran, and G. A. Taylor, "Integration and Adoption of Open Data Standards for Online and Offline Power System Analysis," *2014 49th International Universities Power Engineering Conference (UPEC)*, Cluj-Napoca, 2014, pp. 1-6.

1.7.2 Journal Publications

C. M. Shand, A. W. McMorran, N. Vempati and G. A. Taylor, “Power Flow in a Cloud to give Improved Access to Users” (In preparation for submission to IEEE Transactions on Power Systems in 2018)

1.8 Structure of the Thesis

This thesis is organised into eight chapters:

Chapter 1 has introduced the research presented in this thesis. It has provided background information on the topics covered in the thesis and outlined the objectives behind the work. The principal contributions to research are detailed, as well as conference and journal publications.

Chapter 2 is a comprehensive literature review of relevant work surrounding PMUs, µPMUs, and their current uses. Other researchers have documented various studies and implementations, on the applicable network, and these are examined and commented upon where appropriate.

Chapter 3 discusses the use of standards within the power industry and their necessity to ensure the reliability and security of power both on the grid and while being supplied to consumers. It covers high and low frequency exchanges, network model exchanges, and the support of electricity markets. Also included is a section on standard interoperability.

Chapter 4 looks at different types of networks. Transmission and distribution networks are reviewed separately, followed by a discussion on their variances.

Chapter 5 explains how the installation and running of µPMUs on distribution networks can benefit from the use of a trusted cloud platform. It looks at secure communications between devices and the cloud. This is followed by the proposal and testing of an event-based trigger system to reduce the synchronisation frequency of µPMU data, and the benefits that this will bring to TSOs and DNOs.

Chapter 6 presents work completed on running power flow analysis in a cloud environment. The process of the research is described, including information on the software and simulations used for testing. The results are discussed in terms of how they will affect the future of running a power flow and its contingencies.

Chapter 7 builds on the work done in Chapter 6 by automating the process of running power flow analysis in a cloud environment. It confirms the results obtained in the previous chapter before further investigating the optimal settings for both N-1 and N-2 contingency analysis.

Chapter 8 concludes the thesis. It provides a summary of the research presented in this thesis and the key findings. It also proposes future work that will build on what has already been achieved.

Chapter 2 Literature Review

2.1 Introduction

This chapter reviews existing work that relates to the topics discussed in the thesis. This includes PMUs, state estimation, developments in distribution and transmission networks and examples of current use of a cloud environment in power system analysis.

2.2 PMUs

A PMU is a high frequency device that samples analogue voltage and current data in synchronisation with a GPS-clock [4]. Data is recorded at a rate of 48 samples per 50/60 Hz cycle are sent to a PDC, for sorting and validation. Each phasor has a time-stamp from the GPS clock; this is used to ensure that all PMU samples taken at the same time across a network are processed together. If a PMU data packet is late to arrive at a PDC it can be identified via the time-stamp.

There is discussion by [18] into synchronised measurements in phasor form. It focuses mainly on communication and data processing at the transmission level, with the use of PMUs. This is contrasted with the distribution network, where the waveforms have higher distortions and there is increased DER, which suggests a less stable system. Their proposal is a Synchronous Measurement Unit (SMU), which is an evolved PMU. It would have increased functionality in many aspects of the power system, including energy transfer characterisation, system awareness and control measurements. New algorithms would be required to apply to distribution level data.

2.2.1 µPMUs

µPMUs are a highly accurate but lower-cost variation of a standard PMU. Their economic benefit is necessary for µPMUs to be viable as multiple devices will be required. They are intended for installation in distribution networks, as phase angles at the distribution level change by a much smaller amount than in transmission [19]. Voltage angles on a distribution network can be as much as two orders of magnitude smaller than those on a transmission network, due to shorter transmission lines and lower power flow [20]. Connecting µPMUs to the distribution network is much cheaper to do as they can be installed via an already connected relay, and as such many newer relays are now built with synchrophasor capability.

Examples of the use of µPMU data are found in monitoring and characterisation of high renewable penetration feeders and generators, phase identification, and state estimation. Work has been done in [21] to investigate the hardware limitations of µPMUs with regards to their operation in the distribution network. This is done by way of building an open-box µPMU that complies with IEEE C37.118 standards [22], [23]. The device was built for around £350, so it is a cheaper version of transmission level PMUs but provides an equal view of the distribution system.

The measurement accuracy of a µPMU and its dependencies in the form of the required instrument transformers are discussed in [84]. In this work it was found that there is an inherent accuracy barrier in the utilisation of advanced sensor data and in the existing electrical network analysis models. These modelling accuracy barriers can impose a limit on distribution grid development in areas such as distributed generation interconnection and advanced automation and control schemes, for example active network management systems. The sensor accuracy data is not a key factor in the existing state of distribution planning, since the data needs for real time operational control objectives are different from those required for the initial assessment in the planning context. The applications of the data collected, whether from µPMUs or other sensors, must be appropriate based on the bandwidth for error. This accounts for the entire measurement chain from the wire to the point where data is utilised. An integrated approach for control algorithms and validation is required.

2.3 State Estimation

State estimation is the prediction of voltage and current magnitude and angle, at all buses or nodes on a network, using the existing voltage, current, real and reactive power measurements, and a system that accounts for missing or bad data [19]. For accurate state estimation the topology processor must identify points of bad data on the network and look for errors in the status of switching devices.

A United States (US) Department of Energy (DOE) demonstration project, started in 2009, is a study into the implementation of a three-phase linear tracking state estimator by Virginia Electric and Power Company [24]. When line impedances are known, PMUs in a substation can effectively estimate the bus voltages in neighbouring substations. For the project, a PMU-only three-phase estimator that runs at 30 times a second is installed on a 500 kV system. In this instance the synchrophasor estimation is a linear estimation problem, and it considers each new frame a separate problem:

$$\mathrm{z} = \begin{bmatrix} V \\ I_{flow} \end{bmatrix} = \begin{bmatrix} II \\ yA + \ y_s \end{bmatrix} x + e$$

The state estimator was run inside the openPDC software. Although it was running successfully by July 2012 the results in this paper are from a test set of 10 substations. The estimator has potential, but until it is physically implemented with actual results, the results have variances of 0.002 pu for voltage phasors and 0.01 pu for current phasors.

The implementation, testing, and performance of phasor measurements in an industrial state estimation at the New York Power Authority (NYPA) are described in [25]. The method that is being used for state estimation is minimising the weighted least squares (WLS) objective function:

$$Min. J(\mathrm{X}) = [Z - h(X)]^T R^{-1} [Z - h(X)]$$

In order to use the state estimator as a phasor state estimator new rows were added to the Jacobian matrix for the inclusion of phase angle measurements. Reference buses for estimation were selected among the PMU buses, as the phase angles will need an appropriate reference. In an attempt to identify bad data within the network, errors are searched for in the phase angle reference measurements. If any errors are found, the reference measurement that was used is declared anomalous and the process is repeated with a new reference measurement. The study concluded that even small errors in phasor measurements could lead to sizable errors in metering accuracy, which indicates a higher probability of bad data identification.

NYPA observed that the results of both normal state estimations and phasor state estimations were similar. More measurements were required in the phasor state estimator to produce the same observability. Unsurprisingly, it was concluded that the effectiveness of the phasor state estimator depended on the precision of the phasor metering, but that using the optimal number and locations of PMUs in the network was beneficial.

The work in [20] uses a linear three phase state estimator for applications in distribution systems. It is simulated on the IEEE 13-bus feeder. The estimator that has been proposed will make use of synchrophasor measurements, which will be provided by a µPMU. Their non-linear state estimator has been solved by the WLS method:

$$z = h(x) + e$$

The authors have suggested that a more accurate state estimation can be achieved by using the 3-phase representation of the distribution network:

$$Z = HV + e$$

The simulation results show that the state estimator can provide accurate 3-phase estimations. Given the high sampling rate and high precision measurements of µPMUs, it is proposed that this linear 3-phase state estimator has the potential for real time monitoring in distribution systems.

The uncertainty of voltage profiles in PMU-based DSSE is discussed in [26]. Distribution system state estimation (DSSE) tools estimate the operating conditions of power distribution networks. The accuracy of the data that is estimated using DSSE plays a key role in management and control functions. The paper presents a mathematical analysis to emphasise the factors that most affect the voltage profile. The method used is the WLS approach, using a branch-current estimator. The theoretical analysis was verified with a 95-bus network and showed that the accuracy of PMU-based DSSE is vital for the efficiency of management and control functions.

2.4 Distribution

Reference [19] discusses the challenges involved in the integration of µPMU data with distribution network operations. Two projects are discussed, and they are OpenPMU at Queen's University Belfast (QUB) and a µPMU project between the Council for International Educational Exchange (CIEE), Lawrence Berkeley National Laboratory (LBNL) and Power Standards Laboratory (PSL). An example is given of where µPMU data can be used: in the monitoring and characterisation of high renewable penetration feeders and generators. The main premise behind this is that the voltage phase angle is a uniquely informative state variable for power distribution systems. Changes in phase angles at the distribution network level are much smaller than the changes at transmission level. µPMUs either store their collected data internally or communicate it to a utility or analysis station. Using an application such as the Advanced Research Projects Agency – Exchange (ARPA-E), informational data that is more 'human readable' can be provided to operators. It must then be integrated with the communications system and other sensor data, before travelling on to distribution planning and operations tools. Distribution planning and operating tools need to process large volumes of data (measurement points), and potentially run steady state and dynamic analyses with variable grid conditions. Grid stability can often depend upon these advanced simulations. Another major issue of integrating µPMU data with distribution network operations is that the naming of components is often unsynchronised between departments and companies. For example, substation names may not be identical across applications and therefore time must be spent to match them in order to connect the entire model. The paper concludes that there are many benefits of load and state estimation in distribution. These could be realised through improved measurement and simulation integration (including improved reliability), power restoration, less constraints, and quicker adaptation of the grid to changing and potentially new conditions.

Reference [27] focuses on using data from Advanced Metering Infrastructure (AMI) for real time monitoring and control of a distribution system. It uses the branch current based state estimation (BCSE) method, similar to that used in [26], based on a WLS approach. This uses branch currents instead of node voltages to estimate the state of the system. The paper was published in 2009, and although it is 8 years old the conclusion that AMI measurements can provide a fairly accurate state estimation of a distribution system is still relevant.

Reference [28] describes the development of a Hardware-in-the-Loop (HIL) test platform for the performance assessment of a PMU-based Real-Time State Estimator (RTSE) for Active Distribution Networks (ADN). The Discrete Kalman Filter (DKF) process is used for state estimation. The work has been completed on a simulation, with virtual PMUs, after designing a HIL setup with a real ADN. The performance was assessed in terms of RTSE accuracy and time latency. State estimation results showed accuracy of 0.001 %, and the process was completed in less than 55 ms. These results show that it is possible to accurately track the state of ADNs using PMUs and RTSE.

According to [29], Active Network Management (ANM) is an effective approach to release more distribution network capacity for connecting renewable generation without expensive network reinforcements. To investigate this further, Scottish Power Energy Networks (SPEN) have carried out a pilot project using the Psymetrix ANM approach, focusing on wind farm connections on a 33 kV distribution network on Isle of Anglesey, North Wales. This method is Angle Constraint Active Management (ACAM); where a renewable generation is constrained based on voltage angle difference signals produced by PMUs. A set of angle constraints are derived using offline network simulations and then applied to the real-world network. Data collected by PMUs on the network is sent to PhasorPoint (a Psymetrix platform used for phasor-based network monitoring, in this case being housed in a cloud and accessed via a 3G mobile network) before being downloaded and analysed offline. The angle difference threshold between buses is found, and the calculated real time angle difference must be kept below the threshold to maximise wind farm output.

2.5 Transmission

An introduction to the use of PMUs in the Czech Republic is documented in [30]. It was written in 2009, after PMUs were connected on the power networks across 128 switching stations (475 nodes) in an attempt to improve power systems performance and reliability, and investigates the possibility of fault localisation. The data used is from a fault in July 2006 (when no PMUs were connected). For advanced fault localisation, an *H* matrix with Blondel constraints is used. This allows more inclusion of elements in a line, thus creating a network model much closer to the physical one. This paper looks into the close agreement between frequency measurements at low and high voltage levels during a fault in the transmission network. It also suggests that these events will be visible in low voltage measurements on the distribution system.

2.6 Integration

A study of integrating AMI measurements with the grid infrastructure is presented in [31]. It focuses on decentralised strategies of integration and delivering meter readings to the appropriate substation. The meter readings must be accurately and efficiently delivered, preferably in a way that allows two-way communication between the substation and the network. Although this paper focuses more on smart meters than PMU integration, it does show that it is possible to gain accurate meter readings without sampling each node in the network.

2.7 Practical Work

Reference [32] shares the experience of the deployment and tuning of real-time advanced applications for distribution state estimation at the BC Hydro control centre. The project is not entirely relevant as it does not use PMUs for input, instead load profile pseudo measurements are used. The state estimator and load flow are tuned to ensure the robustness and consistency of the results. Load flow is tuned first, as it identifies higher-level errors that will benefit both applications. The state estimator is tuned by identifying inconsistencies then analysing the solution until convergence. Many iterations are required.

A real-time three-phase state estimator for distribution networks has been deployed on the École Polytechnique Fédérale de Lausanne (EPFL) campus in Switzerland [33]. The 20 kV distribution network is comprised mainly of short lines, and has a variable load with active power injections. The inputs to the state estimation are PMU measurements sent via a PDC, and the results have an accuracy range of a few percent. This is potentially the first practical real-time state estimator for distribution-networks. PMUs are connected to the medium-voltage side of transformers, recording voltage and current waveforms. A low-latency PDC was developed on LabVIEW. This takes care of decapsulation, time-alignment and the replacement of missing measurements in order to feed the state estimator with a consistent and complete set of data. The PDC communicates with the PMUs to decapsulate their data in line with IEEE C37.118.2 [23]. A subset of the measurements is sent to the state estimator with minimum time latency. An algorithm was developed to determine how long each set has to wait until it is complete, with the timeout expiring when a measurement with a later timestamp turns up. A dedicated communications network was built to improve the security of the project.

Two different SE methods were trialled in this project: WLS and DKF. DKF provided more accurate estimations and was therefore the chosen method. The overall accuracy obtained was in the range of a few percent. The authors concluded that the performance of the developed system appears to be compatible for coupling with real-time protection and control functionalities that are expected to be developed for ADNs.

2.8 Optimal PMU Location

Placement of PMUs in real applications is dictated less by maximising observability and more by physical constraints such as planned outage and maintenance schedules, availability of fibre infrastructure, and the number of available channels per device. For full observation of a distribution system using PMUs, each node of the network must have a PMU, or a neighbouring PMU that can obtain its measurements and those of adjacent devices. A bus is directly observable if it has a

PMU placed at it, and the voltage magnitude and angle are measured. A calculated bus is one that is observable by neighbouring PMUs. In a completely observable system every bus is either calculated or observed [34].

2.8.1 Methods

- The Immunity Genetic Algorithm (IGA) is based upon the evolution of life as it adapts to its changing environment [35]
- Binary Search Algorithm considers all possible combinations of PMU locations and uses a binary search formulation to narrow them down to the optimal solution; if multiple options are available the optimal solution is the one with the highest redundancy [35]
- Binary Particle Swarm Optimisation (BPSO) is based upon a 'swarm' with members that cooperate with each other to find their optimal location, where positions are updated continuously and compared with previous states until a solution is found [35]
- In Ant Colony Optimisation (ACO) a set of 'ants' work together to find the optimal solution, moving through adjacent states building paths and dropping 'pheromones' on benches to let other ants know where they have already explored [36]
- Greedy Algorithm makes decisions according to a single rule: choose the node that would allow the PMU to access the largest number of nearby nodes (those without any current coverage by their own PMU or one a single node away) [36]
- Integer Linear Programming (ILP) uses an integer-based formulation to find the optimal solution, using the Bus Observability Index (BOX) and the Summation of Redundancy Index (SORI) [35]
- The Spanning Tree Approach makes use of the depth of observability of nodes by PMUs. It starts at the 'root' of the network and branches off at nodes as it grows [35]
- Matrix Manipulation was designed as a basic exhaustive algorithm that manipulates the connectivity matrix and compares past and present conditions [37]

2.8.2 Studies

A variety of optimal placement techniques are reviewed in [35]. The top three best performing algorithms are found to be IGA, BPSO and ILP. ILP was proven to be the most effective on a larger power system.

References [36] and [38] study the placement problem of PMUs in a distribution system with relation to system reconfiguration. They focus on the ACO algorithm and the Greedy algorithm. Reconfiguring a distribution network means changing its topology by changing the status of switches (open/closed) to transfer loads amongst feeders. ACO and Greedy algorithms are combined to see how reconfiguration affects the optimal location of PMUs. The result is that network reconfiguration does affect the optimal placement of PMUs, and that ideally they should be installed with this in mind.

Work done in [39] improves upon the BPSO method. The theory is that reducing the search space for a particle will improve the convergence time. This is done by excluding radial buses from the solution and pre-assigning PMUs to the neighbours of radial buses. It means that the neighbour, the radial bus and the neighbour’s neighbour will all have observance without wasting any PMUs. The improvement was implemented on the IEEE 30 bus system. It was found that IBPSO (Improved BPSO) converged faster than BPSO by 26 iterations. The SORI was higher for IBPSO, indicating a higher quality of PMU placement.

An evaluation of some optimisation algorithms on their ability to handle different sized networks is presented in [37]. The outcome was similar to that found in [35], showing that BPSO and ILP are suitable for all network sizes, but that others may be less suited to larger systems. Matrix Manipulation has an inability to handle larger networks due to its long calculation times.

The Binary Search Algorithm is used to place PMUs in [34], prior to running a fault location and diagnosis scheme. The aim is to see how accurately various faults can be detected with the optimal placement of the minimum number of PMUs. The proposed scheme gave results of 98 % accuracy over eight different test faults.

Using Greedy and Integer programming algorithms, the optimal placement for PMUs for full observability of a three-phase state estimation is calculated [40]. The Greedy algorithm is used first to find the optimal PMU deployment, and then an integer programming method reformulates the solution into an optimisation problem. The results are positive. The state estimation performance of the optimised method approaches the global optimal solution and becomes closer the more PMUs are installed.

2.9 Power System Analysis in the Cloud

Cloud environments are already used some in power systems applications. The main issues that surround the cloud are security, scalability and optimisation.

Trusted cloud computing can remove uncertainties regarding security. This is addressed in [41] with a focus on intrusion detection and infrastructure security for applications in the energy sector. Implementing a trusted cloud environment gives a measurement of the reliability of the components, including hardware, firmware and the operating system. Cloud administrators monitor the organisation and integrity of their cloud, while end-users must access it via a secure channel.

A different approach to cloud security is taken in [42], where the aim is to obfuscate information about contingency analysis. Disguising the presence of a contingency violation when using a cloud environment run by a third-party attempts to stop the system being exposed and potentially attacked. Obfuscation must be chosen carefully to avoid introducing an error into the contingency analysis.

The proposition of [43] is a cost-orientated optimisation model to support the development of demand side management (DSM) in smart grids. DSM requires a large amount of information to be processed quickly, which requires a large amount of computing power. Using cloud-based computing resources will allow this to happen regardless of the fluctuation in demand due to their scalability. The potential parallelisation of tasks and on-demand availability are highlighted as benefits of using cloud computing.

Parallelisation of the analysis of power systems in a cloud is discussed further in [44], stemming from computing and resource scalability. A parallel algorithm for power flow calculation using Map-Reduce framework scaled well and achieved fast results. This paper is focused more on the power flow algorithm than the cloud environment, but it is still relevant as it proves the use of clouds in power system analysis.

The uptake of using cloud computing in the power industry prompted a study of distributed load balancing algorithms [45]. The size and complexity of load balancing on a network renders centralised job assignment infeasible. This requires an optimised system that allows the system state to be inferred using locally gathered data so that it can be scaled as necessary within a cloud environment.

A PMU-based state-estimation application is mapped to a cloud architecture to support smart grid analysis and data storage [46]. Security of the cloud environment is overcome via encryption of data and a public key infrastructure. The key contributions include flexible scaling based on dynamically changing the number of PMUs, economy of computation using the maximum cores and bandwidth available in the cloud and synchronised data processing of PMU data. The approach to data storage is to cache data in memory on arrival and move it to the volume storage in optimally sized amounts.

Reference [47] discusses high performance computing (HPC) applications to help deal with the increasing complexity of modelling the power grid. HPC benefits from parallelisation across multi-core processors. 17 papers are included in the journal; they examine HPC for transmission applications, distribution applications and algorithm and framework development. HPC is similar to cloud computing in its use of parallelisation and machines with multiple cores. The applications and use cases for HPC discussed in [47] will potentially be able to be run in a cloud environment, saving individual investment in physical machines by instead accessing them as-needed from a cloud provider.

2.10 Concluding Remarks

This chapter has summarised relevant research that relates to the subjects covered in the remainder of the thesis. It reviewed current and recent papers relating to the use of PMUs and µPMUs in both transmission and distribution networks, as well as the optimal placement of these devices. It also discussed the use of cloud computing and parallelisation to improve power system analysis.

Chapter 3 Standards

3.1 Introduction

There are many overlapping standards within the power industry that support communication and data exchange between processing systems. These standards have a variety of often incompatible naming and identification rules. This can cause devices and assets to have different names and identities within models and systems that represent the same thing but are operating at different levels of the network. This chapter will look at the current state of power system standards and how they are used: from the exchange of high-level network planning models; to device communications including PMUs and smart meter readings; how they can be linked to one another and the benefits this will provide to both real-time operation and offline network analysis.

There are a number of advantages in allowing different systems and applications to automatically understand that data is coming from the same device and the same point in the network, even if it originates from different source systems or from sensors conforming to different, incompatible standards. There are a number of challenges that must still be addressed to achieve this, including the requirement to automatically correlate common identifiers, and convert between different data models and serialisation formats. This chapter will look at ways in which these challenges can be addressed to promote integration across multiple levels of the power system, with the aim of providing a more detailed, accurate view of the system, improving the quality of data exchanged, and making it easier to share network information both within a company and with external parties.

This chapter will discuss the standards used in Chapters 4 - 7 within the context of IEC 62357-1, which can be seen in Figure 3.1. This is a reference architecture for power system information exchange that is intended to aid the visualisation and understanding of where different standards and protocols apply within the power system and how they relate to each other, as well as identifying the boundaries between them and where interoperability is required [48]. This chapter will focus on a subset of this standard in conjunction with de facto standards related to it.

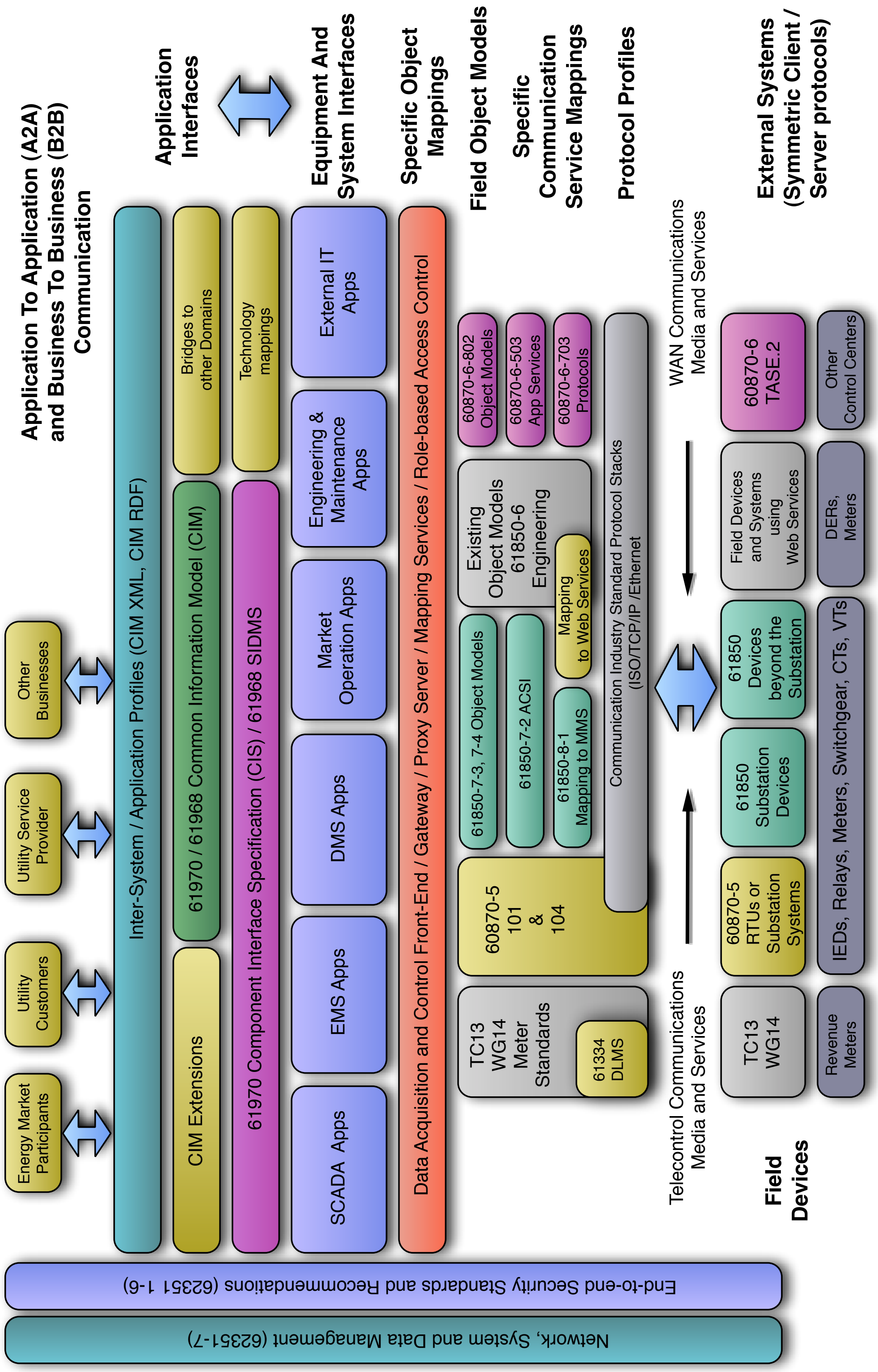

Application To Application (A2A) and Business To Business (B2B) Communication
Energy Market Participants
Utility Customers
Utility Service Provider
Other Businesses
Application Interfaces
Inter-System / Application Profiles (CIM XML, CIM RDF)
CIM Extensions
61970 / 61968 Common Information Model (CIM)
Bridges to other Domains
61970 Component Interface Specification (CIS) / 61968 SIDMS
Technology mappings
Equipment And System Interfaces
SCADA Apps
EMS Apps
DMS Apps
Market Operation Apps
Engineering & Maintenance Apps
External IT Apps
Specific Object Mappings
Data Acquisition and Control Front-End / Gateway / Proxy Server / Mapping Services / Role-based Access Control
Field Object Models
TC13 WG14 Meter Standards
61334 DLMS
60870-5 101 & 104
61850-7-3, 7-4 Object Models
Existing Object Models 61850-6 Engineering
60870-6-802 Object Models
Specific Communication Service Mappings
61850-7-2 ACSI
60870-6-503 App Services
61850-8-1 Mapping to MMS
Mapping to Web Services
60870-6-703 Protocols
Protocol Profiles
Communication Industry Standard Protocol Stacks (ISO/TCP/IP /Ethernet
Telecontrol Communications Media and Services
WAN Communications Media and Services
External Systems (Symmetric Client / Server protocols)
TC13 WG14
60870-5 RTUs or Substation Systems
61850 Substation Devices
61850 Devices beyond the Substation
Field Devices and Systems using Web Services
60870-6 TASE.2
Field Devices
Revenue Meters
IEDs, Relays, Meters, Switchgear, CTs, VTs
DERs, Meters
Other Control Centers
End-to-end Security Standards and Recommendations (62351 1-6)
Network, System and Data Management (62351-7)

Figure 3.1: IEC 62357-1: IEC Technical Committee 57 Reference Architecture for Power Systems Information Exchange [48]

3.2 Low Frequency Exchanges

For the purposes of this research, low frequency communications are considered to be exchanges of data that occur outside of real-time. This can be at intervals of minutes or hours, at regular intervals throughout the day, or irregular exchanges on a weekly or monthly basis in a manual, ad-hoc fashion. These exchanges can take place either through manual user interaction, by automated triggering within a system, or a combination of both.

3.2.1 Network Data

The exchange of network planning models is vital to network companies for short-term forecasting and for longer term network extension planning. These exchanges happen both internally and externally to the organisation. DNOs are required to balance their electricity network continuously on a real-time basis, meaning that they must converse regularly with TSOs.

3.2.2 Network Planning

Standards exist for network model exchanges between planners that use bus-branch models to represent an equivalent model of the underlying network that can be derived from the detailed node-breaker model. These exchanges are often made using proprietary formats internal to the planning applications being used, on an ad-hoc basis, and are usually carried out manually between engineers.

Engineers use bus-branch models for short and long-term planning of the network. It is not uncommon for these planning models to be maintained separately from the detailed node-breaker models used for network operations. This can be due to a combination of cultural and technical issues: departments within any large organisation have a tendency to operate within 'silos' and have no mechanism for regularly distributing and aligning their models to those used by other departments. When different applications and systems are being used for different purposes it is not uncommon for them to use incompatible data formats. Any data conversion into another bus-branch format or back into a detailed node-breaker format requires significant manual effort that cannot be justified on a regular basis. As such it is now common for planning and operations models to be maintained separately, with irregular manual effort required to ensure they are aligned.

3.2.3 Load Congestion Forecasting & Network Conditions

TSOs and DNOs have an increasing need to communicate information about the network in real time. In the past the TSO could treat the DNO network as a predictable load. Generation was connected at the transmission level and any changes to the distribution network were unlikely to have an impact on the overall system behaviour. As more generation is being added at the distribution level, and sophisticated network management schemes are being implemented, the TSO needs greater visibility of the configuration and state of the distribution network. This means that DNOs need to exchange the current status of their network, such as open switches or offline equipment, as any changes occur, as well as exchange of information about planned outages or network updates.

Information about planned outages would be provided days, weeks or months in advance and the requirements for the level of detail, structure and format of the data may be agreed between the parties or set by regulatory organisations such as Ofgem (Office of gas and electricity markets) in the UK or NERC (North American Electric Reliability Corporation) in North America. The network operators may independently choose to agree on exchanges that exceed those required by the regulator to aid their network operations.

These planned outages and updates to network configurations may be exchanged as changes to the network models using machine-readable formats like the Common Information Model (CIM) [49], or as files native to the proprietary software used by both companies. Short-term changes to expected generation or changes to switch statuses may often be communicated person-to-person, using email or phone conversations. Such manual exchanges very quickly run into scalability issues as the size and scope of the network data exposed to other network operators increases.

3.3 Standards to Support Network Model Exchange

3.3.1 CIM IEC 61970-452/456/IEC 61968-13

CIM is used to share network configuration models and their status between network operators including between TSOs within a country; between TSOs and DNOs; and between SOs in interconnected countries/areas. IEC 61970-452/456 [50] covers transmission network model exchange including equipment data, the starting conditions for network analysis and the resulting solution from applications such as power flow or state estimation. IEC 61968-13 [51] deals with distribution network model exchange by extending the network data defined in IEC 61970-452/456 with support for unbalanced, low voltage networks, network asset information and geographical data. This data is subsequently used with the standard serialisation format, CIM RDF XML [52], covered by IEC 61970-552 [53], to create machine-readable files that define complete network models in a standard data model in an open, standard file format.

The CIM network model started as a data model for describing node-breaker network models used by real-time control systems, primarily the Energy Management System (EMS). Since this work began under the Electric Power Research Institute's (EPRI) Control Center Application Programming Interface (CCAPI) project in 1996 [54] it has been expanded to cover different application use-cases for transmission, distribution and market operations. These extensions include the addition of a package to define the computed bus-branch model (known as the 'Topology Model' in CIM) that then has a direct reference to the detailed underlying node-breaker model from which it was derived. This enables the simultaneous exchange of node-breaker and bus-branch models.

The CIM is currently being used by ENTSO-E (European Network of Transmission System Operators for Electricity) for its Common Grid Model Exchange Standard (CGMES) that is used by European TSOs to support Day-Ahead Congestion Forecasting (DACF). This has included a number of interoperability tests [55] with support from major vendors.

3.3.2 IEC 61850

IEC 61850 is a collection of international standards that describe the devices in an electrical substation and how they exchange information. One of the main features of IEC 61850 is the Substation Configuration Language (SCL) that defines substation topology, any logical or protection devices and the data network configuration. A single line diagram is created to define the internal substation topology then descriptions of functions and IED (Intelligent Electronic Device) parameters are added to the devices. IEC 61850 is considered to be an important group of standards as it removes inconsistencies in system structure and requirements, which enables interoperability between applications both within the substation and in external systems.

3.3.3 IEEE Common Data Format

Published in 1973, the IEEE Common Data Format (CDF) is a bus-branch model for network exchanges of solved load flow data. Data is stored in a text file in sections of lines containing the data items in specific columns [56]. IEEE CDF is typical of early power system planning formats with a fixed-column text-based format that can be easily created and edited in a text editor. Its simple structure and format keeps the file sizes small. This was a critical concern in 1973 as data storage costs were several orders of magnitude more than they are today.

The format is a simple bus-branch format that was suited to early power-flow applications. These required a description of the basic components that were used to populate the matrices in a Newton-Raphson [17], [57] power-flow: buses and branches. The IEEE CDF was found to be limited as the complexity of power-flow applications expanded, and it lacked the detail and extensibility to allow it to be used for detailed network modelling. It is no longer in regular use.

3.3.4 UCTE DEF

The UCTE (Union for the Co-ordination of Transmission of Electricity) Data Exchange Format (DEF) is a text format that was developed to aid the communication of load flow data, three-phase short circuit studies and the interconnected high voltage network. UCTE was an association of TSOs in Continental Europe that operated from

1951 to 2009 before its operational tasks were transferred to ENTSO-E. The UCTE DEF format enabled a standardised exchange of network data among the 29 member TSOs. The first version was released in 2003, but when ENTSO-E took over the operational tasks from UCTE in 2008 it was decided that the UCTE DEF format was not suited for the increasing requirements for data exchange among the 43 TSOs that are part of ENTSO-E. It was agreed that using CIM would be more suitable than extending and enhancing UCTE DEF.

3.3.5 Proprietary Formats

Vendor formats, such as PSS/E RAW from Siemens or PSLF EPC from General Electric, are often considered as de facto standards. This is because the formats are fixed width or white-space delimited and thus relatively easy to interpret. They are widely supported by many other vendors, and are both text based formats dealing with a three-phase balanced equivalent bus-branch model that can be viewed or edited in a simple text editor.

Many vendors support the import and export of network data in RAW or EPC within their applications, but they are not open standards and are maintained and defined by the vendor to meet the requirements of their specific tools. Although popular and widely used, the supporting documentation is owned and licensed by the vendor and therefore may require a license or agreement for third party use.

3.4 Electricity Markets

3.4.1 Network Data

Electricity markets also require network data with Market Management Systems (MMS) requiring information about the real-time state of the network from the control system. It also requires real-time data about current and expected load on the network from load-forecast models and data provided by market participants.

3.4.2 Market Operations

Generators communicate with TSOs daily, or more frequently if required, to provide details of their expected generation along with any information they have about network availability such as maintenance or faults. Consumers (who take the form of retail providers in the GB market), generation providers and the SO communicate in advance, with final decisions for each 30-minute block of energy made 1 hour prior to real-time. As the SO knows the expected generation, and consumers can provide the expected demand, an arrangement will be made in terms of supplied electricity and its cost to ensure the system is balanced.

Organisations such as ENTSO-E and BETTA (British Electricity Trading Transmission Arrangements) exist to oversee the electricity markets within their member states. They ensure the fair exchange of generation and aim to improve the coordination between countries. They are also involved in creating a fair and competitive environment with respect to bids and offers submitted by generators and TSOs for the payment and trade of electricity.

There is a drive towards the integration of electricity networks and markets across Europe [58]. To realise this, it is critical that there are accurate, automated exchanges of both network and market data between countries. Many bordering or interconnected countries have different specifications for their market structures, network data and naming conventions that can all hinder communications.

3.5 Standards to Support Markets

3.5.1 IEC 62325

IEC 62325 is the CIM for Market standard that shares market data between countries, geographical areas, generators/consumers and TSOs. IEC 62325 extends the CIM to cover exchanges of data across different stages of settlement in an electricity market, with a number of classes defining the data exchanged for bidding, clearing and settlement. It involves the expected energy generated, pricing nodes and the expected energy required for a given period of time. This includes bids and offers from participants, to be shared and analysed by a coordinating body. Previous data showing past statistics and trends are also shared to improve future network operations.

3.5.2 IEC 61970

The IEC 61970 CIM standard, as discussed previously, defines the electrical network model at both node-breaker and bus-branch level. Crucially the combined IEC 61970/62325 data model has direct references between the market data elements and the electrical network elements. It defines associations between Generators in the market system and the corresponding Generating Units in the electrical network model. Similarly for nodal-markets there is a direct link between an Electrical Connectivity Node and the Market Node. This enables a direct correlation between the operational network and data from market participants.

3.6 High Frequency Exchanges

High frequency communications occur at intervals of several seconds, or more commonly at millisecond or microsecond intervals, so they must be automated. These exchanges are usually between devices in the power network or between a device and a DNO or TSO.

3.6.1 Device Communication

Automated devices on a network communicate their status on a regular basis using protocols like ICCP (Inter-Control Centre Communications Protocol) [59] or IEC 61850. This allows the devices to convey status updates about themselves and parts of the surrounding network, as well as information about the safety and security of the network. Older devices that are not using the same network models will be communicating using similar protocols to ensure that regular updates are delivered. Devices that can be included in these communications are relays, transducers, transformers, PMUs, substation computers and a network master station.

SCADA (Supervisory Control and Data Acquisition) is a control system architecture that enables the collection of real-time data as well as issuing process commands. It is used as a control system in many environments, not just the power industry [60]. Control systems that collect SCADA data use the information gathered to evaluate the status of the devices sending it and the network surrounding them.

ICCP is a protocol used to share data in real time. Data is exchanged among utilities and control centres using wide area networks. These interconnections enable regulated data communications across international borders and between TSOs, using a client/server principle where a server may create associations with multiple clients at the same time.

3.6.2 Phasor Measurement Units

PMUs are high frequency devices that measure waveforms, typically at a rate of 48 samples per 50/60 Hz cycle. These measurements commonly consist of voltage and current data for three phases, a GPS time stamp and any other data that is deemed necessary in the current environment. μPMUs are smaller, cheaper versions of PMUs that are designed for installation on the distribution network. They support cellular data communications but have the ability to send data at a similar rate to the PMUs installed at transmission level. Newer designs have seen sample rates of up to 512 samples per 50/60 Hz cycle [61].

3.6.3 Smart Meters

Smart meters collect and send data detailing the energy usage of consumers, replacing traditional mechanical meters with digital meters that include automated communication amongst electrical network operators or retail providers. This can occur regularly for a specified period of time. Implementations can involve exchanges of usage data every 5 minutes, 15 minutes, hourly, daily, or data can be averaged and sent monthly. The time period involved varies depending on the country and its regulations, and how the data is used. For example, in the US the data flow is often bi-directional and can be used to determine real-time pricing dependant on the weather, or to enable real-time control of consumer appliances. For example, during periods of high temperature there is a greater demand for electricity to supply air conditioning, which can result in prices peaking during the hottest time of the day. The bi-directional control functionality can allow operators to ‘flatten’ this load by cycling air conditioning units across the network in a ‘round robin’ schedule, resulting in the total load being reduced and avoiding any ‘spikes’.

3.7 Standards to Support High Frequency

3.7.1 IEC 61850

The IEC 61850 group of standards aims to standardise the definition of network substation models including the description of the devices within the substation and their configuration. It is a relatively new protocol that has quickly been adopted in Europe but is still uncommon in North America. IEC 61850 covers many specification, documentation and development issues with one standard, including the ability of devices to automatically describe themselves. This makes it highly valuable if implemented correctly as control systems can automatically build up a description of a substation’s internal workings by querying each device about its characteristics and functionality. Alongside configuration standards, IEC 61850 also includes the use of GOOSE (Generic Object Orientated Substation Events) messages, which are sent via Ethernet packets to enable the devices in a network to communicate. These messages are low latency with the standard requiring each data set to be transmitted within a time period of 4 milliseconds [62].

3.7.2 DNP3

The Distributed Network Protocol (DNP3) is attempting to optimise the interoperability between substation computers and master stations, including Remote Terminal Units (RTUs) and IEDs. It was developed to aid the transmission of control commands and data between computers and SCADA applications using serial and IP communications. DNP3 can also be used on fibre optic, cellular and radio systems.

Based on IEC standards, DNP3 has been designed as an open and public protocol, which is owned and administered by the DNP3 Users Group. As it is an open standard, products that support DNP3 can be acquired and used anywhere in the world. This means that a utility can use it with a master station from one vendor and computing equipment from another and they will be able to communicate.

3.7.3 IEEE C37.118.1/2 (PDCs/PMUs)

PMUs sample analogue current and voltage data at their point of connection on the network in the form of synchrophasors, which all have a common time base taken from a GPS timestamp. PMU data is collected by PDCs and sorted according to the timestamp, before being forwarded to another application such as a data historian or control system. It is sent in the form of data frames, starting with a SYNC word (2-bytes for synchronisation and frame identification), a FRAMESIZE word (2-bytes), an IDCODE (2-bytes identifying the unit sending the message) and a time-stamp (8-bytes). The data to be transmitted follows, in 2 and 4-byte words, and the frame is finished with a 2-byte check word (CHK) [63].

PMUs send data to PDCs, which then communicate with other PDCs and the main control system. PDC data is used with SCADA and DNP3 to provide wide area monitoring data for the entire network.

3.7.4 Smart Meters (IEC 61968-9)

IEC 61968-9 is the Interface Standard for Meter Reading and Control. It contains a number of standard interfaces for reading and control of smart meters, as well as communication between Meter Data Management applications and other enterprise systems. The main areas monitored by smart meters are the collection of device statuses and readings, the transmission of these to a data management system, the transmission of power system reliability, event data and the communication of the network health information to the relevant systems [64].

3.8 Standard Interoperability

One issue within a power network is different standards being used for the same functions due to application or system incompatibilities, or the installation of new equipment and systems alongside legacy applications or equipment that cannot support modern standards. This creates challenges for interoperability due to format or protocol incompatibilities, or differences in naming conventions or identities for equipment and data elements.

3.8.1 Competing Standards

3.8.1.1 Network Model Exchanges

Bus-branch network models such as IEEE CDF and UCTE DEF, as well as proprietary formats such as PSS/E RAW and PSLF EPC, are still widely used even with a standardised format such as CIM XML providing an alternative. There are a number of reasons for this including: the complexity of CIM XML compared with white-space delimited text-based formats makes it difficult for users to view or edit outside of an application; vendors not supporting CIM XML; and existing processes and work flows being dependent on a proprietary format.

Where utilities are implementing new processes or systems, proprietary formats are seeing competition from the CIM standard. This occurs as users seek to break any vendor “lock in” and move to open standards that enable them to use best-of-breed software. The support for bus-branch models within CIM, using its Topology classes, allows it to support the operational models used by EMS and other control systems as well as planning applications such as PSS/E and PSLF.

The challenges of moving between CIM and proprietary formats are often due to the proprietary formats having restricting naming and identity rules compared with what is allowed in CIM. This requires either the exchanging parties to restrict the names and identities used in CIM to ensure they are compatible with legacy formats, or for internal registries being used to track where alternative names and identities have to be generated during an export.

3.8.2 Device Level Communication

IEC 61850 and DNP3 cover similar areas of the power network as they are both attempting to standardise communication amongst devices in a network. However, they have differing structures and formats that cause them to be in competition with each other. DNP3 is an older standard that is already widely used and communicates over low-bandwidth channels, whereas IEC 61850 is newer, less well known and primarily uses higher bandwidth, low latency Ethernet based communication for its GOOSE messaging. Until IEC 61850 becomes the protocol of choice for all utility companies, translation systems must be used to convert data between it and DNP3.

3.8.3 Overlapping Standards

3.8.3.1 Network Modelling

IEEE CDF and UCTE DEF are two open standards that overlap with CIM. They, like PSS/E and PSLF, are bus-branch models while CIM is a node-breaker model with the option of bus-branch. There is a requirement to generate bus-branch representations of node-breaker models (e.g. for power flow) so they need to be interoperable. Interoperability from CIM into these bus-branch formats has been demonstrated previously [65].

IEC 61850 SCL overlaps with low frequency network configuration model communications. There are similarities between the models, and integration between the two would be beneficial to allow the automatic generation of CIM substation electrical network models from SCL.

3.8.3.2 CIM Network & CIM Markets

These standards have an obvious overlap as they both use CIM. They should be integrated automatically when both parties are using CIM for their data exchanges and the common, integrated model shows how two domains can be shared.

3.8.3.3 IEC 61850 SCL & CIM

Although IEC 61850 SCL and CIM both cover aspects of network modelling, primarily the former which describes substation configuration and internal connectivity, they are not replacements for each other. It would be beneficial to be able to integrate data from each standard to have a more inclusive view of the network. This would allow the auto-discovery supported by IEC 61850 to enable a system to build the internal substation electrical network in a CIM format. It would ensure that the substation and network control systems were kept synchronised and that any changes to the internal substation configuration would automatically be reflected in a CIM model that is then used by the control, markets and planning systems.

3.8.3.4 IEC 61850 & PMU

These standards overlap where PMU and PDC data is exchanged as IEC 61850 messages. Integration between CIM, IEC 61850 and PDC layers would allow automatic navigation from a PMU to its place on the network via the PDC data.

3.8.4 Future Integration

3.8.4.1 CIM Network & PMU

Integration of CIM network data and PMU data would allow users to see exactly where the PMUs are in the wider network. This differs from the previously mentioned overlap of IEC 61850 and PMU data by removing the need for PDCs. It would allow PMUs to have knowledge about their own location on the network. This integration would also allow PMUs to know their relationship with neighbouring components and would provide consumers with a greater overall view of the system.

3.8.4.2 CIM Markets & Smart Meters

Integration of CIM market data and smart meter data in close to real-time is a pre-requisite for introducing true real-time pricing at the consumer level. This could be especially effective if it is combined with congestion forecasting and real-time control, incorporating the use of bi-directional smart meters.

3.9 Concluding Remarks

This chapter has looked at some of the challenges surrounding data exchange in the power industry, including the use of incompatible data structures, identification rules and communication protocols. Overcoming these challenges will enable the integration of data from different areas of the power network. There are numerous benefits for both real-time operation and offline network analysis. These potential benefits include: enabling different systems and applications to automatically recognise that data is coming from the same device at the same point in the network, but originating from different sources and conforming to different, incompatible standards; the ability to integrate real-time data with offline analysis tools; and utilising smart-meter data to enable true real-time pricing for electricity markets.

Non-interoperable standards complicate integration across multiple levels of the power system and there are a number of challenges that must still be met to achieve this. One of the main challenges is the need to automatically match common identifiers and convert between different data models and serialisation formats. The aim of this is to provide a more detailed and accurate view of the power network by improving the quality of data exchanged and making it easier to share network information. Ultimately this will enable utilities to assess an issue more quickly within the power system, and it will allow them to discover and understand what the problem is and where it has occurred. The overall result will be a faster solution to any issues that arise and a more reliable power system for consumers and operators alike.

Chapter 4 Networks

4.1 Introduction

The changing networks and increased DER mentioned in Chapter 1 are creating unprecedented challenges for network owners and system operators. There is an ever-increasing need for improved visibility of both the distribution and transmission networks, but this is an issue that not all system operators have realised yet. There is a lack of monitoring capabilities on many distribution networks at the lower voltage levels, which could lead to difficulties in forward planning, network control and fault management.

This chapter will look at the similarities and differences between transmission and distribution networks, the main challenges faced by Transmission Owners (TO) when observing large distribution networks, and the challenges for installing µPMUs in resource constrained areas.

4.2 Transmission Networks

4.2.1 Transmission WAMS

Wide Area Monitoring Systems (WAMS) incorporate wide area synchronised phasor measurements, which are produced at a rate of 50 frames per second. These provide unparalleled monitoring of the dynamic behaviour of larger electrical systems, especially when compared to SCADA data that is unsynchronised and only sampled at 1 frame per second. VISOR [66], [67] is a Network Innovation Competition (NIC) project being run on the GB network. It aims to show how an adjustment in power system monitoring can be used to lower operational and capital expenditure. This is realised by maximising asset consumption and increasing resilience against events that are low probability but high impact and can cause network disruption, equipment damage or blackouts.

This objective is being achieved through the novel use of monitoring, analysis and visualisation techniques that give a better understanding of the full potential of the power system. These techniques also monitor how close the system is to being operated at maximum capacity in real-time. Improved understanding enables WAMS to give operators and planners more confidence to use the full capability of the power system when facing new challenges. Network security is improved by using real-time measurements to identify threats in advance. Although WAMS offers a wide range of benefits to power system operators and planners, VISOR is focused on four key areas that have been predicted to be the most advantageous to the GB system in the short to medium term:

- Real-time monitoring of sub-synchronous oscillations between 0.002 Hz and 46 Hz
- Dynamic model validation using post mortem analysis of WAMS data
- Hybrid state estimation
- Potential use of angle-based security limits to increase power flow on the B6 boundary between Scotland and England [66]

4.2.2 Current GB WAMS

WAMS is currently being used on the GB system to improve transfer capabilities and observe interactions and their subsequent oscillations.

The existing GB transmission WAMS includes wide area monitoring data from three GB TOs - National Grid (NG), SPEN and Scottish Hydro Electric (SHE) - and the GB system operator. This integrated WAMS concentrates on three main areas. Each area incorporates planning, real-time and event/trend analysis [2]:

- Management of system risks and events: early warning, response and analysis
- Reducing uncertainty: improved situational awareness, confidence in system models & limits
- Maximising assets: efficient and effective use of WAMS and transmission infrastructure

The critical infrastructure of the GB transmission is shown in Figure 4.1 [66]. It is comprised of three regional WAMS 'Datacentres', a central WAMS 'Data Hub' server and an arrangement of Waveform Measurement Units (WMUs) and PMUs.

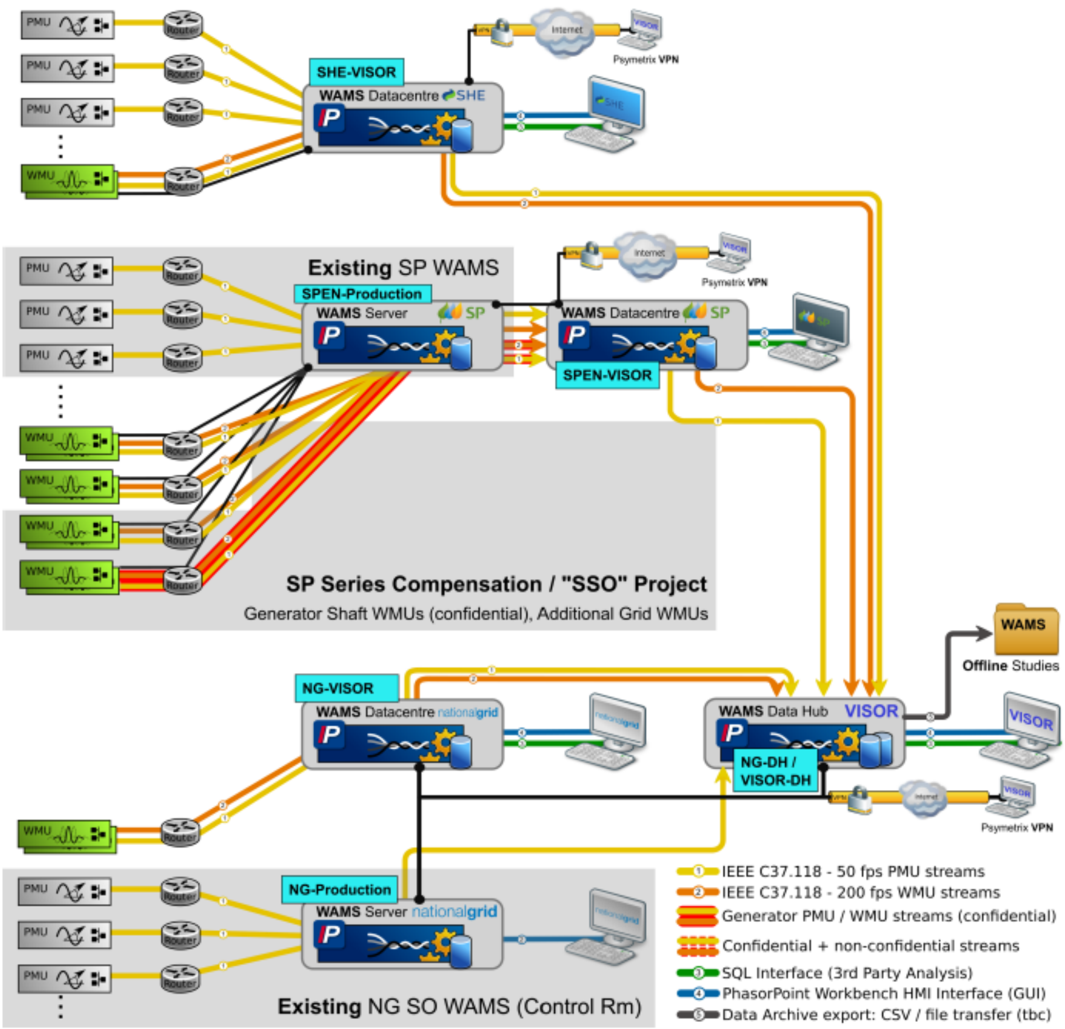

PMU
Router
WMU
SHE-VISOR
WAMS Datacentre SHE
Internet
VPN
VISOR
Psymetrix VPN
Existing SP WAMS
SPEN-Production
WAMS Server SP
WAMS Datacentre SP
SPEN-VISOR
SP Series Compensation / "SSO" Project
Generator Shaft WMUs (confidential), Additional Grid WMUs
WAMS
Offline Studies
NG-VISOR
WAMS Datacentre nationalgrid
WAMS Data Hub VISOR
NG-DH / VISOR-DH
Paymetrix VPN
NG-Production
WAMS Server nationalgrid
Existing NG SO WAMS (Control Rm)
1 IEEE C37.118 - 50 fps PMU streams
2 IEEE C37.118 - 200 fps WMU streams
Generator PMU / WMU streams (confidential)
Confidential + non-confidential streams
3 SQL Interface (3rd Party Analysis)
4 PhasorPoint Workbench HMI Interface (GUI)
5 Data Archive export: CSV / file transfer (tbc)

Figure 4.1: Critical Infrastructure of the GB Transmission WAMS [66]

Data collection occurs at the Data Hub and Datacentres. They act as PDCs, as well as providing storage, analysis and visualisation of the WAMs data for their specific area.

WMUs are comparable to PMUs in that they provide synchronised measurements of voltages and currents and use the IEEE C37.118 [63] communication standard. WMUs differ from PMUs by recording point-on-wave measurements of a voltage or current waveform at a 200 Hz sampling and reporting rate. This offers a greater visibility than the current PMU capability of around 10 Hz, up to 46 Hz to include sub-synchronous oscillation (SSO) behaviour that covers series capacitors, generator torsional behaviour and power electronic systems.

4.2.3 Current Metering & Analysis Options

Transmission network data is typically collected using SCADA from substation data centres and from PMUs installed on the lines, the majority of which are at substations. The readings are sent to data centres every few seconds to be analysed, providing a real time overview of the network. Transmission line parameters used for analysis can be 25 % - 30 % in error compared to measured SCADA values due to the expense of recording and updating the values of parameters calculated from ideal line geometry. These include line resistance, capacitance and reactance, and could be costing utilities millions of dollars in estimated transmission losses [68].

4.2.4 Transmission Network Modelling

The transmission network model is often maintained in the control systems as a node-breaker model. This model is integrated with the SCADA measurements for displaying the real-time view of the network including switch positions and telemetered points. It includes all network components at a high level of detail.

A transmission planning model uses a bus-branch representation: a simpler reduced view of the network where closed switches are ‘collapsed’ into computed buses with lines and transformers maintained as branches. These models are used as input for analysis, such as power flow and state estimation, where the bus-branch structure maps into the Jacobian matrix used as part of a standard Newton-Raphson [69] or Fast Decoupled [70] power flow. Planning models also reflect the evolution of power system planning. Initial plans are at a high-level to determine the impact on the overall system, and they are refined when projects move into detailed design and construction phase.

4.3 Distribution Networks

4.3.1 Extending WAMS to Distribution

Enhanced modelling capabilities will ideally include WAMS across all voltage levels. Digital computing should be capable of safely enabling system services and transferring power. It should also allow operators to utilise the existing system safely and to its maximum capacity. However there are a number of challenges surrounding the deployment of PMUs on the distribution network [71]. Secure and reliable communication is currently not present in all areas of a network and is especially inadequate in distribution networks. This means that communications with low-latency and high-bandwidth cannot be assumed. The main challenges for network owners are: improving the existing communications infrastructure and resources to enable data flow, the merging of IT systems, and ensuring cyber security [2]. It is important that information is provided to the appropriate people at the correct time, irrespective of whether they are the DSO or TSO.

4.3.2 Current Metering & Analysis Options

Economy is essential for the distribution grid. If smart meters were used to monitor WAMS thousands of devices would be required and the information reported would be limited. One or two µPMUs installed on the same network could report more useful information with greater accuracy and fidelity.

Assuming that there are 1000 customers reporting single-phase phasor data at a frequency of 1 Hz, this would produce 2.25 GB of data each day. This is calculated on the assumption that the data is sent as an IEEE C37.118 data frame [63] with payload data (ignoring identity codes and checksum fields):

- 2 phasors with magnitude and angle, one each for current and voltage with 4 bytes per value (16 bytes total)
- Frequency of 2 bytes
- Rate of Change of Frequency of 2 bytes
- Timestamp (taken from the two IEEE C37.118 values for timestamp and fraction of a second) of 6 bytes
- Status of 2 bytes

This means that every data frame for each customer is:

Phasors + frequency + rate of change of frequency + timestamp + status =

[(2 * 2 * 4) + 2 + 2 + 6 + 2] = 28 bytes

With 86,400 data frames sent per day at 1 Hz this becomes:

Data frame size * frequency * frames per day =

28 * 1 * (60 * 60 * 24) = 2,419,200 bytes

Resulting in 2.3 MB per customer, or 2.25 GB of data for 1000 customers each day.

Alternatively, a µPMU collecting three-phase data at 50 Hz would have six phasors for the voltage and current for each phase creating a frame of:

Phasors + frequency + rate of change of frequency + timestamp + status =

[(3 * 2 * 2 * 4) + 2 + 2 + 6 + 2] = 60 bytes

Data frame size * frequency * frames per day =

60 * 50 * (60 * 60 * 24) = 259,200,000 bytes

Producing 247.2 MB of data per µPMU, or 494.4 MB for two µPMUs each day.

The µPMUs would be collecting less data by volume but with higher fidelity. This is less data to store, sort and analyse and a smaller number of devices to maintain.

4.3.3 Distribution Network Modelling

Distribution network modelling is typically focused on asset maintenance. The majority of network models were constructed using geographical data. They have limited monitoring and visibility as traditionally there was little to no generation connected to the distribution networks, therefore power flow was one-way and predictable.

There is a lack of monitoring capabilities at lower voltages and a lack of accurate data about cable properties. This may be due to historical records being stored non-digitally, and because until the arrival of DER it was sufficient to operate the network by treating feeders as predictable loads. Connectivity data was commonly used to determine outages on a network rather than to build power flow models. Without power flow models, equipment data and the ability to closely monitor the network, the level of analysis that can be done on a distribution network is limited.

4.4 Comparison of Transmission & Distribution

While transmission networks generally cover larger geographical areas than distribution networks, the bulk of power transmission networks have a smaller number of lines and substations operating at higher voltages. The size of the network data is therefore relatively small in comparison to that of the distribution networks they feed. For example, SHE Transmission Limited (SHETL) covers the same geographical footprint as the Scottish Hydro Electric Power Distribution (SHEPD) distribution network. SHETL has approximately 5,000 km of electricity transmission infrastructure [72] while the SHEPD network has approximately 47,500 km of underground and overhead cables [73]. Transmission networks may be an order of magnitude smaller in size, but a failure at the transmission level can impact a larger part of the network (and more customers) than an issue at distribution. This means that transmission networks are more closely monitored and have their electrical network models maintained to a higher level of accuracy to enable simulation.

Distribution networks produce more data than transmission networks. In terms of lines and cables there is roughly 10 times the amount of information available, and there are millions of load points where the cable is taken to a metering device.

Transmission monitoring uses network analysis intensively in both planning and operations. Distribution planning usually relies less on network analysis, and distribution operations may not use any analysis [74].

4.5 Useful Improvements

Operators do not necessarily need 100 - 200 samples per second when communications infrastructure is restricted, but they do require the outcome of analysis in order to respond to network events. Investing in distribution management centres and widespread sensing capabilities would provide increased operational visibility, however the availability of data communication and analytics infrastructure is limited. Transmission system operators have little-to-no visibility of faults and outages beyond the grid supply point. Chapter 5 considers µPMUs as part of a data framework, including communications and analytics frameworks. Visible information would be prioritised for both communications and the system operators, although they may each require the data to be filtered and translated into an output that they will recognise. Some operators will prefer to see the action that should be taken when the phase angle deviates as opposed to just the measurement.

An improvement that is currently happening is the change in the requirements of distribution networks due to DER and price responsive demand. This is challenging the traditional view that distribution networks are a fixed demand on transmission and transmission networks are a fixed source for distribution [74].

The University of Texas at Austin have created an independent synchrophasor network [75]. This is a network that uses PMU measurements at the distribution level from sites across the state of Texas. The aim of the research was to validate the quality of PMU measurements and to show that the measurements are acceptable for power system analysis, by comparing them to measurements taken at the transmission level. Using three weeks of measurements, the data from both voltage levels are closely matched. This suggests that the installation of PMUs on the distribution network is beneficial for all stages of network analysis.

4.6 Concluding Remarks

This chapter has studied the similarities and differences between transmission and distribution networks. Although transmission networks cover a larger geographical area, the amount of network data is much less than that of a distribution network. The level of monitoring and analysis is greater at transmission level due to the availability of real time measurements, accurate network models and the increased impact that a high voltage failure will have on the operations of the entire network.

The main challenges encountered by TOs when observing large distribution networks have been discussed, in terms of visibility and control. Improving the existing communications infrastructure is key to improving this, as is maintaining satisfactory cyber security.

There are many challenges involved with installing µPMUs in resource constrained areas, such as the lack of secure and reliable communications in all parts of the network and the need to ensure cyber security across all IT systems, data flows and communication channels. However, as described in 4.3.2, µPMUs would collect less data than other metering options by volume, but with a greater reliability of data, giving them huge potential for improving monitoring and control of distribution networks.

Chapter 5 Triggering μPMUs & Communications

5.1 Introduction

The widespread adoption of PMUs in transmission networks has been extensively studied and shown to offer improved visibility: enhancing grid stability and avoiding low probability, high impact events such as widespread outages or complete system blackouts. For many TSOs, there is almost no visibility of the real-time state of distribution networks beyond the grid supply points. With low and medium voltage networks seeing an increasing penetration of DER, the state, behaviour and load levels on the distribution network cannot be assumed to follow historical patterns. The transition of DNOs from network asset operators to Distribution Systems Operators (DSO) is requiring them to deal with a more dynamic grid containing DERs and intelligent network control and management schemes. This requires the DNO/DSOs to have better visibility of the entire grid and for the TSOs to have visibility of the network state, especially around grid supply points. Real time visibility is key for network operators when deploying wide area control strategies and active network management schemes. This implies active sharing of high quality data between transmission and distribution monitoring systems in real time.

There are a number of challenges for utilities seeking to extend the use of PMUs to distribution networks. Many of the lessons and best practices developed during PMU deployments by TSOs on the transmission network do not translate to the operation of distribution networks. This is because equipment is often installed in remote locations, lacking both the physical security and access to high-speed telecommunications that is ubiquitous at transmission level. The exponential increase in network complexity makes it unfeasible to perform complex analysis of the complete network model in real-time using traditional approaches to analysis. Chapter 6 will demonstrate how cloud computing can be used to overcome many of these restrictions. This chapter presents a methodology for addressing the problems faced in deploying μPMUs at the distribution level.

5.2 Communication Technologies

Transmission systems cover large geographical areas but have relatively few, large-scale substations compared with distribution networks. As such, TSOs are in a position to install and manage their own dedicated communications network that provides secure, high-bandwidth, low-latency communications. These communication networks will often be on dedicated fibre optic cables that are installed in parallel with the transmission lines, thus connecting the primary substations to the control room over dedicated, secure links. When installing a PMU in a transmission substation, the existing communications network is generally used as it provides the bandwidth, latency and security levels required.

When installing µPMUs on the distribution network however, it is less common to have a dedicated communications network installed across the DNO's service territory. It is more common for the communications links to use public networks, such as cellular network services, fixed-line telephone/broadband connections or wide area radio networks. The performance of these communications networks can vary depending on the location of the equipment and local availability of communication services. For distribution networks covering dispersed, rural populations it is common for equipment to be in areas that are not served by any high bandwidth communication links.

The installation of µPMUs on a distribution network must therefore address a number of issues when choosing locations for the equipment, including those posed by the availability of communication links:

- Reduced and/or unpredictable communications bandwidth
- Variable or high latency connections
- Monthly data usage caps causing restrictions in cumulative transmission of data
- Use of public communications networks requiring additional security layers and controls
- Lack of direct control over network outages when a network goes down (e.g. outages caused by scheduled maintenance by the service operator)

Previous research has looked at the comparative advantages and disadvantages of different communications technologies [76]. Modern cellular networks, especially the latest fourth generation technology, can provide sufficient bandwidth for µPMU communications. However cellular links will generally include monthly usage constraints and the cost of a contract with sufficient capacity to transmit all the data collected by a µPMU can be prohibitive. Other users can have an impact on the latency, bandwidth, and availability of the communications link when operating on a public network and that is outwith the control of the network operator.

5.3 Secure Communications

5.3.1 Direct Access via Virtual Private Networks

Virtual Private Networks (VPN) [77] have previously been used within the power industry in conjunction with cellular communication networks [78] to enable secure access over public networks, and are widely used for back-office communications both within the power industry and across other industries. Transport Layer Security (TLS) [79] cryptographic protocols allow for secure, encrypted end-to-end communications over public networks. Open, standard data communication protocols such as HTTPS [80] are built on TLS and are widely used for secure web browsing and e-commerce as well as machine-to-machine communications. Previous work [81] has looked at the security of the Smart Grid and how public key infrastructures can be used on top of existing mechanisms such as TLS to enable secure communications.

For µPMUs deployed on distribution networks using public communication networks, the architecture for receiving commands or data and subsequently providing the collected data to systems can vary depending on the security requirements and the parties and systems requiring access to the data. A VPN can be established between the µPMU and the utility's internal network and configured to allow the µPMU to communicate directly with systems within the DNO's own internal network. Any further sharing of the data with other parties, such as the TSO, would require additional links between the DNO and TSO to be established in the same way as any other data-sharing link such as ICCP.

The advantage to this approach is that systems requiring the data from the µPMU can communicate directly with the devices in the same manner as those deployed on a transmission network. µPMUs appear as network connected devices on the utility's internal network and can be addressed directly. There are a number of potential security issues with this approach. The µPMU devices in the field are unattended and are typically connected to a public communications network with the trusted VPN connection established. It is unfeasible for distribution network operators to provide the same level of physical security at every site where they have equipment, although this is expected at major transmission substations. As such, these devices may be considered physically insecure but with a trusted network connection into the secure internal network. This has already been identified as an avenue of attack [80] for third parties to gain access to a utility's internal network. There is a serious risk that a misconfigured VPN connection or unknown vulnerability would provide an avenue for malicious third parties to gain access into the DNO's internal network, either physically via the device or from unauthorised access via the public communications network. This architecture also requires the DNO to set up data sharing from the µPMU to the TSO via another mechanism, adding complexity and latency compared with the TSO accessing the µPMU data directly.

5.3.2 Distributed Access using Cloud Computing

A second approach is to keep the µPMU outside the utility's internal network, and instead configure the µPMU to communicate with a cloud system that stores the data and shares it with authorised parties [82]. This cloud architecture would involve both the µPMU and the cloud server connecting to the public internet, but would also allow users within the DNO and TSO to pull data from the cloud server directly. The cloud server would be installed within a data centre with an 'always on' connection, either within a commercial, public cloud hosting provider such as Amazon Web Services (AWS) or Microsoft Azure, or within a dedicated externally connected hosting facility within the utility. For devices with low bandwidth communication links, they would connect and push data only when required.

This offers advantages over the direct VPN connection in that the users within a secure TSO and DNO network are only 'pulling' data from the cloud server then performing the integrations locally. By restricting the connection to providing only data feeds it is easier to secure from both the cloud server and utility operators perspective. The cloud server does not need a VPN connection and does not have to manage any passwords for ‘pushing’ data into potentially critical control systems. From the network operators’ perspective, they can restrict access to the server to only specific addresses and treat it as a feed of data that can be validated and checked prior to integration with any critical systems.

If the cloud server was to be compromised any malicious third party would have access to the data being collected, but they would not have VPN access into the secure networks. The µPMU devices in the field can similarly only connect to the cloud server to ‘push’ data, but do not have any connection into the TSO/DNO network. If they were compromised, a third party could alter the data being sent, but would not be able to issue commands or access the critical control systems.

The disadvantage to this architecture is that the system would be storing potentially sensitive real-time sensor data on a server connected directly to the public internet. If properly configured [77] these risks can be mitigated, but recent cyber-attacks [83] have shown that it is difficult to robustly secure systems. The priority would be to mitigate the risks to the secure, internal networks if a remote device is compromised.

As mentioned in 1.3.1, the security of a cloud platform is extremely important when sensitive information is being stored on it. Utilities are understandably wary about allowing their data to be kept in a cloud for any length of time, but there is a substantial difference in the levels of security available on public clouds and private clouds. Clouds used for the collection of µPMU data would ideally be a Trusted Cloud (1.3.2) with varying levels of access for the TSO, DSO, any other departments viewing the data and µPMUs pushing their data. Every account that has access to the Trusted Cloud would require at least two-factor authentication and verification before communication with the cloud is permitted. As utilities will all have different requirements surrounding the security of their network assets it was unable to be investigated fully within the scope of this research.

An architecture that restricts the communications between critical, secure internal systems and the data collected by the µPMU devices to 'pull' commands issued by the internal system to a cloud server has noticeable advantages. If the system is built on the assumption that the data could be compromised then it can be automatically validated and cleansed prior to any interpretation or integration. By limiting connections from the control system to specific protocols, ports and internet addresses in the cloud, it restricts the scope of what an attacker can accomplish. If a µPMU is compromised, all the attacker can achieve is to stop the data feed or compromise the data being sent. This could lead to erroneous readings and a potentially incorrect interpretation of network conditions, but it would prevent direct access to any control system as occurred during the recent cyber-attacks.

5.4 Use of Triggers

5.4.1 SCADA Triggers

SCADA systems on transmission networks currently use a form of event triggering to monitor the current state of the network. Measurements from RTUs and IEDs are collected periodically and stored by time stamp in a database, and when newer information arrives the old values are overwritten or archived. When an event is triggered on the network, SCADA will request measurements from all nearby devices to provide an up-to-date view of the current system state. Events can include breakers opening or closing, equipment such as a capacitor being engaged, an event predefined by the TSO, or a network alarm being triggered by a critical or unexpected occurrence on the network.

5.4.2 µPMU Synchronisation with Triggers

By using encrypted communication protocols over public communication networks, µPMUs on a distribution network can be quickly connected with the cloud platform managing device coordination. Where µPMU devices are operating in variable or low communication bandwidth environments, it is not feasible to provide full 50 – 200 Hz data as would be expected in a transmission network. Instead the synchronisation frequency of data would be reduced to regular intervals of a much lower frequency, unless triggered by configurable metrics.

These localised triggers would force the µPMU to transmit more data to the cloud server and indicate an event has occurred. Triggers can be set based on a device's position on the electrical network with pre-defined metrics. For example, WAMS at transmission may automatically trigger alarms with a pre-defined condition such as the rate of change of frequency exceeding pre-defined limits. Similarly a change in magnitude or angle may go above or below a defined maximum/minimum, triggering the µPMU to notify the cloud server of an event.

These events are sent to the cloud platform, allowing it to interpret the event and any data sent with it, at which point it may request all other devices it controls to force a synchronisation of their data. This allows a centralised analysis of the data from all devices at a high level of detail. A notification to any subscribers (e.g. the DSO and TSO) about the change in network conditions would also occur at this point.

With automated synchronisation, messaging and integration between the µPMU and the cloud server, and the µPMU having knowledge of the underlying network topology, more intelligent local analytics can be used to create dynamic triggers. These would require the device to use its knowledge of the local topology to intelligently determine whether an event has occurred that requires notification to the cloud server. Based on its view of local, regional and overall network conditions, the cloud server can update these triggers automatically, distributing new conditions and trigger configurations to the devices dependent on the known communication restrictions as well as electrical network conditions.

5.5 µPMU Data

5.5.1 Type of µPMU

The µPMUs used in the LBNL and utility test network were developed by PSL [7] with costs that are a fraction of the PMU devices installed and used at transmission level by TSOs. Using these smaller, cheaper devices allows for a wider deployment and use beyond real time monitoring. This research focuses on building a bottom-up view of the electric power distribution system using µPMU measurements. It occurs in the context of DNOs having limited availability of accurate system models and supplemental measurements (such as SCADA and AMI) which are becoming more important for network operations and planning as DNOs see more DER and dynamic components being added to their networks. To this end, µPMU measurements are utilised to parameterise approximations to system power flows in order to highlight discrepancies between measured network parameters and those in the network models.

5.5.2 Data from µPMUs

The µPMUs installed on the LBNL network have a sampling frequency of 120 Hz. For each sample the dataset contains:

- The magnitude and angle of the voltage for all three phases
- The magnitude and angle of the current for all three phases
- The time stamp [7]

This data is available for a number of µPMUs across two test networks installed at LBNL. As part of this research one line of 12.47 kV (line-to-line voltage) with a µPMU at either end was examined. This data proved problematic to analyse due to the lack of a significant voltage drop across the line caused by low loading levels. In order to carry out a more meaningful assessment of the technique’s feasibility, a line from a partner utility was used along with a transformer from the LBNL network and the corresponding network models.

5.5.3 Network Model

Figure 5.1 shows the configuration of µPMUs for estimating the transformer parameters. An estimation of the impedance of a transformer is implemented using the measurements from µPMU A2 and µPMU A3. There is a short cable connecting µPMU A2 to the transformer, however the impedance of this cable was assumed negligible with respect to the impedance of the transformer.

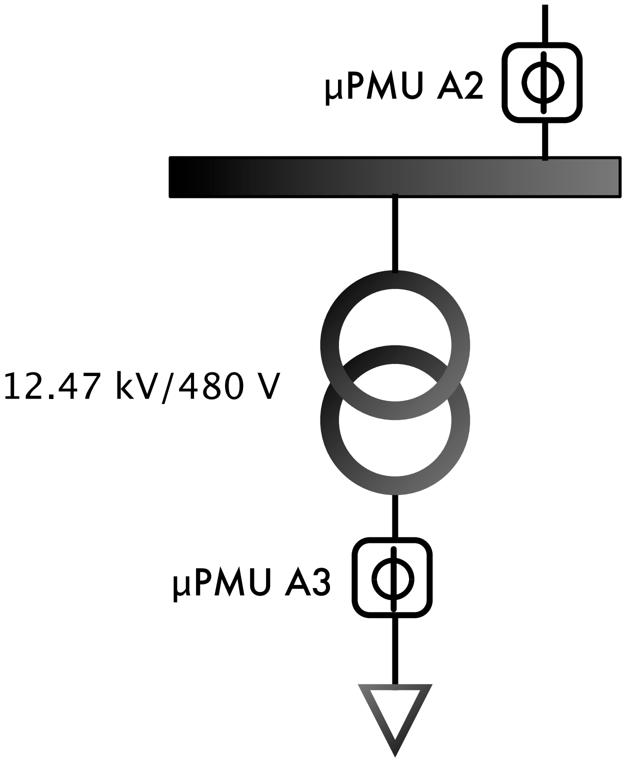


Figure 5.1: Device Configuration for Transformer Estimation

In the second data case there are two µPMUs at either end of an overhead line, from a partner utility, which is approximately 2 miles long with significant daytime loading. This line was examined as it was determined that the loading of the cables on the LBNL test network was insufficient to result in appreciable voltage drops along the cable.

5.6 Preliminary Equations

The intention of this work is to help operators and planners identify any obvious, large discrepancies between the measured parameters of the network and those present in the network models that are being used for online and offline analysis. The assumption is that distribution network models are often built using nominal parameters and that it is common for the constructed network to differ from what was planned or what was assumed when the network models were created. If large, unexplained discrepancies are discovered during this analysis, or if a gradual change over time is detected, a more detailed survey of the equipment would be undertaken. This would allow the operator to update their network model, or to send field crews to perform remedial action should the analysis indicate that there had been a failure or damage to equipment.

As a proof of concept, an analysis of the output data using two different methods for single line power flow was used to determine the impedance magnitude across the line. The lines were treated as a single-phase cable and mutual impedances were ignored.

Following this, the impedance of the unbalanced three-phase line located at the partner utility in Southern California was estimated using an Ordinary Least Squares (OLS) method. Finally the impedance of the delta-grounded wye transformer located at LBNL was estimated.

5.6.1 Simple Methods for Impedance Magnitude Estimation

5.6.1.1 Ohm's Law

The first equation applied to the data is to use Ohm's Law with the measured voltage and current phasors to determine the impedance magnitude across the line:

$$\begin{gathered} \Delta V = I \cdot Z \\ |Z| = \frac{|\Delta V|}{|I|} \end{gathered} \qquad (1)$$

This allows for a quick, high-level comparison of the calculated impedance with the model impedance. If a significant error is detected between this estimated impedance and the impedance in the electrical network analysis model, a more in-depth analysis would be required using a more complex model accounting for any loading unbalance in the system.

5.6.1.2 Cespedes' Method

Cespedes' formula provides a method of calculating the power flowing into a bus across a line with given impedance. The formula [85] is shown in equation (2) below:

$$\begin{gathered} V_r^4 + [2(PR + QX) - V_s^2]V_r^2 + \\ (P^2 + Q^2)(R^2 + X^2) = 0 \end{gathered} \qquad (2)$$

For a distribution line the estimated X/R ratio (ρ) can be used, calculated from equation (1), and thus ρR can be substituted for X in equation (2) and rearranged into the quadratic form:

$$\begin{gathered} R^2[P^2(1+\rho^2)+Q^2(1+\rho^2)]+ \\ R(2PV_r^2+2Q\rho V_r^2)+ \\ (V_r^4-V_s^2V_r^2)=0 \end{gathered} \qquad (3)$$

The calculated powers P and Q are used as the values from the *to* side of the line, and based on the *to* side voltage and current across the line equation (3) can be solved to provide a value of R for the assumed X/R ratio.

5.6.2 Impedance Estimation in Unbalanced, Three-Phase Lines

The three-phase, unbalanced case was investigated using a setup where the measurement data is for a single line connecting a substation to the point of common coupling for a large solar installation. The solar installation and the substation are both monitored by µPMUs. A few seconds worth of voltage and current measurements from the two µPMUs is sufficient to make an approximation of the impedance of the connecting cable.

The problem can be written as an Ohm's Law equation in matrix form. It considers both the long distribution line connecting the substation, which is the focus in this estimation, and the shorter line connecting the photovoltaics (PV). The shorter line is assumed to have a negligible voltage drop with respect to the primary line of interest.

$$\vec{V}_{sub}-\vec{V}_{PV}=\hat{Z}_{sub}\vec{I}_{sub} \qquad (4)$$

Where the *Z* terms above are matrices of the form:

$$\hat{Z}=\begin{bmatrix} Z_{AA} & Z_{AB} & Z_{AC} \\ Z_{AB} & Z_{BB} & Z_{BC} \\ Z_{AC} & Z_{BC} & Z_{CC} \end{bmatrix} \qquad (5)$$

Re-arranging (5) allows the elements of the substation impedance matrix to be collected in a vector and solved using the OLS method.

5.6.3 Transformer Impedance Estimation

The transformer under investigation was a delta-grounded wye transformer on the LBNL test network. To perform an impedance estimation, the primary side voltage was referred across the transformer to the secondary side, as shown in equation (6), whereby uppercase subscripts refer to the primary side:

$$[A_t][VLN_{ABC}]-[VLG_{abc}]=[Z_{abc}][I_{abc}] \qquad (6)$$

With the following matrices having the form:

$$[A_t]=\frac{1}{n_t}\begin{bmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \qquad [Z_{abc}]=\begin{bmatrix} Z_a & 0 & 0 \\ 0 & Z_b & 0 \\ 0 & 0 & Z_c \end{bmatrix} \qquad (7)$$

$$n_t=\frac{VLL_{\text{Rated High Side}}}{VLN_{\text{Rated Low Side}}}$$

Where *LL* denotes the line-to-line voltage and *LN* denotes the line-to-neutral voltage.

5.7 Preliminary Results

5.7.1 Simple Methods for Calculating Impedance Change

Table 5.1 shows a comparison of the impedance magnitude estimates for the unbalanced three-phase line at the partner utility. The model data is compared with the results calculated from using Ohm's Law (1) on the data collected in one day.

Table 5.1: Impedance Magnitude Estimates

	Model Data	Ohm's Law
$\lvert Z_{aa}\rvert$	1.2497	0.984
$\lvert Z_{bb}\rvert$	1.2497	0.906
$\lvert Z_{cc}\rvert$	1.2497	0.969

There are a number of potential reasons to explain these discrepancies:

- The potential transformers being used were not calibrated correctly resulting in inaccurate readings
- The network model does not accurately reflect the as-built network
- There is an issue with the network construction and the readings reflect degradation to the cable or attached equipment

The reasons for these discrepancies are being investigated by: looking at the data over a longer period of time to see if longer trends for the two lines show changes that would indicate degradation, or patterns that correlate with load or weather patterns; checking the calibration of the equipment; and verifying the configuration of the installed equipment. While it has yet to be determined whether this analysis of the data has identified an issue with the network, it highlights how the measured values can differ from what is expected from the network model.

If the errors between the measured data and the model data are substantial enough to cause any issues, they could indicate an increased likelihood of equipment failure. It is up to the utility and the network engineers to decide whether or not there is an issue, but for the remainder of this research the measured values will be used.

The purpose of this work was not necessarily to directly update network models with measured values, but instead to help identify where there could be issues with the network model. It could be that in many cases the discrepancies are not significant enough to impact on the operation of the network. However, planners need to be aware of these potential issues when analysing how the network would be impacted by other changes to the grid such as the installation of DER or other dynamic equipment on the network. To determine the exact unbalanced network impedance more complex analytics must be applied to the data. The simplistic method described above could indicate a problem in the data whereas the complex methods seek to correct the impedance value and update the network model.

5.7.2 Unbalanced, Three-Phase Results

Following the testing of the simple analysis method, the robust three-phase impedance calculation methodology was applied. When this method was tested on field data collecting in one day, the estimated impedance matrix was calculated as:

$$\hat{Z}_{estimated} = \begin{bmatrix} 0.5+1.29j & 0.12+0.37j & 0.21+0.18j \\ 0.12+.37j & 0.59+1.24j & 0.10+0.29j \\ 0.21+.18j & 0.10+0.29j & 0.45+1.08j \end{bmatrix}$$

This can be compared against the values given by the utility feeder model:

$$Z_{real} = \begin{bmatrix} 0.523+1.135j & 0.146+0.387j & 0.146+0.387j \\ 0.146+0.387j & 0.523+1.135j & 0.146+0.387j \\ 0.146+0.387j & 0.146+0.387j & 0.523+1.135j \end{bmatrix}$$

The variation between the calculated mutual impedance values is likely due to the measurement error introduced by instrumentation transformers in the circuit. The estimation is reasonable for the least-squares method that was used, and with more advanced methods that treat that error explicitly significant reductions in that variation are expected.

5.7.3 Transformer Impedance Estimations

The initial estimate of the transformer impedance over the course one day, referred to the primary side, is shown below:

$$\hat{Z}_{estimated} = \begin{bmatrix} 1.38+6.47j & 0 & 0 \\ 0 & 0.86+6.18j & 0 \\ 0 & 0 & 0.71+7.13j \end{bmatrix}$$

This can be compared against that of the nameplate data:

$$Z_{real} = \begin{bmatrix} 0.899+5.886j & 0 & 0 \\ 0 & 0.899+5.886j & 0 \\ 0 & 0 & 0.899+5.886j \end{bmatrix}$$

The comparison between the two matrices shows that the estimated impedances lie within a reasonable range of the actual values. The discrepancies may exist due to the low loading of the transformer, approximately 15 – 20 % of its rating, and thus the no-load losses may be attributed disproportionally to each individual amp relative to lab testing. Another reason for the discrepancy may be the effect of non-linear loads on the transformer. These effects may compound the effects of the instrumental transformers utilised in measuring the values.

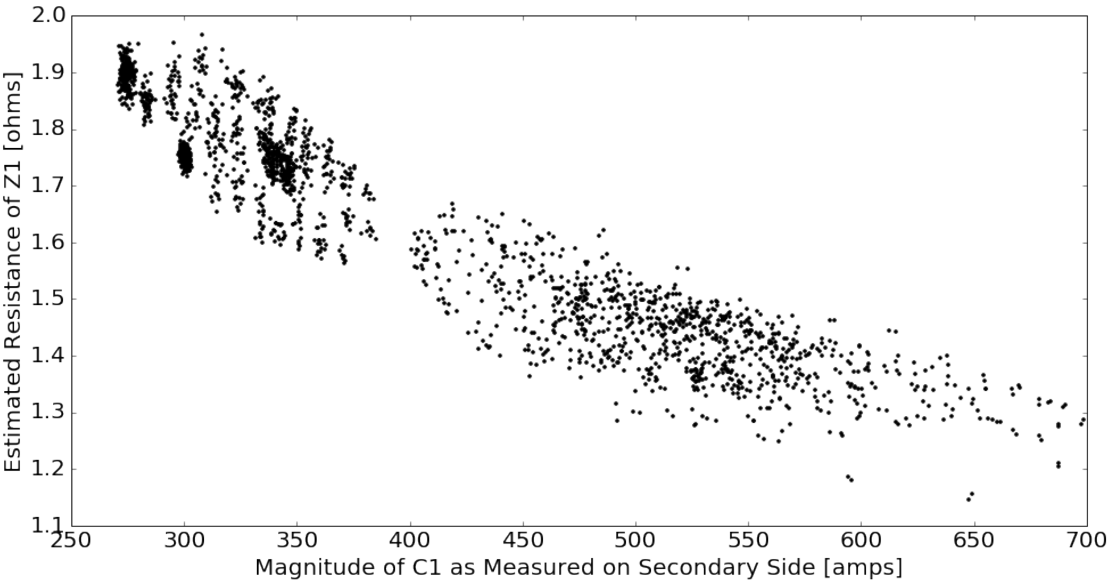


Figure 5.2: Estimates of Transformer Resistance as a Function of Current

Figure 5.2 shows the estimated resistance of Z_1 corresponding to each loading level. It can be seen that the estimates at higher loading levels are significantly above the nameplate rating. This is postulated to be due to the assumption of the magnetising current being negligible with respect to the load current in the deployed transformer model. As the load current increases this assumption becomes more valid and the estimated values tend towards the nameplate rating. Further analysis across a higher proportion of the transformer loading range is necessary to draw conclusive results regarding the behaviour of transformers at lower loading levels.

5.8 Trigger-Based Testing

Testing of the trigger-based data collection to reduce the bandwidth mentioned in 5.4 was implemented using a java program that processes historical data from 3 µPMUs on the LBNL network.

The application parses the µPMU data and simulates the real-time checks by testing for one of the trigger conditions. The trigger values used in this research were chosen based on discussions with engineers about what would be considered abnormal operating conditions on a network, with the understanding that in a real-world situation they would be chosen based on the utility's needs and prior knowledge of their own network. In this case the condition was any phase voltage magnitude being ± 5% of nominal. If this condition occurs more than 100 times within a specific time period (in this case 10 seconds) a trigger is activated. This will trigger the µPMU to transmit the current sample plus all the samples recorded during the time period (in this case the last 10 seconds). The application ensures that overlapping windows of data are identified so that the samples are not duplicated, thus avoiding the transmission of duplicate data that fell within overlapping windows if triggers occurred within 10 seconds of each other.

Initially 8 hours and 20 minutes of data covering 3.6 million samples was tested with the trigger of ± 5%. This produced no triggers on any of the µPMUs as all voltages were within these limits. The amount of data tested was increased to 27 hours and 46 minutes, covering 12 million samples, with the same result. The trigger value was reduced to ± 4.5% and this triggered 25 samples at a6_bus1 over 8 hours and 20 minutes. When including samples for the 10 seconds before each trigger, 10,222 samples would be sent. Again the amount of data tested was increased to 27 hours and 46 minutes, and this resulted in 20,258 triggered samples at a6_bus1 and 8,947 triggered samples at grizzly_bus1_2. These would send 1,376,388 and 980,053 total samples respectively. To allow for closer monitoring of the system the trigger value was further reduced to ± 4% and the test repeated for both time periods. As expected this increased the number of triggered samples and therefore the total number of samples that would be transmitted to the cloud server for analysis. A summary of these results is shown in Table 5.2.

Running the application had a low processing overhead. It took 7 minutes to parse 27 hours and 46 minutes of µPMU data and simulate the real-time checks with a trigger of ± 5%. It took 38.5 minutes to parse the same amount of data and simulate the real-time checks with a trigger of ± 4%. This suggests that the application could be run in real-time with minimal latency issues.

Figure 5.3 is a network diagram showing the location of two of the µPMUs that data was collected from. On the diagram, UPMU1 is grizzly_bus1_2 and UPMU2 is a6_bus1. The third µPMU, bank_514, is situated in a different area of the network and cannot be seen in Figure 5.3.

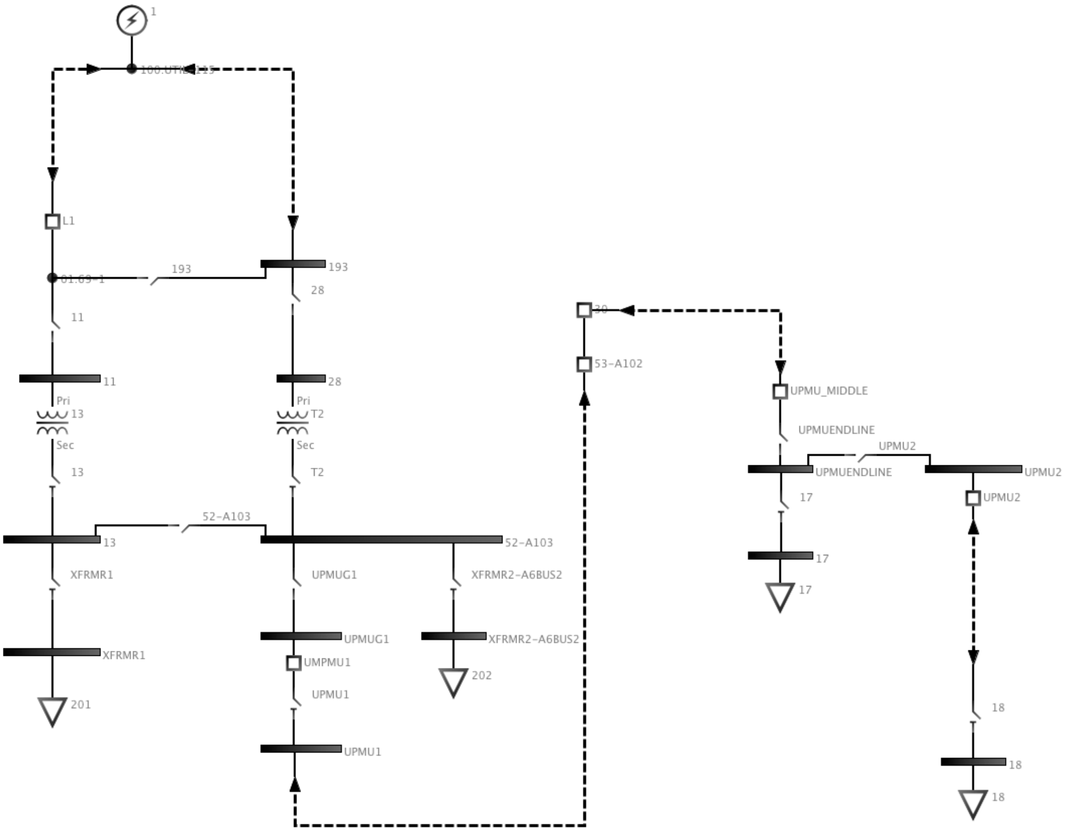


Figure 5.3: Location of µPMUs on the LBNL network

Table 5.2: Testing of Trigger-Based Data Collection

Trigger Value	Hours of data	µPMU	Total samples	Triggered samples	Samples sent
± 5%	8.33	a6_bus1	3,600,000	0	0
		bank_514	3,600,000	0	0
		grizzly_bus1_2	3,600,000	0	0
± 5%	27.78	a6_bus1	12,000,000	0	0
		bank_514	12,000,000	0	0
		grizzly_bus1_2	12,000,000	0	0
± 4.5%	8.33	a6_bus1	3,600,000	25	10,222
		bank_514	3,600,000	0	0
		grizzly_bus1_2	3,600,000	0	0
± 4.5%	27.78	a6_bus1	12,000,000	12,520	1,376,388
		bank_514	12,000,000	0	0
		grizzly_bus1_2	12,000,000	8,947	980,053
± 4%	8.33	a6_bus1	3,600,000	20,258	2,356,672
		bank_514	3,600,000	0	0
		grizzly_bus1_2	3,600,000	11,580	1,583,214
± 4%	27.78	a6_bus1	12,000,000	82,981	5,799,545
		bank_514	12,000,000	0	0
		grizzly_bus1_2	12,000,000	53,601	4,659,246

As calculated in 4.3.2, a µPMU sample (also known as a data frame) is 60 bytes. Using this value, the reduction in µPMU data transfer due to triggering can be calculated for the results in Table 5.2. This is shown in Table 5.3, where the reduction is also displayed as a percentage of the total samples that would be sent without any triggering.

Table 5.3: Reduction of Data Sent due to Triggering

Trigger Value	Hours of data	µPMU	Normal Data Sent (MB)	Data Sent due to Triggering (MB)	Reduction (%)
± 5%	8.33	a6_bus1	206	0	100
		bank_514	206	0	100
		grizzly_bus1_2	206	0	100
± 5%	27.78	a6_bus1	687	0	100
		bank_514	687	0	100
		grizzly_bus1_2	687	0	100
± 4.5%	8.33	a6_bus1	206	0.58	99.7
		bank_514	206	0	100
		grizzly_bus1_2	206	0	100
± 4.5%	27.78	a6_bus1	687	78.8	88.5
		bank_514	687	0	100
		grizzly_bus1_2	687	56.1	91.8
± 4%	8.33	a6_bus1	206	134.8	34.6
		bank_514	206	0	100
		grizzly_bus1_2	206	90.6	56
± 4%	27.78	a6_bus1	687	331.9	51.7
		bank_514	687	0	100
		grizzly_bus1_2	687	266.6	61.2

Although a 100 % reduction in data seems good, system operators would not be satisfied with zero data being received from a µPMU. Data must still be transferred at regular intervals, such as once a second, to ensure that the devices are still functioning and that a comprehensible record of network operations can be retained. Assuming a frequency of 50 Hz, a µPMU would produce 247.2 MB of data per day as calculated in 4.3.2. If the frequency of data transfer was reduced to once a second, this would result in 4.9 MB per µPMU per day, which is a reduction of 98 %. Combining the reduction in regular data transfer with the use of trigger-based data collection would result in a substantial reduction of the amount of data recorded by each µPMU.

The example used above shows the voltage moving outside of the limits on 100 or more occasions during a 10 second period (covering 1,200 samples). This limit could be lowered or raised which would impact the number of triggers that occur, as demonstrated in Table 5.2. Additional, more complex triggers could be applied to test for a combination of angle and magnitude values. The period over which the signals are analysed could also be extended. If a number of small disturbances were detected over a longer period of time it may trigger a warning, as this could be the symptom of a developing fault or imminent equipment failure.

The requirement for dynamic triggering would become apparent in an emergency situation, such as trees falling into remote power lines during a storm. Under these conditions it may be necessary to reduce the frequency of triggering as it would become a constant alert to the SO. Different triggers might be more appropriate in an atypical network situation, but as these triggers may not have been required previously they will require dynamic implementation in real time. An alternative solution would be to include an option to cancel all triggers for a set period of time and collect the full amount of data available from the relevant µPMUs.

5.9 Future Work

5.9.1 Complex Analysis

As stated previously, this is intended as a method of identifying where the network construction and network model may differ to help prioritise areas that need to be resurveyed. The method and equations used are well established and are deliberately using the voltage and current measurements from each µPMU independently of any construction information within the network model. This represents preliminary testing to check the feasibility of the proposed method. The purpose was to identify any impedance discrepancies between network models and field measurements. The accuracy of readings is also important when dealing with low voltage distribution networks, as the change in voltage and current is small compared to the changes that are seen in transmission networks.

The mutual impedances and line susceptance were intentionally disregarded to begin with to show that this is a viable method of observing when readings are not as expected and to help identify potential issues with the network. Future work will focus on validating the readings and applying the data to more complex methods, including complex mesh analysis. The calculation of impedance in a meshed network will allow identification of any change in the condition of network topology by determining where the change has occurred. This work has already begun, and will take into account more complex analysis using cloud-computing resources that provide access to large amounts of computing power.

The initial results of the analysis that included mutual impedances from the three-phase line impedance calculations are positive. There is, however, further room for improvement. Future work will begin with supplementing OLS with more sophisticated algorithms that account for instrumentation transformer error, and potential environmental factors. When this has been accomplished and µPMU-based impedance estimation methods fully developed, further research will involve integrating impedance monitoring into control strategies that make use of µPMU-measured voltage and current angles as operating variables.

The transformer parameter estimation provided reasonable results for the algorithm that did not take account of potential stable errors introduced by the instrumental transformers. An analysis of the impedance estimations at various loading levels, as well as various temperatures, may shed light on the operational behaviour of distribution transformers and in turn improve modelling practices.

5.9.2 Micro-Historians

'Micro-historians' can be deployed at µPMUs to locally store high-frequency data for retrieval in both the short and long term. These will act similar to a PDC with real-time data access, but will differ with a requirement for historical querying capabilities. Data will be sent to the cloud at a relatively low frequency, with a combination of local and remote triggers, and used to initiate a transfer of high-frequency data for centralised analysis. Localised data storage can be implemented on the micro-historians by the addition of a memory card to the device. Assuming the maximum amount of 247.2 MB of µPMU data is being collected each day, a 16 GB memory card would be able to store approximately 2 months of data. These memory cards are inexpensive and easily available from most electronics retailers.

In the event of a major outage a system operator may require all the available network sensor data at full resolution for post fault analysis. As the devices are installed on low bandwidth communication networks it is not feasible to extract days or weeks worth of data from them remotely. However the data could be obtained from the local storage of the micro-historians manually by dispatching field personnel to physically access each device and manually download the data from its memory card.

5.9.3 Extension of C37.118

The existing IEEE C37.118 [63] standard can be extended to accommodate the advanced historical querying of data from remote devices. This would work in combination with localised storage to allow for post-event extraction of high frequency data in the existing standardised form to work alongside the existing streaming protocol.

5.10 Concluding Remarks

This chapter has examined the problems faced in deploying µPMUs at the distribution level and discussed an approach to address these challenges.

The use of µPMU data to support improvements to network model analysis was investigated. Impedance equations were applied to real-world network data captured by µPMUs on the LBNL distribution network as preliminary testing to prove the viability of the proposed method.

A cloud-based architecture was designed to deploy µPMUs in remote locations. Combined with localised triggers that are implemented at the device level it would greatly reduce the amount of data being sent by µPMUs while still supporting network operation. An example trigger of the phase voltage magnitude being ± 5% of nominal more than 100 times in a time period of 10 seconds was tested, with the trigger-value being decreased to 4.5 % and 4 % to demonstrate the difference in the amount of data that µPMUs would send to system operators.

Chapter 6 Power Flow in a Cloud

6.1 Introduction

As computing power has increased, the time taken for power flow software to run individual power flow analysis cases has decreased rapidly, even for large network models. However, as the complexity of models has increased the number of potential contingencies to be analysed has also increased, therefore the time required to run the analyses for all potential contingencies on a network can take several hours.

Contingency analysis is an important part of power system operation. Operators need to know what changes will occur on a network when a specific piece of equipment fails and whether they will still be able to supply all connected loads. When contingency analysis results are obtained in real-time, they can be used to predict the outcome of certain network problems, such as those that may be caused by severe weather. This enables the development of a plan to preserve the network and use it to its best ability. When faced with emergency conditions and the need for quicker feedback, there has to be a compromise between the number of contingencies analysed, the complexity of the network models and the available computing power.

A novel approach to enabling fast, detailed analysis of all possible contingencies is to use a highly parallelisable power flow based within a trusted cloud platform. This would decrease the time taken to run through hundreds of thousands of contingency permutations, without requiring a utility to have the computing capacity in-house.

The objective of this chapter is to implement a cloud-based power flow, compare the performance and cost for different methods of contingency analysis and determine how beneficial they would be in practice. An N-1 contingency set will be run on a representative real-world power network on a commercial power flow engine in a cloud environment. Subsequently, testing will be undertaken to find the optimal settings in terms of the number of machine instances and number of cores per instance. This will include the length of time taken, the efficiency of each method, and the resulting cost.

6.2 Power Flow Software

As mentioned in 1.4, there are different methods of performing power flow analysis and each has its own strengths. According to Open Electrical [86], there are at least 37 commercial power flow analysis software available and at least 28 non-commercial, open source engines.

Some of the most commonly used commercial power flow packages are:

- PSS/E (Power System Simulator for Engineering), the power transmission system planning software from Siemens PTI [87]
- PSLF (Positive Sequence Load Flow), a power system analysis simulation engine from GE Energy Consulting [88]
- Ipsa 2, power system analysis software from TNEI [89]
- PowerFactory, the power system analysis software application from DIgSILENT [90]
- SCOPF (Security Constrained Optimal Power Flow), also known as SCOPE, part of the Grid360 Transmission Analytics solution from Nexant [91]

All of the software mentioned above are capable of running basic power system analysis including load flow, optimal power flow and contingency analysis. There will be variations in the exact methods used to calculate results, but these are undisclosed for commercial reasons. Other functionalities are often packaged together such as load balancing, protection simulations and harmonic analysis. The viability of using more commercially known software such as PSS/E or PSLF became impractical due to the cost being based on a single license and the nature of the research requiring multiple licenses to be used simultaneously. This is similar to the reason for discounting IPSA; it has a single-system license key restriction.

Open source power flow analysis software include:

- GRIDLAB-D, a power distribution system simulation and analysis tool that was developed by the US DOE at Pacific Northwest National Laboratory (PNNL) [92]
- OpenDSS, an electrical power system simulation tool developed by EPRI [93]
- SmartGridToolbox, a C++ library for developing smart grid simulations developed by National ICT Australia (NICTA)'s Optimisation Research group [94]
- PSAT (Power System Analysis Toolbox), a Matlab toolbox for electric power system analysis and simulation created by Federico Milano [95]
- MATPOWER, a Matlab power system simulation package [96]

The main reason that open source software was not used in this research is because technical support for it may not always be available. Another reason is that although all the software have the same core functionalities as those commercially available, they are not all designed to process large networks with the same efficiency. SmartGridToolbox, for example, is intended for academic research instead of a real-world simulation application, which would not be suitable for this work. GRIDLAB-D and OpenDSS are focused on unbalanced three-phase distribution system modelling which is more commonly undertaken on US network models.

In this research Nexant's power system analysis software SCOPE is used. It was chosen as it is a high performance, scalable, industry leading application and because of the support that Nexant were willing to provide as they have an interest in the outcome of the work. Software from a different company was trialled but its single-system licensing model made it unsuitable for cloud deployment where multiple instances are a necessity. Using open source software was considered, but it was decided that having support from Nexant would be invaluable to ensuring the research progressed efficiently. Nexant provided the SCOPE software engine and some initial test data for testing the cloud platform deployment. They also provided support to verify that the large-scale real-world model used was valid and in a format compatible with the SCOPE software. Support from Nexant was given by Narsi Vempati, Herminio Pinto, Brian Stott and Guanji Hou.

6.3 Network Model

6.3.1 Overview

The test model being used is a 21,000-bus model provided by ENTSO-E as part of their CGMES interoperability tests. The network is an anonymised version of a real-world transmission network model with 24,252 power lines, 4,832 generators, and 93 transformers.

6.3.2 Power Flow Results

Nexant validated the 21,000-bus network model using their SCOPE engine to ensure it solved and represented a valid, real-world network. The only modification required to enable convergence was to increase the QMax for the four generators at the swing bus (250042). Each was increased from 360 to 1360, with the number 1360 being chosen arbitrarily during tuning. This was required because all the slack (MW imbalance) in an outage case is being supplied by the swing bus. In this network the swing bus is extremely remote and is being asked to supply all the outaged MW for each generator outage, plus the change in MW losses. Without the change in maximum reactive power, any solution would put unrealistic stress on the transmission network, especially near the swing bus, and produce unrealistic flows and voltages that impede convergence. The alterative would be to use a distributed slack, so that multiple generators pick up the outaged MWs, but for the purposes of this research it was simpler to increase the reactive power generation and leave the rest of the network model unaltered.

6.3.3 Power Flow Results in a Cloud

The model was subsequently run through the same software (SCOPE) inside a cloud instance on AWS to ensure that the power flow simulation solved. This confirmed that the cloud instance was set-up correctly and was suitable to run further testing on.

The Alpha cloud instance that was used is a t2.medium model [97], with 32 GB of external storage added to the internal 4 GB. Installed on the instance are the SCOPE software, copies of the network model and a file that is used to run the power flow analysis. It was accessed via the command line terminal on a MacBook Pro.

The command used to run the power flow is:

```
$ time /opt/nexant/scope134/scope < runPowerFlow.mac
```

The time statement allows the user to see the duration of a command; the time in minutes and seconds is displayed on the terminal when the analysis is complete. The contents of the 'runPowerFlow.mac' file are:

```
------------------
- read in pca data
------------------
dat non
ENTSOE_21k_PSSE30_v0_14Sep2009.pca

----------------
- run power flow
----------------
PFL NWT QLC TVC YVC PCN
```

An AWS image was created from the Alpha instance with all the files, settings and software saved in their current state. The Amazon Machine Image (AMI), as described in 1.3.1, can be used to start up a new instance on any required machine type and size, which will be ready to run a power flow contingency analysis of the 21,000-bus network.

6.4 N-1 Contingencies

6.4.1 Overview

For the purposes of this research an N-1 contingency will be considered as the loss of a single transmission line or other network element. In real-world network operations N-1 represents a single contingency event, which may involve one or more power system network elements. The network must still function as close to normal operation as possible, meaning that the rest of the network must be able to reorganise itself to safely accommodate the fault. This involves other transmission lines operating at a higher capacity to ensure power flows and generators producing more power to balance out losses.

The N-1 contingency for the 21,000-bus network was generated using a custom application to analyse then add the appropriate contingency data. This Java application was written to parse the network file, extract each network element then add a contingency entry representing that element being taken offline. The elements that were selected are the transmission lines and generator units, because they have the largest impact on the operation of the network. Each required network element was saved, with the necessary details, to a list that was printed into a text file. The details of each contingency were then added to a contingency analysis macro on the cloud. The ‘runPowerFlow.mac’ file was updated to add contingency analysis:

```
------------------
- read in pca data
------------------
dat non
ENTSOE_21k_PSSE30_v0_14Sep2009.pca

----------------
- run power flow
----------------
PFL NWT QLC TVC YVC PCN
CTG
```

6.4.2 AWS System Architecture

The AWS Elastic Compute Cloud (EC2) provides secure and scalable compute capacity in the cloud. It allows the user to control many server instances on the AWS computing environment. The system architecture of AWS is not fully available for the public to view, to avoid competing cloud providers such as Microsoft Azure or Google Cloud Platform from being able to access it. As such it is impossible to know how this research will function when implemented in a different environment. It is assumed that the results will be similar, but differences in specifications such as processor features and the distribution of cores in a machine will have an impact on the overall performance.

AWS has computing resources in 16 regions around the world. It is not specified how the physical machines in a region are used to accommodate the number of cores requested by a user. For example, it could be that a 16-core machine is actually created using 16 cores from a 256-core physical machine. There is an unseen layer of AWS that manages how its resources are distributed.

A block diagram of how the services used in this research fit into the AWS architecture can be seen in Figure 6.1. EC2 instances are created within the AWS servers using the secure AMI that was created in 6.3.3, and an EBS volume is attached to each instance. There are many other AWS services available that support cloud deployment, such as databases, load balancing and management tools [9], but these were outside the scope of this work.

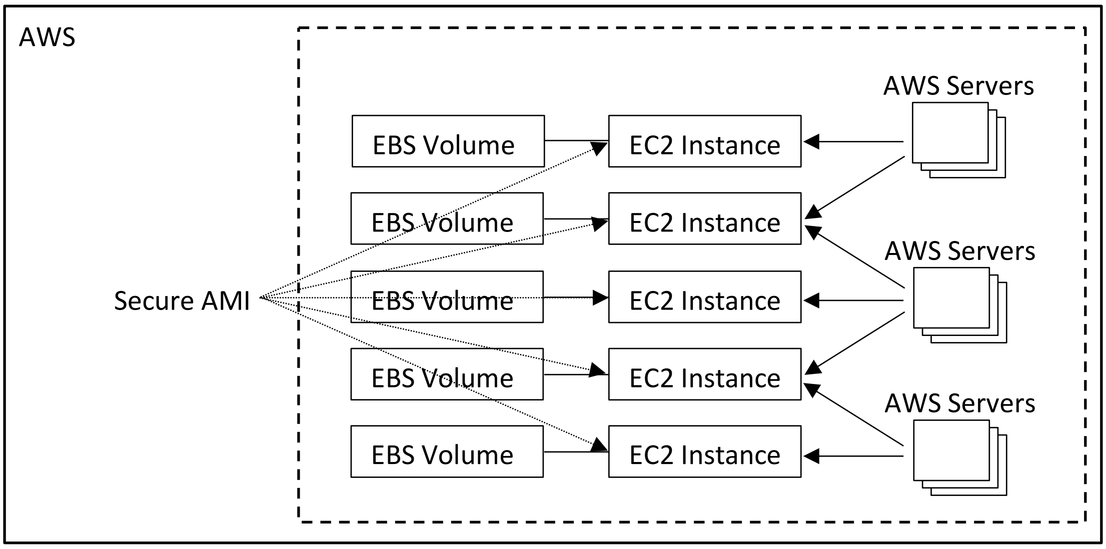


Figure 6.1: AWS Architecture

Table 6.1 is an excerpt of the AWS m4 machine specifications [103], which are the primary machines used in this research. It shows the variation in performance as the number of cores in a machine increases. The processors used in the m4 machines are 2.4 GHz Intel Xeon E5-2676v3 or 2.4 GHz Intel Xeon E5-2686v43 with AES-NI encryption, AVX performance improvement and Turbo performance boosting technology. It can be seen that specifications increase proportionately with the number of cores (vCPUs) in a machine.

The AWS EBS storage volumes used in this research are general purpose solid-state GP2 drives [9]. It has been designed to minimise latency and have a consistent baseline performance of 3 IOPS/GB with a throughput of 160 MB/s per volume. This provides a minimum of 100 IOPS and a maximum of 10,000 IOPS. The storage volumes will be attached to the machine instances and used in the same way as using physical block storage, meaning it will exist independently from the instances launched.

Table 6.1: AWS m4 Machine Specifications [103]

Family	Type	ECUs	vCPUs	Physical Processor	Clock Speed	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support	Processor Architecture
General Purpose	m4.large	6.5	2	Intel Xeon E5-2676v3	2.4 GHz	8	EBS only	Yes	Moderate	Yes	64-bit
General Purpose	m4.xlarge	13	4	Intel Xeon E5-2676v3	2.4 GHz	16	EBS only	Yes	High	Yes	64-bit
General Purpose	m4.2xlarge	26	8	Intel Xeon E5-2676v3	2.4 GHz	32	EBS only	Yes	High	Yes	64-bit
General Purpose	m4.4xlarge	53.3	16	Intel Xeon E5-2676v3	2.4 GHz	64	EBS only	Yes	High	Yes	64-bit
General Purpose	m4.10xlarge	124.5	40	Intel Xeon E5-2676v3	2.4 GHz	160	EBS only	Yes	10 Gigabit	Yes	64-bit
General Purpose	m4.16xlarge	188	64	Intel Xeon E5-2676v3	2.3 GHz	256	EBS only	Yes	25 Gigabit	Yes	64-bit

6.4.3 Single Processor

Transmission line contingencies were created first and resulted in 24,252 individual contingency entries. Using the Alpha cloud instance described above, the contingencies were applied to the network model. The software solved the transmission line contingencies in 12 minutes and 33.4 seconds. The results of this are shown in Figure 6.2. The most important outputs of the SCOPE analysis have been highlighted; these are the convergence status (POWER FLOW SOLVED) and the end of run timings.

At the top of Figure 6.2 the user (ec2-user@ip-172-31-41-137) has entered the appropriate command to run the power flow simulation and contingency analysis. To start, the title details the software that is being used, SCOPE 13.4, and that it is a Nexant Inc. product. Commands from the contingency analysis macro allow the simulation to run without any further human interaction required. The warnings listed on the message output file are not important to this research. The majority of the warnings are related to predefined network limits: line series reactance is smaller than the user definable parameter XMIN or generator specified active power is outside the specified range. These parameters will not cause any issues when running power flow. The remaining warnings are in response to data header files not found in the SCOPE software. Nexant did not provide these files because they are not required for contingency analysis.

The Newton-Raphson power flow convergence process can be seen next. This process details every iteration of the power flow; including the real and reactive powers and per unit value of the maximum mismatch, the bus IDs currently being used, and network limits. Following the iteration information there is a Newton-Raphson power flow convergence summary. Once the power flow simulation on the network has been solved the contingency analysis is run. Warning messages are also listed on the message output file, again they have no impact on the research being conducted. These warnings detail all the contingency cases that contain islands with no MW load that are excluded from the solution, and a final warning that power flow failed to converge in contingency case island(s). The line 'End of run' indicates to the user that the analysis has been completed.

As mentioned previously, the time command was used to display the simulation and analysis time on the screen. Real time is the total time taken for the process, user time is how long the CPU process was used directly for the simulation, and system time is how long the CPU was used indirectly.

Generator unit contingencies were created next, with 4,832 identified within the test model. For the purpose of this research it was decided to use only these two types of network elements for testing, as any others would generate so few results that it would add negligible contingencies and have minor impact on the results. Other network elements that are often included are buses, transformers and DC links. The total of 29,084 contingencies were solved in 16 minutes and 33.8 seconds, this is shown in Figure 6.3. On average, the time for transmission line outages is 0.031 seconds per contingency. The time for generator outages is 0.049 seconds per contingency, which is an increase of approximately 60 % on the transmission line outage time. This is likely to be due to the single slack bus discussed in 6.3.2.

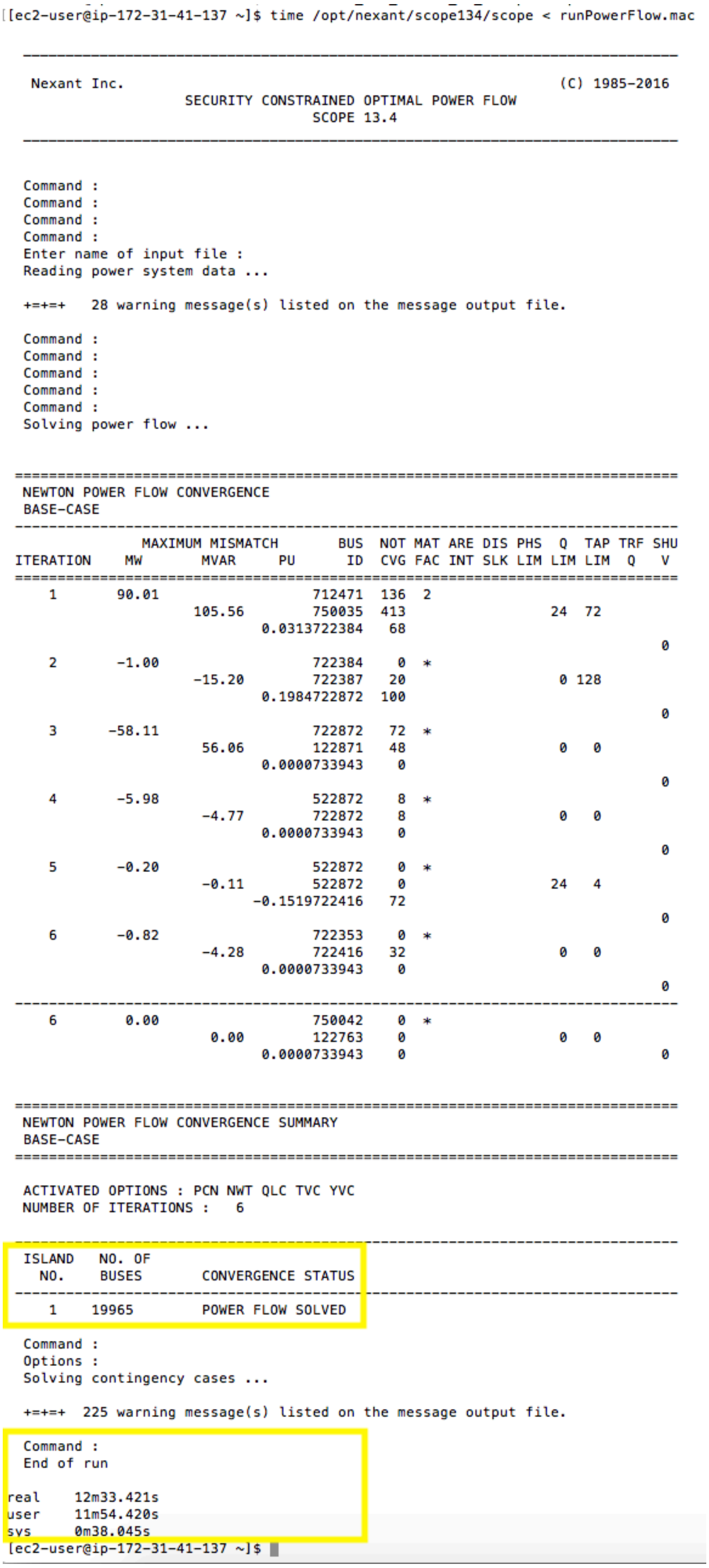
6.4.4 System Resource Restrictions

A complication that was discovered while running the contingency analysis is that for large networks the SCOPE software requires more memory than is available on the system to store the results of the simulation. This requires the use of 'swap space' and temporary files alongside the internal memory to hold sections of memory that are not currently required and allow the current process to continue. The time taken to move temporary files between the swap space can be very slow compared with moving data within memory, however this was expected and a normal part of execution.

During the initial contingency analysis execution on an AWS instance the time taken was measured at over 50 minutes, instead of the expected 16 minutes. It was initially thought that the machine instances were somehow broken or incorrectly set up because they had not completed the simulation after 30 minutes, 14 minutes longer than the expected time, so they were shut down. After allowing an instance to run until it stopped (51 minutes and 15.72 seconds) the issue was realised. A second machine was allowed to complete its simulation (71 minutes and

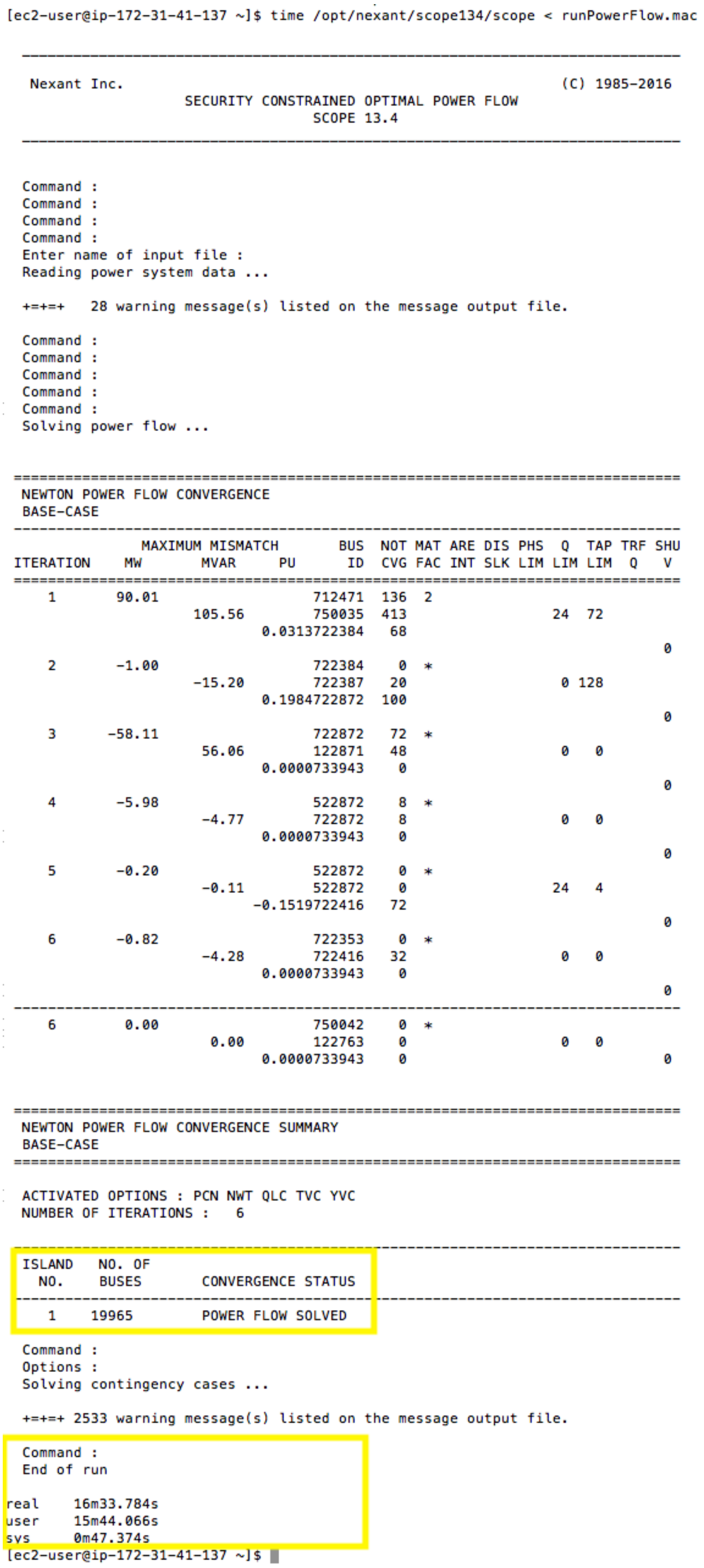
15.76 seconds) to confirm that the set up was not the issue. Repeated executions on a single instance produced the same result.

An investigation noticed that the SCOPE execution was producing approximately 1 MB of temporary data for each contingency. As the run involved close to 30,000 contingencies this was producing almost 30 GB of temporary data. The default instances being used had only 32 GB of storage, so these temporary files, when combined with the data storage requirements of the operating system and installed software, meant that the execution was running out of storage. The initial setup and test instances to validate SCOPE had been based on existing running instances that happened to be configured with more storage and so the issue had not arisen. The solution was to add more storage to the instances being used for SCOPE, ensuring that the process would not have to deal with running out of volume storage space. When rebuilt and run with the extra storage added, both machine simulation times were between 16 and 17 minutes.



```
[[ec2-user@ip-172-31-41-137 ~]$ time /opt/nexant/scope134/scope < runPowerFlow.mac

 -----------------------------------------------------------------------------

  Nexant Inc.                                                     (C) 1985-2016
                    SECURITY CONSTRAINED OPTIMAL POWER FLOW
                                   SCOPE 13.4
 -----------------------------------------------------------------------------


 Command :
 Command :
 Command :
 Command :
 Enter name of input file :
 Reading power system data ...

 +=+=+   28 warning message(s) listed on the message output file.

 Command :
 Command :
 Command :
 Command :
 Command :
 Solving power flow ...


==============================================================================
 NEWTON POWER FLOW CONVERGENCE
 BASE-CASE
------------------------------------------------------------------------------
               MAXIMUM MISMATCH       BUS  NOT MAT ARE DIS PHS  Q  TAP TRF SHU
ITERATION    MW       MVAR     PU      ID  CVG FAC INT SLK LIM LIM LIM  Q   V
==============================================================================
    1       90.01                  712471  136  2
                     105.56        750035  413                 24  72
                             0.0313722384   68
                                                                            0
    2       -1.00                  722384    0  *
                     -15.20        722387   20                  0 128
                             0.1984722872  100
                                                                            0
    3      -58.11                  722872   72  *
                      56.06        122871   48                  0   0
                             0.0000733943    0
                                                                            0
    4       -5.98                  522872    8  *
                      -4.77        722872    8                  0   0
                             0.0000733943    0
                                                                            0
    5       -0.20                  522872    0  *
                      -0.11        522872    0                 24   4
                            -0.1519722416   72
                                                                            0
    6       -0.82                  722353    0  *
                      -4.28        722416   32                  0   0
                             0.0000733943    0
                                                                            0
------------------------------------------------------------------------------
    6        0.00                  750042    0  *
                       0.00        122763    0                  0   0
                             0.0000733943    0                              0


==============================================================================
 NEWTON POWER FLOW CONVERGENCE SUMMARY
 BASE-CASE
==============================================================================

 ACTIVATED OPTIONS : PCN NWT QLC TVC YVC
 NUMBER OF ITERATIONS :    6

------------------------------------------------------------------------------
 ISLAND   NO. OF
   NO.    BUSES           CONVERGENCE STATUS
------------------------------------------------------------------------------
    1    19965            POWER FLOW SOLVED

 Command :
 Options :
 Solving contingency cases ...

 +=+=+  225 warning message(s) listed on the message output file.

 Command :
 End of run

real    12m33.421s
user    11m54.420s
sys     0m38.045s
[ec2-user@ip-172-31-41-137 ~]$
```

Figure 6.2: N-1 Contingency Analysis for Transmission Lines on Alpha Cloud



```
[ec2-user@ip-172-31-41-137 ~]$ time /opt/nexant/scope134/scope < runPowerFlow.mac

  -----------------------------------------------------------------------------

   Nexant Inc.                                                   (C) 1985-2016
                     SECURITY CONSTRAINED OPTIMAL POWER FLOW
                                    SCOPE 13.4
  -----------------------------------------------------------------------------


  Command :
  Command :
  Command :
  Command :
  Enter name of input file :
  Reading power system data ...

  +=+=+   28 warning message(s) listed on the message output file.

  Command :
  Command :
  Command :
  Command :
  Command :
  Solving power flow ...


 ==============================================================================
  NEWTON POWER FLOW CONVERGENCE
  BASE-CASE
 ------------------------------------------------------------------------------
                MAXIMUM MISMATCH       BUS  NOT MAT ARE DIS PHS  Q  TAP TRF SHU
 ITERATION    MW       MVAR     PU      ID  CVG FAC INT SLK LIM LIM LIM  Q   V
 ==============================================================================
     1       90.01                  712471  136  2
                      105.56        750035  413                  24  72
                              0.0313722384   68
                                                                             0
     2       -1.00                  722384    0  *
                      -15.20        722387   20                   0 128
                              0.1984722872  100
                                                                             0
     3      -58.11                  722872   72  *
                       56.06        122871   48                   0   0
                              0.0000733943    0
                                                                             0
     4       -5.98                  522872    8  *
                       -4.77        722872    8                   0   0
                              0.0000733943    0
                                                                             0
     5       -0.20                  522872    0  *
                       -0.11        522872    0                  24   4
                             -0.1519722416   72
                                                                             0
     6       -0.82                  722353    0  *
                       -4.28        722416   32                   0   0
                              0.0000733943    0
                                                                             0
 ------------------------------------------------------------------------------
     6        0.00                  750042    0  *
                        0.00        122763    0                   0   0
                              0.0000733943    0                              0


 ==============================================================================
  NEWTON POWER FLOW CONVERGENCE SUMMARY
  BASE-CASE
 ==============================================================================

  ACTIVATED OPTIONS : PCN NWT QLC TVC YVC
  NUMBER OF ITERATIONS :    6

 ------------------------------------------------------------------------------
  ISLAND   NO. OF
    NO.    BUSES        CONVERGENCE STATUS
 ------------------------------------------------------------------------------
     1    19965         POWER FLOW SOLVED

  Command :
  Options :
  Solving contingency cases ...

  +=+=+ 2533 warning message(s) listed on the message output file.

  Command :
  End of run

real    16m33.784s
user    15m44.066s
sys     0m47.374s
[ec2-user@ip-172-31-41-137 ~]$ ▮
```

Figure 6.3: N-1 Contingency Analysis for Transmission Lines and Generator Units

6.4.5 Multiple Processors

To increase the efficiency of their contingency analysis Nexant are able to split the execution across multiple processors, running several contingencies in parallel threads. This reduces the time taken, allowing more studies to be run when there may be time limitations and fully utilising the multiple cores found in modern processors. The benefit of using multiple processors is obvious, but this comes with a cost and the price may not justify any savings in execution time. The specific cloud instance and storage required to run a contingency analysis using multiple processors may be more expensive for a short time than the cost of a more basic cloud instance running a single processor for a longer period.

The commands for running SCOPE with multiple processors are:

```
$ /opt/nexant/intel/impi/intel64/bin/mpd --daemon
```

This command starts Intel's MPI Library service as a background *daemon*. The MPI library enables the multi-purpose daemon (mpd), which in turn allows the contingency analysis to be run using all the available processors.

```
$ time /opt/nexant/intel/impi/intel64/bin/mpiexec -n 7
/opt/nexant/scope134/scope < runPowerFlow.mac
```

This command then starts the SCOPE engine within a multi-processor execution wrapper that accesses the *mpd* daemon to split the execution across multiple threads. In this example, the power flow analysis runs with 7 processors, as it is for an 8-core machine, but the number '7' in the above command can be changed as required.

When the above commands were executed on an m4.2xlarge (8-core) machine instance [97] the contingency analysis ran in 4 minutes and 34.7 seconds; the results of this are shown in Figure 6.4. Running the contingency analysis on the same machine using a single processor took 16 minutes 50.85 seconds, meaning that using 7 processors was 3.7 times faster.

Contingency analysis using multiple processors was run on the m4 family of machines, with both 32 GB and 64 GB of extra storage, using the appropriate number of processors for the number of machine cores. The number of parallel processors used is the number of cores minus one, as the remaining core is required for operating system functionality and coordination of the other cores. The results can be seen in Table 6.2 and Table 6.3. The m4 machines were chosen because they are General Purpose instances, providing a good balance of compute, memory, and network resources [97]. A column has been included to display the approximate speedup of executing the contingency analysis for a second time.

Table 6.2: m4 Machines with 32 GB Storage

Machine Type	No. of Cores	Time Taken - first run (s)	Time Taken - after removing temporary files (s)	Approximate Speedup
m4.large	2	2418.78	96 4.59	2.5
m4.xlarge	4	2154.38	57.07	3.8
m4.2xlarge	8	1921.73	277.18	6.4
m4.4xlarge	16	1403.03	275.32	5.1
m4.10xlarge	40	2330.46	283.18	7.8
m4.16xlarge	64	2273.63	304.46	7.6

Table 6.3: m4 Machines with 64 GB Storage

Machine Type	No. of Cores	Time Taken - first run (s)	Time Taken - after removing temporary files (s)	Approximate Speedup
m4.large	2	1595.68	979.83	1.6
m4.xlarge	4	1285.4	558.41	2.3
m4.2xlarge	8	873.98	243.23	3.6
m4.4xlarge	16	595.97	133.66	4.6
m4.10xlarge	40	78.22	73.91	1.1
m4.16xlarge	64	68.28	67.33	1.0

It can be seen in Table 6.3 that as the number of cores in a machine increases, the time taken for both the initial run through and the run after removing temporary files becomes almost identical. The simulation time is also faster than the time taken by machines with fewer cores. Although the time to remove temporary files increases, as there is one file for every processor used to run the contingency analysis, this is no longer a necessary part of the process. These results suggest that if it is a choice between 40 or 64 cores there is minimal difference, so a 40-core machine should be used.

6.4.6 EBS Volume Allocation Issue

A complication that was discovered while running the contingency analysis with multiple processors is that the time taken to allocate and initialise storage space can impact greatly on the time taken to run a simulation. Running the test model with 29,084 contingencies could use up to 64 GB of EBS volume space to store temporary files and results. On AWS, when an instance is initialised from a pre-existing image its storage is allocated on an as-needed basis irrespective the amount requested [98]. For example, with the Alpha instance under 2 GB of storage was being used for the operating system and power flow software, so even if the instance was created with 64 GB of allocated storage, the cloud platform initialised only 2 GB of storage then initialised anything beyond 2 GB as requested.

This issue was not present when the instances were manually set up from a standard Linux image (e.g. Amazon Linux or Red Hat Linux) as 'New EBS volumes receive their maximum performance the moment that they are available and do not require initialisation' [98]. Since the purpose of this work was to automate execution, it was necessary to create standard images that could automatically execute the data with the SCOPE engine and orchestration framework software pre-installed.

The problem encountered was that the allocation process was found to be relatively slow. When an 8 GB stream of zero values was written to a temporary file with the command:

```
dd if=/dev/zero of=/home/ec2-user/myfile.TMP bs=1M count=8192
```

The output of the command was:

```
8192+0 records in
8192+0 records out
8589934592 bytes (8.6 GB) copied, 495.077 s, 17.4 MB/s
```

This shows that it took 8 minutes and 15 seconds with a speed of 17.4 MB/s, significantly slower than was expected. This temporary file was removed with the command:

```
rm myfile.TMP
```

The previous command was run again, on the same instance, but with an output of:

```
8192+0 records in
8192+0 records out
8589934592 bytes (8.6 GB) copied, 53.163 s, 162 MB/s
```

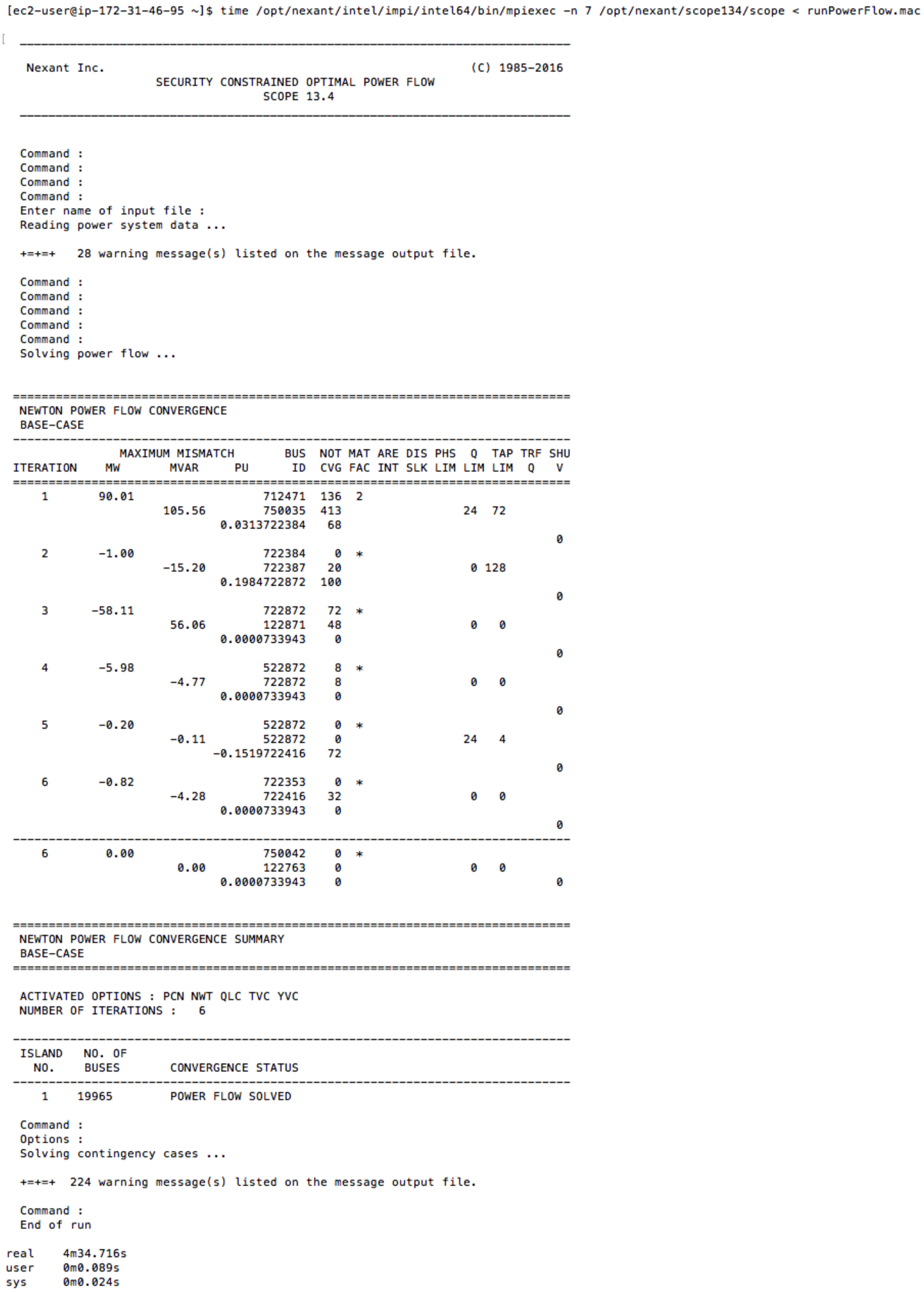
The same command now takes only 53 seconds to write the same amount of data on the same volume. This is because the volume space has already been allocated and the write speed is limited only by the underlying storage medium. For a process that writes up to 32 GB of memory this can add 30 minutes and explains the slow initial runs when a custom image is used.

One solution to this is an initialisation process known as pre-warming. To pre-warm the instance, data needs to be written and then read from the instance before use thus ensuring the volume space is allocated and can then be accessed at full speed.

This can take a long time, but as it only needs to occur the first time a storage volume is used the time demands are often overlooked as it is seen as necessary. Pre-warming a 256 GB storage volume can take over 4 hours making it unsuitable for using the instances on-demand. The solution that was used in this research was to create another image of the Alpha AMI that had already been used but instead of removing the temporary files prior to creating the image, they were retained as part of the image. This way when the instance was created all 256 GB of storage was in-use and thus allocated during start-up. Allocating the volume space during start-up did not result in a measurable time penalty. When three instances were created, one with 32 GB of unallocated storage, one with 256 GB of unallocated storage and one with 256 GB of pre-allocated storage, all three started up in the same time (205, 203 and 198 seconds respectively, or 201.5 seconds ± 3.5 seconds - a deviation of 1.74 %).

The temporary files stored on this image were then deleted prior to execution of the power flow to free up the space for running and storing the results of the contingency analysis. Although this action seems to be an inconvenience, the process can be removed from human requirements by inserting it within a script that is run upon starting an instance. This ensures that it will take the same time for every instance and can be removed from further benchmark results and calculations.

It turned out that the initialisation solution was not ideal, as the results in Table 6.2 show. For an m4.xlarge, 4-core cloud setup with 32 GB of extra storage, the simulation took 35 minutes 54.38 seconds to run. As the expected time was less than 16 minutes this result was unexpected. Temporary files on the instance were removed and the simulation was run again; this time it finished in 9 minutes and 30.07 seconds, which is more reasonable. The reason for the extensive first runtime is still not fully known, but it is suspected to be a combination of swap space and volume storage initialisation. When the same instance type was tested with 64 GB of extra storage (shown in Table 6.3), the first run time decreased by approximately 14.5 minutes while the run time after removing temporary files is similar.



```
[ec2-user@ip-172-31-46-95 ~]$ time /opt/nexant/intel/impi/intel64/bin/mpiexec -n 7 /opt/nexant/scope134/scope < runPowerFlow.mac
[ -----------------------------------------------------------------------------

  Nexant Inc.                                                     (C) 1985-2016
                    SECURITY CONSTRAINED OPTIMAL POWER FLOW
                                   SCOPE 13.4
 -----------------------------------------------------------------------------


 Command :
 Command :
 Command :
 Command :
 Enter name of input file :
 Reading power system data ...

 +=+=+   28 warning message(s) listed on the message output file.

 Command :
 Command :
 Command :
 Command :
 Command :
 Solving power flow ...


==============================================================================
 NEWTON POWER FLOW CONVERGENCE
 BASE-CASE
------------------------------------------------------------------------------
               MAXIMUM MISMATCH       BUS  NOT MAT ARE DIS PHS  Q  TAP TRF SHU
ITERATION    MW       MVAR     PU       ID  CVG FAC INT SLK LIM LIM LIM  Q   V
==============================================================================
    1       90.01                  712471  136  2
                     105.56        750035  413                 24  72
                             0.0313722384   68
                                                                            0
    2       -1.00                  722384    0  *
                     -15.20        722387   20                  0 128
                             0.1984722872  100
                                                                            0
    3      -58.11                  722872   72  *
                      56.06        122871   48                  0   0
                             0.0000733943    0
                                                                            0
    4       -5.98                  522872    8  *
                      -4.77        722872    8                  0   0
                             0.0000733943    0
                                                                            0
    5       -0.20                  522872    0  *
                      -0.11        522872    0                 24   4
                            -0.1519722416   72
                                                                            0
    6       -0.82                  722353    0  *
                      -4.28        722416   32                  0   0
                             0.0000733943    0
                                                                            0
------------------------------------------------------------------------------
    6        0.00                  750042    0  *
                       0.00        122763    0                  0   0
                             0.0000733943    0                              0


==============================================================================
 NEWTON POWER FLOW CONVERGENCE SUMMARY
 BASE-CASE
==============================================================================

 ACTIVATED OPTIONS : PCN NWT QLC TVC YVC
 NUMBER OF ITERATIONS :    6

------------------------------------------------------------------------------
 ISLAND   NO. OF
   NO.    BUSES       CONVERGENCE STATUS
------------------------------------------------------------------------------
    1    19965        POWER FLOW SOLVED

 Command :
 Options :
 Solving contingency cases ...

 +=+=+  224 warning message(s) listed on the message output file.

 Command :
 End of run

real    4m34.716s
user    0m0.089s
sys     0m0.024s
```

Figure 6.4: Multiple Processor Contingency Analysis

6.4.7 Split Contingency List

Another option to consider is splitting the contingencies into multiple lists. In theory, a smaller contingency analysis list will run faster, allowing X lists to be run in parallel on X machines in a shorter overall time which may reduce the overall cost.

This was tested by splitting the contingencies into 3 lists, and running each one simultaneously on a separate, but identical, cloud instance. Each list was limited to 10,000 contingency entries. The longest time taken for any of the three machines to complete the contingency analysis is recognised as the result and will be the value used to calculate costs. The results of split contingency list testing are in Table 6.4. These results show that reducing the number of contingencies in a list helps to overcome the memory restrictions mentioned in 6.4.4. As there are less analysis tasks for the cloud instance to deal with it requires less memory and therefore less storage to complete the analysis. This could be improved further by splitting the contingencies into more lists – meaning fewer contingencies in each list – and this is done as part of the automated power flow in a cloud in Chapter 7.

Table 6.4: m4 Machines with Split Contingency List

Number of Cores	Time Taken for the Longest Cloud Instance (s)			
	32 GB Storage		64 GB Storage	
	first run	removed temp files	first run	removed temp files
2	539.09	353.96	643.88	364.71
4	470.93	209.06	394.81	208.01
8	220.79	116.74	176.74	93.96
16	58.30	49.09	58.27	58.18
40	31.12	28.44	31.35	29.11
64	32.01	24.64	30.19	24.67

6.5 N-1 Discussion

The results used in this section are those recorded after removing temporary files from an instance. It is assumed that the machine will already have been used to perform contingency analysis and is being woken from hibernation each time it is required.

6.5.1 Cores vs. Time

As seen in Table 6.3, the time taken to run an N-1 contingency analysis levels out as the number of cores in the machine running it increases. Figure 6.5 displays the contents of Table 6.3 in a scatter graph. The blue line represents the time taken to run the contingency analysis for the first time, and the orange line is the time taken to run the contingency analysis again. The time is in seconds and each marker on the lines corresponds to the number of cores in the machine. The graph shows that increasing the number of cores in a machine is beneficial to decreasing the run time.

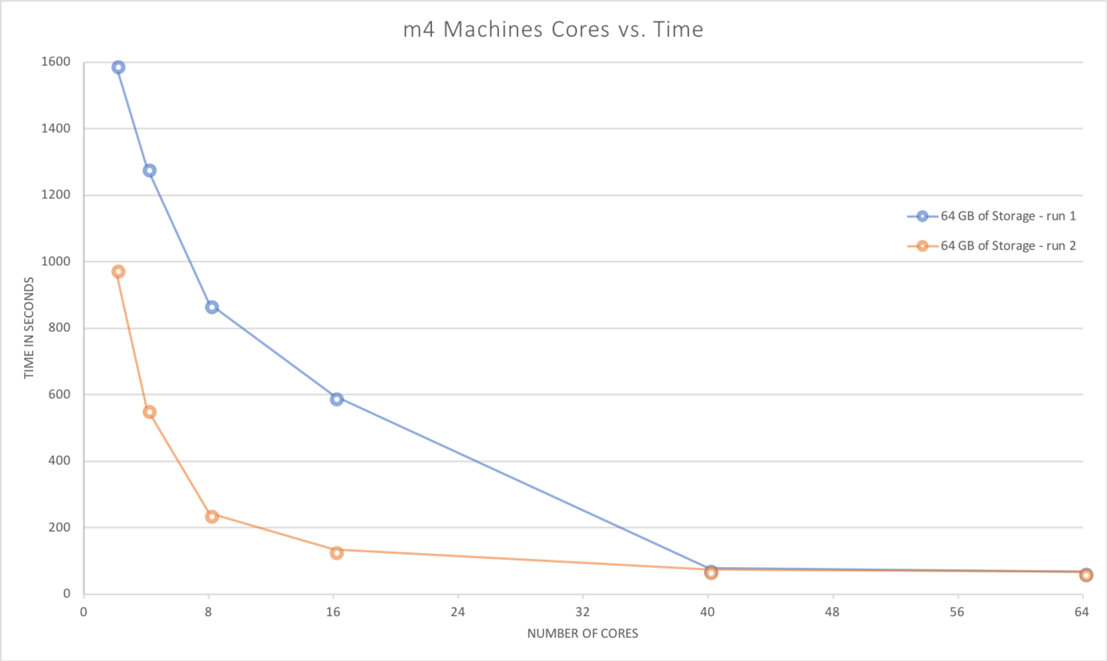


Figure 6.5: Cores vs. Time for N-1 Contingency Analysis

6.5.2 Storage vs. Time

The graph in Figure 6.6 displays the differences between having 32 GB and 64 GB of extra storage on an instance. The times are in seconds and have been calculated from the fourth columns of Table 6.2 and Table 6.3. The resulting figure shows that the amount of storage used can affect the performance of machines with 16 or more cores, by an average of 200 seconds.

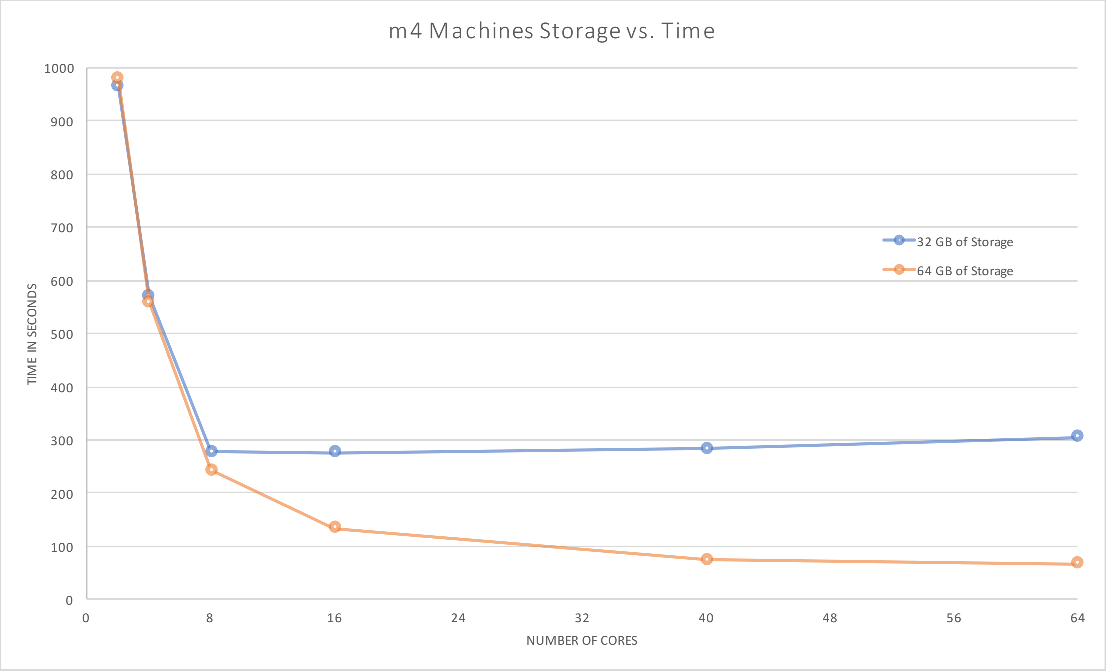


Figure 6.6: Storage vs. Time for N-1 Contingency Analysis

6.5.3 Cost Analysis

Cost should be taken into consideration when there is a choice of all machine types and therefore number of cores. To calculate the price per simulation, the run times after removing temporary files recorded in 6.4.5 were converted into hours and then multiplied by the cost per hour of the machine used. The cost per hour is taken from the Instance Pricing page on the AWS website [99] and the results are shown in Table 6.5.

Table 6.5: Simulation Costs for m4 Machines with 32 GB and 64 GB Storage

Machine Type	Number of Cores	Cost/hour ($)	Price per Simulation - 32 GB Storage ($)	Price per Simulation - 64 GB Storage ($)
m4.large	2	0.111	0.030	0.030
m4.xlarge	4	0.222	0.035	0.034
m4.2xlarge	8	0.444	0.034	0.023
m4.4xlarge	16	0.888	0.068	0.033
m4.10xlarge	40	2.22	0.175	0.046
m4.16xlarge	64	3.552	0.300	0.066

The cost of extra storage was not added to the prices calculated as it is deemed negligible. According to [100] the price per GB-month for an EBS general purpose SSD storage volume in the EU (Ireland) region is $0.11. Taking into account the 60 second minimum billing increment implemented by AWS, the increase in price for a 64 GB volume would be $0.000163 per minute. A 32 GB storage volume would cost $0.000082 per minute. As all recorded times being used to calculate costs are under 17 minutes, the price of extra storage is under $0.00277 and will not have an impact on the overall results.

6.5.4 Comparison – Time

Table 6.6 confirms that it is faster to split a contingency list and run it in parallel on separate machine instances than to run it as one larger list. The final column shows the maximum time taken to run a split contingency list as a percentage of the time taken to analyse the full contingency list. On average, using split contingency lists in parallel takes 35 % of the time taken to analyse the full list. This is as expected; there are 3 contingency lists consequently taking around a third of the time.

Table 6.6: Time Comparison of Full and Split Contingency Lists with 64GB Storage

Machine Type	Number of Cores	Full Contingency List (s)	Split Contingency List (s)	Split Time / Full Time (%)
m4.large	2	979.823	364.71	37.22
m4.xlarge	4	558.41	208.01	37.25
m4.2xlarge	8	243.23	93.96	38.63
m4.4xlarge	16	133.66	58.18	43.87
m4.10xlarge	40	73.91	29.16	39.38
m4.16xlarge	64	67.33	24.67	36.64

6.5.5 Comparison – Cost

The price per simulation for running split contingency lists in parallel was calculated in the same way as before, with the cost being calculated for the longest running cloud instance and multiplied by 3. This provides the maximum cost, although it is acknowledged that the total price will be calculated for each instance and will cost slightly less. Table 6.7 contains the simulation costs for instances with 64 GB storage, after removing the temporary files. Again, the price for extra storage was deemed negligible.

Table 6.7: Simulation Costs for m4 Machines with Split Contingency Lists

Machine Type	Number of Cores	Time Taken (s)	Cost for 1 Instance ($)	Total Cost ($)
m4.large	2	364.71	0.011	0.034
m4.xlarge	4	208.01	0.013	0.039
m4.2xlarge	8	93.96	0.012	0.035
m4.4xlarge	16	58.18	0.015	0.043
m4.10xlarge	40	29.16	0.018	0.054
m4.16xlarge	64	24.67	0.024	0.073

Table 6.8 displays the cost of running 1 machine instance with the full contingency list, compared to running 3 parallel instances each with one of 3 smaller contingency lists. The final column is the difference in cost between the two list types. It can be

seen for the majority of machine types that it costs less than $0.01 more to run the simulation with multiple contingency lists.

Table 6.8: Cost Comparison of Full and Split Contingency Lists

Machine Type	Number of Cores	Full Contingency List ($)	Split Contingency List ($)	Split – Full ($)
m4.large	2	0.030	0.034	0.0035
m4.xlarge	4	0.034	0.039	0.0041
m4.2xlarge	8	0.023	0.035	0.0118
m4.4xlarge	16	0.033	0.043	0.0101
m4.10xlarge	40	0.046	0.054	0.0082
m4.16xlarge	64	0.066	0.073	0.0066

6.5.6 N-1 Conclusions

It is faster to run multiple parallel instances each with a smaller contingency list, and it will only cost $0.01 more. It is also more efficient to use 64 GB storage to increase the run time, although this stops being beneficial above a 40-core machine.

The results in this chapter suggest that the best option for running contingency analysis on a cloud platform is to use multiple 40-core machines with 64 GB of storage in parallel, with contingency lists split among machines.

6.6 N-2 Contingencies

An N-2 contingency covers the loss of two contingency events at the same time. For the purposes of this research an N-2 contingency will be considered as the loss of two transmission lines or other network elements at the same time. The total number of N-2 transmission line and generator unit contingencies on the 21,000-bus network is 422,924,986.

Due to the cost constraints of running 422,924,986 N-2 contingency analysis on AWS, a Monte Carlo method was applied: running the contingency analysis on several smaller, randomised samples of data and then extrapolating. This is covered in Chapter 7.

6.7 N-1-1 Contingencies

An N-1-1 contingency covers the loss of a single contingency event, which may be comprised of a loss of one of more network elements, after the system has adjusted for a previous loss (N-1).

Due to the required system adjustments it was not feasible to simulate N-1-1 in the confines of this research. Although the capability to retrieve the output files of the contingency analysis is there, this work is focused on execution of the power flow and contingency analysis in a cloud environment and the optimal conditions for doing this rather than the complex network analysis required to then determine how to formulate an N-1-1 contingency based on the results. It is also not generally possible to generate a complete N-1-1 contingency list automatically, as it requires input from engineers with knowledge of the system's vulnerabilities and behaviour.

SCOPE contingency analysis is typically executed as a preliminary step before the execution of the optimal active or reactive power flow. This is one of the reasons that temporary files are created with each analysis: they are save/restore files for each contingency that can be used later on during the optimisation process. Another important output from the contingency analysis is the summary of violated constraints and a note of the contingency case that caused the largest violation.

There are a number of avenues for looking at N-1-1 including:

- Centralised interpretation of the contingency results: every contingency is returned centrally to be analysed together, new contingencies are formulated based on the results and then dispatched to the running cloud instances for analysis
- Distributed interpretation of the contingency results: each cloud instance analyses the results of the contingencies it has processed and identifies new contingencies that should be analysed

Centralised interpretation has the downside of removing the parallelisation and network bandwidth. This could be an issue if the contingency results for 10,000+ cases need to be serialised and retrieved from the server. For example, a compressed network model file in the native PCA format for the 21,000-bus model used is over 1 MB (uncompressed it is approximately 15 MB). The benefit is that all of the contingencies can be analysed together rather than only within the context of the set sent to the server.

Distributed interpretation removes the need to transmit all the results back for centralised processing, potentially making it faster to build and then run the N-1-1 cases. For this to work efficiently the set of contingencies sent to each server cannot be randomly created, but instead must be organised to ensure that it covers a set of equipment that is in electrical proximity where the failure of more than one device is likely to have an adverse effect.

6.8 Concluding Remarks

This chapter has researched the viability of running a cloud-based power flow simulation and contingency analysis. Using Nexant's SCOPE engine, a 21,000-bus network model was analysed in an AWS cloud environment. An N-1 contingency set was generated for transmission lines and generator units, chosen for the impact they have on the operation of the network. Testing was done to find the optimal cloud conditions.

Simulations were run on a range of cloud instances and with different set-ups for the contingency analysis. The number of machine cores ranged from 2 – 64, and the amount of storage on a machine was tested between 32 GB and 64 GB. Contingency analysis was run on one processor, on multiple processors depending on the number of cores available, and as 3 separate lists of fewer contingencies on parallel machines. After a discussion of the options it was concluded that the best option for running contingency analysis on a cloud platform is to use multiple 40-core machines with 64 GB of storage in parallel, with contingency lists split among machines.

Chapter 7 Automated Power Flow in a Cloud

7.1 Introduction

It was obvious while running the contingency analyses in Chapter 6 that the work was not being carried out in an ideal way. There was too much user input: manually starting each machine instance and setting it up correctly after it had started was inefficient and time consuming. To advance the work done previously the steps taken to run power flow analysis in a cloud environment were automated. This was done by wrapping the processes into a Scope Cloud Client and a Scope Cloud Server, which require only a basic user input before starting the required number of instances and running the appropriate contingency analysis scenario. It removes the potential for user error, and also decreases the overall time taken as the manipulation of the cloud environment is automatically done upon start up.

The objective of this chapter is to confirm the results obtained previously and determine if they can be improved upon. Further investigation into the optimal settings for running a cloud-based power flow will include running a different type of AWS machine with the same settings and inputs. Similar research will be carried out for N-2 contingency analysis, forecasting the cost of running it on a real-world power network.

7.2 Scope Cloud Client and Scope Cloud Server

7.2.1 Overview

The process that was used in Chapter 6 to reach the endpoint of running the contingency analysis on an AWS instance was automated. Every step that was taken from working out the contingencies, to starting an instance, to running SCOPE, was wrapped into a more compact procedure and automated. The main elements of this are the Scope Cloud Client and the Scope Cloud Server.

When run, the Scope Cloud Client will build the contingencies for the input file. This is still focused only on transmission lines and generators, as it is the same code that was previously used but in a new format. It creates contingency sets for N-1 and N-2.

The Scope Cloud Client will start up AWS instances as requested by the user. Each instance will be started using the same AMI, which is already set up with SCOPE and has 256 GB of storage. The storage has been pre-allocated, so one of the start-up tasks for each instance will be to remove these files. This solves the volume storage allocation issue discussed in 6.4.6. Another stage in the start-up of each instance is to enable multiple processors to be used. The number of processors used will be worked out as one less than the number of cores in the machine.

The Scope Cloud Client will create a Scope Cloud Server instance. This is a web application server that runs SCOPE in the AWS instances. Security was taken into consideration when automating this process. The Scope Cloud Client only sends the commands ‘startInstances’, ‘checks done?’, and ‘terminateInstances’ to AWS. All files are sent directly to the Scope Cloud Server and they are encrypted for further security.

7.2.2 Process

Figure 7.1 shows the UML sequence diagram for the process of running power flow in a cloud environment.

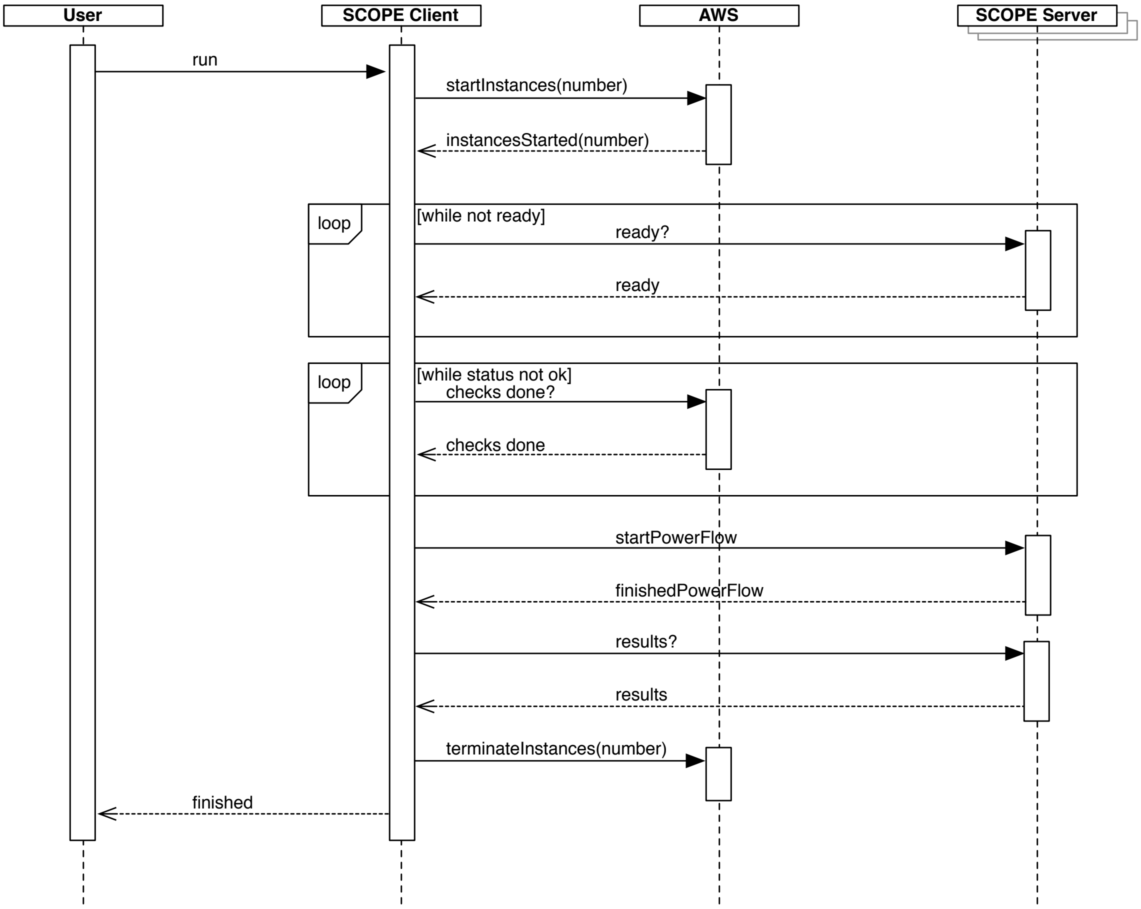


Figure 7.1: UML Sequence Diagram for Scope Cloud Client and Scope Cloud Server

The Scope Cloud Client is the interaction point with users. The parameters are in the form:

```
{InstanceType.M4Large.toString(), "n1", "3", “10000”,
"/Users/corinne/entsoe21.pca", BASE+"/n1-32i.csv"}
```

Where the inputs are:

Instance type	`InstanceType.M4Large`
Contingency analysis type	`"n1"`
Number of required instances	`"3"`
Number of contingencies to run on each instance	`“10000”`
Input file with network data	`"/Users/corinne/entsoe21.pca"`
Output file for results	`"/n1-32i.csv"`

In the above example there are 3 m4.large machines being used to run N-1 contingency analysis on the file `/Users/corinne/entsoe21.pca` with 10,000 contingencies on each instance, and the results are saved to the file `/n1-32i.csv`.

When instructed by the user, the Scope Cloud Client will contact AWS and start up the required number of instances with a suitable, pre-set AMI that contains the correct SCOPE engine and analysis software. The Scope Cloud Client will initiate the Scope Cloud Server to check if the AWS instances are ready. It will continue to check until all the instances are available. To allow for the small variance in start-up times for AWS instances, the Scope Cloud Client will check with AWS to find out if all the instances have completed their start-up checks. The next stage will not start until all the instances are ready for use. The next step is for the Scope Cloud Client to tell the Scope Cloud Server to start the power flows, including how many contingencies are to be run on each instance, which sends the split contingency data to the relevant instance. When the power flow is complete the Scope Cloud Server will alert the Scope Cloud Client. The client then asks for the results, and once they have been received it tells AWS to terminate all instances. The Scope Cloud Client will tell the user details of each instance as it finishes, and that it has terminated all the instances.

There is the option of the user requesting multiple power flows to be run simultaneously. This would result in the Scope Cloud Client starting multiple Scope Cloud Servers, each one focused on a separate power flow request and following the same procedure as above. Only the start-up checks within that server will determine when it can run the power flow, but the AWS instances will not be terminated until the contingency analysis for all requests is complete.

7.3 N-1 Confirmation of Previous Work

7.3.1 Start Up Time

The average start-up time for an AWS instance was calculated to be 4 minutes and 26 seconds. The start-up time for an instance was calculated by removing the run time of the power flow and contingency analysis from the total time of the cloud instance being used. This is shown in Table 7.1.

Table 7.1: m4 Machines with Start Up Time

Machine Type	Number of Cores	Number of Instances	Total Time (s)	Run Time (s)	Start-Up Time (s)
m4.large	2	1	1866	1593.7	272.3
		2	939	667.73	271.27
		4	591	324.26	266.74
		8	497	188.86	308.14
		16	410	102.98	307.02
m4.xlarge	4	1	1660	1454.92	205.08
		2	788	552.53	235.47
		4	473	205.1	267.9
		8	385	112.99	372.01
		16	310	65.05	244.95
m4.2xlarge	8	1	1494	1261.1	232.9
		2	643	403.3	239.7
		4	350	113.73	236.27
		8	329	59.0	270
m4.4xlarge	16	1	1114	849.21	264.79
		2	355	89.86	265.14
		4	360	61.81	298.19
m4.10xlarge	40	1	373	107.23	265.77
		2	300	64.24	235.76
m4.16xlarge	64	1	386	87.93	298.07
		2	287	51.63	235.37

The time to remove temporary files from a machine was recorded for a single instance of each type; this has been included in Table 7.2. The average time to remove temporary files is 18 seconds. This reduces the average start up time to 4 minutes and 8 seconds.

Table 7.2: Single Instance m4 Machines with Start Up Times

Machine Type	Number of Cores	Total Time (s)	Run Time (s)	Remove Files (s)	Start-Up Time (s)
m4.large	2	1866	1593.7	18.12	254.18
m4.xlarge	4	1660	1454.92	18.46	186.62
m4.2xlarge	8	1494	1261.1	16.89	216.01
m4.4xlarge	16	1114	849.21	17.55	247.24
m4.10xlarge	40	373	107.23	20.7	245.07
m4.16xlarge	64	386	87.93	16.98	281.21

For the purpose of this research the time to remove temporary files and any system processes not included in the run time of the SCOPE engine, including the start-up time of an instance, will be assumed to be constant. These processes cannot be changed or improved upon. For the remainder of the chapter the time that will be used to assess machine efficiency is run time. The total time will still be included in the results tables because it is needed for cost analysis. Although the start-up time for every instance can be assumed to be the same, the price per instance is different and therefore the cost of the start-up time will be different.

7.3.2 Single Instance

The Scope Cloud Client ran contingency analysis on a single instance of each of the m4 family of machines. The results can be seen in Table 7.3.

Table 7.3: Single Instance m4 Machines

Machine Type	Number of Cores	Total Time (s)	Run Time (s)
m4.large	2	1866	1593.7
m4.xlarge	4	1660	1454.92
m4.2xlarge	8	1494	1261.1
m4.4xlarge	16	1114	849.21
m4.10xlarge	40	373	107.23
m4.16xlarge	64	386	87.93

Table 7.4 shows a comparison of the results in Table 7.3 with the results of a single instance with 64 GB of storage that are recorded in Table 6.3. The results are similar, but show that using the Scope Cloud Client is noticeably slower for the 4 and 8 core machines. The reason for this is unknown, but the most likely explanation is the difference in execution environment. The Scope Cloud Client instantiates a Scope Cloud Server instance, uploads and runs the power flow analysis, and then terminates the instance. The instances used in Chapter 6 were instantiated then used multiple times each with multiple, manual invocations of the power flow. As demonstrated in Table 6.2 and Table 6.3, the performance of an instance tended to improve from the first run, often seeing improvements in the second and third run even after the issue of volume storage allocation was resolved. The underlying architecture of AWS machines is not directly visible, so it is not possible to know the exact reason for the difference in performance. These results will have an influence on future design decisions, as the results indicate that performance is improved when the analysis is executed on an instance that is already running.

Table 7.4: Single Instance m4 Machines Comparison

Machine Type	Number of Cores	Chapter 7 Run Time (s)	Chapter 6 Run Time (s)
m4.large	2	1593	1596
m4.xlarge	4	1455	1285
m4.2xlarge	8	1261	874
m4.4xlarge	16	849	596
m4.10xlarge	40	107	78
m4.16xlarge	64	88	68

7.3.3 Split Contingency List

The Scope Cloud Client was used to run contingency analysis on the m4 family of machines with a split contingency list. The list was automatically split between the 3 requested instances of each machine type, meaning each instance had approximately 10,000 contingencies to analyse (the same as in 6.4.7). The results can be seen in Table 7.5.

Table 7.5: Split Contingency List m4 Machines

Machine Type	Number of Cores	Total Time (s)	Run Time (s)
m4.large	2	685	413.31
m4.xlarge	4	621	350.95
m4.2xlarge	8	377	141.65
m4.4xlarge	16	304	69.38
m4.10xlarge	40	325	56.27
m4.16xlarge	64	343	47.8

Table 7.6 shows a comparison of the results in Table 7.5 with the results of splitting the contingencies into 3 lists and running each one on a separate, but identical, cloud instance that are recorded in Table 6.4. Again the results are similar.

Table 7.6: Split Contingency List m4 Machines Comparison

Machine Type	Number of Cores	Chapter 7 Run Time (s)	Chapter 6 Run Time (s)
m4.large	2	413.31	643.88
m4.xlarge	4	350.95	394.81
m4.2xlarge	8	141.65	176.74
m4.4xlarge	16	69.38	58.27
m4.10xlarge	40	56.27	31.35
m4.16xlarge	64	47.8	30.19

7.4 N-1 m4 Machines

7.4.1 Overview

Table 7.7 shows the results of running contingency analysis using the Scope Cloud Client on all practical combinations of machine types/cores and number of instances in the m4 machines family. When the time taken to run the contingency analysis on each machine type was reduced to around 5 minutes there could be very little improvement due to start up times, and it was not run again with more instances. For the m4.large machine there was an AWS instance limit of 20 instances. Although the m4.4xlarge machine did not lower to a time of 5 minutes, the price of running 4 instances was sufficiently high to discount running it with 8 instances. The total cost was calculated using the same method as 6.5.3.

Table 7.7: m4 Machines Contingency Analysis

Machine Type	Cores	Number of Instances	Cost/ hour ($)	Total Time (s)	Run Time (s)	Cost / Instance ($)	Total Cost ($)
m4.large	2	1	0.111	1866	1593.7	0.058	0.06
		2		939	667.73	0.029	0.06
		4		591	324.26	0.018	0.07
		8		497	188.86	0.015	0.12
		16		410	102.98	0.013	0.20
m4.xlarge	4	1	0.222	1660	1454.9	0.102	0.10
		2		788	552.53	0.049	0.10
		4		473	205.1	0.03	0.17
		8		385	112.99	0.024	0.19
		16		310	65.05	0.019	0.31
m4.2xlarge	8	1	0.444	1494	1261.1	0.184	0.18
		2		643	403.3	0.079	0.16
		4		350	113.73	0.043	0.17
		8		329	59.0	0.041	0.32
m4.4xlarge	16	1	0.888	1114	849.21	0.275	0.28
		2		355	89.86	0.088	0.18
		4		360	61.81	0.089	0.36
m4.10xlarge	40	1	2.22	373	107.23	0.23	0.23
		2		300	64.24	0.19	0.37
m4.16xlarge	64	1	3.552	386	87.93	0.381	0.38
		2		287	51.63	0.28	0.57

The average cost to run a contingency analysis is $0.22. The closest results to this are running m4.large with 16 instances and running m4.10xlarge with 1 instance. These have been highlighted in red.

The average time to run a contingency analysis is 401 seconds. The closest result to this is running m4.2xlarge with 2 instances. This has been highlighted in green.

Running 2 instances of m4.16large is the fastest, but it is also the most expensive. Running 1 of the simplest m4.large instances is the cheapest option, but it is also one of the slowest.

From the results in Table 7.7, it can be speculated that the best option is running 4 m4.2xlarge instances, taking a total time of 5 minutes and 50 seconds and costing $0.17. Due to start up time of an instance the fastest that the contingency analysis can be run is around 5 minutes, and this is the cheapest option for that time. This 'best option' has been highlighted in blue.

It was noticed that the results obtained when running the contingency analysis on more than one instance of the same machine type are increasing by more than is expected. For example, the time taken to run one instance of the m4.xlarge completes in 1454.9 seconds. The time taken to run the contingency analysis on two m4.xlarge instances is 552.53 seconds. This is 3 times faster instead of the expected 2 times faster. The reason for this is unknown, but repeating these contingency analyses produced the same results and it can be seen in Table 7.7 that it was similar for other machine types. A possible explanation for these results is that the memory allocation or storage space of an instance is reaching its limit towards the later contingencies when all 29,084 are being run in a single list.

When the contingency list is split it lowers the number of contingencies to be analysed on each instance, which will reduce the memory required to process them. If a process can avoid using the storage volume for either swap space or temporary files it offers a significant saving in time. Volume storage performance, even at 162 MB/sec (as discussed in 6.4.6) is still an order of magnitude slower than memory read/write, which is over 20 GB/sec for even the slowest memory configuration used with the m4 instance types [101].

Discussions with the power flow vendor confirmed that reducing the number of contingencies would have a corresponding reduction in memory requirements and storage operations. This is a variable that cannot be tested within the boundaries of this research as the memory configurations on the instance types are fixed, however it highlights that the amount of memory available in each instance can have a more dramatic impact on the performance than the addition of more processor cores.

7.4.2 Outlier

It was observed that the result of the m4.10xlarge machine running a single instance was inconsistent with the other configurations from the m4 family. The results of all machine types running a single instance have been plotted in the scatter graph in Figure 7.2.

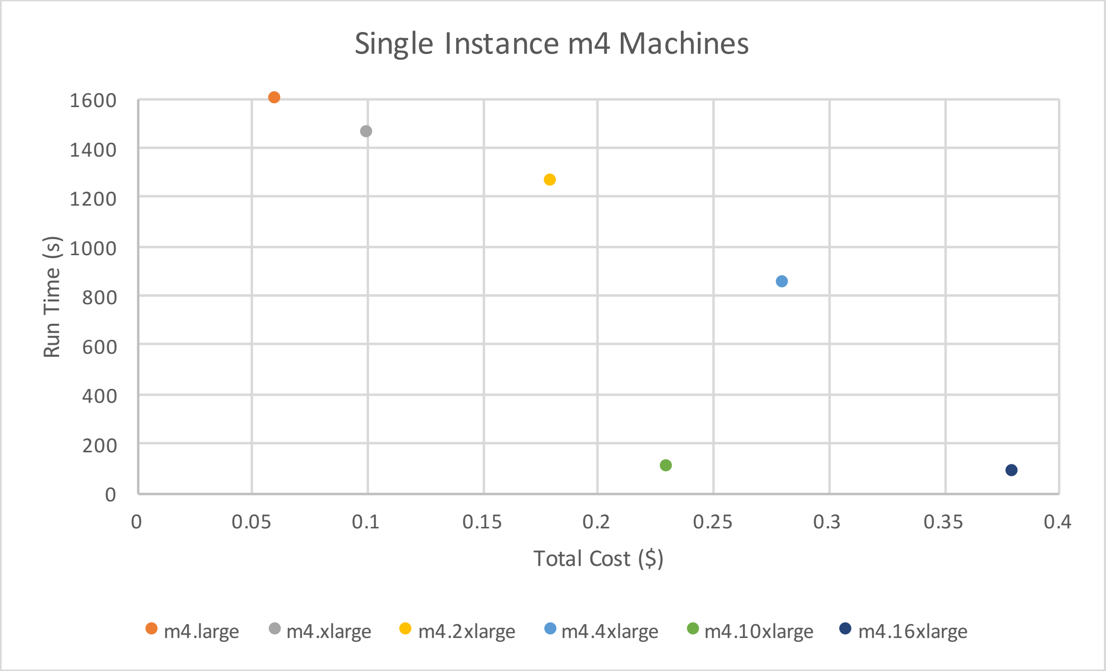


Figure 7.2: Single Instance m4 Machines

Attention is drawn to the green dot, which represents a single instance of an m4.10xlarge machine. It shows that running a single instance of an m4.10xlarge machine is almost as fast as an m4.16xlarge machine, but costs less than an m4.4xlarge machine. The AWS website does not specifically mention the m4.10xlarge machine having better performance than the rest of the m4 machines.

Table 6.1 shows the specifications of each machine type. The only difference between m4.10xlarge and any of the smaller machines is in the Network Performance column. According to AWS, network performance indicates the performance level of the rate of data transfer [102] but the power flow process does not use the network interface other than to receive the data, and the payload is sufficiently small that this time is negligible across all instances. All other specifications increase proportionately to the number of cores the machine has, whereas the network performance increases from 'High' to '10 Gigabit'. Using a machine with 10 Gigabit network connectivity will ensure reliable network bandwidth between an EC2 instance and an EBS volume [103], but the value of 'High' network connectivity is not stated.

One theory is that the memory allocation of the m4.10xlarge and m4.16xlarge instances, being 160 GB and 256 GB respectively, may have increased to a point where the power flow engine could optimise outputs to ignore volume storage input/output during its run. As was covered in 6.4.6 the volume input/output, even after pre-allocation, is still relatively slow. If this was being reduced it could account for the significant drop in execution time.

7.4.3 Improvements

The uncertainty of the result from running a single m4.10xlarge instance prompted a software update. The Scope Cloud Client was improved to record the average of all instances of the same type. For example, if running 4 m4.large instances the Scope Cloud Client would record the run time for each instance but provide them to the user at the end as an average. Previously the user was recording the longest of the 4 times and using that for all results tables and calculations. Being more aware of the slight differences in the performance of each instance and the risk of user error also prompted the contingency analysis to be run 3 times for each core/instance combination. The average of the 3 average run times is now used in results tables and for further calculations.

The new results of running contingency analysis on all practical combinations of machine types/cores and number of instances in the m4 machines family are shown in Table 7.8.

Table 7.8: m4 Machines Improved Contingency Analysis

Machine Type	Number of Cores	Number of Instances	Average Total Time (s)	Average Run Time (s)	Average Total Cost ($)
m4.large	2	1	1594	1316.48	0.05
		2	848	497.68	0.05
		4	638	246.82	0.08
		8	674	132.39	0.17
		16	655	68.39	0.32
		32	614	45.53	0.61
m4.xlarge	4	1	1427	1165.02	0.09
		2	722	413.50	0.09
		4	515	149.62	0.13
		8	551	84.83	0.27
		16	586	49.39	0.58
m4.2xlarge	8	1	1265	1030.73	0.16
		2	613	282.77	0.15
		4	438	75.57	0.22
		8	476	45.30	0.47
m4.4xlarge	16	1	956	671.19	0.24
		2	419	84.04	0.21
		4	406	48.76	0.40
m4.10xlarge	40	1	382	90.69	0.24
		2	427	54.27	0.53
m4.16xlarge	64	1	379	80.64	0.37
		2	382	48.98	0.75

The new results for the single m4.10xlarge instance are better than the original, proving that it is indeed an outlier in terms of price/performance. This is shown in a scatter graph of all machine types running a single instance in Figure 7.3. The results for running a single instance of the 8 and 16 core machines are similar to those recorded in 7.3.2, which suggests that if there is an issue causing the discrepancy in Table 7.4 it occurred during the analysis in the previous chapter. The improvement to the Scope Cloud Client was beneficial to the integrity of this research by providing more detailed output and granularity.

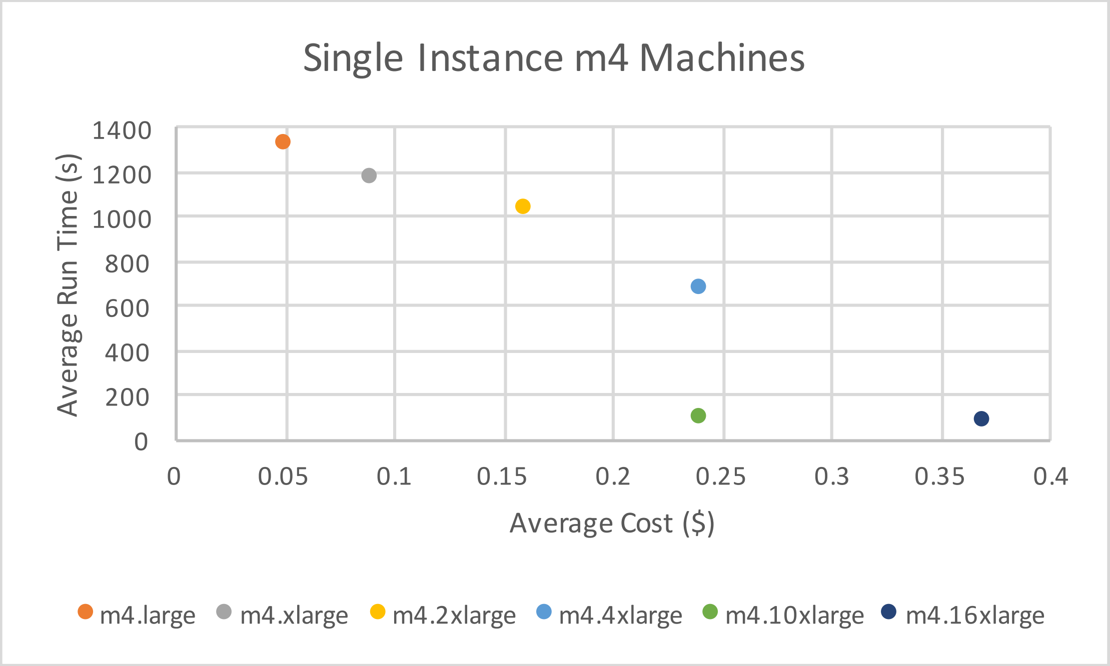


Figure 7.3: Improved Single Instance m4 Machines

7.4.4 Results – Cost

The improved results for the m4 machine instances were sorted by the total cost. This is shown in Table 7.9. The average cost to run a contingency analysis is $0.28. The closest result to this is running m4.xlarge with 8 instances. This has been highlighted in red. This is slightly increased from the previous results (Table 7.7) by $0.06.

7.4.5 Results – Time

The improved results for the m4 machine instances were then sorted by average run time. This is shown in Table 7.10. The average time to run a contingency analysis is 304 seconds. The closest results to this are running m4.2xlarge with 2 instances and running m4.xlarge with 2 instances. These have been highlighted in green. The average run time has decreased by 97 seconds from 7.4.1. It is assumed that running the contingency analysis multiple times and taking averages has produced more accurate results.

Table 7.9: m4 Machines Contingency Analysis Sorted by Cost

Machine Type	Number of Cores	Number of Instances	Average Total Time (s)	Average Run Time (s)	Average Total Cost ($)
m4.large	2	1	1594	1316.48	0.05
m4.large	2	2	848	497.68	0.05
m4.large	2	4	638	246.82	0.08
m4.xlarge	4	1	1427	1165.02	0.09
m4.xlarge	4	2	722	413.5	0.09
m4.xlarge	4	4	515	149.62	0.13
m4.2xlarge	8	2	613	282.77	0.15
m4.2xlarge	8	1	1265	1030.73	0.16
m4.large	2	8	674	132.39	0.17
m4.4xlarge	16	2	419	84.04	0.21
m4.2xlarge	8	4	438	75.57	0.22
m4.4xlarge	16	1	956	671.19	0.24
m4.10xlarge	40	1	382	90.69	0.24
m4.xlarge	4	8	551	84.83	0.27
m4.large	2	16	655	68.39	0.32
m4.16xlarge	64	1	379	80.64	0.37
m4.4xlarge	16	4	406	48.76	0.4
m4.2xlarge	8	8	476	45.3	0.47
m4.10xlarge	40	2	427	54.27	0.53
m4.xlarge	4	16	586	49.39	0.58
m4.large	2	32	614	45.53	0.61
m4.16xlarge	64	2	382	48.98	0.75

Table 7.10: m4 Machines Contingency Analysis Sorted by Run Time

Machine Type	Number of Cores	Number of Instances	Average Total Time (s)	Average Run Time (s)	Average Total Cost ($)
m4.2xlarge	8	8	476	45.3	0.47
m4.large	2	32	614	45.53	0.61
m4.4xlarge	16	4	406	48.76	0.4
m4.16xlarge	64	2	382	48.98	0.75
m4.xlarge	4	16	586	49.39	0.58
m4.10xlarge	40	2	427	54.27	0.53
m4.large	2	16	655	68.39	0.32
m4.2xlarge	8	4	438	75.57	0.22
m4.16xlarge	64	1	379	80.64	0.37
m4.4xlarge	16	2	419	84.04	0.21
m4.xlarge	4	8	551	84.83	0.27
m4.10xlarge	40	1	382	90.69	0.24
m4.large	2	8	674	132.39	0.17
m4.xlarge	4	4	515	149.62	0.13
m4.large	2	4	638	246.82	0.08
m4.2xlarge	8	2	613	282.77	0.15
m4.xlarge	4	2	722	413.5	0.09
m4.large	2	2	848	497.68	0.05
m4.4xlarge	16	1	956	671.19	0.24
m4.2xlarge	8	1	1265	1030.73	0.16
m4.xlarge	4	1	1427	1165.02	0.09
m4.large	2	1	1594	1316.48	0.05

7.4.6 Results – Best Option (m4 Machines)

The results from Table 7.8 for each number of instances run on all machine types were plotted in the graph in Figure 7.4 in the form of average cost against average run time. This is similar to Figure 7.3 but includes running multiple instances. The colour of each machine type has been kept the same for continuity.

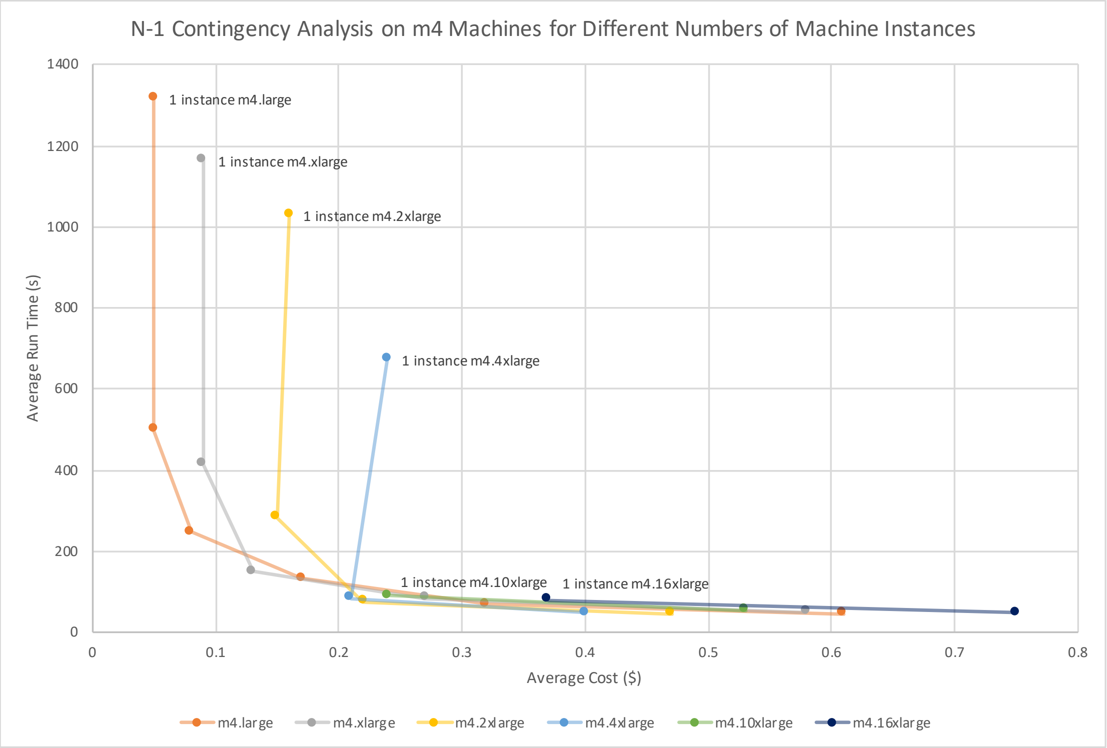


Figure 7.4: N-1 Contingency Analysis on m4 Machines

The results from each machine type have been connected by a line to make the graph more readable. This also highlights that the average time will eventually level out at around 50 seconds, but the cost will continue to increase as more instances are used.

The average cost where the majority of instances plateau in terms of time is just above $0.20. This is true for the m4.xlarge, the m4.2xlarge, the m4.4xlarge and the m4.10xlarge. Of these machine types, a single instance of m4.10xlarge is the best option in terms of price/performance.

7.5 N-1 c4 Machines

7.5.1 Overview

To further the research of running power flow and contingency analysis in a cloud environment, the N-1 process was repeated with a different family of AWS machine instances. The same version of Scope Cloud Client was used as in 7.4.3 and the same numbers of instances were run on the corresponding machine type in terms of number of cores. Table 7.12 shows the results of running contingency analysis on the c4 machines family.

The c4 machines were chosen because they are Compute Optimized, whereas the m4 machines are General Purpose. AWS recommends Compute Optimized machines for intensive workloads to deliver cost effective high performance at a low price-per-compute ratio [97].

The total cost was calculated using the Amazon EC2 Pricing [104]. A summary of the price per hour for each c4 machine type is shown in Table 7.11.

Table 7.11: c4 Machines Pricing

Machine Type	Number of Cores	Instance Price / Hour ($)
c4.large	2	0.113
c4.xlarge	4	0.226
c4.2xlarge	8	0.453
c4.4xlarge	16	0.905
c4.8xlarge	36	1.811

Table 7.12: c4 Machines Contingency Analysis

Machine Type	Number of Cores	Number of Instances	Average Total Time (s)	Average Run Time (s)	Average Total Cost ($)
c4.large	2	1	1660	1379.46	0.05
		2	842	516.81	0.05
		4	553	260.77	0.07
		8	578	113.21	0.15
		16	638	60.57	0.32
		32	630	39.49	0.63
c4.xlarge	4	1	1589	1328.45	0.10
		2	794	509.20	0.10
		4	515	179.47	0.13
		8	536	85.40	0.27
		16	592	44.52	0.59
c4.2xlarge	8	1	1529	1264.49	0.19
		2	755	437.30	0.19
		4	493	112.07	0.25
		8	434	46.46	0.44
c4.4xlarge	16	1	1427	1139.05	0.36
		2	629	324.40	0.32
		4	443	45.72	0.45
c4.8xlarge	36	1	1130	835.45	0.57
		2	374	53.47	0.38

7.5.2 Results – Cost

The results for the c4 machine instances were sorted by the total cost. This is shown in Table 7.13. The average cost to run a contingency analysis is $0.28. The closest result to this is running c4.xlarge with 8 instances. This has been highlighted in red.

This result is the same as the result of sorting the contingency analysis of m4 machine instances by cost in 7.4.4. The average of those results suggested running an m4.xlarge machine, which also has 4 cores, with 8 instances, the same as the average here.

Table 7.13: c4 Machines Contingency Analysis Sorted by Cost

Machine Type	Number of Cores	Number of Instances	Average Total Time	Average Run Time (s)	Average Total Cost ($)
c4.large	2	1	1660	1379.46	0.05
c4.large	2	2	842	516.81	0.05
c4.large	2	4	553	260.77	0.07
c4.xlarge	4	1	1589	1328.45	0.1
c4.xlarge	4	2	794	509.2	0.1
c4.xlarge	4	4	515	179.47	0.13
c4.large	2	8	578	113.21	0.15
c4.2xlarge	8	1	1529	1264.49	0.19
c4.2xlarge	8	2	755	437.3	0.19
c4.2xlarge	8	4	493	112.07	0.25
c4.xlarge	4	8	536	85.4	0.27
c4.large	2	16	638	60.57	0.32
c4.4xlarge	16	2	629	324.4	0.32
c4.4xlarge	16	1	1427	1139.05	0.36
c4.8xlarge	36	2	374	53.47	0.38
c4.2xlarge	8	8	434	46.46	0.44
c4.4xlarge	16	4	443	45.72	0.45
c4.8xlarge	36	1	1130	835.45	0.57
c4.xlarge	4	16	592	44.52	0.59
c4.large	2	32	630	39.49	0.63

7.5.3 Results – Time

The improved results for the c4 machine instances were then sorted by average run time. This is shown in Table 7.14. The average time to run a contingency analysis is 439 seconds. The closest result to this is running c4.2xlarge with 2 instances. This has been highlighted in green.

Table 7.14: c4 Machines Contingency Analysis Sorted by Run Time

Machine Type	Number of Cores	Number of Instances	Average Total Time (s)	Average Run Time (s)	Average Total Cost ($)
c4.large	2	32	630	39.49	0.63
c4.xlarge	4	16	592	44.52	0.59
c4.4xlarge	16	4	443	45.72	0.45
c4.2xlarge	8	8	434	46.46	0.44
c4.8xlarge	36	2	374	53.47	0.38
c4.large	2	16	638	60.57	0.32
c4.xlarge	4	8	536	85.4	0.27
c4.2xlarge	8	4	493	112.07	0.25
c4.large	2	8	578	113.21	0.15
c4.xlarge	4	4	515	179.47	0.13
c4.large	2	4	553	260.77	0.07
c4.4xlarge	16	2	629	324.4	0.32
c4.2xlarge	8	2	755	437.3	0.19
c4.xlarge	4	2	794	509.2	0.1
c4.large	2	2	842	516.81	0.05
c4.8xlarge	36	1	1130	835.45	0.57
c4.4xlarge	16	1	1427	1139.05	0.36
c4.2xlarge	8	1	1529	1264.49	0.19
c4.xlarge	4	1	1589	1328.45	0.1
c4.large	2	1	1660	1379.46	0.05

This result is similar to the result of sorting the contingency analysis of m4 machine instances by run time in 7.4.4. One of the suggested options from the average of those results was running an m4.2xlarge machine, which also has 8 cores. The main difference is that the average run time for m4 machines was 135 seconds faster.

7.5.4 Results – Best Option (c4 Machines)

To check if there were any outliers among the c4 machines a scatter graph of all machine types running a single instance is in Figure 7.5. It shows the expected pattern of a decrease in run time coupled with an increase in cost as the number of cores in a machine increases.

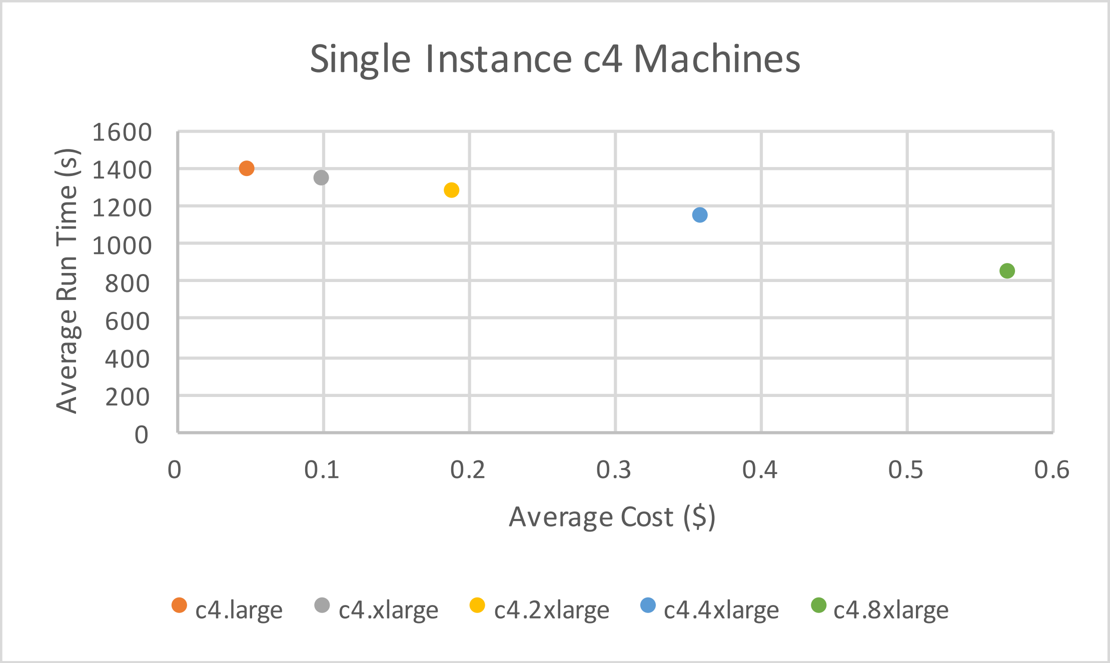


Figure 7.5: Single Instance c4 Machines

The results from Table 7.12 for each number of instances run on all c4 machine types were plotted in the graph in Figure 7.6 in the form of average cost against average run time. The colour of each machine type has been kept the same as in Figure 7.5 for continuity, and they have also been kept in the same order as the m4 machine types.

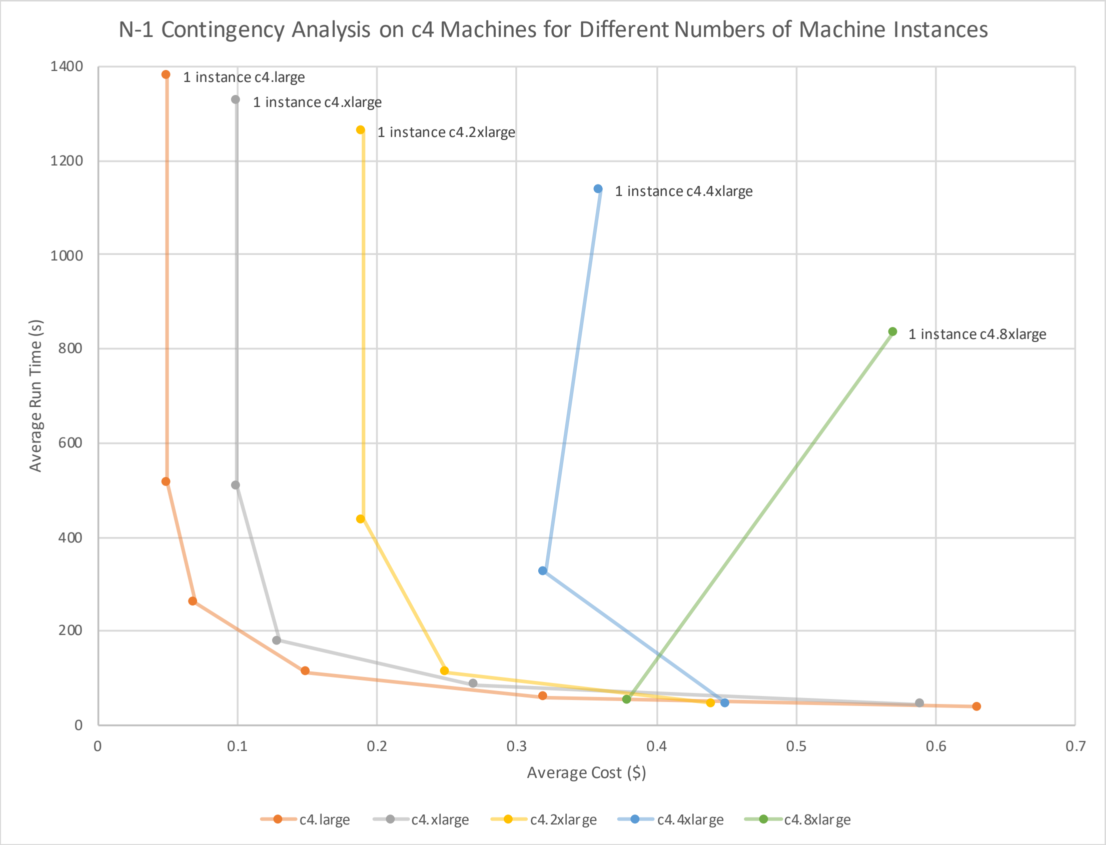


Figure 7.6: N-1 Contingency Analysis on c4 Machines

The results from each machine type have been connected by a line to make the graph more readable. Again, this highlights that the average time will eventually level out, but that the cost will continue to increase as more instances are used.

There are two points of interest in Figure 7.6. The first is where the machines with fewer cores start to plateau in terms of time: around 80 seconds and costing $0.27. This is true for the c4.xlarge and the c4.2xlarge, with the c4.large close behind at $0.32. The other point to note is that the c4.8xlarge machine has a significant jump in performance when 2 instances are run; the run time decreases by 13 minutes and the total cost also decreases by the significant amount of $0.19. As the time for running 2 instances of the c4.8xlarge machine is 53 seconds, and the total cost is $0.38, it is the favourable option when compared to the plateau point for the majority of the c4 machine types.

7.6 N-1 Summary

The machine type and number of instances that was proposed as the best option for the m4 family and the c4 family have been compared in Table 7.15. The results are similar.

Table 7.15: Best Option Comparison

	m4.10xlarge	c4.8xlarge
Number of Cores	40	36
Price per Instance ($)	2.222	1.811
Number of Instances	1	2
Average Run Time (s)	90.69	53.47
Average Total Cost ($)	0.24	0.38

The m4.10xlarge machine has 4 more cores than the c4.8xlarge machine, but it is only running 1 instance whereas the c4.8xlarge is running 2. The number of instances required makes the m4 machine the better option because it costs less. The c4 machine is faster, but only by 37 seconds and this is not fast enough to justify the extra cost of $0.14.

7.7 Extended N-1

In order to look at how this approach could be used with more realistic N-1 contingencies for a large, real-world network, the 29,084 contingencies were expanded to cover additional contingencies involving more than one component being offline. This was accomplished by analysing the network topology and component parameters in the form of:

- Parallel transmission lines and the faults caused by two or more of them going offline: for 2 parallel lines looking at a case of both being taken out of service; for 3 lines looking at the permutations of 2 or 3 being out of service etc.); 2,076 parallel lines were identified

- Two generators over a certain size going offline simultaneously: the largest 1,460 generators representing approximately 30 % of those on the system were used with a contingency for any two of them going offline

This created 1,067,750 contingencies for the network. The work detailed in 7.4.3 indicated that the m4.10xlarge instance type was the best choice in terms of price and performance, so the Scope Cloud Client was configured to split the network across 36 instances of these machines with 29,660 cases per instance.

As with the other benchmarks there was a delay while waiting for the instances to start ranging from 223 seconds to 283 seconds. 35 of the 36 instances started in 223 - 224 seconds; only one instance had an unusual, outlier time of 283 seconds. As the setup was configured to wait for all instances to be ready before starting the next section this meant that the next stage of power flow invocation waited for all 36 instances to be ready. The instances then ran in parallel with execution times across the 36 instances ranging from 124 seconds to 160 seconds.

The total execution time from the Scope Cloud Client starting was 477 seconds (7 minutes and 57 seconds). This included waiting for the instances, splitting the contingency cases, transmitting the data, waiting for the analysis and obtaining the results.

Running this extended N-1 contingency analysis demonstrated that the platform could be used to solve large numbers of pre-defined contingencies from the same network in parallel, using 1,440 processor cores split across 36 separate instances.

7.8 Amdahl’s Law and Gustafson’s Law

Parallelisation of contingency analysis in the cloud has so far conformed to the theory behind Amdahl’s Law: that the speedup seen by an increase of parallel resources will be constrained by any serial ‘housekeeping’ tasks.

Amdahl’s Law is paraphrased as:

$$Speedup = \frac{1}{r_s + \frac{r_p}{n}} \qquad (8)$$

Where $r_s + r_p = 1$ and r_s represents the ratio of the sequential portion in one programme [105]. This can be understood in relation to this research where r_s is equal to the start-up time of an instance, r_p is the runtime and n is the number of instances. It means that there can be any amount of parallel instances run to increase the speed of contingency analysis but the total time will always be limited by the time required to start each instance. Instance start-up is what Amdahl refers to as ‘housekeeping’ because it happens in series; all instances must have completed their start-up checks before the Scope Cloud Client will allow any of them to proceed with running power flows. This can be seen in Figure 7.4 and Figure 7.6, where the graphs level out with regards to time as the runtime decreases with added parallel instances but the start-up time remains the same.

Gustafson’s Law questions the validity of Amdahl’s Law outwith academic research, because the size of a problem will increase with the number of parallel processors used [106]. This suggests linear scalability, as the amount of work that can be done in parallel will vary linearly with the number of processors that are used. It also suggests that larger problems can be solved in the same amount of time if the appropriate resources are available.

7.9 N-2 Contingency Analysis

7.9.1 Overview

As mentioned in 6.6 an N-2 contingency covers the loss of two contingency events at the same time, but for this research will be considered as the loss of two transmission lines or other network elements at the same time. There are 422,924,986 N-2 transmission line and generator unit contingencies on the 21,000-bus network that is being used for testing.

Running all the N-2 contingencies on AWS is outwith the budget for this research, so they will be tested using the Monte Carlo method. A number of randomly selected N-2 contingencies will be run using the Scope Cloud Client and Scope Cloud Server, and these will be used to estimate the total time and cost for running the complete set of N-2 contingencies.

As it was proposed as the best option for N-1 contingency analysis the m4.10xlarge machine will be used for running the N-2 contingency analysis. An example of the input into the Scope Cloud Client is:

```
{InstanceType.M410xlarge.toString(),    "n2",    "50",    "1500000",
"/Users/corinne/entsoe21.pca", BASE+"/n2-50i.csv"}
```

This is requesting N-2 analysis on 50 instances of the m4.10xlarge machine, with a total of 1,500,000 contingencies to be split amongst them. It will save the results to the file `/n2-50i.csv`. The contingencies that are analysed are chosen randomly each time to ensure a fair result.

7.9.2 Results

The N-2 contingency analysis was run 5 times on 5 different m4.10xlarge cloud arrangements. The results are shown in Table 7.16.

Table 7.16: N-2 Contingency Analysis on m4.10xlarge Machines

Number of Instances	Number of Contingencies	Total Time (s)	Average Run Time (s)	Total Cost ($)
5	150,000	374	100.43	1.15
10	300,000	380	96.18	2.34
20	600,000	350	95.38	4.32
25	750,000	394	95.67	6.07
50	1,500,000	491	95.25	15.13

These results have been displayed as a scatter graph of number of instances against total cost in Figure 7.7. The predicted linearity can be seen clearly. Although it would be improved by including results from running between 25 and 50 instances this was not feasible due to cost constraints.

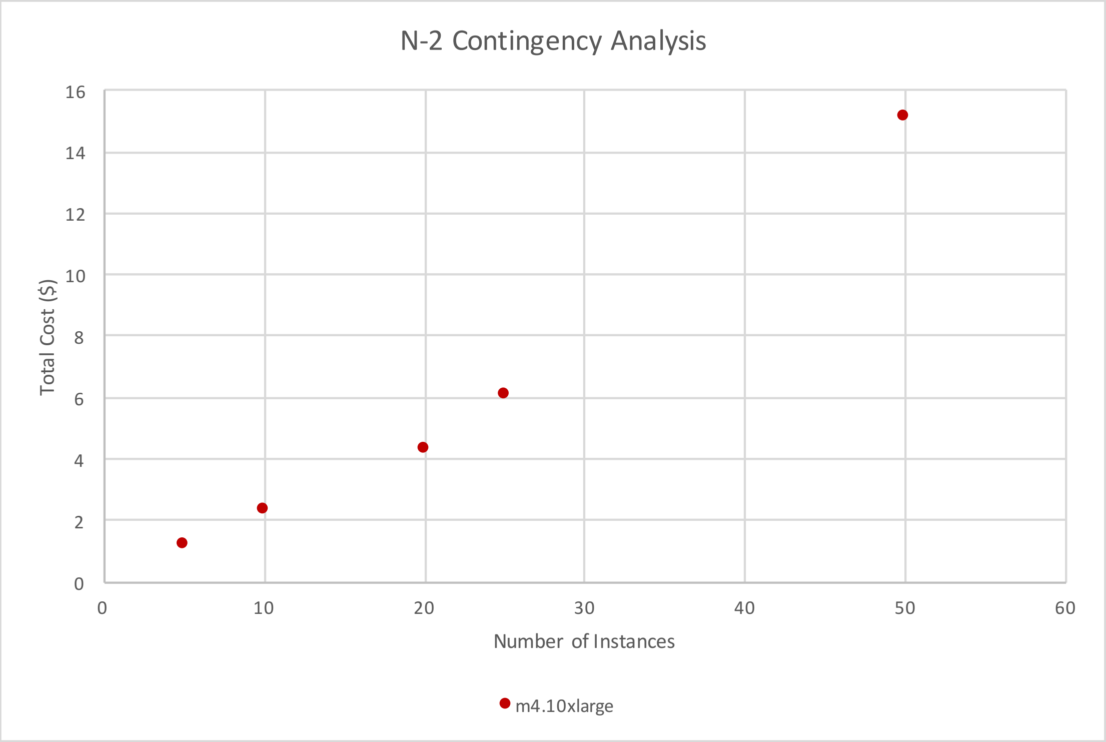


Figure 7.7: N-2 Contingency Analysis on m4.10xlarge Machines

7.9.3 N-2 Conclusions

The linear scalability predicted by Gustafson's Law [106] can be seen in Figure 7.7.

Parallelisation is demonstrated by the fact that the average run time for all variations of running N-2 contingency analysis on m4.10xlarge machines is 100 seconds ± 5 %.

The version of SCOPE being used is limited to 30,000 contingencies per instance. Since there are 422,924,986 N-2 contingencies it would require 14,098 machines to analyse them all. Using Microsoft Excel a linear forecast to 14,098 machines was added to the results graph; this is shown in Figure 7.8. It predicts that it would cost around $4,400 to run a complete N-2 contingency analysis on the 21,000-bus network. Extrapolating the results of running 50 instances of the m4.10xlarge machine supports this:

$$Cost\ to\ run\ a\ single\ instance = \frac{total\ cost}{number\ of\ instances} = \frac{15.13}{50} = \$\ 0.3026$$

$$Cost\ to\ run\ 14{,}098\ instances = Cost\ to\ run\ a\ single\ instance * 14{,}098 \tag{9}$$

$$= \ 0.3026 * 14{,}098 = \$\ 4{,}266$$

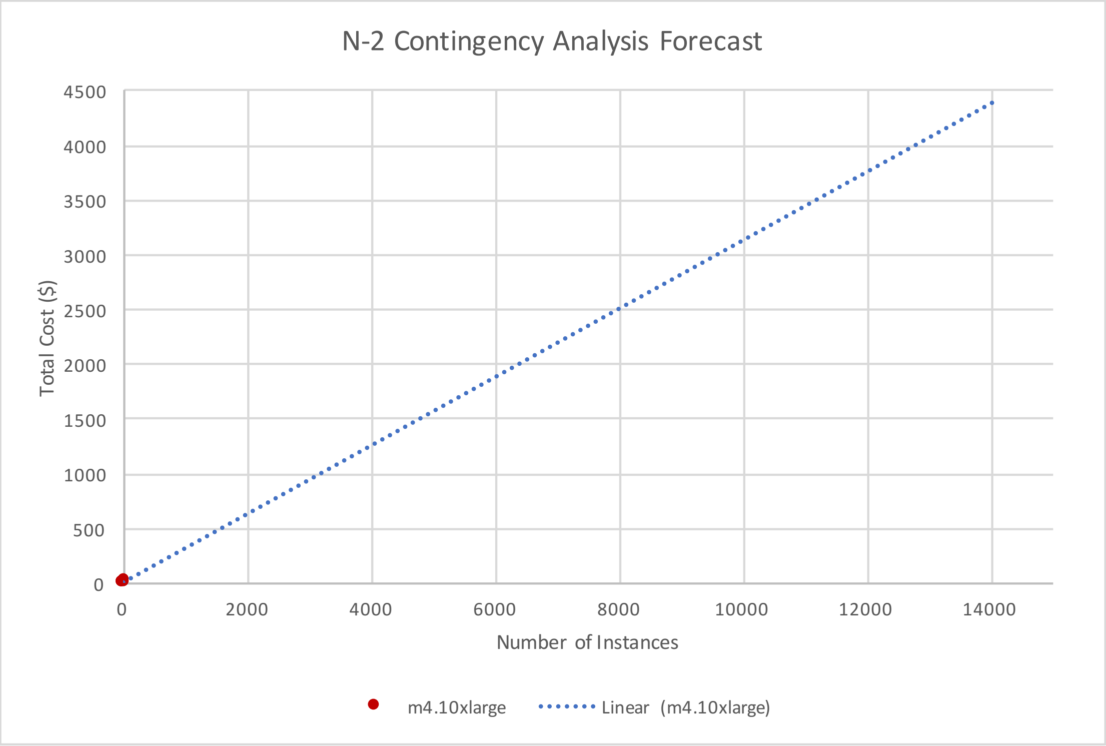


Figure 7.8: N-2 Contingency Analysis Forecast on m4.10xlarge Machines

It is theoretically possible to analyse all 422,924,986 N-2 contingencies in 100 seconds ± 5 % for approximately $4,400, not including the start-up time of an instance. When the start-up time of approximately 300 seconds is added to this, and time is included for the retrieval of results, the time taken to analyse all 422,924,986 N-2 contingencies would be under 10 minutes.

However, AWS may not allow this as it requires starting up and running over 14,000 instances and it is unclear how many instances are available to any one user. It is likely that commercial operators may have access to such large quantities, but AWS does not advertise this. If this is not available, the possible options are to split the instances across regions and therefore across AWS data centres or to arrange a commercial agreement with AWS. Assuming it is not possible to run 14,098 AWS instances, the solution would be to run multiple sets of contingency analysis on each instance. For example: an instance would start up, run a contingency analysis set and return the results, then receive a new contingency analysis set to run. This will take longer but may be cheaper and is definitely more feasible in terms of commercial viability.

If the price per hour for an m4.10xlarge instance = C_I = \$2.22:

Average run time = T_R = 100 seconds

Average start up time = T_S = 300 seconds

Number of instances = I

Contingency analyses run per instance = $A = \frac{14{,}098}{I}$

The total time for a contingency analysis would be:

$$T_{total} = (T_R * A) + T_S \qquad (10)$$

The total cost for a contingency analysis would be (dividing T_{total} by 3,600 to get the time in hours):

$$C_{total} = I * \left(C_I * \frac{T_{total}}{3600}\right) \qquad (11)$$

If the total number of instances is I = 141:

A = 100

$$\begin{aligned} T_{total} = (T_R * A) + T_S &= (100 * 100) + 300 = 10{,}300\, seconds \\ &= 2\ hours\ 52\ minutes \end{aligned} \qquad (12)$$

$$C_{total} = I * \left(C_I * \frac{T_{total}}{3600}\right) = 141 * \left(2.22 * \frac{10{,}300}{3600}\right) = 141 * 6.35 = \$896 \qquad (13)$$

Although this will take almost 3 hours longer, the price has decreased by a significant amount, equating to a fall of almost 80 % over the initial cost estimate. This is because the start-up time is the most expensive part of the process and this will only happen once per instance. Depending on the circumstances, a utility using this method to run contingency analysis could choose whether to optimise by speed or by cost, as it often will not be necessary to have results back in under 2 minutes.

If the total number of instances is I = 940:

A = 15

$$T_{total} = \ (T_R * A) + \ T_S = \ (100 * 15) + 300 = 1{,}800\, seconds \\ = 30\, minutes \tag{14}$$

$$C_{total} = I * \left(C_I * \frac{T_{total}}{3600}\right) = 940 * \left(2.22 * \ \frac{1{,}800}{3600}\right) = 940 * 1.11 = \$1{,}043 \tag{15}$$

Increasing the number of instances to 940 would decrease the total run time by over 2 hours, while only increasing the price by $147. If more instances were available to a utility then they would have a choice between the speed of the contingency analysis or the cost. In this case $147 seems to be a good price for the drastic reduction in time.

If the total number of instances is I = 50:

A = 281

$$T_{total} = \ (T_R * A) + \ T_S = \ (100 * 281) + 300 = 28{,}400\, seconds \\ = 7\ hours\ 53\ minutes \tag{16}$$

$$C_{total} = I * \left(C_I * \frac{T_{total}}{3600}\right) = 50 * \left(2.22 * \ \frac{28{,}400}{3600}\right) = 50 * 17.36 = \$867 \tag{17}$$

Reducing the number of instances to 50 has very little impact on cost of running the contingency analysis on 141 instances, but increases the time taken by 5 hours. Again this suggests that if the option is available then more instances should be used.

These results show that it is much more possible to run an affordable N-2 contingency analysis in a reasonable amount of time using the cloud than it was when it had to be run on a single machine or a number of parallel in-house computers that were limited by the price of a full machine.

7.10 Future Work

This research has dealt with single contingencies, as opposed to contingency events that include one or more elements going offline. An extension of this work is to incorporate the detection and analysis of events, as demonstrated in 7.7, by using more sophisticated network analysis techniques. This will also be utilised in the N-2 contingency analysis.

A method to incorporate N-1-1 contingency analysis within the cloud environment will be investigated in collaboration with the commercial power flow vendor. Although they have stated that in their experience it is generally not possible to fully automate the generation of N-1-1 contingencies, it may be possible to use the computational power provided by a cloud computing platform to leverage network analysis, data analytics and machine learning techniques to generate the majority of the N-1-1 contingencies. This would reduce the manual workload required for cases that are difficult to automatically identify without specialist knowledge of individual network configurations.

From a commercial standpoint it would be beneficial to standardise the process. It could be offered to clients as ‘Power Flow as a Service’ and would remove the need for paying per software license and the processing restrictions caused by vendors implementing per-machine or per-core licensing policies. It would be possible for a vendor to purchase their own high performance machines to be used in-house for running their own analysis, but which could also be rented out to clients using a private cloud infrastructure. Utility clients are still wary of using public cloud platforms. They are more likely to accept off-site processing of sensitive data on a cloud that is operated by an existing, trusted vendor and supplier of power system analysis software. This cloud will already have undergone the pre-requisite regulatory background and security checks. They would then be charged per instance and usage time as with public cloud infrastructures, with the option for ‘surge pricing’ during busy times to limit demand or cover costs of additional infrastructure.

7.11 Concluding Remarks

This chapter has confirmed and improved upon the work that was previously completed on N-1 contingency analysis. Using Nexant's SCOPE engine wrapped within a Scope Cloud Client a 21,000-bus network model was analysed in an AWS cloud environment by the Scope Cloud Server. Testing was done to discover the optimal conditions for running an N-1 contingency analysis in terms of machine family, number of cores and number of instances. The best option was proposed as running a single instance of an m4.10xlarge machine, which analysed 29,084 contingencies in approximately 91 seconds for a cost of $0.24.

This chapter also investigated N-2 contingency analysis within the budget constraints of the research. A variety of contingency analysis was run on m4.10xlarge machine instances, with the specific contingencies being randomly assigned to each instance when it was started. Due to the linear scalability of the results it can be estimated that all 422,924,986 N-2 contingencies can be analysed on 14,098 instances in less than 10 minutes for approximately $4,400.

Chapter 8 Conclusions & Future Work

8.1 Conclusions

The research in this thesis has investigated the design and development of novel, cutting-edge solutions to enable advanced power system analysis. It has explored and demonstrated the use of the cloud as a fast, scalable, secure and robust environment for analysing and storing complex power systems data.

Communication and data processing standards within the power industry were discussed in Chapter 3, within the context of reference architecture IEC 62357-1 (Figure 3.1). Types of data exchanges and the standards to support them are detailed for low frequency operations such as network planning models, high frequency communication between devices and electricity market operations.

Challenges surrounding data exchange within the power industry include the use of incompatible data structures, identification rules and communication protocols. Overcoming these challenges will enable the integration of data from different areas of the power network, with benefits for both real-time operation and offline network analysis. These include: enabling different systems and applications to automatically recognise that data is coming from the same device at the same point in the network but originating from different sources and conforming to different, incompatible standards; the ability to integrate real-time data with offline analysis tools; and utilising smart-meter data to enable true real-time pricing for electricity markets.

Standard interoperability is a common issue, as many overlapping standards within the power industry can cause devices and assets to have different names and identities within models and systems that represent the same thing but are operating at different levels of the network. One of the main complications in achieving integration across multiple levels is the need to automatically match common identifiers and convert between different data models and serialisation formats. This would provide a more detailed and accurate view of the power system by making it easier to share network information. It would enable utilities to assess an issue more quickly within the power system, allowing them to discover and understand what the problem is and where it has occurred.

As the power industry moves towards more active distribution networks there is a requirement for more complete visibility of the current state of the network. The current level of observability differs between the transmission network and the distribution network, and this was studied in Chapter 4.

Although transmission networks cover a larger geographical area, the amount of network data produced is significantly less than that of a distribution network. Due to the fact that a high voltage failure will have a greater impact on the operations of the network as a whole, the level of monitoring and analysis is greater at the transmission level. It also benefits from the availability of real-time results and accurate network models.

The main challenges in observing a distribution network are improving the existing communications infrastructure and maintaining satisfactory cyber security to encourage the installation of µPMUs in resource constrained areas. The benefits would be a lower volume of data with higher reliability than current metering options, with huge potential for improving monitoring and control of the network.

Chapter 5 presented a methodology for addressing the problems faced in deploying µPMUs at the distribution level. One of the principal contributions of this thesis is the use of remote data collection from µPMUs to support improvements to network models for analysis. Impedance equations were applied to real-world µPMU network data to prove the viability of using it to identify areas of the network where the analytical models differ from the as-built models. If a discrepancy is discovered during this analysis, or if a gradual change over time is detected, a more detailed survey of the network and its equipment would be undertaken to support future installation of localised generation or advanced control schemes.

Another principal contribution to research is the design of a cloud-based architecture for deploying µPMUs in geographically remote locations with unpredictable communication bandwidth and low levels of physical security. Communications technology was discussed in terms of securely installing µPMUs on the distribution network. Precautions must be taken in case a device is compromised. Using a cloud architecture to restrict communication between critical internal systems and a µPMU would limit what an attacker could accomplish to stopping the data feed or compromising the data being sent. It would prevent direct access to any control system.

Combining the cloud architecture with localised triggers implemented at the device level would significantly reduce the amount of data being sent by µPMUs while still supporting real-time network operations and notification of unusual system events. An example trigger of the phase voltage magnitude being ± 5% of nominal more than 100 times in a period of 10 seconds was demonstrated in 5.8, with the trigger-value being decreased to 4.5 % and 4 % to demonstrate the difference in the amount of data that µPMUs would send to system operators.

A novel approach to enable fast, detailed analysis of all possible contingencies on a network by using a highly parallelisable power flow based within a cloud environment was developed in Chapter 6 and Chapter 7. Using Nexant's SCOPE engine, a 21,000-bus network model was analysed in an AWS cloud environment. An N-1 contingency set was generated and simulations were run on a range of cloud instances with varying set-ups.

A principal contribution to research is the demonstration of challenges faced when using a commercial cloud platform to inexpensively solve computationally intensive power flow problems. These include the restriction of system resources, the allocation of volume storage space and the presence of results that are outwith the expected performance of a machine.

These challenges prompted the development of a Scope Cloud Client and Scope Cloud Server to execute a highly parallelised power flow with minimal human interaction and a standard start-up procedure and configuration, using the commercial power flow engine. This is another principal contribution of the research.

The cloud client and server application was used to investigate the optimal settings for running an N-1 contingency analysis for a real-world 21,000-bus transmission model in the cloud environment in relation to both execution time and the financial costs. Tests were run on a range of cloud instances from different machine families with variations in the number of cores and number of instances being used. Contingency analysis of approximately 30,000 different contingencies was run on one processor, on multiple processors (dependent on the number of cores available for each machine type), and as multiple lists splitting the contingencies among instances. The best option was proposed as running a single instance of an m4.10xlarge machine, which analysed 29,084 contingencies in approximately 91 seconds for a cost of $0.24.

Finally, the linear scalability of running N-2 contingency analysis was examined. The cloud client and server application was used with the optimal settings proposed from the N-1 research. This included the use of Monte-Carlo simulations on a random selection of N-2 contingencies. Extrapolation of the results estimated that 422,924,986 N-2 contingencies could be analysed on 14,098 instances in less than 10 minutes for approximately $4,400.

8.2 Future Work

This thesis has focused on a number of application areas related to cloud computing, with emphasis on data collection, security and analysis. Chapter 5 investigated the use of cloud platforms to collect µPMU data, using event-based triggering to reduce bandwidth requirements and showing how this data can be used to identify and improve network data quality. Chapter 7 demonstrated how commercial cloud computing platforms could be used to perform contingency analysis on a large-scale, real-world network using a commercial power flow engine.

The final research contribution of this thesis is the proposal of an architecture that would allow commercial power system analysis vendors to offer a standards-based, on-demand analysis service to utilities using either public or private cloud platforms. It would allow integration of the real-time data being recorded by µPMUs with network models to automatically perform complex, computationally intensive analysis. This would provide greater visibility of a power system's status and would pro-actively respond to network disturbances.

Integration of real-time µPMU data with detailed network data and access to high performance computing on-demand (in the cloud) will allow utilities to perform continuous analysis of current network conditions. For example: when a µPMU is triggered by a disturbance this will invoke the transmission of its data to the cloud platform, but it can also prompt the cloud platform to request data from other µPMUs and sensors in the surrounding area. This would provide a highly detailed view of the network at the time of the disturbance, and potentially for a period of time before the original trigger occurred.

The triggered data can automatically be combined with a network model stored in the cloud to run a combination of pre-defined and dynamic contingencies, based on the conditions of the network. For example: if the µPMU data was showing voltage fluctuations on a section of the network the contingencies would be based on a combination of generators and branches in that section going offline, as well as running the pre-defined N-1 contingency analysis. For larger networks this could result in millions of different contingencies, however it was shown in Chapter 7 that these could be analysed in a matter of minutes for a relatively economical cost. By configuring a platform to automatically instantiate and run multiple high-performance cloud instances, operators can be pro-actively informed of potential issues with the network without requiring manual analysis.

From a commercial point of view, vendors of power system analysis software may see benefits in transitioning from the traditional model of charging per software license issued to offering ‘Power Flow as a Service’. This would charge utilities for each analysis run. The software vendor could set up and manage a trusted cloud platform to encourage the use from utilities that are wary of using public cloud infrastructure. This cloud will already have undergone the pre-requisite regulatory background and security checks. It will thus be more appealing to utilities as it is being run by an existing, trusted vendor and supplier of power system analysis software. This would represent a paradigm shift for the power industry, but it would also open up new opportunities for both vendors and utilities to exploit the large amounts of computing power now available on demand.

Currently, utilities are unlikely to undertake a full N-2 analysis, as running through millions of different contingencies would take months to complete on a single core computer. A high performance computing workstation would still require weeks to run all the different permutations, rendering it unsuitable for real-time control and operations of changing network conditions. Reducing the number of N-2 contingencies to a selective one million would still require hours of processing, but could be completed in minutes with the use of a cloud platform. This makes the cloud a useful tool for operators that want to identify the most vulnerable areas of a network and determine which outages will have the largest impact, both on a static network model and for the current operational state of the network.

From a vendor's perspective this is a new business model that is offering on-demand analysis on a commercial, public or private trusted cloud platform. If multiple vendors adopted 'Power Flow as a Service', it would be possible for utilities to choose which vendor's cloud to use each time analysis is required. Costs would vary depending on the availability of resources and the processing time and power requested, which in turn would vary with the complexity of analysis required.

To allow a choice in vendors there would have to be a compatibility of interfaces amongst the provided cloud options. Utility systems would need to be able to automatically connect and invoke the analysis in an open, standard format. This requires standardisation of:

- Network models
- Case configuration (including contingencies)
- Analysis parameters
- Analysis results
- Service operations

The IEC CIM provides a standardised data model for network data. This covers balanced and unbalanced networks as well as operational and planning configurations. The IEC 61970-456 profiles cover starting conditions with the Steady State Hypothesis profile, and the State Variables profile covers the resulting solution from a power flow. The IEC 61970-552 standard defines a difference model format for incremental changes that can be used to define contingencies as changes to the base model.

Invoking the analysis could be as simple as “runPowerflow” with an input of a network model and the output of a solution. However a utility may want to perform other operations prior to the execution of a power flow, such as a validation of the data or a connectivity analysis to determine if there are multiple islands in the system. Each operation could therefore have additional parameters that would impact on the result. For example: the CIM contains classes such as *EnergySource*, which in some distribution networks represents the power provided by the transmission system and is thus modelled as a swing generator or *infinite source* for analysis purposes. In other networks *EnergySource* is used to model small-scale generation where the output is fixed or capped. When running the power flow it may be required to instruct it how to interpret these elements otherwise the results could vary significantly.

The CIM does not cover these input parameters or other application-specific analysis parameters. For example: it does not cover power flow specific parameters such as the convergence limits or the type of power flow to be used (i.e. Newton Raphson or Fast Decoupled). The CIM can define reactive power limits for generation or the tap changer settings on a transformer. It does not, however, define standardised parameters allowing a user to tell an analysis engine to ignore reactive limits or to use an alternative algorithm for regulating tap changers. This would require vendors to agree the input and output data for a simulation, as well as a standardisation of the additional parameters that are used to influence the execution of the analysis.

Assuming such agreements could be put in place, this would allow a utility to use ‘Power Flow as a Service’ from a number of different vendors, in the same way that they could use multiple cloud service providers or internet service providers. Vendors would compete on the service and price they offer rather than because a utility is tied to their platform and format.

References

[1] J. M. Melillo, T. Richmond and G. Yohe, "Climate change impacts in the United States," *Third National Climate Assessment,* 2014.

[2] C. M. Shand *et al*, "Improving actionable observability of large distribution networks for transmission operators to support improved system control, fault detection and mitigation," *CIRED - Open Access Proceedings Journal,* vol. 2017, *(1),* pp. 1215-1218, 2017.

[3] C. Shand *et al*, "Exploiting massive PMU data analysis for LV distribution network model validation," in *2015 50th International Universities Power Engineering Conference (UPEC),* 2015, pp. 1-4.

[4] Anonymous "IEEE Guide for Phasor Data Concentrator Requirements for Power System Protection, Control, and Monitoring," *IEEE Std C37. 244-2013,* pp. 1-65, 2013.

[5] EPRI and US DOE, "North American SynchroPhasor Initiative ," Available: https://www.naspi.org/ [Accessed September 2017].

[6] E. M. Stewart *et al*, "Addressing the challenges for integrating micro-synchrophasor data with operational system applications," in *IEEE Power & Energy Society General Meeting,* 2014, pp. 1-5.

[7] Power Sensors Ltd., "Synchrophasors for Distribution, Microgrids: PQube 3 MicroPMU," 2015.

[8] E.M. Stewart, S. Kiliccote, and C. McParland, "Software-Based Challenges of Developing the Future Distribution Grid," *LBNL Report Number 6708E,* 2014.

[9] AWS, "Amazon Web Services," Available: https://aws.amazon.com/ [Accessed August 2014].

[10] AWS, "Amazon Web Services Glossary," Available: http://docs.aws.amazon.com/general/latest/gr/aws-general.pdf#glos-chap [Accessed November 2017].

[11] D. Wallom *et al*, "myTrustedCloud: Trusted cloud infrastructure for security-critical computation and data managment," in *IEEE Third International Conference on Cloud Computing Technology and Science,* 2011, pp. 247-254.

[12] M. J. Sule *et al*, "Fuzzy logic approach to modelling trust in cloud computing," *IET Cyber-Physical Systems: Theory & Applications,* vol. 2, *(2),* pp. 84-89, 2017.

[13] A. R. Metke and R. L. Ekl, "Security Technology for Smart Grid Networks," *IEEE Transactions on Smart Grid,* vol. 1, *(1),* pp. 99-107, 2010.

[14] C. Shand, A. McMorran and G. Taylor, "Integration and adoption of open data standards for online and offline power system analysis," in *49th International Universities Power Engineering Conference (UPEC),* 2014, pp. 1-6.

[15] J. D. Glover, M. S. Sarma, T. Overbye, *Power System Analysis and Design.* (6th Edition, SI ed.) Cengage Learning, 2017.

[16] Lynn Powell, *Power System Load Flow Analysis (Professional Engineering).* McGraw-Hill Education, 2005.

[17] B. Stott and O. Alsac, "Fast Decoupled Load Flow," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-93, *(3),* pp. 859-869, 1974.

[18] M. Albu, G. T. Heydt and S. C. Cosmescu, "Versatile platforms for wide area synchronous measurements in power distribution systems," in *North American Power Symposium 2010,* 2010, pp. 1-7.

[19] E. M. Stewart *et al*, "Addressing the challenges for integrating micro-synchrophasor data with operational system applications," in *2014 IEEE PES General Meeting | Conference & Exposition,* 2014, pp. 1-5.

[20] X. Chen, K. J. Tseng and G. Amaratunga, "State estimation for distribution systems using micro-synchrophasors," in *2015 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC),* 2015, pp. 1-5.

[21] B. Pinte, M. Quinlan and K. Reinhard, "Low voltage micro-phasor measurement unit (µPMU)," in *2015 IEEE Power and Energy Conference at Illinois (PECI),* 2015, pp. 1-4.

[22] Anonymous "IEEE Standard for Synchrophasor Measurements for Power Systems," *IEEE Std C37. 118. 1-2011 (Revision of IEEE Std C37. 118-2005),* pp. 1-61, 2011.

[23] Anonymous "IEEE Standard for Synchrophasor Data Transfer for Power Systems," *IEEE Std C37. 118. 2-2011 (Revision of IEEE Std C37. 118-2005),* pp. 1-53, 2011.

[24] K. D. Jones, J. S. Thorp and R. M. Gardner, "Three-phase linear state estimation using phasor measurements," in *2013 IEEE Power & Energy Society General Meeting,* 2013, pp. 1-5.

[25] A. Ghassemian and B. Fardanesh, "Phasor assisted state estimation for NYS transmission system --implementation & testing," in *2009 IEEE/PES Power Systems Conference and Exposition,* 2009, pp. 1-8.

[26] C. Muscas *et al*, "Uncertainty of Voltage Profile in PMU-Based Distribution System State Estimation," *IEEE Transactions on Instrumentation and Measurement,* vol. 65, *(5),* pp. 988-998, 2016.

[27] M. Baran and T. E. McDermott, "Distribution system state estimation using AMI data," in *2009 IEEE/PES Power Systems Conference and Exposition,* 2009, pp. 1-3.

[28] S. Sarri *et al*, "A hardware-in-the-loop test platform for the performance assessment of a PMU-based real-time state estimator for active distribution networks," in *2015 IEEE Eindhoven PowerTech,* 2015, pp. 1-6.

[29] D. Wang *et al*, "PMU-based angle constraint active management on 33kV distribution network," in *22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013),* 2013, pp. 1-4.

[30] J. Tlusty *et al*, "The monitoring of power system events on transmission and distribution level by the use of phasor measurements units (PMU)," in *CIRED 2009 - 20th International Conference and Exhibition on Electricity Distribution - Part 1,* 2009, pp. 1-4.

[31] V. Ramesh, U. Khanz and M. D. Ilicx, "Data aggregation strategies for aligning PMU and AMI measurements in electric power distribution networks," in *2011 North American Power Symposium,* 2011, pp. 1-7.

[32] D. Atanackovic and V. Dabic, "Deployment of real-time state estimator and load flow in BC hydro DMS - challenges and opportunities," in *2013 IEEE Power & Energy Society General Meeting,* 2013, pp. 1-5.

[33] M. Pignati *et al*, "Real-time state estimation of the EPFL-campus medium-voltage grid by using PMUs," in *2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT),* 2015, pp. 1-5.

[34] A. Rajeev, Angel T S and F. Z. Khan, "Fault location in distribution feeders with optimally placed PMU's," in *2015 International Conference on Technological Advancements in Power and Energy (TAP Energy),* 2015, pp. 438-442.

[35] W. Yuill *et al*, "Optimal PMU placement: A comprehensive literature review," in *2011 IEEE Power and Energy Society General Meeting,* 2011, pp. 1-8.

[36] H. A. Abdelsalam *et al*, "Impact of distribution system reconfiguration on optimal placement of phasor measurement units," in *2014 Clemson University Power Systems Conference,* 2014, pp. 1-6.

[37] A. A. Fish, S. Chowdhury and S. P. Chowdhury, "Optimal PMU placement in a power network for full system observability," in *2011 IEEE Power and Energy Society General Meeting,* 2011, pp. 1-8.

[38] H. A. Abdelsalam, A. Y. Abdelaziz and V. Mukherjee, "Optimal PMU placement in a distribution network considering network reconfiguration," in *2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014],* 2014, pp. 191-196.

[39] N. H. A. Rahman, A. F. Zobaa and M. Theodoridis, "Improved BPSO for optimal PMU placement," in *2015 50th International Universities Power Engineering Conference (UPEC),* 2015, pp. 1-4.

[40] Yue Yang and S. Roy, "PMU placement for optimal three-phase state estimation performance," in *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm),* 2013, pp. 342-347.

[41] M. J. Sule *et al*, "Deploying trusted cloud computing for data intensive power system applications," in 2015, pp. 1-5.

[42] O. VukoviÄ‡, G. DÃ¡n and R. B. Bobba, "Confidentiality-preserving obfuscation for cloud-based power system contingency analysis," in 2013, pp. 432-437.

[43] Z. Cao *et al*, "Optimal Cloud Computing Resource Allocation for Demand Side Management in Smart Grid," *IEEE Transactions on Smart Grid,* vol. 8, *(4),* pp. 1943-1955, 2017.

[44] W. Sheng *et al*, "On parallelizing analysis of power systems in cloud environment," in 2016, pp. 576-580.

[45] M. Randles, D. Lamb and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," in 2010, pp. 551-556.

[46] K. Maheshwari *et al*, "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud," in 2013, pp. 1-6.

[47] Z. H. Huang *et al*, "Guest Editorial High Performance Computing (HPC) Applications for a More Resilient and Efficient Power Grid," *IEEE Transactions on Smart Grid,* vol. 8, *(3),* pp. 1363-1365, 2017.

[48] Anonymous "Power systems management and associated information exchange – Part 1: Reference architecture," *International Standard IEC 62357-1,* 2012.

[49] A. W. McMorran *et al*, "A common information model (CIM) toolkit framework implemented in Java," *IEEE Transactions on Power Systems,* vol. 21, *(1),* pp. 194-201, 2006.

[50] Anonymous "Energy management system application program interface (EMS - API) – Part 452: CIM static transmission network model profiles," *International Standard IEC 61970-452,* 2013.

[51] Anonymous "Application integration at electric utilities – System interfaces for distribution management – Part 13: CIM RDF Model exchange format for distribution," *International Standard IEC 61968-13,* 2008.

[52] A. de Vos, S. E. Widergren and J. Zhu, "XML for CIM model exchange," in *22nd IEEE Power Engineering Society International Conference on Power Industry Computer Applications,* 2001, pp. 31-37.

[53] Anonymous "Energy management system application program interface (EMS - API) – Part 552: CIM XML Model exchange format," *International Standard IEC 61970-552,* 2013.

[54] J. Mattioni *et al*, "Guidelines for Control Center Application Program Interfaces," *Electric Power Research Institute,* 1996.

[55] ENTSO-E, "Interoperability Test 'CIM for System Development and Operations' 2013," 2013.

[56] Working Group, "Common Format For Exchange of Solved Load Flow Data," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-92, *(6),* pp. 1916-1925, 1973.

[57] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-86, *(11),* pp. 1449-1460, 1967.

[58] ENTSO-E, "Mission and Vision," Available: https://www.entsoe.eu/about-entso-e/inside-entso-e/mission-and-vision/ [Accessed May 2014].

[59] Anonymous "Telecontrol equipment and systems – Part 6 - 503: Telecontrol protocols compatible with ISO standards and ITU-T recommendations – TASE.2 Services and protocol," *International Standard IEC 60870-6-503,* 2002.

[60] B. Galloway and G. P. Hancke, "Introduction to Industrial Control Networks," *IEEE Communications Surveys & Tutorials,* vol. 15, *(2),* pp. 860-880, 2013.

[61] A. von Meier *et al*, "Micro-synchrophasors for distribution systems," in *Innovative Smart Grid Technologies Conference,* 2014, pp. 1-5.

[62] Anonymous "Communication networks and systems for power utility automation - Part 5: Communication requirements for functions and device models," *International Standard IEC 61850-5,* 2013.

[63] Anonymous "IEEE Standard for Synchrophasor Measurements for Power Systems," *IEEE Std C37. 118. 1-2011 (Revision of IEEE Std C37. 118-2005),* pp. 1-61, 2011.

[64] Anonymous "Application integration at electric utilities – System interfaces for distribution management – Part 9: Interfaces for meter reading and control," *International Standard IEC 61968-9,* 2013.

[65] A. W. McMorran *et al*, "Translating CIM XML power system data to a proprietary format for system simulation," *IEEE Transactions on Power Systems,* vol. 19, *(1),* pp. 229-235, 2004.

[66] P. Wall *et al*, "VISOR Project: Opportunities for Enhanced Real Time Monitoring and Visualisation of System Dynamics in GB," .

[67] SP Energy Networks, "DSO Vision," 2016.

[68] G. L. Kusic and D. L. Garrison, "Measurement of transmission line parameters from SCADA data," in *IEEE PES Power Systems Conference and Exposition,* 2004, pp. 440-445 vol.1.

[69] W. F. Tinney and C. E. Hart, "Power Flow Solution by Newton's Method," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-86, *(11),* pp. 1449-1460, 1967.

[70] B. Stott and O. Alsac, "Fast Decoupled Load Flow," *IEEE Transactions on Power Apparatus and Systems,* vol. PAS-93, *(3),* pp. 859-869, 1974.

[71] E. M. Stewart *et al*, "Addressing the challenges for integrating micro-synchrophasor data with operational system applications," in *IEEE Power & Energy Society General Meeting,* 2014, pp. 1-5.

[72] Scottish Hydro Electric Transmisson Plc, "Information Boards," 2014.

[73] Scottish and Southern Energy Power Distribution, "Key Facts about our Network," 2015.

[74] J. Britton and P. Brown, "Using the Common Information Model for Network Analysis Data Management," *Electric Power Research Institute,* 2014.

[75] A. J. Allen *et al*, "Validation of distribution level measurements for power system monitoring and low frequency oscillation analysis," in *IEEE Power Electronics and Machines in Wind Applications,* 2012, pp. 1-5.

[76] V. C. Gungor *et al*, "Smart Grid Technologies: Communication Technologies and Standards," *IEEE Transactions on Industrial Informatics,* vol. 7, *(4),* pp. 529-539, 2011.

[77] R. Venkateswaran, "Virtual private networks," *IEEE Potentials,* vol. 20, *(1),* pp. 11-15, 2001.

[78] H. J. Zhou, C. X. Guo and J. Qin, "Efficient application of GPRS and CDMA networks in SCADA system," in *IEEE PES General Meeting,* 2010, pp. 1-6.

[79] IETF Network Working Group, "The Transport Layer Security (TLS) Protocol," 2008. Available: https://tools.ietf.org/pdf/rfc5246.pdf [Accessed November 2017].

[80] IETF Network Working Group, "HTTP Over TLS," 2000. Available: https://tools.ietf.org/pdf/rfc2818.pdf [Accessed November 2017].

[81] A. R. Metke and R. L. Ekl, "Security Technology for Smart Grid Networks," *IEEE Transactions on Smart Grid,* vol. 1, *(1),* pp. 99-107, 2010.

[82] M.P. Andersen and D.E. Culler, "BTrDB: Optimizing storage system design for timeseries processing," in *14th USENIX Conference on File and Storage Technologies (FAST '16),* Santa Clara, CA, USA, 2016, pp. 1-15.

[83] R.M. Lee, M.J. Assante and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid," 2016.

[84] E. M. Stewart *et al*, "Accuracy and validation of measured and modeled data for distributed PV interconnection and control," in *IEEE Power & Energy Society General Meeting,* 2015, pp. 1-5.

[85] R. G. Cespedes, "New method for the analysis of distribution networks," *IEEE Transactions on Power Delivery,* vol. 5, *(1),* pp. 391-396, 1990.

[86] Open Electrical, "Power System Analysis Software," Available: https://wiki.openelectrical.org/ [Accessed November 2017].

[87] Siemens, "Power Transmission System Planning Software," Available: https://w3.siemens.com/smartgrid/global/en/products-systems-solutions/software-solutions/planning-data-management-software/planning-simulation/pages/pss-e.aspx; [Accessed November 2017].

[88] GE Energy Consulting, "PSLF," Available: http://www.geenergyconsulting.com/practice-area/software-products/pslf; [Accessed November 2017].

[89] IPSA POWER, "Power System Analysis Software – Ipsa 2," Available: https://www.ipsa-power.com/?page_id=766; [Accessed November 2017].

[90] DIgSILENT, "PowerFactory Application," Available: https://www.digsilent.de/en/powerfactory.html; [Accessed November 2017].

[91] Nexant, "Grid360," Available: http://www.nexant.com/software/grid360/grid360-transmission-analytics; [Accessed November 2017].

[92] GridLAB-D, "Welcome," Available: http://www.gridlabd.org; [Accessed November 2017].

[93] EPRI Smart Grid Resource Centre, "Simulation Tool – OpenDSS," Available: http://smartgrid.epri.com/SimulationTool.aspx; [Accessed November 2017].

[94] Smart Grid Simulation Library, "SmartGridToolbox," Available: http://nicta.github.io/SmartGridToolbox/; [Accessed November 2017].

[95] Federico Milano's Webpage, "PSAT," Available: http://faraday1.ucd.ie/psat.html; [Accessed November 2017].

[96] Power Systems Engineering Research Centre, "MATPOWER," Available: http://www.pserc.cornell.edu/matpower/; [Accessed November 2017].

[97] AWS, "Amazon EC2 Instance Types," Available: https://aws.amazon.com/ec2/instance-types/; [Accessed November 2017].

[98] AWS, "Initializing Amazon EBS Volumes," [Accessed November 2017].

[99] AWS, "Amazon EC2 Pricing," Available: https://aws.amazon.com/ec2/pricing/on-demand/; [Accessed November 2017].

[100] AWS, "Amazon EBS Pricing," Available: https://aws.amazon.com/ebs/pricing/; [Accessed November 2017].

[101] D. Sklavos, "DDR3 vs. DDR4: Raw bandwidth by the numbers," 2015. Available: https://www.techspot.com/news/62129-ddr3-vs-ddr4-raw-bandwidth-numbers.html [Accessed December 2107].

[102] AWS, "Amazon EC2 Instance Configuration," Available: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-ec2-config.html [Accessed November 2017].

[103] AWS, "EC2 Management Console: Step 2 - Choose an Instance Type," Available: https://eu-west-1.console.aws.amazon.com/ec2/v2/home?region=eu-west-1#LaunchInstanceWizard:ami=ami-710ccf08 [Accessed November 2017].

[104] AWS, "Amazon EC2 Pricing," Available: https://aws.amazon.com/ec2/pricing/on-demand/;; [Accessed November 2017].

[105] G.M. Amdahl, "Validity of the single processor approach to achieving large ScaleComputing capabilities," in *AFIPS Spring Joint Computer Conference,* 1967, pp. 1-4.

[106] J. L. Gustafson, "Reevaluating Amdahl's Law," *Communications of the ACM,* vol. 31, *(5),* 1988.