Assessment of a Computational Grid for the Replay of Digitally-Recorded Holograms

J.J. Nebrensky and P.R. Hobson

School of Engineering and Design
Brunel University, UK

Acknowledgements:
EGEE, GridPP and LT2 institutes for computing resources

BRUNEL
UNIVERSITY
WEST LONDON

Topics

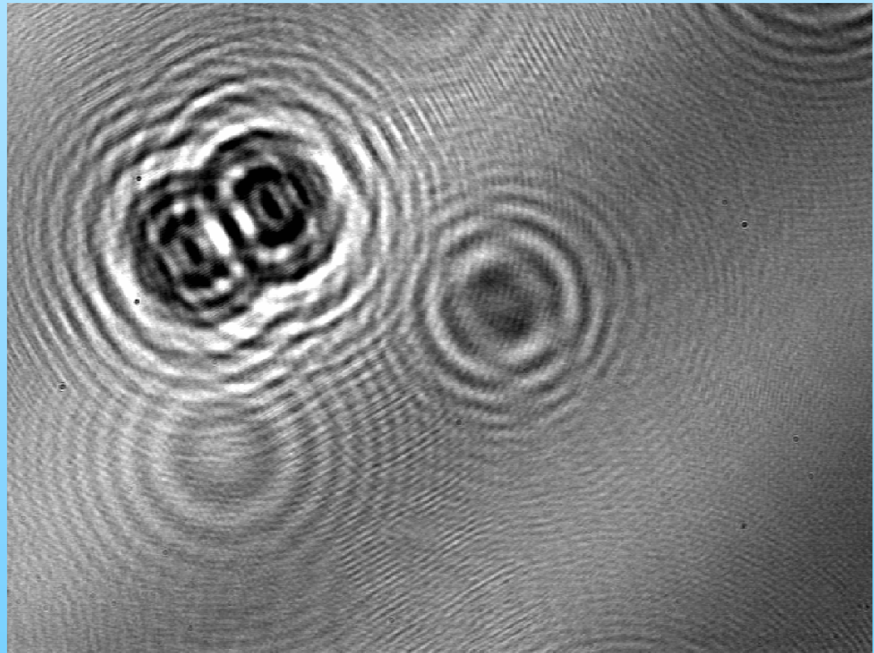
- **Digital holography**
  - In-line (Gabor) hologram captured directly on CCD
  - Reconstruct sample volume numerically inside computer
  - You just heard about it!

- **What is Grid computing?**

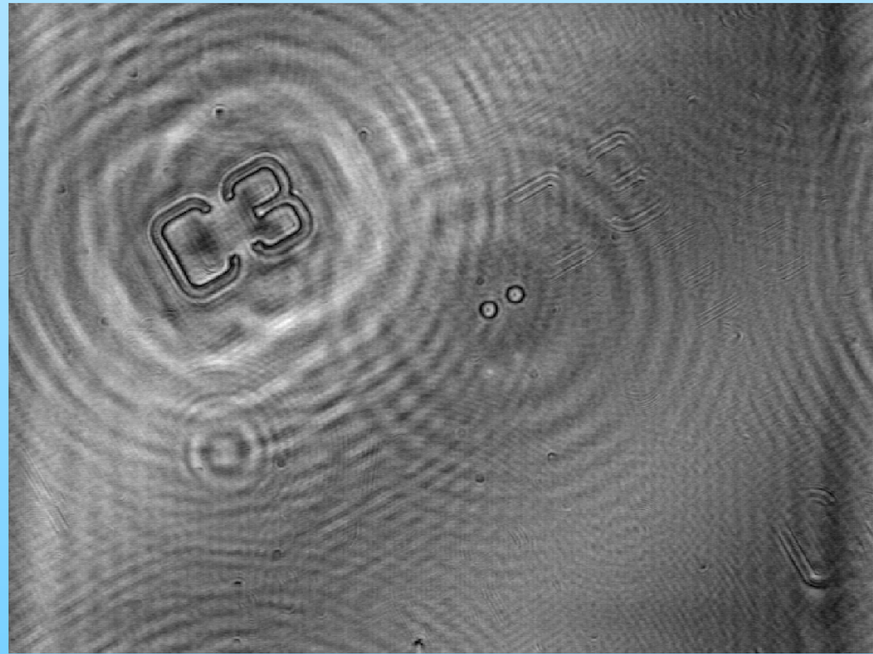
- **How can it be used with digital holography?**

Digital Holography

Sample results obtained from an array of pairs of opaque discs:



An in-line hologram of a test target, captured from a CCIR videocamera



The in-focus objects, regenerated within the computer

(UG project work by Marc Fournier-Carrié)

Digital Holography

We wish to identify the properties of and relationships between particles in water. We want the largest possible sample volume so as to get the best statistics.

To find the objects for further analysis, we must reconstruct a series of slices and then search through each in turn (the brute force approach, as we already have an existing object-finding code from an earlier project).

Numerical reconstruction is computationally heavy and intermediate data volumes are huge – too much for single PC.

Digital Holography

Our test volume is a tank of water 80 mm across, with cenospheres stirred into it (mostly 100-300 μm dia.). We used a collimated beam from a 1 mW red HeNe.

We have our own FFTW-based reconstruction software “HoloReco” for single image planes from in-line holograms. This has been tested with images from an Atmel Camelia camera (8 Mpixel, 12-bit depth) †.

The same source code compiles and runs both on Visual C++ 6 on Windows 2000, and on GCC 3.2 on Linux (RedHat 7.3 and Scientific Linux 3).
[Plan to release as open-source]

† based on UG project work by Paul Fryer

How heavy?

Timings based on original reconstruction software:

f: P2 400 MHz 384 Mb (RH6.2)

SAVVAS1.PGM	768x576	8bit	DFT	with no padding	(0.44Mpixels)	914 s
SAVVAS1.PGM	768x576	8bit	DFT	pad to 1024x1024	(1Mpixels)	3305 s
SAVVAS1.PGM	768x576	8bit	FFT	pad to 1024x1024	(1Mpixels)	17 s
8MC.PGM	1024x512	8bit	FFT	with no padding	(0.5Mpixels)	8 s
8MC.PGM	1024x512	8bit	DFT	with no padding	(0.5Mpixels)	1240 s
8MWIRE.PGM	2300x3500	8bit	FFT	pad to 4096x4096	(16Mpixels)	671 s
8MWIRE.PGM	2300x3500	8bit	DFT	pad to 4096x4096	(16Mpixels)	216504 s
8MWIRE.PGM	2300x3500	8bit	DFT	with no padding	(8Mpixels)	73272 s

Image enhancement & species identification:

P2 400MHz 128Mb (NT 4)

42 mins HS20120A ci6a 768x566 1-459 no objects

6'04 HS20120A co2a 768x574 1-61 no objects (<10 seconds per frame)

How heavy?

For each CCIR frame, object tracking and extraction is quicker than the reconstruction.
(We don't yet know how this scales with frame size)

Timings based on original reconstruction software:

```
g: Athlon XP2600+ (1900MHz) 1024Mb (RH7.3)
SAVVAS1.PGM   768x576 8bit FFT pad to 1024x1024 (   1Mpixels)      3 s
8MC.PGM      1024x512 8bit FFT with no padding  ( 0.5Mpixels)      2 s
8MWIRE.PGM  2300x3500 8bit FFT pad to 4096x4096 (  16Mpixels)    101 s
```

(but object tracking and extraction is quicker, too!)

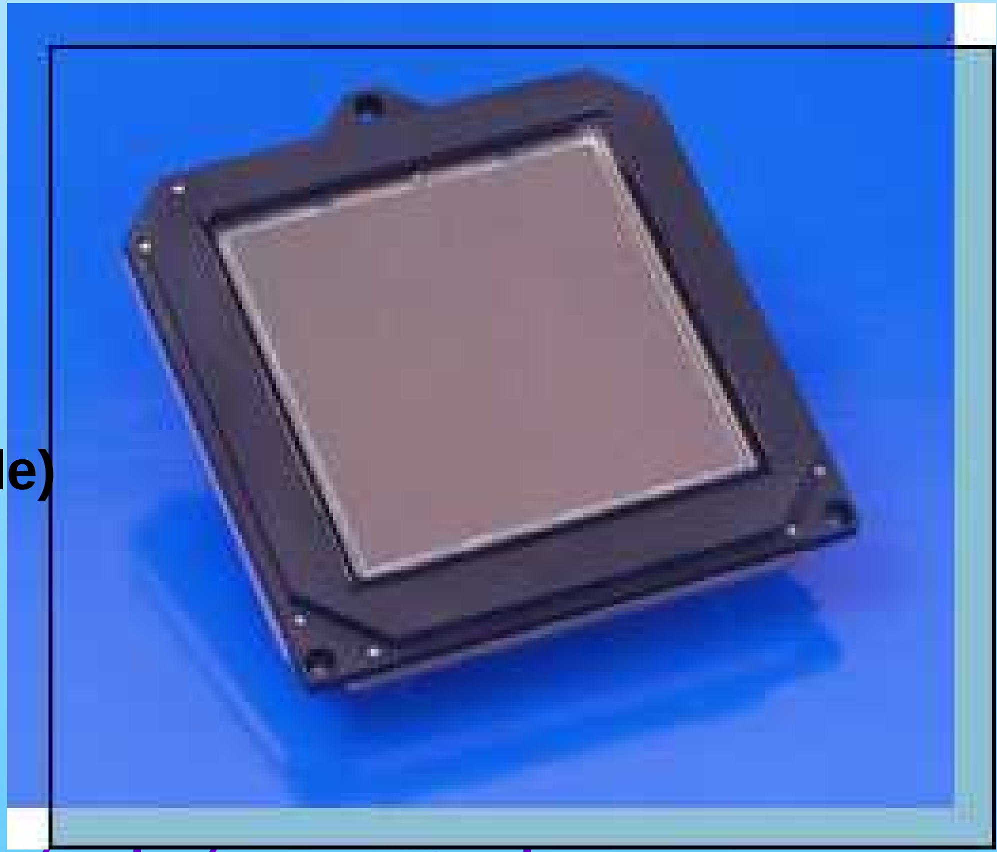
Now about one minute for each plane – but *many* planes (100's) in a single hologram!

How heavy!

The end-results are statistics driven – smarter algorithms would drive a move to larger sample volumes and increased framerate.

Fairchild Imaging CCD595:

- **9216 x 9216 full frame CCD**
- **8.75 µm x 8.75 µm pixels**
- **80 mm x 80 mm image area**
- **100% fill factor**
- **€50-100 000 (maybe no longer available)**



85 Mpixels!

http://www.fairchildimaging.com/main/area_595.htm

What is Grid computing?

More than just "distributed computing":

- **General purpose**
- **"Virtual Organisation" VO – allows its resources to be shared among its members**
- **Wide-Area deployment – heterogeneous resources spread over globe**

- **Foster and Kesselman – the Globus Project**
  - Anatomy of the Grid `http://www.globus.org/alliance/publications/papers/anatomy.pdf`
  - Physiology of the Grid `http://www.globus.org/alliance/publications/papers/ogsa.pdf`
  - I. Foster and C. Kesselman: *"The Grid: Blueprint for a New Computing Infrastructure"*

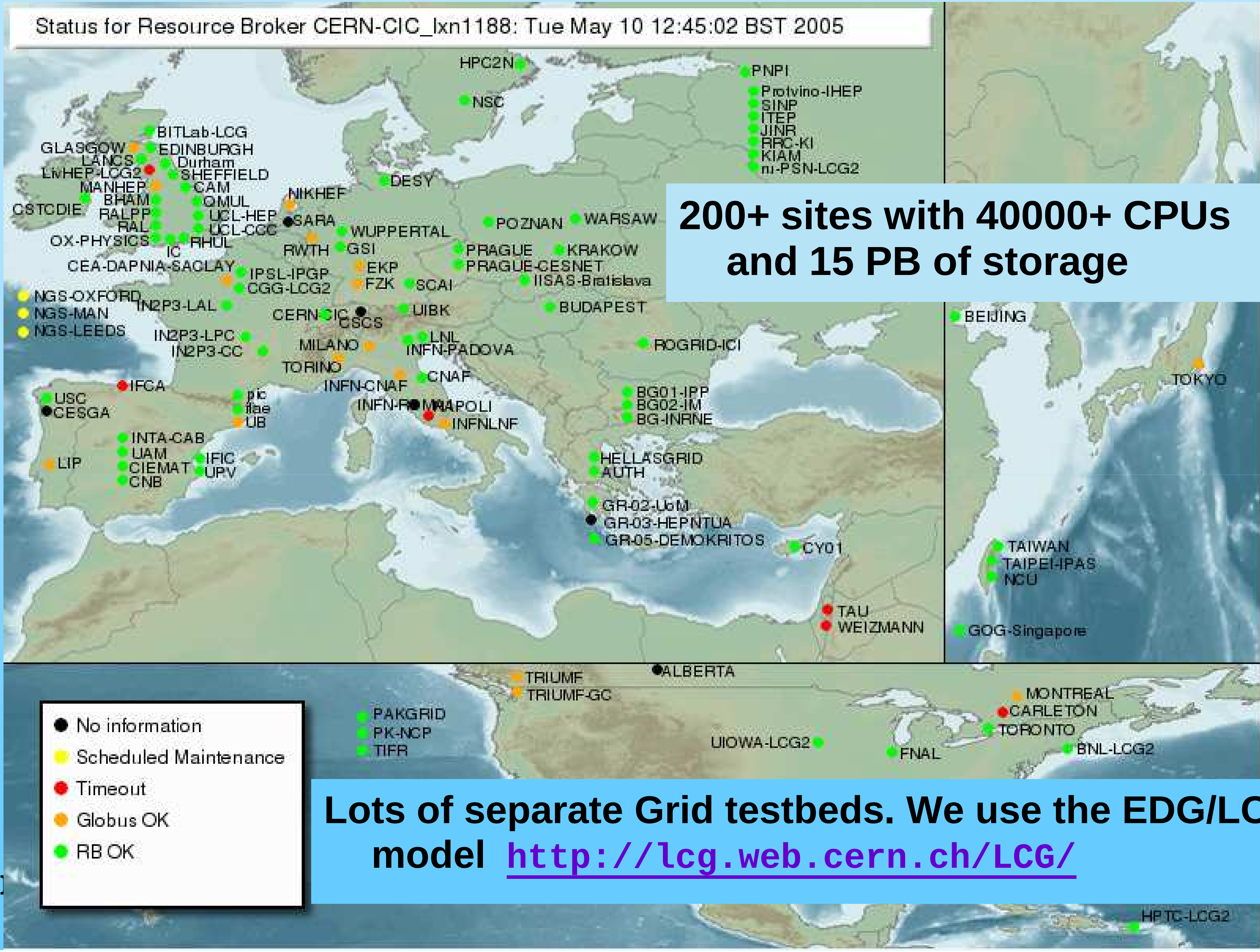
What is Grid computing?

- The Web provides seamless access to data
- Underlying protocols don’t care about who you are

- The Grid provides seamless access to computing
- Users identified via X.509 certificates:
  must be in a suitable VO, then single sign-on

The key is not that the Grid makes it easy to “scavenge” idle resources, but rather that it makes cooperation to avoid wasting them in the first place much easier.

Constantly changing acronyms - terminology here is a personal mish-mash L



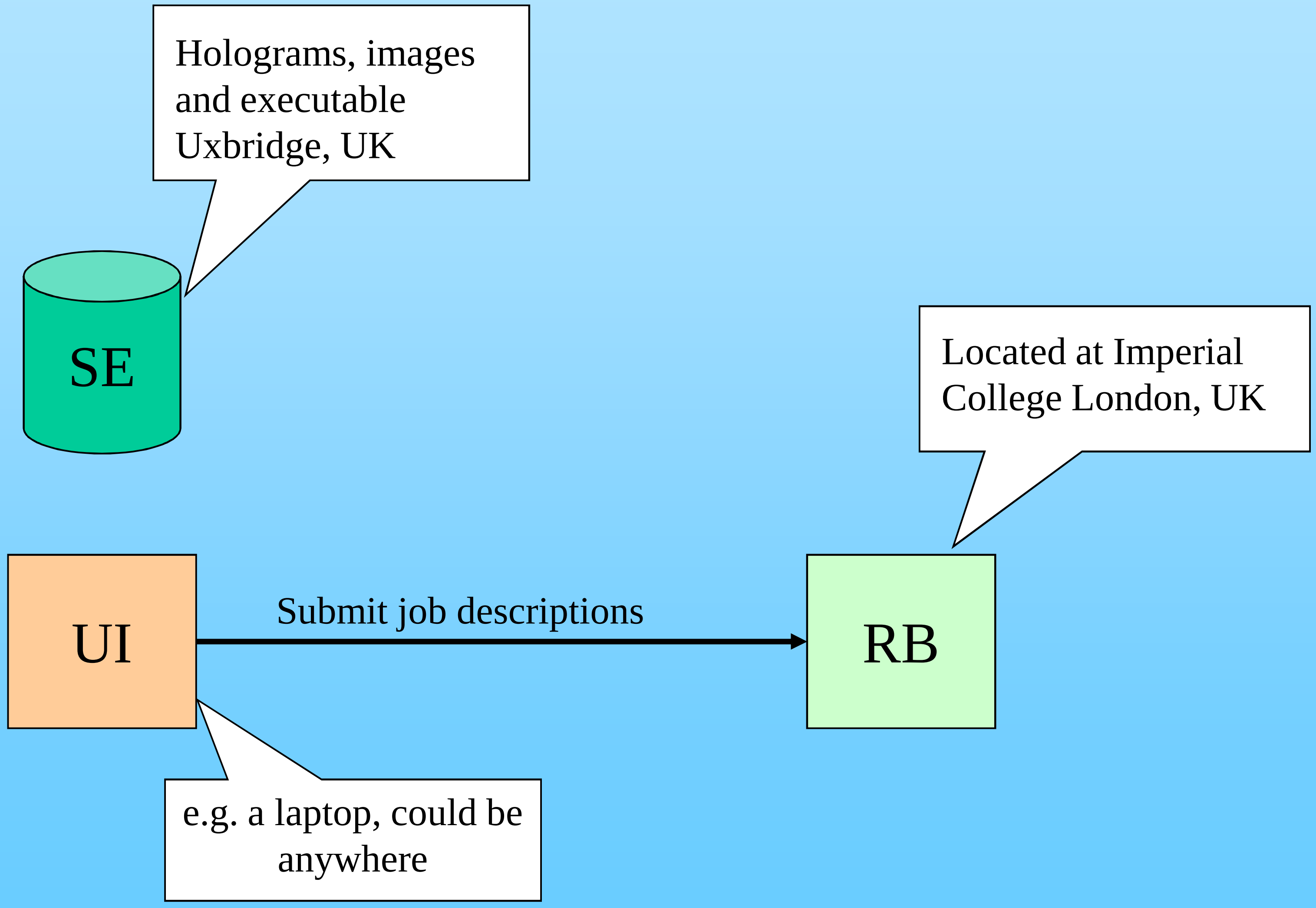
200+ sites with 40000+ CPUs and 15 PB of storage

Lots of separate Grid testbeds. We use the EDG/LCG model http://lcg.web.cern.ch/LCG/

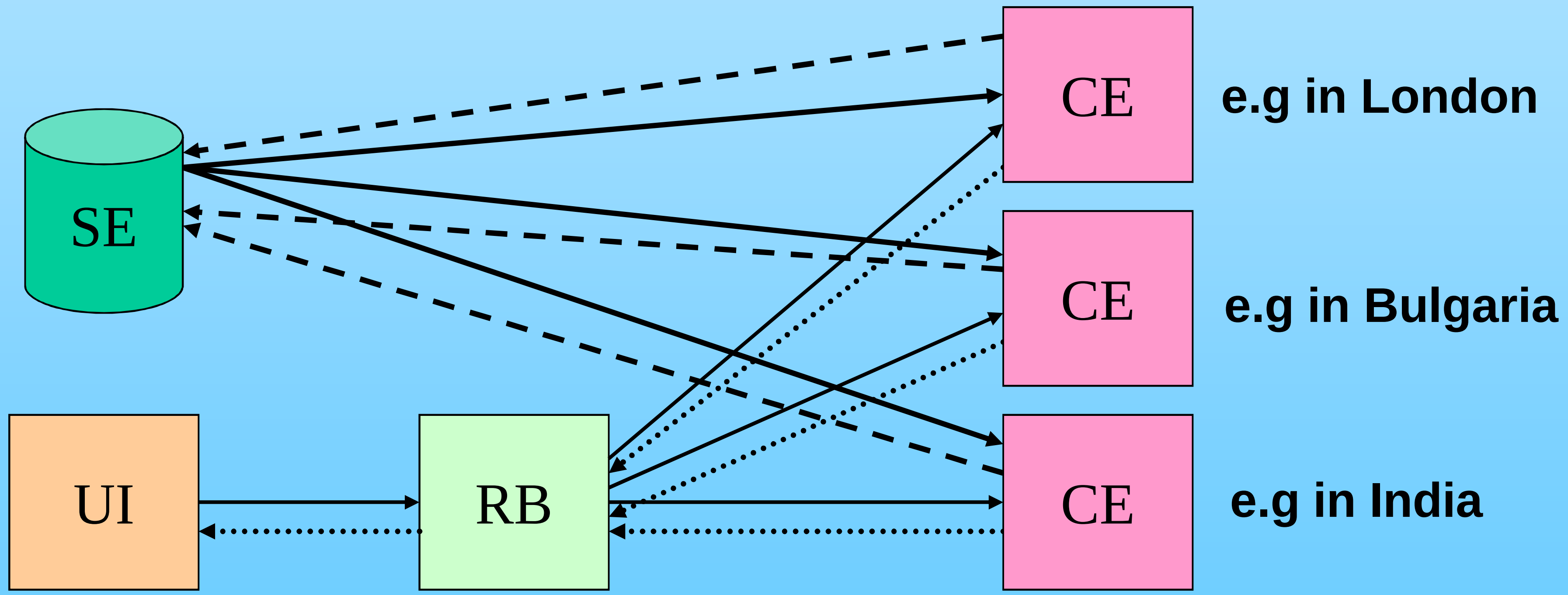
How does a Grid job happen?

- **At a “UI”, specify executable, data files and other requirements using JDL and submit**
- **The UI client passes this to a Resource Broker (RB) which identifies the best place to run the job amongst the cloud of resources available:**
  - a Computing Element provides CPUs
  - a Storage Element provides storage space
- **A CE consists of a Gatekeeper (GK) which receives the job, and a set of Worker Nodes (WN) that do the actual work – similar to traditional batch farm**

Some of the players



Reconstruction on the Grid



Grid Computing in a Nutshell

BBC Cult (bbc.co.uk)

The venerable "Electrical Power" analogy

How do we use the Grid for Digital Holography?

Reconstruction of any slice is independent of all the others (“embarrassingly parallel”), so use Grid to reconstruct many planes at the same time:

- store digital hologram (and binary) at SE
- submit control file for each plane (`edg-job-submit`) and monitor with LCG GUI monitor (`edg-wl-ui-gui-jobmonitor`)
- recover image planes to UI via local SE
- use existing tracking and identification code

Need to keep track of reconstructed planes (which Grid job produced the next image) – an extra layer of software.

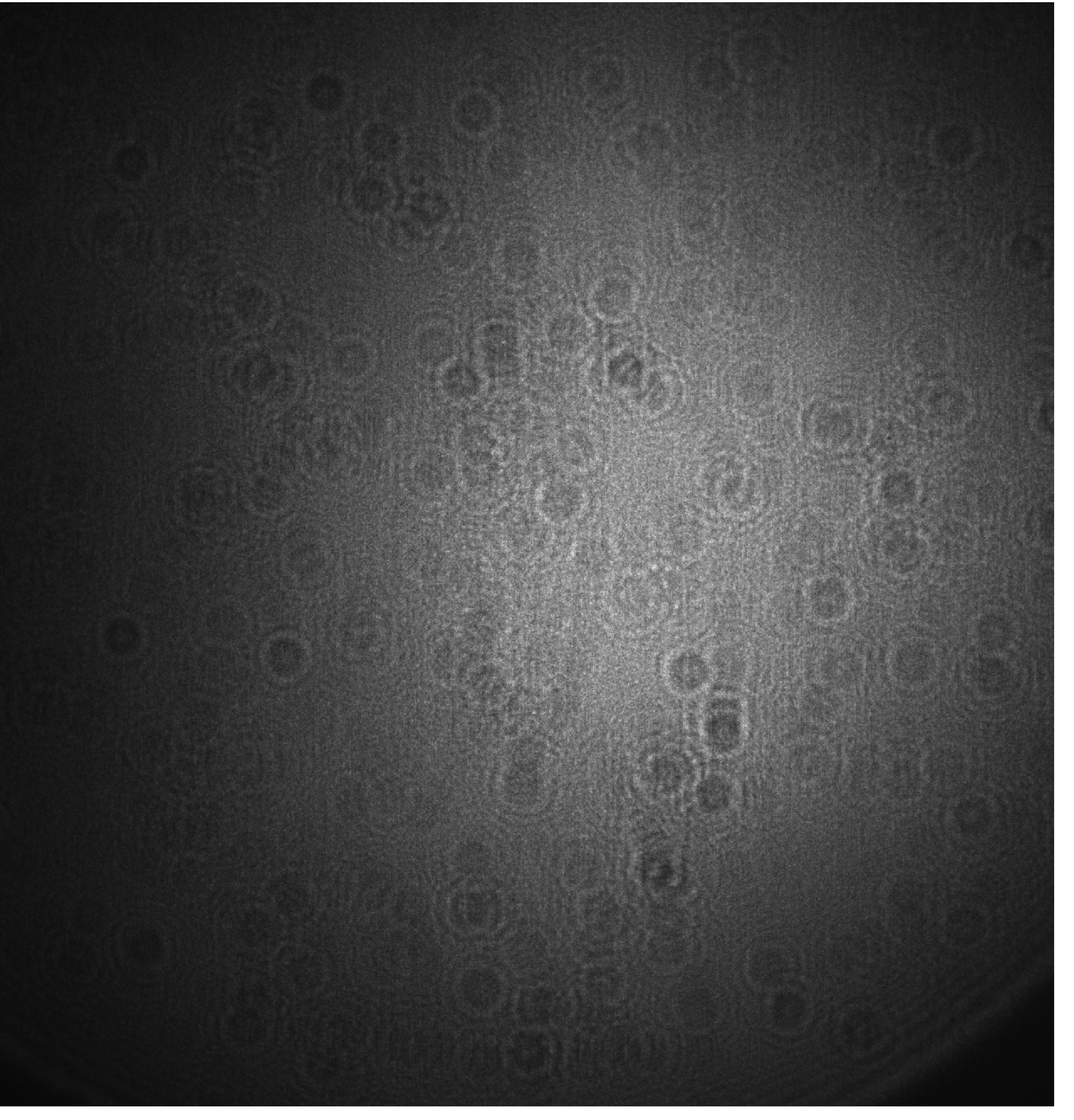
Digital Holography on the Grid

We uploaded the hologram file to the SE associated with the BITLab facility at Brunel University.

HoloReco was then run on resources around the Grid, with the reconstructed images being stored on the Brunel SE. The work was done within the LTWO VO, giving access to resources within the London Tier 2 of the UK GridPP project, that forms part of the EGEE Grid. LT2 is a collaboration which currently has a total of ~3000 CPUs across 7 institutes in London.

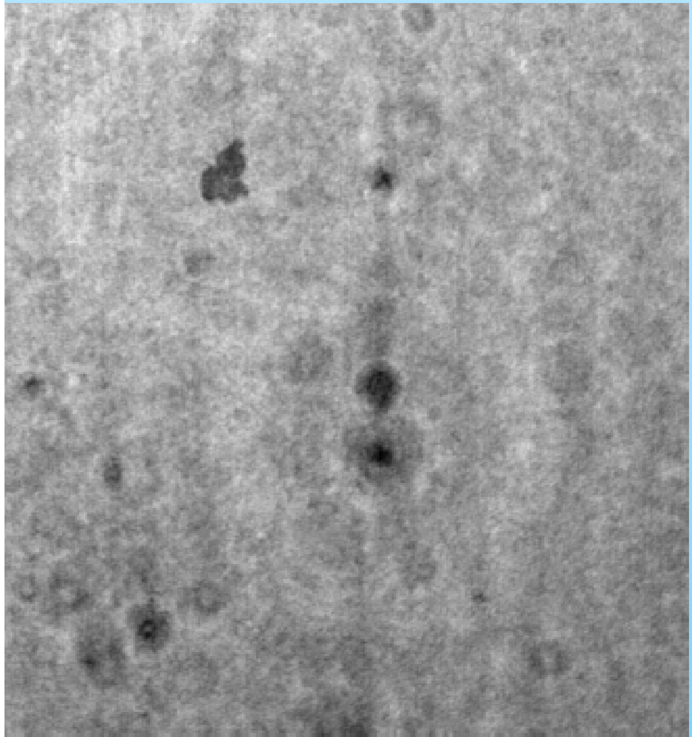
Our sample holograms are 2300 by 3500 pixels with 12-bit depth. We use PGM format image files, which can be up to 40MB in size. The hologram and images are gzipped before being uploaded to the SE (also allows integrity checking).

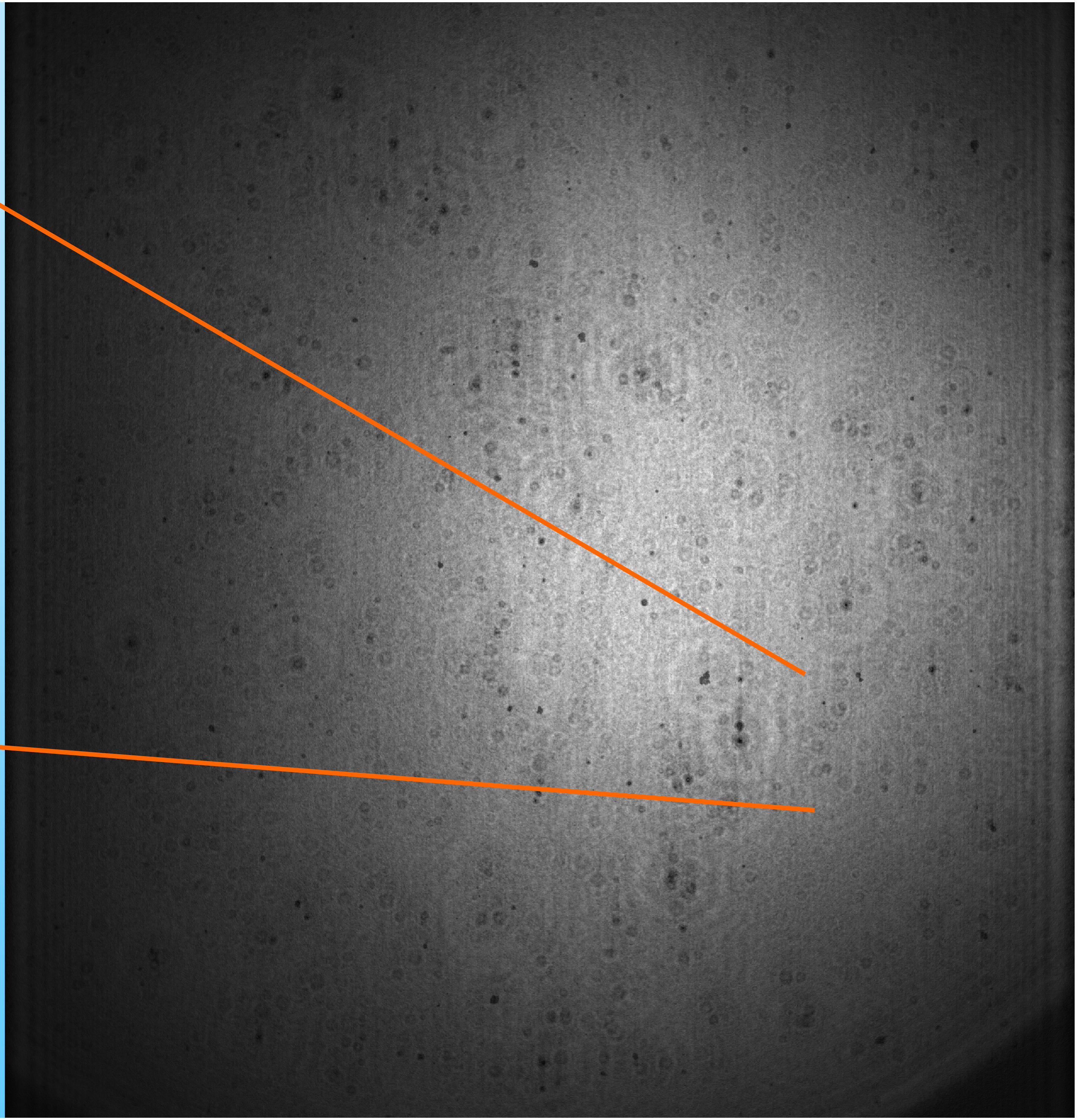
From this…



IoP Hydrological Optics

To this...





Method

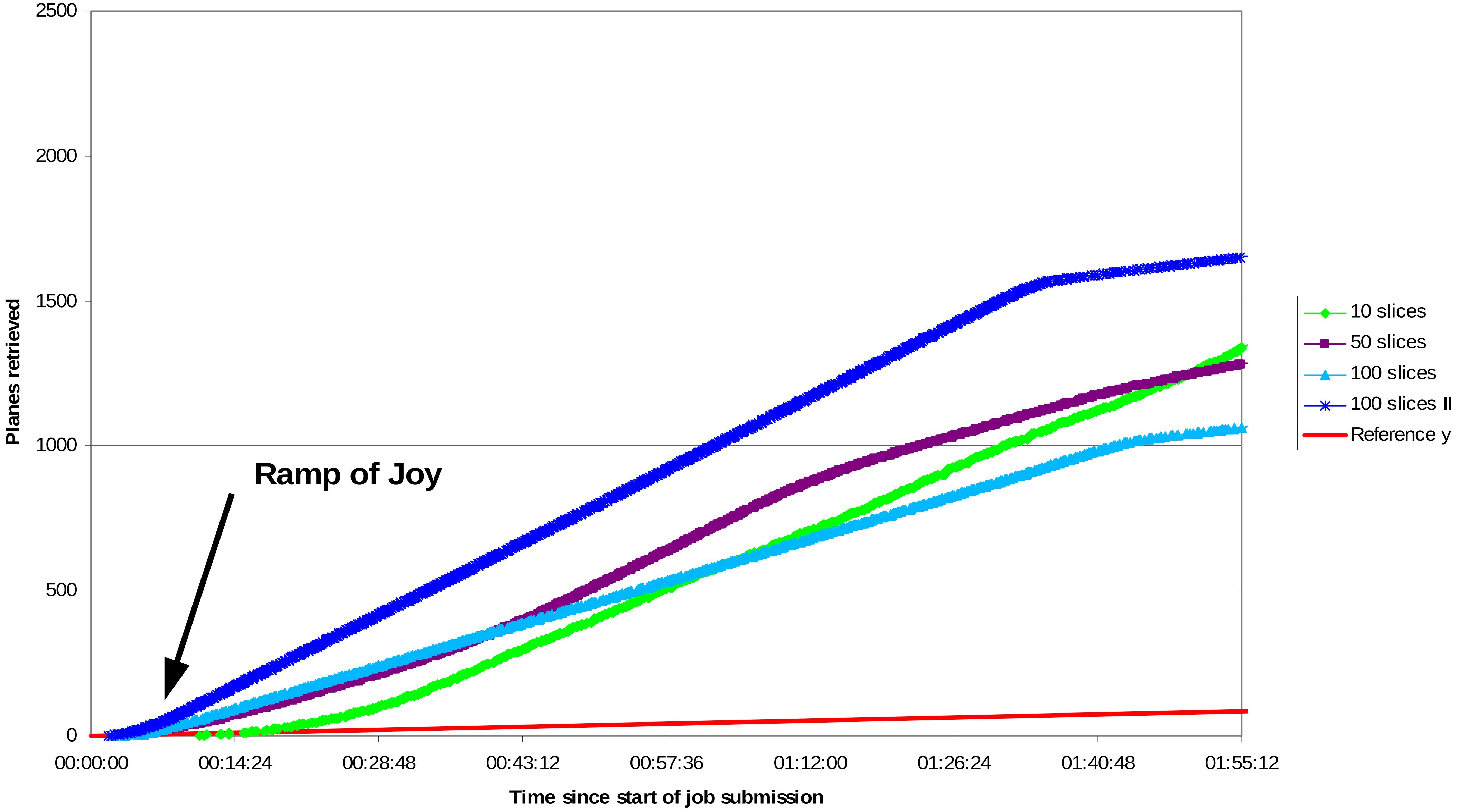
We've submitted batches of Grid jobs each between 10 and 100 single slices to reconstruct the water tank with 0.1mm axial spacing (total 2200 slices = 91 GB), and looked at how long it takes between starting the submission and all the replayed images arriving at the SE.

Each job loops, replaying a slice and immediately sending it back to the SE.

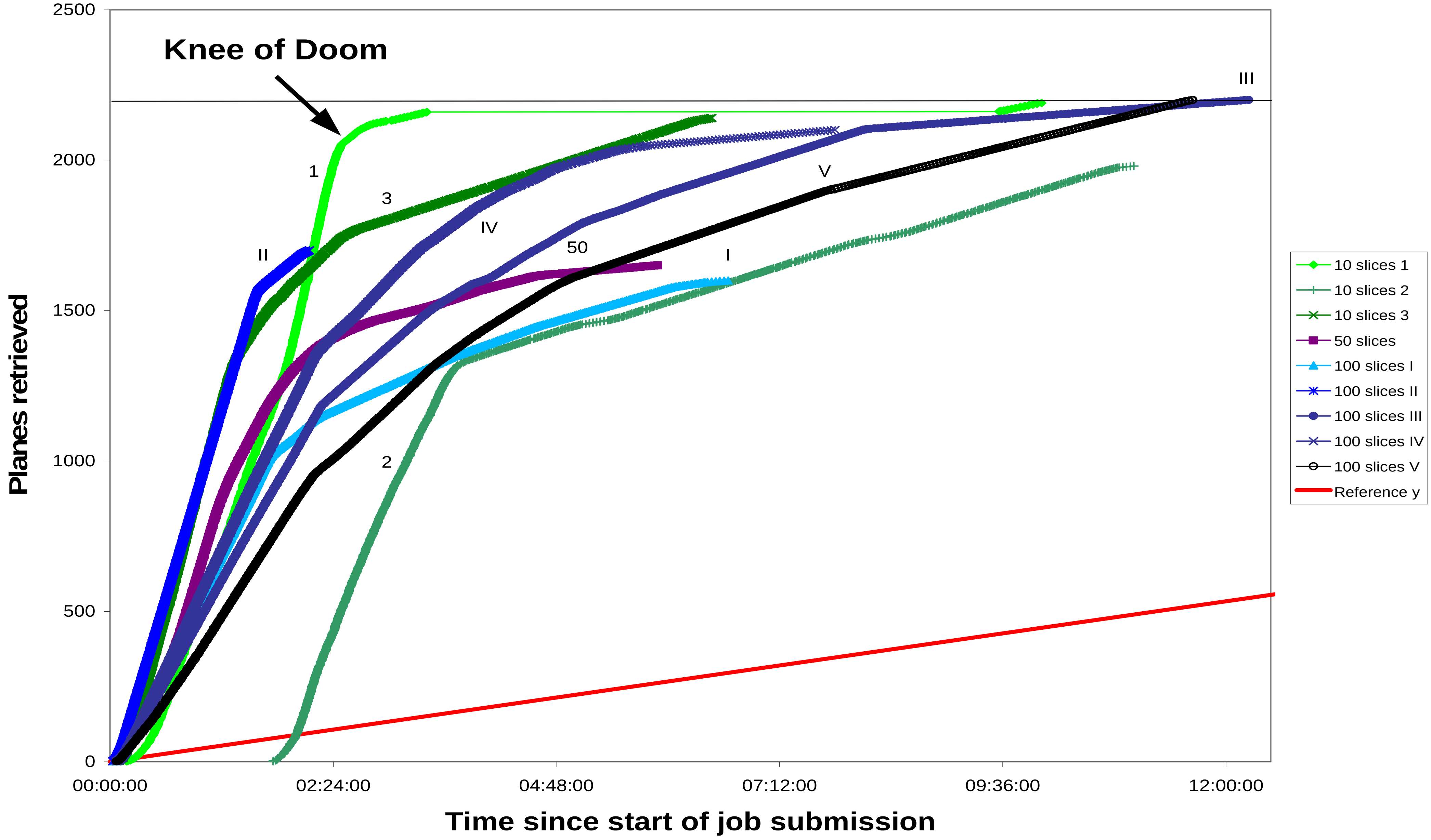
Compare this with replaying the slices sequentially on a single machine (~1'20" each on the UI).

- Most of the volume is generated much faster on Grid!
- Some stragglers, causing rate to slow.
- A significant proportion of jobs get lost completely

Job Progress



Job Progress



Results

- **Misconfigured sites and middleware problems mean that Grid job efficiency is about 90% - i.e. 10% of jobs fail completely (never return data).**
  The Grid infrastructure can resubmit them internally, but this can take >12 hours…
  (This feature was turned off here)

- **Need something to track progress so that gaps can be filled in quicker – e.g. automatically re-send all jobs not yet running when “Knee of Doom” reached.**

Conclusions

- We have a numerical replay code for digital holography that can be run on a major production Grid, and
- We have demonstrated the reconstruction and storage of a large sample volume using the Grid

- Latency poor, (but may not be an issue: “High Performance” vs. “High Throughput” Computing)

- Results will apply to any task with similar computing requirements and data access patterns – not specific to the algorithms used here

- Can “the Grid” help with the workflow – be more than just a number-crunching engine?

The End

Related hologrammetry applications

High-resolution imaging & measurement in 'dense' media:

- **Offshore inspection**
  - archiving, corrosion pitting, damage, dimensional measurement
- **Nuclear fuel inspection**
- **Bubble chamber diagnostics -** analysis of nuclear particle tracks
- **Marine life, organisms, bubble fields**
  - recording / monitoring of coral reefs
  - sediment transport / tracking
  - cavitation nuclei
- **Separation processes - crystallisation, flocculation/sedimentation**
- **3-d micrography of human eye**
- **Sampling from ice cores**
  - pollen, micro-meteorites, mammoths
- **Objects trapped in amber**

Other particle analysis:

- **Combustion processes & liquid atomisation**
  - water droplets, aerosols, clouds, snow and fog
- **Insect swarms**

To-do List

- Allow the Grid to resubmit on failure, time from SE not WN
- Optimise compilation and FFTW use
- CPU-specific optimisation.
- Bulk upload at end of job
- Scaling of object tracker
- Fast location of targets
- Virtual water!
- Focus metrics