Performance of the Distributed Central Analysis in *BaBar*

A. Khan, *Member, IEEE*, R. K. Mommsen, W. Gradl, M. Fritsch, A. Petzold, W. Roethel, and D. A. Smith

***Abstract*—The total dataset produced by the BaBar experiment at the Stanford Linear Accelerator Center (SLAC) currently comprises roughly 3×10^9 data events and an equal amount of simulated events, corresponding to 23 Tbytes of real data and 51 Tbytes simulated events. Since individual analyses typically select a very small fraction of all events, it would be extremely inefficient if each analysis had to process the full dataset. A first, centrally managed analysis step is therefore a common pre-selection ('skimming') of all data according to very loose, inclusive criteria to facilitate data access for later analysis. Usually, there are common selection criteria for several analysis. However, they may change over time, e.g., when new analyses are developed. Currently, $\mathcal{O}(100)$ such pre-selection streams ('skims') are defined. In order to provide timely access to newly created or modified skims, it is necessary to process the complete dataset several times a year. Additionally, newly taken or simulated data has to be skimmed as it becomes available. The system currently deployed for skim production is using 1800 CPUs distributed over three production sites. It was possible to process the complete dataset within about 3.5 months. We report on the stability and the performance of the system.**

***Index Terms*—Data handling, data management, data processing, distributed computing.**

I. Introduction

THE *BaBar* detector [1] at the Stanford Linear Accelerator Center (SLAC) in the US collects data from e^+e^- collisions at the B meson resonance. 230 fb^{-1} of physics quality data were collected in the years 1999 to 2004. This corresponds to 3.3×10^9 events or 23 Tbytes of reconstructed quantities, not counting raw data ($\approx$ 100 Tbytes) or multiple reconstruction passes. In addition, 3.4×10^9 Monte Carlo (MC) events have been simulated and reconstructed, adding another 51 Tbytes of reconstructed data.

A typical physics analysis selects only a tiny fraction of all events. Therefore, it is extremely inefficient for each analysis to loop over all data. Additionally, allowing many analysis jobs to access the central event store puts a high load on the system and requires a large local computing farm to provide the necessary CPU power and disk capacity. In order to circumvent these issues, the concept of a centrally managed pre-selection of events was introduced as first step in the analysis chain (see Fig. 1). The central analysis separates the events into $\mathcal{O}(100)$ streams, called 'skims', according to loose, inclusive physics criteria defined by the analysis groups or individual analysts. Therefore, the analysis-specific skim contains a higher purity of interesting events for a given analysis. On average, every event is found in 2 skims. The reconstruction quantities for selected events can be duplicated and are included in the skim. Thus, they can be easily distributed to different computing sites, grouped according to physics interests. At the same time, analysis-specific quantities can be added to the skims. This empowers the analysts to perform the final data fitting without a massive ntuple production on their own.

The skim definitions change over time as a better understanding of the physics and the detector evolves. Also new skims are devised, as new analysis topics emerge. Therefore, it is necessary to have several skim production cycles per year. Additionally, newly taken or simulated data needs to be processed as it becomes available. Since all analyses depend on the central skim production, a timely and reliable processing is essential.

II. Data Organization

In fall 2003 a new computing model [2] was deployed in BaBar. All data except the raw data is stored as ROOT [3] files. The ROOT files are administered by a relational database. The basic entity is a *collection*, uniquely defining a set of ROOT files with a common denominator. A *dataset* holds multiple collections with similar content. A dataset is dynamically defined by properties associated with a collection. E.g., a dataset can hold all collections from a given run period. Datasets are updated automatically when new collections fitting the corresponding definition become available. Tags, similar to CVS tags, are used to freeze a dataset. Tags can be used to mark a common data sample, e.g., the data used for analyses targeted at a specific conference.

The ROOT files are accessed using the xrootd protocol [4]. This protocol is implemented in the ROOT framework and allows a distributed access to the data. Additional tools have been developed to copy local files into the xrootd event store. For the central analysis, two types of xrootd event stores are used: a

Manuscript received January 3, 2006; revised June 16, 2006.

A. Khan is with the School of Engineering and Design, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K (e-mail: akram@slac.stanford.edu).

R. K. Mommsen was with the University of California, Irvine, Irvine, CA 92664 USA. He is now with the University of Manchester, Manchester M13 9PL, U.K. and Fermilab, Batavia, IL 60510 USA (e-mail: mommsen@slac.stanford.edu).

W. Gradl is with the Department of Physics, University of Edinburgh, Edinburgh EH9 3JZ, U.K. (e-mail: wgradl@slac.stanford.edu).

M. Fritsch is with the Inst. f. Experimentalphysik 1, Universitätsstrasse 150, Ruhr Universität Bochum, D-44780 Bochum, Germany (e-mail: miriam@ep1.ruhr-uni-bochum.de).

A. Petzold is with the Inst. f. Kern- u. Teilchenphysik, Technische Universität, Dresden, D-01062 Dresden, Germany (e-mail: a.petzold@physik.tu-dresden.de).

W. Roethel is with the University of California, Irvine, Irvine, CA 92697 USA (e-mail: roethel@slac.stanford.edu).

D. A. Smith is with the Stanford Linear Accelerator Center, Menlo Park, CA 94025 USA (e-mail: douglas@slac.stanford.edu).

Digital Object Identifier 10.1109/TNS.2006.881737

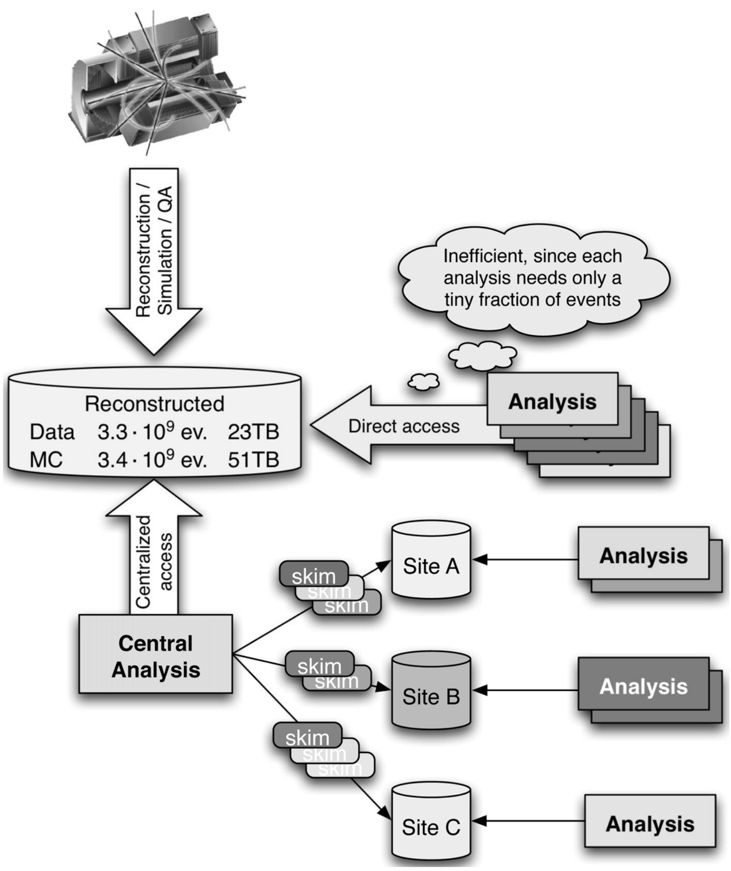
Reconstruction / Simulation / QA
Inefficient, since each analysis needs only a tiny fraction of events
Reconstructed
Data $3.3 \cdot 10^9$ ev. 23TB
MC $3.4 \cdot 10^9$ ev. 51TB
Direct access
Analysis
Centralized access
Central Analysis
skim
Site A
Site B
Site C
Analysis
Analysis
Analysis

Fig. 1. Sketch of the analysis flow in BaBar. The reconstructed data originating from the BaBar detector and Monte Carlo (MC) simulation is stored in a central event store. It is inefficient and requires large local resources to allow all analysis applications to directly access the central event store. Therefore, a central analysis as first step in the analysis processes splits the physics events in $\mathcal{O}(100)$ skims according to physics criteria defined by the analysis groups. These skims can be distributed to several remote farms where each analysis can access a purified sample of interesting events.

public event store where the reconstructed events and the skims are stored, and for temporary storage of the intermediate output produced during the skim production.

III. Central Analysis

The central analysis consists of several tasks handling different types of data. Each task uses as input one or several datasets. The tasks are distributed to different computing centers and assigned priorities for the processing. Each task processes all collections belonging to its associated datasets. As the processing of one collection typically takes several CPU days, it turned out to be more efficient to split each collection into smaller chunks that can be processed within 4 to 5 hours. This allows a more flexible handing of the CPU queues. It also saves a considerable amount of CPU time if a job crashes due to computing problems. Each job produces all skims which are defined for the given skim cycle. Each skim corresponds to a ROOT file, that is typically less than a few kbytes, since only a tiny fraction of the events is selected. Mass storage systems inefficiently handle small files and there is a considerable overhead in opening many small files during the analysis. Therefore, a second step merges several files belonging to the same skim and having some commonality into larger files. These files are imported into the mass storage system and their properties are inserted into the bookkeeping database, where the corresponding collections are created and added to datasets. The analyst can access the skimmed collections using datasets.

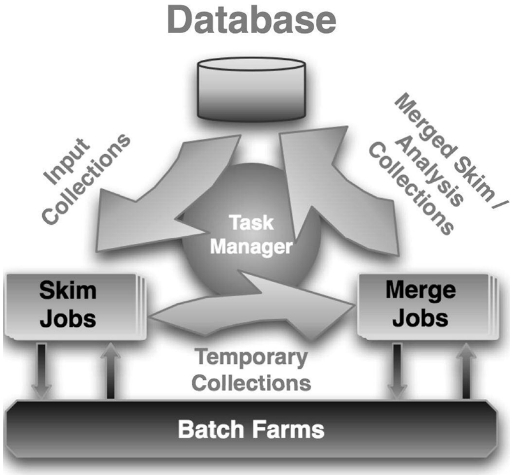

Fig. 2. The Task Management controls the job creation and batch execution as well as the bookkeeping of processed input collections and produced temporary and final collections.

The central analysis needs a huge amount of CPU power to process all available data in a reasonable time. In order to achieve a timely production, the load is distributed over multiple

TABLE I
PROCESSED EVENTS AND OUTPUT PRODUCED DURING THE FIRST FIVE SKIM CYCLES

Skim Cycle	# Skims	Processed Events		# Collections Produced		Total Size (Tbytes)	
		data	MC	data	MC	data	MC
R14	125	3.3×10^9	2.3×10^9	22 768	80 272	32.6	54.0
R14a	23	1.7×10^9	–	2 208	–	4.3	–
R16a	161	3.3×10^9	3.1×10^9	15 384	67 865	24.1	65.8
R16b	31	3.3×10^9	2.6×10^9	5 182	21 326	8.7	22.8
R16c	7	3.3×10^9	2.6×10^9	1 162	4 809	2.5	5.3

computing centers. Up to now we used the computing centers at the University of Padova/INFN (Italy), GridKa (Karlsruhe, Germany) and SLAC (California, US). The assignment of tasks is agreed upon before a given skimming cycle starts. The data distribution is done using multi-stream file transfer tools. In order to facilitate the bookkeeping of processed and available collections, all input data was transfered from SLAC to the remote sites and the finished skim collections were copied back into the SLAC event store before redistributing them to the analysis centers.

The Task Management (TM) framework [5] is used to manage the central analysis (Fig. 2). The TM handles the job submission and control. It creates the skimming jobs for the input collections, assuring that each input collection is only processed once. It submits them to the batch system and checks for their successful execution. In case of job failures, a basic failure classification is done and jobs likely to succeed on a further try are resubmitted. The TM keeps track of the temporary output collections and determines which ones can be merged together. It also handles the creation and submission of the merge jobs as well as the final import into the event store and bookkeeping database.

Further tools were developed during the skim production to improve the flexibility in automatically handling multiple parallel tasks on the batch system and to keep the batch queues filled. Monitoring tools were used to create websites with progress and status information. Additional tools were needed for managing the data distribution to the remote sites and for database consistency checks.

IV. PERFORMANCE

Given that all analyses depend on the availability of skims, the timely production of skims is essential for the success of the physics program. The aim is to have 3 to 4 skimming cycles per year. In each cycle, only modified or newly defined skims need to be produced unless a new reconstruction was performed. For newly available data all defined skims need to be produced.

Table I lists the skim cycles completed so far. The first skim cycle R14 was started at the beginning of 2004 and produced all skims defined at that time. The R14a skim cycle fixed a few bugs present at the beginning of data skimming. The subsequent R16a skim cycle started in late December 2004 producing 91 newly defined and 70 redefined skims. The following two skim cycles produced a total of 38 new skims. Currently, a full reconstruction and new simulation effort is ongoing which is being skimmed as the reconstructed data becomes available.

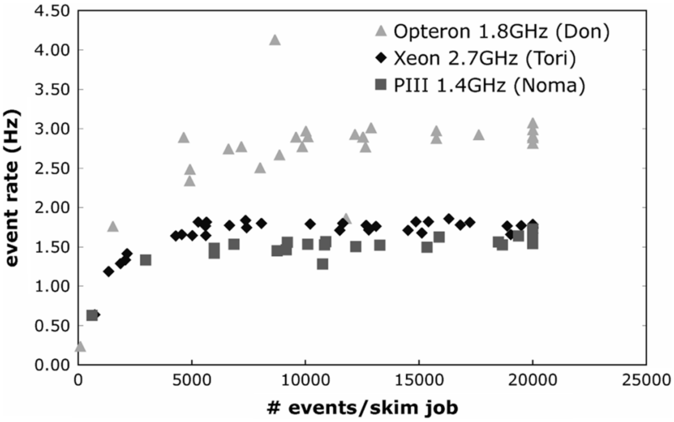

Fig. 3. Event rate for data skimming as function of the number of events processed by one job, comparing the different CPU types available at SLAC. The default number of events processed by one job is 20000. The event rate is using wall clock time and includes the overhead for initializing and checking of the output.

In the following, we will look at the performance and reliability of the last large scale skim production cycle R16a.

The batch queue used at SLAC consists of approximately 450 PIII 1.4 GHz, 300 Xeon 2.6 GHz, and 220 Opteron 1.8 GHz machines. The actual number of CPUs varied due to machine failures and queue allocation policies. The Opteron machines became available during January 2005 and their number gradually increased over the following month. All machines are dual-CPU machines with 2 Gbytes memory. Fig. 3 shows the processing rate versus the number of events processed by a skim job running over real data. The processing time is wall clock, i.e., it includes all overheads from starting job and checking the output for consistency, but does not including the batch queuing time. It can be clearly seen that below 5000 events the overhead for initializing and checking dominates, while for larger chunks the processing rate is constant. The skim performance of PIII and Xeon processors is comparable, while the rate of the Opteron machines is 1.8 times higher. A similar behavior is observed for MC events. The processing rate depends on the type of simulated events being processed. The processing rate is reduced by a factor 5 for the most demanding MC ('B generics') which include the full spectrum of possible B decays overlaid with background data.

Fig. 4(a) shows the number of successfully skimmed events versus time. After a slow startup phase over Christmas 2004, the real data was skimmed within 1.5 months. The bulk of the MC skimming was finished two months later. Roughly 10% of the

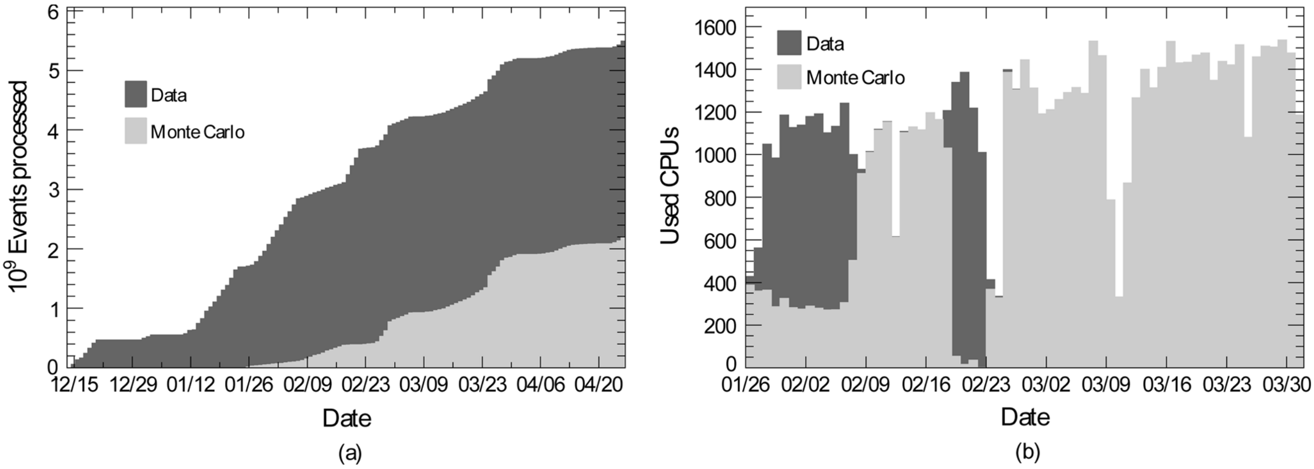

Fig. 4. Performance of the last large scale skim production cycle R16a. On the left we show the cumulative number of events skimmed versus time (shown in month/day format for the years 2004/2005). After a slow startup phase over Christmas 2004, the real data was skimmed within 1.5 months, while the bulk of simulated Monte Carlo events was skimmed 2 months later. On the right the number of CPUs used at SLAC versus time during the main production period in 2005 is shown. The number of available CPUs increased during this time and except for short periods of time, all CPUs were busy. (a) Number of successfully skimmed events versus time. (b) Number of CPUs used at SLAC versus time.

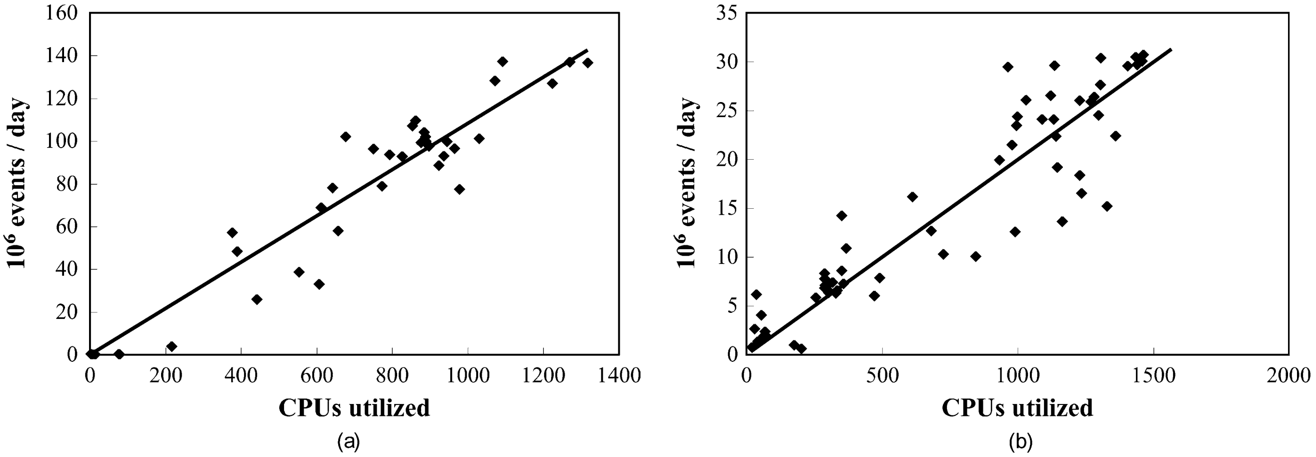

Fig. 5. Scalability of the system shown for data (left) and Monte Carlo (right). The number of processed events per day scales nicely with the number of CPUs used. The linear fit to the data points yields 10M (2M) events/day/100 CPUs for data (Monte Carlo).

jobs had to be resubmitted due to machine failures or I/O problems. Fig. 4(b) depicts the queue usage at SLAC during the main production period. It shows that it was possible to constantly use all available CPUs at SLAC. The occasional lower CPU usage was due to the limitations in the available disk space used for storing the temporary collections.

V. Scalability

The number of CPUs devoted to a given task changed over time. This allows the investigation of the scalability of the system as function of used CPUs. Fig. 5 shows the number of events being successfully processed per day versus the number of CPUs utilized. We observe a linear behavior. The scatter of the points around the fitted line is mainly due to the fact that we did not correct for the different processing speeds of the machines involved. The linear fit yields 10 M (2 M) events/day/100 CPUs for data (Monte Carlo). This rate includes the time needed for initializing the jobs, checking the produced output, and rerunning of failed jobs. It also includes the overheads from bookkeeping, but does not account for batch queuing time.

VI. Conclusion

The concept of a central analysis within the new computing model based on ROOT files has been successfully implemented. Since its deployment in fall 2003, 5 skim production cycles have been completed. We demonstrated that it is possible to process the huge data sample in a reasonable time and to provide skims to the analysts. With a special effort before a major conference, we succeeded to have skims available for data taken only eight days before. The system enables a reliable and large scale operation and shows an excellent scaling behavior. Given the recent experience, the target of running 3 to 4 skim cycles per year in a timely fashion is achievable. We successfully processed more than 25×10^9 events and produced 220 Tbytes of skims using in excess of 600 000 CPU-days at three computing centers.

For future development, skim production in the grid is an interesting option. We are already running at three different computing centers, and the Task Management system demonstrated the feasibility for running in a distributed environment. The centrally defined skim applications are more stable than individual user analyses which makes the code distribution easier. In contrast to simulation tasks which already run routinely in the grid, skimming could be an application to test the data access within the grid, without the difficulties of unpredictable access patterns imposed by normal analysis tasks.

Acknowledgment

This work builds on a large body of development by the BaBar Computing Group, and would not have been possible without a strong collaborative effort.

References

[1] B. Aubert, "The BaBar detector," *Nucl. Instrum. Methods Phys. Res. A*, vol. A479, pp. 1–116, 2002.

[2] D. N. Brown, "The new BaBar analysis model," in *Proc. CHEP '04*, Interlaken, Switzerland, Sep. 2004.

[3] R. Brun and F. Rademakers, "ROOT—An object oriented data analysis framework," *Nucl. Instrum. Methods Phys. Res. A*, vol. A389, pp. 81–86, 1997.

[4] A. Dorigo, P. Elmer, F. Furano, and A. Hanushevsky, "XROOTD—A highly scalable architecture for data access," in *Proc. WSEAS Conf.*, Prague, Czech Republic, Mar. 2005.

[5] W. Roethel, "The BaBar analysis task manager," in *Proc. CHEP '04*, Interlaken, Switzerland, Sep. 2004, pp. 348–353.