Improving Performance of Object Oriented Databases, *BABAR* Case Studies

Jacek Becla[1]

[1] Stanford Linear Accelerator Center, USA. For the *BABAR* Computing Group

Abstract

The *BABAR* experiment at SLAC is designed to perform high-precision studies of the decays of the B-meson, that are produced in e+e- collisions at the PEP-II accelerator. Since it was brought on-line in May 1999, it is continuously increasing luminosity approaching designed goal of recording 30 hadronic events per second, or $3*10^8$ events per year. In order to promptly process and store persistently all generated events, a powerful processing software and hardware is essential. Current estimates of the number of computing nodes needed by first level reconstruction range from 200 to 350. Providing correctly configured hardware and software for physicists' analysis become equally important and even more complex task. The part of the system responsible for concurrency issues in this environment is a database component. The paper will describe performance studies, chosen hardware and software configuration of the underlying Objectivity/DB database system used by *BABAR* Project.

Keywords *BABAR*, databases, Objectivity/DB

1. Introduction to *BABAR* Database System

The goal of the *BABAR* experiment is the detailed study of the difference between matter and antimatter. The project is composed of an international collaboration of physicists and engineers from 10 countries, and it is headquartered at the Stanford Linear Accelerator Center (SLAC).

The PEP-II linear accelerator used by the experiment (also located at SLAC) has been designed to generate data at a rate of about 32 MB/sec, or $3*10^8$ events per year. All that data has to be stored persistently, and then reconstructed within eight hour of the data's collection. Reconstruction runs asynchronously with data taking, on multiple computing nodes in a fully controlled environment. The output from the reconstruction process is then passed to physicists, who are analyzing data. Data analysis is performed in a non-controlled way, where over a hundred physicists are allowed to look at the data as well as generate new persistent data.

The component of the *BABAR* Software responsible for handling all the persistent data is called a Database System. The estimated size of 1-year's worth of real and simulated data is 300 TB, and the total amount of data generated during the lifetime of the experiment is expected to approach the multi-PB region[1]. Handling this much data is non-trivial: the software has to be very robust, well optimized, and it requires a lot of computing power. The system is already in production since the first data was taken in May'99, but it is still not in its final shape. The process of optimizing the system: choosing the right hardware configuration and tuning the software is an ongoing and very lively activity inside the *BABAR* Database Group.

[1] We are not aware of any other system maintaining comparably large amount of data using an ODBMS.

2. Some System Design Aspects

The Database System used within *BABAR* is based on two commercial products: High Performance Storage System (HPSS) and an Object Oriented Database System: Objectivity/DB. The persistent part of the system is hidden behind a transient-to-persistent and persistent-to-transient wrapper, relieving users from the burden of learning and directly using the two mentioned systems. For more information, please refer to [1], [2].

The Data Acquisition System (DAQ), Online Prompt Reconstruction (OPR) and Data Analysis are three, fairly independent parts of the system, each having different set of requirements. In order to provide each of them with the environment they need, as well as to avoid unnecessary collisions between them, they have been assigned separate *federations*[2] and dedicated servers. Data is exchanged between the federations using specialized, homegrown export/import tools. The rest of this document will focus on the most interesting aspects of tuning an undoubtedly very large system.

2.1. Tuning the system

When the project was launched in May'99, the database system was far from being able to satisfy the stringent requirements of the experiment. OPR was designed to reconstruct events with an input rate of 100 Hz, and it was expected that approximately 200 computing nodes would have to be used in parallel to achieve that[3]. In practice the system was able to deliver only 6-8 Hz, and saturation was observed on a 50-node farm. Similarly, data analysis was running at a much slower than expected pace.

2.1.1. Testbed

Since OPR is running in a fully controlled environment, and there is no good way of controlling data analysis environment, it was natural to start optimizing the OPR farm. It was also clear, that many of the problems and bottlenecks in both cases are the same. Thus, the optimizations for the OPR farm could be easily applied to the data analysis farm. The decision to establish a dedicated test-bed and focus on scalability and performance issues was made soon after the start of the experiment. The test-bed initially consisted of 100 processing nodes and two data servers. However, when the tests progressed, the number of nodes reached 230, and 3 more data servers were added. Jobs run in the testbed were an exact copy of the real production jobs. They were connected to test federation(s), created specifically for the sake of the tests.

A monitoring system was developed in order to understand the causes of the various bottlenecks and delays. Due to a large number of shared resources (such as Objectivity/DB data servers, Objectivity/DB Lock Server, NFS and AFS servers, network switches, database files, files containing input data) the monitoring system ended up being quite complex.

2.1.2. Testbed results

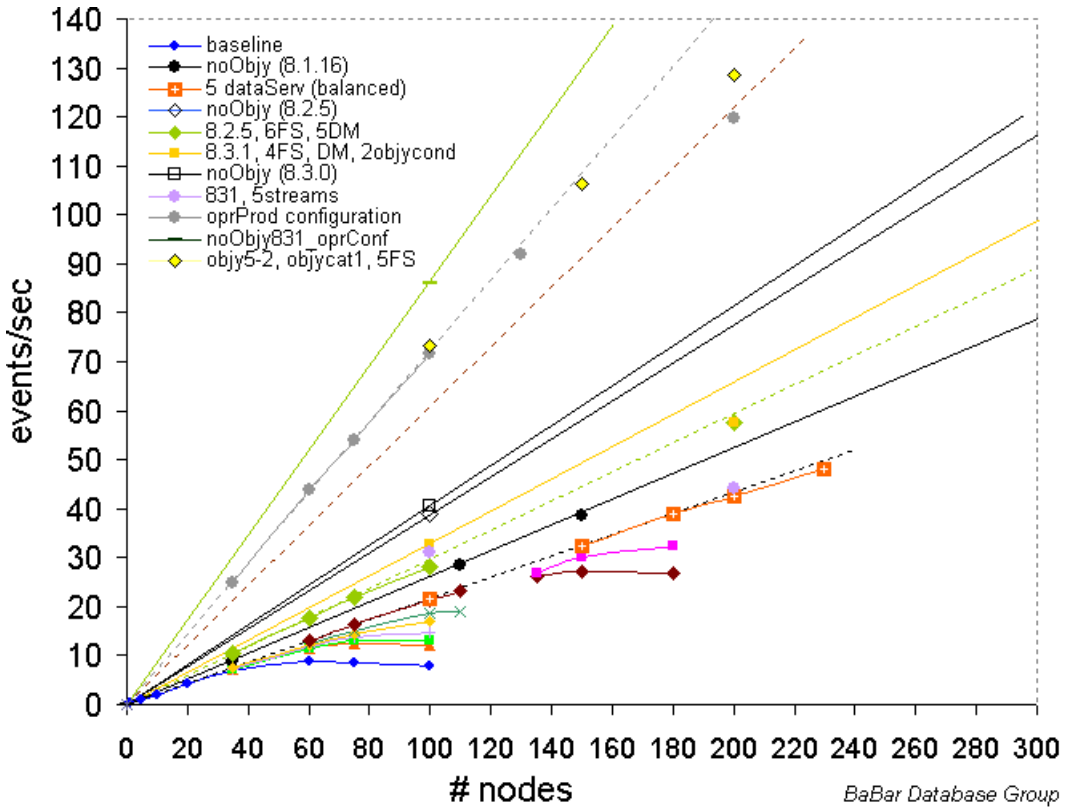
After 5 months of operation of the testbed, changing hardware configuration, improving the software and adjusting various settings, the system is able to deliver around 130 Hz, and scale beyond 200 nodes.

---

[2] "Federation" is the highest level in the Objectivity/DB logical storage hierarchy.

[3] Soon the accelerator will start producing events with rates higher then the designed 100 Hz, and the estimate number of nodes used for Prompt Reconstruction is expected to grow to 300-350.

The following plot shows some major phases of the tests and their influence on the overall performance of the system. Due to paper length constraints, the plot cannot be explained in detail. However, the top most line represents the best possible performance obtainable and is limited by the reconstruction code. Subsequent lines represent the processing rate for various configurations. When the tests were started, all of the events were fully processed. Subsequent to changes in the OPR production configuration, we started to filter out some events in the first phase of the reprocessing and ceased to persistently store them. This changed the processing rate measured on the input side by about a factor of 3.



Tuning the system involved adjusting many configuration parameters. Below are only some of them, especially those that gave the most significant improvements.

- Increased the file descriptor limit on all machines running Objectivity/DB data servers. The default value used by the Unix is much too low for the servers, which have to communicate with 200 clients simultaneously trying to open multiple connections.
- Increased the number of data servers, number of file systems, and balanced the load evenly.
- Ran multiple instances of Objectivity/DB data servers per machine[4].
- Pre-sized containers during creation to reduce the number of extensions.
- Used several database clusters[5]; we currently use 4.
- Tuned and randomized the transaction length, as well as the Objectivity/DB cache size (on the client side)
- Reduced dependencies on NFS and AFS to the minimum.

[4] This is still the case in the latest Objectivity/DB release even when running multi-threaded servers.
[5] A *database cluster* is a set of databases shared by a subset of clients. Each subset shares only meta-data, databases containing real data are not connected to more then one subset of clients.

Most of the improvements were already transferred from the testbed into the production environment. Unfortunately, due to a growing number of required production nodes, we are no longer able to maintain an independent production farm and test-bed. Most of the tests are now scheduled and require production outages.

Although we are currently able to process data with a rate higher than the designed 100 Hz, further improvements are still needed. The quoted processing rate is measured during steady processing. Due to various operational delays (mostly non-database related) the average processing rate measured over 24 hour period is still much lower then the one required. Fortunately, there are still dozens of improvements to be explored and applied; all of which we hope will give a significant performance boost. These include adding more data servers, reducing payload per event, improving performance of the Objectivity/DB Lock Server[6] and reducing startup time.

2.1.3. Analysis results

Similar improvements are gradually being applied for physics analysis. Of particular importance are the matching of the number of database servers, the number of cpus per server, and the number of filesystems per server. Performance optimizations in the analysis programs themselves have resulted in improvements from 35Hz to 2kHz for one particular benchmark where events are selected on the basis of so-called tag filters.

3. Conclusions

The *BABAR* Database System is responsible for persistent storage and providing access to the data. Given the volume of data, which is likely to enter the petabyte region in two years, the system is intrinsically very complex and requires a lot of computing power. Currently, the major focus is on improving the performance and scalability of the system.

Tuning the system in an on-going process. After the first 5 months of work, we were able to increase the throughput from ~8 Hz up to 130 Hz. At the same time scalability has been improved from ~50 nodes to over 200 nodes. Most likely the processing farm will be expanded up to 300-350 nodes in the near future, together with the throughput requirements. The performance tests will continue for at least the next few months.

Objectivity/DB, one of the very few commercial systems used within the *BABAR* software, proved to be a very robust and reliable system. It was able to handle many terabytes of data along with hundreds of simultaneous clients.

References

1 J. Becla, "Data clustering and placement for the BaBar database", CHEP'98, Chicago, Summer 1998
2 A. Hanushevsky, "Pursuit of Scalable High Performance Multi-Petabyte Database", 16th IEEE Symposium on Mass Storage Systems, San Diego, CA, Spring 1999
3 D. Quarrie, "Operational Experience with the BaBar Database", CHEP'00, Padova, Winter 2000
4 Objectivity/DB, Inc website: http://www.objectivity.com

[6] Scheduled for one of the next Objectivity/DB releases in the next few months.