The BaBar Database: Challenges, Trends and Projections

Jacek Becla
Stanford Linear Accelerator Center
For the BaBar Computing Group
### Acknowledgements

- Adeyemi Adesanya¹
- Igor Gaponenko²
- Andrew Hanushevsky¹
- Adil Hasan¹
- Akbar Mokhtarani²
- David Quarrie²
- Simon Patton²
- Artem Trunov¹

¹ Stanford Linear Accelerator Center
² Lawrence Berkeley National Laboratory
BaBar

- Studying difference between matter and antimatter (CP violation)
- Based at SLAC
- ~800 physicists, 12 countries
- In production
  - Run 1: Oct’99 – Oct’00, ~23.5 fb\(^{-1}\)
  - Run 2: Feb’01 – Jul’02, up to 20.3 fb\(^{-1}\) now
Making It Persistent

- Persistency mechanism based on an Object Oriented Database Management System - Objectivity/DB
- Enhanced by custom-designed C++ code
It’s All In The Data

350 TB

- Raw data
  - 10^8 simulated events
  - 189*10^6 hadronic events
  - 2*10^9 events
  - 550 K collections
- Rec data
- Esd data
- Aod data
- Aod data

170,000 Files
Data Size Growth

BTW, officially the largest db in the world: 10.5 TB
BaBar Software

Reconstruction, Analysis and Simulation Code
- Over 3,000,000 lines of C++ code, developed by hundreds of physicists and engineers

Transient-Persistent Wrapper
- Over 500,000 lines of C++ code:
  - complex persistent schema
  - 500 persistent classes
  - placement/clustering strategies
  - metadata maintenance
  - indices, admin tools

BaBar Database Software
- Objectivity/DB Client and Server
  - supports complex schema
  - provides C++ bindings
  - acts as typical database engine
    - transaction processing
    - journaling, recovery
    - page layout
    - indices

SLAC Enhancements
- Veritas File System
  - optimized logical file system
    - disk cache manager
    - mass storage system interface
    - real time monitoring
    - added security

Hundreds of disks combined into a small number of large volumes. Serves as “disk cache”

Home-Grown Commercial

HPSS
- Database scalability to a petabyte and beyond is achieved by writing all database files to tapes managed by HPSS. Access for physics analysis requires that the files be staged back to the tens of terabytes of disk cache.
Data Flow
DB System Evolution

- Production started in Oct’99
  - Initial design requirements (100 Hz, 200 TB/year) met last year
  - Many unexpected issues, bottlenecks
  - Changing physics requirements
  - Challenge to keep up with improving accelerator performance

- Improving, optimizing and tuning, redesigning, scaling
  - since day one, still ongoing
Initial Challenges

- Architectural limit
  - 64 K DB files per federation

- Robustness
  - A single failure may affect the whole farm

- Main bottleneck - lock server

- Miscellaneous
  - Does it really scale?…x10?, x20, x100?
  - Poor space utilization
    - Event size 3-4x larger than the design goal
  - Poor efficiency per server
64 K Database Limit

- Forced to deal with large files (10 and 2 GB)
  - Inefficient (tape drive usage, data distribution)
  - Issues with large file support
  - Does not solve the problem, just delays it

- Working with Objectivity on multi-petabyte support
  - Several possibilities examined
  - Final choice: multiple federations
Multiple Federations

- A single application can simultaneously access multiple federations
- BaBar DB architecture maps well
  - Concept of domains (event store, conditions….)
  - Now domains mapped onto different fds, rather than regions of one fd (*divested domains*)
Inter-FD Navigation

♦ Long reference
  - Initially (May’99) not available
  - Backwards compatibility issues – cannot be widely used
  - Use restricted to event collections
    - Bridge collections
      see talk #4-017

This has been implemented, released, and is being introduced in production now. True long ref support by Objectivity ~ Oct 2002
Online Prompt Reconstruction

Run 1
- Kept up with data (on the edge)
- A lot of tuning, optimization involved

Run 2
- Major change in physics requirements: streaming data into 21 streams (used to be 4)
  - 5 times more locks to handle
  - 5 times more files to handle (higher load on data servers)
- Smaller files (2 and 0.5 GB)
- Clustering hint server and oid server + more tuning, more hardware
Centralized Metadata Mgt

◆ CORBA Clustering Hint Server
  - Pre-creates, pre-sizes, de-sparsifies, distributes, closes full databases
  - Features:
    • Reduces lock traffic (no container extensions), de-randomizes I/O, improves placement (containers reused), optimizes disk space usage, full integration with HPSS, many others

◆ CORBA OID server
  - Locates condition interval, distributes to clients
  - Features:
    • Reduces disk I/O, lock traffic

◆ See talk #4-039 for more details
Scaling OPR

- 2 CORBA servers + tuning (commitAndHold, number of client file descriptors, many others)
  - Performance (21 streams) ~ the best seen with 4 streams

- Number of data servers rapidly grows

- High-end CPU-s for lock server desired

- It is keeping up now, but
  - clearly that is not enough
    - continuous luminosity increase
Recorded and available luminosity as of 20010829

- **Level3 lumi ON+OFF**
- **Processed lumi**
A Typical OPR Run

![Graph showing processing rate over time](image-url)
Current OPR Configuration

EVENT STORE + COND

Data Server
Data Server
Data Server
Data Server

220 clients

100 Hz (events/sec)
...Where Do We Go From Here?

- Use the power of multiple federations
  - Multiple event store federations
  - One federation per data server
- Separate rolling calibrations from reconstruction
- Process multiple runs in parallel (asynchronously)
- Reconfigure disks
  - ~3x client/server improvement
- Focus on improving performance of Objectivity data server (AMS)
Future OPR Configuration

Expected to be ready ~ March 2002

Data Server

EVS001
LS
JNL
CHS
100 Clients
45 Hz

EVS002
LS
JNL
CHS
100 Clients
45 Hz

EVS003
LS
JNL
CHS
100 Clients
45 Hz

EVS004
LS
JNL
CHS
100 Clients
45 Hz
Benefits of the New System

- Allows to scale
  - Just plug in new federations and servers, no unexpected surprises with scalability, no new bottlenecks

- Lock server is no longer a bottleneck
  - Many lock servers used, (BTW, does not increase cost)
  - Now AMS performance dictates per-server performance

- Very robust
  - A single failure brings down a small piece of the system

- Improved daily average efficiency
  - Shorter startup time, no finalize, steady processing rate dominates
  - Shorter outages (each fd independently)

- Easy to test
  - one data server needed (cannot afford tens of servers in test-bed)
Reaching the current level was very difficult to achieve, see various papers May99-Sep01.
Analysis

- 200 CPUs (~Sun Netra T1 like)
- 17 servers, 27 TB Objectivity disk cache
- On demand staging turned off
- Read only databases

- Veritas File System reconfiguration
  - direct I/O instead of buffered I/O
    - more than doubles effective data rate

- Disk space – always a problem
  - mini – 4.7 KB/event (esd)
  - micro – 5.4 KB/event (aod, col, tag, evt, evshdr)

Recent computing model: enhance Objy, deprecate Kanga (ROOT based): freeze by Mid 2002, produce Kanga files till late 2002
Scaling Analysis

- Multiple-federations
- A new fd every few weeks
- A bridge federation and bridge collections bind all federations together
- A pool of lock servers - scalable
Data Distribution

- Micro-level data mirrored @ in2p3
- Run2 – mirror raw as well
- Current tools do not scale with increased data volume
  - a lot of manual work
- Will try using data grid based tools soon
- See #10-040: US Grid Projects (plenary), and #4-021
Space Optimizations

- **Is it worth to compress?**
  - CPU cost vs disk cost
    - 1 TB =$= $ Sun E420 now
    - disk cost going down faster than CPU cost
  - See talk #4-022 to find out!

- **Working on inefficiencies in data model and placement**
  - Size data types appropriately
  - Avoid repeating patterns
What Powers the BaBar DB
(SLAC only)

- **Servers**
  - 44 data servers (mostly 4-CPU Sun)
  - 57 Secondary servers

- **Disk Space**
  - 51.1 TB total:
    - 26.6 Analysis
    - 19.3 OPR/REPRO
    - 3.6 SP
    - 1.6 User federations

- **Clients:**
  - 1000 Sun Netra T1 (1 CPU, 450 MHz, 256 MB RAM)
  - 512 Pentium III (2 CPU, 866 MHz, 1 GB RAM)
Platforms

- Support for Sun Solaris and Linux
- Servers – all Sun Solaris
- Clients
  - Analysis – mostly Linux
  - OPR – still Sun
  - REPRO – Linux
  - Simulation Production - Sun at SLAC, Linux elsewhere
- Linux clients
  - 2.2x faster (comparing to Sun Netra T1)
  - Still not stable enough
    - SLAC/VA Linux specific problem

reprocessing being made to work in Italy (INFN), timescale mid 2002, all Linux based, including servers
Initial Challenges → Solution

- **Architectural limit** → multiple federations
  - 64 K DB files per federation

- **Robustness** → one federation per data server
  - A single failure may affect the whole farm

- **Main bottleneck - lock server** → one fd per server

- **Miscellaneous**
  - Does it really scale?...x10?, x20 → one fd per server, x100? → ?
  - Poor space utilization → removing inefficiencies in data model
    - Event size 3-4x larger than the design goal
  - Poor efficiency per server → dispersify, reconfigure disks
# Prognosis

<table>
<thead>
<tr>
<th></th>
<th>FY00</th>
<th>FY01</th>
<th>FY02</th>
<th>FY03</th>
<th>FY04</th>
<th>FY05</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak luminosity $[10^{33} \text{ cm}^{-2} \text{ sec}^{-1}]$</td>
<td>2.5</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>13</td>
<td>24</td>
</tr>
<tr>
<td>Yearly integrated luminosity $[\text{fb}^{-1}]$</td>
<td>25</td>
<td>40</td>
<td>80</td>
<td>115</td>
<td>135</td>
<td>225</td>
</tr>
<tr>
<td>Total Integrated luminosity $[\text{fb}^{-1}]$</td>
<td>25</td>
<td>65</td>
<td>145</td>
<td>260</td>
<td>395</td>
<td>620</td>
</tr>
</tbody>
</table>
Future Activities

◆ We still have many interesting problems to solve
  – How to handle 10x more data per day?
  – How to place a petabyte of data on 50 TB of disks?
  – What is the ultimate OPR configuration?
  – How to index $>10^{15}$ tags, preferably on tens of attributes?
  – How to speed up analysis?
    ▪ Parallel iteration
    ▪ Dynamic load balancing
Summary

Is managing 100s of TBs of data understood?
- OPR – finally we know how to scale it
- Current focus on Analysis and reducing costs
  - We have ideas, time will show whether we succeeded
  - It is definitely non-trivial
  - Hard work, costly (money and time)

ODBMS technology
- A good choice – fits BaBar requirements for major center-based production and analysis
  - Still a perceived challenge to set up Objectivity in small centers
- A lot of custom-designed code written on top
Compilation of BaBar Talks

- **Oral**
  - 4-017 Support in the BaBar Objectivity Database for Multiple Federations
  - 4-021 The BaBar Distributed Computing Model
  - 4-039 Optimizing Parallel Access to the BaBar Database System Using CORBA Servers
  - 4-022 To Compress or Not to Compress: Objectivity/DB Compressed Database Support

- **Poster**
  - 4-023 Distributing File-based Data to Remote Sites Within the BaBar Collaboration
  - 4-042 Managing the BaBar Object Oriented Database