CHEP 2000, Padova, Feb 2000

Operational Experience with the BABAR Database

David R. Quarrie
Lawrence Berkeley National Laboratory
for *BABAR* Computing Group
DRQuarrie@LBL.GOV

Acknowledgements

D. Quarrie[5], T. Adye[6], A. Adesanya[7], J-N. Albert[4], J. Becla[7], D. Brown[5], C. Bulfon[3], I. Gaponenko[5], S. Gowdy[5], A. Hanushevsky[7], A. Hasan[7], Y. Kolomensky[2], S. Mukhortov[1], S. Patton[5], G. Svarovski[1], A. Trunov[7], G. Zioulas[7]

for the *BABAR* Computing Group

[1] Budker Institute of Nuclear Physics, Russia

[2] California Institute of Technology, USA

[3] INFN, Rome, Italy

[4] Lab de l'Accelerateur Lineaire, France

[5] Lawrence Berkeley National Laboratory, USA

[6] Rutherford Appleton Laboratory, UK

[7] Stanford Linear Accelerator Center, USA

Introduction

- Many other talks describe other aspects of *BABAR* Database
  - A. Adesanya, *An interactive browser for BABAR databases*
  - J. Becla, *Improving Performance of Object Oriented Databases, BABAR Case Studies*
  - I. Gaponenko, *An Overview of the BABAR Conditions Database*
  - A. Hanushevsky, *Practical Security in large-Scale Distributed Object Oriented Databases*
  - A. Hanushevsky, *Disk Cache Management in Large-Scale Object Oriented Databases*
  - E. Leonardi, *Distributing Data around the BABAR collaboration's Objectivity Federations*
  - S. Patton, *Schema migration for BABAR Objectivity Federations*
  - G. Zioulas, *The BABAR Online Databases*
- Focus on some of the operational aspects
  - Lessons learnt during 12 months of production running

Experiment Characteristics

Characteristic		*Size*
No. of Detector Subsystems		7
No. of Electronic Channels		~250,000
Raw Event Size		~32kBytes
DAQ to Level 3 Trigger	2000Hz	50MByte/sec
Level 3 to Reconstruction	100Hz	2.5MByte/sec
Reconstruction	100Hz	7.5MByte/sec
Event Rate		10^9 events/year
Storage Requirements (real & simulated data)		~300TByte/year

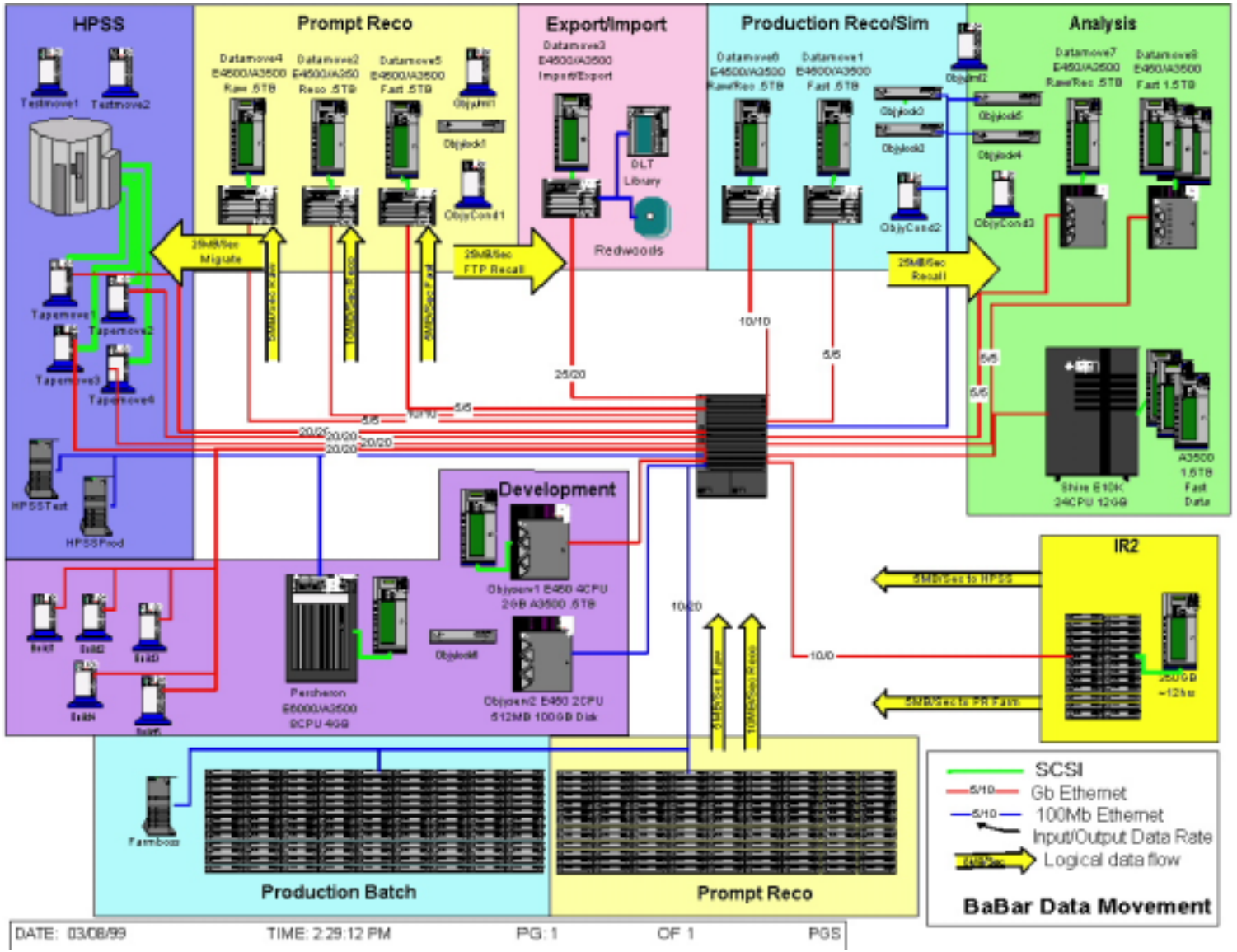
Performance Requirements

- Online Prompt Reconstruction
  - Baseline of 200 processing nodes
  - 100 Hz total (physics plus backgrounds)
    - 30 Hz of Hadronic Physics
      - Fully reconstructed
    - 70 Hz of backgrounds, calibration physics
      - Not necessarily fully reconstructed
- Physics Analysis
  - DST Creation
    - 2 users at 10^9 events in 10^6 secs (1 month)
  - DST Analysis
    - 20 users at 10^8 events in 10^6 secs
  - Interactive Analysis
    - 100 users at 100events/secs

Functionality Summary

- Basic design/functionality ok
- No performance or scaling problems with conditions, ambient and configuration databases
- Security and data protection APIs added
  - Internal to a federation
  - Access to different federations
- Startup problems being resolved
  - Scaling problems with event store
    - Online Prompt Reconstruction
    - Physics Analysis
  - Data Distribution
    - Internal within SLAC
    - External to/from remote Institutions

SLAC Hardware Configuration



Production Federations

- Developer Test
  - Dedicated server with 500GB disk & two lockserver machines
    - ← Saturation of transaction table with a single lock server
  - Test federations typically correspond to *BABAR* software releases
  - 5 federation ids assigned per developer
  - Space at a premium – separate journal file area
- Shared Test
  - Developer communities
    - ← (*e.g.* reconstruction)
  - Share hardware with developer test federations
    - ← Space becoming a problem – dedicated servers being setup
- Production Releases
  - Used during the software release build process.
    - ← One per release and platform architecture
  - Share hardware with developer test federations

Production Federations (2)

- Online (IR2)
  - Used for online calibrations, slow controls information, configurations
  - Servers physically located in experiment hall
- Online Prompt Reconstruction (OPR)
  - Pseudo real-time reconstruction of raw data
    - Designed to share IR2 federation as 2nd autonomous partition
    - Intermittent DAQ run startup interference caused split
      - Still planned to recombine
  - Input from files on spool disk
    - Decoupling to prevent possible deadtime
    - These files also written to tape
  - 100-200 processing nodes
    - Design is 200 with 100Hz input event rate
  - Output to several database servers with 3TB of disk
    - Automatic migration to hierarchical mass store tape
- Reprocessing
  - Clone of OPR for bulk reprocessing with improved algorithms *etc*.
    - Reprocessing from raw data tapes
    - Being configured now – first reprocessing scheduled for March 2000

Production Federations (3)

- Physics Analysis
  - Main physics analysis activities
  - Decoupled from OPR to prevent interference
- Simulation Production
  - Bulk production of simulated data
  - Small farm of ~30 machines
    - Augmented by farm at LLNL writing to same federation
    - Other production site databases imported to SLAC
- Simulation Analysis
  - Shares same servers as physics analysis federation
  - Separate federations to allow more database ids
    - Not possible to access physics and simulation data simultaneously
- Testbed federation
  - Dedicated servers with up to 240 clients
  - Performance scaling as function of number of servers, filesystems per server, cpus per server, and other configuration parameters

Integration with Mass Store

- Data Servers form primary interface
- Different *regions* on disk
  - *Staged*: Databases managed by staging/migration/purging service
  - *Resident*: Databases are never staged or purged
  - *Dynamic*: Neither staged, migrated or purged
    - Metadata such as federation catalog, management databases
    - Frequently modified so would be written to tape frequently
    - Only single slot in namespace but multiple space on tape
    - Explicit backups taken during scheduled outages
  - *Test*: Not managed.
    - Test new applications and database configurations
- Analysis federation staging split into two servers
  - *User*: Explicit staging requests based on input event collections
  - *Kept*: Centrally managed access to particular physics runs

Movement of data between Federations

- Several federations form coupled sets
  - Physics
    - Online, OPR, Analysis
  - Simulation
    - Production, Analysis
- Data Distribution strategy to move databases between federations
  - Allocation of id ranges avoids clashes between source & destination
  - Use of HPSS namespace to avoid physical copying of databases
    - Once a database has been migrated from source, the catalog of the destination is updated and the staging procedures will read the database on demand
- Transfer causes some interference
  - Still working to understand and minimize
  - Two scheduled outages per week (10% downtime)
    - Other administrative activities
      - Backups, schema updates, configuration updates
  - ~2 day latency from OPR to physics analysis
    - Have demonstrated <6 hours in tests

Physicist Access to Data

- Access via event collections
  - Mapping from event collection to databases
    - ← Data from any event spread across 8 databases
      - Improved performance to frequently accessed information
      - Reduction in disk space
  - Scanning of collections became bottleneck
    - ← Needed for explicit staging of databases from tape
  - Mapping known by OPR but not saved
  - Decided to use Oracle database for staging requests
    - ← Single scan of collection to databases mapping
    - ← Also used for production bookkeeping
    - ← Data distribution bookkeeping
- On-demand staging feasible using Objy 5.2
  - Has been demonstrated
  - Prefer explicit staging for production until access better understood

Production Schema Management

- Crucial that new software releases compatible with schema in production federations
  - New software release is a true superset
- *Reference* schema used to preload release build federation
- If release build is successful, the output schema forms new reference
  - Following some QA tests
- Offline and online builds can overlap in principle
  - Token passing scheme ensures sequential schema updates to reference
- Production federations updated to reference during scheduled outages
- Explicit schema evolution scheme described elsewhere

Support Personnel

- Two database administrators
  - Daily database operations
  - Develop management scripts and procedures
  - Data distribution between SLAC production federations
  - First level of user support
- Two data distribution support people
  - Data distribution to/from regional centers and external sites
- Two HPSS (mass store) support people
  - Also responsible for AMS back-end software
    - Staging/migration/purging
    - Extensions (security, timeout deferral, etc.)
- Five database developers provide second tier support
  - Augmented by physicists, visitors

Database Statistics (Jan 2000)

- Data
  - ~33TB accumulated data
  - ~14000 databases
  - ~28000 collections
- 513 persistent classes
- Servers
  - 12 primary
    - Database servers
  - 15 secondary
    - Lock servers, catalog & journal servers
- Disk Space
  - ~10TB
    - Total for data, split into staged, resident, *etc*.

Database Statistics (2)

- Sites
  - >30 sites using Objectivity
    - USA, UK, France, Italy, Germany, Russia
- Users
  - ~655 licensees
    - People who have signed the license agreement
  - ~430 users
    - People who have created a test federation
  - ~90 simultaneous users at SLAC
    - Monitoring distributed oolockmon statistics
  - ~60 developers
    - Have created or modified a persistent class
    - A wide range of expertise
      - 10-15 experts

Ongoing and Future Operational Activities

- Improved performance
  - Both hardware and software improvements
  - Reduced payload per event
  - Design goals almost met
- Improved automation
  - Less burden on the support staff
- Reduced downtime and latency
  - Outage level <10%
  - Latency <6 hours
- Large file handling issues
  - Problems handling 10GB database files for external distribution
- Better cleanup after problems
- Remerge online and OPR federations

Conclusions

- Basic design and technology ok
  - No killer problems
- Initial performance problems being overcome
  - Ongoing process
  - Design goals almost met
- Still learning how to manage a large system
  - Multiple federations
  - Multiple servers
  - Multiple sites
  - Large user community
  - Large developer community
- Still more to automate
  - Many manual procedures
- More features on the way
  - Multi-dimensional indexing
  - Parallel iteration