



BABAR™

Database
Operations

Operational Aspects of Dealing with the Large BaBar Data Set

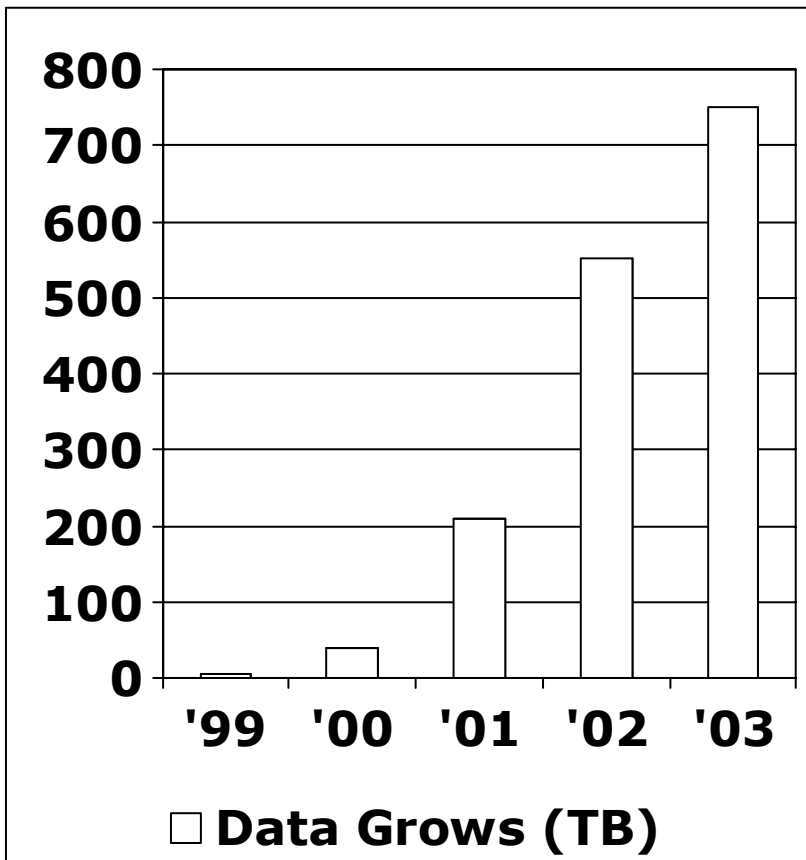
Tofigh Azemoon, Adil Hasan, Wilko Kroeger, **Artem Trunov**
For the BaBar Computing Group

Stanford
Linear
Accelerator
Center

Statistics

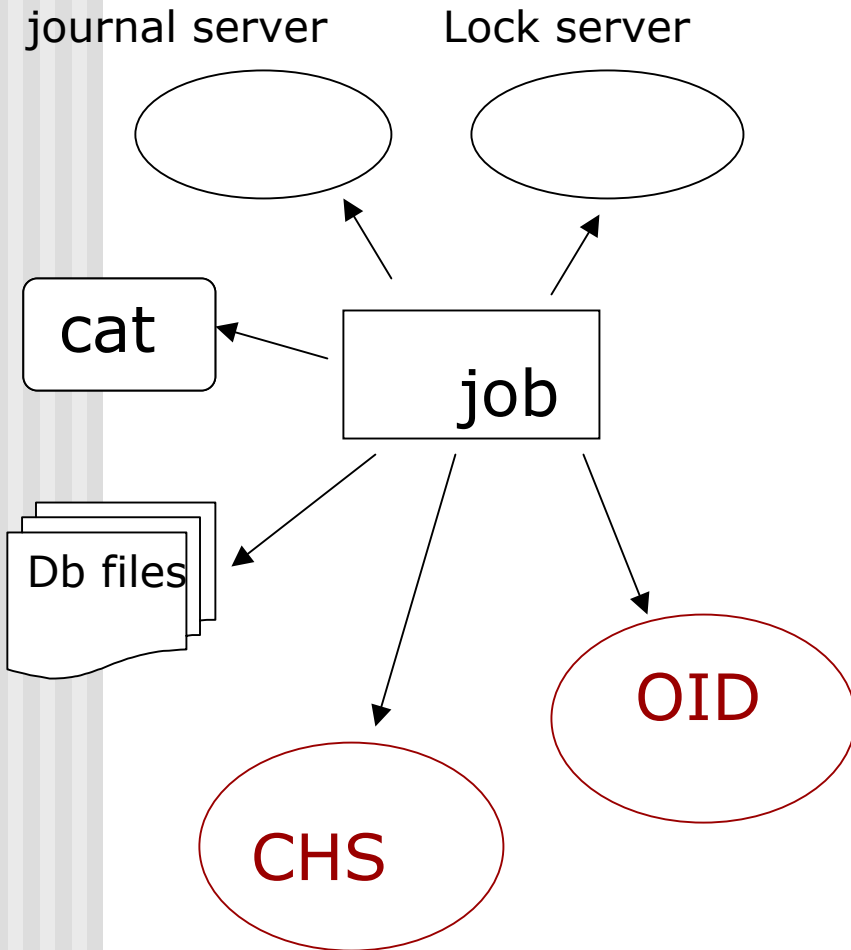
- Total size 750 TB
- 576000 database files
- Over 100 Objectivity/DB federations
- 88 TB of disk space - 50 servers
- Over 50 other servers
 - Lock servers, journal servers
- 60+ million collections

Data grows



- Slow down in grows - moving off exponential curve...
- ...But still well ahead of Moore's law
- Data grows rate ~ 3 TB/wk
- No more raw, rec in production
- Produce more collections per run (~ 120)
- Expect to hit PetaByte range next year.

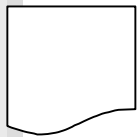
Increasing Complexity



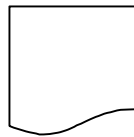
- Pure Objectivity is not enough for good performance and scalability
- Clustering Hint Server – procreates database files, dispatching containers to jobs
- Object ID server – caching object IDs for conditions database
- Used in production environment – need to manage and support 24x7

Increasing Complexity

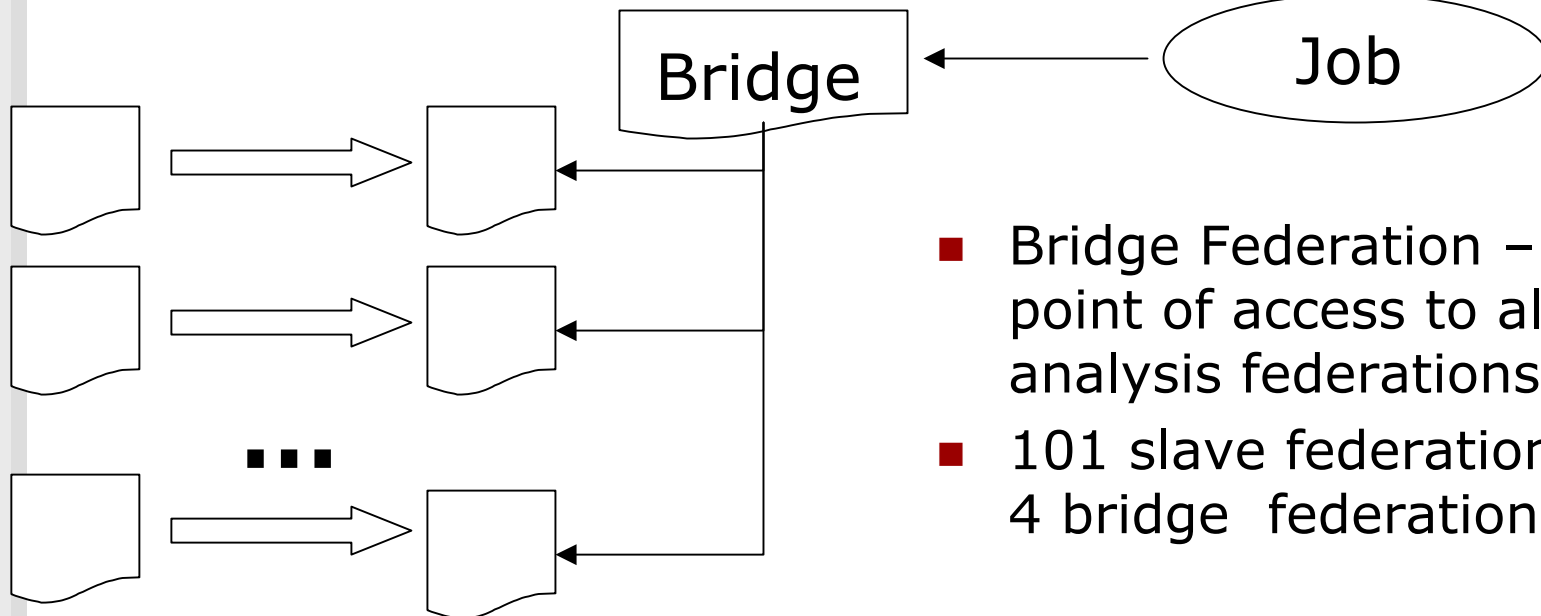
Production



Analysis



- From one production and one analysis federations to multiple federations



- Bridge Federation – single point of access to all analysis federations
- 101 slave federations under 4 bridge federations

Operational Problems and Goals:

Minimize downtime, maximize performance

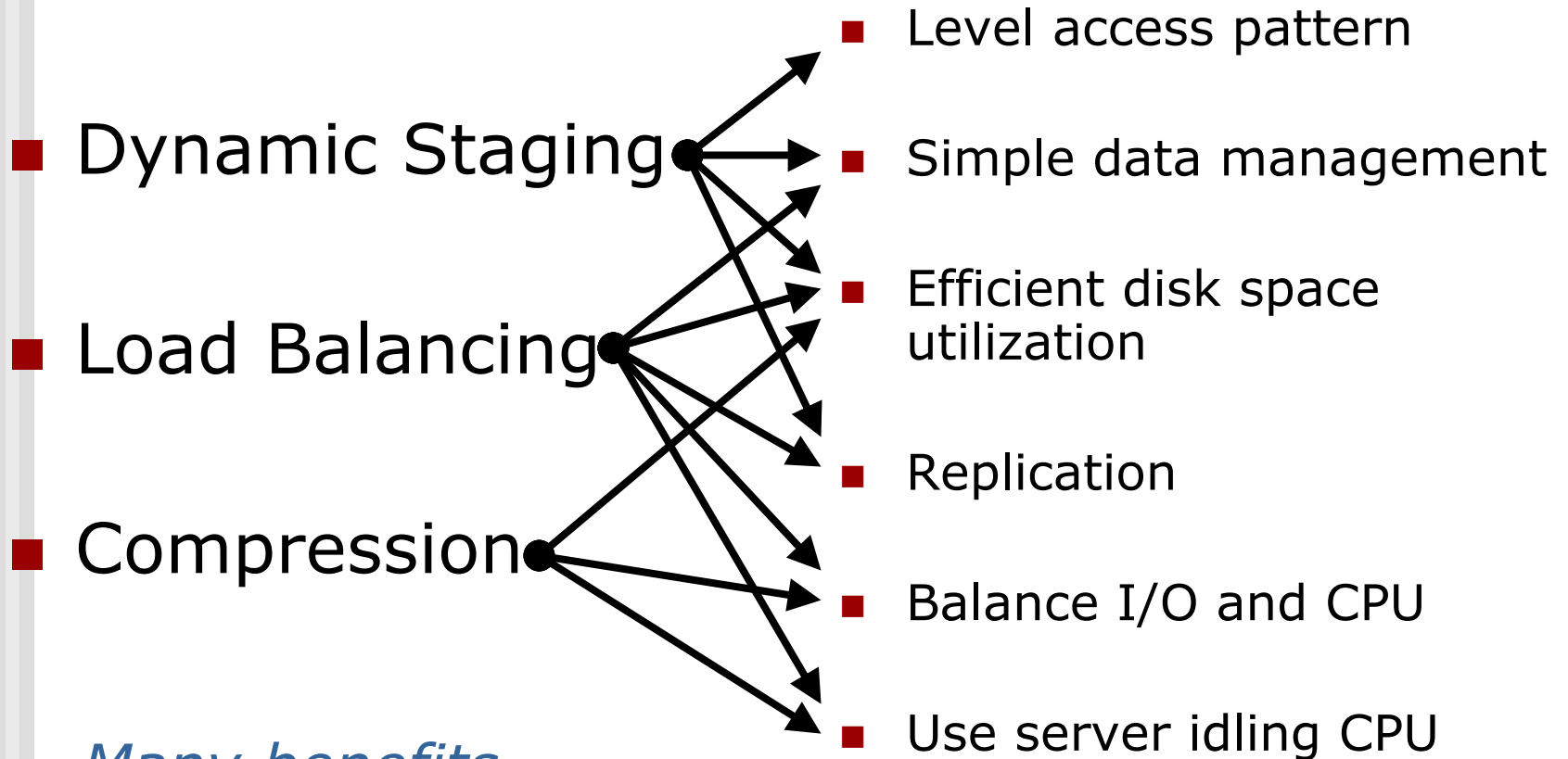
- Goal – 96% availability time
- Most admin task requires outages of a federation – blocking access for all users.
- 5 hours per week scheduled maintenance – new data and conditions loading, schema upgrade, backups etc...
- Most maintenance jobs are transparent to batch user – inhibit system prevents from starting new transaction when a federation is locked.
- Admin task is performed on one federation at a time – only small fraction of data is unavailable.
- Now more downtime is accounted for “external” reasons – hardware failures and maintenance.

Performance issues

- Data must be distributed over many hosts, file systems
- Data must not be clustered
- Data must be available automatically in timely matter (i.e. without user-admin interaction)

- AMS is a bottleneck
 - Use multiple AMS on each host
- Lock server saturation
 - Read-only databases

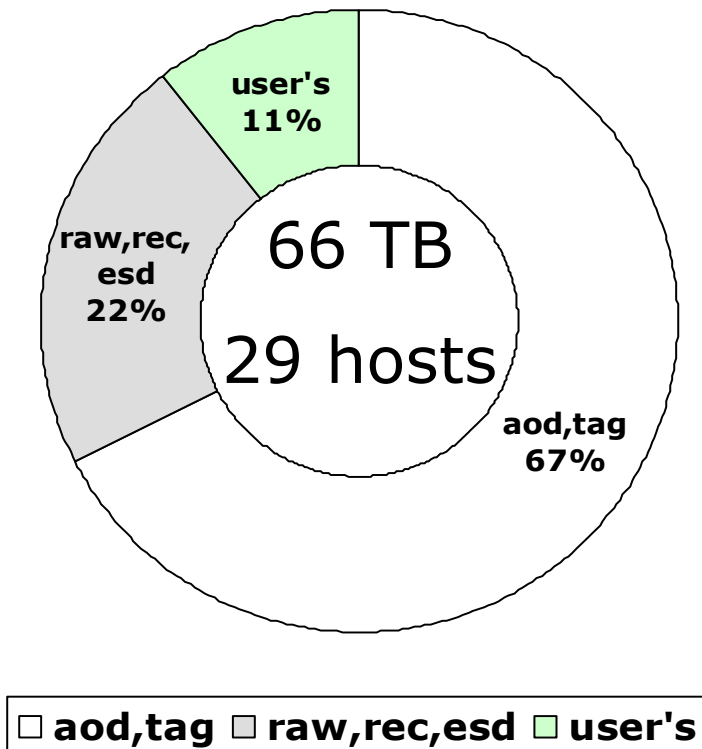
Maximize performance



*Many benefits
no tradeoffs*

Maximize performance – load balancing

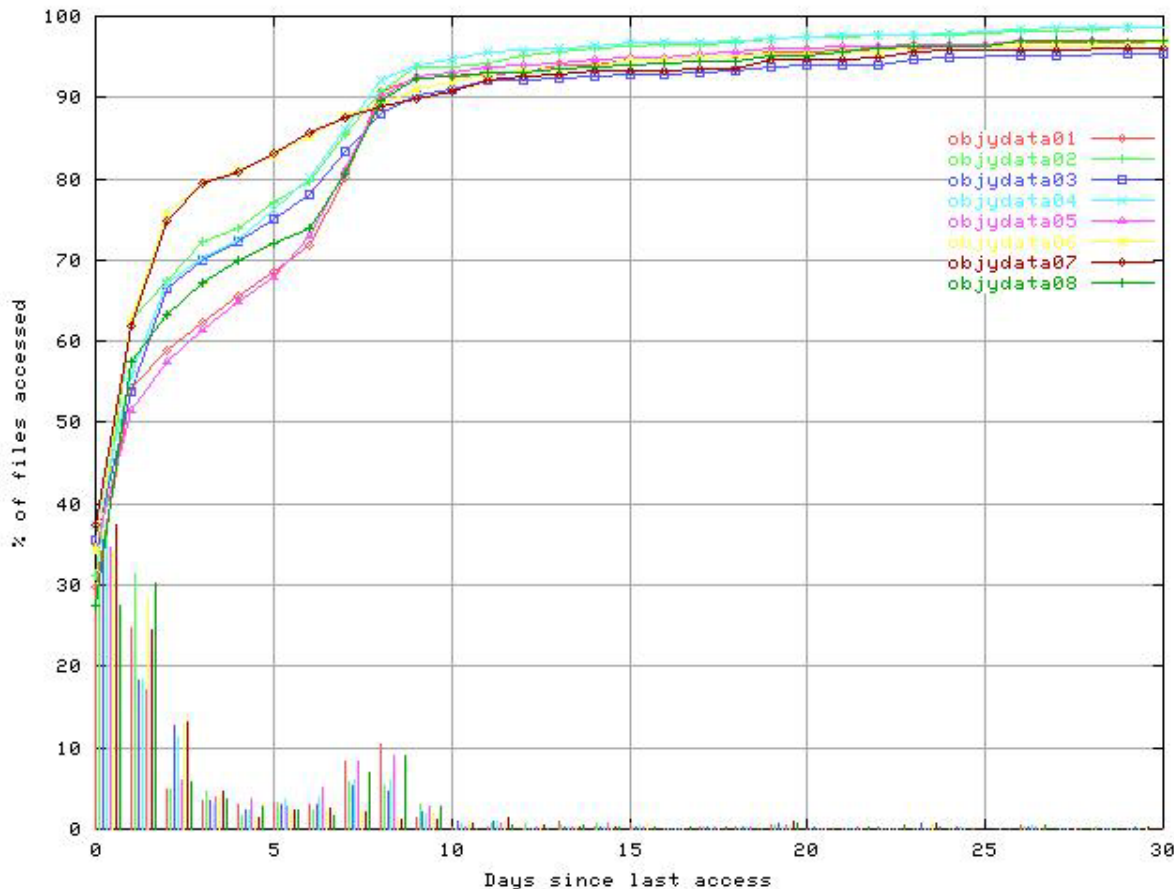
Analysis disk space



- 3 server pools - data distributed evenly within the pool
- Load balancing allows dynamically stage in files on the least loaded host in the pool.
- Allows to redirect client's requests away from faulty hosts – transparent to user's jobs

In deployment stage

Maximize performance – dynamic staging



Most files accessed in last 10 days

Other files can be purged to recover disk space

Dynamic staging and purging helps efficiently manage space

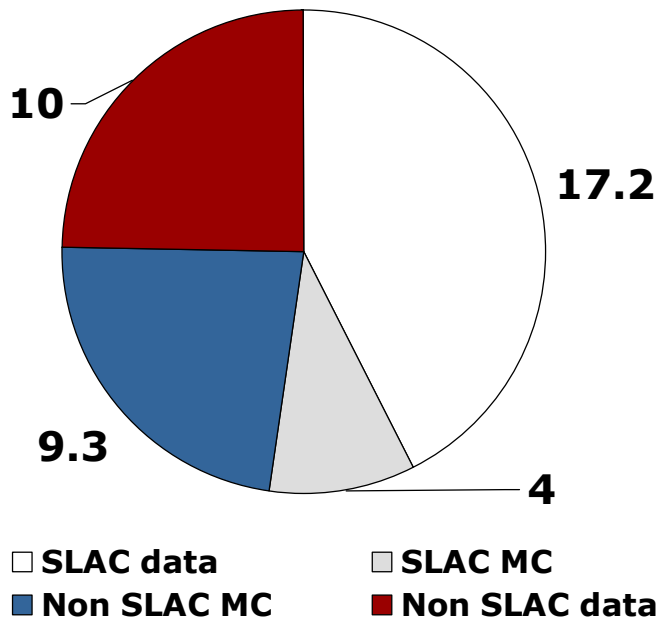
Fraction of files accessed vs. time /picture – courtesy of Bill Weeks/

Maximize performance - compression

- Access to data is I/O bound – when disk is 100% busy, server CPU is far from saturation
- Can use idling CPU for compression
- Achieving 2:1 compression ratio on “micro”
- Decompression on client side - Objy now supports it
- Can do server side decompression as well
- Compression done selectively, based on usage pattern.

In deployment stage

Data Transfer, Imports, Exports



Production since Nov'02 (TB)

- Non SLAC production ratio increased from 0 to 42%
- 25 institutions contribute to simulation production, 1 site (INFN-Padova) runs Event Reconstruction
- Export of full Objy dataset to IN2P3
- High performance copy programs – bbcp, bbftp
- 4 hosts dedicated for import/export operations

Import of data at SLAC

- Data transfer starts with *production cycle* – full databases are transferred immediately
- Production site doesn't keep a copy of the data, deletes it after transfer is complete
 - Allows to use smaller disk space at small institutions
- Data is imported to special import federation as soon as it arrives, then backed up.
- All import operation are automatic; status of imports is logged in a bookkeeping database.
- Error handling – not only MD5, but also database integrity check.
 - Automatic error handling of some errors
- Same protocol and tools for all imports – simulation and reconstruction data, but different transfer agents
- Method of “active client” – all transfers are initiated by remote sites, no handshakes.

Automation of Admin Tasks

- Real time monitoring of servers, including Objy specific monitoring – lock server load, AMS load.
- Recovery of some failures – restart services.
- Lock cleanup
- Routine maintenance – schema upgrade, back up, conditions transfers
- Managing multiple (over 200) federation - template driven: creation of new federations, authorization etc...
- Goal to integrate all admin tools – use single configuration file/module.

Metadata

- Objectivity is slow for catalog and metadata operations
 - Expected to improve in Objectivity V8.0 release
- Use of relational database for metadata
- Collection \Leftrightarrow database mapping
 - Built in new redesigned Event Store
- Storing file stats
- Bookkeeping of import/export operations
- Bookkeeping of conditions transfers.

Summary

- We can keep up with large data volume
- Objectivity brings complexity in data management, but the benefits are worth it.
- 30,000+ lines of code in database management tools (mostly Perl)
- No use of GRID yet, but working on it

Acknowledgments

- DB Operations team: *Tofigh Azemooon, Adil Hasan, Wilko Kroeger, Artem Trunov*
- *Andy Salnikov* – online DB operations
- *Yemi Adesanya* – real-time monitoring
- *Andy Hanushevsky* – mass storage, compression, load balancing development
- *Bill Weeks* – mass storage support

- People who contributed in the past: *Jean-Noel Albert, Cristina Bulfon, Lawrence Mount, Hammad Saleem*

More info

- BaBar Database Group Web page
 - www.slac.stanford.edu/bdb
- Main Bdb talk on CHEP'03
 - Jacek Becla "On the Verge of One PetaByte – the Story Behind the BaBar Database System"
- Bbcp – high performance copy tool
 - www.slac.stanford.edu/~abh/bbcp