

# *BaBar Databases*

Jacek Becla

Stanford Linear Accelerator Center



# Outline

- ◆ Bdb responsibilities

*Why do we exist?*

- ◆ Activities

*What were we doing since last review?*

*Why were we doing it?*

- ◆ Future plans

*What did we fail to do?*

*What are we going to do about it?*

- ◆ Manpower

*Can we really do what we want?*

# *Bdb Responsibilities*

*Provide efficient way of  
managing BaBar's data*

## ◆ Development

- Customizing underlying database engine to fit BaBar's needs
  - Based on ODBMS: Objectivity/DB
  - ~500K lines of code on top

## ◆ Administration

- Running system in production
- User support
- Data distribution

# Outline

- ◆ Bdb responsibilities

*Why do we exist?*

- ◆ **Activities**

*What were we doing since last review?*

*Why were we doing it?*

- ◆ Future plans

*What did we fail to do?*

*What are we going to do about it?*

- ◆ Manpower

*Can we really do what we want?*

# *Bdb Activities (Sep 2000 -> now)*

- ◆ Extending address space
- ◆ Tuning and scaling
- ◆ Reworking inefficient parts
- ◆ Improving operations and data distribution tools
- ◆ ...+ running the system and maintaining the code

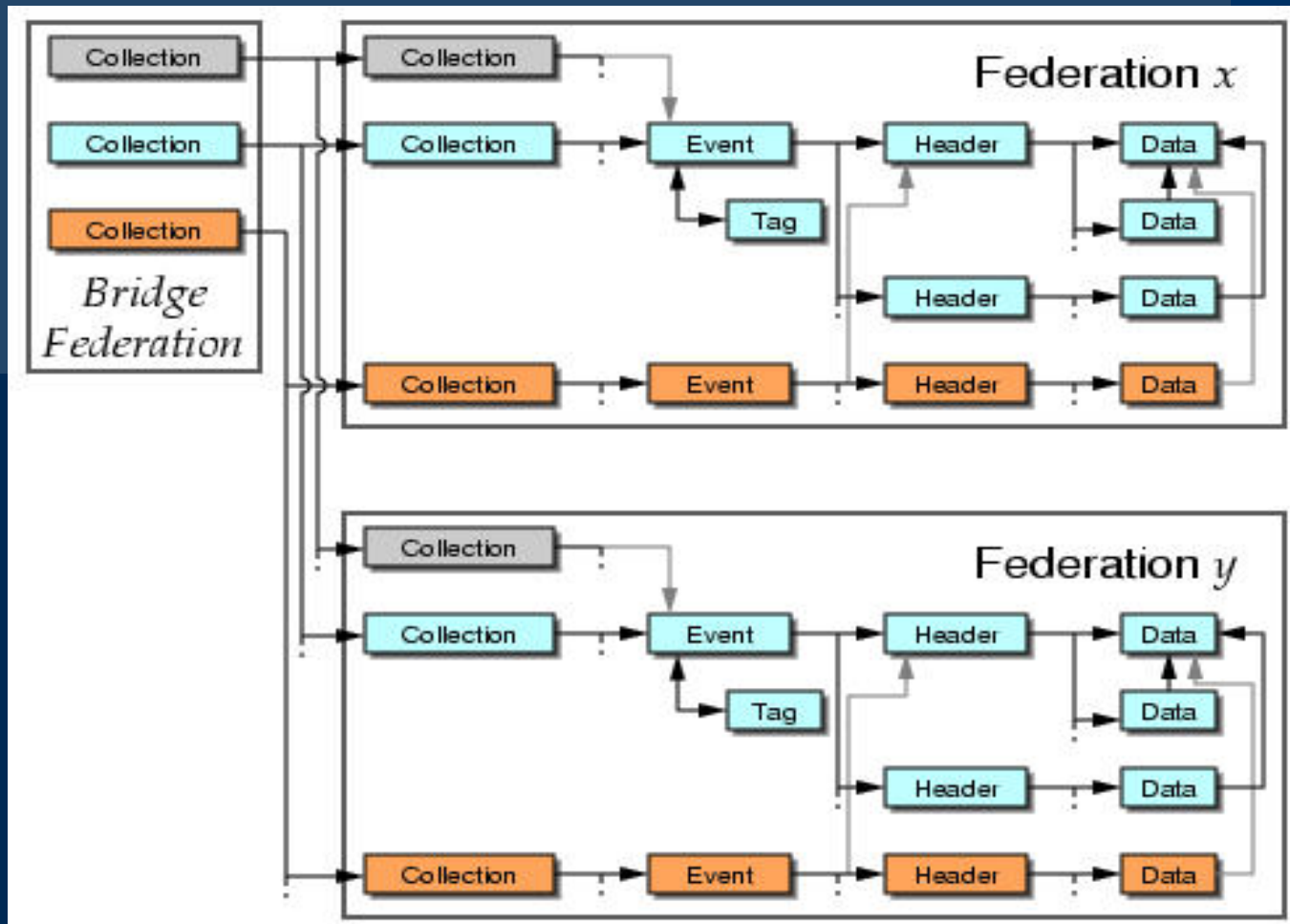
# *Extending Address Space*

- ◆ Max # files per federation: 64 k
  - not sufficient
  - large files difficult to manage and distribute
- ◆ Bridge technology
  - Allows access to multiple federations
  - A lot of work in Bdb code
    - to make it backwards compatible and transparent
  - + some in Objy

# *Bridge Technology*

- ◆ *Bridge collections* introduced to bridge across federations
  - Non-trivial
    - transparent transaction mgmt, “context” switching, iterators, inhibit system, many others
- ◆ Finally have sufficient address space  
(no need for major redesign in this area)

# Bridge Collections



# Bridge Technology - Status

- ✓ **Domain divestiture** (each domain can live in separate fd, don't have to maintain too many copies of Conditions)
- ✓ **Phase I & II** (reading & writing via bridge fd)
  - ✓ done now but delivered long after schedule
  - writing still not demonstrated in production
- ◆ **Phase III** (input fd != output fd, for data export)
  - due June 30, '02, needed for data distribution
- ◆ **Phase IV** (multiple input fds, for event mixing)
  - in discussions
- ◆ **General perception: late, late**
  - Clearly, effort underestimated

# *Tuning & Scaling (OPR)*

Req. #3.4.1: *“OPR must have the capacity to be able to process the data coming in from the detector with a max delay of 24 hours”*

- ◆ Move from 4 streams -> 21 streams
  - Streams cannot share files
    - ! Lock server traffic x5
    - ! Load on data servers x5
  - Non trivial, a lot of extra work, tuning required

# *Tuning & Scaling (OPR)*

- ◆ Centralized metadata operations (two CORBA servers)
- ◆ Solved random write problem
  - I/O more efficient, e.g. file migration speed 25 MB/sec (was 1.5)
- ◆ Code tuned, reduced lock traffic
- ◆ Proposed one-federation-per-dataserver configuration

# *OPR – Current Status*

- ◆ Up to 2x175 nodes in production
- ✓ OPR keeps up with data
- ✓ Expected to keep up in the future
  - + use servers much more efficiently
- Large scale tests done by Bdb group in the past
  - Should be in OPR hands from now on

# Compression

- ◆ Compression done by configurable daemon
- ◆ Decompression done by server (AMS)
- ◆ Still questionable if we want to compress
  - Space vs. CPU cost
    - Example compression factors:  
~4:1 (evshdr), ~5.6:1 (evt), ~2.5:1 (aod), ~1.8:1 (esd), ~1.3:1 (new mini)
    - Needed x2 CPU to uncompress (server side)
    - Practically, compress ~10-20% of data
    - Automatic load balancing will help
  - 👉 **Store data more efficiently rather than compress**
- ◆ Scheduled: 1<sup>st</sup> Q'02
  - First compressed files in production ~next week
- ◆ Considering compression on client side

# Optimizing Event Size

- ◆ Started to look Jan 2002
  - Have tools to determine size & overhead breakdown
  - Investigating deficiencies now
    - Found a few already
  - Will start modifications soon
- ◆ Expected to *significantly* reduce size by end of 2002

# *Load Balancing*

- ◆ Will improve performance
  - Distribute load
  - Make it dynamically scalable
  - Reduce influence on disk failures
    - Requires additional client-side coding
- ◆ Coding in progress
- ◆ Due 2<sup>nd</sup>Q'02

# *Reworking Condition DB*

- ◆ Will address currently known problems
  - Improves scalability
  - Lightweight and flexible
  - More efficient data distribution
- ◆ Due mid-June, '02
  - Challenging schedule

# Operations (1)

- ◆ Improved sweep tools
  - For import/export and internal data transfers
- ◆ Staging implemented, in production
  - Micro (analboot2): automatic
  - Mini: automatic and on-demand
    - Calibration process demonstrated on 2002 data
  - Staging at SLAC being reviewed
    - Proposal is being drawn up to address needs of new mini

# Operations (2)

- ◆ Tools to manage bridge fds
  - Attaching collections
  - Fixing teething problems
    - full paths, wrong # events
- ◆ Fds not closed often enough
  - Db id range for future rep/analysis
  - Difficult to spread lock server load

# Analysis

## ◆ Done

- Introduced bridge technology
- Reduced outage time
- Tuning and scaling
  - Added capacity
    - more data servers, disks and lock servers
  - Reduced lock collisions
  - Introduced read-only dbs
- Automatic & on-demand staging

## ◆ A lot more being worked on

- Expecting much higher load soon
- Objy analysis in tier C ~end of this year

# *Communication*

- ◆ with users

- Webpage
- Hypernews

- ◆ with others

- Stanford DB group
- Monitoring LHC effort, RTAG, ...
- Press release

# Outline

- ◆ Bdb responsibilities

*Why do we exist?*

- ◆ Activities

*What were we doing since last review?*

*Why were we doing it?*

- ◆ Future plans

*What did we fail to do?*

*What are we going to do about it?*

- ◆ Manpower

*Can we really do what we want?*

# Future Plans

	FTE-months still needed (development)	deadline
◆ To be finished		
– Bridge fd phase 3	3	June 30'02
– Load balancing	3+	Q2'02
– Condition DB redesign	4	June 15'02
– Optimizing event size	~9	~Q4'02
– Tune bridge fds	2	~Q2'02
◆ New projects		
– Reworking collection metadata	~11	~Q4'02 (?)
– Reducing disk & network I/O	~12	~Q4'02 or Q1'03
– Bridge fd phase IV?	2	Q3'02
– High performance AMS	4	Q3'02
– Contingency plan?	?	?
◆ Plus		
– Switch PR to new system		
– Assist REP in Padova		
– Maintain software, run production, user support, data distribution, analysis in tier C		

# *Reworking Col Metadata*

- ◆ Automatic col->files mapping
- ◆ Attaching collections slow now
  - PR/REP ~3-4, SP ~7 hours per week
  - Could be a few minutes if we had time to improve
- ◆ New design in discussions
  - No deadlines/schedules yet

# *Reducing Disk & Network I/O*

- ◆ Asked objy to add ooStatisticsMgr
  - To understand disk & network traffic, cache hits & misses, many others...
  - Delivery expected 2<sup>nd</sup> (3<sup>rd</sup>?)Q'02
- ◆ Already aware of some issues
  - Expect visible reduction
    - On top of improvements from reduced event size

# Outline

- ◆ Bdb responsibilities

*Why do we exist?*

- ◆ Activities

*What were we doing since last review?*

*Why were we doing it?*

- ◆ Future plans

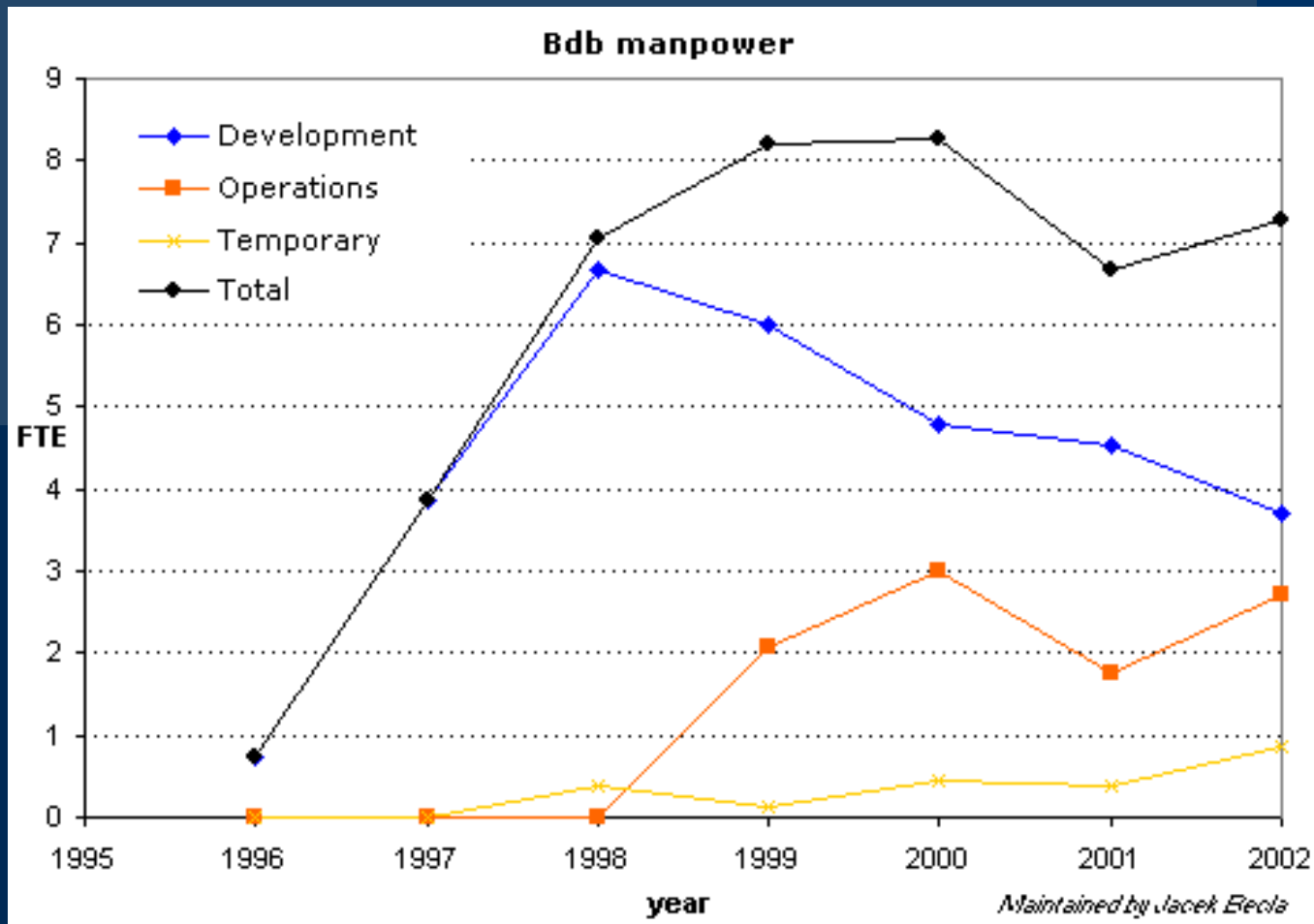
*What did we fail to do?*

*What are we going to do about it?*

- ◆ **Manpower**

*Can we really do what we want?*

# Manpower



Total = ~45 FTE-years

# *Manpower*

- ◆ Need 12+ FTE
- ◆ Now has ~7 FTE (including students)
- ◆ New position (development)  
@ SLAC opened now
  - realistically, a new person ~Jan'03

# *How are we managing?*

## ◆ Objectivity

- Excellent support, new features, improvements
  - Reduces the impact of insufficient manpower
- One data corruption bug found and fixed

## ◆ External sites

- Tier A sites offloading SLAC
  - IN2P3 happy with objy-based solution
- ~15 tier C sites run objy-based MC prod successfully
- Objy-analysis in tier C soon
  - Will physicists want to switch from Kanga?

# Summary

## ◆ Per system

- OPR – ok
- Analysis, serious effort to optimize (speed and event size) started recently
- SP – ok (mostly using tune-ups done for others)
- Data distribution covered by Adil

## ◆ Really tight on manpower

- Bdb group overworked
- Many features possible, have to focus on most important only

## ◆ ODBMS, Objectivity, bridge fds, event size ...

- Schizophrenic view of Objectivity
- Many recognize & appreciate the effort
- We think it is all doable