Distributed Offline Data Reconstruction in BaBar

Peter Elmer
(For the BaBar computing group)
Princeton University

CHEP2003 – March 24, 2003

BaBar PromptReco Development Team

- Caltech: Anders Ryd
- Colorado: Doug Johnson
- Ohio: Teela Pulliam
- Padova: Alberto Crescente, Alvise Dorigo, Fulvio Galeazzi, Mauro Morandin, Roberto Stroili, Gionni Tiozzo, Gabriele Vedovato
- Rome: Francesco Safai Tehrani
- SLAC/INFN: Antonio Ceseracciu, Martino Piemontese
- Wisconsin: Sridhara Dasu

The BaBar Experiment

- BaBar is an experiment built primarily to study B-physics at an aymmetric high luminosity electron positron collider PEP-II at the Stanford Linear Accelerator Center (SLAC)
- BaBar is an international collaboration involving 560 physicists from 76 institutions in 10 countries
- We have been taking data since May, 1999 and are currently in the middle of “Run3”, which will run through Jun2003
- To date we have taken ~110 inverse fb of data (~1.1 billion events), we expect to increase this to o(0.5) inverse ab by the end of 2006

BaBar Prompt Reconstruction (PR)

- The BaBar Prompt Reconstruction system is the primary means for bulk reconstruction of BaBar data in raw format
- The raw data is written from the L3 trigger into a flat file ("xtc file"), this file serves as input to the PR system. PR is *not* in the dead time path for datataking. The xtc file is migrated to mass storage from the online system and can be obtained by from there or copied over directly from the online buffer disk.
- The PR system is used both for processing of new data as it is taken by the experiment as well as reprocessing of older data.
- In addition to event reconstruction the system is also responsible for providing a means to perform automated "rolling" calibrations and monitoring. A dedicated control system was developed to orchestrate the PR system (see talk by A.Ceseracciu, "BaBar Data Reconstruction Control System")

Parallel processing model

- Given the large number of events per run (currently corresponding to a single xtc file) it is not practical to process the entire run on a single cpu, thus parallelizing the processing over many cpus is desirable.
- Our model for parallel processing in PR is a central server (the “Logging Manager”) which reads the xtc file and distributes the events to a set of client processes.
- The Logging Manager has the logic to insure that all events are processed once and only once. It also handles naturally the death of any given client process.
- Each of the client cpu processes writes its eventstore output to an Objectivity federation (see talk by J.Becla, “The story behind the BaBar database system”, ...)

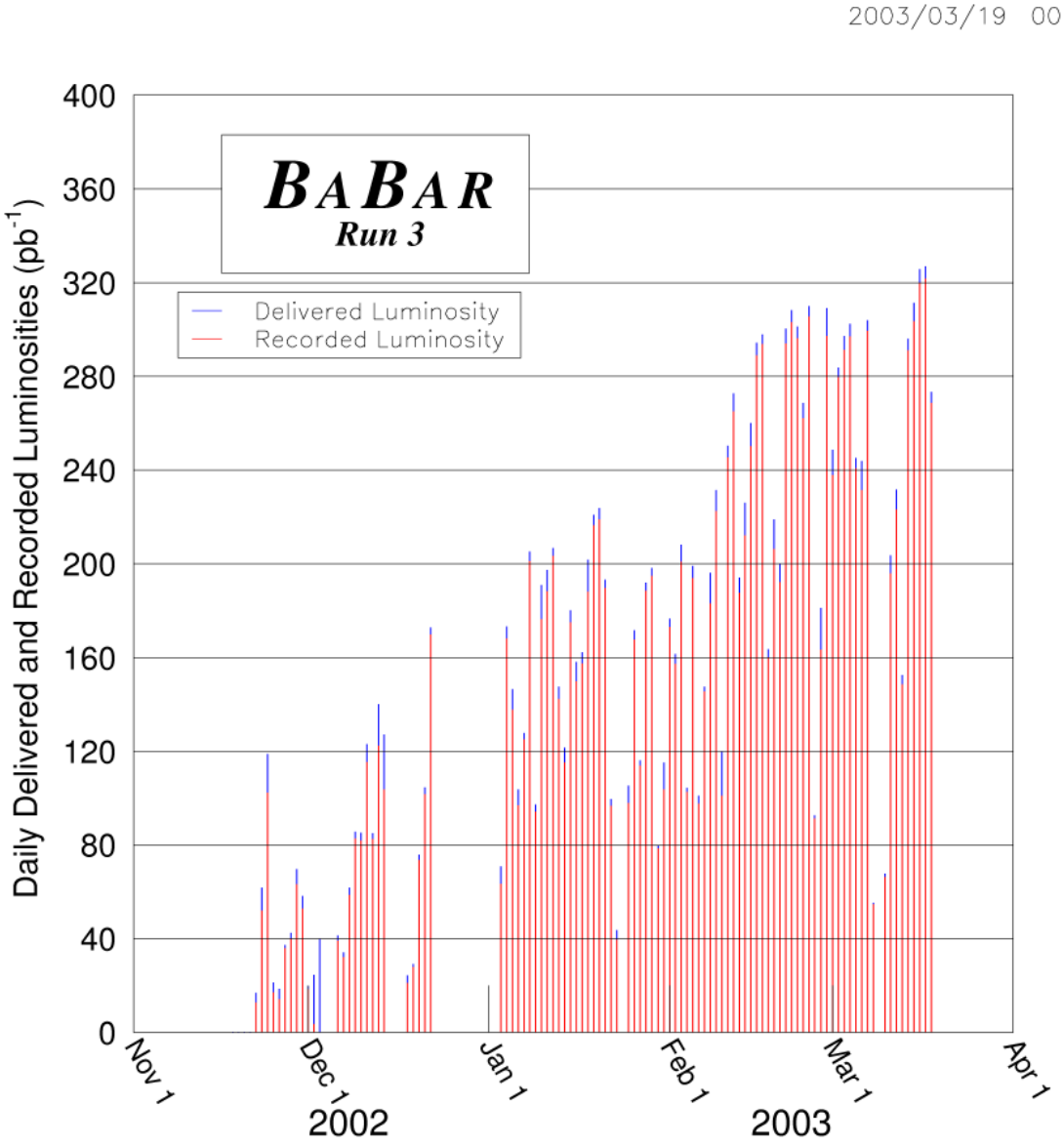
Reconstruction numbers – input

- Input event size (raw) ~ 30kB
- One single file per run is written by L3 trigger containing all events (physics, calibration, ...). The typical file size is of order 10GB, containing 300-350 kEvents. In the past we have written files containing up to 1M events each.
- We record more data than we fully process, all but approximately 35-40% of the events in the raw data xtc file are filtered by dedicated filters early in the reconstruction executable before full reconstruction is performed.

Reconstruction numbers – output

- On output we can write the following components to the eventstore: tag, micro, mini, reco, raw
- tag/micro has been primary analysis format
- At the end of 2001, we deprecated writing the "reco" data (100kB/event+) on output as its function is largely replaced by a redesigned mini (see talk by D.Brown, "The BaBar Mini").
- In summer, 2002 we also deprecated writing a copy of the raw data (50kB/event) into the Objectivity eventstore, recognizing that a full reprocessing from the raw in Objectivity was unlikely.
- Output event size ~ 20kB (tag/micro/mini), with physics selections: 4 physical streams/117 pointer collections

Processing new data

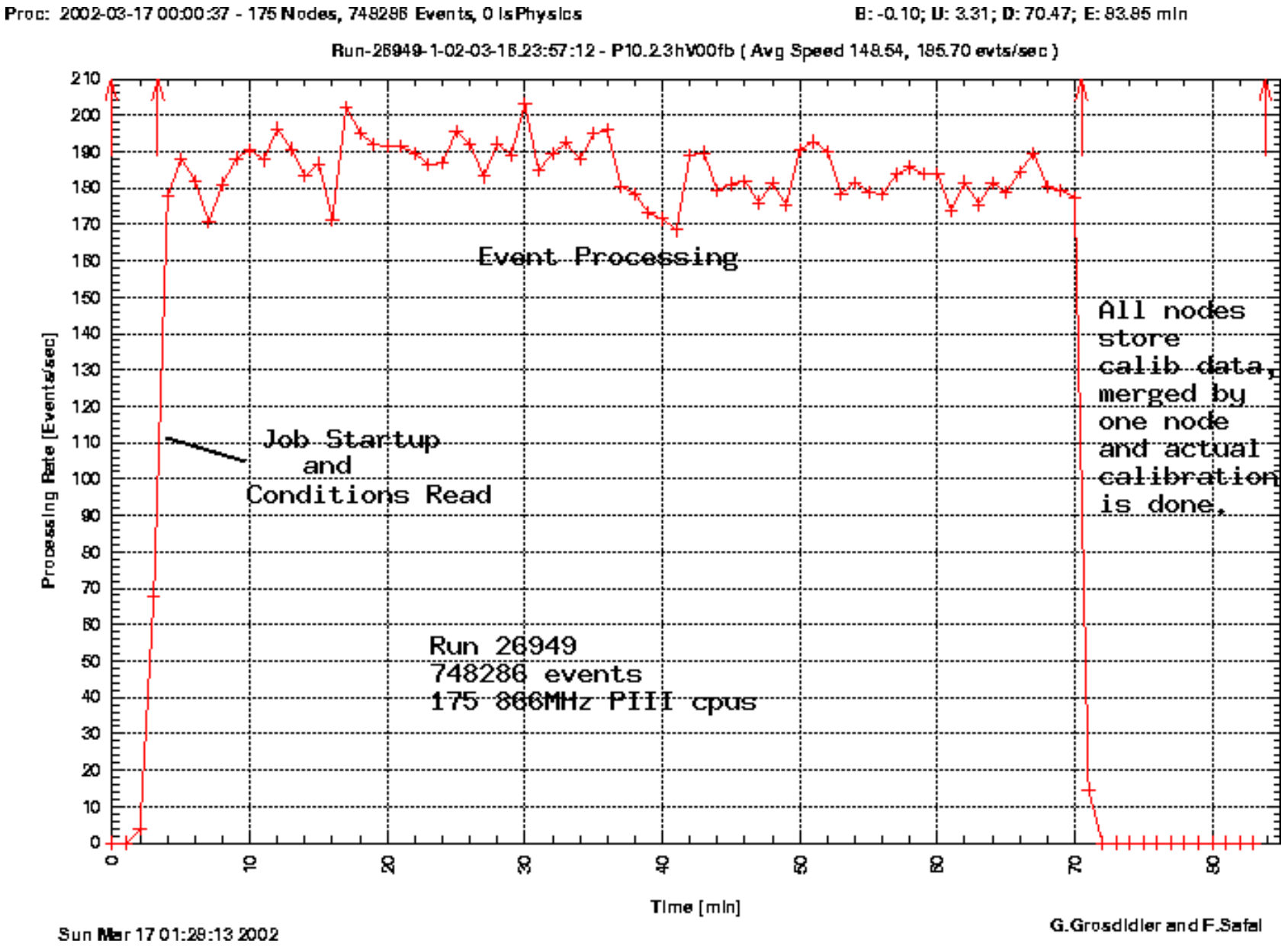


- BaBar/PEP-II now producing 300+ ipb/day
- The current system should scale easily to at least twice this in order to keep up with incoming data

Reconstruction/Calibration Model (old model)

- Through the end of the last run, in July2002, we performed the reconstruction in a *single* pass. Calibration information was collected during the processing of the events from the run itself. Calibrations were performed and written to the conditions database at the end of the run. The calibrations determined from run N are then used as input for run N+1.
- This model avoids overhead (CPU and I/O) from multiple passes through the data, but locks us into processing the runs in the order they were taken. Scaling the farms to accomplish this has had a number of scaling problems.
- In addition the calibrations used for each run are not the best possible for quantities that change quickly.

Typical processing rate (old model)



New Reconstruction/Calibration Model

- As of this past fall, we are now reconstructing data with a more classic model with *two* passes:
  - The first "prompt calibration" (PC) pass processes only a fraction (fixed rate) of events and writes out only the resulting conditions (i.e. does not write to eventstore)
  - The second "event reconstruction" (ER) pass actually processes the full set of events writing to the eventstore.
- Runs are processed in the order they were taken on the PC farm and then the run can be scheduled on any ER farm available. The conditions must be swept from the PC farm to the ER farm periodically, benefit from new conditions design.
- Run N is processed with calibrations determined from run N

New PR farms architecture

- New architecture: Prompt Calibration (PC) farm processes runs in sequential order followed by the scheduling of the run onto one of several (larger) Event Reconstruction (ER) farms.

Reprocessing

- The total throughput needed for *reprocessing* may actually exceed that needed for processing new data. Capacity needed is defined by availability of stable reconstruction executable and deadline by which data must be reprocessed (e.g. for analysis for conferences)
- Scaling for reprocessing can be accomplished by breaking the conditions timeline into separate intervals and creating a separate Prompt Reconstruction system, with one PC farm and N ER farms, for each time interval.

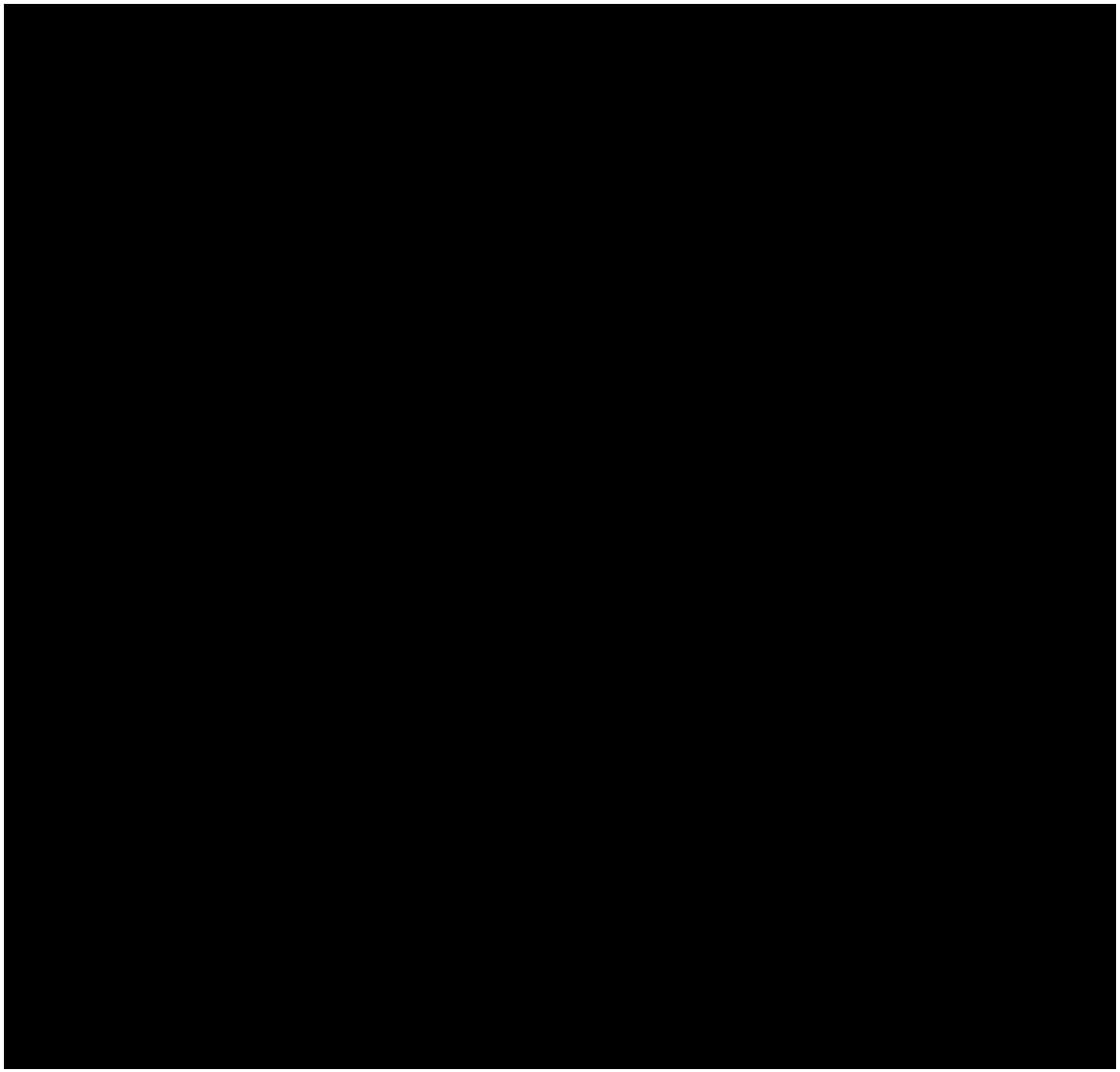
Current processing capacity

- 3 PC farms at SLAC ~32 1.4GHz cpus (1 for new data, 2 for reprocessing) each
- 5 ER farms at SLAC ~ 64 1.4GHz cpus each
- 4 ER farms in Padova ~ 60 1.26GHz cpus each
- Distributed system (1 PC farm at SLAC feeds 4 ER farms in Padova, see poster “A facility for large scale reprocessing of BaBar raw data”)
- Each PC farm can do 600ipb/day and each ER farm can do 150ipb/day. Both types of farms still have significant deadtime between runs, so there is room for improvement. For new data, we expect that the PC pass will be done in <8hours, ER pass within 24 hours.

PC farms

- Shown is the event rate over a 24 hour period.
- Deadtime is due to the calibrations calculations
- All nodes write temporary calibrations info, which is read back into a single node for calibrations calculations

ER farms



- Shown is the event rate over a 24 hour period.
- Deadtime is due to overhead from Objectivity eventstore (Clustering Hint Server cleanup, collection precreation) and other administrative work
- Working on reducing overhead, also planning to keep CPU's spinning by running background simulation batch jobs.

Outstanding issues

- Stability, stability, stability: many improvements have been made and many new features have been introduced, but the system still requires too much intervention by human beings.
- Monitoring: Constant struggle to make sure that appropriate monitoring is available to spot problems with reconstruction
- Corruption: We have had a number of problems with data corruption: some due to hardware problems, some from Objectivity.
- Data availability: Data written to dedicated production servers, migrated to mass storage and “swept” to analysis servers. Due to current limitations of the Bdb/Objectivity eventstore, data is not made available for up to 10 days after it is processed, however.

Summary

- The BaBar Prompt Reconstruction system is used for processing of new data and reprocessing of the data set. We have recently moved from a single pass architecture to two pass system.
- In order to exploit available resources, we have moved to a distributed system using farms in multiple sites (SLAC and Padova)
- We have made significant progress on building a stable, scalable system and now believe that we have an architecture better positioned to scale well through future luminosity upgrades.