The *BABAR* Prompt Reconstruction System or

Getting the Results out Fast: an evaluation of nine months experience operating a near real-time bulk data processing system

The BABAR Prompt Reconstruction Group:
T. Glanzman[1], *J. Bartelt*[1], *R.F. Cowan*[2], *S. Dasu*[3], *G. Grosdidier*[4], *F. di Lodovico*[5] and *F. Safai Tehrani*[6]

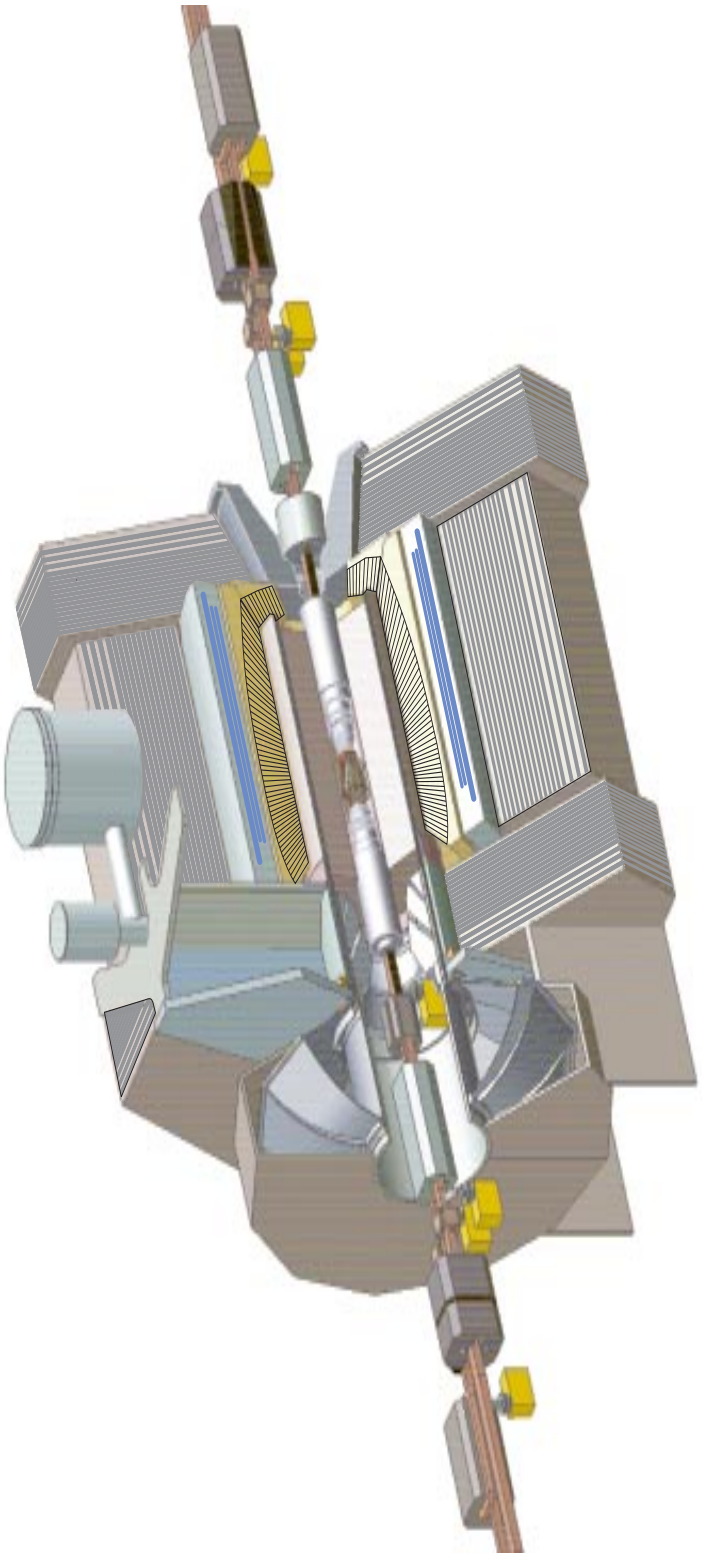
(for the *BABAR* Computing Group)

[1] Stanford Linear Accelerator Center, Menlo Park, CA, USA
[2] Laboratory for Nuclear Science, M.I.T., Cambridge, MA, USA
[3] University of Wisconsin, Madison, WI, USA
[4] LAL-IN2P3-CNRS, Université Paris-Sud, Orsay, France
[5] University of Edinburgh, Scotland, UK
[6] INFN, Sezione di RomaI, Rome, Italy

A true story (with a moral) about how a flood of data, buggy code, a recalcitrant database, a newly assembled and unreliable infrastructure, and inexperienced operators deluged an enthusiastic development team during the first months of *BABAR*'s lifetime.

Life Begins: 04:30 PDT 26 March 1999

The *BABAR* Detector



Key features:

- Intermediate “flat” input file (.xtc)
- Output to Objectivity database federation
- Quasi-realtime (latency, reliability, etc.)

```
04:38 PDT 26 May 1999 - first colliding beam
“revenue” Run 5354 begins

07:40 PDT 26 May 1999 - First Data processed and
available in Objectivity (18k events)
```

Event processing rate was 7.2 events/sec (maximum) across 20 Sun 333MHz Ultra5 machines.

Four developers were running 7x24 shifts and the entire system was unreliable....

A short diversion, the *mission statement*:

> Process 100% of all BABAR physics events within 2 hours of its acquisition, including filtering/tagging, reconstruction, constants generation, monitoring and logging into an Objectivity database. This is known as the online *Prompt Reconstruction* system

Key numbers:

Input rate = 100Hz
Raw event size = 32kB
Processing latency ≤ 2 hours

Birthing Pains: from 26 May to the *BABAR* collaboration meeting of 23 June 1999:

Problems:

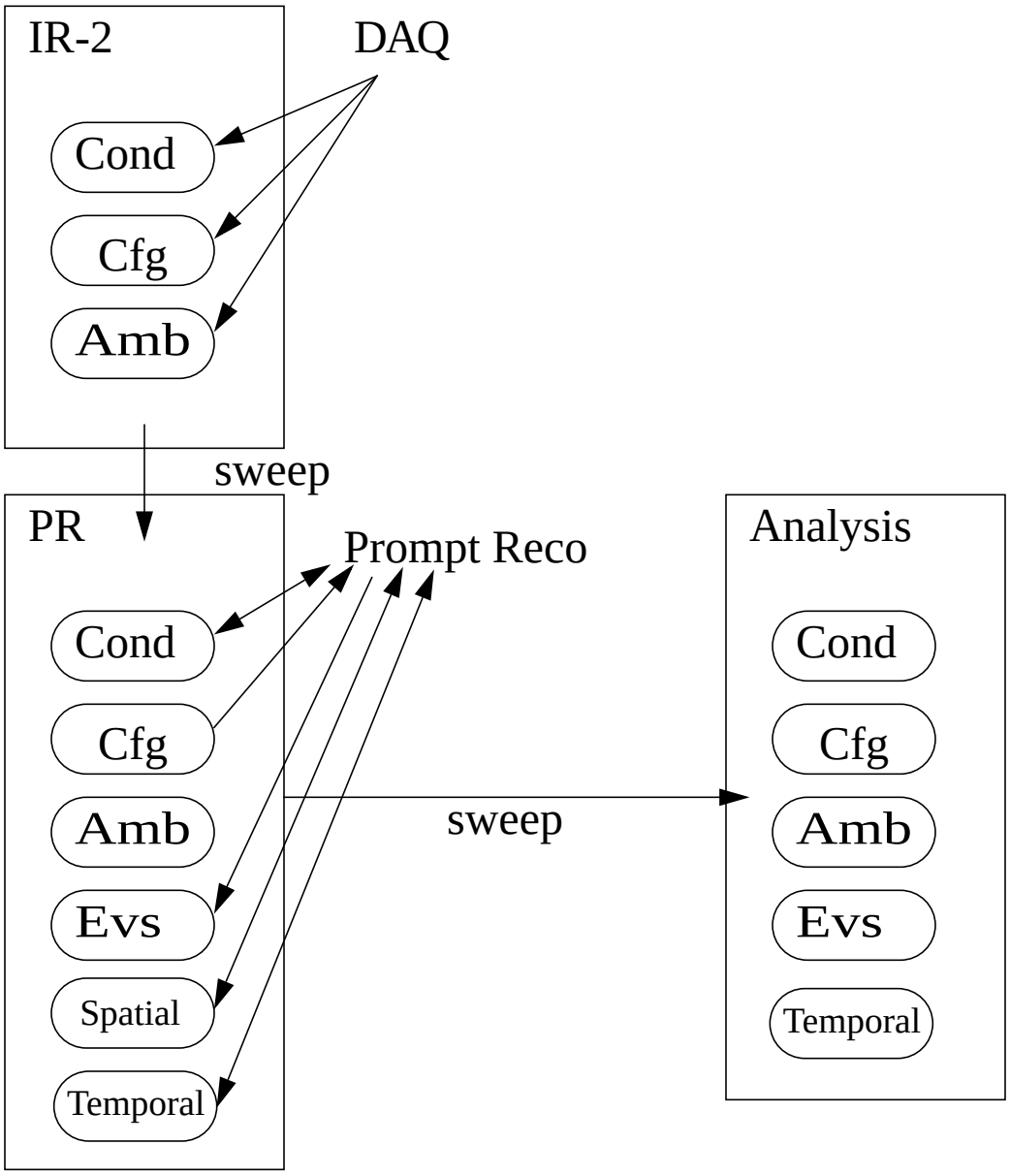
- kernel memory leak in Solaris 2.5 running in key server
- cpu problem in E450 machine (4-cpu)
- Veritas file system corruption/stack overflow
- Autoclient instability (see paper #115 by A.Telnov at this conference)
- unacceptable AFS dependences
- NFS loading
- unstable reconstruction application (9 releases in 4 weeks)
- Objectivity instability (deadlocks, dead locks, dreaded locks!)
- Objectivity non-scalability (≤20 machines)
- peak processing rate 6-7 Hz, overall average rate 1-3 Hz

Responses:

- twice weekly code review meetings - filter and regulate new releases
- set up operations group (10 persons to cover 24 x 7)
- disable rolling calibrations
- beat on Sun Microsystems for fixes
- ***split single federation into many*** (whole new infrastructure...and industry!)
- continue to focus on Objectivity performance, enabled massive logging

Split federations...

Multi-federation Model



Growing pains: BABAR Computing Review, 2-4 August 1999

By that time,

- Prompt Reconstruction was able to successfully run on 50 machines
- Another eight (8) code versions had been released
- Automated job sequencing, *troll*, commissioned (previous talk, #180 at this conference)

(excerpt from review committee...)

> For example, OPR and the offline system still have significant difficulties in some areas, in particular with data access and software performance. As a consequence, reconstruction is barely able to keep up with the data-logging rate and the system is unable to provide simultaneous data analysis by many users. Up to now there is no strategy in place to control access to these limited resources, which leads to long waiting times and hence, a lot of frustration.

Committee recommendations (in part):

- Separate development and operations teams
- Focus on improving Objectivity performance
- Change priorities from “keeping up” to “fixing problems”
- Pursue non-Objectivity data output option (*for the near-term*)
- Recombine Objectivity federations (enable rolling calibrations)
- Aim for new code release only every 3 months

And some immediate responses:

- Create 100-machine + servers Objectivity test facility
- Schedule DB downs -- 25% of week during two periods!
- Create distinct OPR operations group for running processing shifts (developers remained as on-call experts, but otherwise free to continue development)

Development ramps up while operations continue: Autumn 1999

Problems remained:

- training shifters in far-away places => low operational efficiency
- accelerating number of requests to perform full-scale tests
- need to ***reprocess*** large amounts of data
- cancellation of an eagerly anticipated accelerator down time in Aug-Sep (but unexpected down period (Nov-Dec) to fix vacuum leak)
- between 3 August and 31 December there were 13 new code releases
- key member of development team resigns from SLAC
- event processing rate in production remained low
-

But on the bright side:

- much progress with Objectivity event processing performance in dedicated test facility (see paper 110 by J.Becla at this conference)
- farm ramped up to routinely running with 100 machines
- major build up and rework of network infrastructure
- commissioning of *global farm daemon* core code (see paper #161 by G.Grosdidier at this conference)
- Prompt Reconstruction shifters given collaboration credit for their efforts
- parallel reprocessing team started -- along with duplicate hardware
- significant gains in reconstruction code efficiency
- HPSS outages much less than expected (only ~1 shift/month)

February 2000 - Gearing up for 100 Hz

Latest (final?) measures taken or contemplated:

- Expand farm capacity with additional CPUs (150 440MHz Sun T1)
- Expand Objectivity backend
  - 2->3 large “datamover” servers (with HPSS backend)
  - 2->4 RAIDs on datamovers
  - 2->3 small servers (lock, journal, catalog)
- Continue to tune and improve Database code (and to work around problems within Objectivity)
- Focus on “edge effects” of processing a run (i.e. start-up and shut-down), including lengthening data runs
- Commission job monitor, *imp*, to reduce MTTR
- Reduce (or eliminate) certain types of data from the event stream (e.g. backgrounds, events for efficiency studies, calibration events, etc.)
- Reduce scheduled DB down periods, but increase frequency of “sweeps”
- (Re)enable Rolling Calibrations
- Phase out the Sun Autoclient system

Performance Summary and Future

BABAR Prompt Reconstruction System has been commissioned and in 24 hour/day by 7 day/week production for nearly nine months.

For the period 26 May 1999 through 14 January 2000 (33 weeks):

- 179M events processed
- 250M events processed (includes reprocessing)
- 55 Hz steady-state processing rate (100 machines)
- 26 Hz average processing rate over select 11-day period
- Latency (not yet a running priority):
  - 3178 runs processed
  - First Data Run processed within 3 hours
  - Total runs processed within 2 hours: 7 (0.2%)
  - Total runs processed within 2 days: 858 (27%)
  - Total runs processed within 2 weeks: 2268 (71%)

Goal: 100Hz processing rate (averaged over days) by 15 March 2000 while not throwing away priority physics.

We are just now assembling the hardware and software to meet this challenge...and this is a big challenge!

Lessons Learned

- **Infrastructure selection** -- The choice of computers (Sun), operating system (Solaris), programming languages (C++, scripting), networking (100Mbps/1Gbps with Cisco equipment) was generally good. All components required tuning to operate in a *large* and *heavily used* system.

- **Retrospect (hindsight?) engineering** -- Expect that a complex system with many dependencies and built upon many assumptions will need flexibility to cope with the real world.

- **Development->Operations sooner rather than later** -- Gear up for operations well before it is actually needed or the development team will find themselves swamped. This includes lining up personnel and management for those people.

- **Just-in-time computer hardware purchasing** -- The longer you wait the cheaper and better the product. True for computing hardware, but the amount of time for ordering, site-preparation, installation, commissioning, and turning into a reliable production system must not be underestimated. The timescale is months, not weeks.

- **Quality of commercial software** -- Buying software from a reputable company is no guarantee of flexibility or quality. Often its “phase space” is sparsely tested which leaves critical debugging to the customer. This can happen with operating systems, file systems and object databases. Our experience with carefully selected freeware has been no worse than with expensive payware.