New distributed offline processing scheme at Belle

Ichiro Adachi
Nobuhiko Katayama
Frédéric Ronga
High Energy Accelerator Research Organization
KEK – Tsukuba – Japan

CHEP'04 — Interlaken

Objective and Issues

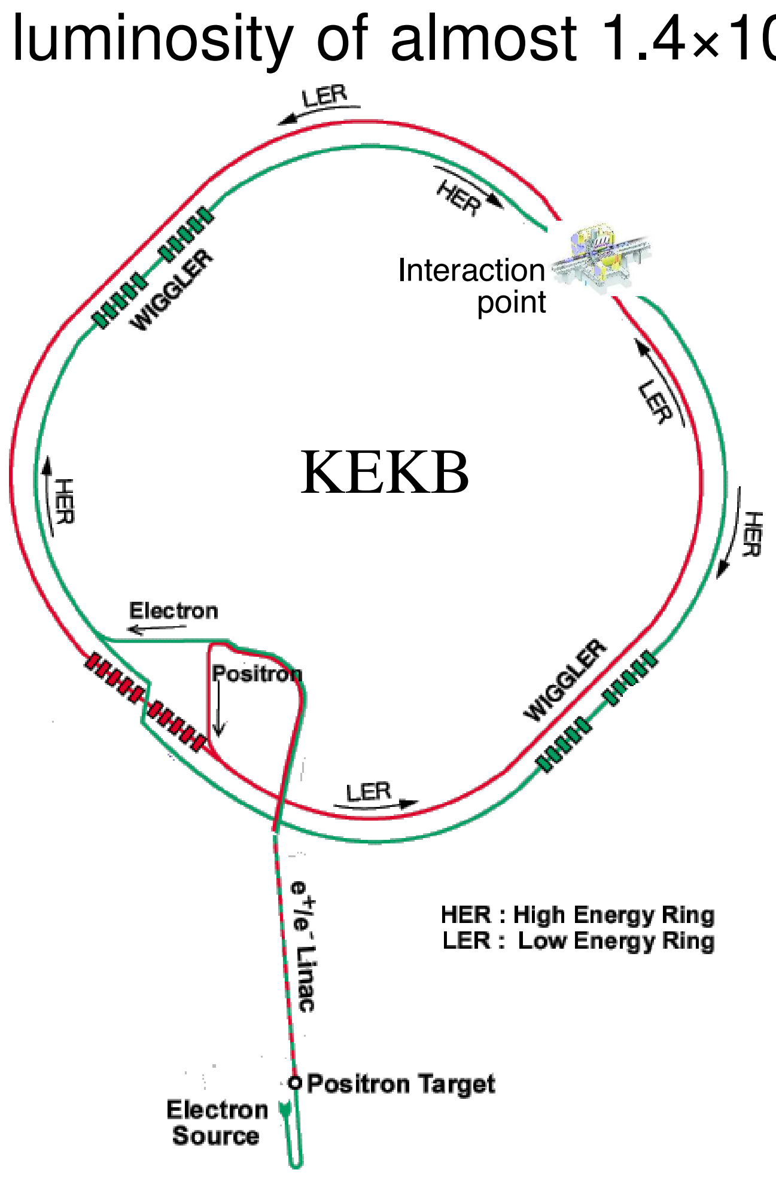
The offline processing of the data collected by the Belle detector has been recently upgraded to cope with the excellent performance of the KEKB accelerator. The 127 fb^{-1} data (120 TB to be processed) collected between autumn 2003 and summer 2004 has been processed in 2 months, thanks to the high speed and stability of the new, distributed processing scheme. We present here the physics and computing environment, before introducing the processing scheme and showing its performance.

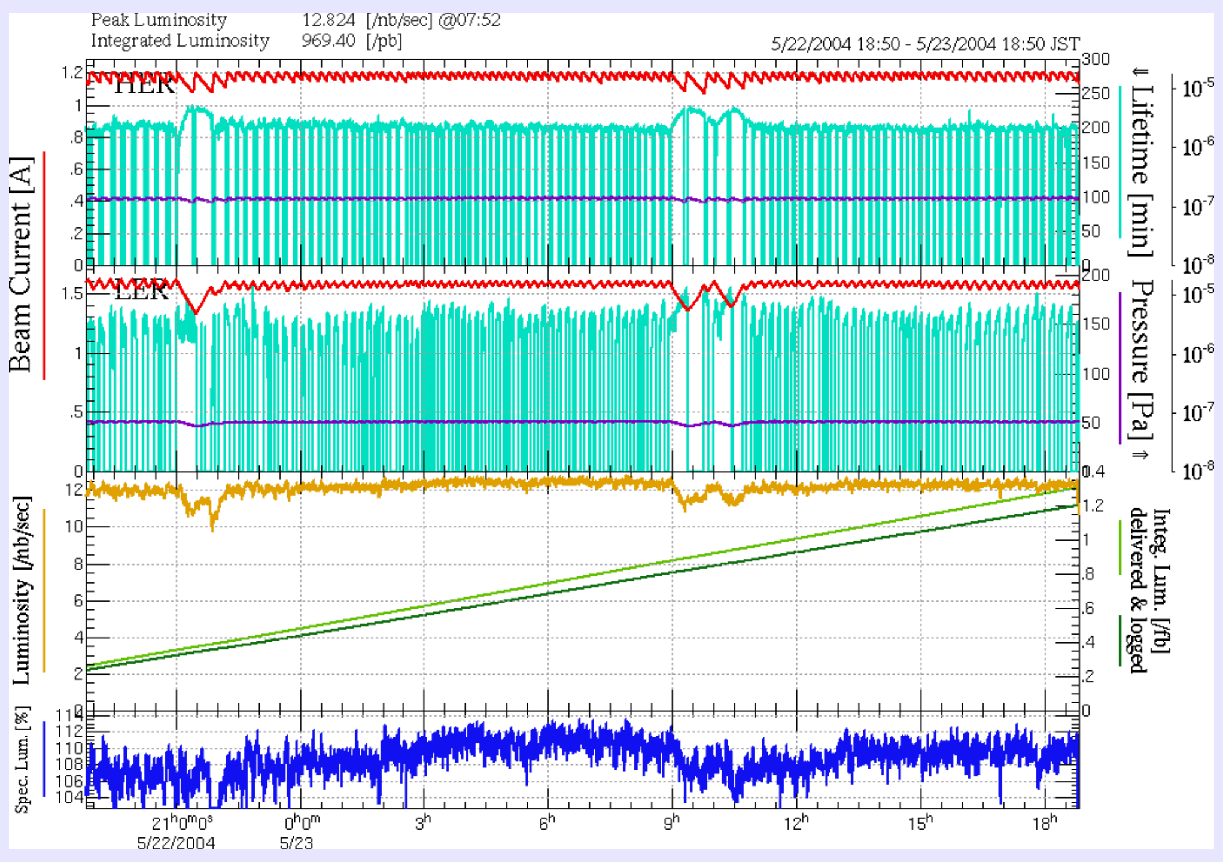
The KEKB collider

S. Kurokawa *et al.*, *Nucl. Instr. and Meth.* A**499**, 1 (2003)

The KEK B-factory is an asymmetric e^+e^- collider, which consists of an 8 GeV high-energy electron ring (HER) and a 3.5 GeV low-energy positron ring (LER). The total energy in the center-of-mass corresponds to the mass of the Υ(4*S*) resonance, which decays into pairs of *B*–anti-*B* mesons.

KEKB started operation in 1999 and has been steadily improving its operation since then. A peak luminosity of almost 1.4×10^{34} cm^{-2}s^{-1} has been achieved in June 2004 (40% more than the design luminosity). The daily integrated luminosity delivered to the Belle experiment has exceeded 1/fb (which corresponds to more than a million B meson pairs per day). Finally, the total luminosity logged by the Belle experiment corresponds to 290/fb, among which 127/fb were collected in the last 9 months.





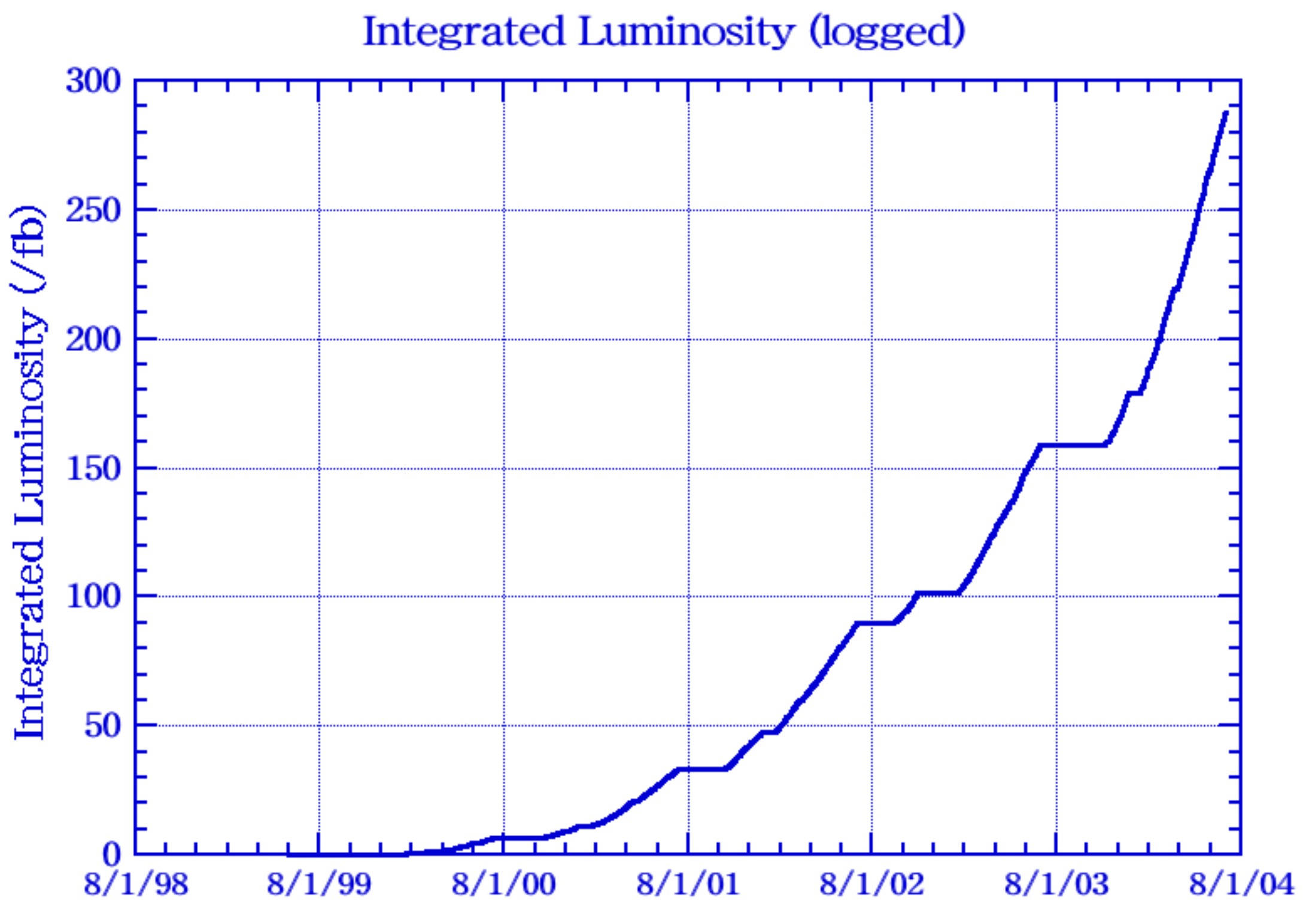
Accelerator operation in continuous injection mode.

Since January 2004, KEKB has turned to a **continuous injection mode**, where beam particle losses are compensated by “continuously” injecting beam from the linear accelerator, without interrupting data taking (see above figure of the best 24 hours). In this new mode of operation, KEKB was able to increase its integrated luminosity by 30%.

Summary of KEKB performance

- *Peak luminosity:* 1.39×10^{34} cm^{-2}s^{-1}
- *Integrated luminosity:* 944/pb/day
- *Total luminosity:* 290/fb
  ⇒ *since last summer:* 127/fb
- *Total B meson pairs:* 275 million
  ⇒ *since last summer:* 123 million

N.B. These are all world records!



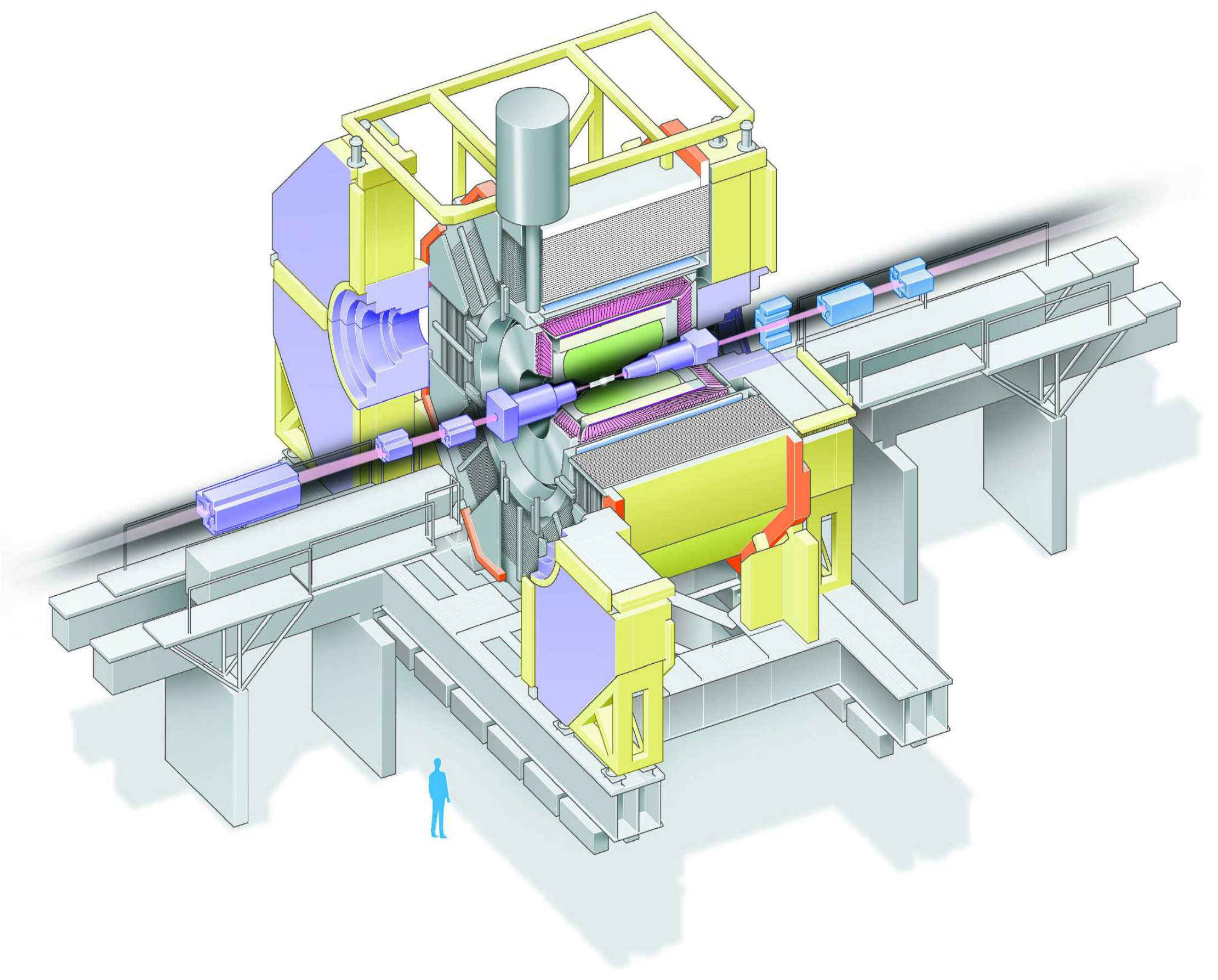
Total integrated luminosity from the start of KEKB

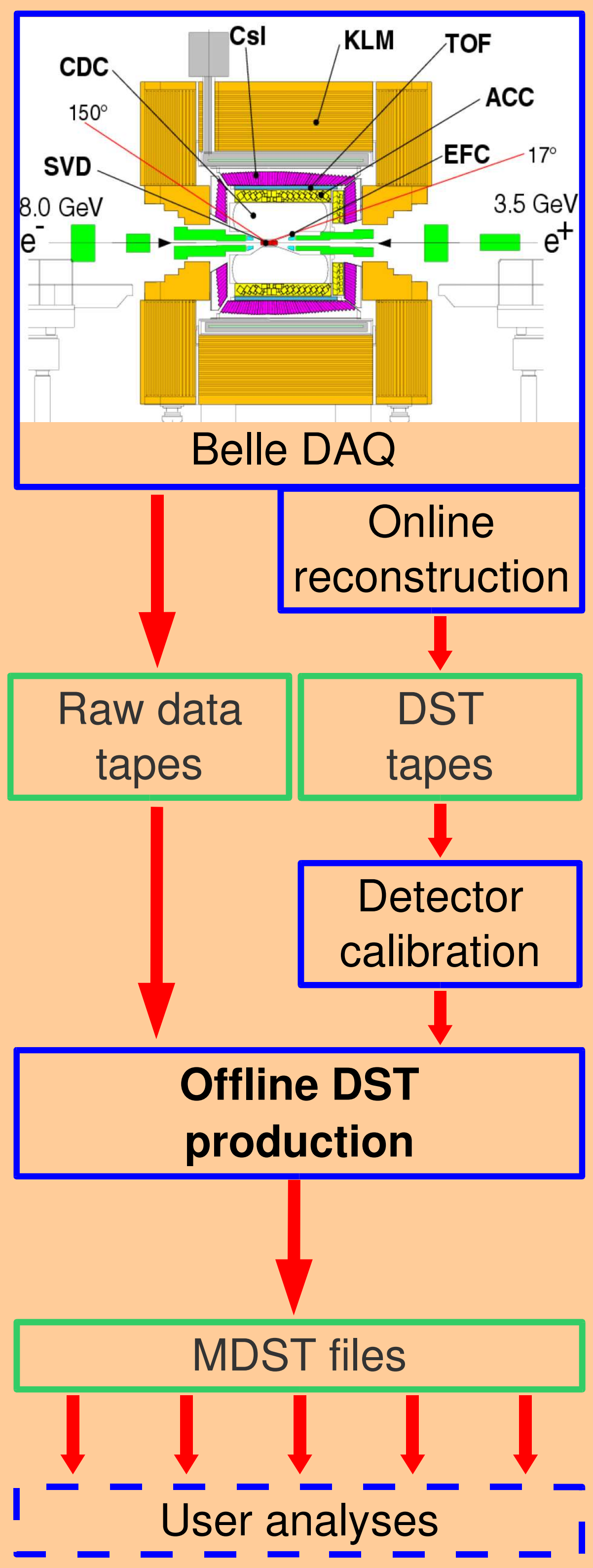
The Belle experiment

A. Abashian *et al.*, *Nucl. Instr. and Meth.* A**479**, 117 (2002)

The main goal of the Belle collaboration is the **study of the origin of CP violation**, which breaks the symmetry between matter and anti-matter. CP violation is expected to be especially large in the *B* meson family (and, indeed, Belle has already observed sizeable CP violation in several *B* meson decays).

In order to achieve this goal, Belle profits from the large sample of *B* meson pairs provided by the KEKB collider. Excellent tracking and particle identification are also required to precisely measure time-dependent asymmetries in particle decays. This information is extracted from the various subdetectors of Belle.





The Belle data flow

The subdetector information is collected by the Belle data acquisition system (DAQ) at an output rate of about 230 events per second.

This raw data is written on tapes and, at the same time, processed by the online reconstruction farm (RFARM), which outputs *reconstructed data* (DST) used for detector calibration.

The raw data tapes are then reprocessed offline, using the calibration constants, for full DST production which is stored in the form of mini-DST (mdst—essentially containing 4-vector information), to be accessed by users for physics analysis.

Belle data acquisition and processing figures

Output event rate: 230 Hz
Output data rate: 8.9 MB/s

Total raw data: 247 TB
⇒ *since last summer:* 120 TB

Total DST data: 390 TB
⇒ *since last summer:* 186 TB

Raw event size: 38 kB
DST event size: 60 kB
mdst event size: 12 kB

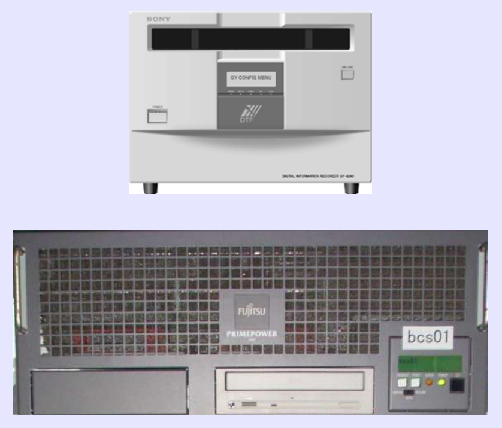
Total mdst data: **80 TB**
⇒ ***since last summer:*** **40 TB**

Computing hardware

Data storage

Raw/DST data: stored on SONY DTF2 tapes

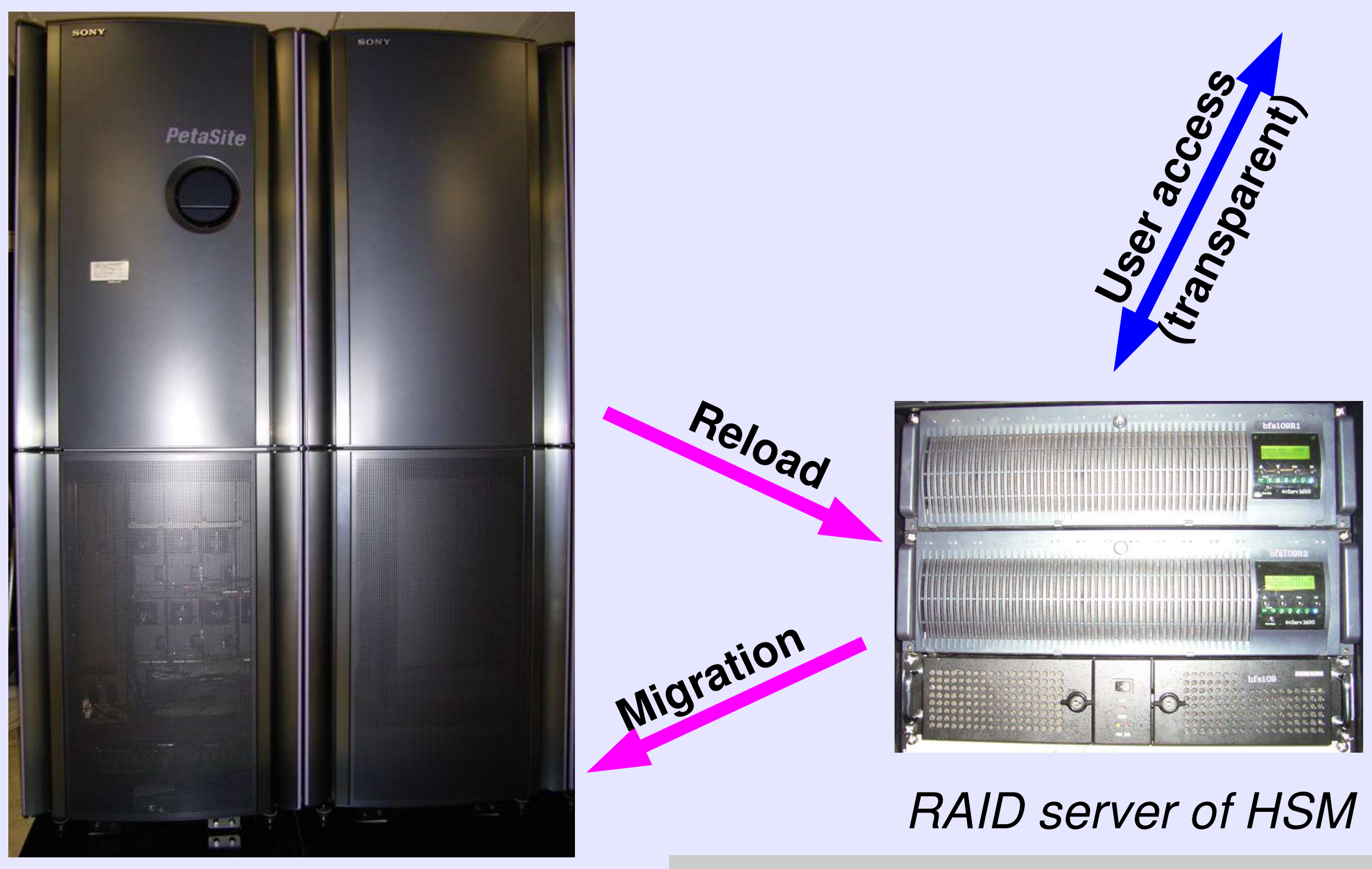
- 200 GB tapes × 2600 (500 TB total)
- 20 tape servers (0.5 GHz × 4 CPUs)
- 2 tape drives / server (40 total)
- 24 MB/s readout rate



Tape drive and server

mdst data: stored on HSM system (SONY):

- Hierarchical Management System
- Hybrid tape/disk storage
- 500 GB tapes × 900 (450 TB total) → *1.2 PB end of 2004*
- 4 SONY SAIT tape drives
- 30 MB/s readout rate
- 1.6 TB RAID disks × 16 (26 TB total)
- 8 disk servers (4 CPU Xeon 2.8 GHz)

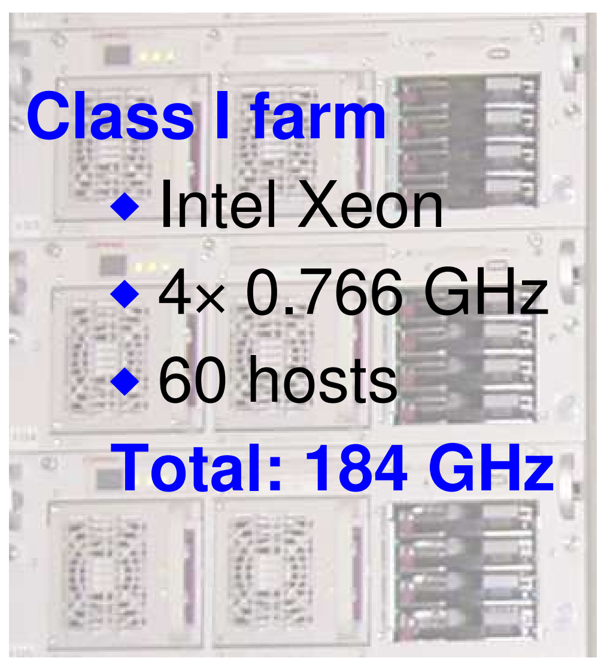


Tape library of HSM

RAID server of HSM

See also N. Katayama, *New compact hierarchical mass storage system at Belle*, poster session 1

Computing farms



Class I farm

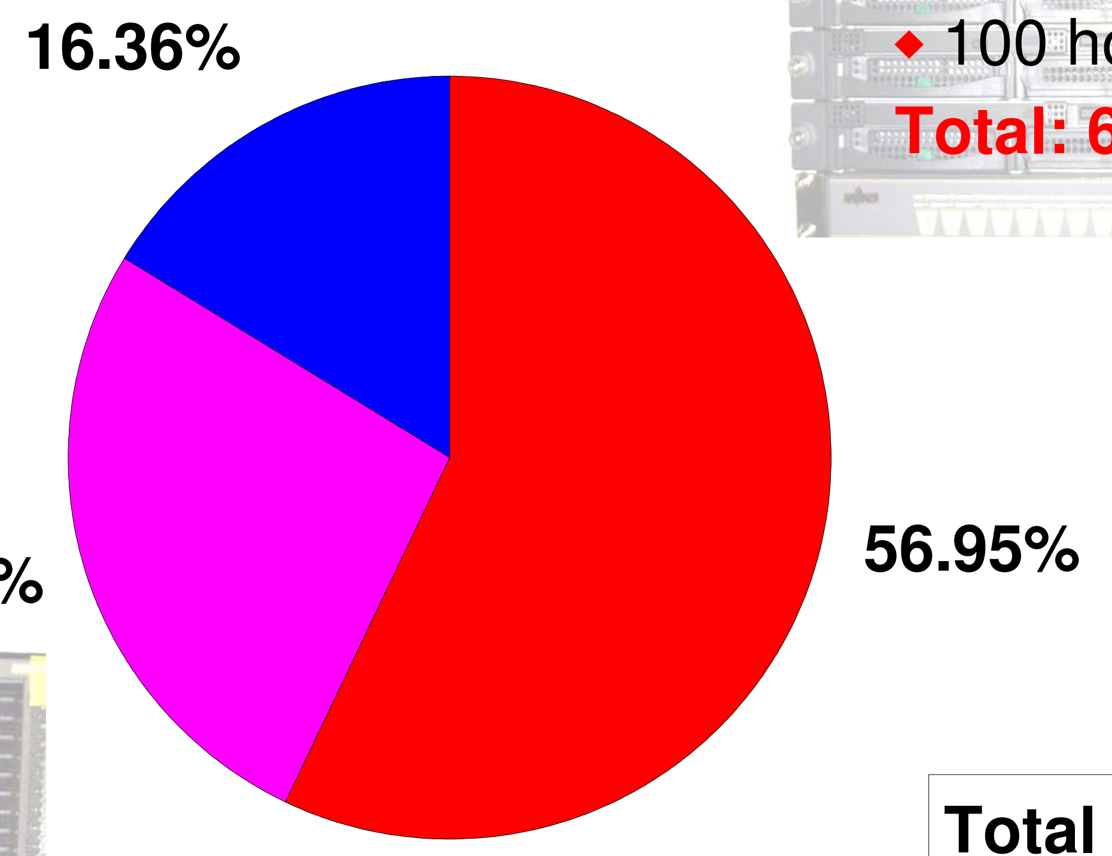
- Intel Xeon
- 4× 0.766 GHz
- 60 hosts

Total: 184 GHz

Class III farm

- Intel Xeon
- 2 × 3.2 GHz
- 100 hosts

Total: 640 GHz



Class II farm

- Intel Pentium III
- 2 × 1.26 GHz
- 119 hosts

Total: 300 GHz

Total CPU power
1.12 THz
279 hosts

N.B. These figures pertain to the DST production farms.

Network

The various parts of the computing facilities used for DST production are connected to **Gbit ethernet switches**.

Software tools

See also R. Itoh, *Experience with Real Time Event Reconstruction Farm for Belle Experiment*, Online Computing session

Belle Analysis Software Framework

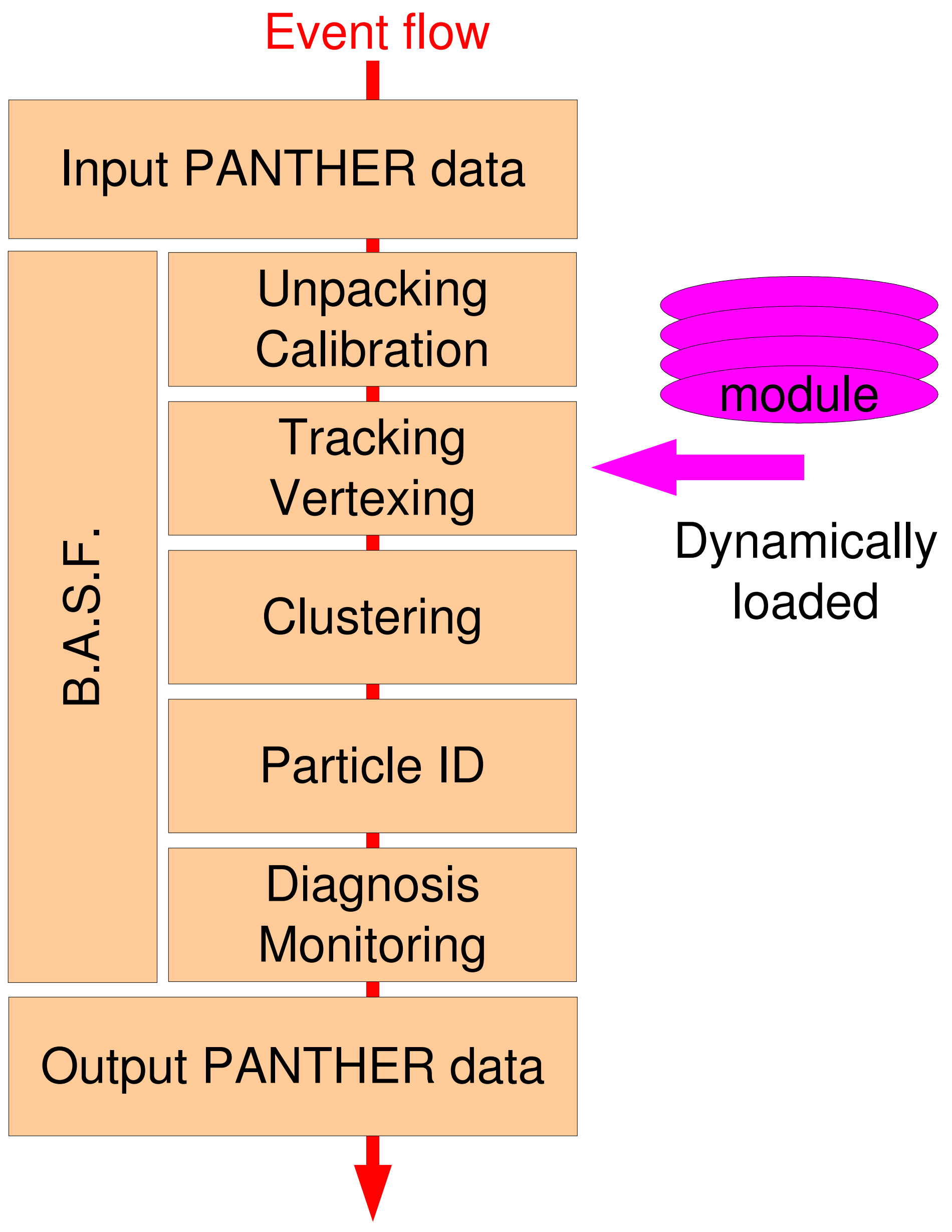
B.A.S.F. is the software framework used for **all Belle data analyses**, from data acquisition to end-user analysis and DST production.

B.A.S.F. provides an **interface** to external programs *(modules)*, dynamically loaded as shared objects at the start of a processing job. The interface includes begin and end run calls, event calls, histogram definitions, as well as a shared memory utility.

External modules actually process the event information. Several modules can be called at will, in the order specified by the user.

B.A.S.F. is written in **C++** (and so are modules).

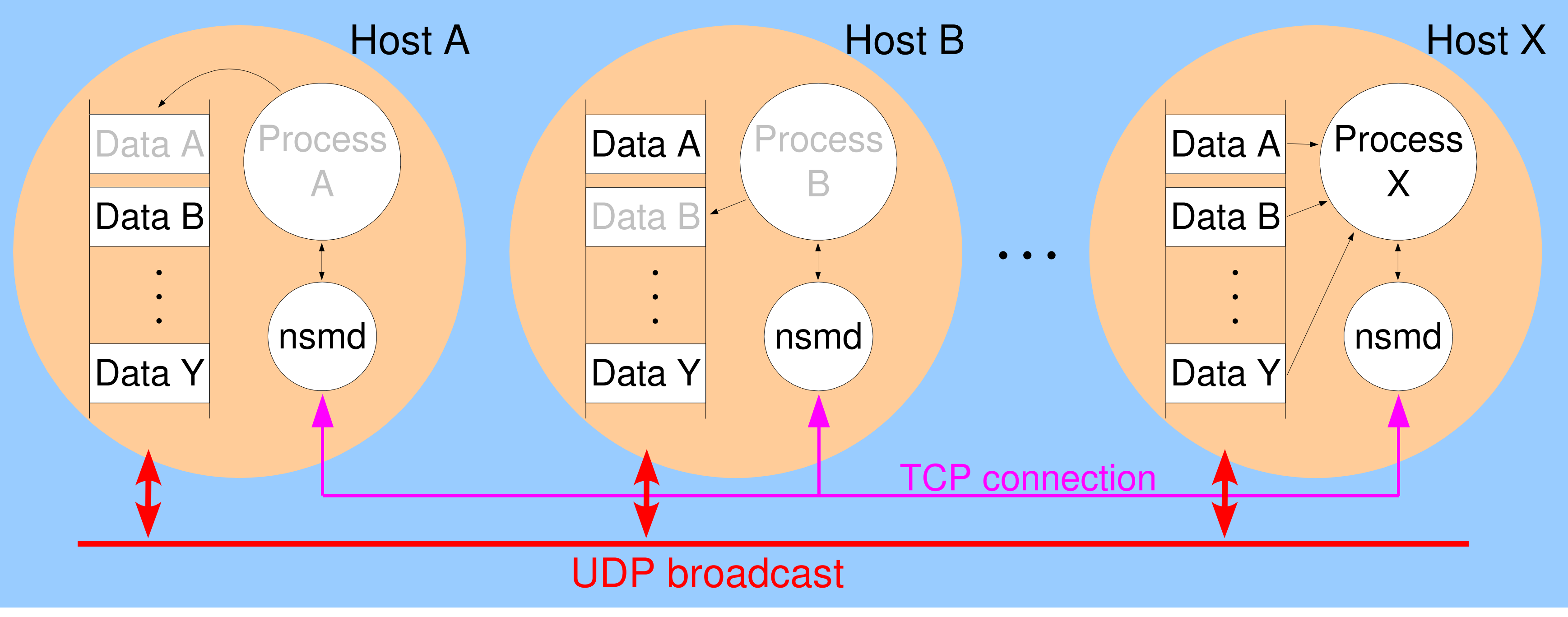
Finally, B.A.S.F. supports Symmetrical Multiprocessing (SMP), thus allowing **parallel processing** of events on a multi-processor machine.



Schematic view of the event flow in DST production. Reconstruction modules are dynamically loaded in the BASF framework and called in a given order to process each event.

Network Shared Memory

The NSM package provides tools for **information exchange** over a TCP/IP based LAN. It allows processes running on different machines to share memory across the network, or send requests and messages to each other.



The DST production scheme relies on a number of different software tools listed here, three of which are “home made”:

- **BASF** – the framework of all data analyses
- **PANTHER** – the Belle data format
- **NSM** – inter-process communication over the network

The PANTHER format

The input and output of Belle data, as well as transfer between modules, is managed by the *PANTHER* system. It is used consistently from raw data to user analysis. The PANTHER format consists of compressed tables *(banks)*, using the standard *zlib* libraries. A cross-reference system is implemented in order to allow navigation between the tables. The table formats are defined in B.A.S.F. in ASCII header files, which are loaded before the modules. Users may define their own tables.

Other tools

- **postgresql database system**
  An important part of the information relevant to DST production is stored in databases:
  - calibration constants;
  - raw data, DST and mdst files information;
  - tapes and tape drives information;
  - PC farms information

  DST production uses a dedicated postgresql server, mirrored from the main database server.

- **LSF batch queues**
  Tape servers are operated through the LSF queing system.

- **Redhat and Solaris**
  The computing farms run on various versions of Redhat Linux: Redhat 6.2 (class I), 7.2 (PC class II) or 9.0 (PC class III).
  The tape servers run Solaris (Sun OS 5.7).

Overview

The new DST processing scheme has been implemented between March and May 2004.

It is based on a **Distributed version of B.A.S.F.** *(d-basf)*, first developped for the online processing (RFARM) and adapted to the configuration of offline processing.

The computing facility used for DST processing was divided in a number of **dbasf clusters** that independently process groups of events *(runs)*.

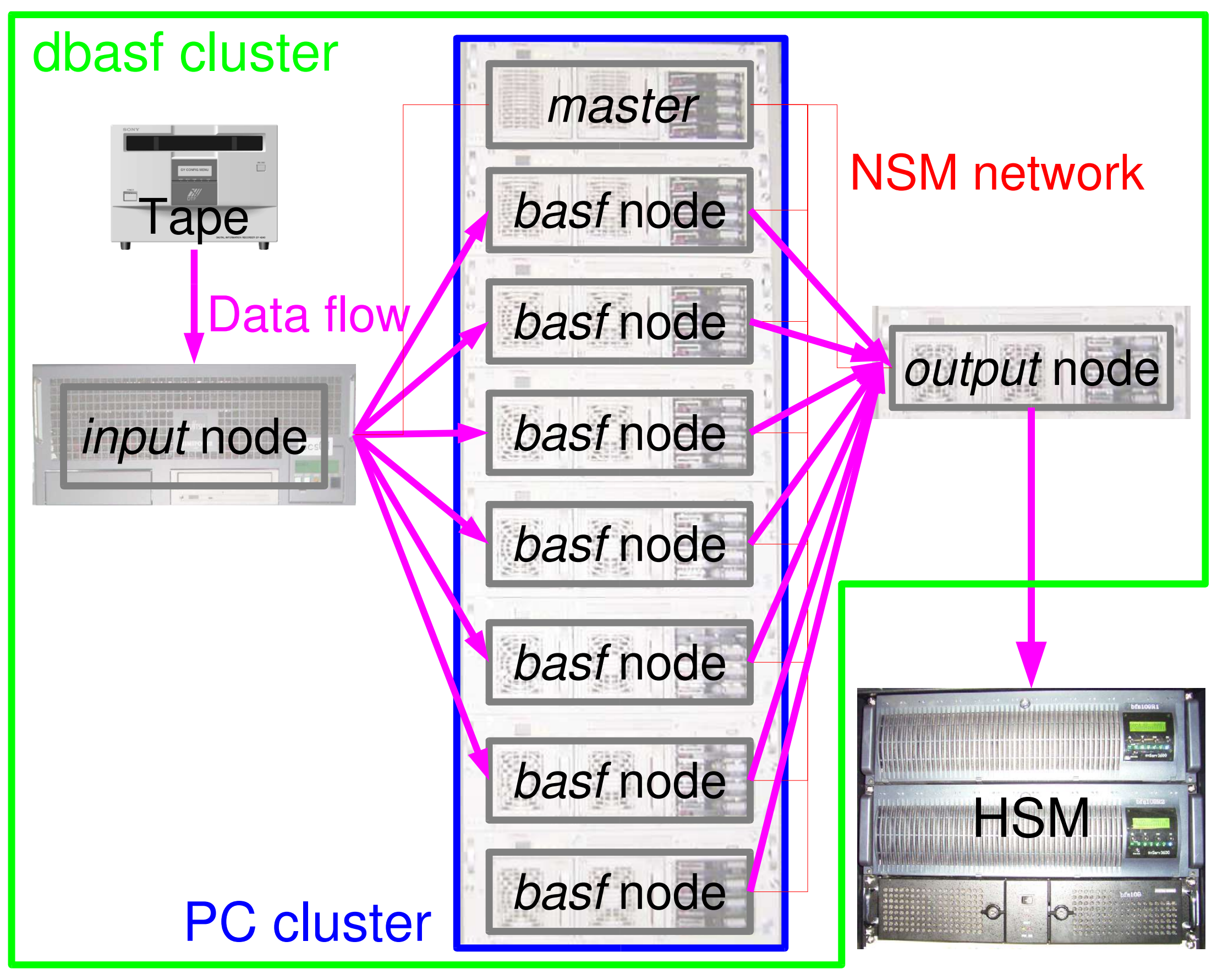
The production is fully managed by a **steering Perl script** *(dcruncher)* that allocates jobs and surveys the global DST processing operations.

The task of the new scheme was to complete the processing of all the data accumulated from autumn 2003 to summer 2004 before the summer physics conferences.

The Distributed B.A.S.F.

In order to increase its **parallel-processing** ability, B.A.S.F. was extended to a distributed version that relies on the NSM system. A **dbasf cluster** physically consists of a tape server and **30 to 40 PC hosts** (PC cluster).

The data is distributed by an *input node* to all PC hosts *(basf nodes)*, on which basf processes are running. The output is redirected to a single *output node* (one of the PC hosts) that send it to the HSM storage system.



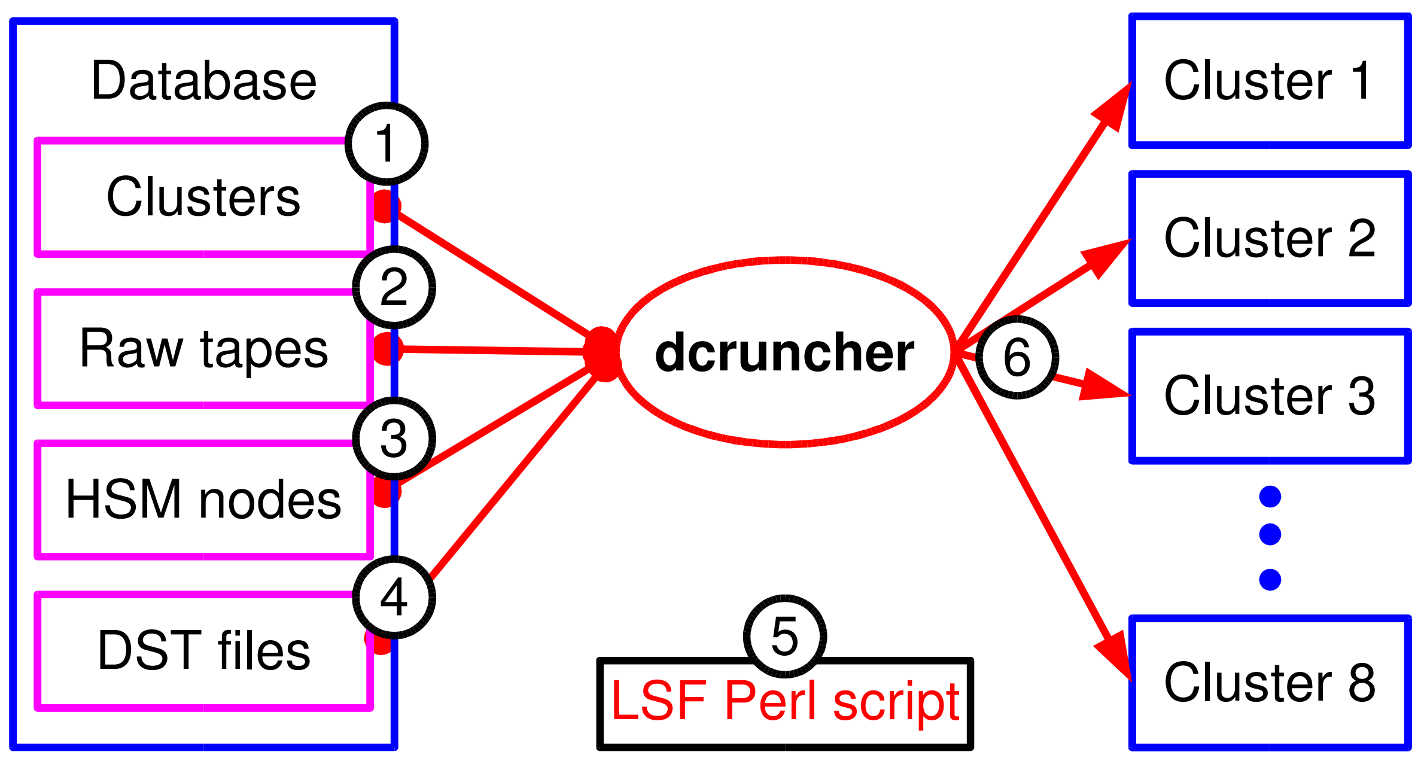
Finally, the synchronization among the various nodes is managed by a *master node* runnning on one of the PC hosts. It is also used to externally check the cluster status.

The processing scheme

The production is driven by the **dcruncher** Perl script that runs on a mainframe Solaris server (similar to tape servers). In order to allocate jobs, it interacts with the database dedicated to DST processing.

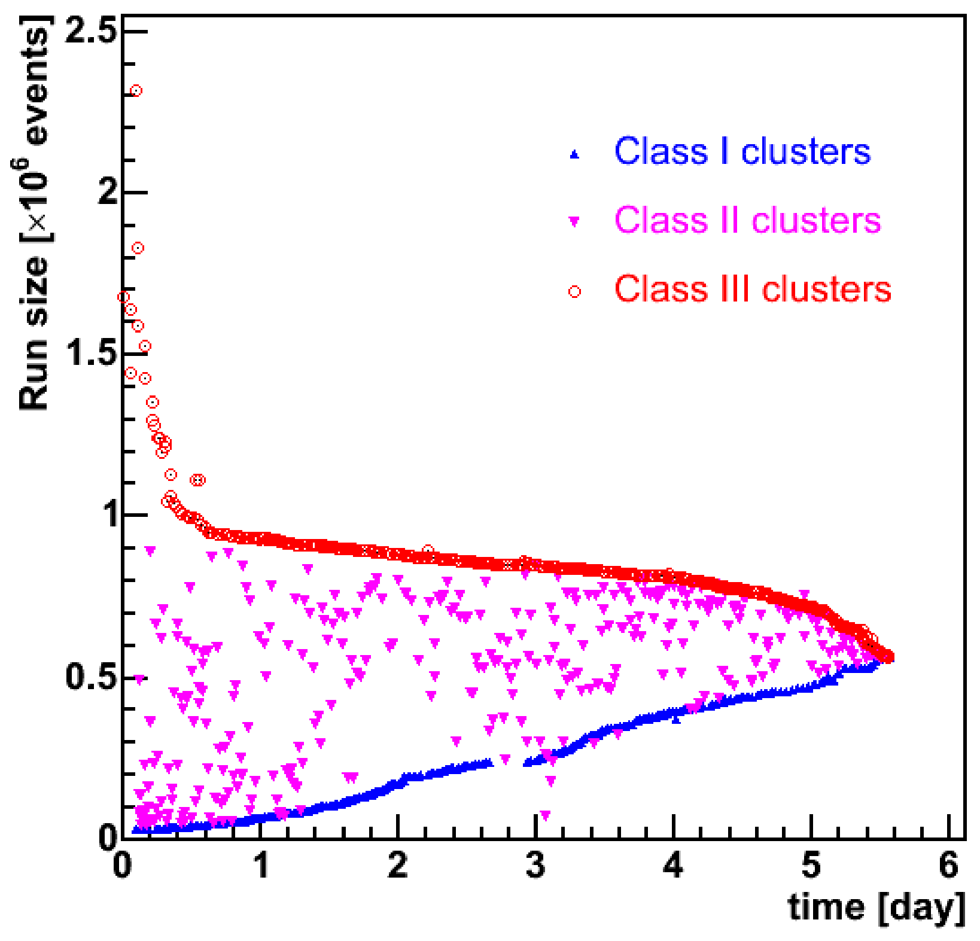
During the job, the LSF script checks the operation of the dbasf cluster. **After** it has finished, dcruncher checks that output files exist and have sensible sizes. In case of any **failure**, the DST team is immediately informed by e-mail.

1. Check for free **clusters** (fastest first).
2. Check for available unprocessed **raw tape** (depending on cluster type: see speed optimization).
3. Find location for output file on **HSM**.
4. Insert corresponding entry in **DST files** table.
5. Write Perl script job to submit to LSF batch queue.
6. Send job to relevant cluster (the script job is run on the tape server of the cluster).
7. Wait 5 minutes and restart the loop.



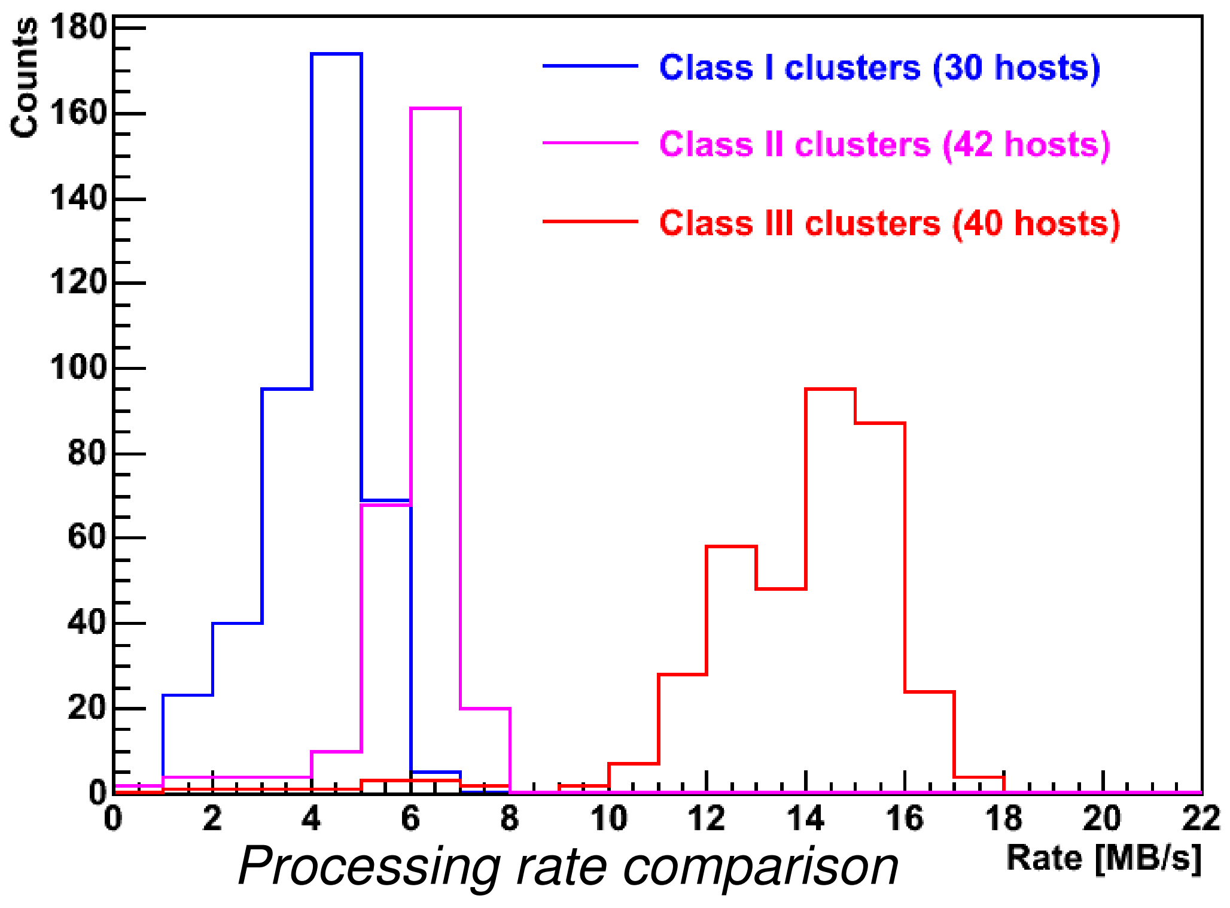
Speed optimization

A simple algorithm was implemented to make the best use of the computing power: faster clusters process runs by decreasing size, and slower cluster by increasing size. This is nicely illustrated on the figure below.



Performance

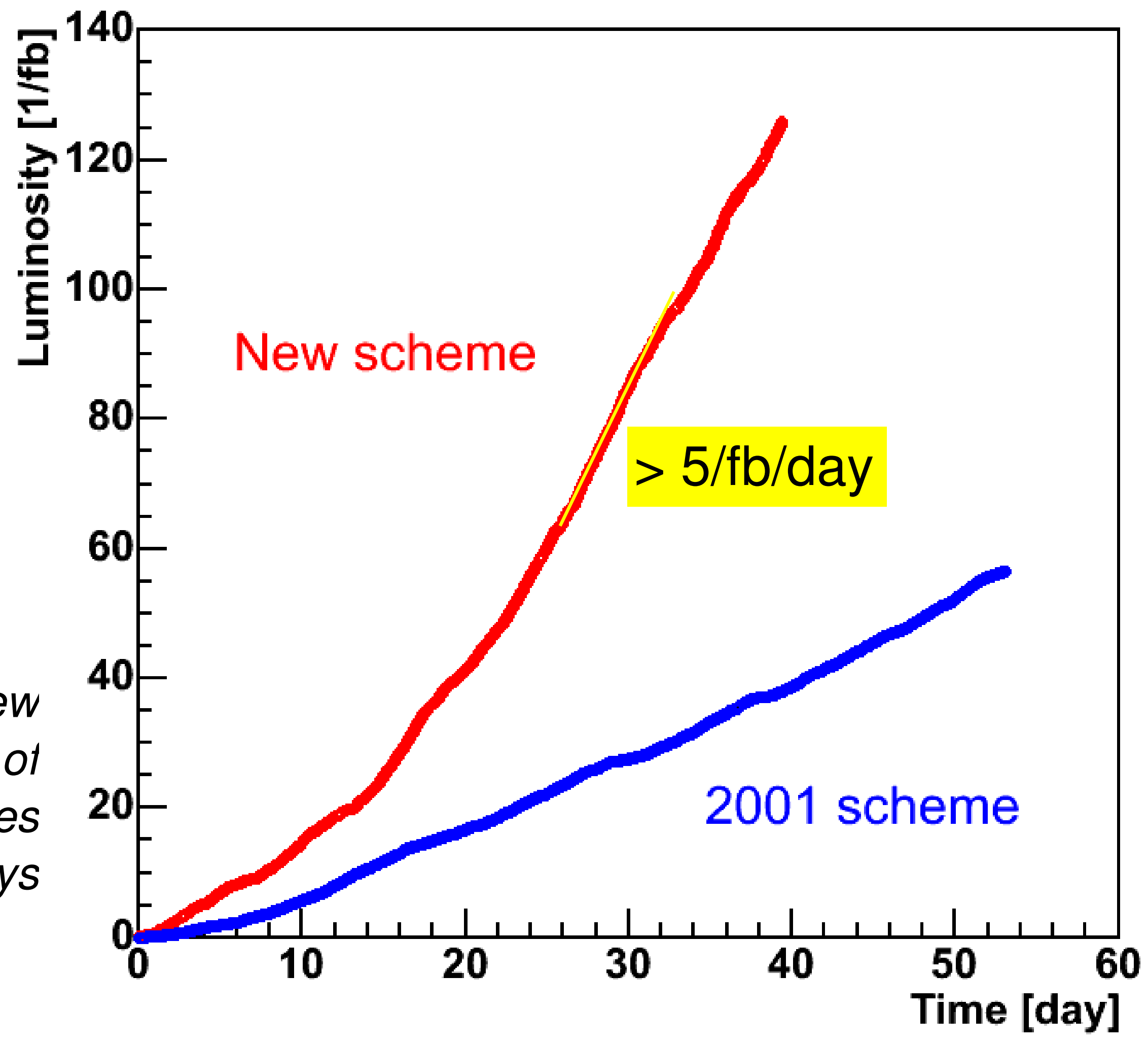
●*Processing time:*	56 days	
→ *alive:*	39 days	
●*Input data size:*	127/fb 3.3×10^9 events 120 Tbytes	
●*Output data size:*	40 Tbytes	
●*Number of jobs:*	3408	
●*Processing rate*		
average:	3.2/fb/day	37 MB/s
maximum:	**5.1/fb/day**	60 MB/s
old system:	1.1/fb/day	



Processing rate comparison between the 3 classes of PC farms

Failure rate

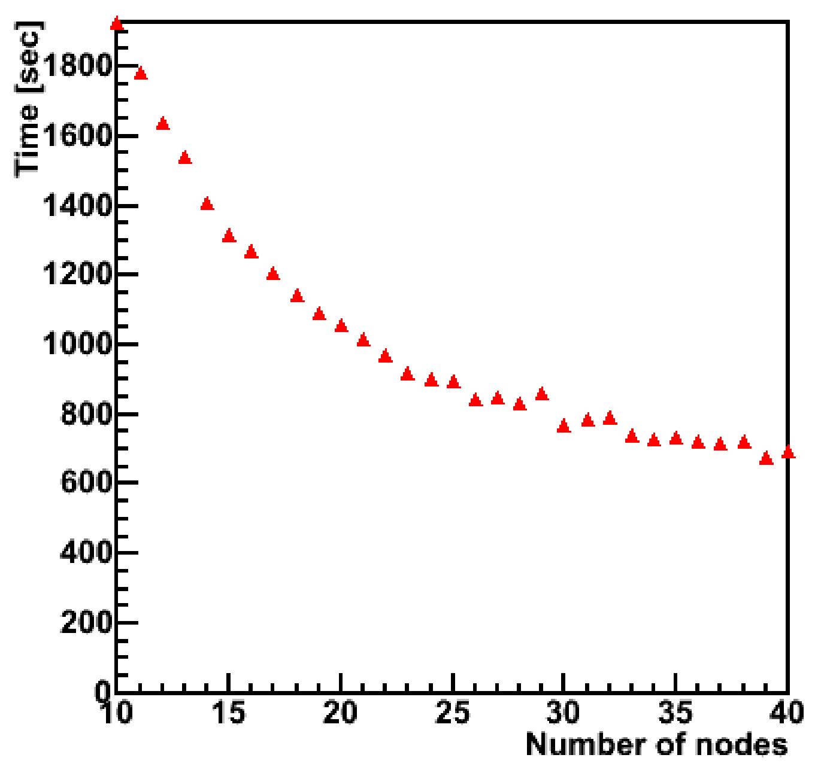
Inter-process communication	0.2%
Database access	0.1%
Tape drives	<0.1%
Network	0.3%
Total	**0.7%**



Performance comparison between the old and the new scheme. The old scheme used a slightly different version of dbasf. In the best period, the new scheme processed at rates greater than 5/fb/day. The dead-time periods due to delays independent of the DST processing are subtracted.

Limitations

- ◆ **Database access** is one of the main issues, since processing makes heavy use of the database, in particular at the start of a job.
- ◆ The limited **CPU power** of tape servers which distribute the data the dbasf clusters is the present bottleneck (see figure).
- ◆ The **network bandwith** between the input server and basf nodes may eventually limit the processing power.



Conclusion

The new offline processing scheme started running on May 20, 2004. In 39 days of stable running, it succesfully processed the 3 billion events collected by KEKB with a maximum rate of 5/fb/day, 5 times faster than the data acquisition and the previous scheme. With the expected increase of the KEKB luminosity, however, the DST production will face further challenges… Room for improvement of this scheme still remains.