14 June 1999

MONARC testbed and a preliminary measurement on Objectivity AMS server

K. Amako, Y. Karita, S. Kawabata, A. Manabe
Y. Morita*), T. Sasaki, H. Sato, Y. Watase
KEK - Japan
*) on leave to CERN IT/ASD

The aim of the MONARC Testbed Working Group is to provide the measurements of the key parameters which govern the behaviour and the scalability of the various models of the distributed computing system. Testing enviroments have been set up at CERN and at KEK with a raw data modelling on Objectivity/DB based on the ATLAS 1TB milestone program. A preliminary measurements on Objectivity AMS performance has been made.

1. Introduction

The primary goal of the MONARC project is to identify baseline computing models that could provide viable and cost-effective solutions for the data analysis of the LHC experiments [1]. A simulation tool has been developed by the MONARC Simulation Working Group to evaluate the performances of various models of distributed data analysys environment [2]. The role of the Testbed Working Group is to validate the accuracy of the simulation model and to directly measure the performances of various types of the distributed data analsys models with different "use-case" of Object Database Management System (ODBMS), such as database replication and caching.

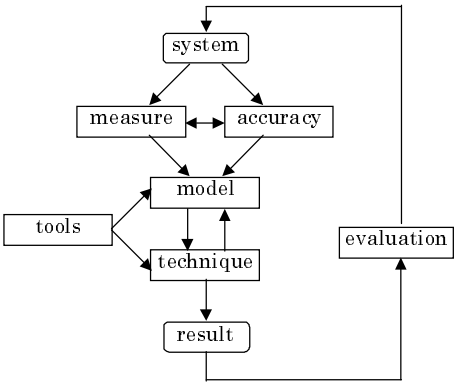

Fig. 1 Performance Evaluation Cycle

Evaluation of the computer and network system is the iteration of the processes such as measurement, modeling, development of simulation tools and validation of the simulation

techniques (Fig. 1). With sufficient iterations of the above cycle, one could predict the behavior of the system for various types of loads with sufficient accuracy. Therefore the measurements needed for the MONARC project should be closely related to the specific modeling of the simulation tools and techniques.

A number of "use-case" software has been identified within the Testbed WG, and a series of measurements have been started at various sites such as Milano [3].

In this document, MONARC-dedicated hardware and software setup at CERN and KEK, wide-area network configuration is described. Also preliminary measurements made on these configurations are presented.

2. Hardware and Software Configurations

2.1 Set up at CERN and KEK

A dedicated SUN Enterprise 450 server has been set up by Tim Smith of CERN IT/PDP as **monarc01.cern.ch**. It has four 400MHz CPUs with 512 MBytes of memory, equiped with two 9 GB disks. The power of this machine at single CPU is rated as 113 CU (CERN Unit) [4]. SPECint95 is rated as 17.4 and 14.0 for SPECint_base95 [5]. At KEK, a SUN Ultra60 workstation (CPU 360MHz) with 100 GB disk has been dedicated for the test of performance measurements. SPECint95 and SPECint_base95 are 16.1 and 12.8 respectively. Following software has been set up on these machines:

- SUN Solaris 2.6 with C++ v.4.2
- Objectivity/DB v.5.1

In addition to the above machines, two more SUN workstations of ATLAS group were used in this measurement.

- **atlobj01.cern.ch** (used as a Lock Server)
  SUN Ultra5 (270MHz CPU), 192 MB memory
- **ablobj02.cern.ch**
  SUN Enterprise450 (300 MHz dual CPU), 512 MB memory (85 CU)
  Disks: Hitachi RAID (rated around 25 MB/sec)

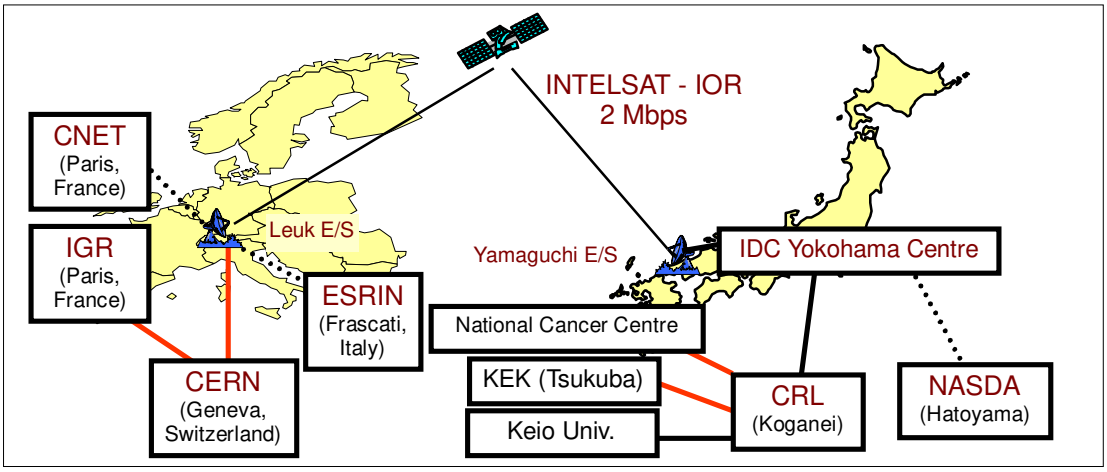

Fig. 2 Outline network topology of Japan-Europe Gamma Project

2.2 Network Between CERN and KEK

To evaluate the impact of different kind of wide-area network configurations, various types of network will be utilized in the Testbed WG. The network between CERN and KEK will be alternated between two kind of links. One is the surface-based public link of Japanese NACSIS academic network with 2 Mbps bandwidth, which is shared by many other academic institutes and universities between Japan and Europe. The other link is a dedicated ATM Permanent Virtual Circuit in Japan-Europe Gamma project, which utilizes the 2 Mbps link of INTELSAT satellite (Fig. 2). The satellite link between CERN and KEK workstations are routed statically by specifying IP numbers at each end of the routers. This link is planned to be tested until the end of August 1999, with a possible extension of the project with much higher bandwidth.

3. Stress Tests on Local and Client/Server Configurations

3.1 Software Setup

To populate the database with fully-simulated ATLAS raw data, ATLAS 1 TB milestone program has been used [6]. It generates raw data with GEANT3 and the event data model has been converted into LHC++ environment. Fig. 3 is the class diagram of the event structure of this program. A pointer of the vectors to the detectors are stored in "Event container", while the VArray of detector digits are stored in "raw data container (Fig. 4). A hundred events corresponding to 57 MBytes in the database is used in this test.

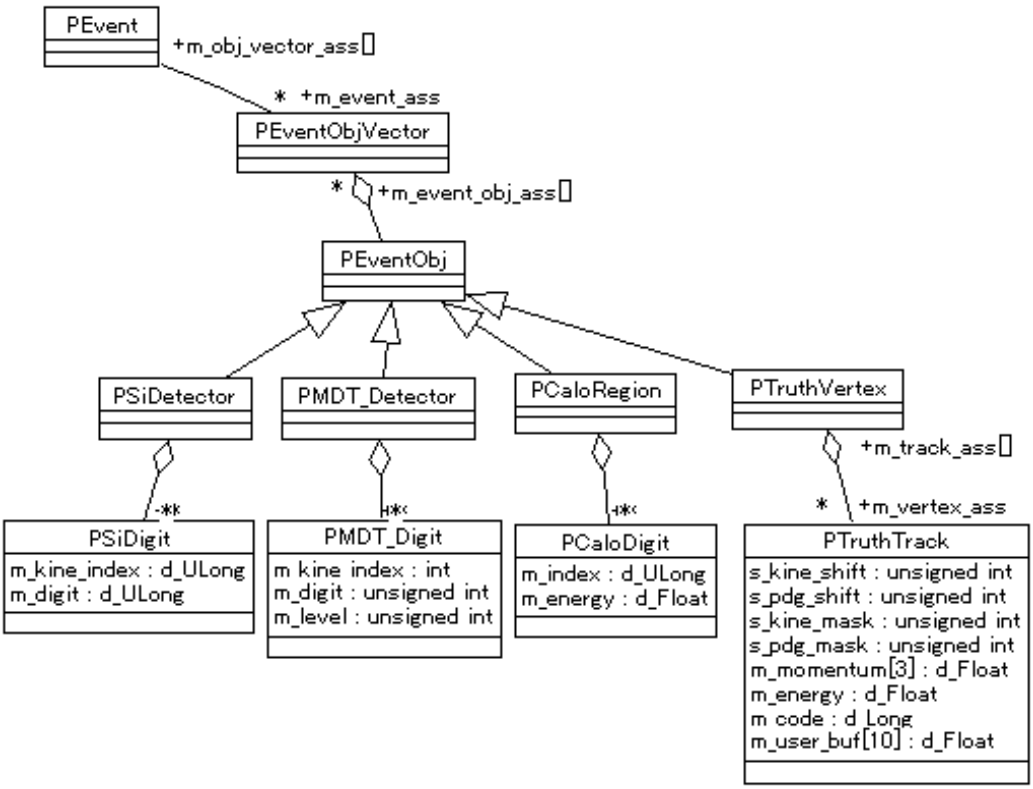

Fig. 3 Class diagram of the ATLAS 1TB milestone test [6].

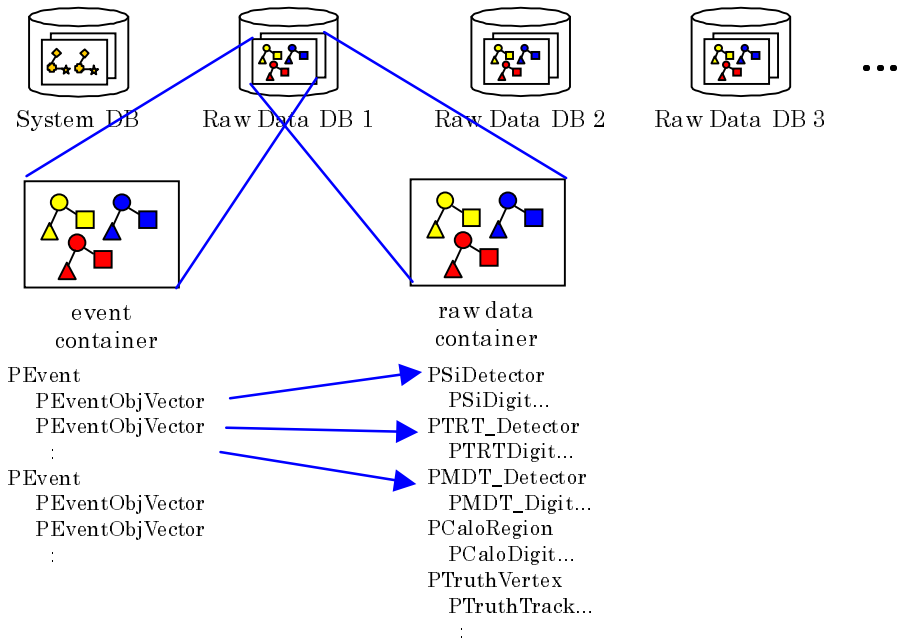

Fig. 4 Data Model of the ATLAS 1 TB Milestone

Clustering of the objects in the containers will affect the performance of the writing and reading transactions, depending on the different data models and access patterns of the application. A study of tuning the data modeling is being carried out in ATLAS Database Group [7]. However, the measurement of AMS behavior will benefit from various types of data modeling and access patterns with some extreme cases, and the model used in our measurement should be regarded as just a starting point of many possible data modeling.

3.2 Database Configurations

Two local federation configurations ("mon" and "atl") and one client-and-server configuration ("atl-mon") have been used, as shown in Fig. 5. In "mon" and "atl" configurations, the event database is located on the same machine as the analysis jobs. In "atl-mon" configuration, the database is located on the server (atlobj02) while the analysis jobs ran on the client node (monarc01). The client and server machines are connected into the same CERN internal network with a Gigabit ethernet switch. The Objectivity page size is fixed at 32 KBytes in this test.

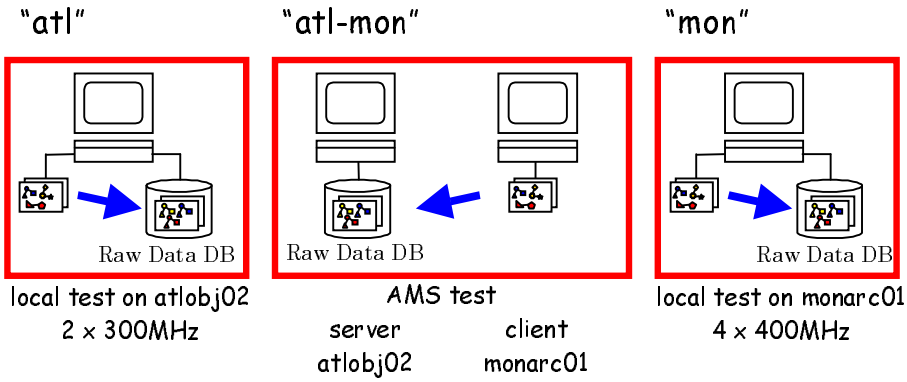

Fig. 5 Local and Client/Server Configurations of the Database

3.3 Stress Test

A short C++ program, which reads and navigates through all objects in the database with the object iterator, is run on each configuration concurrently up to 32 jobs. The program scans all objects but does little on "user analysis". Therefore the CPU cycle spent on each job is mostly dedicated to the I/O system call, object reformatting and

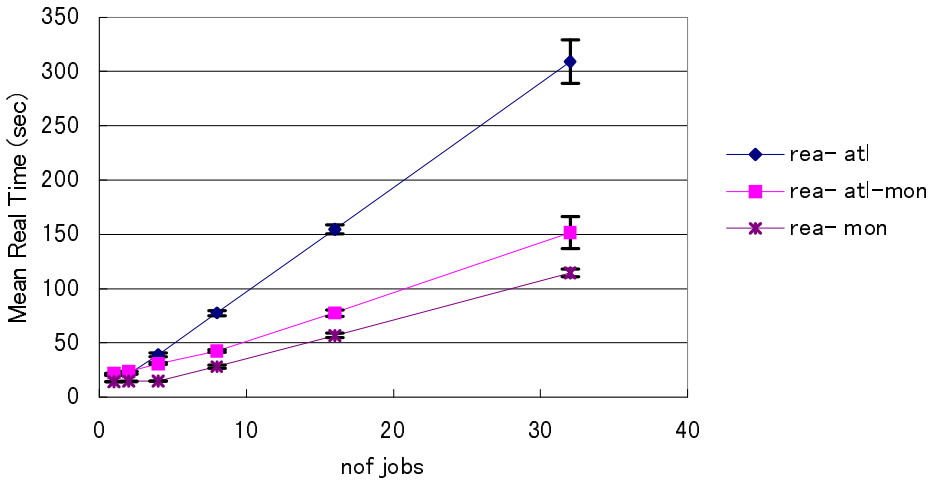

Fig. 6 Average job execution time vs number of concurrent jobs

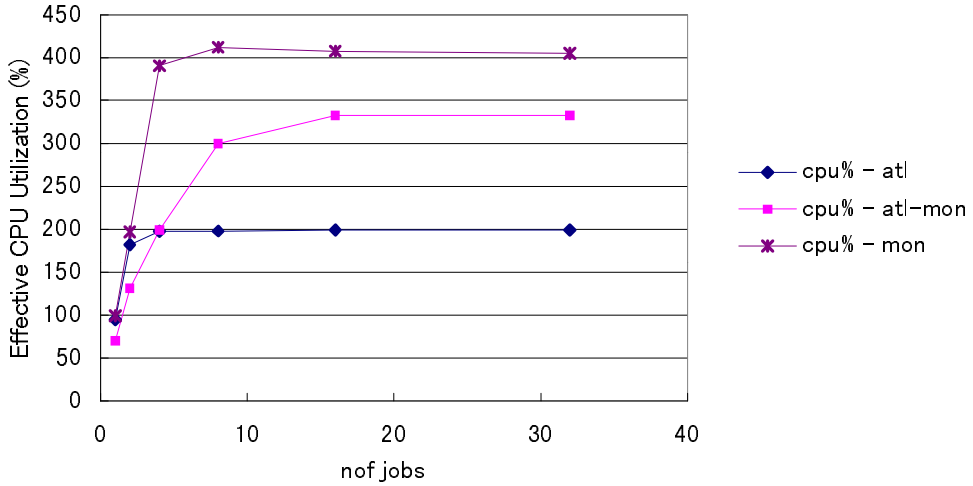

Fig. 7 Aggregate CPU utilization vs number of concurrent jobs

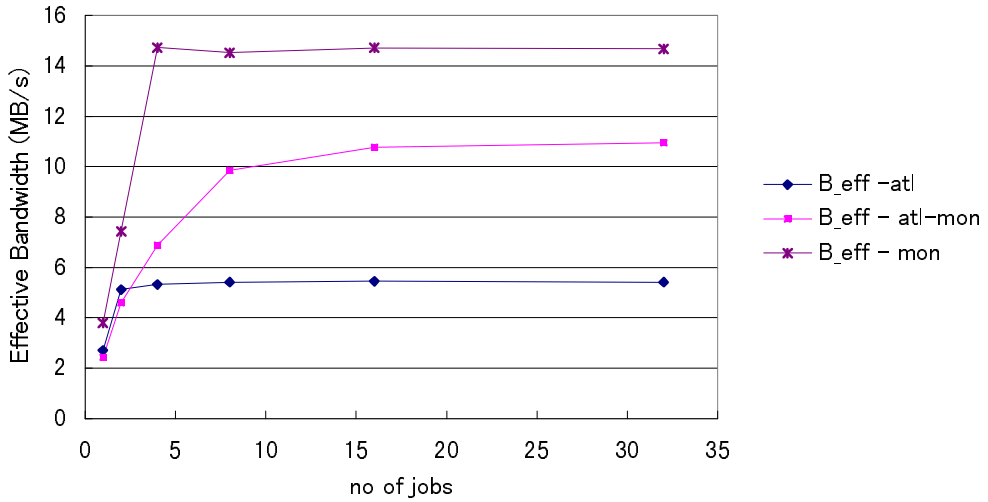

Fig. 8 Aggregate I/O Bandwidth vs number of concurrent jobs

other internal operations of Objectivity. Job execution time, CPU utilization time and fraction of CPU usage are measured with UNIX *time* command.

The average job execution time with different numbers of concurrent jobs is shown in fig. 6. In the local configuration test "atl" and "mon", job execution time stays same until the number of jobs outnumbers the number of CPU on the system (two in "atl" and four in "mon"). With more jobs running on the system, job execution time grows lineary with different slopes due to the performance difference between atlobj02 and monarc01. In the client and server configuration using Objectivity AMS server ("atl-mon"), job execution time is slower than the local test on atlobj02, when number of concurrent jobs is less than four. For more than four jobs running on monarc01, job execution time on "atl-mon" outperforms the local test of "atl".

Shown in fig. 7 is the aggregate CPU utilization percentage of the concurrent jobs. Due to imprecision of the way UNIX *time* command estimates the CPU utilization of each process, and also due to the number of digits of the output of the command, this measurement should be regarded as a rough approximation of the system utilization. Nevertheless, the CPU utilization of the local configurations are almost full both in "atl" and "mon", while the CPU utilization of the client machine (monarc01) in the client and server configuration is about 70% below the full utilization (18% per CPU). This suggests that the system in this test is in the CPU-bound state in the local configurations, as Objectivity I/O transaction requires certain amount of CPU cycle per operation [8]. In the client and server configuration, certain amount of client CPU cycles is wasted due to the transaction-related bottleneck or overhead elsewhere. It will be interesting to test the CPU load on the server side with large number of client nodes with large number of concurrent jobs. Further measurements on locating the source of the bottleneck are being planned.

Fig. 8 is the plot of aggregate effective global throughput, calculated form the total size of the database accessed divided by the average job execution time, versus the number of concurrent jobs. Similarity of the plot to the CPU utilization again suggests that the analysis jobs used in this test is CPU-bound in the local configurations.

Table 1 summarizes the results of the measurements and table 2 lists the calculated effective global throughput.

	monarc01 local			atlobj02 local			atlobj02-monarc01		
jobs	time (s)	rms	cpu%	time (s)	rms	cpu%	time (s)	rms	cpu%
1	14.23	0.00	99.1	19.93	0.00	94.5	22.08	0.00	69.7
2	14.44	0.13	196.9	21.04	0.09	181.8	23.43	0.06	131.4
4	14.62	0.05	390.6	38.81	1.95	197.6	30.63	0.79	199.3
8	27.96	1.48	412.0	77.48	2.38	198.0	42.40	1.12	300.1
16	56.59	2.06	407.4	154.41	4.23	199.1	77.50	2.99	332.6
32	114.33	3.57	404.8	309.23	20.08	199.1	151.59	14.57	332.6

Table 1 Job execution time and CPU utilization

Jobs	monarc01	atlobj02	atlobj02-monarc01
1	3.8	2.7	2.4
2	7.4	5.1	4.6
4	14.7	5.3	6.9
8	14.5	5.4	9.8
16	14.7	5.5	10.8
32	14.7	5.4	11.0

Table 2 Calculated effective global throughput [MB/s].

4. Hardware Performances of the Testbed Configurations

To normalize the results of the stress test measurements, the raw I/O speed of the disks and network on monarc01 and atlobj02 are measured. The disk speed is measured with a simple C program using UNIX system calls *write()* and *read()*. The speeds of block writing and block reading are measured on the same disk and file system as in the above measurements (Table 3). Block writing and block reading speeds are measured for two different block sizes.

File size [MB]		54		864	
Block size [KB]		54	1024	54	1024
monarc01	write [MB/s]	14.8	16.2	13.5 $\rightarrow$ 9.2	15.9
	read [MB/s]	207	207	18.5	22
atlobj02	write [MB/s]	18.3	25.9	20.7 $\rightarrow$ 8.0	27
	read [MB/s]	31	44	22.8	28.3

Table 3 Disk I/O speed measured with UNIX system calls.

Actual writing and reading speed depend on many factors such as: how files are located on the disk sectors, utilization of the caches at the disk drive or at the UNIX system level. In the case of writing an initial 864 MB file with 54 KB blocks on monarc01 for example, the writing speed was faster (13.5MB/s) than overwriting by re-opening the existing file (9.2MB/s). The disk on monarc01 has a large cache and the reading speed of 54 MB file is faster than the writing. One should be carefully compare the

performance with other measurements with the same circumstance.

The network interfaces of monarc01 and atlobj02 are Gigabit ethernet, with one router is placed in between. The minimum/average/maximum round trip time of the 84 bytes packets were 0.72/0.82/1.02 msec respectively. FTP transfer speeds of the same database file used in the above test are given in Table 4.

		MB/s	rms
monarc01 $\rightarrow$ atlobj02	put	7.17	0.12
	get	7.26	0.58
atlobj02 $\rightarrow$ monarc01	put	6.48	0.16
	get	6.56	0.08

Table 4 FTP transfer speed between monarc01 and atlobj02.

5. Conclusions

Preliminary measurements of Objectivity AMS behavior has begun. The aims of the measurements are to provide the model parameters of Objectivity to the MONARC simulation tools, with enough precision and the scalability. This test should be regarded as a starting point of the set of measurements in this direction.

Performance of the Objectivity applications is expected to vary from a use-case to use-case, with different data models and different access patterns, and also with different tuning of the Objectivity internal parameters. A simple event loop access program used in this test, with little CPU utilization in the user analysis, showed that the system behaves lineary with a suggestion of CPU-bound state if the database is in the same machine as the analysis job. In the client and server configuration, a similar linearity has been obtained, with a small amount of CPU cycles (~18%) in idle state. A bottleneck or overhead in the database transactions over the network is suggested, and the exact mechanism of the overhead will be investigated in further measurements.

Job execution time summarized in Fig. 6 and in Table 1 are the sums of various internal response times, such as raw disk I/O speed, network speed, and the CPU cycles required to process the Objectivity transaction. Depending on the available power of the resources and the varying the loads of the system with multiple jobs, the state of the system changes from one bound state to another, where the waiting time of one type of the resource (CPU, disk I/O or network) is dominant over the other types of the resources. Under some simple assumptions, it is possible to extract the response time

of each component and then compare the result with simulation.

In the world-wide distributed data analysis environemnt, the wide-area network are of particular importance. A behavior of the client and server database configuration depends heavily on the performance and the profiles of the network. A performance test with different characteristics of the wide-area networks is being planned.

A discussion between the Simulation Working Group has been started to define the measurements of model parameters. A small set of parameters will be chosen to confirm the validity of the simulation model, such as the parameterization of AMS and NETWORK. Testbed WG will validate the simulation tool with several iterations of modeling and measurements. To make a prediction of the future powerful CPU and Netowork configurations, measurements on the various types of hardware configurations will be necessary.

References

[1] The WWW Home Page for the MONARC Project
http://www.cern.ch/MONARC/

[2] Concurrent Modelling and Simulation in Java
http://home.cern.ch/~cil/

[3] M. Boschini *et al.*, Preliminary Objectivity tests for MONARC project on a local federated database: http://www.cern.ch/MONARC/docs/monarc_docs/99-04.ps

[4] T. Smith, Private communication.

[5] SPEC: Standard Performance Evaluation Corporation
http://www.spec.org/
http://www.spec.org/osg/cpu95/results/cint95.html
http://www.spec.org/osg/cpu95/results/res98q4/cpu95-981109-03155.html

[6] R.D. Schaffer, ATLAS 1 TB Milestone
http://home.cern.ch/s/schaffer/www/slides/db-meeting-170399-new/

[7] M. Schaller, Benchmarks: Raw Data Event Model
http://home.cern.ch/~schaller/Notes.RawDataEventModel.ps

[8] K. Holtman: CPU requirements for 100 MB/s writing with Objectivity:
http://home.cern.ch/~kholtman/monarc/cpureqs.html