KEK Preprint 2000-111
October 2000
D

# Validation of the MONARC Simulation Tools

Y. MORITA (KEK)

for the MONARC Collaboration

*To be appeared in the Proceedings of the International Conference on Computing in High Energy Physics 2000, Padova, Italy, February 7-11, 2000. Submitted to Computer Physics Communication.*

**High Energy Accelerator Research Organization (KEK)**

KEK Reports are available from:

Information Resources Division
High Energy Accelerator Research Organization (KEK)
1-1 Oho, Tsukuba-shi
Ibaraki-ken, 305-0801
JAPAN

Phone: +81-298-64-5137
Fax: +81-298-64-4604
E-mail: adm-jouhoushiryou1@ccgemail.kek.jp
Internet: http://www.kek.jp

# Validation of the MONARC Simulation Tools

The MONARC Collaboration

M. Aderholz[t] K. Amako[q] E. Auge[r] G. Bagliesi[y] L. Barone[aa]
G. Battistoni[u] M. Boschini[u] A. Brunengo[m] J.J. Bunn[h]
J. Butler[ℓ] M. Campanella[u] P. Capiluppi[f] M. D'Amato[d]
M. Dameri[m] A. di Mattia[aa] A. Dorokhov[i] U. Gasparini[w]
F. Gagliardi[i] I. Gaines[ℓ] P. Galvez[g] A. Ghiselli[j] J. Gordon[z]
C. Grandi[f] F. Harris[v] K. Holtman[h] V. Karimaki[o] Y. Karita[q]
J. Klem[o] I. Legrand[h] M. Leltchouk[k] D. Linglin[s] P. Lubrano[x]
L. Luminari[aa] M. Michelotto[w] I. McArthur[v] Y. Morita[q]
A. Nazarenko[ac] H. Newman[g] V. O'Dell[ℓ] S.W. O'Neale[e]
B. Osculati[m] M. Pepe[x] L. Perini[u] J. Pinfold[b] R. Pordes[ℓ]
F. Prelz[u] A. Putzer[n] S. Resconi[u] L. Robertson[i] S. Rolli[ac]
T. Sasaki[q] H. Sato[q] L. Servoli[x] R.D. Schaffer[r] T. Schalk[c]
M. Sgaravatto[w] J. Shiers[i] L. Silvestris[d] G.P. Siroli[f] K. Sliwa[ac]
T. Smith[i] R. Somigliana[ac] C. Stanescu[ab] H. Stockinger[i]
D. Ugolotti[f] E. Valente[p] C. Vistoli[j] I. Willers[i] R. Wilkinson[g]
D.O. Williams[i]

[b]*Alberta*
[c]*BaBar*
[d]*Bari/INFN*
[e]*Birmingham*
[f]*Bologna/INFN*
[g]*Caltech*
[h]*Caltech/CERN*
[i]*CERN*
[j]*CNAF/INFN*
[k]*Columbia*
[ℓ]*FNAL*
[m]*Genova/INFN*
[n]*Heidelberg*

*Preprint submitted to Elsevier Preprint* *25 October 2000*

[o] *Helsinki*
[p] *INFN*
[q] *KEK*
[r] *LAL-Orsay/IN2P3*
[s] *Lyon CC/IN2P3*
[t] *MPI*
[u] *Milano/INFN*
[v] *Oxford*
[w] *Padova/INFN*
[x] *Perugia/INFN*
[y] *Pisa/INFN*
[z] *RAL*
[aa] *Roma1/INFN*
[ab] *Roma3/INFN*
[ac] *Tufts*

## Abstract

The objective of the MONARC project is to identify baseline computing models that could provide viable solutions meeting the data analysis needs of the LHC experiments. A powerful and flexible set of simulation tools has been developed to model the performance of distributed computing resources for a set of reconstruction and analysis tasks. In this paper we report the validation of the simulation tools using the testbed environments with Objectivity/DB over LAN and WAN connections. A simple and effective way to parameterize and evaluate the concurrent database access over network has been established.

*Key words:* MONARC; Regional Center; Performance Simulation; Objectivity; ODBMS

## 1 Introduction

The LHC experiments at CERN [1] face stringent performance demands on the data analysis at an unprecedented scale. Resources such as CPU, storage and network bandwidth have to meet the requirements of the vast amount of data analysis performed by physicists distributed world-wide. The primary goal of the MONARC project [2] is to identify baseline computing models that could provide viable and cost-effective solutions for the LHC experiments [3]. A powerful and flexible set of simulation tools has been developed to model and evaluate the performance of distributed computing resources for the baseline computing models.

The evaluation of the computing and networking systems involves the iteration of system measurements, modeling of the system behavior, development of the simulation tools and the validation of the simulation technique [4]. By iterating this evaluation cycle, one can predict the behavior of the system with the required accuracy for various types of activities. Validation of the MONARC simulation tools is therefore closely related to the required "level of detail" as the project aims for improved accuracy with greater detail in the system modeling.

## 2 Simulation Model and Technique

The technique used by the MONARC simulation tools developed by Legrand *et al.* [5] is based on "process oriented discrete event simulation". It provides a dedicated scheduling mechanism that is based on semaphores for the "Active Objects", which represent the concurrently running jobs and the traffic of data in the system. Tasks of active objects are simulated on an "interrupt" driven mechanism implemented in the simulation engine (Fig. 1). Shared resources, such as CPU, I/O links, network bandwidth of LAN and WAN, are represented as normal objects with a mutual exclusion mechanism to allow the simulation of accessing atomic parts of the system operation.

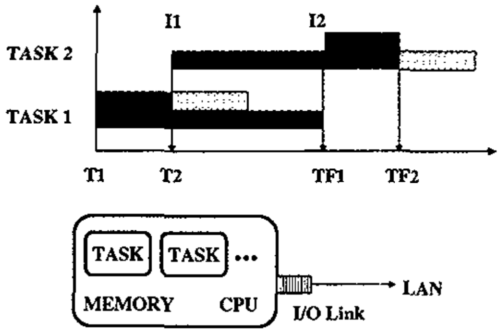

Fig. 1. Modeling of the multitask processing based on an interrupt scheme.

When a first job (Task 1) starts, the time it takes is evaluated ("original" TF1), and this "Active" object enters into a wait state for this amount of time unless it is interrupted. If a new job (Task 2) starts on the same resource, it will cause an interrupt to the first task. Both tasks now share the same CPU resource and the time to complete for each of them is re-computed (new

TF1 and "original" TF2), assuming the tasks will consume the CPU resource equally. Then the jobs will enter into a wait state and listen to other interrupts. When the first job is finished, it creates another interrupt to the remaining job(s) and the time to finish is re-computed (new TF2).

The access to the experiment data is modeled based on an existing client-server ODBMS architecture, Objectivity/DB. The database architecture is modeled using parameters such as database page sizes and the network time response between clients and servers. The clustering of the objects in different containers can be modeled and studied using the tool.

## 3 Testbed Measurements

To validate the performance simulation of distributed data analysis using the ODBMS in LAN and WAN environments, the response time functions of the following components are crucial and needed to be evaluated:

- the sharing of the CPU and I/O resource
- the queuing mechanism
- the performance of the ODBMS with network
- the sharing of the network bandwidth

The queuing mechanism in the tool has been compared with the queuing model by Dorokhov *et al.* [5], and good agreement has been obtained. To evaluate other key components, several testbed environments have been set up at various laboratories, such as CERN and INFN in Europe, KEK in Japan, Caltech, and SLAC in the United States. These sites are connected with several types of wide area network, such as satellite ATM PVC circuit with dedicated bandwidth, and dedicated QoS network services. Several example HEP analysis applications, which utilize Objectivity/DB, have been developed and tested over these environments.

### *3.1 Local and LAN Concurrent ODBMS Access*

To understand the behavior of an Objectivity AMS server as an example model of the distributed ODBMS, the following SUN Solaris 2.6 machines were used [6].

- **machine A:** SUN Enterprise 450 (400MHz x 4 CPUs), 512 MB, two 9 GB disks
- **machine B:** SUN Ultra5 (270MHz CPU), 192 MB memory (Objectivity/DB lock server)

- **machine C:** SUN Enterprise450 (300 MHz dual CPU), 512 MB memory, RAID Disks

*All 3 machines were running SUN Solaris 2.6 with C++ v.4.2 and* Objectivity/DB v.5.1.

Monte Carlo simulated ATLAS raw data was converted into Objectivity/DB database format. An event was stored in an event container as a persistent vector of pointers referring to the list of sub-detectors, which were stored in a single separate container. Each subdetector holds a vector of hit objects which contain energies and hit coordinates. To evaluate the performance of concurrent database read access over the network, a simple C++ program was created. It reads through every hit object from the generated events by using an event iterator and following the associations of sub-detectors therein. Multiple jobs were run concurrently on the system with three configurations:

(1) Local file database access on one machine (machine A)
(2) Local file database access on another machine (machine C)
(3) a pair of machines acting as client and server of Objectivity/DB AMS.

The execution time and CPU utilization of each job were measured using the UNIX *time* command. The local disk I/O speed was measured by a simple C program using *read()* and *write()* system calls. The network speed between the two machines was measured using FTP transfers. The results of the measurements of the database accesses are given in Table 1. These are compared to modeling results in Section 4 of this paper.

| | machine A local | | | machine C local | | | machine A and C | | |
|---|---|---|---|---|---|---|---|---|---|
| jobs | time ($s$) | rms | CPU % | time ($s$) | rms | CPU % | time ($s$) | rms | CPU % |
| 1 | 14.23 | – | 99.1 | 19.93 | – | 94.5 | 22.08 | – | 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. Average job execution time and aggregated CPU utilization

### *3.2 ODBMS Access over WAN*

The packet-level behavior of an Objectivity AMS client/server system has been measured with a dedicated 2Mbps satellite link as well as general purpose 2 Mbps and 4 Mbps academic network links between CERN and Japan [7]. The relatively large round trip times (660 msec for satellite link, 280 msec for surface link) enabled the detail observation of the packet behavior.

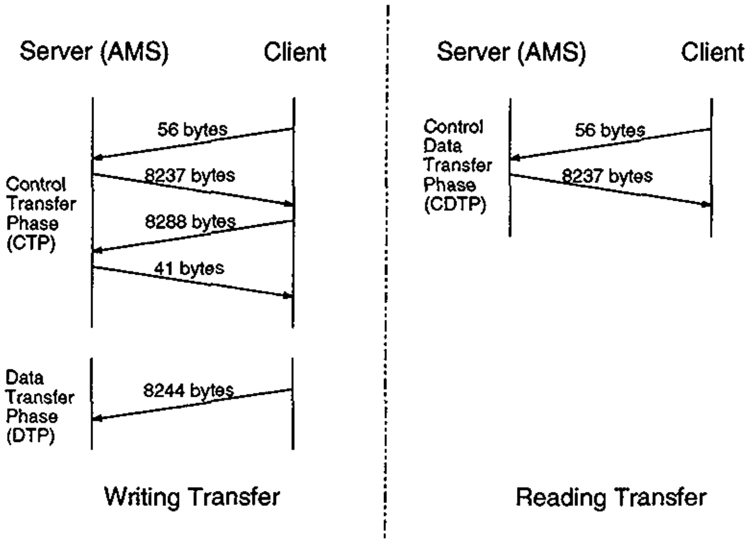

Fig. 2. Packet transfer sequences between AMS server and client

Figure 2 shows the packet transfer sequences between the AMS client and server for reading and writing operations. In the reading operation, a handshaking of unit page size at the application layer of the protocol makes the transfer rate inefficient for the large latency networks. In the writing operation, several handshakes of control phase appear at the beginning of each transaction, but a simple data transfer phase dominates for a bulk transfer operation. In this case the TCP/IP congestion window size becomes the crucial parameter for good performance (Fig. 3). Proposals have been made to Objectivity that for a reading operation to either increase the size of the handshaking unit paging or to stop the handshaking at the application layer for large latency networks. The AMS writing speed outperforms the default implementation of FTP client and server on Solaris.

Another set of measurements was performed on a QoS network using various links speeds between the AMS server and clients [8]. Monte Carlo generated data is analyzed and the analysis object data such as the reconstructed muon momentum are stored into database files with a program ATLFAST++. In this program, one single container is used to store all data and there are no associations among the objects. By filtering the values stored into the database, system performance of concurrent read access with various different values of

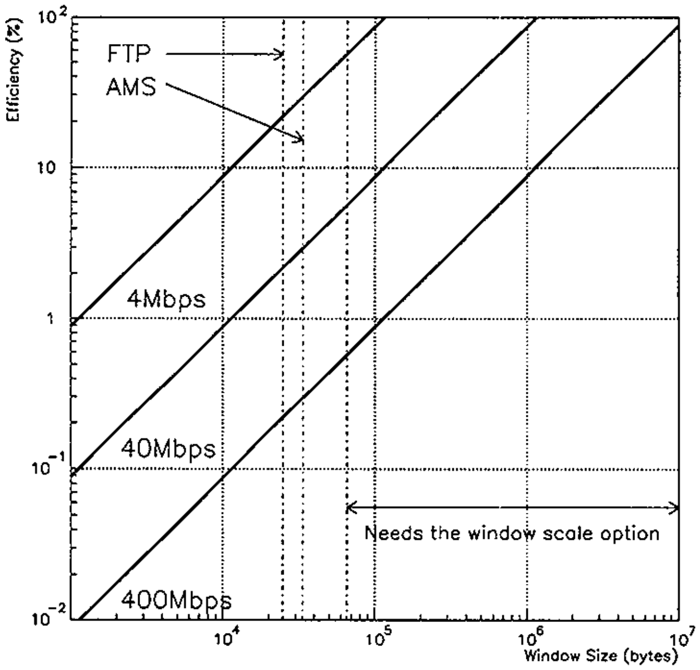

Fig. 3. Efficiencies of bulk data transfer vs. congestion window size at round trip time 280 msec for various network bandwidths

muon momentum has been measured [9]. With this application program, it was noted that the balance between the CPU power and the network bandwidth becomes important both on the server and client sides, and the combination of clients with fast and slow network links brings a bottleneck related to the slowest client in the AMS server.

## 4 Comparison with Simulation

To simulate the behavior of a given testbed application, the following parameters must be provided to the simulation: the CPU cycles per event, the CPU SI95 rating, the size of event data, disk I/O speed, and the network speed.

By making a simple assumption, the estimates of the parameters from the testbed measurements have been made. They are extracted from a single job accessing a local file database (cases (1) and (2) in Section 3.1).

(1) Machine A local access

$$T_{job}(A) = 14.23 \text{ sec} \quad ; \quad CPU_{Util}(A) = 99.1\%$$

$$Disk_{rate}(A) = 207 \text{ MB/s} \quad ; \quad CPU_{Power}(A) = 17.4 \text{ SI95}$$

(2) Machine C local access

$$T_{job}(C) = 19.93 \text{ sec} \quad ; \quad CPU_{Util}(C) = 94.5\%$$

$$Disk_{rate}(C) = 31 \text{ MB/s} \quad ; \quad CPU_{Power}(C) = 13.05 \text{ SI95}$$

Assuming

$$T_{job} = T_{diskread} + T_{process}$$

$$\text{and} \quad \frac{T_{process}(A)}{T_{process}(C)} = \frac{CPU_{Power}(C)}{CPU_{Power}(A)} = \frac{13.05}{17.4}$$

$$\text{and} \quad \frac{T_{diskread}(A)}{T_{diskread}(C)} = \frac{Disk_{rate}(C)}{Disk_{rate}(A)} = \frac{31}{207},$$

then

$$T_{process}(A) = 14.06 \text{ sec}$$

$$T_{process}(C) = 18.74 \text{ sec.}$$

By using this set of single job parameters, the simulation tool reproduces the multiple job configurations for the local database access, as shown in Fig. 4. This is an indication that the simulation tool handles the concurrent access of CPU and database file properly. The machines A and C are 4- and 2-CPU SMP machines respectively. However, the simulation model is constructed with 4 and 2 single CPU nodes connected with a fast network, and the simulated results reproduces the measurements.

For the configuration (3), a client/server pair of machines, we monitored the TCP/IP packets between the AMS client and server. The handshaking at the application layer which was observed in section 3.2 was modeled into the network response time in the simulation tool, and the simulated results reproduce the measured values satisfactorily ("AMS" in Fig. 4).

Although the simulation parameters are not tuned for individual jobs, the qualitative behavior of the simulated job execution time is similar to the measurements (Fig. 5). This agreement indicates that the time sharing of the CPU resources with multiple jobs as well as the database I/O queuing in the system is well modeled and simulated.

The testbed measurements over the QoS network were also reproduced with

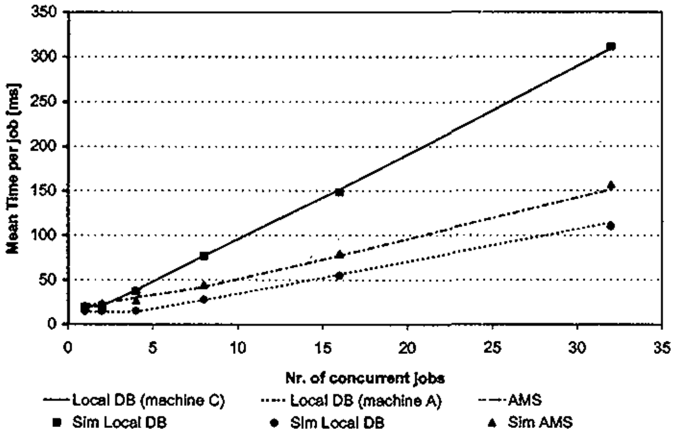

Fig. 4. Simulated and measured job execution time in Objectivity client and server configuration

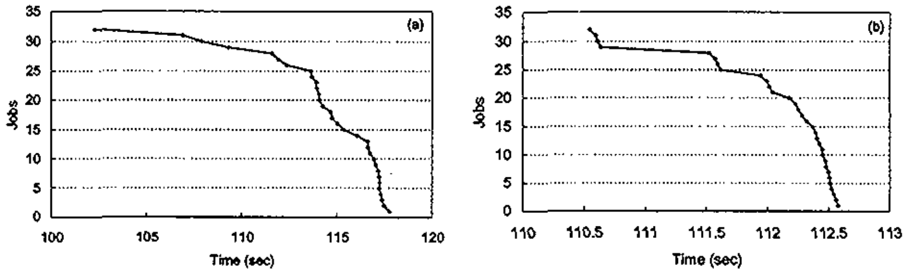

Fig. 5. Measured (a) and simulated (b) job execution time competing for the same resource

the simulation (Fig. 6) [10], which reassures that the network transaction in the simulation model can describe the real measurements with the parameters of bandwidth and latency, regardless of whether the connections are LAN or WAN.

## 5 Conclusions

The MONARC project has developed a powerful set of simulation tools to evaluate the performance of complex computing and network systems, which consist of CPU farms, database servers and local and wide area networks. A Java based simulation program, using a process oriented discrete event simulation technique, reproduces the predictions of analytical queuing models and a set of testbed measurements.

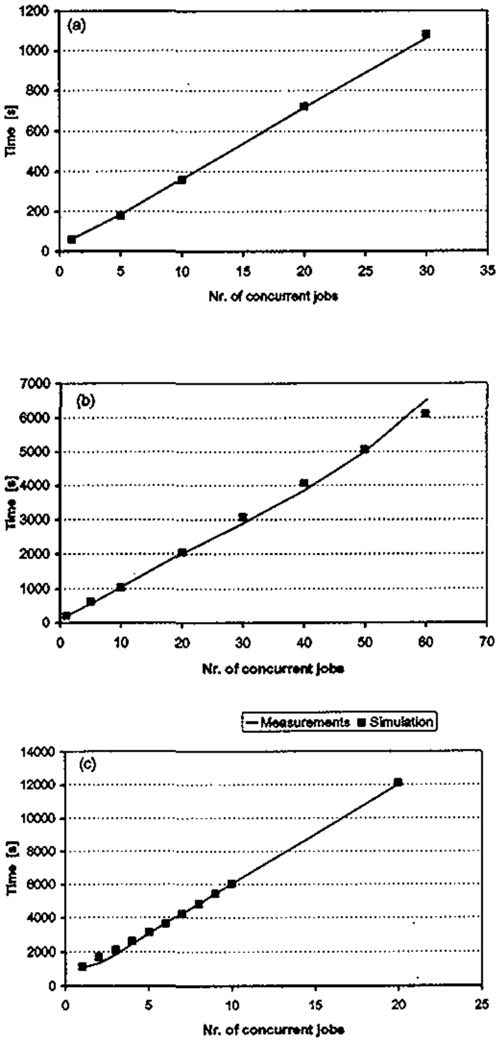

Fig. 6. Average execution time of concurrent number of jobs for different network configurations – (a) 1000BaseT, (b) 10BaseT, (c) 2 Mbps WAN

It has been demonstrated that a complex set of transactions of local ODBMS access can simply be modeled into a sum of time responses of the CPU processing and the disk I/O processing. A set of job profile parameters reproduces the testbed measurements for multiple concurrent jobs. For the client and server configuration, the behavior of the protocol was monitored at the TCP/IP packet level and the handshaking model at the network application layer well reproduces the testbed measurements.

However, it is important to understand that the evaluation of the system performance is a continuous cycle of refining the modeling, testing and validation. To make a reliable prediction of the system performance at the startup time of LHC era, careful considerations of the future software performance tunings as well as hardware technology tracking are necessary. An effort of simulating baseline LHC computing models has already commenced in the MONARC project [3].

Evaluation of the hierarchical mass storage system requires a set of performance measurements with a combination of ODBMS use cases. How to achieve an effective use of ODBMS on tape robotics with various possible object reclustering strategies based on the real HEP data analysis use cases, will be one of the major focus for the next phase of the project [11][12].

Effective use of the wide area network is also one of the most crucial aspect in the LHC era computing. Relatively large latency (several hundreds of milliseconds in round trip time), typically seen in world-wide WAN and in satellite connections, combined with a high bandwidth (several hundreds of Mbps to a few Gbps) foreseen for the future, imposes a serious efficiency problem for the current implementation of TCP/IP applications. An improved slow-start algorithm for quickly enlarging the congestion window size has appeared in the literature [13], which is needed to be tested in the HEP data analysis environments. Also several efforts are now underway to implement better implementations of file transfer applications, such as *gsiftp* with an ability to control the congestion window size. All components of the software of world-wide computing in the LHC era should accommodate to the high latency network, ranging from the TCP/IP layer to the application layer. The MONARC project should continue to test and validate the models of networked HEP data analysis with the new generations of software components.

## Acknowledgements

MONARC is pleased to acknowledge the strong support and collaboration of the CERN/IT Division in this project, especially the PDP group and for its help and support for the testbed and simulation systems, and the CS Group for its efficient management and operation of CERN's network links to Europe, Japan and the US which were important for our tests over wide area networks. We gratefully acknowledge the support of our respective funding agencies: IN2P3 in France, INFN in Italy, MONBUSHO (Ministry of Education, Science, Sports and Culture) and YUSEISHO (Ministry of Posts and Telecommunications) in Japan, PPARC in UK, and the US Department of Energy and National Science Foundation.

## References

[1] LHC Project: http://www.cern.ch/LHC/

[2] "Models of Networkd Analysis at Regional Centres for LHC Experiments: Phase 2 Report", CERN/LCB 2000-001, also in http://www.cern.ch/MONARC/

[3] I. Gaines, "Modelling LHC Regional Computing Centers with the MONARC Simulation Tools", submitted to this conference.

[4] B. R. Haverkort, "Performance of Computer Communication Systems", John Wiley & Sons Ltd.

[5] I. Legrand, "Multi-threaded, discrete event, simulation of distributed computing systems", submitted to this conference.

[6] K. Amako et al., "MONARC testbed and a preliminary measurement on Objectivity AMS server", MONARC-99/7 in:
http://www.cern.ch/MONARC/docs/monarc_docs.html

[7] H. Sato and Y. Morita, "Evaluation of Objectivity/AMS on the Wide Area Network", submitted to this conference.

[8] A. Brunengo et al., " LAN and WAN tests with Objectivity 5.1", MONARC-99/6 in:
http://www.cern.ch/MONARC/docs/monarc_docs.html

[9] M. Boschini et al., "Preliminary Objectivity tests for MONARC project on a local federated database", MONARC-99/4 in:
http://www.cern.ch/MONARC/docs/monarc_docs.html

[10] I. Legrand, LHC Computing Board Marseilles Workshop, 1999 September
http://lcb99.in2p3.fr/ILegrand/Slide1.html

[11] K. Holtman, "Prototyping of CMS Storage Management", Ph.D. thesis (proefontwerp), May 2000, Eindhoven University of Technology, ISBN 90-386-0771-7

[12] MONARC Phase 3 Letter of Intent:
http://www.cern.ch/MONARC/docs/phase3_loi.doc

[13] Y. Nishida, J. Murai, Computer Software, Vol. 16, No. 4 (1999) 33.