# HUGE MEMORY SYSTEMS FOR DATA-INTENSIVE SCIENCE

R. P. Mount, Stanford Linear Accelerator Center, Menlo Park, CA 94025, USA

*Abstract*

We propose a revolutionary leap forward in the application of information technology to the many important basic science fields which are data-intensive – high-energy physics, biology, nuclear physics, combustion simulation, and astronomy/cosmology. In these fields we expect our proposed approach to transform the interaction between scientists and their data, and in doing so, transform the very science being performed.

Our goal for ~2008 is a new "huge-memory" architecture inserting a quarter-petabyte of solid-state memory between disk storage and the high-performance memory serving the processors. This architecture will remove the increasingly crippling effects of the factor $10^5$ latency mismatch between disks and high-performance memory.

In the first three years we will build a relatively modest 'demonstration and development' system designed to cache the BaBar micro DST that currently has a size of ten terabytes.

## INTRODUCTION: A REVOLUTIONARY ADVANCE IN DATA ANALYSIS

We address a key need of those basic sciences that face the overwhelming challenges of analyzing growing volumes of experimental, observational or simulated data. Specifically, we propose a new architecture that will improve the response time for the many (and growing) data-intensive areas of basic science by up to two orders of magnitude, and will remove the scientific paralysis that will otherwise occur as analysis times for high-volume data stretch out to years.

A recent DOE Office of Science Data Management Workshop [1] brought together a wide spectrum of activities, many of which are facing current or near-future challenges of petabytes of data. Storing and accessing petabytes of bulk data with streaming, often parallel, I/O is non-trivial, but is made possible by the onward march of storage capacities. A much greater challenge is the analysis of petabyte-scale datasets with more complex structure to which individual scientists require sparse or even random access. Such datasets or databases are not usually the bulk data streamed from a detector, telescope or supercomputer, but rather the feature databases abstracted from the bulk data and acting either as the index to the bulk information or as stand-alone targets for analysis.

Specific science challenges and opportunities are well-illustrated by presentations at the data management workshop:

- Biology is becoming increasingly challenged by "huge data volumes, great schema complexity [and the] need for new types of databases (hardware and software)," [2] For example, proteomics data that amounts to scarcely 10 terabytes today, but will exceed a petabyte by 2005 and grow to hundreds of petabytes early in the next decade.
- Nuclear physics already faces petabyte data volumes at the RHIC collider and much greater volumes at the LHC. Studies of the quark-gluon plasma tend to require statistical rather than "needle-in-haystack" analyses. This helps to explain why about half of the RHIC data volume is the so-called "summary data" used intensely for physics analysis. Access to data is recognized as a major problem: "the extraction and the delivery of distilled data subsets to physicists for analysis currently the principal limitation on NP analyses." [3]
- Combustion simulation is currently limited to two-dimensional models generating about 2 terabytes of raw and 2 terabytes of processed data per run. Within 5 years three-dimensional models will generate about 3+3 petabytes of raw+processed data. "Feature-borne analysis and redundant subsetting of data for storage" are now the normal approach. In the future, this will allow scientists to "find small ROI's in a large 3D domain" and "retrieve and analyze only what [they] need." [4]
- Current high-energy physics experiments have petabyte databases of processed "feature" information. The physics analysis may be characterized as thousands of physicist-initiated tasks requiring sparse access to small (100 bytes to 10 kbytes) objects. Today's petabyte databases will become orders of magnitude larger in the high-energy physics experiments of 2010. Already, glacial sparse access places major inhibition on the use of intuition and unplanned (i.e. new) ideas in physics analysis. To avoid being crippled by data access, the physicists devote great energy, many months of elapsed time and data-manipulation hardware costing millions of dollars to organizing the data for reasonable performance for mainstream analysis patterns.
- Jim Gray drew on his own experience in addressing issues in data access to astronomical data sets: "You can *grep* 1 megabyte in a second" or "you can *grep* 1 petabyte in 3 years" by streaming data to a scientist's workstation, If you "want ~1 minute response" you could do it by

"brute force" by spending $300M on disks to grep in parallel[5].

Obviously, no scientist would be naïve enough to search serially through a petabyte, but as soon as more intelligent query approaches are used, the abysmal random-access performance of disks brings rapid disappointment.

The technical problem to be addressed by a new architecture can be summarized as "disks are no longer 'effective' random-access devices". This is illustrated in Figure 1. Disks have a mechanical/rotational latency of about 10 milliseconds. This has remained essentially unchanged for a decade whereas processors have increased in speed by close to three orders of magnitude. The predictions for the next decade closely mirror history. Processors will get faster, disks will have more capacity, but the number of sparse or random retrievals per second will not increase.

In its simplest form, our proposed approach to solving the technical problem is almost obvious: all data for which 10ms latency is unacceptable must reside in memory. This immediately improves the latency by a factor of over 1000 and "solves" the problem. There remains the issue that, very approximately, memory costs 100 times as much as disk – a problem whose partial solution is to note that the memory performance required to beat disk by a factor of 1000 in latency and 10-100 in bandwidth is quite modest. The problem can be addressed by a cost-optimized architecture that inserts a layer of the cheapest reliable memory between the disk storage and the processors equipped with high-performance memory.

## DEVELOPMENT SYSTEM DEPLOYMENT PLAN

### *Design Principles*

The design principles for the development system are:

- It should be attractive to scientists who will be motivated to exploit its new capabilities to revolutionize their work and explore and demonstrate its capabilities.
- It should have 1000 or more processors. This is similar in size to the current BaBar data analysis system at SLAC.
- It should support access from any analysis processor in the system to any memory-resident data within the system with a latency of less than 100 microseconds. This is distinctly unaggressive performance by memory-access standards, but beats disk latencies by a factor of 100.
- The total bandwidth for processor to memory-resident data access should be at least 10 times higher than that for streaming access to a similar quantity of disk-resident data.
- The system should be able to offer memory cache sizes in the range 3 to 10% of the size of the disk-resident working set for an access-challenged community. The initial target community is the BaBar Collaboration. By late 2004, BaBar will have a total data volume of around 1.5 petabytes with a disk-based working set of about 200 terabytes. A system with 10 terabytes of memory cache would offer ~5% caching to the full working set for BaBar analysis activities. More specifically this is the current size of the micro-DST real data

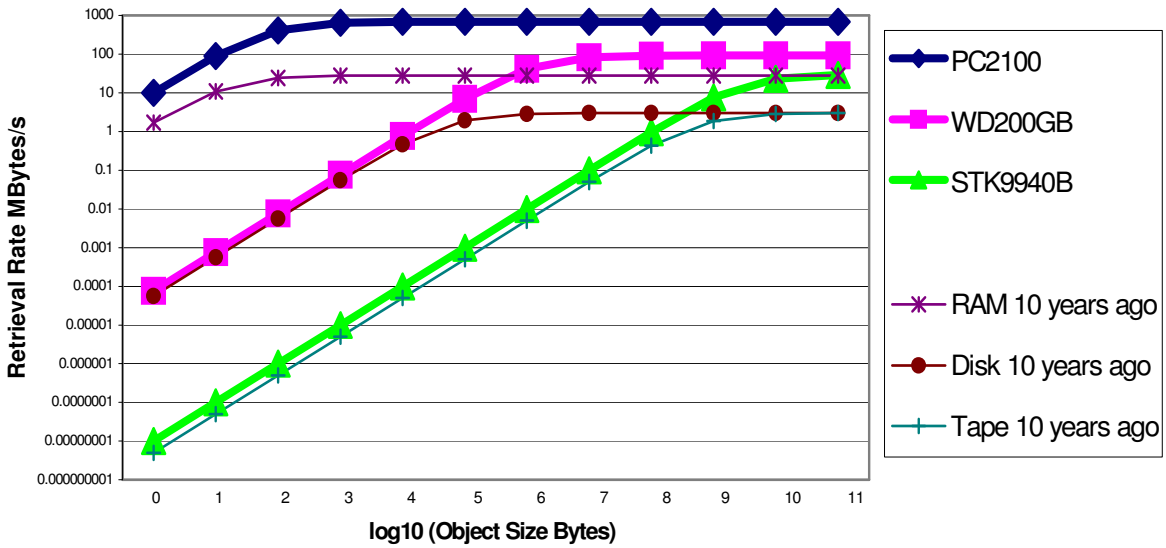


Figure 1**:** The effects of latency on retrieval rate for memory (PC2100), disk (WD200GB) and tape (STK9940B). The curves show the average data rate for random-access to objects ranging in size from one byte to 10 gigabytes. Typical performance of devices 10-years ago is also shown.

events which are the target of the most intense data-access

- The system should be as cost-effective as possible, keeping in mind that achieving a reasonable level of reliability and fault detection/recovery is required for cost-effectiveness.

### *Design Choices*

The principles lead to the following practical design choices:

- The system should use commodity processors such as (Intel/AMD) with server-class mainboards that can accommodate as much error correcting (ECC) memory per processor as possible. Currently, dual-processor mainboards accommodate more memory per processor than quad-processor boards. Server mainboards for two processors with 8 memory (dimm) slots and two gigabit Ethernet ports are available from many sources. Blade servers with 8 processors and 8 dimm slots per processor will be attractive if they become available in 2005.
- Cost effectiveness currently dictates the use of 2-gigabyte dimms (under $1000) each. Four-gigabyte dimms have been announced, intially at $7000 each. This price is expected to decline rapidly.
- The operating system used on the system nodes should be able to address the memory sizes – at least 32 gigabytes – that will be achieved during the life of the system. At present this makes 64-bit Solaris running on AMD Opteron processors an attractive choice.
- The system should use a cost-effective switch fabric of adequate total any-to-any capacity for processor-memory access. Unlike a cluster that tries to emulate a tightly coupled supercomputer, there is no imperative to achieve the low latencies associated with technologies such as Myrinet, and a large Ethernet switch becomes a serious candidate if it provides more total any-to-any throughput.
- Since all candidate nodes have two gigabit Ethernet interfaces, a separate switch fabric can be used to allow the system to access external disk servers.
- In the first year, the system will have access to the high-performance FibreChannel disk servers installed for the BaBar experiment. In the future, serial ATA drives are expected to be a good choice disks from which the memory-resident data cache will be loaded.

### *Initial Deployment*

The year-1 system was proposed to be 650 dual-Opteron nodes, each with 16 gigabytes of ECC memory to reach a total system memory of 10 terabytes. The initial funding will allow 20% of this system to be constructed and exhaustively tested. Each node will run an instance of the 64-bit version of Solaris x86. Additional funding will be sought in 2005 to expand the system to at least 10 terabytes, probably moving all the memory into a new system based on 8-way blade servers.

The interconnect will be centred on an Ethernet switch, initially a Cisco 6509, supporting 10/100/1000 Mbit ports. The 10-terabyte machine will require a highrt throughput switching core or at least 1.6 terabit/s

A simplified diagram of the 2005 configuration of the system and its interconnection with SLAC-BaBar storage is shown in Figure 2.

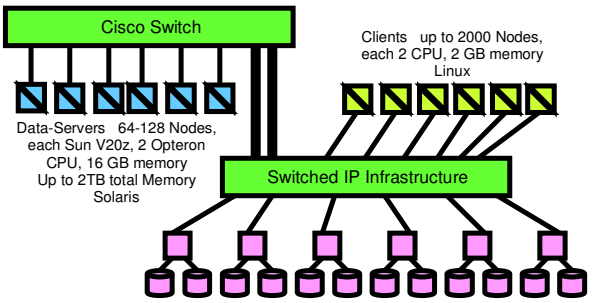


Figure 2: Initial configuration of the memory-based data-analysis system

## DEVELOPMENT SYSTEM EXPLOITATION PLAN

BaBar's current analysis model, which is not optimized to exploit a huge-memory architecture, would already become more productive by being deployed on the 10-terabyte development machine. Typically many physicists work concurrently on their analysis, and thus each disk used to serve the analysis farm is completely busy answering the various unrelated requests. This is exacerbated by the fact that each physicist must submit hundreds or even thousands of jobs to exploit the trivial parallelism of high-energy physics data analysis.

However, BaBar's data analysis approach has been developed to survive with existing disk storage, and does not currently offer access to individual objects without retrieving the complete event. To motivate the work needed to unleash truly sparse access to objects, it will be essential to guarantee that the full micro-DST will always be memory resident.

Today's technology can be applied to begin to explore this revolution. BaBar uses a purpose-built daemon to serve analysis data (xrootd [6]). The analysis system contains load-balancing instances of the xrootd that diverts clients to the machine actually holding the data (or one willing to serve it if it is not already on disk – it can automatically stage data from tape if needed). With a large compute facility that contains machines with large local memory, an instance of xrootd could be run on each node. The memory not required for normal operation would then be mounted as a local file system. Each node

would communicate with the load balancer to describe which data it would serve to the other nodes. In the event of power failures we could either have the xrootd on each node retrieve the data they serve from disk (or tape) as it is requested by users or in a more organized mode once power has been restored (and is stable).

The initial 1-2 terabyte data-server system will be an excellent platform for the study of the immediate barriers to scaling up random access. The CPU load caused by high-packet-rate network traffic is likely to be an immediate concern to be addressed both by hardware, such as tcp-offload network cards, and software, such as tuning or improving the tcp stack within the operating system. While the xrootd technology allows the data-server task to be arbitrarily subdivided, it is likely that the optimum for cost-effective, hot-spot-insensitive data serving will lie somewhere between dual-cpu severs and large symmetric multiprocessor servers. During 2005, the option of using larger data-server machines will be tested.

The xrootd approach constitutes an existence proof that a large-memory cluster can be exploited rapidly for high-energy physics analysis. Alternative approaches, likely more relevant to the application of large memory to other sciences, include the investigation of the caching of shared file systems such as Lustre and SAMFS. A key characteristic of high-energy physics data, and almost certainly a large range of other basic scientific data, is that they are generally accessed in either "read-only" or "write-only" modes. Very simple locking mechanisms are usually viable alternatives to highly granular locks or full cache-coherency.

Figure 3 shows the initially proposed, and not yet funded, configuration of the development machine. The plan to combine data-serving and data analysis functions on each node is ambitious and perhaps dangerous. There is ample evidence from day-to-day experience at SLAC that physics analysis code can crash or paralyze the machines that run it. No operating system since IBM VM has been robust enough to be resist crashes caused by user code. While such a configuration might provide an excellent demonstration of the fault tolerance of xrootd, it might also be too ambitious. It may also remain advantageous to decouple the choice of operating system for the data-servers and the data-analysis clients. The likely alternative is a smaller number of dedicated ~8-way symmetric multiprocessor data servers connected to the switching fabric by 10 gigabit Ethernet, leaving a slightly reduced number of dual-processor data-analysis clients.

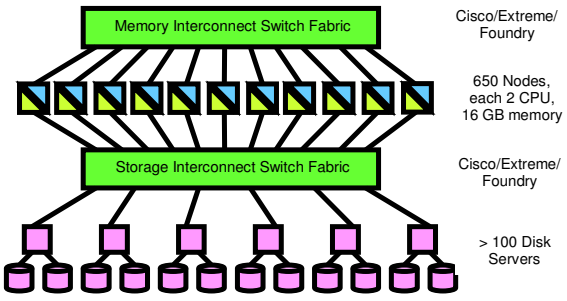


Figure 3: The initially proposed configuration for the development machine. All nodes perform both data serving and data analysis functions.

## LONGER-TERM GOALS

It seems very probable that within five years "storage-class memory" [7] will become a reality, inserting solid-state products into the latency and cost gap between memory and disk.

Our aim is to prepare the architectural and data-analysis approaches that will allow successful exploitation of machines with a petabyte of storage-class memory by the end of the decade. Even without new technology, the downward evolution of memory prices, and the increasing needs of many sciences, lead us to believe that our proposed 250 terabyte can be funded by 2008 and will yield major advances for data-intensive science.

## ACKNOWLEDGEMENTS

Many people have contributed to the ideas behind this work and to the success in obtaining initial funding. At SLAC David Leith, Andy Hanushevsky and Randy Melen have brought ideas, enthusiasm and energy. Technical discussions with Sun Microsystems have been extremely valuable.

## REFERENCES

[1] http://www-user.slac.stanford.edu/rmount/dm-workshop-04/Final-Report-Work-Area/

[2] G. Michaels, Pacific Northwest National Lab., http://www-conf.slac.stanford.edu/dmw2004/slacworkshop/talks/michaels/SLAC_GSM_3_16_04.ppt

[3] B. Gibbard, Brookhaven National Lab., http://www-conf.slac.stanford.edu/dm2004/slacworkshop/talks/gibbard/NP%20Data%20Management%20Needs.ppt

[4] J. Chen, Sandia National Lab., http://www-conf.slac.stanford.edu/dmw2004/slacworkshop/talks/chen/Data_management.ppt

[5] J. Gray, Microsoft, http://www-conf.slac.stanford.edu/dmw2004/slacworkshop/talks/gray/SLAC_Data_Management_Workshop_Gray.ppt

[6] A. Hanushevsky, "The Next Generation root File Server", presentation at this conference.

[7] J. Menon, "Grand Challenges facing Storage Systems", presentation at this conference.