

High Performance Computing and I/O Architectures for Database and Knowledge Discovery: The System Design Perspective*

Richard Lundeen
CS Program, College of Engineering
Idaho State University
Pocatello, ID, U.S.A.

Steve C. Chiu
CS Program, College of Engineering
Idaho State University
Pocatello, ID, U.S.A.

Abstract

Research in parallel database (DB) and data mining (DM) algorithms has experienced a significant growth due to advancements in high performance computing (HPC) systems. Enabling technologies such as multi-core processors, object-based storage and high-bandwidth interconnects helped propel innovations to address fast increasing demands in scientific and commercial computing. Large-scale applications involving data in the order of tera-bytes or beyond, not uncommon, require characterization of the HPC system designs to address potential performance bottlenecks. This paper will attempt to design and characterize HPC architectures with novel micro-electromechanical system (MEMS) based storage, where parallel DB and DM algorithms are utilized for inferences and knowledge discovery. A visualization system exploiting parallel message passing interface and open source libraries will be developed. Findings and multiple components from the proposed research may also be extensible to other scientific areas.

Keywords: Parallel I/O, Database, MEMS, Data Mining, Knowledge Discovery, Active Storage.

1. Introduction

In recent years, the way in which basic and applied scientific research is conducted has changed significantly as simulation and computer modeling are increasingly being used to augment, and frequently replace, physical experimentation. Complex systems can now be visualized, analyzed and better understood through computational models. For many fields, including astrophysics, climate modeling and seismic studies, computational simulation has become the primary method for exploring and verifying theories. Sophisticated designs such as aircrafts and advanced pharmaceuticals can be synthesized and

better produced by way of computational optimization and computer-based testing and experimentation. With the huge amounts of data that is often collected in these applications, it is not feasible to analyze the results using more traditional methods. Computation has, and will continue, to play a critical role, not only in addressing advanced scientific and engineering problems, but also in helping solve commercial, social and environmental issues. In particular, we observed that the progress in science and engineering depends to a large extent on the access to and utilization of computing resources.

In recent years, it has become more expensive to improve upon existing hardware. As the price of computers to the user continually falls, it becomes increasingly expensive for producers to supply these. For example, the cost to tapeout (the design phase at which the blueprint of a chip is sent to be produced) a chip at 0.18um was roughly \$300,000. The cost to tapeout a chip at 90nm exceeds \$750,000, and the cost is expected to exceed \$1.0M for 65nm [1]. Although it may arguably be possible for Moore's law to extend past the current technology design, it is generally accepted that given the current hardware design, there are performance limits that will be approached or reached within or around one or two generations.

Because of these restrictions, parallel computing has drastically increased in popularity over time. Many of the computationally intensive algorithms are in fact parallelizable, and almost linear speedups can be achieved by increasing the number of processors.

The evolution of HPC architectures and their associated processing paradigms have demonstrated a trend of increasing computing power in peripheral devices, coupled with larger-than-exponential growth in the size of data being processed and analyzed. To address the fast increasing computational requirements of emerging scientific and commercial applications, several enabling computing technologies are being researched. Chip-level integration of electronics with growing density makes possible on-device processors that offload user-

* This research was partially supported by Grant #964, from the Faculty Research Committee, Idaho State University, Pocatello, ID, U.S.A.

level application processing to the peripheral [2]. Given rapid advancements in device hardware such as MEMS (for micro-electromechanical system), it is conceivable that an integrated processor-memory-storage MEMS chip could soon be realized [3]. These hardware innovations, coupled with object-based storage and parallel file systems [4], represent a logical direction for future HPC architectures. Semantically smart I/O system designs that provide peripheral devices, e.g. storage controllers, with information beyond block-level scheduling protocols further offload the file manager -- often a performance bottleneck [5]. Parallel DB and DM algorithms are experiencing a significant interest by researchers for the potential performance gains they would provide with these recent hardware platform innovations. This interest in DB/DM is made all the more apparent by the demands in scientific and commercial computing that often require constant analysis of large data sets produced at short intervals [6][7].

In this paper, we study the design of HPC and I/O architectures both for database and data mining (a.k.a. knowledge discovery) applications from a systems design perspective. Section 2 reviews the background work in the related areas. Section 3 presents our design methodology, including the system setup. Section 4 discusses the architectural and device-level models used in our investigation. In Section 5, we specify the simulation platform to be used for this work. Finally, conclusions and future work are discussed in Section 6.

2. Background

2.1. Scientific Simulation Cycle

Figure 4 depicts the characteristics of a scientific simulation cycle composed primarily of 5 stages: domain decomposition, simulation, data analysis, visualization, and parameter adjustment. The cycle is iterated until the data analysis and visualization determine that no additional parameter adjustment is necessary, based on certain application-specific criteria. Such scientific applications can be compute-intensive, data-intensive, or both. Because of enormous sizes of the data in the computations involved, scientists often rely on inferences drawn from the data analysis stage, along with 3-D images or animations rendered in the visualization stage, to determine if the results of the simulation cycle have reached the desired precision or outcome. With an automated on-line processing model, where DM algorithms can be embedded within the simulation stage, the data analysis and visualization stages play a critical role in the simulation cycle [8].

Examples of images produced during visualization stage include those shown in Figure 5 and Figure 6. In

Figure 5, data sets produced by the analysis stage (with parallel data clustering DM algorithms) from a cosmology simulation using ENZO [9] are displayed to represent the formation of the galaxies from beginning of the universe to the present day, given 491520 particles in this case. Figure 6 illustrates the visualization of the multi-scale algorithm named AMR (for Adaptive Mesh Refinement, used in ENZO), in which the algorithm commences with a coarse uniform grid covering the entire domain, and will continue to refine certain regions by adding finer sub-grids, where higher computational needs, i.e. higher resolutions, exist. Such parallel DM algorithms are widely used in several scientific and engineering research areas, including physics, astronomy, fluid and structural dynamics, magnetism, and so forth. Data clustering algorithms, also a class of parallel DM algorithms, help scientists discover interesting patterns (a.k.a. data clusters) from large data sets. These interesting patterns often exist in high-dimensional data space, and efficient data clustering algorithms need to address both data and noise that exist in high-dimensional spaces and sub-spaces. It is important that these algorithms address scalability, data distribution, understanding of end-results, and sensitivity to input order. The parallel adaptive-grid and density-based algorithm named pMAFIA [10] represents such algorithms. Some major applications that exploit these algorithms include image processing, seismic studies, and organization of document data sets, among others.

2.2. Parallel and Distributed Databases

Parallel and distributed databases, having long been deployed in large corporations for managing inventories, pay rolls, and other operating resources, are also experiencing a tremendous growth in performance and capacity. This is due to the ever-increasing demand for information, and the afore-mentioned HPC architectural innovations that supply a solution for this demand. DB algorithms and query optimization are thus receiving a renewed interest by researchers seeking to improve the DB processing power and query throughput [11]. While these are primarily commercial systems, they nevertheless share many fundamental characteristics with other scientific applications. Both scientific and commercial applications deal with the same basic types and sizes of data and frequently require DB and DM algorithms to aid in quantitative analysis and decision-making. Figure 1 illustrates the Oracle 8 parallel database that could have a scientific or a commercial function [12][14]. The work in [13] proposed an HPC processing model for the TPC-H [15] benchmark ad hoc queries used for measuring the performance of large, commercially deployed databases. Figure 7 depicts the processing model for the query

execution plans of some of the queries from the TPC-H benchmark suite.

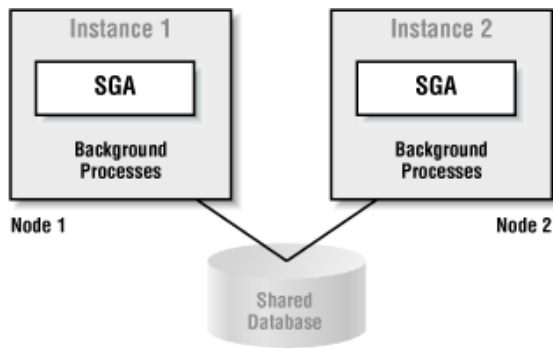


Figure 1. A parallel database model

2.3. MEMS-based I/O Storage Devices

MEMS-based I/O storage devices are micron-sized devices fabricated from the normal IC photolithographic processes [16]. These devices consist of a moving rectangular medium sled and an array of read/write probes. The medium sled is spring-mounted under the probes, and can be moved in the X, Y, and Z dimensions by way of actuators. As an I/O storage device, this system “seeks” in the X dimension and reads/writes data in the Y dimension. The Z dimension is then used to control the contact between the probes and the medium sled. Such a prototype MEMS I/O system has been developed at Carnegie Mellon University [17]. Other MEMS-based I/O devices include those being developed by Agilent Technologies [18], IBM [19], and Nanochip [20]. The differing characteristics of these devices have had a significant impact on the architectural designs of next-generation HPC systems. Based on the data organization and access designs proposed in [2][7], the work in [21] investigated such impact from the perspective of parallel DB/DM algorithms, showing the fundamental shifts of DB/DM workload characteristics.

Since product technologies for MEMS-based I/O storage devices are still under development and multiple design approaches exist, characterization and performance analysis for HPC architectures are primarily done through simulations at the present time, and few have been performed. Subsystem-level simulation tools developed and validated by hardware designers of MEMS devices were used in author’s earlier work. These simulations have been designed and carried out on massively parallel computers and Beowulf clusters with properly installed libraries and application programming interfaces.

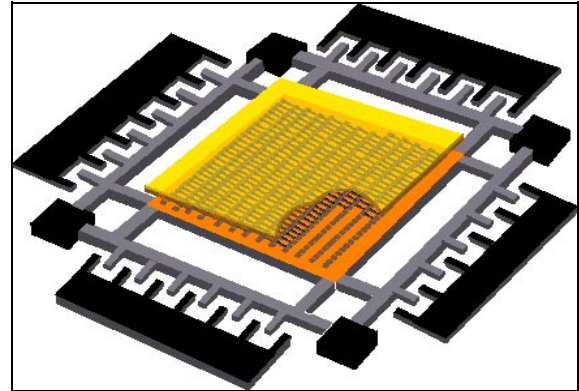


Figure 2. MEMS-based storage device [17]

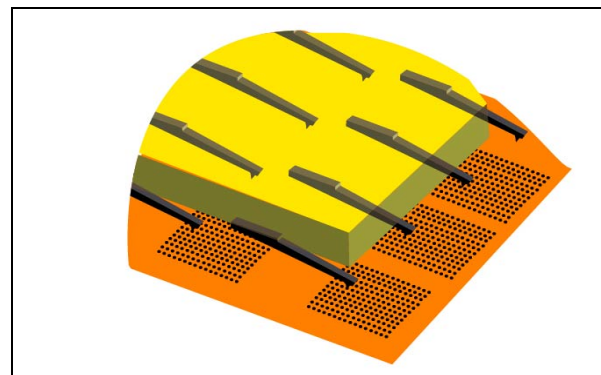


Figure 3. Close-up view of the R/W tips [17]

3. Design Methodology and Procedures

3.1. Design Methodology

Our study employs the following methods:

- (1) Literature search of existing HPC architectures with active I/O storage,
- (2) Literature search of applicable parallel DB and DM algorithms,
- (3) Design and prototyping of a MEMS-based HPC and I/O architecture,
- (4) Adaptation of the applicable DB/DM algorithms as workloads for the HPC architecture,
- (5) Instrumentation of a re-configurable visualization system for data analysis, and
- (6) Performance characterization of our prototype architecture with the DB/DM algorithms

Literature search on active I/O (compute-in-storage) HPC architectures, and parallel DB and DM algorithms for these architectures, will be completed as the first step. Based on the results of this literature search, device-specific libraries, API, and system-level simulation tools will be identified to construct the HPC architecture's software components. Validation and verification work will be performed using published data and device parameters to prototype the architecture on a distributed-memory parallel computer, e.g. a Beowulf cluster. Since MEMS-based I/O devices are still under development, and hence not commercially available for hardware integration into the architecture, these devices will be implemented with software components. Characterization of this "baseline" platform will be approached empirically by means of parallel processing interfaces. Essential performance parameters of this prototype platform will be measured and analyzed. Results from this phase of the proposed research can be considered for publication at conferences related to computer architecture and high performance computing.

Adaptation of the DB and DM algorithms for the HPC architecture will follow the characterization of the platform. This portion of the work implements the algorithms to enable SPMD (Single Program Multiple Data) processing on the architecture. While the simulation cycle depicted in Figure 4 is primarily seen in scientific computing applications, algorithms for data clustering and mining are often exploited for commercial applications. The DB algorithms, particularly the TPC-H queries, are designed to benchmark the performance of large databases used by commercial organizations. With respect to implementation, the DM algorithms perform mining and knowledge discovery using the data produced from the simulation stage. The DB algorithms, on the other hand, require SQL-like queries to be applied to one or more input tables. Therefore, generation (or collection) of these queries and input tables will be performed as a pre-processing step for the DB algorithms.

Performance characterization obtained for the HPC architecture with DB/DM algorithms will serve as empirical data for benchmarking against other HPC systems whose designs exploit similar novel I/O storage devices such as MEMS. The parallelization of the DB or DM algorithms will provide insight into the design process for these algorithms to be integrated into new and emerging hardware architectures. This holistic approach will be essential, because ultimately the computer hardware vendors require this information to design the instruction set architecture (i.e. the ISA) to support the new platform. There is also optimization performed from the hardware up through the application programming

interface (i.e. the API) level, all aiming to increase the overall system performance.

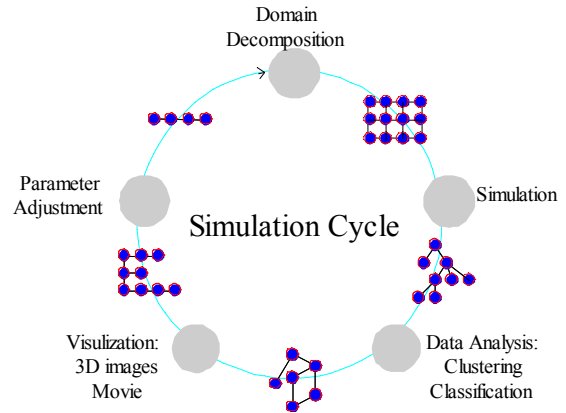


Figure 4. A 5-stage simulation cycle

3.2. Visualization System Setup

To address the visualization needs for this work, where novel and larger-scale DB and DM algorithms are integrated into the HPC architecture, it is necessary to implement a re-configurable, scalable, and cost-effective visualization system for analyzing large data sets generated by the DB and DM algorithms being investigated. The visualization system will also aid in developing the parallel processing programs needed for evaluating the HPC architectures. Several projection- and LCD-based systems have already been designed in [24][25][26][27]. This work will extend the scalable design by exploiting open source software, e.g. Linux, as well as Windows-based environments.

The visualization stage renders 3-D images, animation or other graphical representations to aid in analyzing the data produced by the DB or DM algorithms, which are usually in plain text and interesting patterns such as data clusters may be difficult to capture. For data sets beyond the order of giga bytes (GB), visualization provides significant benefits. This portion of the proposed work entails the design and implementation of a re-configurable and cost-effective visualization system with message passing and the OpenGL [23] libraries. The system uses distributed parallel processing methods to display high-resolution data or scientific images on multiple "tiled" screens (LCD-based), providing zooming, panning, even rotating functions, all on commodity hardware. It is noted that while this visualization system is designed for characterizing the proposed HPC architecture and its DB/DM algorithms, the system is extensible and can be utilized for other scientific applications that may require high-resolution imagery. One solution for the proposed visualization system based on the client-server

programming model would be to tile our display onto several monitors. This design is depicted in Figure 8. The design and implementation, which will be cost-driven, would consider a one PC per video-card design for control of all display elements, using commodity workstations to support distributed processing. With non-expensive dual-output video cards readily available, the design may also be able to support up to two screens per PC, lowering costs substantially. This design will be based on the literature review, and realized during the visualization system design phase to ensure scalability.

4. Architectural and Device Models

4.1. Architectural Model

Our model is based on that presented in [27], which is a distributed active storage design with a switched high performance I/O network, such as InfiniBand or VIA.

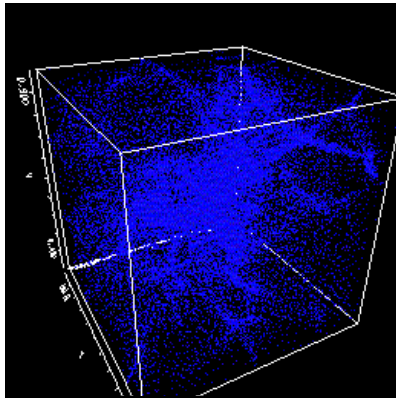


Figure 5. ENZO data visualization

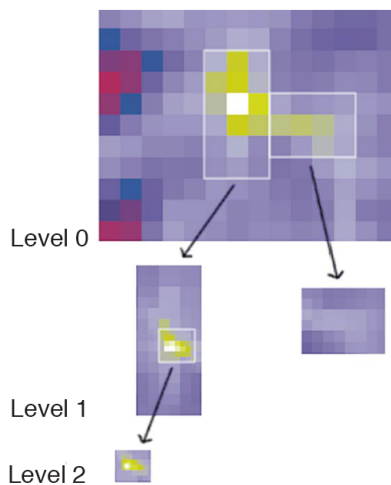


Figure 6. AMR algorithm on sub-grids

4.2. Storage Device-level Models

The work in [21] studied I/O performance gain when shifting from disk- to MEMS-based storage devices. Both of these device models were integrated into the architecture and simulated separately, depending on which of the device models was specified in the initial input parameters for the system. These models are also used in this work, including the G1, G2 and G3 MEMS devices, and the HP C2490A hard disk model, the performance characteristics of which are specified also in [7] and [21].

5. Simulation Platform

Our simulation platform consists of 12 nodes, each with two 2.0-GHz Opteron CPUs and 2 GB of ECC RAM. Each slave node has a 40 GB hard drive for scratch space and the head node has 1.75 TB of redundant RAID5 storage. This platform runs Debian Linux 3.1 Sarge for the AMD64 with a vanilla 2.6.14.3 64-bit kernel and full 64-bit user space. The cluster nodes are connected to a gigabit Ethernet switch with 1-Gbit link for each node for a flat topology.

5.1. Simulation Platform

Our simulation platform will incorporate the *DiskSim* 3.0 [22] simulator to estimate the cost of accessing the storage devices, and is programmed to include the algorithms and data needed to process the given workloads. Message passing interface (MPI) and OpenMP functions will be used for data communication during the processing. The simulation structure used for a single SN includes processes representing the on-device embedded processor, storage controller, the DMA engine and storage mechanism to service the I/O requests.

5.2. Disk- and MEMS-based Devices

The device-level models for the HP C2490A disk and MEMS-based storage devices are as those described in Section 4.2. Our simulation structure will use those two *DiskSim* validated device models for MEMS and disks to obtain accurate I/O access times for the respective devices during the simulations.

Note that these devices models are integrated into our simulation platform to form a base-line architecture for the study, which is then further integrated with the various applicable DB and DM algorithms to enable a holistic view of the entire system. The data output of this system then becomes the input for the visualization phase. It is expected that parameter adjustment will be needed.

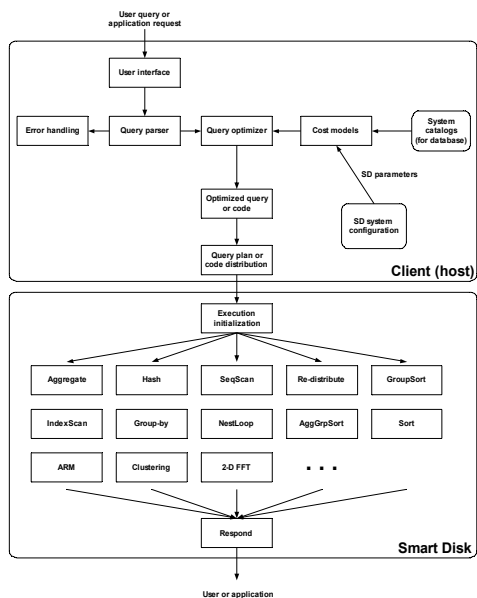


Figure 7. The HPC processing model

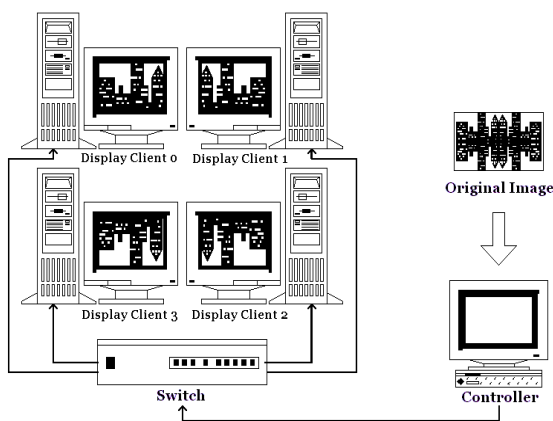


Figure 8. A tiled visualization system

6. Summary and Future Work

In this paper, we presented a feasible framework for investigating new HPC and I/O architectures for parallel database and data mining applications. From a systems design perspective, we proposed our architecture, the target workloads, along with a scalable data visualization system. Our work is based on the methodology adopted by scientific simulation cycles that involve data analysis and visualization. The design goal is to automate the decision-making process for scientific simulations, and thus improve the overall efficiency in running computation-based experiments and data inferences.

Given the proposed design, our future work items include the implementation of the scalable visualization system, which will be followed by the adaptation of selected database and data mining workloads to our proposed architecture. It is noted that these applications will be evaluated using the distributed memory model with message passing for processor synchronization and data transfers. This approach is consistent with the design of our distributed HPC I/O system. Essential performance parameters both at the architectural and application levels will be characterized. Upon completion of this study, we also plan to explore the possibility of automating the scientific simulation cycle based on the 5-stage model as described in this paper.

Acknowledgement

The authors would like to acknowledge the use of the BREMS Cluster of the Idaho Accelerator Center, and the Computer Science Program's Progeny Linux laboratory at Idaho State University's College of Engineering, for the work performed in this paper.

References

- [1] Moore's Law, the Wikipedia URL: http://en.wikipedia.org/wiki/Moore's_law
- [2] S. Schlosser, J. Griffin, D. Nagle and G. Ganger, *Designing Computer Systems with MEMS-based Storage*, Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). January 2000.
- [3] J. Griffin, S. Schlosser, G. Ganger and D. Nagle, *Modeling and Performance of MEMS-based Storage Devices*, ACM SIGMETRICS Performance Evaluation Review, 28(1): 56-65. June 2000.
- [4] P. Braam *et al.*, *Lustre: A Scalable, High-Performance File System*, Lustre Whitepaper Version 1.0, Cluster File Systems Inc. November 2002.
- [5] M. Sivathanu *et al.*, *Semantically-Smart Disk Systems*, Proceedings of the USENIX 2003 Conference on File and Storage Technologies (FAST). April 2003.
- [6] A. Choudhary *et al.*, *Data Management for Large-Scale Scientific Computations in High Performance Distributed Systems*, Cluster Computing: The Journal of Networks, Software Tools, and Applications. Baltzer Science Publishers, 3(2000): 45-60.
- [7] S. Chiu, *Processor-Embedded Distributed Storage for High-Performance I/O*, Ph.D. Dissertation, Northwestern University. June 2004.
- [8] Y. Liu, W. Liao, S. Chiu and A. Choudhary, *On-Line Processing Model for Data Mining in Large Scientific Simulations*, Proceedings of the SIAM Scientific Data Management Conference (SIAM SDM). April 2004.

- [9] G. Bryan, T. Abel and M. Norman, *Achieving Extreme Resolution in Numerical Cosmology Using Adaptive Mesh Refinement: Resolving Primordial Star Formation*, Proceedings of the Conference on Super Computing, November 2001.
- [10] H. Nagesh, S. Goil and A. Choudhary, *Parallel MAFIA: Parallel Subspace Clustering for Massive Data Sets*, Data Mining for Scientific and Engineering Applications. Academic Publishers. March 2001.
- [11] W. Hsu, A. Smith and H. Young, *Projecting the Performance of Decision Support Workloads on Systems with Smart Storage (SmartSTOR)*, Report UCB/CSD-99-1057. University of California at Berkeley. August 1999.
- [12] M. Cannataro, D. Talia, and P. Srimani, *Parallel data intensive computing in scientific and commercial applications*, Parallel Computing Volume 28, Issue 5 Elsevier Science Publishers B. V. May 2002. URL: portal.acm.org/citation.cfm?id=605723
- [13] S. Chiu, W. Liao, A. Choudhary and M. Kandemir, *Processor-Embedded Distributed Smart Disks for I/O-Intensive Workloads: Architectures, Performance Models and Evaluation*, Journal of Parallel and Distributed Computing, Vol. 64, No. 3. Elsevier B.V. March 2004.
- [14] T. Mahapatra and S. Mishra, *Oracle Parallel Processing*, O'Reilly August 2000. URL: www.oreilly.com/catalog/oraclepp/chapter/ch01.html
- [15] *TPC Benchmark Standard Specification Revision 1.4.0 and the TPC Soft Appendix for TPC-H and TPC-R Benchmark Suite*, Transaction Processing Performance Council. June 2001. URL: www.tpc.org.
- [16] L. Carley, G. Ganger, and D. Nagle, *MEMS-Based Integrated-Circuit Mass-Storage Systems*, Comm. of the ACM, 43(11): 72-80. November 2000.
- [17] Carnegie Mellon University CHIPS Project: www.lcs.ece.cmu.edu/research/MEMS.
- [18] Agilent Technologies Foundation Research Program on MEMS URL: www.labs.agilent.com/research/foundation.html.
- [19] IBM Millipede Project URL: www.zurich.ibm.com/st/mems/millipede.html.
- [20] Nanochip Inc. MEMS Research URL: www.nanochip.com.
- [21] S. Chiu, W. Liao, and A. Choudhary, *Processor-Embedded Distributed MEMS-Based Storage Systems for High-Performance I/O*, Proceedings of the 18th International Parallel and Distributed Processing Symposium (IPDPS). April 2004.
- [22] J. Bucy *et al.*, *DiskSim Simulation Environment Version 3.0*, Technical Report No. CMU-CS-03-102, Carnegie Mellon University. January 2003.
- [23] The OpenGL Library and Tools URL: www.opengl.org.
- [24] The NCSA VTK Geometry Viewer Project, URL: www.ncsa.uiuc.edu/TechFocus/Deployment/DBox/downloads.html.
- [25] The PowerWall, URL: www.lcse.umn.edu/research/powerwall/powerwall.html.
- [26] The LLNL Views Visualization Project: Scalable Rendering Infrastructures, URL: www.llnl.gov/icc/sdd/img/infrastructures.shtml.
- [27] Scalable Display Wall, URL: www.cs.princeton.edu/omnimedia/index.html.
- [28] S. Chiu and A. Choudhary, *Impact of Interconnect Protocols and Device-Level Performance on Distributed Active Storage Architectures*, Proceedings of the 2005 International Conference on Parallel and Distributed Processing Techniques and Applications. June 2005.