DSM Perspective: Another Point of View

GORDON BELL AND CATHARINE VAN INGEN

Invited Paper

Distributed shared memory computers (DSM's) have arrived [4], [5] to challenge mainframes. DSM's scale to 128 processors with two to eight processor nodes. As shared memory multiprocessors (SMP's), DSM's provide a single system image and maintain a "shared everything" model.

Large-scale UNIX servers using the SMP architecture challenge mainframes in legacy use and applications. These have up to 64 processors and a more uniform memory access.

In contrast, clusters both complement and compete with SMP's and DSM's, using a "shared nothing" model. Clusters built from commodity computers, switches, and operating systems scale to almost arbitrary sizes at lower cost while trading off SMP's single system image. Clusters are required for high availability applications. Highest performance scientific computers use the cluster (or MPP[1]) approach. High growth markets, e.g., Internet servers, online transmission processing (OLTP), and database systems can all use clusters.

The mainline future of DSM may be questionable because: small SMP's are not as cost effective unless built from commodity components; large SMP's can be built without the DSM approach; and clusters are a cost-effective alternative for most applications to SMP's, including DSM's, for a wide scaling range. Nevertheless, commercial DSM's are being introduced that compete with SMP's over a broad range.

***Keywords*—Distributed memory systems, multiprocessing, multiprocessor interconnection, parallel architectures, parallel processing, shared memory systems.**

I. Introduction to Shared Memory Multiprocessors (SMP's)

Multiprocessors have been compelling since their introduction in the early 1960's due to the following: ability to cover a range of price and performance with fewer designs; incremental upgrades; redundancy for reliability and serviceability; having few physical systems to maintain; and resource fungiblilty. Historically, these compelling advantages have been offset by longer design times limiting product life, limited scaling range, impractical upgrade ability caused by rapid processor or product obsolescence, performance degradation for more processors, lack of O/S and programming support (especially for transparent parallelism), and uncompetitive cost and performance compared to a uni-processor or a cluster of single bus SMP's of the next generation. No doubt, a flaw of multiprocessors' total applications has been the inability to get sustained high performance for single applications—an important subject of this Special Issue. Still for many applications, just being able to run many jobs is enough, not the ability to utilize many processors on a single job.

SMP's are now established and their long-term existence is assured for several reasons. First, users have legacy applications and are likely to prefer a single system image to manage. Second, server manufacturers, e.g., DG, Compaq, Digital (now part of Compaq), HP, IBM, NCR, Sequent, SGI, Siemens, and Sun, are building larger scale SMP's using both DSM and larger switches. For example, SUN's 10 000 Server can have up to 64 SPARC processors, and future servers are being designed to have over 100 processors. Another company has built an SMP with 320 Intel processors. Third, the uniformity of access to memory and other resources simplifies the design of applications. SMP's have evolved to be good enough to replace the mainframe for both legacy and new applications. Parallel apps often use a message-passing program model that a shared memory supports.

II. SMP Evolution

The following section chronicles the multiprocessor evolution.

SMP's with just two to four processors were introduced in the early 1960's when a processor or 16-Kword (64-KB) memory occupied a large cabinet. Machines included: the Burroughs B5500, CDC 3600, Digital PDP-6, General Electric 600-series, and the IBM System/360 Model 50. Their physical structures were all nearly identical—each processor had cables that threaded the memory cabinets that housed part of the distributed cross-point switch. The

Manuscript received March 31, 1998; revised May 26, 1998.

The authors are with Microsoft Corp., Bay Area Research Center, San Francisco, CA 94105 USA (e-mail: gbell@microsoft.com; vaningen@microsoft.com).

Publisher Item Identifier S 0018-9219(99)01540-6.

[1] We use clusters or multicomputers to mean either MPP (for massively parallel processing) and interconnected shared SMP nodes with 2–64 processors. With multicomputers, an operating system (O/S) manages each node. In the early 1990's the technical community defined massive as any system with >1000 processors that did not use a single shared memory and passed messages among the nodes.

0018–9219/98$10.00 © 1999 IEEE

cost was proportional to the product of the processors and memories for cabling and switching plus the memories and processors.

Just a few of these early multiprocessors were delivered, even though the arguments seemed compelling. However, the "cabinet multiprocessor" for the half-dozen processor mainframe has prevailed and become the mainline for Amdahl, Fujitsu, Hitachi, IBM, NEC, and Unisys mainframes. With processor caches and a central switch, memory coherence is expensive, but current mainframes are built with up to 16 processors. Cray Research supercomputers adopted the multiprocessor in 1980 for their XMP, and current supercomputers have up to 32 vector processors that connect to a common memory via a cross-point switch.

In 1971, Bell and Newell [2] conjectured that IBM could have used multiprocessors to cover the same factor of 50 performance range with only two processor types with up to ten processors. It was left as an exercise to the reader as to how this would be accomplished and how it would be used.

The CMU C.mmp project [14] connected 16 modified, PDP-11/20 processors through a centralized cross-point switch to banks of memories. The availability of a large-scale integration (LSI) chip enabled a single, central 16 × 16 cross-point switch that reduced the number of cables to just the sum of the processors and memories. By the time the system was operational, with a new operating system, a single PDP-11/70 could outperform the 16 Model 20 processors.

The CMU Cm* project [6] was the first distributed shared memory (DSM)—a scalable, shared memory multiprocessor. LSI-11 microprocessors were the basic modular building block. Cm* consisted of a hierarchy of modules. Memory accesses were local to a processor, to a cluster of ten, or to the next level in the five-cluster hierarchy. The nonuniform memory access times of the Cm* made programming difficult, and it introduced the need for dealing with memory locality. Several operating systems were built to control Cm*, but a message-passing programming model was required to get reasonable speedup. Today, many highly parallel applications use explicit message passing for communication.

A. Emergence of Mainstream SMP's

Mainstream SMP's based on commodity microprocessors used in PC's and workstations were first introduced in the mid-1980's by Encore[2] and Sequent. All major vendors followed, including Intel beginning in the early 1990's. These "multis" [3] used a common bus to interconnect single chip microprocessors with their caches, memory, and I/O. The "multi" is a natural structure because the cache reduces memory bandwidth requirements and simultaneously can be interrogated, i.e., "snooped" so that memory coherence is nicely maintained across the entire memory system.

Bell correctly predicted that the "multi" structure would be the basis for nearly all subsequent computer designs for the foreseeable future, because the incremental cost for another processor is nearly zero. Two kinds of "multis" exist due to electrical signaling and shared bus bandwidth issues: "single board multis" with two–four processors and memory mounted on one printed wire board (which are the most cost effective) and "backplane multis" consisting of a backplane interconnecting up to 16 modules with two–four processors and their memories. One can foresee "single chip multis" with "on chip" memories.

B. DSM Enters the SMP Picture

In 1992 KSR[3] delivered a scalable computer with a ring connecting up to 34 multis, each with a ring of 32 processors. The KSR-1, was the first cache coherent, nonuniform memory access (cc-NUMA) multiprocessor. DSM was also a cache only memory architecture because memory pages migrated among the nodes. Programs could be compiled to automatically utilize a large number of processors. Like all other computers with nonuniform memory access, the performance gain depended on program locality and communication granularity. Nevertheless, KSR stimulated an interest in all communities for scalable multiprocessors based on the "multi" as a component by providing an existence proof.

Protic *et al.* [10] chronicle the progress and various impediments to DSM. They include reprints of the various systems, e.g., KSR-1, DASH, SCI systems, and components. Attempts were made to use software to create a shared memory environment using clusters [8]. Due to the overhead of a software approach, the important benefit was to stimulate a model and need for a shared memory environment. In 1998, software solutions to provide an SMP environment on multicomputers remains a research topic and challenge. The authors remain skeptical of this approach.

DSM breaks through the "multi" scalability barrier to maintain the simple single system image programming model. The approach is modular: multis are connected with fast cache coherent switching. This modularity allows upgrade ability as well as some expandability over time (and perhaps model changes), but at a penalty determined by the size of the modules, their interconnection bandwidth, and applications. However, significant challenges still exist [9], [10] for them to have a certain future as a standard technique for building SMP's.

In 1998, several manufacturers are delivering cc-NUMA DSM multiprocessors with up to 32 or up to 128 processors that interconnect with client internodes links or switches, e.g., rings or cross-port switches. The SGI Origin with up to 128 processors uses direct links among the two processor and memory nodes and is based on the Stanford DASH project [7]. Other manufacturers, e.g., DG and Sequent, use the Scalable Computer Interface at a relatively low, 1-Gbyte/s rate for maintaining memory coherence. Convex, (now part of HP) used a high bandwidth switch for higher intermodule communication together with SCI to maintain coherence.

[2] Bell was a company founder.

[3] Bell was an investor and advisor to the company.

This slow but steady evolution seems to ensure that DSM's will continue to have a place in future architectures. However, the "optimum" computer measured in ops/sec/$ is still either a uni-processor or "single board multi." With faster processors, minimizing memory latency among the processor accesses becomes critical to performance. With denser silicon, more of the platform interconnect logic can migrate into the processor. If these two trends lead to wider variations in memory timing, maintaining a single system image will exacerbate cost-effective designs. However, for high-performance applications, having a single shared memory is likely to be the critical success factor even if the user has to manage it.

III. Clusters: SMP Competitor and Complement

Clustering is an alternative to the SMP and DSM, while complementing it for reliability and for large-scale systems with many processors. Today, tying together just plain old microcomputers or "multis" claims the world heavyweight title for both commercial and technical applications. To understand clusters as an alternative, we backtrack to the mid-1980's, when the research programs were put in place to build high-performance computers and clusters, i.e., when VAX clusters began to be deployed.

Clusters have been used since Tandem introduced them in 1975 for fault tolerance. Digital offered VAX clusters in 1983 that (like Tandem) virtually all customers adopted because they provided incremental upgrade ability across generations. Users had transparent access to processors and storage. IBM introduced mainframe clusters or Sysplex in 1990. UNIX vendors are beginning to introduce them for high availability and higher than SMP performance.

By the mid-1980's, ARPA's Strategic Computing Initiative (SCI) program funded numerous projects to build scalable computers (e.g., BBN, CalTech, IBM, Intel, Meiko, Thinking Machines). Most of these efforts failed, but the notion of MPP and scalability to interconnect thousands of microcomputer systems emerged. Message-passing standards such as MPI and PVM solidified as applications were modified to use them. If future hardware provides faster message passing, then the need for SMP's for technical computing will decline.

In 1988, Oracle announced development of their parallel database engine Oracle Parallel Server (OPS). Early development was VAX cluster-based and the shared disk design owes much to that heritage. When OPS went into production in 1992, it virtually defined commercial clustering in the Unix market.

In 1998, the world's fastest computer for scientific calculations is a cluster of 9000 Intel Pentium-Pro processors, which operates at a peak-advertised performance of 1.8 Teraflops. The Department of Energy's Accelerated Strategic Computing Initiative (ASCI) is aimed at one Petaflops by 2010. The first round of teraflop sized computers are all clusters from Cray/SGI,[4] IBM, and Intel.

[4] Cray/SGI interconnects four 128-processor DSM computers in a cluster of 512 processors.

Table 1 gives various characteristics of the alternative structures. From the table we see that the key differences are in user transparency of scaling range, and ease of programming. The long-term existence of SMP's favors their use. For many commercial and server apps, the apps hide the need to parallelize and this favors clusters.

Note that DSM and clustering ally for the highest performance but are competitors otherwise. DSM competes with clusters along all the scalability dimensions: 1) arbitrary size and performance; 2) reliability (single image versus one operating system per node); 3) spatial or geographical distributability (computers can be distributed in various locations); and 4) cross-generation upgrades.

IV. The Commercial Market

Commercial computing applications have used multiple cooperating machines for many years. Traditional mail, file, print, database, and online transaction processing servers have long constituted the bulk of the computing market. These servers are now are being joined by new servers for web pages and streaming multimedia. These applications are not small—several web sites qualify among the 100 most powerful computer systems. For example, the website "microsoft.com" uses a cluster of over 200 SMP computers for a total of 600 processors.

Commercial applications can utilize cluster technology because the cluster can be made to provide a transparent environment for applications. Applications have natural parallelism in the parallelized database and queue of transactions that buffer the application developer and end user from that parallelism. The combination of commodity prices and visual database tools are making databases almost ubiquitous—they are inexpensive, easy to use, and more information always seems to be required. Once the database engine has been parallelized and a multithreaded transaction processing monitor supplied, applications which use that environment inherit parallelism.

Commercial systems are evolving a robust infrastructure for distributing applications, through both web and object oriented technologies. Middleware tools for coding and deployment simplify dynamically partitioning a package across servers. Two-tier client–server configurations are being replaced by three-tier client–application-database clusters. Web servers that deliver pages and stream data can be simple clone or affinity clusters.

While commercial computing is naturally parallel, there appear to be a number of practical limits for both multiprocessors and clusters. For example, very large transaction rates are achieved by both parallelism (putting more processors to work on the problem) and then data partitioning (reducing contention for access to storage). Today, the practical (not benchmark-touted) parallelism limit is between 16 and 32 processors for multiprocessors, including the 32 processor DSM's from DG and Sequent—adding more processors within the same box does not result in added throughput. Data and execution partitioning within a cluster is required to go beyond this 32-processor limit.

Table 1
Characteristics of Clusters, SMP's, and DSM's

Characteristic	Clusters Shared nothing	DSM (examples) Nearly shared everything	SMP (examples) Shared everything
Scaling range (examples)	2-Thousands (MPP). Uses commodity nodes and switches. Lowest system cost.	SGI: 2-128 (Large scaling range with one node type.) DG: 4-32 (Low cost to scale. Cluster competitor.)	SUN models: 1 to 2, 8, 14, 30; 16-64 (Many models or configurations required to cover a wide range.)
Commercial (databases, OLTP, Web Servers)	High availability using replication. Commercial apps scale well and transparently.	See SMP Parallelization is easier than for technical apps and hence is a cost-effective alternative to clusters	Legacy commercial apps where database vendors require shared disks Apps scale within the size constraint of SMP
Technical (sans vectors)	Massively Parallel Processing (MPP)	See SMPs . Depends on the interconnection and apps.	Have been used as a vector processor alt. for parallelism
Strengths	Indefinite scaling with commodity nodes. Generations and models may be mixed. Nodes may be dispersed. Used when apps. and system hide parallelism.	See SMPs. Cost-effective alternative to SMPs. SGI demonstrates ability to have a wide range. DG demonstrates the ability use low cost nodes.	A single system for a large range of apps. Easier to build than DSMs Handles legacy apps by given uniform (fungible) access to all resources.
Weaknesses	Separate nodes and operating systems to maintain. Apps that require all resources may be more difficult to parallelize. Databases and OLTP must hide parallelism for users.	See SMPs. Depending on the hardware, software, and apps, SMP benefits may not be realizable.	Large systems require clustering. More expensive than large DSMs. Several models needed to cover large range. Large systems must be co-located. Minimal upgrade across processor generations.

Such partitioning requires significant engineering in the database engine or the applications package. While scalable partitioning is still a niche, it is definitely an expanding niche with visible engineering progress.

The relative growth in the commercial market combined with the viability of clustering in that market will continue to lessen interest and investment in technical computing. The historic difficulties in scaling database engines to very large SMP systems will apply to DSM as well. The newer package development technologies and web servers are targeted to clusters. Higher cost SMP's and DSM's are being sold into the commercial market for certain applications such as decision support where they do have the advantage of a single large address space, IO bandwidth, and familiarity.

V. The Technical Market

Technical applications are traditionally computation and visualization but are increasingly database oriented. Technical applications come from a few independent software vendors (ISV's) or are written directly by users for specific problems. Above the desktop, each parallel application is expensive to create, maintain, and must be tuned to a specific machine. Parallelism is not well hidden from the application developer during coding or tuning. The chief advantage of a shared memory is that it provides fungible resources and fast access for message passing using the message-passing interface (MPI). Automatic parallelization that utilizes two decades of legacy parallel vector programs is still a challenge.

Technical applications are more sensitive to synchronization and communication latency than commercial applications designed to deal with disk latencies. Commercial performance depends on record throughput per second; disk access latency often hides computing or messaging latency. Technical performance depends a great deal on floating point operations per second and hiding latency inherent in distributed computing or DSM is usually difficult. However, more recently the need for large memories and disk arrays also favors low-cost PC technology.

Technical users may not see the need for large SMP systems (including DSM's) for various reasons.

1) Users are content with personal computers that are improving at 60% per year. Today's personal computer would have been classed as supercomputer five years ago.
2) Users with large-scale problems are assembling clusters of 10–100 PC's such as Beowulf, Loki, and Hyglac [12], [13] for specific applications. Two decades ago users deserted computing centers and installed their own VAX computers in a

similar fashion. For example, the number of NSF supercomputing centers has declined from five to two in the last two years.

3) Most technical users do not have the few million dollars necessary to purchase a 64–128 way SMP or DSM that competes with a traditional supercomputer.
4) Above today's 128-node DSM's, MPP's are built as clusters. Programs that require the entire machine see a machine-specific, three-tier hierarchy of multiprocessor, DSM, and cluster. Message-passing is the program model.
5) In the worldwide market, DSM must compete with vector processors that support technical applications in an evolutionary fashion, minimizing end user impact. In the United States, shared DSM resources compete with PC clusters or workstations. Our technical market seems likely to remain centered on the few (two–eight) processor node due to both problem scope and cost effectiveness.

A. High-Performance Technical Applications Rely on All Structures

Looking at the 500 highest performance technical computers in June 1998, there are 107 vector processor supercomputers including clusters of supers, 69 T3D/E Cray and 75 IBM SP2 MPP's (clusters), 25 HP and 91 SGI DSM's, 112 SUN SMP's, and nine other clusters. One cluster of four 64-processor SUN SMP is in the top 50. Only one HP and one SGI DSM are in the top 100. From the data, it is clear that DSM's have yet to impact the highest performance computers, but they are an important component and for smaller sized systems and are apparently cost effective.

We believe the strongest technical computing supporters of large DSM machines are likely to be a few of the U.S. government labs who use them in clusters. Unless adopted widely by commercial users, DSM will remain in the small, higher priced niche. This downward trend will be exacerbated as future PC's are connected with higher performance switches. Since the programming model is often focused on message passing, SMP's offer little advantage over clusters.

VI. Prognosis

DSM is currently utilized where users have legacy code, a compelling application for an SMP, and where managers desire the simplicity of one larger system verses multiple independent computers. The important future for DSM is to be able to take off-the-shelf, commodity-based one–four processor SMP PC's and simply interconnect them. This is the approach used by DG and Sequent for their 32-processor systems. The DG 32-processor DSM system has comparable performance to the Sun 24 processor SMP for commercial OLTP benchmarks but costs significantly less.

Alternatively, the PC barbarians have arrived at the big server gates with do-it-yourself, commodity clusters. While DSM sales are expanding, manufacturers are likely to have a dwindling replacement market for customers with a few million dollars to spend. Retreating to the high-end only works for a few manufacturers, and not forever. The important market segment for DSM's is increasing the scaling range using one–four-node PC components.

Clusters will also compete with small DSM systems because of the cost penalty. DSM still fails two important scalability tests: scaling geographically and across rapidly evolving generations. The commercial market focus on clusters for fault containment, and three-tier application deployment will continue to improve and standardize that alternative. Trends in high-speed networking are closing the gap in system cross-sectional communications (memory) bandwidth, making clusters more ubiquitous.

The next generation of large PC servers could tip the balance away from DSM. On the other hand, applying DSM technology to the PC architecture could ensure its long-term significance, but only if it gets adopted across the entire industry. Commoditization will take at least a three-year development generation. Meanwhile, DSM systems must offer additional value in scalability for not a negligible price penalty since clusters work so easily for high-volume applications.

Acknowledgment

The authors would like to thank the editors and reviewers who stimulated them to solidify and clarify their position so as to present DSM in a less biased light. They are especially indebted their colleague J. Gray for his interaction and tireless editing of two of the drafts.

References

[1] ASCI (descriptions of each of the high-performance compuers). [Online.] Avaliable WWW: http://www.llnl.gov/asci/.
[2] C. G. Bell and A. Newell, *Computer Structures: Readings and Examples*. New York: McGraw-Hill, 1971.
[3] C. G. Bell, "Multis: A new class of multiprocessor computers," *Science*, vol. 228, pp. 462–467, Apr. 26, 1985.
[4] G. Bell, "Ultracomputers a teraflop before its time," *Commun. ACM*, vol. 35, no. 8, pp. 27–47, Aug. 1992.
[5] ——, "1995 Observations on supercomputing alternatives: Did the MPP bandwagon lead to a cul-de-sac?," *Commun. ACM*, vol. 39, no. 3, pp. 11–15, Mar. 1996.
[6] S. H. Fuller, D. P. Siewiorek, and R. J. Swan, "Computer modules: An architecture for large digital modules," in *Proc. 1st Annu. Symp. Computer Architecture (ACM/SIGARCH)*, Dec. 1973, pp. 231–236.
[7] D. Lenoski, J. Laudon, T. Joe, D. Nakahira, L. Stevens, A. Gupta, and J. Hennessy, "The DASH prototype: Implementation and performance," in *Proc. 19th Int. Symp. Computer Architecture*, Gold Coast, Australia, May 1992, pp. 92–103.
[8] K. Li and R. Schafer, "A hypercube shared virtual memory system," in *Proc. 1989 Int. Conf. Parallel System*. Reprinted in J. Protic, M. Tomasevic, and V. Milutinovic, *Distributed Shared Memory: Concepts and Systems*. Piscataway, NJ: IEEE Press, 1998, pp. 121–128.
[9] V. Milutinovic, "Some solutions for critical problems in the theory and practice of DSM," *IEEE TC Comput. Architecture Newslett.*, pp. 7–12, Sept. 1996.
[10] J. Protic, M. Tomasevic, and V. Milutinovic, "Distributed shared memory: Concepts and systems," *IEEE Parallel and Distributed Technol.*, vol. 4, pp. 63–79, Summer 1996.
[11] ——, *Distributed Shared Memory: Concepts and Systems*. Piscataway, NJ: IEEE Press, 1998, p. 365.
[12] D. Ridge, D. Becker, P. Merkey, and T. Sterling, "Beowulf: Harnessing the power of parallelism in a pile-of-PC's," in *Proc. IEEE Aerospace Conf.*, 1997, pp. 79–91.

[13] M. S. Warren, J. K. Salmon, D. J. Becker, M. P. Goda, T. Sterling, and G. S. Winckelmans, "Pentium pro inside: I. A treecode at 430 gigaflops on ASCI red: II. Price/performance of $50/Mflop on Loki and Hyglac," in *Proc. Supercomputing'97*, Los Alamos, CA, 1997. [Online.] Available WWW: http://www.supercomp.org/sc97/proceedings/BELL/WARREN/INDEX.HTM.

[14] W. A. Wulf and C. G. Bell, "C.mmp—A multi-mini-processor," in *Fall Joint Computer Conf. Proc.*, 1972, pp. 765–777.

Gordon Bell is a Senior Researcher at Microsoft Corp., San Francisco, CA. He spent 23 years at Digital Equipment Corporation as Vice President of R&D, where he was responsible for the first mini- and time-sharing computers and led the development of DEC's VAX. He has been involved in the design of many products at Digital and starting a score of companies. As the first Head of NSF's Computing Directorate, he led the National Research Network panel that became the NII/GII and was an author of the first High Performance Computer and Communications Initiative. He is the author of books and papers on computer architecture and entrepreneurial startup companies.

Dr. Bell is a member of various professional organizations, including the National Academy of Engineering, ACM (Fellow), and the American Academy of Arts and Sciences. He was awarded IEEE's von Neumann medal in 1992, and in 1991 he received the National Medal of Technology for his contributions to computing. He is a founder of the Computer Museum.

Catharine van Ingen received the Ph.D. degree in civil engineering from the California Institute of Technology, Pasadena.

She is currently an Architect focusing on SAN (storage area networks) with the Microsoft Windows 2000 File System and Storage Group, San Francisco, CA. Prior to coming to Microsoft, she worked on enterprise class software products for e-commerce and production document management for engineering and plant maintenance. She was the Co-System Architect for the first high end Alpha server (DEC 7000) and holds three patents for high-performance I/O design. In the mid-1980's, she became interested in commodity processor arrays while working on data acquisition systems for large physics detectors at the Fermi National Accelerator Laboratory and Stanford Linear Accelerator Center.