Memory Expansion Technology (MXT): Competitive impact

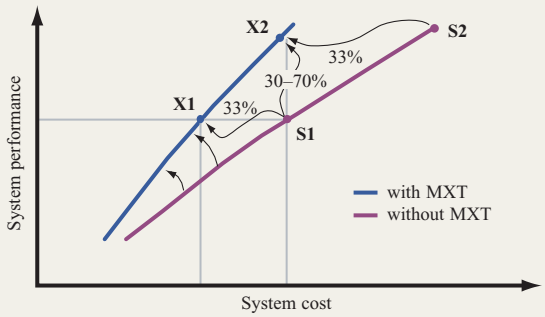
by T. B. Smith
B. Abali
D. E. Poff
R. B. Tremaine

Memory Expansion Technology (MXT™) has been discussed in a number of forums. It is a hardware-implemented means for software-transparent on-the-fly compression of the main-memory content of a computer system. For a very broad set of workloads, it provides a compression ratio of 2:1 or better. This ability to compress and store data in fewer bytes effectively doubles the apparent capacity of memory at minimal cost. While it is clear that a doubling of memory at little cost is bound to improve the price/performance of a system that can be offered to customers, the magnitude of the impact of MXT on price/performance has not been quantified. This paper estimates the range of price/performance improvements for typical workloads from available data. To summarize, the results indicate that MXT improves price/performance by 25% to 70%. The competitive impact of such a large step function in price/performance from a single technology is profound; it is comparable to the entire gross margin in the competitive market for "PC servers."

Introduction

Memory Expansion Technology (MXT*) has been discussed in a number of forums [1–7]. It is a hardware-implemented means for software-transparent on-the-fly compression of the main-memory content of a computer system. For a very broad set of workloads, it provides compression of 2:1 or better [3, 4]. This ability to compress and store data in fewer bytes effectively doubles the apparent capacity of memory at minimal cost. Since memory cost is frequently the single most expensive core component in server systems, it is clear that doubling memory at little cost will surely improve the price/performance of a system. The magnitude or impact of MXT on price/performance has not been quantified or fully appreciated. This paper uses available data on workloads and pricing data to estimate the range of MXT price/performance improvements. To summarize, the results indicate that MXT improves price/performance by 25% to 70%. The competitive impact of such a large step function in price/performance from a single technology is profound. No known alternate technologies or approaches exist that would allow such a large "delta" between two otherwise equivalent implementations using otherwise identical technology. To overcome the price/performance advantage with a simple pricing adjustment would require

0018-8646/01/$5.00 © 2001 IBM



X1 - $7756*
1792 MB real memory
(896 physical)
One 800-MHz proc.
w/133-MHz FSB
One 9.1GB 10K SCSI drive
Two 10/100 Ethernet ports
$150 MXT retail premium

S1 - $11,586*
1792 MB real memory
One 800-MHz proc.
w/133-MHz FSB
One 9.1GB 10K SCSI drive
Two 10/100 Ethernet ports

X2 - $11,226*
2816 MB real memory
(1280 physical)
Two 800-MHz procs.
w/133-MHz system bus
Two 9.1GB 10K SCSI drives
One 10/100 Ethernet port
One gigabit Ethernet port
$150 MXT retail premium

S2 - $17,345*
2816 MB real memory
Two 800-MHz procs.
Two 9.1GB 10K SCSI drives
One 10/100 Ethernet port
One gigabit Ethernet port

* Retail prices — from IBM on-line configurator (10/27/2000)

Figure 1

Price/performance comparison of MXT/non-MXT "twin" systems.

the sacrifice of most if not all of the gross margins common to this market. It would simply be impossible to compete profitably against this technology in this market.

Method for quantifying the price/performance impact of MXT

Extensive evidence exists, and is discussed elsewhere, that MXT can essentially double the capacity of an installed memory to hold instructions and data [2–4, 7]. It has also been extensively verified that the performance of the MXT memory-doubled system is essentially identical to that of a stock system with twice the installed memory of the MXT system [3, 4]. For example, an MXT-enabled system with 1 GB of installed memory will perform to within a few percent of a stock system with 2 GB of memory installed. Since memory is frequently the single most costly component of the core system, it is not surprising that this improves the price/performance of systems.

The method chosen is an attempt to quantify the improvement in price/performance and the competitive impact for typically configured *core servers* (base systems with configured processor and memory). The configured core was chosen as the relevant price/performance domain for several reasons. The most important of these is that the target "PC server" market is now highly disaggregated. Full systems include many components which are separately purchased, either by systems integrators or by the end-user customer when he functions as his own systems integrator. The interfaces or boundaries between system components are standardized, making mix-and-match system construction the norm in this market. Individual purchase decisions are thus made on a component basis, and all other factors being equal, it is the norm to purchase the component with the best price/performance.

For PC servers, the core server is a relevant customer purchase or decision domain; it consists of the base or entry system plus additional memory, additional processors, and/or optional I/O infrastructure upgrades which must be purchased with the base system to configure it for the workload. This excludes PCI or other I/O adapters, which can generally be purchased from a number of competing sources, but includes any I/O infrastructure augmentation, such as options for additional PCI buses that are specific to the base system and would have to be purchased as part of the base configuration. This also excludes disks and network components. Again, these components all compete in their own highly competitive submarkets. Retail price was chosen as the most convenient underlying metric in computing price/performance. Margins and discounts from retail are consistent enough across the industry for one to expect that the fundamentals and conclusion would remain the same if cost or discounted retail were used instead.

Using the above definition for price/performance, we then define the convenience concept of *performance twins*: two server systems which differ in that one is MXT-enabled and the other is a stock or non-MXT-enabled system, but with twice the installed memory of the MXT-enabled system. Other components of the two systems are identical or equivalent. As noted above, evidence has been presented elsewhere that shows that two systems differentiated only in this fashion have essentially identical performance [3, 4]. A similar concept, that of *price twins*, also exists. Price twins are two systems, each balanced for peak price/performance within the same cost constraint; one system is MXT-enabled and the other is not. Price twins cost the same, but have different performance or throughput. They also may have a somewhat different balance of resources (processor, memory, and I/O), given the effective halving of the marginal effective cost of memory in an MXT-enabled system. **Figure 1** illustrates these concepts. In this figure, the S1 and X1 systems are real-world examples of performance twins—systems which are very close clones of one another. The X1 system is MXT-enabled and configured with 896 MB of physical

memory that is expanded to appear as 1792 MB. The S1 system is a stock system configured with 1792 MB of memory, twice that of the MXT-enabled system. These performance-twin systems have nearly identical performance characteristics, despite the significant price difference between them.

In contrast, the X2 system is the MXT-enabled price twin of the S1 system. These systems differ considerably in their performance and have a somewhat different balance of components, but are similarly priced. The degree to which they differ in performance is strongly a function of workload; indeed, the actual optimization of the component or resource balance in each system for peak price/performance is a function of workload.

The performance twin of the X2 system is the S2 system; in general, most MXT configurations can be thought of as having both performance and price twins. That is, the X2 system can be thought of as either a cost-reduced version of S2 (performance twin), or a more capable same-cost version of S1 (price twin).

The prices in Figure 1 are prices as of October 27, 2000 [8] for an IBM e-Server xSeries* 330 as pictured in **Figure 2**. The xSeries 330 is a one-unit (1U), 1.75-in.-high, rack-mount dense server, designed to be packaged with up to 42 servers in a single rack. Such dense packaging has strong market appeal, but illustrates another important point: In principle, it should always be possible to find a performance twin for any configuration, but in practice such a system may be impossible to configure. Because of its dense packaging constraints, an xSeries 330 cannot be configured with more than 4 GB of installed memory. An MXT-enabled configuration exists with 8 GB of apparent memory (4 GB installed memory, expanded to 8 GB), but its performance twin with 8 GB of installed memory cannot be configured. The largest MXT configurations frequently fail to have real-world performance twins. This ability to configure an MXT system beyond the largest stock configuration can provide actual performance advantages that are not fully captured in the analysis that follows. While current 32-bit software constraints may limit the ability to exploit memory beyond 4 GB, thereby limiting the MXT advantage of being able to configure extremely large systems, software vendors, driven by the considerable performance advantages of increased memory size, are rapidly removing these 32-bit system constraints.

The concept of performance twins and price twins suggests two ways to measure the difference in price/performance that could be attributed to MXT. The most natural measure is to use price twins, which can be thought of as similar to the increase in performance achieved by simply turning MXT on without otherwise changing the configuration. Basically this is a direct answer to the question, "Given a fixed number of dollars

Figure 2

IBM xSeries 330 1U rack-mount server.

to spend, how much extra performance does MXT offer?" Since it is extremely workload-dependent, the answer tends to cover a range of values for a range of workloads. For price twins, the improvement is given as

$$\alpha = \frac{MXT_{\text{throughput}}/MXT_{\text{price}}}{STOCK_{\text{throughput}}/STOCK_{\text{price}}} - 1,$$

where MXT and STOCK prices are equal, yielding

$$\alpha = \frac{MXT_{\text{throughput}}}{STOCK_{\text{throughput}}} - 1.$$

For performance twins, the most closely related metric is

$$\beta = \frac{MXT_{\text{throughput}}/MXT_{\text{price}}}{STOCK_{\text{throughput}}/STOCK_{\text{price}}} - 1,$$

where MXT and STOCK throughputs are equal, yielding

$$\beta = \frac{STOCK_{\text{price}}}{MXT_{\text{price}}} - 1.$$

Metric β is less sensitive to workload, because the performance of performance twins is nearly identical for a broad range of workloads. In contrast, the impact of the extra memory for price twins is highly dependent upon workload and the base size of memory, making metric α highly dependent upon workload.

Available data and workloads

Coarsely, the price/performance impact of MXT technology can be computed from measurements on real systems that provide insight into those performance differences which can be attributed to variations in memory size for representative configurations and workloads. As noted, the compressibility of workloads

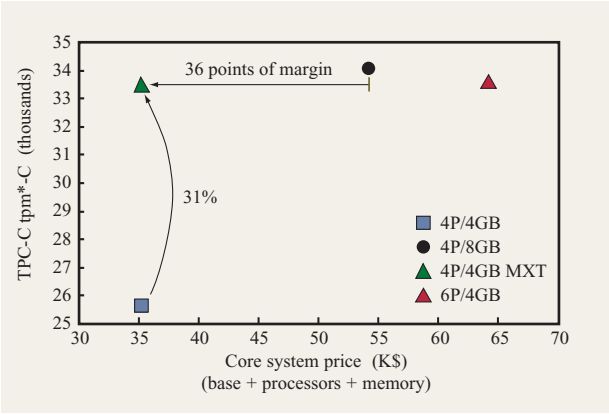


Figure 3

Transaction processing for 550-MHz Intel Xeon processors (2MB L2 caches, Microsoft Windows 2000 server and IIS 5.0). *tpm = transactions per minute. (Source: *http://www.tpc.org*, October 12, 2000.)

in general and the equivalence of physical memory in stock systems to apparent or expanded memory in MXT systems have already been established and documented [3, 4].

Several benchmark databases were examined in order to gain insight from instances in which an industry-significant workload or benchmark was run on similarly configured PC server systems, but with enough variations in memory size to suggest the competitive impact of MXT. Sets of data which met this criterion were found in the reported TPC-C** results of the Transaction Processing Performance Council, and also for reported results for SPECweb99**. The data comparisons by virtue of this methodology are only approximate.

Transaction processing

The TPC-C results for the core PC server components are presented in **Figure 3**. Several similarly configured systems are shown. All systems were multiprocessors using 550-MHz Intel** Xeon** processors with 2MB L2 caches, running Microsoft Windows 2000 Server** and IIS** 5.0. Results were reported for a four-processor, 4GB memory configuration; two four-processor, 8GB memory configurations; and one six-processor, 4GB memory configuration. These results were also informally compared with those for other configurations in the database to establish that all of the results reported here are within norms and are not anomalous data points. A performance twin was then postulated; this is a four-processor configuration with 4 GB of memory installed that has been MXT-expanded to 8 GB. This performance twin was assumed to be intermediate in performance between the two known stock four-processor, 8GB memory configurations being reported. The core price of this performance twin was assumed to be approximately the same as that for the four-processor, 4GB system being reported. Price data uses acquisition prices for the core system components, as reported in the TPC-C executive summary for each reported result. Using this data, it is then possible to infer approximate improvement in price/performance:

$$\alpha = \frac{MXT_{\text{throughput}}}{STOCK_{\text{throughput}}} - 1 = 31\%;$$

$$\beta = \frac{STOCK_{\text{price}}}{MXT_{\text{price}}} - 1 = 56\%.$$

These metrics suggest a 30% to 60% improvement in price/performance, a staggeringly large number in the PC server market.

It is interesting to note that the reported results for a six-processor, 4GB memory configuration suggest that adding 4 GB of memory to the four-processor, 4GB memory system is a more price-effective means to improve performance than is the addition of two processors to the same system. Unfortunately, there were no reported results for a six-processor, 8GB configuration. As an additional measure of price/performance benefit, all of the top ten reported price/performance results for the TPC-C benchmark were examined. The impact on core system throughput/price for performance twins was constructed for each. The average improvement in throughput/price was

$$\beta = \frac{STOCK_{\text{price}}}{MXT_{\text{price}}} - 1 = 33\%.$$

Web-serving workloads

Another set of reported results is for the SPECweb99 benchmark, a web-serving application. The base configurations for these systems are similar in character to those of the IBM e-Server xSeries 330 1U dense server or the comparable COMPAQ DL360**. Both systems are also quite similar to the MXT prototype [7]. Prices on all configurations are for a comparably configured 1U server [8]. All systems were 800-MHz Pentium** III processors running Microsoft Windows 2000 Server and IIS 5.0. **Figure 4** shows the published data from this database: for a single processor with 2 GB of system memory (dark blue squares), a single processor with 4 GB of system memory (orange squares), and a dual processor with 4 GB of system memory (light blue squares). Two hypothetical price twins were constructed, a single processor with 2 GB of memory expanded to 4 GB apparent and a dual processor with 2 GB of memory expanded to 4 GB apparent. Price-twin prices were computed using the list price for a comparably configured stock product, assuming a retail $200 premium for MXT.

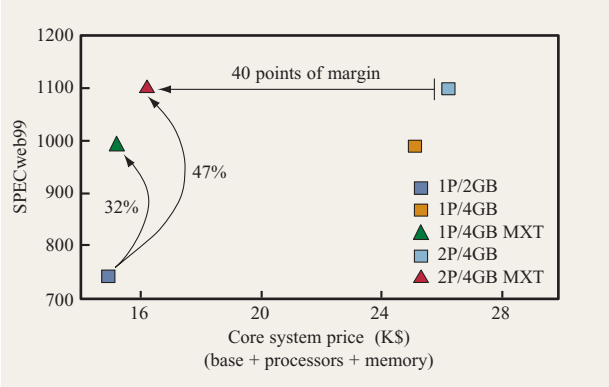


Figure 4

SPECweb99-class workloads for 800-MHz Pentium III processors (Microsoft Windows 2000 server and IIS 5.0).

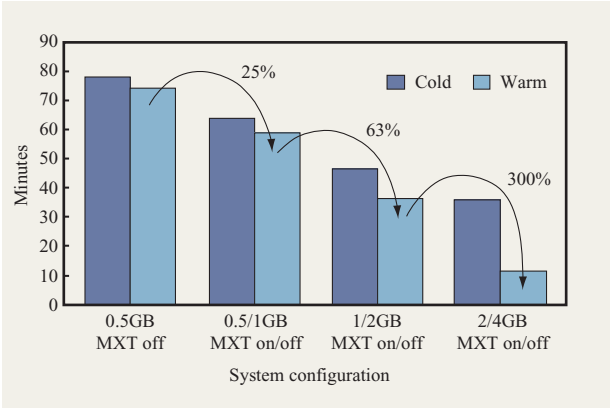


Figure 5

DB2 Windows 2000 regression runs.

These computations led to price/performance metrics of

$$\alpha = \frac{MXT_{\text{performance}}}{STOCK_{\text{performance}}} - 1 = 32\%$$

and

$$\beta = \frac{STOCK_{\text{price}}}{MXT_{\text{price}}} - 1 = 66\%.$$

Again, both metrics suggest a large (30% to 70%) improvement in price/performance. It is interesting to note that in this case the purchase of an additional processor appears to be very cost-efficient. Indeed, the ideal configuration would appear to be a two-processor, 2GB system expanded to 4 GB with MXT.

Measured results

As a final result, the prototype MXT system was measured while running an extract of a commercial company's production database. This configuration is used within IBM primarily as a "quick-look" regression test for ascertaining the impact of DB2* design changes. It is substantially less costly and quicker to run than complex benchmarks such as TPC-C, and is another coarse indication of the general performance characteristics that might be expected. Several configurations were run on the prototype hardware: 512 MB with MXT off, the same 512 MB with MXT on (1 GB expanded), a 1GB configuration with MXT off, the same 1 GB with MXT on (2 GB expanded), and finally a 2GB configuration with MXT off and on (4 GB expanded). Multiple runs were made for each configuration, a "cold" run in which the DB2 buffers were initially empty, and following that a warm run in which the DB2 buffers had been "warmed" by the preceding cold run. The cold run provides an indication of the overhead in the initial load from disk. The warm runs illustrate the advantage of operating out of memory caches once they have been warmed. In all runs there was essentially no significant difference between runs with $2 \times N$ GB of installed memory with MXT off or N GB of installed memory with MXT on (e.g., the 1GB system with MXT off performed exactly like the 0.5GB system with MXT on). The arcs in **Figure 5** indicate the percentage of throughput gain for warm runs when comparing the same system with MXT off and on (e.g., there is a 63% throughput increase when MXT is turned on for a system with 0.5 GB of physical memory installed; these are essentially comparisons between price twins).

For the warm-run comparison between 0.5GB/1GB expanded MXT off/on price twins,

$$\alpha = \frac{MXT_{\text{performance}}}{STOCK_{\text{performance}}} - 1 = 25\%.$$

Similarly, the 1GB/2GB MXT off/on comparison is a comparison between price twins:

$$\alpha = \frac{MXT_{\text{performance}}}{STOCK_{\text{performance}}} - 1 = 63\%.$$

Finally, for the 2GB/4GB MXT off/on comparison price twins,

$$\alpha = \frac{MXT_{\text{performance}}}{STOCK_{\text{performance}}} - 1 = 300\%.$$

It is interesting to note that the benefit of larger memory is more pronounced for this workload for larger memory sizes, which is indicative that both the smaller 512MB memory and 1GB memory configurations are memory-starved.

For this workload, system price/performance is improved by close to 70%. The 2GB memory size is close to the observed "sweet spot" for this class of dense servers.

Conclusions

Memory Expansion Technology has been discussed in a number of forums. It is a hardware-implemented means for software-transparent on-the-fly compression of the main-memory content of a computer system. For a very broad set of workloads, it provides a compression ratio of 2:1 or better. This ability to compress and store data in fewer bytes effectively doubles the apparent capacity of memory at minimal cost. While it is apparent that a doubling of memory at little cost will improve the price/performance of a system that can be offered to our customers, the magnitude or impact of MXT on price/performance has not been quantified nor fully appreciated. Available benchmark and workload data suggest that typical throughput for price/performance improvements of roughly 30% to 70% can be expected. The competitive impact of such a large step function in price/performance from a single technology is profound. In the competitive market for PC servers, this impact is comparable to the entire gross margin in this market.

References

1. S. Arramreddy, D. Har, K. Mak, T. B. Smith, B. Tremaine, and M. Wazlowski, "IBM X-Press Memory Compression Technology Debuts in a ServerWorks NorthBridge," presented at the HOT Chips 12 Symposium, August 13–15, 2000.
2. B. Abali and H. Franke, "Operating System Support for Fast Hardware Compression of Main Memory," presented at the Memory Wall Workshop, International Symposium on Computer Architecture (ISCA2000), Vancouver, B.C., July 2000.
3. B. Abali, H. Franke, D. E. Poff, R. A. Saccone, Jr., C. O. Schulz, L. M. Herger, and T. B. Smith, "Memory Expansion Technology (MXT): Software Support and Performance," *IBM J. Res. & Dev.* **45,** No. 2, 287–302 (2001, this issue).
4. B. Abali, H. Franke, D. Poff, and T. B. Smith, "Performance of Hardware Compressed Main Memory," *Research Report RC-21799*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, July 2000.
5. C. Benveniste, P. Franaszek, and J. Robinson, "Cache–Memory Interfaces in Compressed Memory Systems," presented at the Memory Wall Workshop, International Symposium on Computer Architecture (ISCA2000), Vancouver, B.C., July 2000.
6. J. Chen, D. Har, K. Mak, C. Schulz, B. Tremaine, and M. Wazlowski, "Reliability–Availability–Serviceability Characteristics of a Compressed-Memory System," presented at International Dependable Systems and Networks–2000 (DSN–2000), New York, June 2000.
7. R. B. Tremaine, P. A. Franaszek, J. T. Robinson, C. O. Schulz, T. B. Smith, M. E. Wazlowski, and P. M. Bland, "IBM Memory Expansion Technology (MXT)," *IBM J. Res. & Dev.* **45,** No. 2, 271–285 (2001, this issue).
8. *http://www.pc.ibm.com/eservers/xseries/*, October 29, 2000.

Bibliography

P. Franaszek, J. Robinson, and J. Thomas, "Parallel Compression with Cooperative Dictionary Construction," *Proceedings of the Data Compression Conference, DCC'96*, IEEE, 1996, pp. 200–209.

P. Franaszek and J. Robinson, "Design and Analysis of Internal Organizations for Compressed Random Access Memory," *Research Report RC-21146*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, April 1998.

P. Franaszek, P. Heidelberger, and M. Wazlowski, "On Management of Free Space in Compressed Memory Systems," *Proceedings of the ACM Sigmetrics Conference*, ACM, Atlanta, GA, June 1999, pp. 113–121.

P. A. Franaszek, P. Heidelberger, D. E. Poff, and J. T. Robinson, "Algorithms and Data Structures for Compressed-Memory Machines," *IBM J. Res. & Dev.* **45,** No. 2, 245–258 (2001, this issue).

D. A. Luick, J. D. Brown, K. H. Haselhorst, S. W. Kerchberger, and W. P. Hovis, "Compression Architecture for System Memory Applications," U.S. Patent 5,812,817, 1998.

M. Kjelso, M. Gooch, and S. Jones, "Empirical Study of Memory Data: Characteristics and Compressibility," *IEE Proceedings on Computers and Digital Techniques*, Vol. 45, No. 1, pp. 63–67, IEE, 1998.

U. Vahalia, *UNIX Internals, The New Frontiers*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1996, ISBN 0-13-101908-2.

P. Wilson, S. Kaplan, and Y. Smaragdakis, "The Case for Compressed Caching in Virtual Memory Systems," *Proceedings of the USENIX Annual Technical Conference*, USENIX Association, Monterey, CA, June 1999, pp. 6–11.

http://www5.compaq.com/products/servers/platforms/, October 12, 2000.

http://www.tpc.org, October 27, 2000.

Received October 31, 2000; accepted for publication February 6, 2001

T. Basil Smith *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (tbsmith@us.ibm.com).* Dr. Smith has been a Research Staff Member at the IBM Thomas J. Watson Research Center since 1986. He is currently a Senior Manager responsible for research into the exploitation of high-leverage server innovations and manages the Open Server Technology Department. His work has been on memory hierarchy architecture, reliability, durability, and storage efficiency enhancements in advanced servers. Dr. Smith has received IBM Outstanding Innovation Awards and Outstanding Technical Achievement Awards for his contributions in these fields at IBM. Before joining IBM in 1986, he worked at United Technologies Mostek Corporation in Dallas and at the Charles Stark Draper Laboratory in Cambridge, Massachusetts. He holds more than 20 patents in computer architecture and reliable machine design. Dr. Smith received his Ph.D. degree in computer systems, and his S.M. and S.B. degrees from MIT. He is an IEEE Fellow and a member of the IEEE Computer Society Technical Committee on Fault-Tolerant Computing, and is active in that community. Most recently he was General Chair of the Dependable Systems and Networks Conference (DSN-2000) held in New York in June 2000.

Bulent Abali *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (abali@us.ibm.com).* Dr. Abali has been a Research Staff Member at the IBM Thomas J. Watson Research Center since 1989; he is currently a manager responsible for system software and performance evaluation of advanced memory systems. He has contributed to numerous projects on parallel processing, high-speed interconnects, and memory systems, including RS/6000 SP and MXT. Dr. Abali received his Ph.D. degree in electrical engineering from Ohio State University.

Dan E. Poff *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (poff@us.ibm.com).* Mr. Poff is a System Programmer at the IBM Thomas J. Watson Research Center, where he designs and develops MXT software compression controls. Before joining the Research Center in 1982, he programmed logic chip testers at IBM in East Fishkill, New York. At the Watson Research Center, he first joined a group that developed IBM's first port of UNIX to the first RISC machine, then assisted in porting Carnegie Mellon University's MACH to an early SMP RISC machine. He subsequently assisted in porting MACH to RS/6000. In the early 1990s he joined a group porting Windows NT to the IBM PowerPC. He has received an IBM Outstanding Technical Achievement Award. Mr. Poff received an M.A. degree in history and philosophy of science from Indiana University in 1969 and a B.S. degree in physics from the University of Cincinnati in 1964. He has five patents pending and several publications, and he is a member of the ACM.

R. Brett Tremaine *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (afton@us.ibm.com).* Mr. Tremaine is a Senior Technical Staff Member at the IBM Thomas J. Watson Research Center, where he is responsible for commercial server and memory hierarchy architecture, design, and ASIC implementation. Before joining the Watson Research Center in 1989, he had been at the IBM Federal Systems Division in Owego, New York, since 1982. He has led several server architecture and ASIC design projects, many with interdivisional relationships, and he has received two IBM Outstanding Technical Achievement Awards and several division awards for his contributions. Mr. Tremaine received an M.S. degree in computer engineering from Syracuse University in 1988, and a B.S. degree in electrical engineering from Michigan Technological University in 1982. He has eleven patents pending and several publications, and he is a member of the IEEE.