August 2006
Performance Analysis

Performance of the IBM System x^{TM} 3455

Douglas M. Pase and Matthew A. Eckl
IBM Systems and Technology Group

Abstract

In this paper we examine the performance of the IBM System x™ 3455 server, or x3455. Our analysis includes memory bandwidth and latency, floating-point vector performance and performance of the SPEC® CPU2000 speed and rate benchmarks. In this paper we examine the processor and frequency scaling of the system, and performance implications for supporting 12 DIMMs in the server. During all tests the x3455 performs well. Our findings demonstrate performance consistent with what we expect for a next-generation, two-socket, AMD Opteron processor-based server.

1. Introduction

The IBM System x 3455 joins the lineup of next generation, two-socket, AMD Opteron processor-based rack-optimized server from IBM. This system supports the new Rev. F dual-core processors at speeds of 1.8 GHz, 2.2 GHz, 2.4 GHz, 2.6 GHz and 2.8 GHz. It supports 12 DIMMs of 667 MHz DDR2 in sizes ranging from 512MB to 4GB[1] per DIMM, for a total memory capacity up to 48GB per system. (See Figure 1 below.) The x3455 also supports two I/O slots, one of which is a PCI-Express (PCI-E) x16 slot. The second slot may be configured as a PCI-E x8 slot, or as a Hyper-Transport Expansion (HTX) slot. There is also room for two cold-swap SATA disk drives, several USB ports in the front and back, and two Gigabit Ethernet ports.

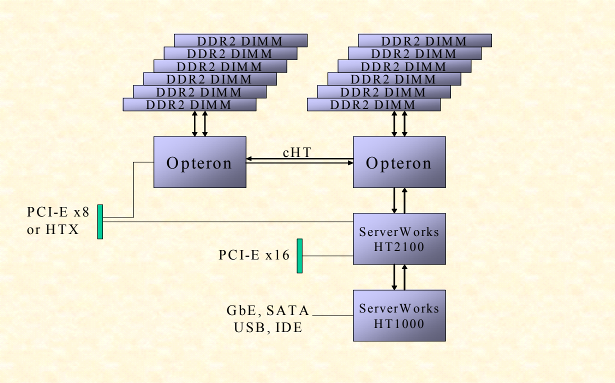

Figure 1. Block Diagram of the x3455

2. Memory Performance

The x3455, like all multiprocessor Opteron designs, is a shared-address, Non-Uniform Memory Access (NUMA) design. The address space spans all memory within the system, so any processor can store or retrieve data anywhere in memory, but memory that is directly attached to the processor can be retrieved more rapidly than memory attached to the other processor. When a reference misses both L1 and L2 cache, the memory subsystem must ask the other processor whether it has the data in its cache. This is called a *snoop request*. The processor must respond with the data, or with a message stating that it does not have the desired data. This is called a

[1] The x3455 will support the 8GB (2x4GB) DIMM option when it becomes available.

snoop response. A reference to local memory requires a snoop request and response before data gathered from local memory can be used. A reference to remote memory (memory attached to the other processor) also suffers from additional latency when the data is transmitted over the HyperTransport link.

Memory performance can be stated in terms of memory bandwidth or throughput and latency. The total bandwidth of a system is the product of the number of channels within the system and the speed of each channel. It is an indication of the upper bound of performance – a measure guaranteed never to be exceeded, but not necessarily something that can be achieved. The bandwidth of a fully configured x3455 is 21,333 MB/s. Memory throughput is a measure of what can actually be achieved. In our tests we used the Stream benchmark [1] for measuring memory throughput. We were able to achieve 12,803 MB/s on a two-processor configuration using the Stream Scale benchmark. We also measured memory latencies from just below 48 ns up to just above 105 ns depending on data locality and memory loading, using a custom benchmark called pChase.

The Stream benchmark was designed to test memory throughput. It uses four common operations from scientific and technical computing. The four operations are copy, scale, add and triad. Copy and scale request one 64-bit floating-point value for each 64-bit value that is stored. Add and triad each load two values for each value stored. All four operations request and store data sequentially over a range of memory that is much larger than the largest cache. In this way it approximates some of the important operations that frequently occur in high-performance computing (HPC) applications.

The pChase benchmark is a pointer-chasing benchmark. A chain of pointers is set up to fill any desired size of memory. The pointers are set up specifically so that they do not reference memory sequentially. Only one pointer per cache line is used. The chain is followed repeatedly until the elapsed time is significantly longer than the system clock resolution, guaranteeing that an accurate measurement has been made. No more than one thread is executed per processor core. The benchmark can measure local and remote latency as well as loaded and unloaded latency. Both pChase and Stream are used to determine the effects of different conditions on memory performance.

Unless otherwise specified, all performance results in this paper were measured on an x3455 using 2.8 GHz processors and 12GB of CL5, 667 MHz DDR2 memory in 12 DIMMs. The BIOS was set to disable Node Interleave and ChipKill, which are the default settings. (For an explanation of the effects of these settings, see “Performance of the IBM System x 3755,” by Douglas M. Pase and Matthew A. Eckl [2].)

2.1 Processor Scaling

Processor scaling is one of the more interesting aspects of performance analysis. Processor scaling shows how performance changes when a processor is added to or removed from a system. Figure 2 shows that the throughput of the two-processor configuration is approximately double that of the single-processor configuration. The reason is that each Opteron processor has its own integrated memory controller, and adding a second processor to a system also adds more channels to memory. Furthermore, the act of snooping the remote processor for modified data completes before the local memory read and therefore doesn't reduce the throughput.

Memory latency seems relatively unaffected by the number of processors, although it is a minor curiosity that the two-processor system is slightly faster than the single-processor system. It is an effect for which we currently have no explanation.

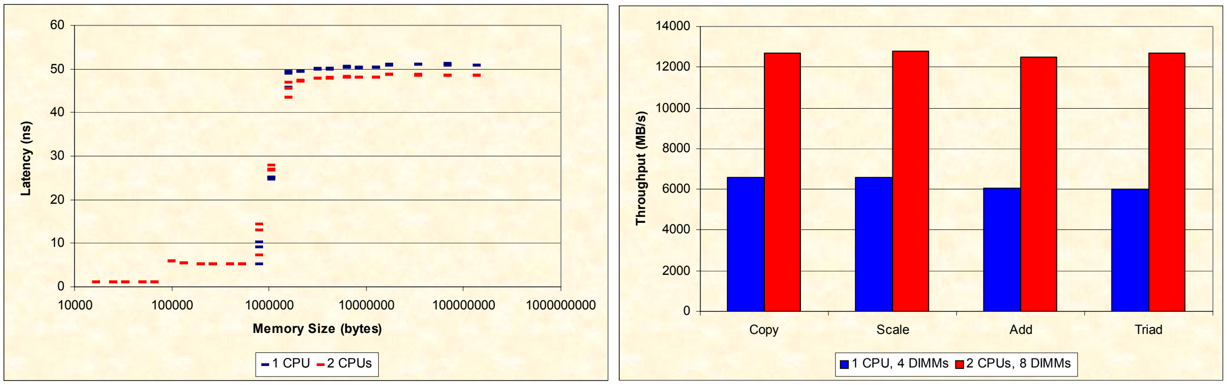

Figure 2. Effect of Processor Scaling on Memory Performance

2.2 Number of DIMMs Populated

The next item we examine is the effect of the number of DIMMs on memory performance. When the number of DIMMs is a power of 2, such as 2 or 4, the memory controller is able to perform certain optimizations that make memory access faster. One such optimization is to interleave addresses between DIMMs in such a way that it reduces conflicts in sequential accesses. This kind of interleave operation is more complex when the number of DIMMs is not a power of two and memory controllers simply don't do it.

So, the interesting question here is what happens when there are 12 DIMMs in the system instead of eight? Are we trading away any performance for the increased memory capacity? Figure 3 shows the answer to this question. The figure shows some interesting and somewhat unexpected results. It shows that for one processor the throughput is about 10% less when four DIMMs are used compared to six DIMMs, and latency is higher. But for two processors the throughput for eight DIMMs is a little over 10% better than for 12 DIMMs, although latency is the same.

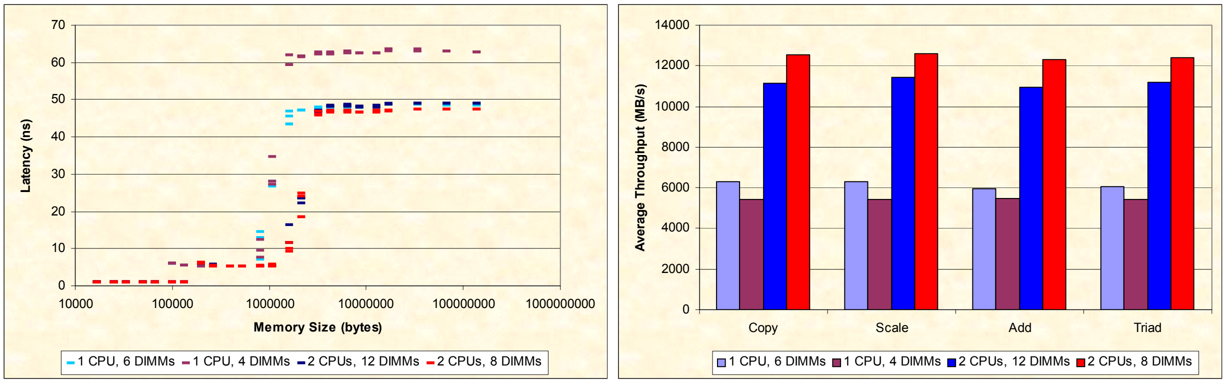

Figure 3. Effect of Populated DIMMs on Memory Performance

2.3 Remote Latency

Every NUMA system has local memory and remote memory. In fact, the great benefit of NUMA is that while flat memory may be easier to work with, it may be possible to achieve better performance at less expense by placing memory closer to some processors than to others. While the emphasis is on exploiting locality, remote references still occur and must be addressed. Figure 4 shows that remote latency is about 30 ns longer than local latency.

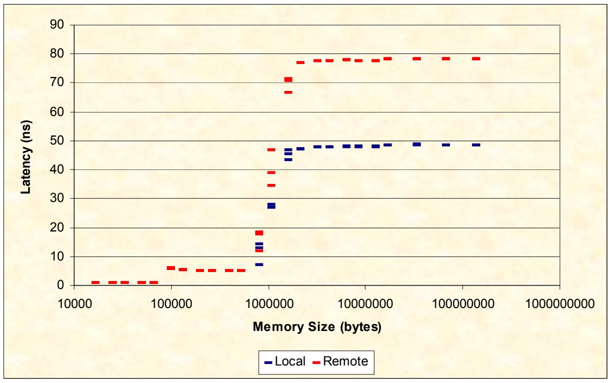

Figure 4. Effect of Remote References on Memory Latency

2.4 Memory Loading

Memory loading refers to the number of concurrent references outstanding at a given moment. Servers are designed to maximize system throughput, and that includes memory throughput. It is quite common to have multiple outstanding memory references being executed concurrently. Stream addresses this issue with the add and triad operations where each iteration requests two operands from memory for each store operation that takes place. The benchmark also uses loop unrolling to initiate the memory references from multiple iterations at once. In this way the Stream benchmark makes heavy use of memory loading.

The pChase benchmark also measures memory loading, but in a more controlled way. Unlike Stream, the number of outstanding concurrent references can be specified directly. When only a single reference is in flight, that is described as *unloaded latency.* When more than a single reference is executed concurrently it is considered *loaded latency.* For the next set of experiments the loading varied between one and two concurrent references over multiple processors. The results are shown in Figure 5.

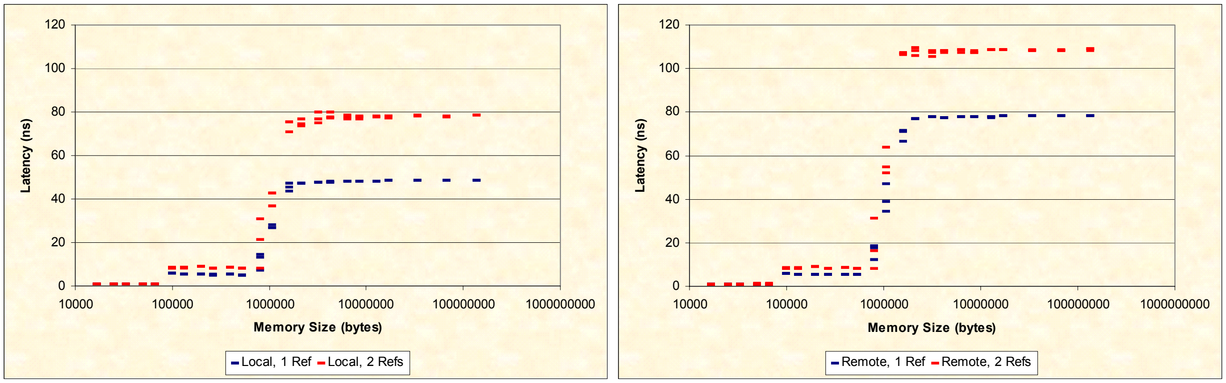

Figure 5. Effect of Loading on Local Memory Latency

As one would expect, loading clearly increases the latency of memory operations, and the effect is about the same whether the references are local or remote. In the local case, the difference in latency between having one and two outstanding references is about 30 ns. In other words, the first reference costs 48 ns, the second costs about 78 ns.

3. 64-Bit Floating-Point Performance

Processor performance is an area where few changes beyond incremental improvements in technology are expected. The 2.8 GHz dual-core processors perform with no surprises. To measure this type of performance we use the High-Performance Linpack (HPL) benchmark.

The Linpack benchmark measures 64-bit floating-point vector performance, with long, easily exploited vectors that fit within available cache [3]. This means memory performance is not a factor. It also means Linpack performance is scalable with the processor clock frequency and the number of vector (SSE2) units in the system. Performance is also a function of the problem size and the available system memory. HPL is a benchmark that employs multiple processes that communicate by sending messages over an interface called MPI[4]. As the problem size grows the benchmark is increasingly dominated by floating-point operations and less influenced by its communication behavior, but the rate of improvement slows as the system approaches its asymptotic performance limit. In our tests using 12 GB of memory the system reached 19.6 GF/s, which is 87.5% of the system peak performance. (With two dual-core processors at 2.8 GHz the hardware peak performance is 22.4 GF/s.) More memory would allow the performance to go even higher, but only by a very small amount. The performance profile of HPL on this system can be seen in Figure 6.

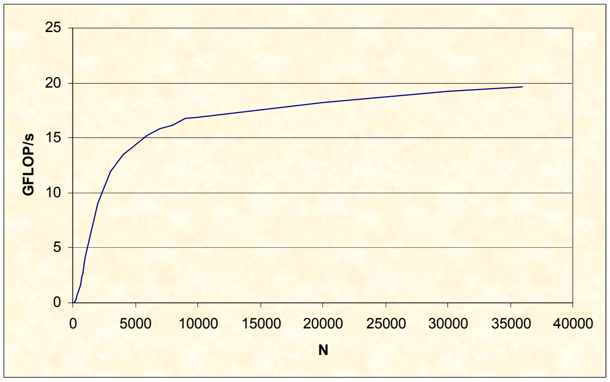

Figure 6. Linpack Performance Using Two 2.8 GHz Dual-Core Processors

4. SPEC CPU2000 Performance

The SPEC CPU2000 benchmark is actually eight benchmarks packaged together [5]. It consists of three independent dimensions that are used together to identify each benchmark. Those dimensions are operation type (integer or floating-point), execution mode (speed or rate), and compilation mode (base or peak). Each run of the benchmark uses a selection of applications taken from the computer industry that reflect the intention of that benchmark. Each application is run three times and given a score based on the median run time of those three runs. All of the application scores are then combined to form a geometric mean[1], which becomes the benchmark score. Eight different benchmark scores are possible, namely integer base speed, integer base rate, integer peak speed, integer peak rate, floating-point base speed, and so on. Base and peak results are usually paired together, so the benchmarks are commonly referred to as SPEC CINT2000 for the integer speed benchmark, SPEC CINT2000 Rates for the integer rate benchmark, and similarly, CFP2000 and CFP2000 Rates for the floating-point speed and rate benchmarks.

The integer benchmarks use 14 integer applications. Most of the applications are highly cacheable, and overall, memory performance has little effect on benchmark performance. The floating-point benchmarks use 12 floating-point applications that generally have strong dependence on both memory performance and floating-point performance, although large caches can also have impact on the outcome.

SPEC CPU2000 can be executed as either a speed benchmark or a rate benchmark. The speed benchmarks execute a single copy of each application serially, using a single processor core, and report the results. These results approximate the speed with which a single task can be completed. Rate benchmarks execute multiple copies of each benchmark concurrently, usually one copy per processor core. This approximates the system throughput under full load. Since servers are designed to complete many tasks at once, sometimes sacrificing the speed of individual tasks, the rate benchmarks are generally a more relevant measure of server performance.

[1] The geometric mean of a series of k numbers is the k^{th} root of the product of those numbers. For example, the geometric mean of x_1, x_2, x_3, x_4 and x_5 would be $(x_1\ x_2\ x_3\ x_4\ x_5)^{1/5}$.

The compilation mode is often the most misunderstood aspect of the SPEC CPU2000 benchmarks. The base and peak benchmarks differ only in what compiler optimization flags may be used to compile the applications. Proper execution of the base benchmarks require that no more than four optimization flags be used, and that all applications use the same optimization flags. Peak results may use any number of flags and each application may be different. The base results approximate an environment such as code development, where a developer wants some combination of flags that works reasonably well, but is unwilling to fine-tune the compilation of the application. In our opinion, base results do not reflect the performance of the system very well. Rather, they reflect how well the compiler has packaged its optimization flags, and how well it has implemented its general-purpose optimization flags such as -O3.

Peak results are unconstrained in their number and choice of optimization flags, so the compiler is free to take advantage of architectural features of the server. As a result, peak results more accurately reflect the performance of the system. In this report only peak results are used.

4.1 Integer Results

Figure 7 shows the results of the integer benchmarks by processor frequency. Because the benchmark is highly cache-friendly, both speed and rate results show near perfect scalability with processor frequency. And although these results are not shown here, it is a fairly safe extrapolation to say that this benchmark scales very closely with the number of processors, too.

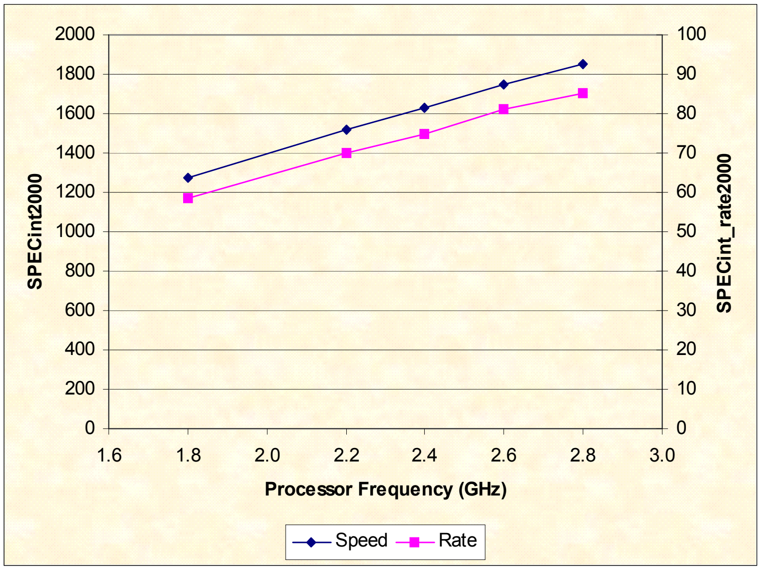

Figure 7. SPEC CINT2000 Speed and Rate Results by Processor Frequency

4.2 Floating-Point Results

Figure 8 shows the results of the floating-point benchmarks by processor frequency. We can see from the charts that performance scales quite linearly with processor frequency. The reason is that there is sufficient memory throughput within the system so that the memory performance does not become the limiting factor.

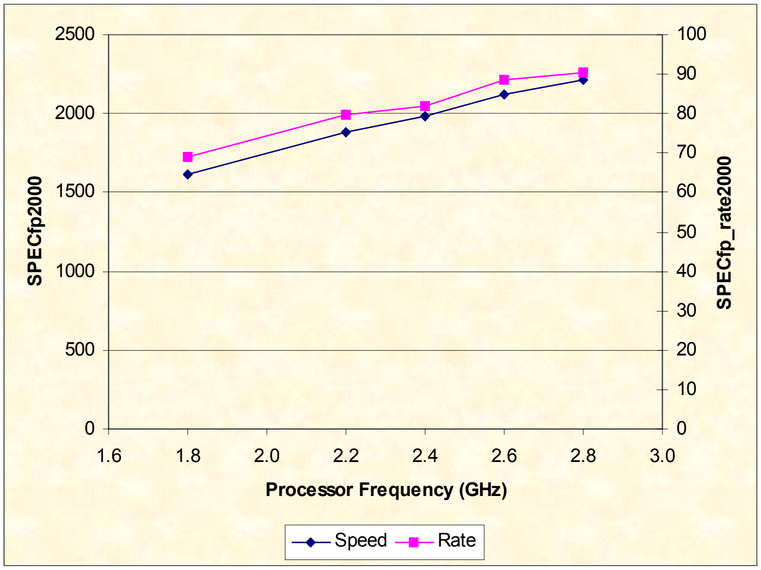

Figure 8. SPEC CFP2000 Speed and Rate Results by Processor Frequency

As mentioned earlier in this section, the performance of this benchmark suite is strongly dependent on memory performance. This touches on two aspects of the Opteron processor design, both having to do with the integrated memory controller. The first is that memory performance is improved with increasing processor frequency. As the processor frequency goes up, requests to the memory controller become faster. This contributes significantly to the linearity of the results in Figure 8.

The second aspect of the integrated memory controller is that memory bandwidth increases as processors are added to the system. We do not show these results, but as processors are added to a system, it is reasonable to expect the rate benchmark performance increase to be very close to linear. We would expect the rate performance for one processor to be half, or perhaps slightly better than half, of the two-processor performance.

5. Conclusions

The x3455 is a new server from IBM that supports one or two AMD Opteron processors. It supports 12 DIMMs per system, or up to 48GB of system memory. Based on the findings presented in this paper, we make the following conclusions:

1. Enabling the Node Interleave setting in the BIOS can result in significant performance loss.
2. Enabling ChipKill in the BIOS increases the robustness of the memory system and does not decrease memory performance.
3. The x3455 shows nearly perfect linear processor scaling between one and two processors because of the integrated memory controller.
4. Local (unloaded) memory latency for one to three processors is fast. We measure 48 ns. Remote latency is somewhat slower at approximately 78 ns.
5. Loading memory with two concurrent references adds 30 ns to memory latency for random accesses.
6. The 64-bit floating-point performance is quite high, achieving 19.6 gigaflops (87.5%) on High-Performance Linpack.
7. The SPEC CPU2000 performance is also in line with expectations.

6. References

[1] Memory Bandwidth: Stream Performance Results, http://www.cs.virginia.edu/stream/.
[2] Douglas M. Pase and Matthew Eckl, “Performance of the IBM System x 3755,” ftp://ftp.software.ibm.com/eserver/benchmarks/wp_x3755_081506.pdf, IBM, August 2006.
[3] Douglas M. Pase, “Linpack HPL Performance on IBM eServer 326 and xSeries 336 Servers,” ftp://ftp.software.ibm.com/eserver/benchmarks/wp_Linpack_072905.pdf, IBM, July 2005.
[4] Marc Snir, Steve Otto, Steven Huss-Lederman, David Walker, and Jack Dongarra, MPI—The Complete Reference, Vol. 1, 2nd Ed., MIT Press, 1996.
[5] SPEC CPU2000, http://www.spec.org/cpu2000/.

IBM Systems and Technology Group

Department MX5A

Research Triangle Park NC 27709

Produced in the USA.

08-06

IBM reserves the right to change specifications or other product information without notice. References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

This publication may contain links to third party sites that are not under the control of or maintained by IBM. Access to any such third party site is at the user's own risk and IBM is not responsible for the accuracy or reliability of any information, data, opinions, advice or statements made on these sites. IBM provides these links merely as a convenience and the inclusion of such links does not imply an endorsement.

All performance information was determined in a controlled environment. Actual results may vary. Performance information is provided "AS IS" and no warranties or guarantees are expressed or implied by IBM. Buyers should consult other sources of information, including system benchmarks, to evaluate the performance of a system they are considering buying.