Performance Study of Virtualized IBM solidDB® Universal Cache and IBM DB2® on the IBM System x3850 X5 server with MAX5, Using VMware vSphere® 5.0

Performance Study

TECHNICAL WHITE PAPER

Table of Contents

Introduction

Increased transaction throughput demands for online transaction processing (OLTP) in today's enterprise applications require high performance technology to ensure speedy response times, IBM's solidDB® Universal Cache in-memory database caching feature is designed to provide extreme performance acceleration for your OLTP applications. Paired with VMware's vSphere infrastructure, multiple virtualized solidDB instances can be deployed easily as on-demand services to accelerate IBM® DB2® performance and dramatically improve throughput using an IBM System x3850 X5 server with the MAX5 memory expansion option.

Performance testing found that solidDB Universal Cache in-memory databases performed well in virtual machines on vSphere 5.0, providing an improvement in throughput of over 300% when compared to the same configuration without solidDB. In addition, the testing found that throughput scalability was essentially linear as up to five solidDB Universal Cache in-memory database nodes were added.

This paper addresses these performance characteristics:

- The performance implications of adding the solidDB Universal Cache in-memory database caching feature into a typical OLTP application environment.
- The performance impact of scaling out with multiple solidDB Universal Cache virtual machines (increasing the number available to the host).
- The practical performance benefit of virtual machines with large virtual memory capacities running on a VMware host with greater than 1TB of physical memory.
- Performance best practices recommendations for the virtualized solidDB environment on VMware vSphere are also provided.

Product Summaries

VMware vSphere® 5.0 is the industry-leading virtualization platform for building cloud infrastructures. It enables users to run business-critical applications with confidence and respond to business needs faster. vSphere accelerates the shift to cloud computing for existing data centers and underpins compatible public cloud offerings, forming the foundation for the industry's only hybrid cloud mode.

IBM solidDB Universal Cache is the industry's first relational, in-memory database caching feature that accelerates virtually all leading relational databases. It uses the familiar SQL language to enable applications to achieve tens of thousands of transactions per second with response times measured in microseconds. When you combine the relational, in-memory data management capabilities of solidDB Universal Cache with the versatility of disk-based databases, applications benefit from the best of both worlds.

With IBM's exclusive eX5, the fifth-generation of X-Architecture®, the System x3850 X5 delivers innovation with enhanced reliability and availability features to enable optimal performance for databases, enterprise applications, and virtualized environments. The x3850 X5 is a versatile 4-socket, 4U rack-optimized, scalable enterprise server that supports up to 2TB of memory (up to 3TB with MAX5). In addition to higher levels of function than its predecessors, the x3850 X5 can be upgraded to the x3950 X5, which offers up to 8-socket SMP operations and up to 6TB of system memory in an 8-socket 2-node complex with the optional IBM MAX5 for System x.

The MAX5 is a scalable, 1U memory expansion unit that provides an additional 512GB in 32 DIMM slots with a memory controller for added performance and a node controller for x3850 scalability. Two MAX5 units can be connected to the 8-socket x3950 X5, providing a total of 192 DIMM slots.

Experimental Configuration and Methodology

The performance studies were done in VMware's internal labs. The purpose of the tests was to measure, analyze, and understand the performance of virtualized IBM solidDB Universal Cache and DB2 running on the x3850 X5 system with MAX5 using VMware vSphere 5.0.

Test-Bed Configuration

An x3850 X5 server was configured with four Intel® Xeon® X7560 processors (with 8 cores per processor) and 1TB of physical memory. The MAX5 with 512GB of physical memory was connected to the x3850. The virtual environment used VMware vSphere 5.0 as the hypervisor. *Table 1* details the hardware configuration.

COMPONENT	DETAILS
System Under Test (SUT)	
Server	IBM System x3850 X5 with MAX5
Processors	4 Intel Xeon X7560 processors @ 2.27GHz, with hyper-threading (8 cores / 16 threads per processor)
Memory	1,536GB DDR-3 DIMMs (1,024GB on server + 512GB attached by MAX5)
HBA	QLogic 8Gb, dual-port QLE2562 Fibre Channel adapter
Virtualization Software	VMware vSphere 5.0
Storage	
Storage Enclosure	EMC VNX 5300 with 2 storage processors
Hard Drives	75 300GB 15K RPM drives

Table 1. Hardware configuration

Figure 1 shows how the application server and database server virtual machines were deployed in our test environment. The topology on the left illustrates a typical application server accessing the DB2 database directly. The topology on the right shows each of the application servers with solidDB Universal Cache (solidDB virtual machine) accessing data in-memory locally. All virtual machines were on the same x3850 X5 server.

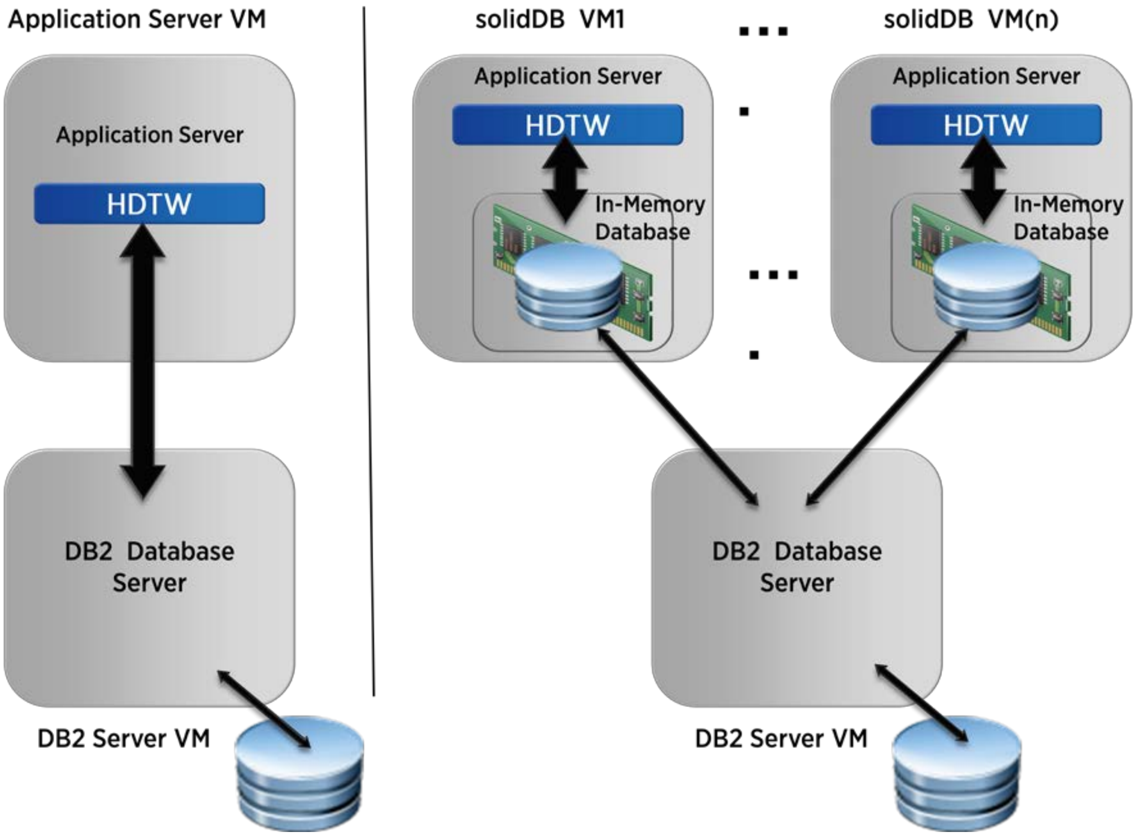

Figure 1. Virtual machine test environment

The DB2 server virtual machine was configured with two 15-disk LUNs for data and indexes with a 10-disk LUN for the DB2 log. The initial size of the database was 190GB. *Table 2* shows the number of CPUs, memory size and software information for each type of virtual machine. All virtual machines used the VMXNET3 virtual network driver.

VIRTUAL MACHINE	DETAILS
DB2 Server	
CPU/Memory	8 vCPUs/256GB virtual memory
Software	Red Hat Enterprise Linux 6.0 DB2 Advanced Enterprise Server Edition V9.7 InfoSphere Change Data Capture V6.3
Database size	190GB
Storage	2 fifteen-disk LUNs for data and indexes and a ten-disk LUN for log
Application Sever	
CPU/Memory	4 vCPUs/208GB virtual memory
Software	Red Hat Enterprise Linux 6.0 DB2 connect V9.7
solidDB (solidDB Universal Cache in-memory database on Application Server)	
Size of CPU/Memory	4 vCPUs/208GB virtual memory
Software	Red Hat Enterprise Linux 6.0 solidDB Universal Cache v6.5 For Linux X86 64 Bit InfoSphere Change Data Capture V6.3 solidDB For Linux X86

Table 2. Virtual machine configuration

Test and Measurement Tools

IBM's Hybrid Database Transaction Workload (HDTW) was used to simulate a complex OLTP workload for an order-entry system of a wholesale supplier. The workload was first run on a single virtualized DB2 instance, then run on the same DB2 instance, but with one or more virtualized solidDB Universal Cache instances acting as front-end caches to the database. The performance of the HDTW workload is measured using two metrics:

- Transaction throughput, measured in transactions per second (TPS)
- Transaction latency, measured in milliseconds (ms)

The throughput is calculated by summing the total number of transactions executed and dividing this sum by the test run duration. The response time is calculated by weighing each transaction response time based on the transaction mix and summing the result. The response time for each transaction is defined as the interval between the time the transaction is started by the application and the time the transaction commit has been executed by the application.

The workload was a mix of 80% read and 20% write transactions. The 20% write transactions were distributed as 9% inserts, 10% updates and 1% deletes transactions. In order to maximize throughput in our test environment, the DB2 server was ramped up to reach a 100% buffer pool hit ratio previous to the HDTW workload experiments being conducted. Read operations accessed the data in the DB2 buffer pool and no disk read I/O was required. Any write operation issued write I/O into storage to persistent data.

For the vSphere server, we used esxtop to record both vSphere and virtual machine–related performance counters. The esxtop tool was configured to log CPU, memory, disk, network, and system counters during HDTW runs.

Test Cases and Test Method

The primary objectives in performing these experiments were to understand the performance implications of scaling out with solidDB Universal Cache virtual machines by comparing this virtual configuration to a DB2 standalone scenario.

For this, the following experiments were conducted:

- DB2 standalone scenario: Application server without solidDB Universal Cache drives HDTW workload.
- solidDB Universal Cache scenario: The solidDB virtual machine drives the HDTW workload through a local solidDB Universal Cache. The number of solidDB virtual machines was incremented from one to five instances for scale out experiments.

Table 3 shows the number of vCPUs and memory sizes for each scenario.

NUMBER AND TYPE OF VIRTUAL MACHINES	TOTAL NUMBER OF VCPUS IN VIRTUAL MACHINES	TOTAL MEMORY (GB) IN VIRTUAL MACHINES
App server + DB2	12	272
1 solidDB + DB2	12	464
2 solidDB + DB2	16	672
3 solidDB + DB2	20	880
4 solidDB + DB2	24	1,088
5 solidDB + DB2	28	1,296

Table 3. Number of vCPUs and memory sizes for virtual machines

Experimental Results and Performance Analysis

Results are presented to show the performance comparison with DB2 standalone and a solidDB virtual machine, scale-out performance of many solidDB virtual machines.

Performance Comparison with DB2 Standalone and a solidDB Virtual Machine

For an apples-to-apples comparison, the application server virtual machine for the DB2 standalone case and the solidDB virtual machine for solidDB Universal Cache case were configured to have the same number of vCPUs and virtual memory size. Both scenarios used the same DB2 server virtual machine.

Throughput and Response Time

The results of the HDTW scores represent the performance impact of adding the solidDB Universal Cache feature into an existing application server configuration. *Figure 2* shows the throughput of the solidDB Universal Cache environment increasing another **315%** from 364 to 1,510 transactions per second. Also, the response time is reduced **59%** from 49 to 20 milliseconds. The solidDB virtual machine case not only can provide **4.15 times** the transactions on the same physical server, but also significantly improves the user experience by reducing response time by **more than half**.

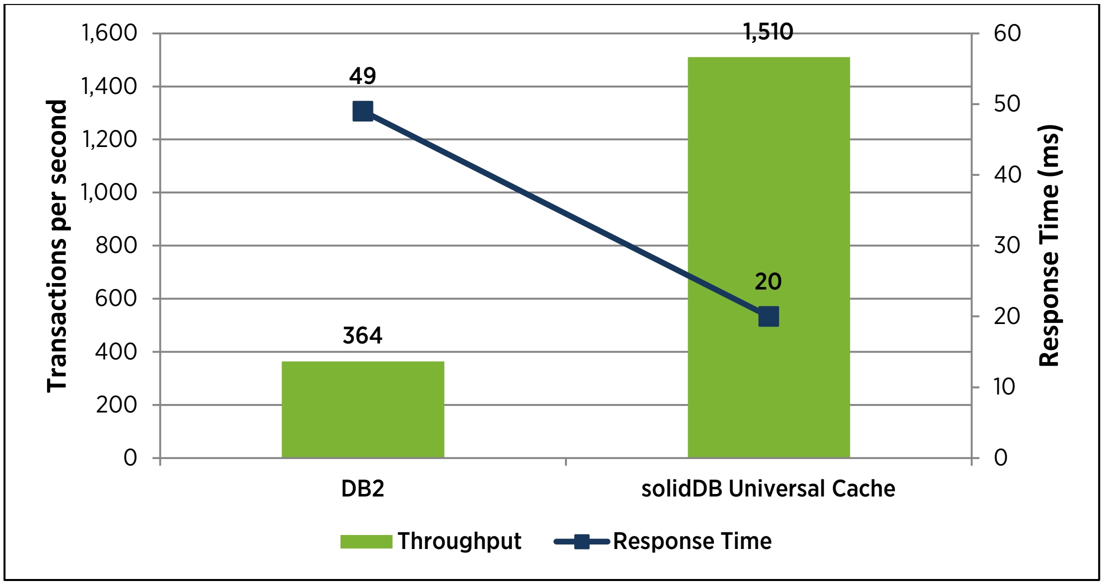

Figure 2. Performance impact by adding a solidDB Universal Cache

Resource Utilization

The overall resource utilization on the vSphere host was measured for each of the test scenarios. Resource utilization is an important metric to evaluate the efficiency of the application workload. The detailed comparisons of CPU and networking component usage between the DB2 standalone and solidDB Universal Cache cases is important in this evaluation of efficiency

	THROUGHPUT	HOST CPU UTILIZATION (%)		NETWORK TRAFFIC (MBITS/SEC)	
		Total	1000 TPS	Total	1000 TPS
DB2 standalone	364	11.5	32	200	549
solidDB UC	1510	14.9	10	46	30
Ratio	4.15	1.30	0.31	0.23	0.06

Table 4. Resource utilization - DB2 standalone vs. solidDB

In the solidDB Universal Cache case, the CPU utilization of the vSphere host increased only 30%, but was able to produce **4.15 times** the throughput. This is a **69%** savings on CPU cost per transaction. The network traffic throughput measured the total networking data transmitted and received for all of the virtual machines in the test environments. The solidDB Universal Cache configuration offered a **94%** per transaction reduction in network traffic to communicate with the back-end DB2 database server. This measurement of network traffic does not include the initial traffic that would occur during the initial sync or caching of data on the solidDB Universal Cache system.

Performance of Multiple solidDB Virtual Machines

The majority of today's OLTP applications utilize a multi-tiered architecture which provides great application scalability and high availability by adding (scaling out) servers. The solidDB Universal Cache feature can be easily deployed for a scaled-out environment. The solidDB virtualization environment was scaled out to demonstrate up to 5 solidDB Universal Cache virtual machines.

Throughput and Response Time

The results of the scale-out experiments show that the throughput increase of the HDTW workload was near-linear with up to 5 solidDB Universal Cache virtual machines. The throughput was increased to **5.09 times** and the response time also was improved to 17ms (**66%**) in the 5 solidDB virtual machine case versus the one solidDB virtual machine case. The x3850 X5 vSphere host was able to maintain the response time to less than 20ms for all of solidDB scale-out cases, which is key to maintain a positive user experience.

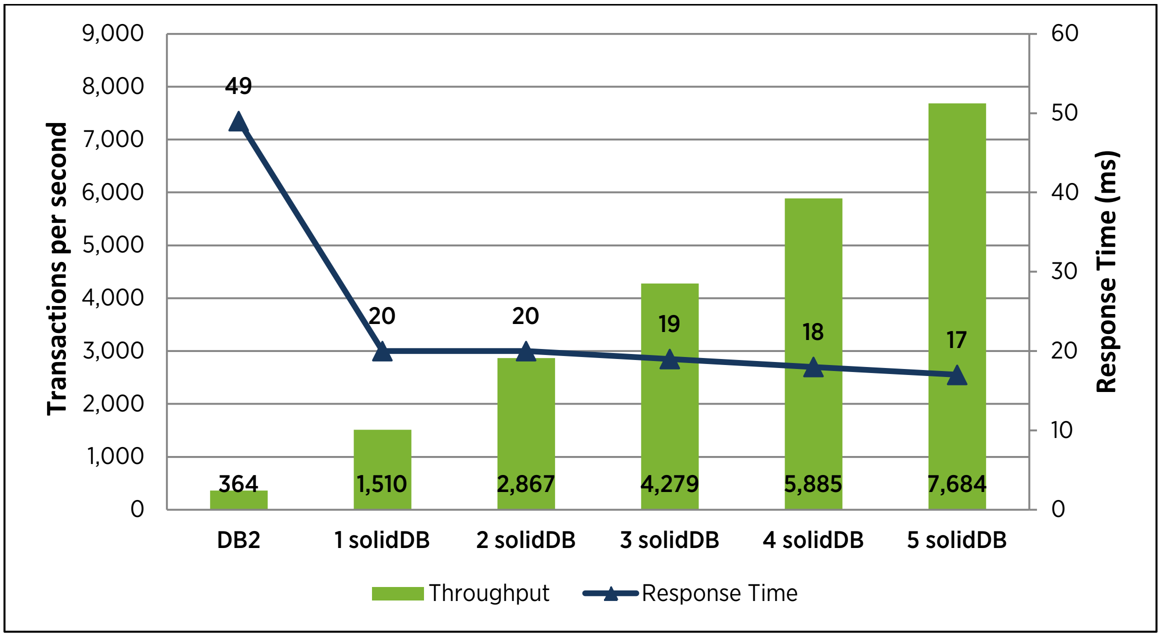

Figure 3. Throughput and response time for scale-out

Processor Utilization

The CPU utilization of DB2, solidDB virtual machines and total CPU usage on the x3850 X5 vSphere host is presented in *Figure 4*. The host CPU utilization for the vSphere 5 server running solidDB instances increased only 6.2 times to produce **21.1 times** the throughput while increasing the number of solidDB instances from zero to five. Because throughput was increased at a much greater rate than CPU usage, the CPU cost per transaction in the five solidDB instances is **71%** better than the DB2 standalone case (without solidDB cache), similar to the 69% CPU savings with one solidDB instance.

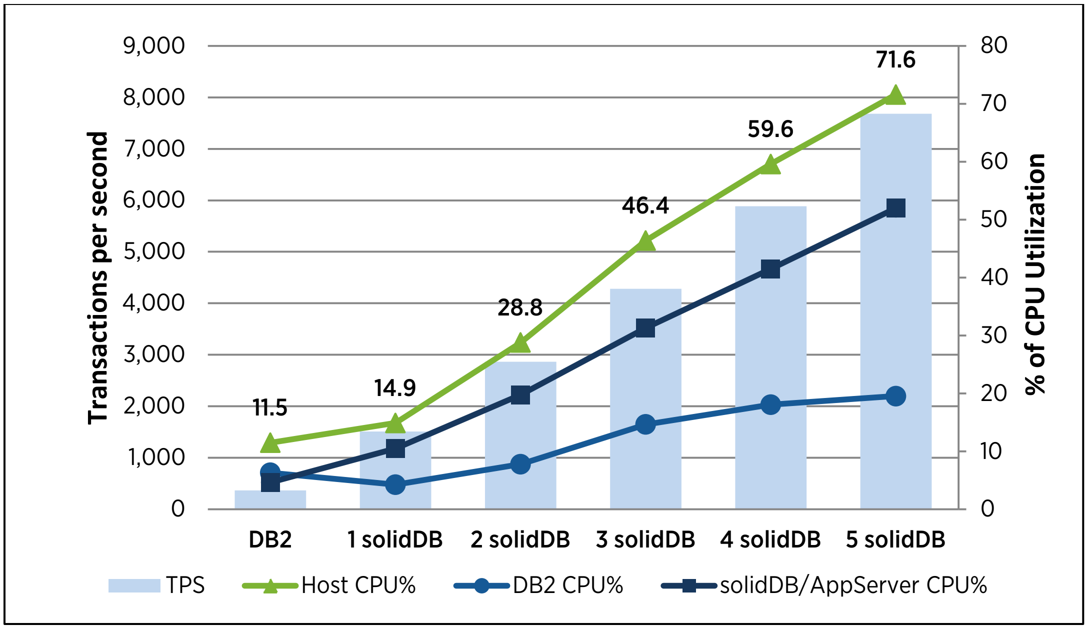

Figure 4. Processor utilization on the x3850 X5 vSphere host for scale-out

Network Traffic

Figure 5 depicts the average of the network traffic per 1,000 TPS. It is initially reduced from 549Mbits/sec in the configuration without solidDB to 30Mbits/sec in the configuration with one solidDB instance. As the number of solidDB instances is increased to five, the rate increases only slightly to 33Mbits/sec. The network traffic was reduced more than **92%** from the configuration without solidDB in all of the configurations with solidDB. This does not include the network traffic that was necessary to populate the caches before the testing began.

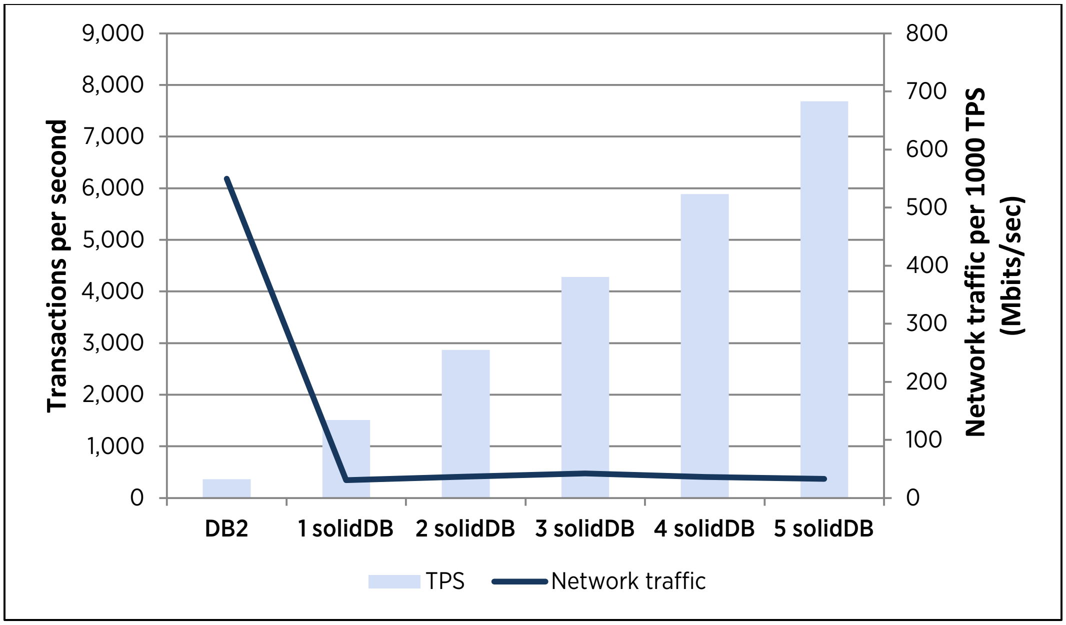

Figure 5. Network utilization on the x3850 X5 vSphere host for scale-out

Best Practices for vSphere 5.0

This section describes some recommendations for improving virtualized solidDB Universal Cache in-memory performance:

- **Enable NUMA for your NUMA processor.**
  The NUMA-aware resource management in vSphere achieves fast memory accesses and balanced load-distribution among NUMA nodes. This provides optimal performance in the NUMA system.
- **Enable hyper-threading for your processor.**
  vSphere automatically makes use of the Hyper-Threading technology, which allows a single physical processor core to behave like two logical processors. The processor can run two independent applications at the same time.
- **Make sure the system has enough memory to avoid ESX host swapping; otherwise, performance in the virtual machines is significantly reduced.**
  Pay attention to the balloon driver's inflation, which may induce guest swapping. Detailed information is available at http://pubs.vmware.com/vsphere-50/topic/com.vmware.ICbase/PDF/vsphere-esxi-vcenter-server-50-resource-management-guide.pdf.
- **Use the VMXNET3 virtual network device.**
  VMXNET3 is the latest virtual network driver. It provides lower CPU cost and better latency. For details, see "Performance Evaluation of VMXNET3 Virtual Network Device" at http://www.vmware.com/pdf/vsp_4_vmxnet3_perf.pdf.
- **Follow IBM practices.**
  For your solidDB Universal Cache virtualized environment, use the same performance tuning and monitoring practices as IBM uses in the following document. See http://publib.boulder.ibm.com/infocenter/soliddb/v6r5/index.jsp?topic=%2Fcom.ibm.swg.im.soliddb.universalcacheuserguide.doc%2Fdoc%2Fs0009029_performancetuning.html.

Conclusion

This paper shows that VMware vSphere on IBM System x3850 X5 server with MAX5 is an excellent platform on which to virtualize and consolidate a high-performance OLTP solution. The addition of multiple, virtualized solidDB Universal Cache instances greatly accelerated the performance of a DB2 database, both in terms of increased transaction throughput and significantly lower transaction response time.

The x3850 X5 with MAX5 server was shown to provide a near-linear scale-out capability from one to five solidDB Universal Cache virtual machines virtualized in vSphere 5.0. With the five solidDB Universal Cache virtual machines deployment, the x3850 X5 vSphere server increased by 21 times the transaction throughput, improved the response time by 66%, and reduced the CPU utilization per transaction by 71%. These excellent HDTW results and significant CPU cost savings in a predictable manner are keys to assisting virtualization and application administrators with capacity prediction and maintenance.

The x3850 X5 with MAX5 can extend the VMware hosts' physical memory capacity beyond the typical 1TB that is supported in 4-processor servers using 16GB DIMMs, to as much as 6TB, if needed. Scaling out multiple solidDB Universal Cache instances in separate virtual machines configured with large virtual memory capacities is a practical use for VMware servers with large physical memory capacities.

The scalability results show how leveraging the Infrastructure-as-a-Service model for a solidDB Universal Cache server topology in the consolidation environment improves transaction response times and maintains a positive user experience. Using vSphere, you can deploy additional virtual machines for a solidDB server needing more resources to meet increased demand. Adding more virtual machines in this way also enhances the efficiency of the physical server processor utilization.

About the Authors

Vincent Lin is a Performance Engineer at VMware. In this role, his primary focus is to evaluate and help improve the performance of VMware products in better supporting key enterprise applications. Prior to VMware, Vincent was a principal performance engineer at Oracle. Vincent received a Master of Science degree from the University of West Florida.

Joe Jakubowski is a Senior Technical Staff Member in the IBM System x Performance Laboratory at Research Triangle Park, NC where he has responsibility for System x® and IBM BladeCenter® performance. He spent 6 years leading a team of engineers focused on developing tools and methodologies to characterize virtualization performance. He holds Bachelor of Science degrees in Electrical Engineering and Engineering Operations from North Carolina State University and a Master of Science degree in Telecommunications from Pace University.

Chris Floyd is an Advisory Performance Engineer at IBM. Chris has been involved in the development and benchmark implementations for many industry-standard benchmarks, including TPC-C, TPC-W, TPC-App, and SPECvirt. Chris was the chairman for the TPC-W and TPC-App benchmarks for several years and is currently the virtualization team lead for IBM System x Performance. Chris holds a Bachelor of Science degree in Computer Science from North Carolina State University.