Judith A. Piantedosi
Archana S. Sathaye
D. John Shakshober

Performance Measurement of TruCluster Systems under the TPC-C Benchmark

Digital Equipment Corporation and Oracle Corporation have announced a new TPC-C performance record in the competitive market for database applications and UNIX servers on the AlphaServer 8400 5/350 four-node TruCluster system. A performance evaluation strategy enabled Digital to achieve record-setting performance for this TruCluster configuration supporting the Oracle Parallel Server database application under the TPC-C workload. The system performance in this environment is a result of tuning the system under test and taking advantage of TruCluster features such as the MEMORY CHANNEL interconnect and Digital's distributed lock manager and distributed raw disk service.

Current industry trends have moved from centralized computing offered by uniprocessors and symmetric multiprocessing (SMP) systems to multinode, highly available and scalable systems, called clusters. The TruCluster multicomputer system for the Digital UNIX environment is the latest cluster product from Digital Equipment Corporation.[1] In this paper, we discuss our test and results on a four-node AlphaServer 8400 5/350 TruCluster configuration supporting the Oracle Parallel Server database application. We evaluate this system under the Transaction Processing Performance Council's TPC-C benchmark to provide performance results in the competitive market for database applications.

The TPC-C benchmark is a medium-complexity, on-line transaction processing (OLTP) workload.[2,3] It is based on an order-entry workload, with different transaction types ranging from simple transactions to medium-complexity transactions that have 2 to 50 times the number of calls of a simple transaction.[4] To run the TPC-C benchmark on a clustered system, the operating system and the database engine must present a single database to the benchmark client. Thus the TruCluster system running the Oracle Parallel Server differs greatly from a network-based cluster system by two significant features. First, the Digital UNIX distributed raw disk (DRD) service enables the distributed Oracle Parallel Server to access all raw disk volumes regardless of their physical location in the cluster. Second, the Oracle Parallel Server uses Digital's distributed lock manager (DLM) to synchronize all access to shared resources (such as in memory cache blocks or disk blocks) across a TruCluster system.

In tuning the system under test, we used the DRD and the DLM services to balance the database across the TruCluster multicomputer system. The configuration includes a specialized peripheral component interconnect (PCI) known as the MEMORY CHANNEL interconnect to greatly improve the bandwidth and latency between two or more member nodes.[5] We tuned the system under test to attain the peak bandwidth of 100 megabytes per second (MB/s) for heavy internode communication during checkpointing by using a dedicated PCI bus for the MEMORY CHANNEL interconnect. We also tuned the system under test to use the very large memory technology and trade off memory for the database cache with memory for DLM locks to improve the throughput. (For a discussion of this technology, see the section Performance Evaluation Methodology.) We measured the maximum throughput, the 90th percentile response time for each transaction type, and the keying and think times. Finally, we compared our measured throughput and price/performance with competitive vendors like Tandem Computers and Hewlett-Packard Company.

The rest of the paper is organized as follows. In the next section, we provide a synopsis of the TruCluster technology and introduce the Oracle Parallel Server, an optional Oracle product that enables the user to use TruCluster technology with the Oracle relational database management system. Following that, we give an overview of the TPC-C benchmark. Next, we describe the system under test and our performance evaluation methodology. Then we discuss our performance measurement results and compare them with competitive vendor results. Finally, we present our concluding remarks and discuss our future work.

TruCluster Clustering Technology

Digital's TruCluster configuration consists of interconnected computers (uniprocessors or SMPs) and external disks connected to one or more shared, small computer systems interface (SCSI) buses providing services to clients.[6] It presents a single raw volume namespace to a client with better application availability than a single system and better scalability than an SMP. A TruCluster configuration supports highly parallelized database managers, such as the Oracle Parallel Server, to provide incremental performance scaling of at least 80 percent for transaction processing applications. The underlying technology to provide this incremental growth includes a PCI-based MEMORY CHANNEL interconnect for communication between cluster members.[6] The MEMORY CHANNEL interconnect provides a 100-MB/s, memory-mapped connection between cluster members.[7] The cluster members map transfers from the MEMORY CHANNEL interconnect into their memory using standard memory access instructions. The use of memory store instructions rather than special I/O instructions provides low latency (two microseconds) and low overhead for a transfer of any length.[7]

The TruCluster for Digital UNIX product supports up to eight (four for commercial DLM/DRD-based applications) cluster members connected to a common cluster interconnect. The computer systems supported within a cluster are AlphaServer systems of varying processor speed and number of processors. The member systems run applications (for example, user applications), as well as monitor the state of each member system, each shared disk, the MEMORY CHANNEL interconnect, and the network. These cluster members communicate over the MEMORY CHANNEL interconnect.[6,8] A MEMORY CHANNEL configuration consists of a MEMORY CHANNEL adapter installed in a PCI slot and link cables to connect the adapters. In a configuration with more than two members, the MEMORY CHANNEL adapters are connected to a MEMORY CHANNEL hub. A typical TruCluster configuration with a MEMORY CHANNEL hub is shown in Figure 1.

Applications can attain high availability by connecting two or more member systems to one or more

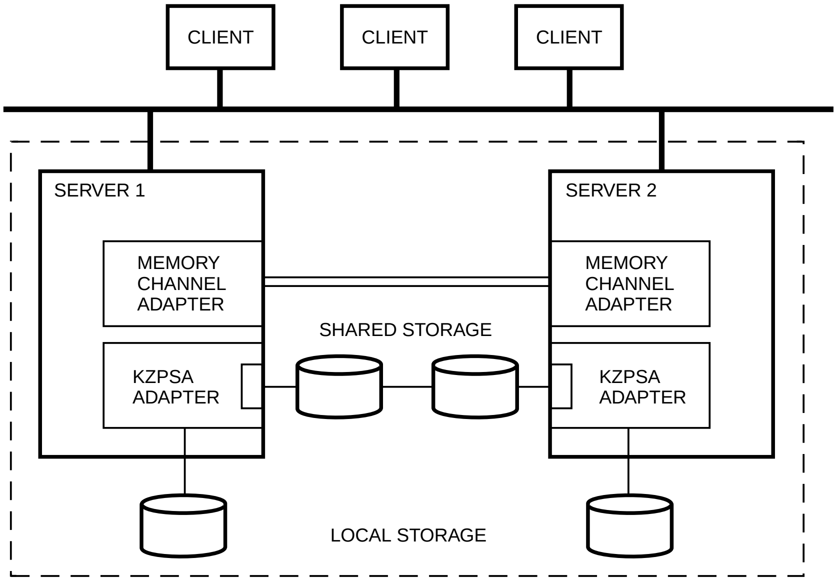


Figure 1
A TruCluster Configuration with MEMORY CHANNEL Hub

shared SCSI buses, thus constructing an Available Server Environment (ASE). A shared SCSI bus is required only for two-member configurations that do not have a MEMORY CHANNEL hub. Although MEMORY CHANNEL is the only supported cluster interconnect, Ethernet and fiber distributed data interface (FDDI) are supported for connecting clients to cluster members. Disks are connected either locally (i.e., nonshared) to a SCSI bus or to a shared SCSI bus between two or more member systems. A single node in the cluster is used to serve the disk to other cluster members. Disks on local buses obviously become unavailable upon failure of the server node. The SCSI controller supported in this configuration is the PCI disk adapter, KZPSA.

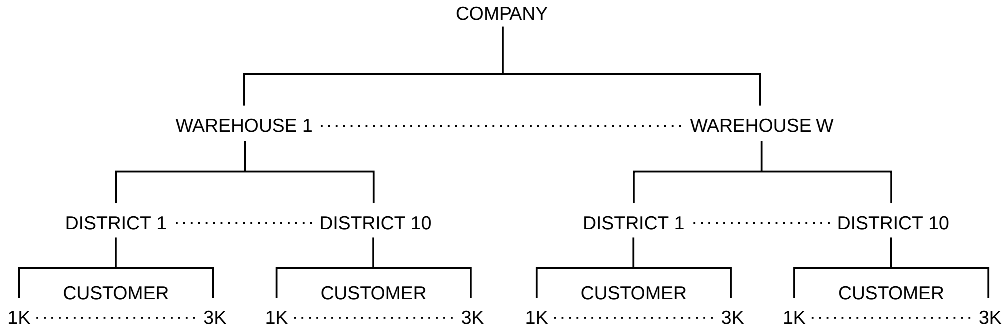
The distinguishing feature of the TruCluster software is its support of the MEMORY CHANNEL as a cluster interconnect, thus providing industry-leadership performance to intracluster communication.[9] The TruCluster software includes the following components: the DLM, the connection manager, the DRD, and the cluster communication service. The DLM facilitates synchronization to shared resources to all member systems in a cluster by means of a run-time library. Cooperating processes use the DLM to synchronize access to a shared resource, a DRD device, a file, or a program. The DLM service is primarily used by the Oracle Parallel Server to coordinate access to the cache and shared disks that have the database installed.[6] The connection manager maintains information about the cluster configuration and maintains a communication path between each cluster member for use by the DLM. The DLM uses this configuration data and other connection manager services to maintain a distributed lock database. The DRD allows the exporting of clusterwide raw devices. This allows disk-based user-level applications to run within the cluster, regardless of where in the cluster the actual physical storage resides. Therefore a DRD service allows the Oracle Parallel Server parallel access to storage media from multiple cluster members. The cluster communication service is used to allocate the MEMORY CHANNEL address space and map it to the processor main memory.

TPC-C Benchmark

The TPC-C benchmark depicts the activity of a generic wholesale supplier company. The hierarchy in the TPC-C business environment is shown in Figure 2. The company consists of a number of geographically distributed sales districts and associated warehouses. Further, there are 10 districts under each warehouse with each district serving 3,000 (3K) customers. All the warehouses maintain a stock of 10,000 items sold by the company. As the company grows, new warehouses and associated sales districts are created. The business activity consists of customer calls to place new orders or request the status of existing orders, payment entries, processing orders for delivery, and stock-level examination. The orders on an average are composed of 10 order lines (i.e., line items). Ninety-nine percent of all orders can be met by a local warehouse, and only one percent of them need to be sold by a remote warehouse.

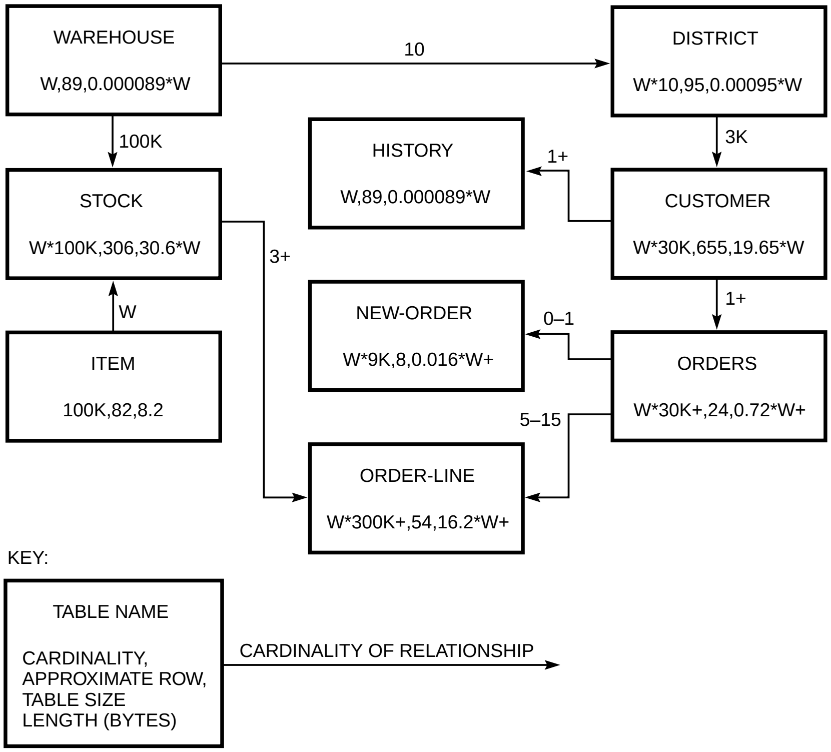
The TPC-C logical database components consist of nine tables.[3] Figure 3 shows the relationship between these tables, the cardinality of the tables (i.e., the number of rows), and the cardinality of the relationships. The figure also shows the approximate row length in bytes for each table and the table size in megabytes. The cardinality of all the tables, except the item table, grows with the number of warehouses. The order, order-line, and history tables grow indefinitely as the orders are processed.

The five types of TPC-C transactions are listed in Table 1.[3] The new-order transaction places an order (of 10 order lines) from a warehouse through a single database transaction; it inserts the order and updates the corresponding stock level for each item. Ninety-nine percent of the time the supplying warehouse is



Source: Transaction Processing Performance Council, *TPC Benchmark C Standard Specification,* Revision 3.0, February 1995.

Figure 2
Hierarchical Relationship in the TPC-C Business Environment



Note: + implies variations over measurement interval as rows are deleted or added.

Figure 3
TPC-C Database Tables Relationship

Table 1
TPC-C Requirements for Percentage in Mix, Keying Time, Response Time, and Think Time[a]

Transaction Type	Minimum Percentage in Mix	Minimum Keying Time (Seconds)	90th Percentile Response Time Constraint (Seconds)	Minimum Mean Think Time Distribution (Seconds)
New order	N/A[b]	18	5	12
Payment	43	3	5	12
Order status	4	2	5	10
Delivery	4	2	5	5
Stock level	4	2	5	5

Notes

[a] Table 1 is published in the Transaction Processing Performance Council's *TPC Benchmark C Standard Specification, Revision 3.0,* February 1995.

[b] Not applicable (N/A) because the measured rate is the reported throughput, though it is desirable to set it as high as possible (45%).

the local warehouse, and only one percent of the time is it a remote warehouse. The payment transaction processes a payment for a customer, updates the customer's balance, and reflects the payment in the district and warehouse sales statistics. The customer resident warehouse is the local warehouse 85 percent of the time and is the remote warehouse 15 percent of the time. The order-status transaction returns the status of a customer order. The customer order is selected 60 percent of the time by last name and 40 percent of the time by identification number. The delivery transaction processes orders corresponding to 10 pending orders, 1 for each district with 10 items per order. The corresponding entry in the new-order table is also deleted. The delivery transaction is intended to be executed in deferred mode through a queuing mechanism, rather than being executed interactively; there is no terminal response indicating the transaction completion. The stock-level transaction examines the quantity of stock for the items ordered by each of the last 20 orders in a district and determines the items that have a stock level below a specified threshold.

The TPC-C performance metric measures the total number of new orders completed per minute, with a 90th percentile response-time constraint of 5 seconds. This metric measures the business throughput rather than the transaction execution rate.[3] It is expressed in transactions-per-minute C (tpmC). The metric implicitly takes into account all the transaction types as their individual throughputs are controlled by the mix percentage given in Table 1. The tpmC is also driven by the activity of emulated users and the frequency of checkpointing.[10] The cycle for generating a TPC-C transaction by an emulated user is shown in Figure 4.

The transactions are generated uniformly and at random while maintaining a minimum percentage in mix for each transaction type. Table 1 gives the minimum mix percentage for each transaction type, the minimum keying time, the maximum 90th percentile response-time constraint, and the minimum think time defined by the TPC-C specification.

The delivery transaction, unlike the other transactions, must be executed in a deferred mode.[3] The response time in Table 1 is the terminal response acknowledging that the transaction has been queued and not that the delivery transaction itself has been executed. Further, at least 90 percent of the deferred delivery transactions must complete within 80 seconds of their being queued for execution. The performance tuning for the system under test determines the number of checkpoints done in the measurement interval and the length of the checkpointing interval. The TPC-C specification, however, defines the upper bound on the checkpointing interval to be 30 minutes.[3]

The other TPC-C metric is the price/performance ratio or dollars per tpmC. This metric is computed by dividing the total five-year system cost for the system under test with the reported tpmC.[11]

Performance Evaluation Methodology

In this section, we first describe the configuration of the system under test (SUT) used for the performance evaluation of the TruCluster system under the TPC-C workload. Then we discuss the testing strategy used to enhance the performance of the SUT.

We show the configuration of the client-server SUT in Figure 5. The server SUT consists of a TruCluster configuration with four nodes; each node is an AlphaServer 8400 5/350 system with eight 350-megahertz (MHz) CPUs and 8 gigabytes (GB) of memory. These nodes are connected together by a MEMORY CHANNEL link cable from the MEMORY CHANNEL adapter on the node to a single MEMORY CHANNEL hub. The local storage configuration for each node consists of 6 HSZ40 redundant array of inexpensive disks (RAID) controllers, 31 RZ28 and 141 RZ29 disk drives, connected to the node by SCSI buses to 6 KZPSA disk adapters. Further, each node is connected to FDDI by a DEFPA FDDI adapter. The nodes communicate with the clients over this FDDI.

The client SUT consists of 16 AlphaServer 1000 4/266 systems, each with 512 MB of memory, one RZ28 disk drive, and one DEFPA FDDI adapter.[12] The remote terminal emulators (RTEs) that are used to generate the transactions and measure the various times (i.e., think, response, or keying time) for each transaction are 16 VAXstation 3100 workstations, each with one RZ28 disk drive. From our logical description of the network topology shown in Figure 6, we see that each of the four nodes in the cluster is connected to four client systems, and each RTE is connected to one client system. The four clients associated with each node are connected to a DEChub 900 switch. Each of the four DEChub 900 products contains two concentrators, one DEFHU-MU 14-port unshielded twisted-pair (UTP) concentrator (for FDDI) and one DEFHU-MH concentrator (for the twisted-pair Ethernet). The DEChub 900 switches are connected to an 8-port GIGAswitch system, which is used to route communications between the client and the server.

The software configuration of the server system is the TruCluster software running under the Digital UNIX version 4.0A operating system and the Oracle Parallel Server database manager (Oracle7 version 7.3) installed on each cluster member. The software configuration installed on each client system is the Digital

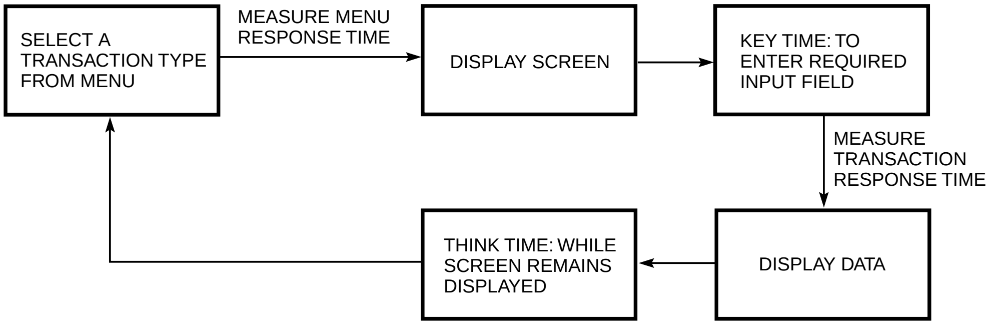

SELECT A TRANSACTION TYPE FROM MENU
MEASURE MENU RESPONSE TIME
DISPLAY SCREEN
KEY TIME: TO ENTER REQUIRED INPUT FIELD
MEASURE TRANSACTION RESPONSE TIME
DISPLAY DATA
THINK TIME: WHILE SCREEN REMAINS DISPLAYED

Figure 4
Cycle for Generating a TPC-C Transaction by an Emulated User

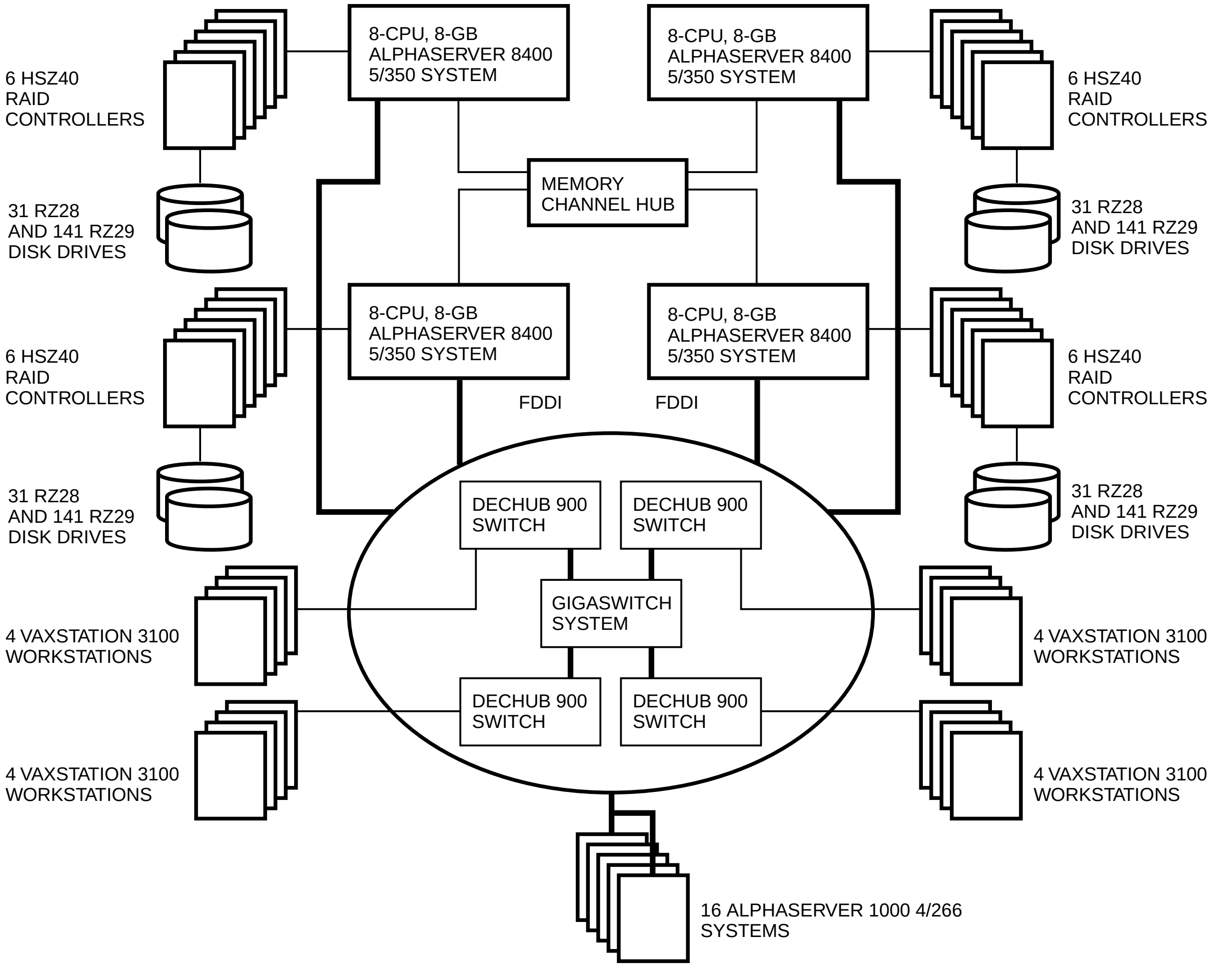

6 HSZ40 RAID CONTROLLERS
8-CPU, 8-GB ALPHASERVER 8400 5/350 SYSTEM
8-CPU, 8-GB ALPHASERVER 8400 5/350 SYSTEM
6 HSZ40 RAID CONTROLLERS
MEMORY CHANNEL HUB
31 RZ28 AND 141 RZ29 DISK DRIVES
31 RZ28 AND 141 RZ29 DISK DRIVES
6 HSZ40 RAID CONTROLLERS
8-CPU, 8-GB ALPHASERVER 8400 5/350 SYSTEM
8-CPU, 8-GB ALPHASERVER 8400 5/350 SYSTEM
6 HSZ40 RAID CONTROLLERS
FDDI
FDDI
31 RZ28 AND 141 RZ29 DISK DRIVES
DECHUB 900 SWITCH
DECHUB 900 SWITCH
31 RZ28 AND 141 RZ29 DISK DRIVES
GIGASWITCH SYSTEM
4 VAXSTATION 3100 WORKSTATIONS
4 VAXSTATION 3100 WORKSTATIONS
DECHUB 900 SWITCH
DECHUB 900 SWITCH
4 VAXSTATION 3100 WORKSTATIONS
4 VAXSTATION 3100 WORKSTATIONS
16 ALPHASERVER 1000 4/266 SYSTEMS

Figure 5
Client-Server System under Test

UNIX version 3.2D operating system and the BEA Tuxedo System/T version 4.2 transaction processing monitor. Further, each RTE runs the OpenVMS operating system and a proprietary emulation package, VAXRTE. In the remainder of this section, we discuss the testing strategy used to generate the transactions on the front end. Then we discuss the tuning done on the back end to achieve the maximum possible tpmC measurements from the SUT.

In conformance with the TPC-C specification, we used a series of RTEs to drive the SUT. The one-to-one correspondence between emulated users on the RTE and the TPC client forms on the client required us to determine the maximum number of users to be generated by the RTE. The main factor we used to determine the number of users was the client's memory size. We assumed that on a client, 32 MB of memory is used for the operating system and 0.25 MB for each TPC client form process. Therefore, with these constraints, each RTE generates 1,620 emulated users. The emulated users then generate transactions randomly based on the predefined transaction mix (as described in Table 1) with a unique seed. This ensures the mix is well defined and a variety of transaction types are running concurrently (to better simulate a real-world environment). We had a local area transport (LAT) connection over Ethernet between each emulated user and a corresponding TPC client form process on the client for faster communication. We show the communication between an RTE, a client, and a server in Figure 7.

We built five order queues on each client corresponding to a transaction type, which allowed us to control the transaction percentage mix. A TPC client form process queues transactions generated by the emulated users to the appropriate order queue using Tuxedo library calls. These transaction requests in each queue are processed in a first in, first out (FIFO) order by the Tuxedo server processes running on the client. We had 44 Tuxedo server processes that were not evenly distributed among the 5 order queues but were distributed so that the number of Tuxedo server processes dedicated to a queue was directly correlated to the percentage of the workload handled by the

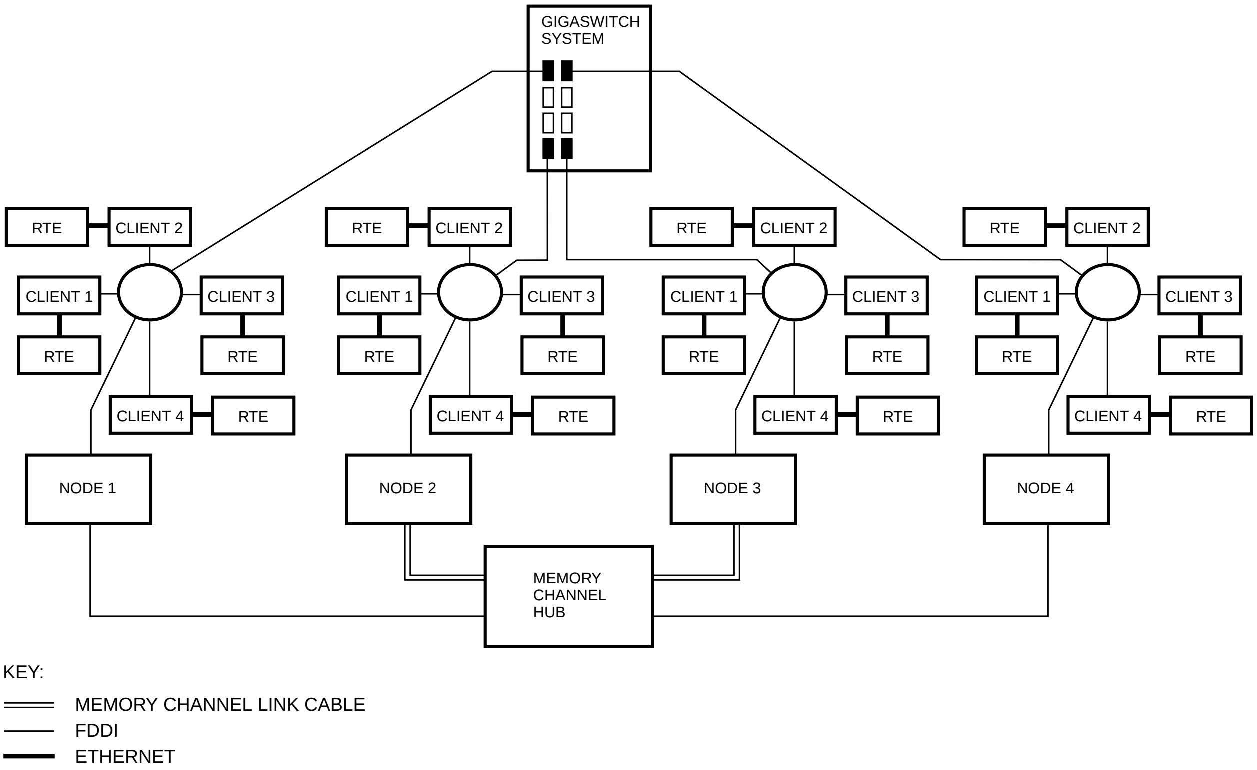
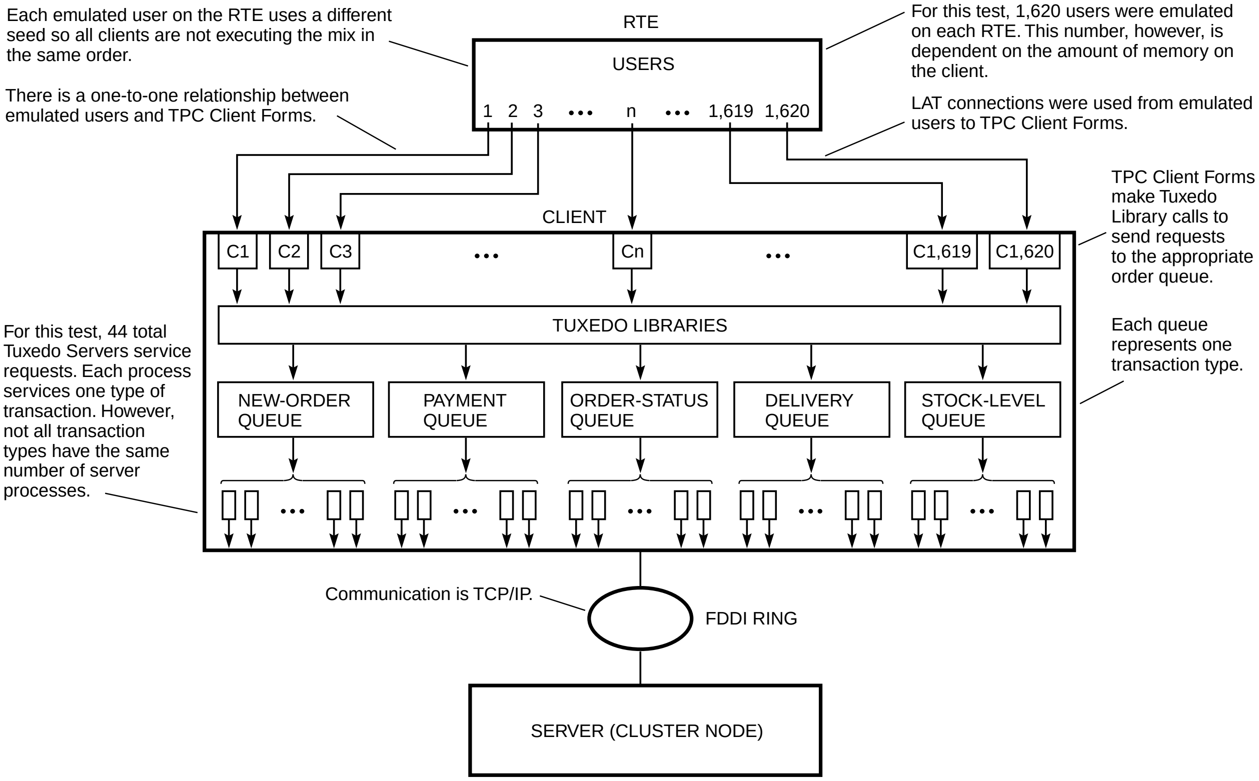


Figure 6
Logical Description of the Network Topology



Each emulated user on the RTE uses a different seed so all clients are not executing the mix in the same order.

There is a one-to-one relationship between emulated users and TPC Client Forms.

For this test, 1,620 users were emulated on each RTE. This number, however, is dependent on the amount of memory on the client.

LAT connections were used from emulated users to TPC Client Forms.

TPC Client Forms make Tuxedo Library calls to send requests to the appropriate order queue.

Each queue represents one transaction type.

For this test, 44 total Tuxedo Servers service requests. Each process services one type of transaction. However, not all transaction types have the same number of server processes.

Communication is TCP/IP.

Figure 7
Communication between an RTE, a Client, and a Server

queue. In other words, the greater the percentage of the workload on a queue, the greater the number of Tuxedo server processes dedicated to that queue. The number of Tuxedo server processes per client is computed based on the rule of thumb that each queue should have no more than 300 outstanding requests during checkpointing and 15 at other times. These Tuxedo server processes communicate with the server system (cluster node) using the Transmission Control Protocol/Internet Protocol (TCP/IP) over FDDI to execute related database operations.[13]

The industry-accepted method of tuning the TPC-C back end is to add enough disks and disk controllers on the server to eliminate the potential for an I/O bottleneck, thus forcing the CPU to be saturated. Once the engineers are assured that the performance limitation is CPU saturation, the amount of memory is tuned to improve the database hit ratio. Because all vendors submitting TPC-C results use this style of tuning, the performance limitation for TPC-C is usually the back-end server's CPU power. In fact, tests have shown that if this method of tuning is not followed on the back-end server, the user will not obtain the optimal TPC-C performance results. Instead, the tests reveal a back-end server configuration that has not fully utilized the server's potential by having unbalanced CPU and I/O capabilities. This type of configuration not only reduces the server's throughput capacity but also adversely affects the price/performance of the SUT.

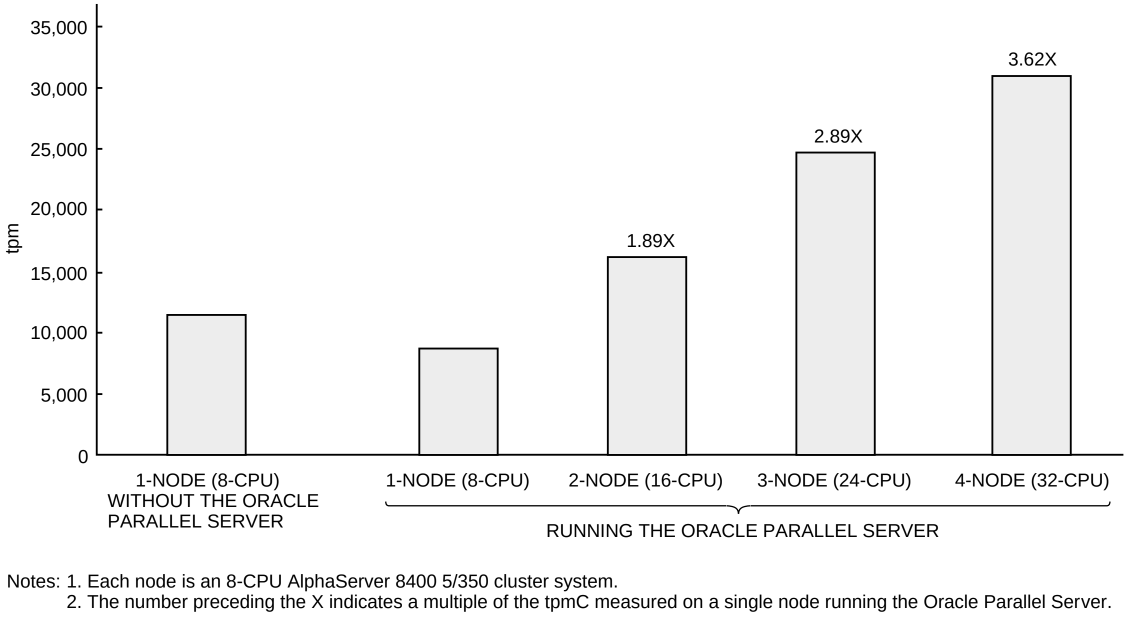
On the back end, we used TruCluster technology features to achieve the maximum possible transactions per minute (tpm).[14] We balanced the I/O across all the RAID controllers and disks of the cluster and distributed the database across all the server nodes. We distributed the database such that each node in the cluster had an almost equal part of the database. The TPC-C benchmark execution requires a single database view across the cluster. We used the DRD and DLM services of the TruCluster software to present a contiguous view of the database across the cluster. If both the database and the indexes could have been completely partitioned, we could have achieved close to linear scaling per node. However, since the Oracle Parallel Server does not have horizontal partitioning of the indexes, we could not completely partition the indexes across the cluster.[15] This resulted in 15 percent to 20 percent of internodal access, which means that 15 percent to 20 percent of the new orders were satisfied by remote warehouses, therefore making our TPC-C results more realistic.

We also tuned the physical memory to trade off memory for database cache and the DLM locks. Heuristically, we observed a 40-percent gain in throughput on a single-node AlphaServer 8400 5/350 server system running TPC-C when the memory size was increased from 2 GB to 8 GB. This is because, with more data being served by memory, the number of processor stalls decreases, and the database-cache hit ratio improves from 88 percent to more than 95 percent.[16] Tuning physical memory beyond 2 GB is called very large memory (VLM). We used the tpm results of the AlphaServer 8400 system to tune the physical memory size and configuration. We show these measured tpm results for the AlphaServer 8400 cluster systems in Figure 8.

To achieve optimal server performance, it is important to tune the amount of memory used by the Oracle System Global Area (SGA) and the DLM. Our testing found that using VLM to increase the size of the SGA to 5.0 GB of physical memory yielded optimal performance in a TruCluster environment. However, it is important to note that on a single-node server that does not run the Oracle Parallel Server, we could assign 6.6 GB of physical memory to the SGA. (One reason that the SGA was smaller in an Oracle Parallel Server environment is that memory needed to be set aside for the DLM.) Consequently, as seen in Figure 8, the tpm on a single-node cluster system running the Oracle Parallel Server (8.4K tpm) is less than a single-node cluster not running the Oracle Parallel Server (11.4K tpm).

In an Oracle Parallel Server environment, we assigned 1 GB of memory to the DLM for the following reasons: The DLM, under the 64-bit Digital UNIX operating system, requires 256 bytes for each lock. In addition, the DLM must be able to hold at least one other location (and sometimes three) for lock callback. As a result, each lock requires between 512 bytes and 1 kilobyte (KB) of physical memory. To tune the system, we added more locks to increase the granularity of the locks and reduce lock contention. We observed that for this configuration, a system of this size supporting the Oracle Parallel Server requires 1 million locks (occupying 1 GB of memory) for the DLM when using 5.0 GB of memory for the SGA. Again heuristically, we observed that if we used less memory for the DLM, the total number of locks per page was reduced. The decrease in locks per page increases contention across nodes and hence reduces the tpm as the number of nodes increases.

With the help of engineers from Digital's MEMORY CHANNEL Group, we were able to use a hardware data analyzer to measure the percentage of the MEMORY CHANNEL interconnect's bandwidth used when running the TPC-C benchmark. By using the data analyzer, we determined that we do not approach saturation of the PCI-based MEMORY CHANNEL hardware during a TPC-C test, even though it is capable of sustaining a peak throughput rate of 100 MB/s. In fact, we observed that the MEMORY CHANNEL bandwidth was not saturated; a TPC-C test required a peak throughput rate of only 15 MB/s to 17 MB/s from the MEMORY CHANNEL. As stated previously, the benchmark specification forces 15 percent of the database accesses



Notes: 1. Each node is an 8-CPU AlphaServer 8400 5/350 cluster system.
2. The number preceding the X indicates a multiple of the tpmC measured on a single node running the Oracle Parallel Server.

Figure 8
TPC-C Results on the AlphaServer 8400 Family

to be remote, resulting in database accesses across the MEMORY CHANNEL. Using the DRD administration service available with the UNIX TruCluster software, we measured the DRD remote read percentages to match the 15-percent remote accesses rate. The DRD remote write performance was only 3 percent to 4 percent during the steady state and rose to 10 percent to 11 percent during a database checkpoint. It is important to note that the TPC-C benchmark performs random 2K I/Os using the Oracle Parallel Server. Small, random I/O transfers are much more difficult to perform than large, sequential transfers. Because the MEMORY CHANNEL interconnect not only has sufficient bandwidth for TPC-C but also provides excellent latency (less than 5 microseconds), we are able to report very good scaling results.

In the section TPC-C Benchmark, we discussed that the time taken for a checkpoint impacts the throughput. Therefore, we focused on improving the checkpointing time to increase the tpmC number. First, we used a dedicated PCI bus on each node for the MEMORY CHANNEL interconnect and thus obtained a 5-percent improvement in performance during checkpointing. Next we implemented the highly optimized "fastcopy" routine in DRD, which packs data on the PCI when transmitting through the MEMORY CHANNEL interconnect.

Performance Measurement Results

In this section, we present our results for the TruCluster configuration running the TPC-C workload and compare them with results from competitive vendors. We conducted the test on a database with 2,592 warehouses and 25,920 emulated users. The database was equally divided, which means each node contained 648 warehouses and served 6,480 emulated users. We show the initial cardinality of the database tables in Table 2. The cardinality of the history, orders, new-order, and order-line tables increased as the test progressed and generated new orders. We conducted the experimental runs for a minimum of 160 minutes.[10] The measurement on the SUT began approximately 3 minutes after the simulated users had begun executing transactions. The measurement period of 120 minutes, however, started after the SUT attained a steady state in approximately 30 minutes. In agreement with the TPC-C specification, we performed 4 checkpoints at 30-minute intervals during the measurement period.

On the SUT, we measured a maximum throughput of 30,390.65 tpmC, which unveiled a new record high in the competitive market for database applications and UNIX servers. We repeated the experiment once

Table 2
Initial Cardinality of the Database Tables

Warehouse	2,592
District	25,920
Customer	77,760,000
History	77,760,000
Order	77,760,000
New order	23,328,000
Order lines	777,547,308
Stock	259,200,000
Item	100,000

more to ensure the reproducibility of the maximum measured tpmC. Digital Equipment Corporation and Oracle Corporation also present a price/performance ratio of $305 per tpmC.

In Table 3, we present the total occurrences of each transaction type and the percentage transaction mix used to generate the transactions in each test run. We compare the percentage transaction mix in Table 1 and Table 3 and observe that our measurements are in agreement with the TPC-C specification. We present the 90th percentile response time measured for each transaction type in Table 4. The 90th percentile response time we measured is well below the TPC-C specification requirement (compare Table 1 and Table 4). In Table 5, we present the minimum, average, and maximum keying and think times. Again, we comply with the TPC-C specification (compare Table 1 and Table 5).

Now we compare the maximum throughput achieved on the AlphaServer 8400 5/350 four-node TruCluster configuration with results from Tandem Computers and from Hewlett-Packard Company (HP).[17] The Tandem nonstop Himalaya K10000-112 112-node cluster reported 20,918.03 tpmC at $1,532 per tpmC. Observe that Digital's measured tpmC are 45 percent higher than Tandem's, and Digital's price/performance is 20 percent of Tandem's cost. In Figure 9, we compare Digital's performance with HP's. The HP 9000 EPS30 C/S 48-CPU four-node cluster system using the Oracle Parallel Server Oracle7 version 7.3 reported 17,826.50 tpmC at $396.[18] Again, observe that the tpmC we measured on Digital's TruCluster configuration are 59 percent higher than HP's at 77 percent of the cost.

Table 3
Measured Total Occurrences of Each Transaction Type and Percentage Transaction Mix

Transaction Type	Total Occurrences	Percentage in Mix
New order	3,645,228	44.47
Payment	3,540,119	43.19
Order status	336,255	4.10
Delivery	337,423	4.12
Stock level	337,730	4.12

Table 4
Measured 90th Percentile Response Time

Transaction Type	90th Percentile Response Time
New order	3.4
Payment	3.2
Order status	0.9
Delivery (interactive)	0.4
Delivery (deferred)	5.0
Stock level	1.7

Table 5
Measured Keying/Think Times

Transaction Type	Minimum (Seconds)	Average (Seconds)	Maximum (Seconds)
New order	18.0/0.00	18.1/12.2	18.8/188.1
Payment	3.0/0.00	3.1/12.1	3.7/201.4
Order status	2.0/0.00	2.1/10.1	2.7/125.6
Delivery	2.0/0.00	2.1/5.2	2.7/74.9
Stock level	2.0/0.00	2.1/5.2	2.7/62.7

Conclusion and Future Work

In this paper, we discussed the performance evaluation of Digital's TruCluster multicomputer system, specifically the AlphaServer 8400 5/350 32-CPU, four-node cluster system, under the TPC-C workload. For completeness, we gave an overview of TruCluster clustering technology and the TPC-C benchmark. We discussed tuning strategies that took advantage of TruCluster technology features like the MEMORY CHANNEL interconnect, the DRD, and the DLM. We tuned memory to use VLM for the database cache and made memory allocation trade-offs for DLM locks to reduce processor stalls and improve cache hit ratios.

One common concern is performance scalability of cluster systems, that is, incremental performance growth with the size of the cluster. In Figure 8, we showed the measured performance of an SMP server, both with and without the Oracle Parallel Server, and cluster configurations with two, three, and four SMP servers. We do not see linear scaling because the Oracle Parallel Server imposes a significant amount of overhead on each cluster node. This overhead equates to approximately a 25-percent reduction in tpmC on a per node basis. However, it is important to note that due to the time constraints of obtaining audited results for the product announcement, the testing team was unable to fully tune the server and saturate the server CPUs. In future testing, additional performance tuning is planned to further optimize server performance.

The performance testing of the TruCluster multicomputer system was time-consuming and expensive. Thus, answering "what if" questions regarding sizing and tuning of varying cluster configurations under different workloads using measurements is an expensive (with respect to money and time) task. To address this problem, we are developing an analytical performance cluster model for capacity planning and tuning.[10] The model will predict the performance of cluster configurations (ranging from two to eight members) with varying workloads and system parameters (for

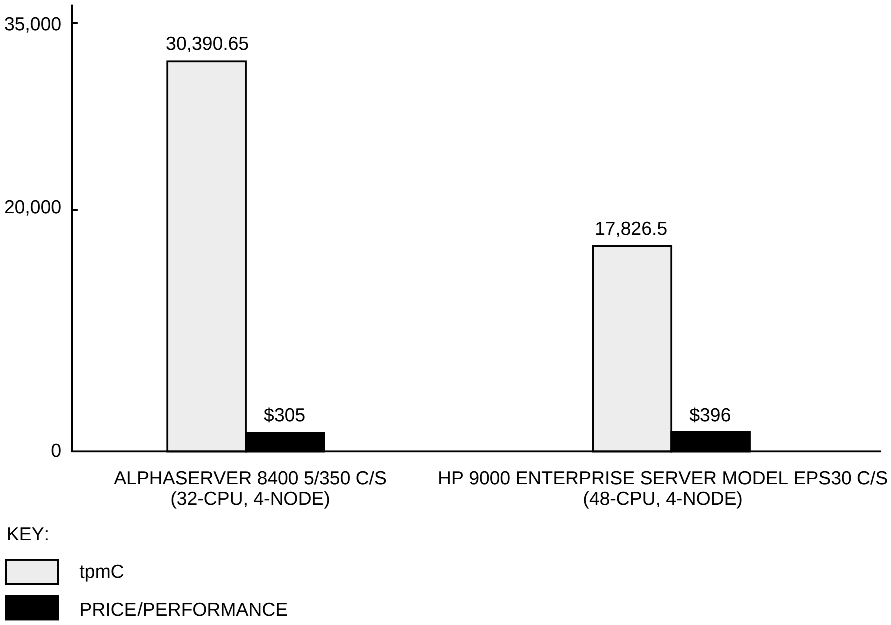


Figure 9
Comparison of TPC-C Results

example, memory size, storage size, and CPU power). We will implement this model in Visual C++ to develop a capacity planning tool.

Acknowledgments

Many people within several groups and disciplines in both Digital and Oracle contributed to the success of this performance project. We would like to thank the following individuals from Digital: Lee Allison, Roger Deschenes, Tareef Kawaf, Maria Lopez, Joe McFadden, Bhagyam Moses, Ruth Morgenstein, Sanjay Narahari, Dave Stanley, and Greg Tarsa of the CSD Performance Group; Brian Stevens and Tim Burke of the Digital UNIX Engineering Group; Jim Woodward, Digital UNIX Performance Team member; Sean Reilly, Doug Williams, and Zarka Cvetanovic of the AVS Performance Group; and Don Harbert and Pauline Nist, the test sponsors. Lastly, we would like to thank Jef Kennedy, Peter Lai, Karl Haas, and Vipin Gokhale of the Oracle Performance Group.

References and Notes

1. Throughout this paper, we use the term cluster interchangeably with TruCluster.

2. W. Kohler, A. Shah, and F. Raab, “Overview of TPC Benchmark C: The Order Entry Benchmark” (Transaction Processing Performance Council, Technical Report, December 1991).

3. Transaction Processing Performance Council, *TPC Benchmark C Standard Specification, Revision 3.0,* February 1995.

4. S. Leutenegger and D. Dias, “A Modeling Study of the TPC-C Benchmark,” *ACM SIGMOD Record,* vol. 22, no. 2 (June 1993): 22–31.

5. R. Gillett, “MEMORY CHANNEL Network for PCI,” *IEEE Micro,* vol. 16, no. 1 (February 1996): 12–19.

6. *TruCluster for Digital UNIX Version 1.0* (Maynard, Mass.: Digital Equipment Corporation, Software Product Description 63.92, October 1995).

7. W. Cardoza, F. Glover, and W. Snaman Jr., “Design of the TruCluster Multicomputer System for the Digital UNIX Environment,” *Digital Technical Journal,* vol. 8, no.1 (1996): 5–17.

8. *TruCluster Software, Hardware Configuration* (Maynard, Mass.: Digital Equipment Corporation, Order No. AA-QL8LA-TE, December 1995).

9. *TruCluster: Software Installation and Configuration* (Maynard, Mass.: Digital Equipment Corporation, Order No. AA-QL8MA-TE, September 1995).

10. Checkpointing is a process to make the copy of the database records/pages on the durable media current; systems do not write the modified records/pages of the database at the time of the modification but at some deferred time.

11. This cost includes the hardware system cost, the software license charge, and the maintenance charges for a five-year period.

12. The AlphaServer 1000 4/266 system can be configured with as much as 1 GB of memory. Due to a supply shortage of denser error correction code (ECC) memory, the clients in the SUT could be configured with a maximum memory of 512 MB.

13. Digital Equipment Corporation and Oracle Corporation, “Digital AlphaServer 8400 5/350 32-CPU 4-Node Cluster Using Oracle7, Tuxedo, and Digital UNIX,” TPC Benchmark C Full Disclosure Report filed with the Transaction Processing Performance Council, April 1996. Also available from the TPC Web page.

14. Note that these results were not audited; per TPC-C specification, we refer to them as tpm instead of tpmC.

15. Horizontal partitioning of the indexes allows the user to have each node in the cluster store indexes that are mapped only to tables that are local.

16. T. Kawaf, D. Shakshober, and D. Stanley, “Performance Analysis Using Very Large Memory on the 64-bit AlphaServer System,” *Digital Technical Journal,* vol. 8, no. 3 (1996, this issue): 58–65.

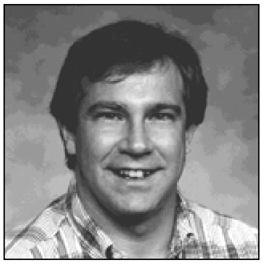
17. These results were withdrawn by Tandem on April 12, 1996, and hence are not included in Figure 9.

18. Hewlett-Packard Company, General Systems Division, and Oracle Corporation, “HP 9000 Enterprise Parallel Server Model EPS30 (4-Node) Using HP-UX 10.20 and Oracle7,” TPC Benchmark C Full Disclosure Report filed with the Transaction Processing Performance Council, May 1996. Also available from the TPC Web page.

Biographies

Judith A. Piantedosi
A principal software engineer in the CSD Performance Group, Judy Piantedosi evaluates I/O performance on Digital UNIX systems, specializing in characterizing NFS file servers. Judy is the project leader of the TruCluster capacity planning modeling effort and Digital’s technical representative to the Standard Performance Evaluation Corporation (SPEC) System File Server (SFS) Subcommittee. Judy joined Digital in 1987 to help solve customer hardware/software problems when using System V. Before joining Digital, Judy was employed at Mitre Corporation. She was the lead software designer on the Joint STARS Radar Evaluation Activity, a radar simulation built to provide proof of concept to the U.S. Air Force for the Joint STARS project. She was responsible for implementing several radar models into the simulation. Judy holds a B.A. from Boston College (1984).

Archana S. Sathaye
Archana Sathaye is currently a consultant to Digital in its CSD Performance Group. From 1987 to 1994, she was an employee of Digital and worked on several reliability, availability, and performability modeling projects for OpenVMS Cluster systems and other high-end CPU products. She resigned from Digital and accepted a position as adjunct assistant professor in the Department of Electrical and Computer Engineering at the University of Pittsburgh. Archana holds a Ph.D. in electrical and computer engineering from Carnegie Mellon University (1993); an M.S. from Virginia Polytechnic and State University (1986), a B.Sc (1981) and an M.Sc (1983) from the University of Bombay, India, all in mathematics. She is an affiliate member of ACM SIGMETRICS and has authored or coauthored several papers on reliability, availability, and performability modeling and control synthesis.



D. John Shakshober
John Shakshober is the technical director of the CSD Performance Group. The Computer Systems Division Performance Group evaluates Digital’s systems against industry-standard benchmarks such as those of the Transaction Processing Performance Council (TPC) and the Standard Performance Evaluation Corporation (SPEC). In this function, John has been responsible for integrating Digital’s state-of-the-art software technologies with Digital’s Alpha-based products since their introduction in 1992. Prior to joining the CSD Performance Group, John modeled the performance of the 21064 and 21164 Alpha 64-bit VLSI microprocessors and was a member of the VAX 6000 Hardware Group. He joined Digital in 1984 after receiving a B.S. in computer engineering from the Rochester Institute of Technology. John also received an M.S. in electrical engineering from Cornell University in 1988.