Order-of-Magnitude Advantage on TPC-C Through Massive Parallelism

Charles Levine
Tandem Computers Inc.
10555 Ridgeview Ct., LOC 252-10
Cupertino, CA 95014
levine_charles@tandem.com

Introduction

TPC Benchmark™ C (TPC-C) is the modern standard for measuring OLTP performance. Running TPC-C, Tandem demonstrated a massively parallel configuration of 112 CPUs which achieved ten times higher performance than any other system previously measured (and today is still better by a factor of five). This result qualifies as the largest industry-standard benchmark ever run.

This paper briefly describes how the benchmark was configured and the results which were obtained.

Results

In 1994, Tandem announced TPC-C results for four configurations of the Himalaya K10000 server. The four results demonstrated the system's scalability and linearity over a large performance range. Indeed, the largest system had 112 CPUs and 1.4 TB of disk capacity. The results are shown in Table 1.

TABLE 1: Himalaya K10000 TPC-C Performance

# of CPUs	tpmC	$/tpmC	Announce Date
16	3,043	1,598	April 25, 1994
32	6,067	1,552	May 3, 1994
64	12,021	1,528	May 17, 1994
112	20,918	1,532	July 5, 1994

Architecture

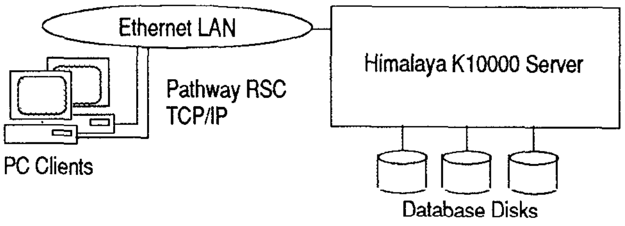
Tandem NonStop Kernel systems are designed around a loosely-coupled shared-nothing architecture. Each CPU has its own memory and I/O channels. CPUs within a node communicate over a packet-based interprocessor bus. A node may have up to 16 CPUs. All process-to-process communication occurs via message passing. This design facilitates fault tolerant operation because any component can fail without affecting the correct operation of the rest of the system.

The Himalaya K10000 runs the MIPS R4400 microprocessor. As used in the benchmarks, the CPU runs at 150 MHz with 4MB of secondary cache and 128 MB of main memory.

Configuration

Our TPC-C tests used a client-server configuration. Presentation services were performed by the client systems; the server executed all database functions. Clients sent transactions to the server using Tandem's RSC (Remote Server Call) product, an extension of the NonStop TS/MP transaction monitor. The client-server link used TCP/IP over an Ethernet LAN. The configuration is shown in Figure 1.

FIGURE 1: Client-server Configuration



The TPC-C database consists of nine tables. The hierarchical design of the TPC-C database lends itself to horizontal partitioning. In Tandem's implementation, all the tables were horizontally partitioned using range partitioning based on primary key. Partitioning the tables accomplishes two important objectives: (1) it spreads the physical I/O load over multiple disks, and (2) it spreads the processing load across the CPUs.

SIGMOD ' 95,San Jose , CA USA
© 1995 ACM 0-89791-731-6/95/0005..$3.50

TABLE 2: I/O Statistics (Approximate)

System	Logical I/Os per sec	Physical I/Os per sec	SQL Selects per sec	SQL Updates per sec	SQL Inserts per sec	SQL Deletes per sec	SQL statements per sec
K10000-16	12,700	1,600	1,700	800	650	50	3,200
K10000-32	25,400	3,200	3,400	1,600	1,300	100	6,400
K10000-64	50,300	6,400	6,700	3,200	2,600	200	12,700
K10000-112	87,600	11,100	11,700	5,500	4,500	350	22,050

The TPC-C database scales linearly with throughput. Thus, to double the throughput requires a database twice as large. For high transaction rates, building the database can be very time consuming if not done in parallel. We solved this problem by designing the database generator program to load a horizontal slice of the database. By running *n* generator processes in parallel, each loading a subset of the total database, the elapsed time was reduced by approximately *1/n*. The size of the database increased as CPUs were added, but the size of the database per CPU remained constant. Consequently, the database took about 6-8 hours to build for all the configurations from 16 to 112 CPUs. This is a good example of scaleup.

Our TPC-C implementation used alternate key indexes on the CUSTOMER and ORDER tables. Because these tables are relatively large (each had 58 million rows in the largest configuration), creating indexes would take a long time if done serially. By using parallel create index, the index build time remained approximately constant at 20 to 30 minutes regardless of database size. This is another good example of scaleup.

Scalability and Linearity

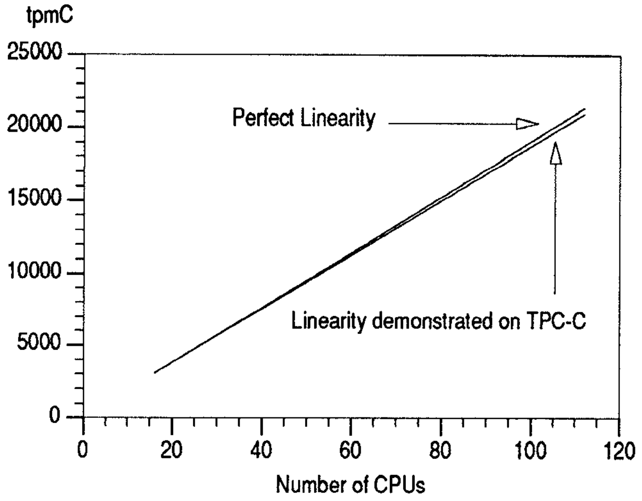
Tandem's TPC-C results demonstrate scaleup and linearity. Figure 2 shows that the linearity achieved was within 2% of perfect linearity. The excellent linearity in the Tandem system results from the shared-nothing design combined with the message based operating system. The system overhead per transaction remains essentially constant regardless of the number of CPUs.

Table 2 shows statistics on the I/O activity and SQL operations per sec. The largest configuration executed approximately 11,100 physical I/Os per sec.

Personnel

The effort required to do this benchmark merits some discussion. TPC-C has substantial start-up costs. Indeed, to produce the first result many components have to be developed such as the benchmark application, database generator, and RTE (Remote Terminal Emulator) driver. In addition, the results must be audited and a full disclosure report written.

FIGURE 2: Linearity Graph



These benchmarks were Tandem's first TPC-C results. The work running the tests was accomplished by two engineers over about four months. This included many of the startup costs described above, although the software was mostly completed beforehand. More interestingly, after completing the first result, subsequent tests took about one week to run and one week to complete the audit and full disclosure report.

Conclusion

Applying a massively parallel system to TPC-C, Tandem has demonstrated performance an order of magnitude greater than conventional SMP systems. Furthermore, the system scales with near-linear performance over a wide range.