Oracle Parallel Technology Empowers AIX Systems

By Sandra Lee and Annie Chen

This article describes how Oracle's parallel technology fully exploits the power of IBM's HACMP/6000 software and SPx hardware (IBM SP1™ and SP2 RISC-based parallel processors). By using parallel database functions—from database backup and restore to the query process itself—the Oracle7 database builds on the strengths of IBM architectures to improve database performance. Although processing is shared across multiple CPUs, users still manage one database with parallel functions for better performance.

Today's businesses need to harness enterprise-wide data. Computing environments are being pushed to their limits by business applications that are growing larger and more complex, requiring more sophisticated computer systems with greater capacity. Many users eliminate data because they cannot economically store it for later access. Other users base their business decisions on data that is days old because their current systems do not have the horsepower to provide real-time analyses.

Oracle has created new solutions to meet these business needs, which fall into three general areas of computing: data mining (queries) on huge amounts of data, high availability, and improved backup/restore and database performance.

Not only does Oracle support different IBM architectures, but Oracle's parallelism addresses the business needs discussed above and enhances performance in all three areas of computing. For example, the Oracle Parallel Server provides high availability in an HACMP/6000 cluster; in an SPx system, the Parallel Server also provides parallel query ability, enhancing the IBM technology. Oracle's parallel technology is a logical next step for companies that need to enhance their computer systems to better meet their business needs.

Oracle Parallel Server

The Oracle7 Parallel Server (OPS) technology has been implemented on IBM's HACMP/6000 and SPx systems. With OPS, multiple instances of the Oracle database server run concurrently and independently. Each runs on its own processing node with its own Shared Global Area (SGA) memory for a database buffers cache and its own set of Oracle7 background processes (including the system monitor, process monitor, and database write) for backup and recovery. Any number of Oracle7 instances can access the same Oracle7 database files and control files on disk, which collectively form an Oracle7 Parallel Server. In Symmetric Multiprocessing (SMP) systems, IBM plans to accomplish this through shared memory; in HACMP/6000, it is done through shared disks in clusters up to four processors; and in the SPx family, it is done through the high-performance switch.

Figure 1 illustrates the Oracle Parallel Server architecture. Each Oracle7 instance concurrently handles the database requests of multiple clients. All Oracle7 instances can execute transactions concurrently against the same database, enabling users to focus the processing power of multiple CPUs (whether loosely or tightly coupled) against the database.

Benefits of Single-Database Design

Oracle's single-database design (as opposed to multiple databases or multiple database partitions) scales up system performance by simply adding nodes and disks. Since all nodes have direct access to the entire common database (instead of

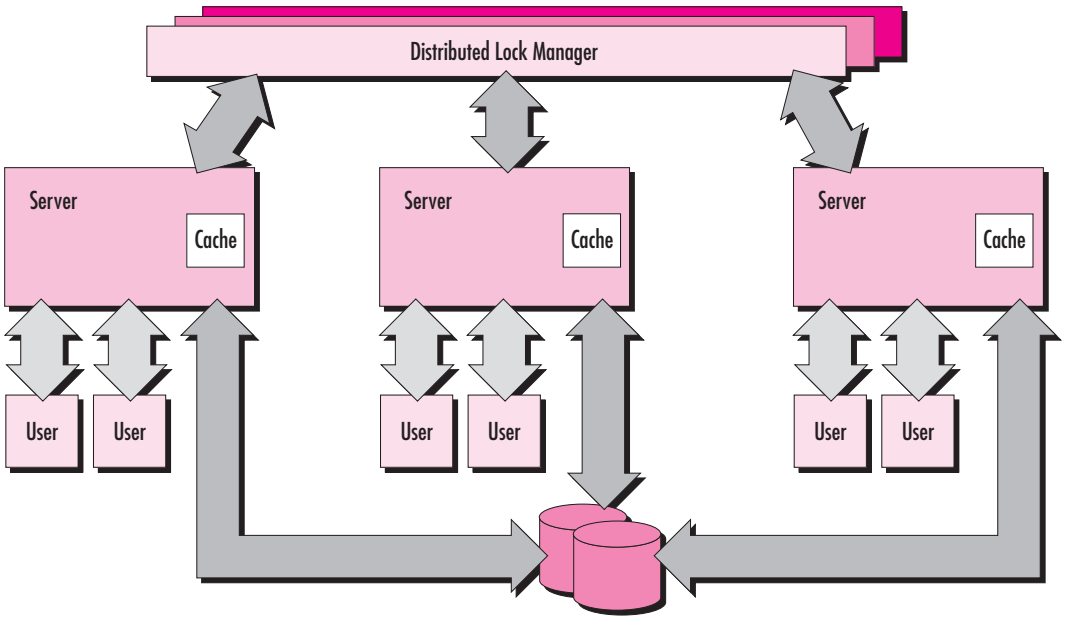

Figure 1. Oracle Parallel Server architecture

each node having exclusive access to a small partition of the database), adding nodes immediately increases system throughput. Because data does not need to be repartitioned across new numbers of nodes and no applications need modification, the administrative work involved in scaling up performance is much simpler. The single-database design further simplifies database administration since there is only one database to start up, shut down, back up, and monitor.

Easy Migration from Uniprocessing to HACMP/6000

Oracle's parallel database environment completely hides its parallelism from users. Since the OPS software uses the identical SQL interface in every Oracle7 database system, the user sees and interacts with the familiar standard Oracle7 database server. Parallelism requires no new commands or extensions to existing commands; the OPS technology handles parallelism and optimizes system resource utilization automatically. The results are twofold: all Oracle tools and applications run unchanged, and neither application developers nor end users need retraining.

Increased Availability

The single-database architecture results in increased data availability. Because each node has full access to the entire database, losing a node does not mean losing access to a part of the database. If a node fails, one of the surviving nodes will automatically detect the failure, recover any work that was in progress on the failed node, and continue processing all client requests. Oracle7 currently supports all modes of HACMP/6000, including concurrent access. Figure 2 shows an HACMP/6000 cluster.

In an HACMP/6000 cluster running in concurrent access mode, the OPS enables different nodes in the cluster to share an application. This means that the HACMP/6000 cluster can run larger Oracle applications than a single RISC System/6000 can handle. This capability breaks the single machine barrier for Relational Database Management System (RDBMS) performance, achieving scalable high performance for many types of applications. Tests have shown that the increase in performance on two-node HACMP/6000 clusters with Oracle7 and HACMP ranges from 1.6 to 1.9 times the performance of the uniprocessor configuration, depending on the application.

Exploiting the Power of the SPx Family

Oracle plans to take high availability one step higher in IBM's massively parallel SPx machines. In an SPx system, the nodes can be divided logi-

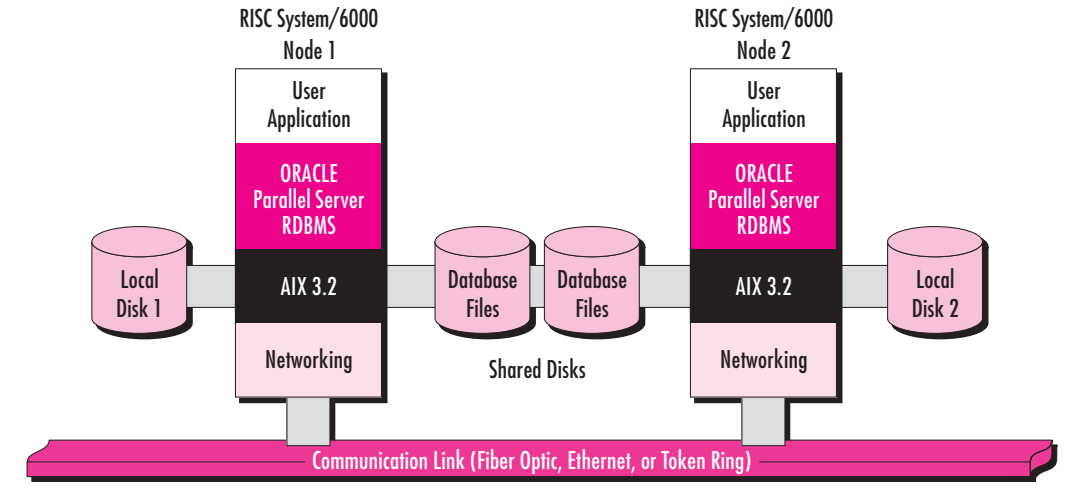

Figure 2. HACMP/6000 cluster

cally into two categories: data servers that host the Oracle database, and application/compute servers where applications reside. An instance of Oracle7 can run on each of the 8 to 512 data server nodes. All instances of Oracle7 have access to one physical database that is located on the SPx's Virtual System Disk (VSD). Any user or application can access any portion of the database from any instance or node of the machine with the same response time. All instances in a parallel server share the data files and control files, but each instance has its own redo log files (known as *threads of redo*).

If an Oracle7 instance or an SPx node fails on an SPx system with dual disk access, the other instances continue to access all portions of the database. First, the SPx will automatically recover the node or the VSD component and then pass control to Oracle7, which will automatically recover on behalf of the failed instance. Another Oracle instance ensures that data committed by the failed instance is written to the database and all uncommitted transactions are rolled back.

Parallel Cache

To further improve performance in an HACMP/6000 cluster, the OPS uses parallel cache management enabled by the cluster's Distributed Lock Manager (DLM). As shown in Figure 3, each node in a cluster has a local cache containing database blocks that have been recently accessed by transactions running on that node. After execution, the blocks are retained in the cache so subsequent transactions can readily access them. Database blocks are removed from the cache to make room for new blocks using a Least Recently Used algorithm. When a node requires a database block that is not already in its cache, it uses the DLM and parallel cache management facilities to determine whether the block is in the cache of another node. If it is, the parallel cache manager coordinates the fast transfer of the database block from the other node. If no other node has the

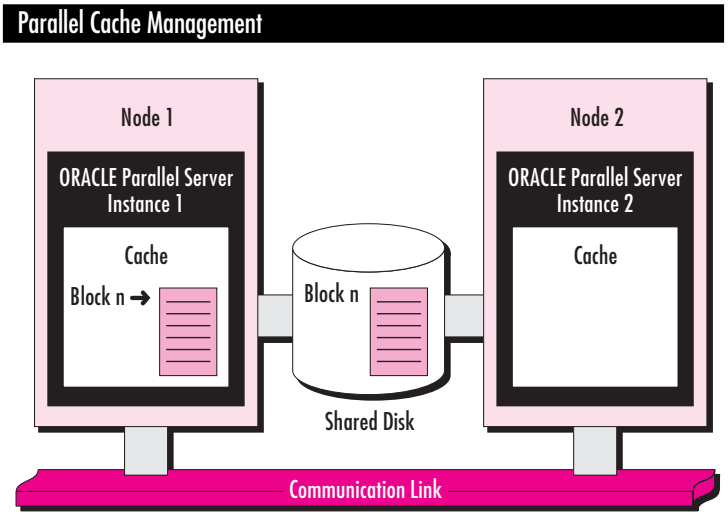

Figure 3. Optimizing parallel cache management

IBM SPx and Gigacache Nodes

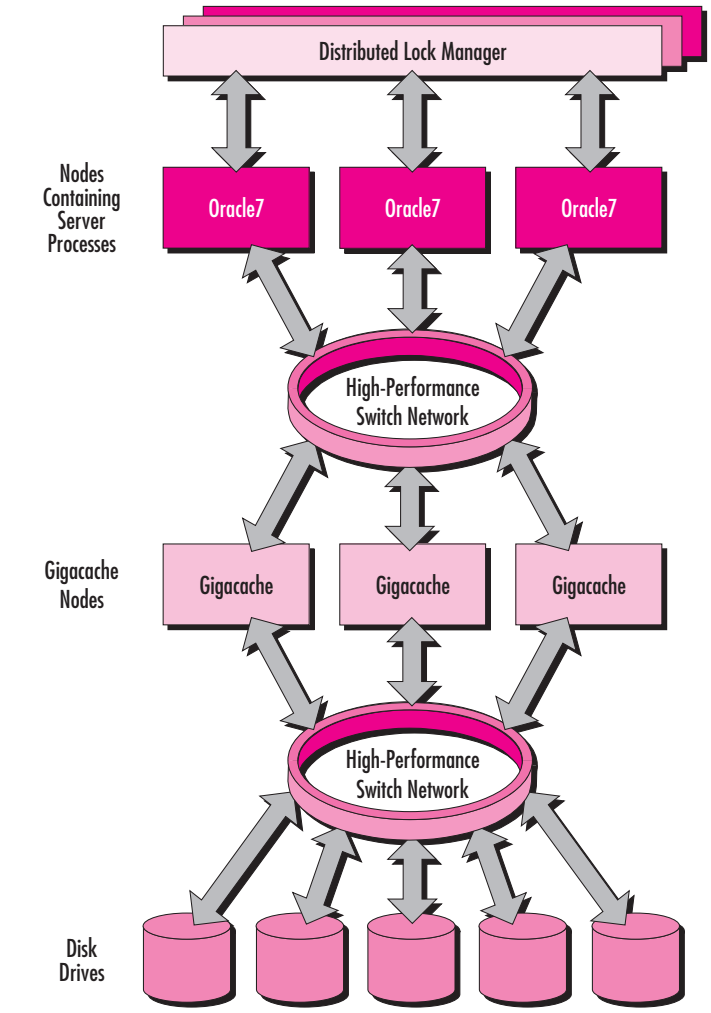

Figure 4. IBM SPx and gigacache nodes

database block, it is read directly from the database.

To increase performance and reduce overhead, the OPS does not release DLM locks at the end of a transaction. It holds the lock until another node requests a DLM lock on the same block. In business applications, especially with disjoint sets of data, the probability of an instance reusing a block that it just accessed is much higher than the chance of another instance asking for the same block. This parallel cache management technique makes the OPS very fast and efficient.

On an SPx system, the OPS speeds parallel cache management even more by reducing the amount of time needed for one instance of Oracle7 to obtain a database block from the cache of another node.

Some SPx processing nodes, each with its own dedicated memory, can function optionally as I/O nodes or *gigacache* nodes—a distributed random-access-memory disk cache that holds some or all of the database being updated or queried. Oracle7 can establish default configurations that designate certain nodes as servers and others as gigacache, as shown in Figure 4. These gigacache processors can reduce the read/write delays imposed by the database block transfer for cache coherency that would otherwise require disk read. In real world applications that frequently access data "hot spots," the gigacache can drastically reduce I/O requests from the parallel cache manager.

Parallel Query

Cache speed is not the only area in query processing that may affect response time. A query on two 500,000 row tables may ultimately provide a single-row answer. But in the computation stage, the end result requires a complex sifting through the tables and performing sorts, aggregates, and joins. To speed this process, operations such as sorts, scans, and joins are parallelized with Oracle 7.1's Parallel Query Option (PQO) for SPx. Different nodes and instances of an SPx can work on different operations in parallel and send the results back to the query coordinator, which resides on a separate node (see Figure 5). Since all the data resides in a central database, the Query Execution Plan can be dynamic, based on the data accessed. This is unlike other RDBMS implementations, which statistically fragment the database when it is created.

Oracle's unique single database approach enables the definition of processor node classes on an SPx. One set of nodes can be defined to process Online Transaction Processing (OLTP) application requests while a different group of nodes processes complex queries. Since each node can directly access any portion of the entire database and execute instructions independently of other nodes in the system, the nodes running complex queries will not drain CPU processing power from the OLTP nodes.

Oracle's PQO is composed of parallel scan, parallel join, and parallel sort technologies that enable multiple processor nodes to automatically share the workload of a single, large, complex

Oracle's Parallel Query

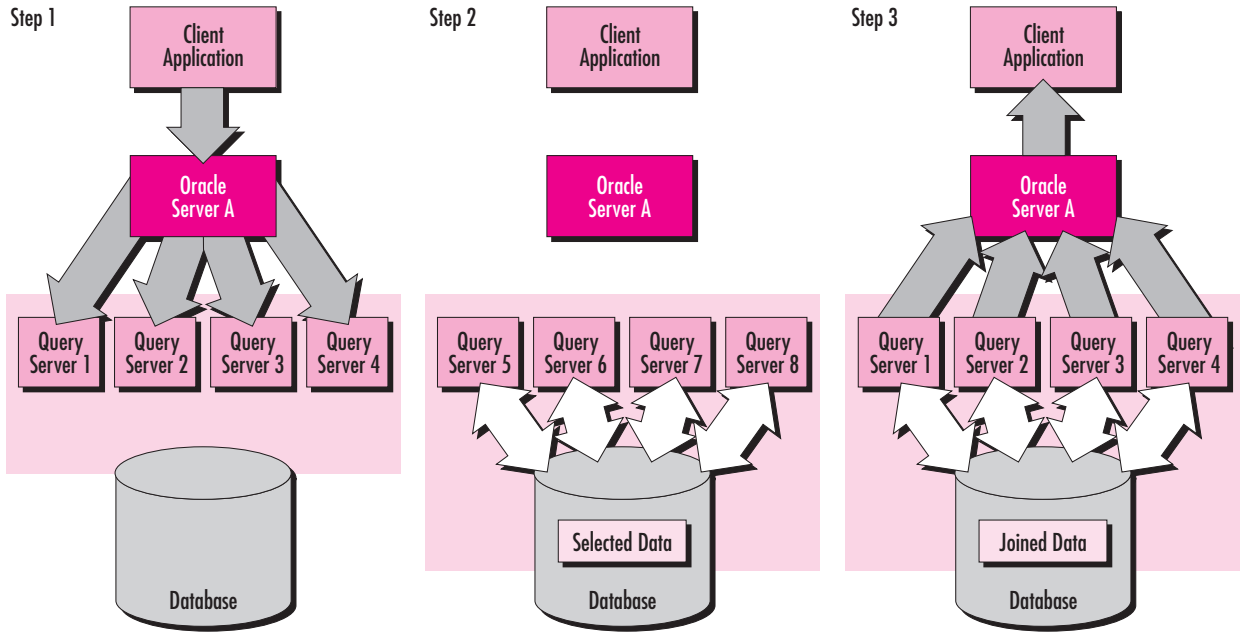

Figure 5. Oracle's parallel query in action

query. With parallel scan, the nodes work in parallel to search through different portions of the database, reducing the work for each node and improving performance. Parallel join enables multiple nodes to scan separate data tables, and then join the selections in parallel to quickly provide the final answer. Parallel sort divides the data into multiple pieces so that each node can sort a small portion of the data. The nodes then work together to quickly combine the smaller sorted lists into one sorted final result.

In the screens in Figures 6 and 7, the benefits of PQO are clearly visible. An Oracle7 database was run on an SPx system with eight nodes—first without the PQO, then with the PQO. The graphical front-end in the screens depicts the results of the experiment pictorially. The clock indicates the running time of the query, while the bar graphs show the CPU load plus I/O (plotted from system variables) for each node in the SPx.

When querying a database sequentially, one CPU must often wait for another CPU to finish processing the initial portion of the query before it can process its portion. Since only some CPUs are used during a query, the system is not used

CPU Usage Without PQO

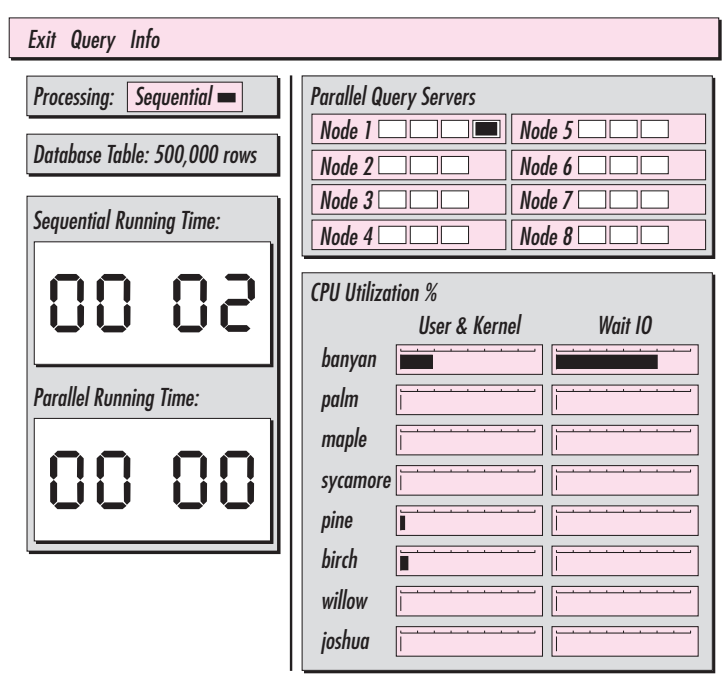

Figure 6. CPU usage without Parallel Query Option

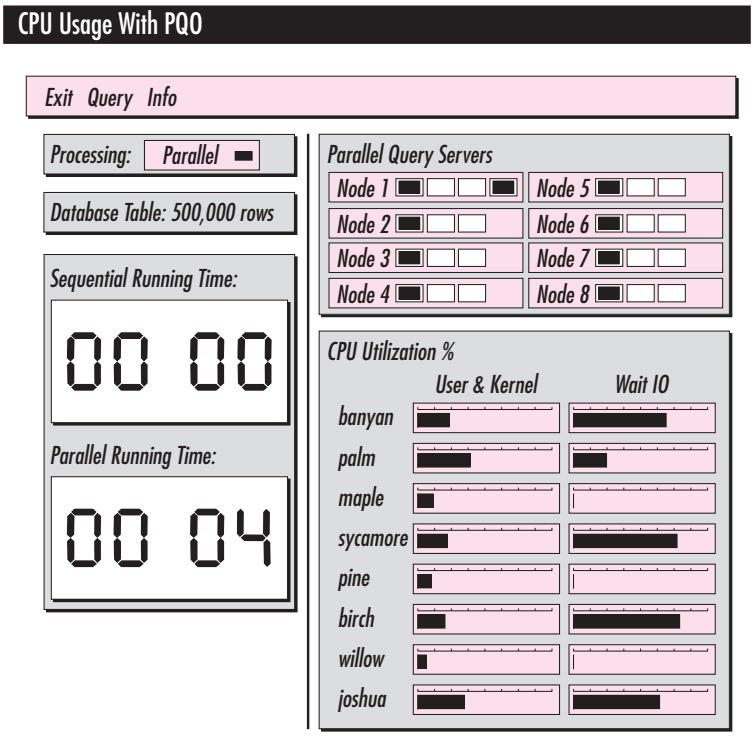

Figure 7. CPU usage with Parallel Query Option

to its capacity, resulting in inefficiency and slower performance.

Figure 6 shows CPU usage during sequential processing of a query. Only a few CPUs are used (banyan, pine, birch); the rest are sitting idle (palm, maple, sycamore, willow, joshua). The query is not taking advantage of all the available CPU power.

Figure 7 shows the result when the PQO is enabled. A query is split into several parts, and a separate CPU works on each part. The data is scanned and sorted in parallel, using all CPUs and decreasing response time, making more efficient use of the available processing power and increasing performance.

Parallel scan, parallel sort, and parallel join deliver exceptional query performance on massively parallel systems, such as the SPx, because of the many processors available to work on the query.

Parallel Load and Parallel Index

For bulk loading of external data into tables of a database, Oracle 7.1's Parallel Direct Load enables users to start multiple concurrent load processes directed to the same table from multiple processing nodes. Parallel Direct Load uses many resources (such as disks and tape drives) to load data into the same table in parallel from different tape drives. Once the data is loaded, index creation can be divided across several data server nodes.

Like PQO, Oracle's parallel index feature will automatically use parallelism if the base table is spread over multiple database files across processing nodes. After the parallel table scan on the database files of the base table, the base table is split into partitions. Each partition contains the keys from the base table for a disjoint portion of the entire index key range. An index is created on each partition simultaneously, and eventually these indexes are merged into a single permanent index.

Parallel Backup and Recovery

The Oracle Parallel Backup/Restore utility and Oracle 7.1's parallel recovery can drastically reduce the time to back up, restore, and recover very large databases—tens to hundreds of gigabytes in size. Available on IBM's SP2 with Oracle 7.1, it will enable multiple data files and tablespaces to be backed up online to different media devices in parallel. The utility works with a third-party media management tool to enable a data center to back up the enterprise—including the Oracle data servers, compute servers, filesystems, and clients—with the same media management software. This is a robust alternative to less reliable, slower utilities, such as `dd`, `cpio`, and `tar`. Once a restore is done with the Backup/Restore utility, Oracle 7.1's parallel recovery feature can be applied, providing a read-ahead capability of the log files in parallel.

Sandra Lee, *Oracle Corporation, 500 Oracle Parkway, Box 659406, Redwood Shores, CA 94065. Internet: shlee@us.oracle.com. Ms. Lee is the product line manager for the AIX platform at Oracle. She has BA degrees in Computer Science and English from the University of California at Berkeley and a graduate degree from Boston University.*

Annie Chen, *Oracle Corporation, 500 Oracle Parkway, Box 659406, Redwood Shores, CA 94965. Ms. Chen is a development manager in the UNIX products division. She has an MS in Computer Science from the University of Pittsburgh in Pennsylvania.*