Data Warehousing on the Cray CS6400: Oracle's Test-to-Scale Benchmark

Brad Carlile, Cray Research, Inc., Business Systems Division

1 INTRODUCTION

Consolidating enterprise-wide data located in disparate databases into a data warehouse provides an opportunity for many companies to develop a competitive advantage. Data warehouses are large repositories of corporate data that can often require Terabytes of data storage. Decision Support System (DSS) is the complete system used to learn more about this data within the warehouse and highlight previously un-explored relationships in large databases. Finding the important data in a data warehouse involves the judicious use of detail data, summarized data, and meta-data (data about the data). Summary data can dramatically speed important basic queries but often this must be balanced with being able to explore all the detail data without losing information due to over-summarization.

Today more and more corporations are planning databases in the Terabyte range. A Terabyte is a literal mountain of data. A book containing one Terabyte of information would be 34,745 feet thick (10590 meters) or 6.6 miles (10.6 kilometers) [note1]-- taller than the summit of Mt. Everest at 29,208 feet (8902 meters). Currently, most large production databases contain tables of several hundred Gigabytes (GB or Gbytes). The use of indexes, disk mirroring, RAID-5, and redundant systems can however increase the amount of disk required by 5 or 10 times.

Oracle created the Test-to-Scale Benchmark to show the Terabyte capabilities of Oracle7. This Oracle certified data warehouse benchmark runs summary table creation, 6-way table joins, drill-down queries, star queries, and index creation on a 1.3 Terabyte database of actual data (not counting temporary indexes or mirroring) the largest table in this benchmark is almost 700 Gbytes and has 6 billion rows. To date this benchmark has been certified on 3 SMP vendors and no MPP vendors. We believe this is the highest performance Test-to-Scale results. The table schema is based on the Transaction Processing Council's decision support benchmark (TPC-D) Scale Factor 1000 (SF1000) database. The SF1000 database contains roughly 1 Terabyte of data (each Scale factor is approximately 1 Gbyte, SF1000=1Tbyte). The TPC-D queries were not run on this benchmark.

In addition, Cray also created another database with 1.9 Terabytes in a single table with 9.3 billion rows. This is the largest Oracle Database in terms of both data volume and number of rows. Several different queries were run on this database to demonstrate full table scan and aggregate performance. This was a test to demonstrate the high capacity of the Oracle7 database. A single large table was created to serve as a stress test of the databases internal structures. The database schema was based on the Wisconsin benchmark schema.

This paper presents Cray's results on the Test-to-Scale benchmark demonstrating high performance, linear scaleup, and excellent scalability on a very large database. The CS6400 is Cray's SPARC-based SMP System that runs the Solaris 2.4 operating system and is fully SPARC binary compatible. The combination of the CS6400 and the parallel features of Oracle7 provide scaleable performance for DSS operations on databases ranging from a Gigabyte [2] to multiple Terabytes in size [13].

2 THE EVOLUTION OF THE DATA WAREHOUSE

A data warehouse combines data from databases dispersed throughout an enterprise. The term Decision Support Systems (DSS) describes the entire system involved in performing analysis of data in a database that supports decision making. Due to the differences between a transactional workload and an analysis workload, DSS operations run most efficiently on a data warehouse architected for DSS. The purpose of performing DSS on a data warehouse is to derive useful information from the large corporate data repository. With this information there is greater potential for making intelligent tactical and strategic decisions based on data, improving customer response time, tailoring to the needs of important customers, enhancing customer service, and creating profitable products.

Data warehouses can be used to perform corporate-wide decision support analysis. Effective ad-hoc DSS analysis may involve both summary data and detail data from the data warehouse. Queries can be grouped into three general categories: summary data creation, drill-down data using that summary data, and ad-hoc queries that entirely use detail data. In the first category, the summary data is typically created by running queries on the detail data. This will typically involve aggregation queries that must perform multi-way joins. In the second category, queries use the oft referenced summary data. Using this global summary data one can perform "drill down" queries that perform analysis at different levels of summarization that contain finer sets of data that focus on more constrained data [6].

With information from the drill down queries, often star queries can be performed on the detail data, making use of star queries. In the third category, ad-hoc queries are performed on the detail data of the entire data warehouse. These queries will usually return a small set of data from scans of a very large data warehouse. These important ad-hoc queries that scan hundreds of Gigabytes or Terabytes of detail data require a system with high compute and DASD (disk) performance.

Data warehouse data volumes are measured in multiple Gigabytes and are increasingly being measured in Terabytes (TB or Tbytes). Capacity, functionality, and performance are key issues when deploying data warehouses. To understand the correct manner to address these issues it is instructive to look at the landmarks in a typical deployment. Data warehouse design and deployment is an incremental process. Many corporations begin by prototyping data warehouses at around the 10 GB level. Initial deployment is usually at the 100 GB level. Upon deployment, the usefulness of the data warehouse is often quickly realized. This success may have its problems. The amount of the data often grows much faster than expected. Within a year of initial deployment, the data warehouse may even grow to Terabytes of data. Many have found it difficult to predict the explosive growth in database size, the user community, and query types. This rapid growth means that prototyping should be done on a platform, such as the Cray CS6400, that provides the performance and capacity for each phase of the data warehouse deployment.

3 DSS CHARACTERISTICS

An important aspect of DSS operations on the data warehouse is their ad hoc nature. According to the Metagroup, a market research firm based in Stamford CT, 75% of all DSS queries are ad hoc. This is due to the iterative process of refining and defining new queries based on the information gathered from previous queries. These queries are ad hoc and unpredictable. It is difficult to pre-plan for these types of queries since they execute once and may access millions or billions of rows [1]. With ad hoc queries there is no perfect data layout, especially when refreshing the database with inserts and updates imbalances the original data layout. Such performance problems are a defining characteristic of MPPs [9] [11]. Alternatively, high-performance SMP systems are very flexible and capable. Data can be placed on disks in a manner that provides consistent data access for a wide variety of query types.

A characteristic of DSS applications that take advantage of parallel processing is the ability to divide a single query into sub-queries. Executing in parallel keeps both processors and disks active reducing execution time. In Oracle, these sub-queries execute on multiple "Query Servers" in parallel and provide results to a Query coordinator that combines results as required by the query. Parallel systems, such as on the CS6400, provide a cost-effective approach to meeting typical DSS performance requirements on large databases.

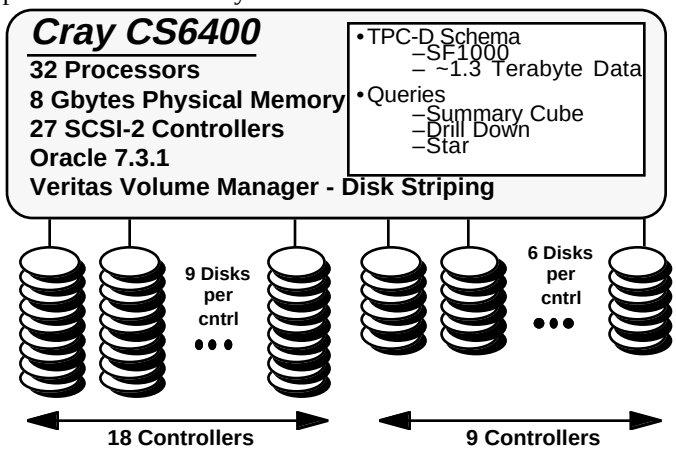
Many data warehouses will incrementally load new data from the operational databases with a lag of 24 hours [6] for data settling. It is important to plan for these incremental batch loads. For some enterprises, this operational data may come from legacy systems, therefore mainframe connectivity is often a requirement. SMPs can handle incremental loads without causing any performance problems. On an MPP system, inserts un-balance the data layout that directly affects performance. Re-partitioning data can overcome these problems, however, re-partitioning can be a time consuming step and it is best done when the database is off-line and unavailable for use. This can be a big performance penalty for a 7 day 24 hour operation.

4 TEST-TO-SCALE BENCHMARK

To demonstrate the Terabyte capacity that Oracle provides on powerful SMP systems, several queries representative of important DSS operations were executed with a CS6400 system. Oracle's Test-to-Scale Benchmark models a real-world business analysis scenario. This test emphasizes the features that are most important for data warehousing customers: data loading, index creation, creation of summary tables, and complex query processing.

4.1 Configuration and Setup

The CS6400's configuration for this benchmark consisted of thirty-two 85 MHz SuperSPARC processors, 8192 MB of physical memory, and two hundred forty-six (9 GB) disks. Each processor has a 2 MByte external cache.



Oracle version 7.3.1 was used together with VERITAS Volume Manager (VxVM) which provided disk management and RAID-0 (striping) for the data files. The data tablespace was evenly spread across 246 disks using a stripe size of 64K. The SORT_AREA_SIZE was 8 MB per query server process, the HASH_AREA_SIZE was 67 MB for each of the sixty-four query server processes for a total of 4.2 GBytes. We think even more benefit would be observed with a larger size and still be able to fit in the large physical memory of the CS6400. The DB_BLOCK_SIZE was 8K, and the number of DB_BLOCK_BUFFERS for the SGA was 10000 (approximately 80 MBytes) since these queries did not make heavy use of the SGA.

The database was composed of 942 database files each of 2048 Mbytes, this gave a total tablespace size of 1880.7 GB or 1.83 TB. For this test, each database file contained one extent of size 2047 MBytes, except of the initial extent which was 16

Kbytes. Aggresive loading was performed on many of these tables. In some tables, 2040 Mbytes of data was loaded into a database file (99.6% of the 2047MB extent). The data tablespaces had a PCTINCREASE of zero. The tuning process only involved adjusting Oracle init.ora parameters [12].

Each database file was stripped across many different disks. The Veritas volume management software VxVM was used to created striped volumes(RAID-0) for each of the database files. Data was not mirrored due to the reliability of the disk configuration and the amount of disk available during the benchmark. Mirroring would be recommended for any production environment. A small stripe size (64k) used with VERITAS VxVM makes it quite simple to avoid all the disk hot-spotting problems that are common with MPPs [4]. For easy-to-manage predictable performance, SMP's can efficiently distribute the data in a fine-grain fashion and still have every processor have equal access time to the data. Without this equal access time of data, bottlenecks can degrade performance by orders of magnitude and serialize the processing on MPPs. Each logical disk volume was composed of a portion of a disk on each controller (one on each controller).

4.2 Data Description

The demonstration involved creating several tables that are representative of data warehousing information. The tables were loaded with randomly generated data to simulate read data. Information about the data in the tables is shown in the table below.

The lineitem table accounts for most of the database size. Row size is highly variable in most data warehouses and therefore the oft used performance metric of Millions of rows per second is a poor measurement and should not be used. Typical data warehouses may have more tables, however this demonstration proves the capacity required to implement production data warehouses.

4.3 Test-to-Scale Query Description

The queries modeled in this demonstration were chosen to be representative of the queries important to a data warehouse involving both detail data and summary data. A single Oracle instance was used with a single user issuing the queries sequentially. The amount of table data was constant during the execution of the tests.

The Terabyte Test-to-Scale simulates a typical sales analysis scenario. Sales executives and brand managers often perform an in-depth examination of variations in sales, attempting to identify brands that performed below expectations during a given period or in a given region.

Identification of such performers is followed by a detailed analysis of the given product and its market. This type of analysis is often called "drill-down" analysis; an analyst begins with a high-level question ("Which product performed below quotas in 1995?"), and successively "drills down" with more detailed questions ("In which region(s) was this product weak?" followed by "In which month(s) was this product weak in the given region?"). This type of query can be accelerated through the used of summary tables.

Summary tables are relatively small tables that contain aggregate data. For instance, a summary table created in the Terabyte Test-to-Scale describes the sales per month for every product and packaging in every nation. This summary table was less than 2 GB, so that accessing this summary data is very efficient. A summary table would be used to evaluate a query such as "What was my best-performing product in the past year?" The summary cube creation query was a 6-way table join that performed aggregations on group-by data. The tables joined were 6 of the largest tables (lineitem, customer, orders, parts and nation {twice}). From the first summary table, 6 additional summary tables were created with higher levels of summarization. The summary table below aggregated data summarizrom 7.2 Billion rows of data.

The drill-down queries accessed these summary tables, starting with a highly summarized table, T_12 and successively

Table	*Tablespace Size (Gigabytes) 1GB=1024MB*	*User Data Size for Query (Gigabytes) 1GB=1024MB*	*Rows*	*Columns*	*bytes/row*	*Index Size*
lineitem	720.0 GB	658.7 GB	5,999,989,709	16	117.9	216 GB
orders	162.0 GB	158.1 GB	1,500,000,000	9	113.1	est. 50 GB
Partsupp	124.9 GB	na	800,000,000	5	167.6	na
parts	26.0 GB	23.0 GB	200,000,000	9	123.4	est. 7 GB
customer	26.0 GB	23.3 GB	150,000,000	8	167.4	na
supplier	2.0 GB	23.3 GB	10,000,000	7	146.4	na
nation	2.0 GB	na	25	4	128.0	na
region	2.0 GB	na	5	3	124.0	na
temp	812.0 GB	na	na	na	na	na
other	8.0 GB	na	na	na	na	na
Total Gbytes	*1880.8 GB*	*863.1 GB*	*8,659,989,739*			
Total Tbytes	*1.84 TB*	*0.84 TB*				

getting to tables with more detail using information from each to refine the next query. This continued until table T_123456 was accessed. The query was designed to answer the following questions in order with each query using the output data from the previous query:

1. What was the worst month in a particular year?
2. What nation had the least product quantity for that (month)?
3. What brand had lowest quantity for that (nation, month)?
4. What size container did they sell least for that (brand, nation, month)?
5. To which nation did they sell the least for that (container, brand, nation, month)?

This data was then feed into a star query to determine additional information such as the individual account balances for this poorly performing product. The star query used indexes on several of the tables to speed execution. A three column concatenated index was created on lineitem, and other indexes were created on parts and orders. Without building indexes this query is requires the use of full table scans on many of the large tables.

The creation of summary tables and indexes are both resource-intensive operations that are typically executed as overnight or weekend jobs in data warehousing environments. A data warehouse may receive new portions of detail data on a daily or weekly basis; after inserting the new detail records, the summary tables and indexes are rebuilt.

4.4 Building and Loading the Terabyte of Data

Using VERITAS Volume manager, 1000 volumes were created and associated with the various tablespaces. Each volume was striped (RAID-0) across 27 controllers for the first 6 disks per controller and then across 18 controllers for the next 4 disks per controller. The stripe size used was 64K, so the first disk on the first controller had the first 64 Kbytes, the second disk on the second controller had the next 64 Kbytes, and so on. The 27th write put 64 Kbytes on the first disk on the 27th controller, and the 28th write placed the next 64 Kbytes on the first disk of the first controller immediately following the first 64k bytes of the file. Each disk contained portions of 111 volumes for the first set of disks and 74 volumes for the second set of disks (since there where fewer controllers used in the second set of disks. The workable size used on each disk was about 8440 Mbytes.

There is no need to partition different tables on different disks, as required for performance on MPPs. In addition with the homogenous data layout there is no time-consuming re-partitioning software required.

Oracle7 allows tablespaces to be created in parallel after an initial data file has initialized. This was accomplished by having the korn-shell scripting language fork multiple instances to add additional tablespaces "alter tablespace transaction add datafile ..." Best performance was observed when the number of *alter tablespace's* backgrounded was equal to the number of processors. The entire tablespace creation occurred at over 225 GBytes/hour.

Data generation and Oracle7 SQL loader (sqlldr) are both compute-bound operations. Both phases were performed simultaneously on the CS6400 due to the high performance of the SPARC processors. Cray was the only vendor with the who chose this route. The output from dbgen was directed to named pipes that were read by sqlldr processes. Data was loaded using the direct path option of the Oracle7 SQL loader (sqlldr). In addition, data was loaded in parallel by using a shell script having each sqlldr process load data in a separate database file in the tablespace.

4.5 Test-to-Scale Results

All Test-To-Scale performance results require a non-disclosure agreement as per Oracle Test-To-Scale rules. The best

Name	T_	T_1	T_12	T_123	T_1234	T_12345	T_123456
Rows	1	2	24	600	15,000	600,000	15,000,000
Columns	1	2	3	4	5	6	7
Entries	quantity	year quantity	month year quantity	cust_nation month year quantity	brand cust_nation month year quantity	container brand cust_nation month year quantity	sup_nation container brand cust_nation month year quantity
Description	Total Sales	Sales per Year	Sales in each month	customers who bought in each month	brands that each customers who bought in each month	containers sizes for each brand for each customers who bought in each month	supplier of the container sizes for each brand for each customers who bought in each month

Test-to-Scale load performance was observed when the number of dbgen processes and the number of sqlldr processes was equal to the number of processors. Since the data generation routines that were used are very compute intensive, it is also instructive to just look at the load rates for other tests. On the 1.9 Terabyte test discussed below a peak load rate using 32 CPUs was 27.7 GB/hour and on a 1.6 Terabyte test [13] using 48 CPUs a peak rate of 38.1 GB/hour was observed.

All Test-to-Scale results were obtained on a 32 processor CS6400 system with 8 Gigabytes of physical memory. The software versions used were as follows: Oracle 7.3.1 pre-release version, production version Solaris 2.4. The CS6400 is the fastest and most powerful system to be certified on Oracle's Test-to-Scale benchmark.

In order to create the summary cube, data from 6 tables must be joined together and aggregated. Faster hash-join performance would be obtained by using more processors and more memory. Hash-join performance improves as more memory is used, since less transfers back and forth between temporary tablespace. The CS6400 can utilize more physical memory than any other SMP server and can set HASH_AREA_SIZE to a larger size than any other system. This gives the CS6400 particular advantage on a key DSS operation in the data warehouse.

The performance on the Terabyte SF1000 database was also compared to a 10 Gigabyte (SF10) database to provide information on Scaleup as database size is increased. The term scaleup is used to describe the difference in performance as the size increases. For instance, perfect scaleup on a problem 100 times bigger would be 100 times longer. Results showed that the SF1000 database, only took 98.9 times longer even though it was 100 times bigger. This case showed slightly better than perfect scaleup due to slight differences in database setups. Perfect scaleup is important since it means that performance is directly related to problem size and what is more important is that it will not worsen as problem size is increased.

Performance on the drill-down queries was very quick due to the relatively minuscule size of these tables. The set of 5 drill-down queries took less than 1 minute to determine the worst selling product under a variety of constrains. This set of queries found worst month in the year, the worst supplier nation in that month, the worst performing brand for that nation, the worst size of container for that brand, and the customer that bought the least of that product. It is easy to see the strategic value of queries of this sort. The importance of summary information is that it can quickly speed queries that ask common questions.

The star query that utilized the summary data from the drill-down queries to determine all the customers for the poorly selling product and various other account information. This query used the detail information stored in the large database to accurately determine other important information. The star query only took 97.1 times longer even though it was 100 times bigger than a SF10 (10 Gigabyte) database. Again this kind of scaleup indicates that performance is directly related to problem size.

4.6 Contrast with Sun's and Hewlett Packard's Test-to-Scale Benchmark

A month after Cray ran the 1.6 Terabyte in August 1995, HP announced results on a preliminary 944 GByte 3-way join. This query was the precursor to Oracle's Test-to-Scale benchmark. HP mirrored the entire database including temp spaces and had 4 Terabytes of disk connected to their system. The following table compares these databases.

In January of 1996, three SMP vendors (Cray, Sun, and HP) announced certification on the Test-to-scale benchmark. Each of these vendors ran a 1 Terabyte query that consisted of a 6-way hash-join that resulted in a summary cube generation. Sun and HP claim these are 4 Terabyte databases when counting all temporary spaces and mirroring. We prefer to describe database size in terms of the only the database structures involved (data, indexes, and temporary) and to exclude the additional disk storage required for mirroring. In addition, we refer to query size as the amount of data store in tables that are required by the query. Others have a much more liberal interpretation.

In February 1996, NCR announced an 11 Terabyte database using their 5100M MPP. This test was more of a OLTP-style test simulating 3,000 users only performing small queries on the database. It did not test the ability to generate the required large summary tables, perform large joins, or create large indexes. It was designed to avoid testing areas where MPPs have architectural problems. No performance results have been published on this test.

5 1.9 TERABYTE BENCHMARK

This benchmark was modeled after the 1.6 Terabyte benchmark that Cray ran using Oracle in August of 1995 [13] The 1.9 Terabyte database contained a 1.9 Terabyte table and was meant as a demonstration of Oracle's data warehousing capacities. It is the largest Oracle table constructed to date.

5.1 1.9 Terabyte Query Description

The "full table scan with aggregates" query scans the table and performs an aggregation on each of the seventeen columns in the table. Aggregation functions used were min, max, avg, sum, and count. The scanned table contained 9.3 billion rows each of approximately 100 bytes for a total of 1.86 Terabytes of user data. This does not count mirroring, temporary tables, or indexes.

5.2 1.9 Terabyte Results

A full table scan (SCAN)of the 1.9 Terabyte table with 17 aggregates using 32 processors took 12 hours twenty-five minutes to complete. This is one of the most data intensive operations in DSS that is performed completely on the 1.9 Terabyte table. The scalability on this query on the 1.9 Terabyte table was slightly better than that obtained on a small database [2] (1 Gigabyte table) which demonstrated near-linear scalability of the Cray-Oracle solution. The small database test showed efficient scaling with a 39-fold performance improvement on 40 processors over one processor. A one processor run was not performed on the 1.6 or the 1.9 Terabyte table due to the time required for a one processor run. In addition, a full table scan

Table	*Database Size*	*"Claimed Size"*	*Rows in Largest Table*	*Largest Table Size*	*Query*	*Query time*	*Query Size*
Cray: 1.6TB no mirroring 8/95	1651 GB	1.6 TB	8,106,000,000	1649 GB	2 way Nested Loop Join	9:30:47	1651 GB
HP: preliminary TTS, 9/95	1300 GB	4.0 TB counted mirroring & scratch	5,999,989,709	746 GB	3-way Hash Join	?	863 GB
Cray: 1.9TB 11/95	1860 GB	1.9 TB	9,273,650,000	1859 GB *Largest to Date*	Full Table Scan with 17 Aggregates	12:25:20	1860 GB *Largest to Date*
HP: TTS SF1000 12/95	1300 GB	4.0 TB counted mirroring & scratch	5,999,989,709	746 GB	6-way Hash Join Summary Cube	?	863 GB
Sun: TTS SF1000 12/95	1300 GB	4.0 TB counted mirroring & scratch	5,999,989,709	746 GB	6-way Hash Join Summary Cube	?	863 GB
Sun: TTS SF1600 1/95	1872 GB	5.5 TB counted mirroring & scratch	9,599,983,534	1246 GB	6-way Hash Join Summary Cube	?	1664 GB
Cray: TTS SF1000 1/95	1300 GB	1.3 TB	5,999,989,709	746 GB	6-way Hash Join Summary Cube	*Fastest to date*	863 GB

(1TB=1024GB, 1 GB=1024MB, TTS=Test-to-Scale, SF1000=1TB, SF1600=1.6 TB)

with only a count aggregate took 9 hours thirty-six hours to complete. These queries would require several weeks to process on other high-end parallel servers and proprietary mainframes.

6 CRAY SUPERSERVER 6400 SYSTEM

The CS6400 is an enterprise-class application or data server for a wide range of tasks such as on-line transaction processing (OLTP), decision support systems (DSS), on-line analytical processing (OLAP), and data warehousing. The result of a technology agreement between Cray Research and Sun Microsystems, the CRAY CS6400 is a binary-compatible upward extension of Sun Microsystems' product line. Its full compatibility with Sun Microsystems' Solaris operating system guarantees the availability of the largest set of third-party solutions in open systems. Large configurations of this SMP system can simultaneously support sixty-four processors, 16 Gigabytes of physical memory, and 10+ Terabytes of online disk storage. The CS6400 also has the capacity to combine DSS and online transaction processing (OLTP) job mixes on the same platform. The CS6400 also provides processor partitioning to segregate these workloads for flexibility in system management. In addition to DSS scalability, the CS6400 has also shown excellent OLTP Scalability. It leads in the industry in TPC Benchmark™ B Results with a performance of 2025.20 tpsB and leads in price/performance with $1,110.14 per tpsB (result date: 6/4/94).

RAS features are a critical part of the design of the CS6400. There is nearly complete redundancy of system components in the CS6400. This includes multiple redundant system buses, N+1 power supplies, dual pathing, RAID devices, disk mirroring, etc. The CS6400 also offers fail-over, hot swap of system boards, dynamic reconfiguration (and expansion), and automatic reboot. A separate service processor including monitoring software (with call home on unplanned reboots) and remote diagnostics.

The speedup factors obtained are the result of joint engineering efforts by Oracle, Cray, and Sun in exploiting the performance features of Solaris 2, such as the multi-threaded architecture of the Solaris kernel, asynchronous I/O, and efficient OS striping. Likewise, the hardware strengths of the CRAY SUPERSERVER 6400 that facilitate good scalability include the quad XDBus bus architecture, fast SCSI controllers, and larger CPU caches to hold frequently referenced data and instructions. Oracle will exploit faster CPUs with larger caches to deliver even bigger performance boosts for future generations.

The SMP architecture allows DSS queries to be optimized for parallel operations, while avoiding the MPP performance and administration problems. MPP performance can be very dependent on data layout. On MPPs, the user has the choice between executing high-performing "good" queries and slow-performing "bad" queries. This has the drawback of potentially "training" users what queries not to submit. In these respects, MPPs are more difficult to tune and to administer. Even on an MPP that uses a shared disk strategy, there can be other problems on an MPP due to coordinating the various IO requests from within the MPP.

7 CONCLUSION

The CS6400 is well suited for data warehouses throughout the entire deployment process from the Gigabyte prototype to the multi-Terabyte production system. The CS6400 has demonstrated the capacity to handle Terabyte data warehouses as shown by many different Terabyte demonstrations. In addition, work with Oracle's Test-to-Scale data warehouse benchmark has shown the CS6400 to be the most powerful system to complete a wide range of queries important to the data warehouse. These queries include generation of a summary cube from a Terabyte of data, drill-down queries, star queries, and large index generation. The 1.9 Terabyte query discussed in this paper also maintains the lead as the largest query ever run using Oracle.

Performance and scalability are particularly important for large data warehouse applications. The CS6400's SMP design allows commercial DBMSs to effectively use its large configuration of processors. Large configurations of the CS6400 provide excellent scalability on DSS operations using the Oracle7 shared-everything implementation. The large memory capacity of the CS6400 also provided a particular performance advantage for the queries that used hash joins.

Oracle7 was extremely robust throughout all of the work on this demonstration. In addition, VERITAS Volume Management software efficiently and reliably performed the disk stripping. We are confident that all major hurdles have been cleared for much larger databases using Oracle7 on the Cray CS6400.

The efficient implementation of the Oracle7 on the C6400 provides near-linear scalability while maintaining all the advantages of SMP systems. Past limits to SMP scalability are avoided by providing sufficient performance at every level in a balanced system.

8 REFERENCES

[1] S. Brobst, "An Introduction to Parallel Database Technology", VLDB Summit, Miller Freeman, Inc, 1995.

[2] B. Carlile, "Linear Scalability on Decision Support Systems: Cray CS6400", *DoD Database Colloquium '95 Proceedings*, August 28-30, pp 603-611.

[3] A. Cockcroft, Sun Performance and Tuning, SunSoft Press, A Prentice Hall Title, 1995.

[4] H. Edelstein, "The Power of Parallel Database", *DBMS: Parallel Database Special*, 1995, pp D-G.

[5] J.L. Hennessy and D. A. Patterson, *Computer Architecture A Quantitative Approach* (Morgan Kaufmann Publishers, San Mateo CA, 1990).

[6] W. Inmon, *Building the Data Warehouse* (A Wiley - QED Publication, New York 1993).

[7] D. McCrocklin, "Scaling Solaris for Enterprise Computing", Cray User's Group, 1995.

[8] *Oracle and Cray Superserver 6400 Linear Scalability*, Oracle Corporation, May 1995.

[9] "Open Computing & Server Strategies", Fourth Quarter Trend Teleconference Transcript, META Group, Dec 13, 1994.

[10] J. Scroggin, "Oracle7 64-Bit VLM Capability Makes Digital Unix Transactions Blazingly Fast", Oracle Magazine, Vol IX, No 4, July/August 1995, pp 89-91.

[11] C. Stedman, "What you don't know... ... will hurt you", Computer World MPP & SMP special Report, March 27, 1995, supplement pp 4-9.

[12] *Oracle for Sun Performance Tuning Tips*, Oracle Corporation, Part # A22554, May 1995.

[13] B. Carlile, "Multi-Terabyte SMP Data Warehouse: The Cray CS6400/Oracle7 Solution", *to be published* , November 1995, pp 1-8.

[note1] 80 chars/per line, 66 lines/page, 2 pages/sheet, 250 sheets/inch (copier paper).

9 AUTHOR INFORMATION

9.1 Speaker's Biographical Sketch

Brad Carlile is a Performance Manager at Cray Research, Business Systems Division. He is responsible for analyzing and characterizing real-world workloads. Background includes work on eight distinct shared and distributed memory parallel architectures on a wide variety of commercial and technical applications. He is author of over 17 technical papers and articles and was the first person to achieve more than a 1.0 Terabyte Oracle database with a 1.6 Terabyte Database. His current focus is Data Warehousing and DSS performance issues.

9.2 Contact Information

Brad Carlile

Cray Research, Business Systems Division

8300 Creekside Ave,

Beaverton, OR 97008

bradc@oregon.cray.com

System Components	Configurations	Specifications
Number of Processors	4-64 SPARC	85 MHz SuperSPARC
Memory Size	16 Gbytes	SMP, Shared Memory
System Bandwidth	1.7 GB, 4 XDBuses	55 MHz
I/O Channels	16 SBuses	800 MB/s
Bus Controllers	64	Full Coherency
Online Disk Capacity	10+ Tbytes	Using 9 GB disks+Raid
Operating System	Solaris 2.4	SVR4, Solaris Enterprise Server