TPC Benchmark™ D Full Disclosure Report

for

IBM Netfinity 7000

using

DB2 Universal Database 5.2.0

Submitted for Review

June 1, 1998

First Edition - June 1998

Trademarks

IBM is a registered trademark and Netfinity is a trademark of International Business Machines Corporation.

The following terms used in this publication are trademarks of other companies as follows: TPC Benchmark, TPC-D, QppD, QthD and QphD are trademarks of Transaction Processing Performance Council; Intel and Pentium Pro are registered trademarks of Intel Corporation; Microsoft and Windows are trademarks or registered trademarks of Microsoft Corporation. Other company, product, or service names, which may be denoted by two asterisks (**), may be trademarks or service marks of others.

Notes

[1] MHz only measures microprocessor internal clock speed, not application performance. Many factors affect application performance.
[2] When referring to hard disk capacity, one GB equals one billion bytes. Total user-accessible capacity depends on the operating environment.

Abstract

IBM Corporation conducted the TPC Benchmark™ D on the IBM Netfinity* 7000. This report documents the full disclosure information required by the TPC Benchmark™ D Standard Specification, Revision 1.3.1, including the methodology used to achieve the reported results. All testing fully complied with this revision level.

The software used on the IBM Netfinity 7000 system includes Microsoft** Windows** NT Server 4.0 operating system and IBM DB2 Universal Database 5.2.0.

The benchmark results are summarized in the following table.

Hardware	Software	Total System Cost	QppD@100GB	QthD@100GB	$ / QphD@100GB
IBM Netfinity 7000	DB2 UDB 5.2.0 Microsoft Windows NT Server 4.0	**$169,942**	**987.2**	**362.4**	**$284**

The results of the benchmark and test metho dology used were audited by Francois Raab of Information Paradigm. The auditor's letter is contained in Section 9 of this report.

Copies of this full disclosure report can be obtained from either the Transaction Processing Performance Council or IBM Corporation at the following address:

Transaction Processing Performance Council (TPC)
c/o Shanley Public Relations
777 North First Street, Suite 600
San Jose, CA 95112 USA
Telephone: (408) 295-8894, Fax: (408) 295-9768

or

IBM Corporation
Mail Drop 23U/205/QQ160
3039 Cornwallis Road
Research Triangle Park, NC 27709

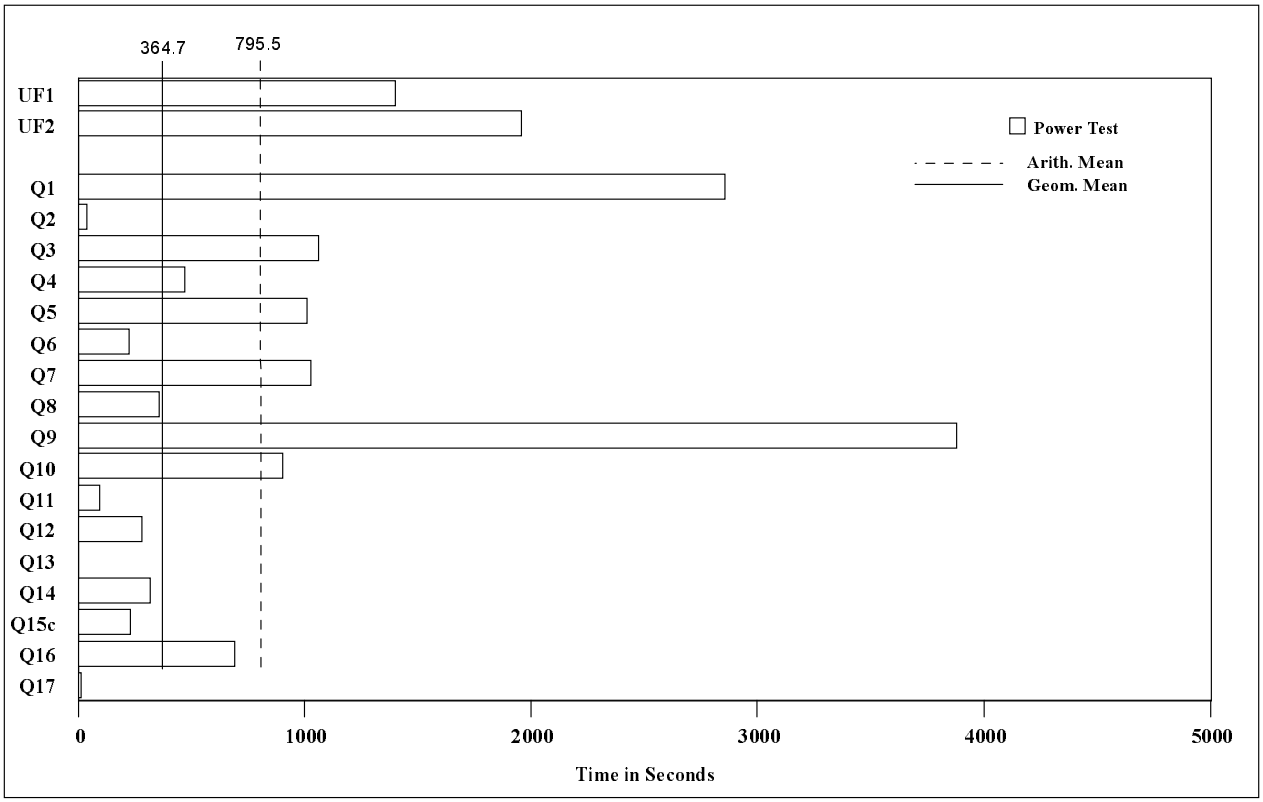
IBM Netfinity 7000 with DB2 UDB 5.2.0

TPC-D Rev 1.3.1

Report Date: 1 June 1998

Total System Cost	TPC-D Power	TPC-D Throughput	Price/Performance
$169,942	**987.2 QppD@100GB**	**362.4 QthD@100GB**	**$284 per QphD@100GB**

Database Size	Database Manager	Operating System	Other Software	Availability Date
100GB	**DB2 UDB 5.2.0**	**Windows NT Server 4.0**	**Visual C++ 4.2**	**October 31, 1998**



Database Load Time: 77h 54m 38s

Total Data Storage/Database Size: 3.75

RAID: N

Components	Qty	
Processors	**4**	**200MHz Pentium Pro / 1MB L2 Cache**
Memory	**2**	**1024MB**
Disk Controllers	**3**	**IBM ServeRAID II Ultra SCSI Adapter**
Disk Drives	**83**	**4.51GB**
Total Storage		**374.33GB**

IBM Netfinity 7000 with DB2 UDB 5.2.0

TPC-D Revision 1.3.1

Report Date: 1 June 1998

Description	Part Number	Source	Unit Price	Qty	Ext. Price	5-Yr. Maint.**
Server Hardware						
IBM Netfinity 7000 / 200MHz PentiumPro/1MB L2 Cache	8651-RH0	1	12,674	1	12,674	5,985
200MHz / 1MB Processor Upgrade	94G7147	1	4,843	3	14,529	0
Netfinity 1024MB Memory Expansion Kit	94G7386	1	5,725	2	11,450	0
IBM ServeRAID II Ultra SCSI Adapter	76H3584	1	1,785	3	5,355	0
Third Channel Cable	76H5400	1	27	2	54	0
IBM 4.51GB 10K Wide Ultra SCSI Drive	01K7960	1	758	3	2,274	0
IBM 4.3M External .8mm SCSI Cable	01K8029	1	56	8	448	0
IBM Auto/4 PCI Token-Ring Adapter	41H8900	1	210	1	210	0
IBM G42 14" (13.2" Viewable) Color Monitor	654000N	1	209	1	209	408
20/40GB SCSI Tape Drive	76H0485	1	3,157	1	3,157	0
Netfinity Rack	9306900	1	1,733	1	1,733	0
Side Panel Kit	94G6669	1	201	1	201	0
				Subtotal	52,294	6,393
Server Software						
DB2 UDB 5.2.0 Workgroup Edition		2	999	1	999	0
DB2 UDB 5.2.0 10-User License		2	1,849	1	1,849	0
Microsoft Windows NT Server 4.0 SP4		3	809	1	809	0
Microsoft Visual C++ and Subscription Service		3	499	1	499	0
Microsoft Priority Plus Database Server Support		3	2,095	5		10,475
				Subtotal	4,156	10,475
Storage Devices						
IBM Netfinity EXP10 Rack Storage Exp.	35201RU	1	2,698	8	21,584	14,400
Netfinity EXP10 4.51GB 10K Wide Ultra SCSI	01K8009	1	758	80	60,640	0
				Subtotal	82,224	14,400
				Total	138,674	31,268

Pricing: 1 - Entex Information Services; 2 - IBM; 3 - Microsoft Corp.

**** The standard 3-year warranty and the extended warranty on all hardware are for 7x24, on site same day coverage.**

Audited by Francois Raab of Information Paradigm.

Five-Year Cost of Ownership:	**$169,942**
QppD @ 100GB:	**987.2**
QthD @ 100GB:	**362.4**
QphD @ 100GB:	**598.1**
$ / QphD @ 100GB:	**$284**

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one-time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specification. If you find that stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

IBM	**IBM Netfinity 7000 with DB2 UDB 5.2.0**	TPC-D Rev 1.3.1 Report Date: 1 June 1998

Measurement Results:

Database Scaling (SF/Size)	100
Total Data Storage/Database Size	3.75
Database Load Time	77:54:38
Query Streams for Throughput Test	0
TPC-D Power Metric (QppD@100GB)	987.2
TPC-D Throughput Metric (QthD@100GB)	362.4
Composite Query-per-Hour Rating (QphD@100GB)	598.1
Total System Price over 5 Years	$169,942
TPC-D Price/Performance Metric ($/QphD@100GB)	$284

Measurement Intervals:

Measurement Intervals in Throughput Test (Ts) = 16,886 seconds

Duration of Stream Execution

Stream ID	Seed Used	Start Date	Start Time	End Date	End Time
UF1		05/27/98	14:18:03	05/27/98	14:41:28
Stream 0	1833	05/27/98	14:18:03	05/27/98	18:59:29
UF2		05/27/98	18:26:50	05/27/98	18:59:29

Timing Intervals (in seconds):

Stream ID	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**
Stream 00	2859.9	44.0	1066.4	473.4	1015.1	226.3	1030.7	363.8	3881.2

Stream ID	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15c**	**Q16**	**Q17**	**UF1**	**UF2**
Stream 00	908.0	99.0	284.3	2.5	319.4	236.0	697.3	15.5	1405.8	1958.4

Table of Contents

Preface

TPC Benchmark™ D Standard Specification was developed by the Transaction Processing Performance Council (TPC). It was released on May 5, 1995, and most recently revised (Revision 1.3.1) on December 4, 1997. This is the full disclosure report for benchmark testing of the IBM Netfinity 7000 according to the TPC Benchmark D Standard Specification.

TPC Benchmark D is a Decision Support benchmark. It is a suite of business-oriented queries and concurrent updates. The queries and the data populating the database have been chosen to have broad industrywide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

- Examine large volumes of data
- Execute queries with a high degree of complexity
- Give answers to critical business questions.

TPC-D evaluates the performance of various Decision Support Systems by the execution of set of queries against a standard database under controlled conditions. The TPC-D queries:

- Give answers to real-world business questions
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- Are executed against a database complying to specific population and scaling requirements
- Are implemented with constraints derived from staying closely synchronized with an on-line production environment.

The TPC-D operations are modeled after the following business environment:

- The database is continuously available 24 hours a day, 7 days a week, for queries or updates against all tables for multiple users, except possibly during infrequent (e.g., once a month) maintenance sessions.
- The TPC-D database tracks, possibly with some delay, the state of the OLTP database through ongoing updates, which batch together a number of modifications impacting some part of the Decision Support database.
- Due to the worldwide nature of the business data stored in the TPC-D database, the queries and the updates may be executed against the database at any time, especially in relation to each other. In addition, this mix of queries and updates is subject to specific ACIDity requirements, since queries and updates may execute concurrently.
- To achieve the optimal compromise between performance and operational requirements, the database administrator can set, once and for all, the locking levels and the concurrent scheduling rules for queries and updates.

The minimum database required to run the benchmark holds business data from 10,000 suppliers. It contains almost 10 million rows representing a raw storage capacity of about 1 gigabyte. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g., 100 gigabytes), as defined in Clause 4.1.3).

The performance metrics reported by TPC-D measure multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the TPC-D query processing power at the selected size (QppD@Size), and the TPC-D throughput at the selected size (QthD@Size) when queries are submitted by one or more concurrent users. The TPC-D Price/Performance metric is expressed as $/QphD@Size and is based on a composite query-per-hour rating derived from QppD and QthD. To be compliant with the TPC-D standard, all references to TPC-D results for a given configuration must include all required reporting components (see Clause 5.4.7).

The TPC-D database was implemented using a commercially available database management system (DBMS), and the queries executed via an interface using dynamic SQL. The specification provides for variants of SQL, as implementors are not required to have implemented a specific SQL standard in full.

Benchmarks results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-D should not be used as a substitute for specific customer application benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

General Items

Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

This benchmark was sponsored by International Business Machines Corporation.

Parameter Settings

Settings must be provided for all customer-tunable parameters and options that have been changed from the defaults found in actual products, including but not limited to:

- *Database tuning options*
- *Optimizer/Query execution options*
- *Query Processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameters*
- *Configuration parameters and options for any other software component incorporated into the pricing structure*
- *Compiler optimization options.*

Appendix A, "Tunable Parameters," contains a list of all DB2 parameters, operating system parameters and compiler options. Session initialization parameters can be set during or immediately after establishing the connection to the database within the tpcdbatch program documented in Appendix D, "Implementation-Specific Layer and Driver Source Code." This result uses the default session initialization parameters established during preprocessing/binding of the tpcdbatch program. The procedure for preprocessing, binding, compiling and linking the tpcdbatch program is documented in Appendix A.5, "Compiler Options for Driver."

Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences. This includes, but is not limited to:

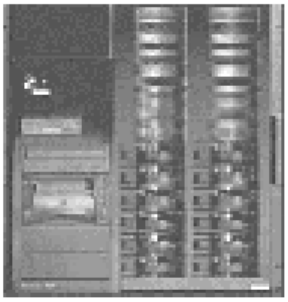
- *Number and type of processors*
- *Size of allocated memory and any specific mapping/partitioning of memory unique to the test and type of disk units (and controllers, if applicable)*
- *Number and type of disk units (and controllers, if applicable)*
- *Number of channels or bus connections to disk units, including their protocol type*
- *Number of LAN (e.g., Ethernet) connections, including routers, workstations, terminals, etc., that were physically used in the test or are incorporated into the pricing structure*
- *Type and run-time execution location of software components (e.g., DBMS, query processing tools/languages, middleware components, software drivers, etc.)*.

The configuration diagram for the tested and priced system is provided on the following page.

IBM Netfinity 7000 Measured and Priced Configuration

Server

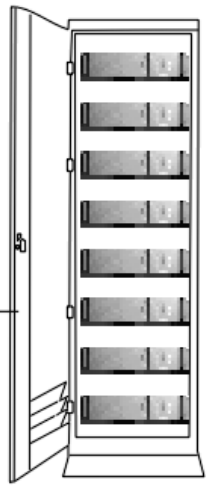
IBM Netfinity 7000
4-way 200MHz Pentium Pro
with 1MB L2 Cache
2 GB Memory



3 x IBM ServeRAID II
Ultra SCSI Adapter
3 x 4.51GB 10K rpm Drives

Storage Subsystem

8 x IBM Netfinity EXP10
Rack Storage Expansion
Enclosures



80 x 4.51GB 10K rpm Drives

Clause 1: Logical Database Design Related Items

Appendix B, "Database Build Scripts," contains the programs and input files used to load the test and qualification databases. The test and qualification databases are built in exactly the way in all respects except the scale factor; they use the same table definitions, indices and partitioning methods. Thus, the buildtpcd script documented in Appendix B.1 was used for both the qualification and test databases except that different input files were used.

There are two phases for the loading of the database: the generation of the flat data files and the building of the database from them. The buildtpcd script executes DDL and other command scripts to create the database, load the data into the tables, execute DDL (e.g., to create indices), gather statistics, and set the configuration. These DDL and other command scripts are documented in Appendix B.

Table Definitions

Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases. (8.1.2.1)

Appendix B, "Database Build Scripts," contains the table definitions and the program used to load the database.

Database Organization

The physical organization of tables and indexes within the test and qualification databases must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.

Appendix B, "Database Build Scripts," contains the DDL for the index definitions.

Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases must be disclosed (see Clause 1.5.4).

Horizontal partitioning was not used.

Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.6).

No replication was used except in the sense that the indexes contained some of the same data that is also contained in the database tables.

Clause 2: Queries and Update Functions Related Items

Query Language

The query language used to implement the queries must be identified (e.g., RALF/SQL-Plus).

SQL was the query language used.

Verification for the Random Number Generator

The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.

The supplied DBGEN version 1.3.1 and QGEN version 1.3.1 were used to generate all database populations.

Substitution Parameters

Method of Generation

The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number and patch level of QGEN must be disclosed.

The supplied QGEN version 1.3.1 was used to generate the substitution parameters.

Query Text

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Appendix C.1, "Qualification Queries and Output," contains the output for each of the queries. The functional query definitions and variants used in this disclosure use the following minor query modifications:

- Table names and view names are fully qualified . For example, the nation table is referred to as "TPCD.NATION." The "order" table is named "orders."
- The standard IBM SQL date syntax is used for date arithmetic. For example, DATE('1996-01-01') + 3 MONTHS
- The semicolon (;) is used as a command delimiter.

Disclosure

All query substitution parameters used for all performance tests must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix C.3, "Query Substitution Parameters," contains the query substitution parameters used in the performance tests.

Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to one of the isolation levels defined in Clause 3.4, additional descriptive detail must be provided.

The isolation level used was “repeatable read.”

Update Functions

The details of how the update functions were i mplemented must be disclosed (including source code of any non-commercial program used).

The update function is part of the im plementation-specific layer/driver code incl uded in Appendix D, “Implementation-Specific Layer and Driver Source Code.”

Database Maintenance Option

The details of the database maintenance option selected (i.e., reset or evolve) must be disclosed (including source code of any non-commercial program used).

The implementation uses the evolve option.

Clause 3: Database System Properties Related Items

The results of the ACID tests must be disclosed, along with a description of how the ACID requirements were met. This includes disclosing the code written to implement the ACID Transaction and query.

Atomicity Requirements

The system under test must guarantee that database transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially completed operations leave any effects on the data.

All ACID tests were conducted according to specification. The Atomicity, Isolation, Consistency and Durability tests were performed on the IBM Netfinity 7000. Appendix E, "ACID Transaction Source Code," contains the source code for the ACID transaction and query.

Atomicity of Completed Transactions

Perform the ACID transactions for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDER, LINEITEM and HISTORY tables.

The following steps were performed to verify the Atomicity of completed transactions.

1. The total price from the ORDER table and the extended price from the LINEITEM table were retrieved a for a random Orderkey. The number of records in the HISTORY table were also retrieved.
2. The ACID transaction was executed for the Orderkey used in step 1.
3. The total price and the extended price were retrieved for the same Orderkey used in steps 1 and 2. It was verified that: EXTENDEDPRICE = EXTENDEDPRICE + ((DELTA) * (EXTENDEDPRICE/QUANTITY)), TOTALPRICE = TOTALPRICE + (COST * (1-DISCOUNT) * (1 + TAX)), and that the number of records in the history table had increased by 1.

Atomicity of Aborted Transactions

Perform the ACID transactions for a randomly selected set of input data, submitting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDER, LINEITEM and HISTORY tables.

The following steps were performed to verify the Atomicity of the aborted ACID transaction:

1. The ACID application is passed a parameter to execute a rollback of the transaction instead of performing the commit.
2. The total price from the ORDER table and the EXTENDEDPRICE from the LINEITEM table were retrieved for a random Orderkey. The number of records in the HISTORY table was also retrieved.
3. The ACID transaction was executed for the Orderkey used in step 2. The transaction was rolled back.
4. The total price and the extended price were retrieved for the same Orderkey used in steps 2 and 3. It was verified that the extended price and the total price were the same as in step 2.

Consistency Requirements

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another.

Consistency Condition

A consist state for the TPC-D database is defined to exist when:

O_TOTALPRICE = SUM(L_EXTENDEDPRICE1-L_DISCOUNT)(1+L_TAX)*

for each ORDER and LINEITEM defined by (O_ORDERKEY=L_ORDERKEY)

The following query was executed before and after a measurement to show that the database was always in a consistent state both initially and after a measurement.

```
SELECT DECIMAL(SUM(DECIMAL(INTEGER(DECIMAL
(INTEGER(100*DECIMAL(L_EXTENDEDPRICE,20,2)),20,2)*
(1-L_DISCOUNT))*(1+L_TAX)),20,2)/100.0,20,2)
FROM TPCD.LINEITEM WHERE L_ORDERKEY = O_ORDERKEY
```

Consistency Tests

Verify that the ORDER and LINEITEM are initially consistent as defined in Clause 3.3.2.1, based on a random sample of at least 10 distinct values of O_ORDERKEY.

The query defined in the "Consistency Condition" section above was run after initial database build and prior to executing the ACID transaction. The query showed that the database was in a consistent state.

After executing 100 ACID transactions, the query, defined in the "Consistency Condition" section above, was run again. The query showed that the database was still in a consistent state.

Isolation Requirements

Isolation Test 1

This test demonstrates isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed.

The following steps were performed to satisfy the test of isolation for a read-only and a read-write committed transaction:

1. First session: Start an ACID transaction with a randomly selected O_KEY,L_KEY and DELTA. The transaction is delayed for 60 seconds just prior to the Commit.
2. Second session: Start an ACID query for the same O_KEY as in the ACID transaction.
3. Second session: The ACID query attempts to read the file but is locked out by the ACID transaction waiting to complete.
4. First session: The ACID transaction is released and the Commit is executed releasing the record. With the LINEITEM record now released, the ACID query can now complete.
5. Second session: Verify that the ACID query delays for approximately 60 seconds and that the results displayed for the ACID query match the input for the ACID transaction.

Isolation Test 2

This test demonstrates isolation for read-write conflict of read-write transaction and read-only transaction when the read-write transaction is rolled back.

The following steps were performed to satisfy the test of isolation for read-only and a rolled back read-write transaction:

1. First session: Perform the ACID transaction for a random O_KEY, L_KEY and DELTA. The transaction is delayed for 60 seconds just prior to the Rollback.
2. Second session: Start an ACID query for the same O_KEY as in the ACID transaction. The ACID query attempts to read the LINEITEM table but is locked by the ACID transaction.

3. First session: The ACID transaction is released and the Rollback is executed, releasing the read.
4. Second session: With the LINEITEM record now released, the ACID query completes.

Isolation Test 3

This test demonstrates isolation for write-write conflict of two update transactions when the first transaction is committed.

The following steps were performed to verify isolation of two update transactions:

1. First session: Start an ACID transaction T1 for a randomly selected O_KEY, L_KEY and DELTA. The transaction is delayed for 60 seconds just prior to the Commit.
2. Second session: Start a second ACID transaction T2 for the same O_KEY, L_KEY and for a randomly selected DELTA2. This transaction is forced to wait while the First Session holds a lock on the LINEITEM record requested by the Second Session.
3. First session: The ACID transaction T1 is released and the Commit is executed, releasing the record. With the LINEITEM record now released, the ACID transaction T2 can now complete.
4. Verify that:
   T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE +
   (DELTA*(T1.L_EXTENDEDPRICE)/T1.L_QUANTITY)

Isolation Test 4

This test demonstrates isolation for write-write conflicts of two ACID transactions when the first transaction is rolled back.

The following steps were performed to verify the isolation of two ACID transactions after the first one is rolled back:

1. First session: Start an ACID transaction T1 for a randomly selected O_KEY, L_KEY and DELTA. The transaction is delayed for 60 seconds just prior to the Rollback.
2. Second session: Start a second ACID transaction T2 for the same O_KEY, L_KEY used by the First Session. This transaction is forced to wait while the First Session holds a lock on the LINEITEM record requested by the Second Session.
3. First session: Roll back the ACID transaction T1 . With the LINEITEM record now released, the ACID transaction T2 completes.
4. Verify that:
5. T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE

Isolation Test 5

This test demonstrates the ability of read and write transactions affecting different database tables to mak e progress concurrently.

The following steps were performed to successfully conduct this test:

1. First session: Start an ACID transaction T1 for a randomly selected O_KEY, L_KEY and DELTA. The ACID transaction was suspended prior to Commit.
2. Second session: Start a second ACID transaction T2, which selects random values of PS_PARTKEY and PS_SUPPKEY and returns all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal to the selected values.
3. T2 completed.
4. T1 was allowed to complete.
5. It was verified that the appropriate rows in the ORDERS, LINEITEM and HISTORY tables were changed.

Isolation Test 6

This test demonstrates that the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

The following steps were performed to successfully conduct this test:

1. First session: A transaction T1, which executes TPC-D query 1 (from TPC-D spec clause 2.3) with DELTA=0, was started.
2. Second session: Before T1 completed, an ACID transaction T2 with randomly selected values of O_KEY, L_KEY and DELTA, was started.
3. Third Session: Before T1 completed, a transaction T3, which executes TPC-D query 1 with a randomly selected value of DELTA (not equal to 0), was started.
4. T1 completed.
5. T2 completed.
6. T3 completed.
7. It was verified that the appropriate rows in the ORDERS, LINEITEM and HISTORY tables were changed.

Durability Requirements

The SUT must guarantee durability: the ability to preserve the effects of committed transactions and ensure database consistency after recovery from any one of the failures listed in Clause 3.5.3.

Permanent Unrecoverable Failure of Any Durable Medium

The tests were conducted on the qualification database . The steps performed are shown below.

Failure of Durable Medium Containing Database Tables

1. The complete database is backed up to disk.
2. The consistency test was verified.
3. The current count of the total number of records in the HISTORY table was determined giving hist1.
4. A test to run 100 ACID transactions on each execution was started.
5. One of the disks containing database tables was powered off after at least 20 ACID transaction had completed.
6. The applications running the ACID transactions terminated after receiving an error from DB2.
7. The disk from step 4 was powered back on.
8. The DB2 database was restored from the backup copy in step 1.
9. DB2 was restarted and its transaction log was used to roll forward the transactions that had completed but had not been written to disk before the failure.
10. Step 3 was performed giving hist2. It was verified that hist2 - hist1 was equal or greater than the number of records in the success file.
11. Consistency condition was verified.

Failure of Recovery Log Data and System Crash

1. The consistency test was verified.
2. The current count of the total number of records in the HISTORY table was determined giving hist1.
3. A test to run 100 ACID transactions on each execution was started.
4. One of the disks containing DB2 transaction log data was powered off after at least 20 ACID transactions had completed.
5. The run was not affected because of log mirroring and ACID transactions continued to execute successfully.
6. The system was shut down by switching off the Emergency Power Off button, after at least another 20 transactions had completed.
7. The system was powered back on and rebooted.

8. The mirrored partition on the disk was reestablished and resynchronized.
9. Step 2 was performed, giving hist2. It was verified that hist2 - hist1 was equal or greater than the number of records in the success file.
10. Consistency test was verified.

Clause 4: Scaling and Database Population Related Items

Cardinality of Tables

The cardinality (e.g., the number of rows) of each table of the test database, as it existed at the completion of the database load (see Clause 4.2.5), must be disclosed.

The following table contains the TPC Benchmark D defined tables and the number of rows for each table as they existed upon build completion.

Table Name	Rows
Lineitem	600,037,902
Orders	150,000,000
Customer	15,000,000
Supplier	1,000,000
Part	20,000,000
Partsupp	80,000,000
Nation	25
Region	5

Distribution of Tables and Logs

The distribution of tables and logs across all media must be explicitly depicted for the tested and priced systems.

The priced configuration used 83 disks of 4.51GB each; 80 of these disks were in rack-mounted external enclosures. The remaining three disks were housed in the SUT's internal drive bays.

Three Internal Disks

The three internal disks provided a total formatted capacity of 13.53GB (3 x 4.51GB) but were made into a RAID-5 array with a total formatted capacity of 8606MB. The database recovery logs were kept on this RAID-5 array.

Eighty External Disks

The 80 external disks were arranged in eight RAID-0 arrays of 43030MB of formatted capacity each. Each RAID-0 array was seen by NT to be one 43030MB disk. The first of these eight "disks" had a 2048MB boot partition (C: drive). Each of the remaining seven disks contributed 2048MB to an NT stripe set of 14336MB (7 x 2048MB). This stripe set was used to hold the pagefile and the UF flat data files and to store miscellaneous files.

Subtracting 2048MB (boot partition) from each of the eight 43030MB disks left 40982MB per disk, which was apportioned as follows:

- 10500MB per array for the LINEITEM_TABLE tablespace
- 14000MB per array for the LINEITEM_INDEXES tablespace
- 7500MB per array for the OTHER_STUFF tablespace
- 8982MB per array for the TEMP_TABLES tablespace

Thus, each of these four tablespaces was striped across all eight "disks." For each of the four tablespaces, these eight disks were represented by four drive letters. Each drive letter represented two partitions in an NT stripe set. For example, the LINEITEM_TABLE tablespace was on drives D, E, F and G where drive D is an NT stripe set of a 10500MB partition on Disk #0 and a 10500MB partition on Disk #1.

Disk 0	C: 2048 MB	D: 10500	H: 14000	L: 7500	P: 8982
Disk 1	X: 2048	D: 10500	H: 14000	L: 7500	P: 8982
Disk 2	X: 2048	E: 10500	I: 14000	M: 7500	Q: 8982
Disk 3	X: 2048	E: 10500	I: 14000	M: 7500	Q: 8982
Disk 4	X: 2048	F: 10500	J: 14000	N: 7500	R: 8982
Disk 5	X: 2048	F: 10500	J: 14000	N: 7500	R: 8982
Disk 6	X: 2048	G: 10500	K: 14000	O: 7500	S: 8982
Disk 7	X: 2048	G: 10500	K: 14000	O: 7500	S: 8982

Thus, the total disk capacity for the LINEITEM_TABLE tablespace was 84000MB (8 x 10500MB), and so on for the remaining three tablespaces.

Note: An additional 23 disks (20 x 4.51GB external, plus 3 x 9.1GB internal) were used to store the 100GB flat data files and were not priced.

Implementation of RAID

Implementations may use some form of RAID to ensure availability. If used for data, auxiliary storage (e.g., indexes) or temporary space the level of RAID used must be disclosed for each device.

RAID-5 was used for the database recovery logs. RAID-0 was used elsewhere.

DBGEN Modifications

The version number, release number, modification number, and patch level of DBGEN must be disclosed. Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The standard distribution DBGEN version 1.3.1 was used for database population.

Table Contents

The contents of the first 10 rows of each table in the test database must be disclosed.

Appendix C.2 lists the contents of the first 10 rows of each table in the test database.

Database Loading

The database load time for the test database (see Clause 4.3) must be disclosed.

The Numerical Quantities Summary contains the database load times for the system tested in this full disclosure.

Data Storage Ratio

The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in Clause 4.1.3.1. The ratio must be reported to the nearest 1/100th, rounded up.

The calculation of the data storage ratio is shown in the following table.

Disk Type	Number of Disks	Space per Disk	Total Disk Space	Database Size	Data Storage Ratio
Wide Ultra SCSI (10K rpm)	83	4.51GB	374.33GB	100	3.75

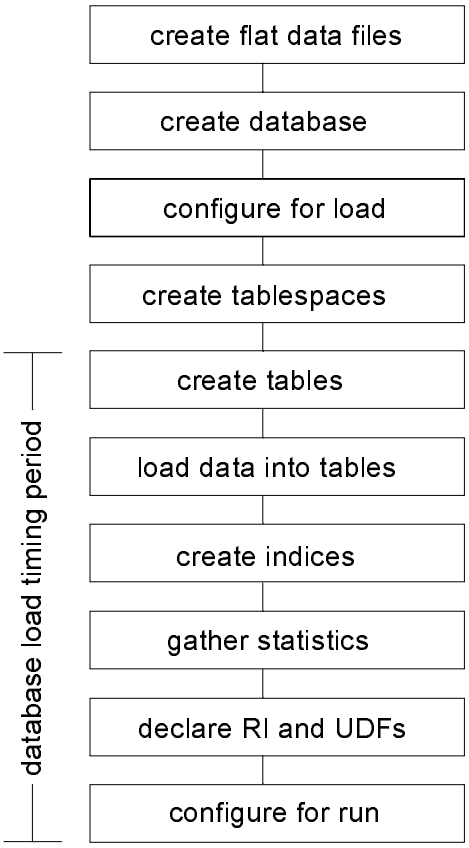
Details of Database Loading

The details of the database load must be disclosed, including a block diagram illustrating the overall process. Disclosure of the load procedure includes all steps, scripts, input and configuration files required to completely reproduce the test and qualification databases.

Flat files for each of the tables were created using DBGEN.

The tables were loaded as depicted in Figure 4-1.

Figure 4-1. Database Load Procedure



Clause 5: Performance Metrics and Execution Rules

Power Test

Implementation

The details of the steps followed to implement the power test (e.g., system boot, database restart) must be disclosed.

The following steps are performed prior to running the power test:

1. The database manager is stopped. This clears any DB2 buffers from memory.
2. The database manager is restarted.

The power test is then initiated. A warm-up run is not used. The power test comprises a single stream that runs the update function UF1, followed by the queries using the sequencing given by stream number 0 in Appendix A of the TPC-D specification, followed by the update function UF2. The tpcdbatch program (see description of the Implementation Specific Layer in the TPC-D specification) is used to run all of the queries.

Timing Intervals

The timingintervals (see Clause 5.3.6) for each query of the measured set (i.e., the query set that follows the warm-up set, see step 4 of Clause 5.3.2.2) and for both update functions must be reported for the power test.

The Numerical Quantities Summary contains the timing intervals for the power test.

Throughput Test

The number of query streams used for the throughput test must be disclosed.

A single query stream throughput metric was calculated using the timings from the power test, as indicated in Clause 5.3.1.4. A separate throughput test was not run.

Stream Times

The start time and finish time for each query execution stream must be reported for the throughput test.

The Numerical Quantities Summary contains the start and stop times for the query execution streams run on the system reported.

Measurement Interval

The total elapsed time for the measurement interval (see Clause 5.3.5) must be reported for the throughput test.

The Numerical Quantities Summary contains the timing intervals for the throughput test run on the system reported.

Update Functions

The start time and finish time for each update function in the update stream must be reported for the throughput test.

The Numerical Quantities Summary contains the timings for the update functions.

Timing Intervals

The timing intervals (see Clause 5.3.6) for each query of each stream and for each update function must be reported for the throughput test.

A throughput test was not run. Timings for the power test appear in the Numerical Quantities Summary.

Performance Metrics

The computer performance metrics, related numerical quantities, and the price/performance metric must be reported.

The Numerical Quantities Summary contains the performance metrics, related numerical quantities, and the price/performance metric for the system reported.

Reproducibility

A description of the method used to determine reproducibility of the measurement results must be reported. This must include the performance metrics (QppD, QthD, and QphD) from the reproducibility runs (see Clause 5.4.6).

Two consecutive runs were performed. The following table contains the reproducibility metrics for the system reported.

	QppD	QthD	QphD
Run 1	988.1	362.4	598.4
Run 2	987.2	362.4	598.1

Clause 6: SUT and Driver Implementation

Driver

A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.

Appendix D, "Implementation-Specific Layer and Driver Source Code," contains the source code used for the driver and all scripts used in connection with it.

The power test is invoked by calling tpcdbatch with the stream number 0 specified, an indication that the update functions must be run, and the SQL file that contains the power stream queries.

Implementation-Specific Layer

If an implementation-specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.

The implementation-specific layer is a single executable SQL application that uses embedded dynamic SQL to process the EQT generated by QGEN. The application is called tpcdbatch to indicate that it processes a batch of TPC-D queries, although it is completely capable of processing any arbitrary SQL statement (both DML and DDL).

A separate instance of tpcdbatch is invoked for each stream. Each instance establishes a distinct connection to the database server through which the EQT is transmitted to the database and the results are returned through the implementation-specific layer to the driver. When an instance of tpcdbatch is invoked, it is provided with a context of whether it is running a power test, query stream or update stream, as well as an input file containing the 17 queries and/or update functions. Then, tpcdbatch connects to the database, performs any session initialization as well as prepares output files required by the auditor. Then it proceeds to read from the input file and processes each query or update function in turn.

For queries, each query is prepared, and a cursor is opened and used to fetch the required number of rows. After the last row has been retrieved, a commit is issued. For UF1, the data is split into n equal portions for both the lineitem and orders tables taking care that the records for the same orderkey remain in the same set. For UF2, the data is split into m equal portions. During the run, when tpcdbatch encounters a call to execute UF1, it forks off n children, each of which performs an insert with subselect. The insert is into the Orders and Lineitem tables. The subselect is from User Defined Table Functions, which present the flat data files as relational tables. When tpcdbatch encounters a call to executed UF2, it calls a shell script that starts m applications, one for each of the m portions of delete data. Each application reads the keys for its portion and performs the deletes using these keys.

Clause 7: Pricing Related Items

Hardware and Software Components

A detailed list of the hardware and software used in the priced system must be reported. Each item must have a vendor part number, description and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) must also be reported.

A detailed list of all hardware and software, including the 5-year price, is provided in the Executive Summary at the front of this report. The price quotations are included in Appendix F at the end of this document.

Five-Year Cost of System Configuration

The total 5-year price of the entire configuration must be reported, including hardware, software and maintenance charges. Separate component pricing is recommended. The basis of all discounts must be disclosed.

A detailed list of all hardware and software, including the 5-year price, is provided in the Executive Summary at the front of this report. The price quotations are included in Appendix F at the end of this document.

Availability Dates

The committed delivery date for general availability (availability date) of products used in the price calculations must be reported. When the priced system includes products with different availability dates, availability date reported on the Executive Summary must be the date by which all components are committed to being available. The Full Disclosure Report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided (see Clause 7.3.1.3).

The hardware and system software used in this test are generally available at the time of publication. (Service Pack 4 for NT Server 4.0 will be generally available in third quarter 1998.) The availability date for the database software is October 31, 1998.

Country-Specific Pricing

Additional Clause 7 related items may be included in the Full Disclosure Report for each country-specific priced configuration. Country-specific pricing is subject to Clause 7.1.7.

The configuration is priced for the United States of America.

Clause 9: Audit Related Items

Auditor

The auditor's name, address, phone number, and a copy of the auditor's attestation letter indicating compliance must be included in the Full Disclosure Report. (8.1.9.1)

This implementation of the TPC-D benchmark was audited by Francois Raab of Information Paradigm. The auditor's attestation letter is provided in this section.

Availability of the Full Disclosure Report

The Full Disclosure Report must be readily available to the public at a reasonable charge, similar to the charges for similar documents by the test sponsor. The report must be made available when results are made public. In order to use the phrase "TPC Benchmark™D," the Full Disclosure Report must have been submitted to the TPC Administrator as well as written permission obtained to distribute same. (8.2)

Requests for the TPC Benchmark D Full Disclosure Report should be sent to:

Transaction Processing Performance Council
c/o Shanley Public Relations
777 North First Street, Suite 600
San Jose, CA 95112-6311

or

IBM Corporation
Mail Drop 23U/205/QQ160
3039 Cornwallis Road
Research Triangle Park, NC 27709

Information Paradigm

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL

Certified Auditor

Test Sponsor: William D. Hall
Mgr., Server Systems Performance
IBM Personal Systems Group
3039 Cornwallis Road
Research Triangle Park, NC 27709

May 29, 1998

I verified the TPC Benchmark™ D performance of the following configuration:

Platform: IBM Netfinity 7000
DataBase Manager: DB2 UDB Version 5.2.0
Operating System: Microsoft Windows NT Server Version 4.0

The results were:

CPU's Speed	Memory	Disks	QppD@100GB	QthD@100GB
IBM Netfinity 7000				
4 x Pentium Pro (200 Mhz)	1MB L2/CPU 2GB Main	83 x 4.5 GB	987.2	362.4