



Red Hat – Industry Standard Benchmarks

TPC-H Benchmark™ with Sybase® IQ on HP ProLiant DL785 G6 using Red Hat® Enterprise Linux® 5.3 (102,375 QphH @ 1 TB)

TPC-H Benchmark™

**Sybase® IQ Single Application
Server Edition v15.1**

Red Hat® Enterprise Linux® 5.3

**HP ProLiant® DL785 G6
(8 sockets x 6 cores/socket = 48 cores
AMD Opteron® 8439)**

Version 1.0

February 2010





**TPC-H Benchmark™ with Sybase® IQ
on HP ProLiant DL785 G6 using
Red Hat® Enterprise Linux® 5.3
(102,375 QphH @ 1 TB)**

1801 Varsity Drive
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The following terms used in this publication are trademarks of other companies as follows:

TPC Benchmark, TPC-H, QphH and \$/QphH are trademarks of the Transaction Processing Performance Council (TPC).

Sybase is a registered trademark of Sybase, Inc.

Hewlett-Packard, the Hewlett-Packard logo, Insight Manager, ProLiant, HP Serviceguard and HP StorageWorks are registered trademarks of Hewlett-Packard Development Company, L.P., in the United States and other countries..

Linux is a registered trademark of Linus Torvalds

Red Hat, Red Hat Enterprise Linux and the Red Hat "Shadowman" logo are registered trademarks of Red Hat, Inc. in the United States and other countries

All other trademarks referenced herein are the property of their respective owners.

© 2010 by Red Hat, Inc. This material may be distributed only subject to the terms and conditions set forth in the Open Publication License, V1.0 or later (the latest version is presently available at <http://www.opencontent.org/openpub/>).

The information contained herein is subject to change without notice. Red Hat, Inc. and IBM Inc. shall not be liable for technical or editorial errors or omissions contained herein.

Distribution of modified versions of this document is prohibited without the explicit permission of Red Hat Inc. and Vertica Systems Inc.

Distribution of this work or derivative of this work in any standard (paper) book form for commercial purposes is prohibited unless prior permission is obtained from Red Hat Inc. and IBM Inc.

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E



Table of Contents

1. Executive Summary	4
2. Overview of the TPC Benchmark H: The Decision Support Benchmark	6
2.1 The TPC-H Business Environment	7
2.2 The TPC-H Schema	9
2.3 TPC-H Queries	10
3. TPC-H Methodology & Metrics	13
3.1 TPC-H Runs	13
3.1.1 TPC-H Power Test	13
3.1.2 TPC-H Throughput Test	14
3.2 TPC-H Performance Metric	14
3.3 TPC-H Price/Performance Metric	14
3.4 Comparing TPC-H Results	15
4. Hardware Configuration	16
5. Software Configuration	18
5.1 Parameter Settings	18
6. TPC-H Benchmark Results	19
7. References	21
Appendix A Parameter Settings	22
A.1 bm_setup	22
A.2 sysctl.conf	24
A.3 tpch.cfg	25
A.4 options.sql	25



1. Executive Summary

The recently published TPC-H benchmark using a Sybase database on a HP ProLiant DL785 G6 server provide Red Hat Enterprise Linux a number of leadership points in the 1TB TPC-H scale factor category:

- #1 performance among x86 results in the non-clustered category (see Figure 1)
- #1 Eight-socket performance
- #1 performance among Linux OS results in the non-clustered category
- #1 performance among Sybase IQ results in the non-clustered category

RHEL is the best platform for running Decision Support applications on x86_64 hardware. Figure 1 compares the TPC-H Composite Queries Per Hour (QphH) metric for RHEL and Windows Server 2008 on identical hardware.

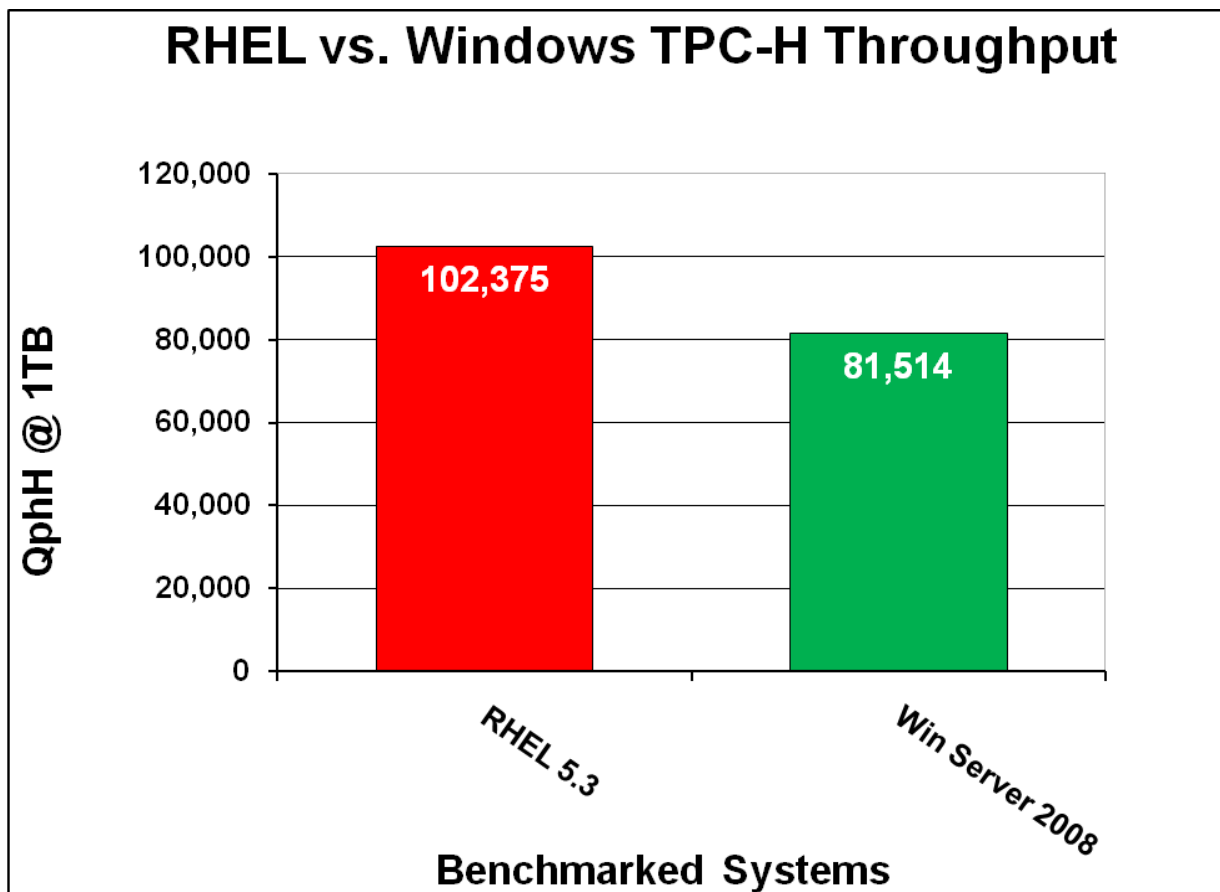


Figure 1: RHEL versus Windows TPC-H Results



In February 2010, HP published the following TPC-H benchmark results:

Platform: **HP ProLiant DL785 G6**
Database Manager: **Sybase IQ Single Application Server Edition v15.1 ESD#1**
Operating System: **Red Hat Enterprise Linux 5.3**
Performance: **102,375.3 QphH**
Price/Performance: **\$3.63/QphH**

For more details on the Red Hat Enterprise Linux TPC-H benchmark refer to:

1. TPC-H Benchmark Executive Summary by HP.
http://www.tpc.org/results/individual_results/HP/HP_TPC_H_020110_ES-DL785-Sybase-1TB.pdf
2. TPC-H Benchmark Full Disclosure Report by HP.
http://www.tpc.org/results/FDR/tpch/HP_TPC_H_020110_FDR-DL785-Sybase-1TB.pdf

In November 2009, HP published following TPC-H benchmark results:

Platform: **HP ProLiant DL785 G6**
Database Manager: **Microsoft SQL Server 2008 Enterprise Edition**
Operating System: **Microsoft Windows Server 2008 R2 Enterprise Edition**
Performance: **81,515 QphH**
Price/Performance: **\$2.90/QphH**

For more details on the Windows TPC-H benchmark refer to:

1. TPC-H Benchmark Executive Summary by HP.
http://www.tpc.org/results/individual_results/HP/HP_DL785_1000G_091109_ES_v2.pdf
2. TPC-H Benchmark Full Disclosure Report by HP.
http://www.tpc.org/results/FDR/tpch/HP_DL785_1000G_091109_FDR_v2.pdf



2. Overview of the TPC Benchmark H: The Decision Support Benchmark

The TPC Benchmark [™] H (TPC-H) is a decision support benchmark. It consists of a suite of business oriented ad-hoc queries and concurrent data modifications. The queries and the data populating the database have been chosen to have broad industry-wide 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-H evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries(e.g., via a point and click GUI interface);
- 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 database.

TPC-H operations are modeled as follows:

- The database is continuously available 24 hours a day, 7 days a week, for ad-hoc queries from multiple end users and updates against all tables, except possibly during infrequent (e.g., once a month) maintenance sessions;
- The TPC-H database tracks, possibly with some delay, the state of the OLTP database through on-going updates which batch together a number of modifications impacting some part of the decision support database;
- Due to the world-wide nature of the business data stored in the TPC-H 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 ACID (Atomicity, Consistency, Isolation, Durability) 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 ten million rows representing a raw storage capacity of about 1 GB. Compliant benchmark implementations may also use one of the larger permissible database populations (e.g. 1000 GB).
- The performance metrics reported by TPC-H measure multiple aspects of the capability of the system to process queries. The TPC-H metric at the selected size (QphH@Size) is the performance metric. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components. The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.
- The TPC-H database must be 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 implementers are not required to have implemented a specific SQL standard in full. TPC-D uses terminology and metrics that are similar to other benchmarks, originated by the TPC and others. Such similarity in terminology does not in any way imply that TPC-H results are comparable to other benchmarks. The only benchmark results comparable to TPC-H are other TPC-H results compliant with the same revision.
- Despite the fact that this benchmark offers a rich environment representative of many decision support systems, this benchmark does not reflect the entire range of decision support requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.
- Benchmark results are highly dependent upon workload, specific application requirements, and systems design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, TPC-H should not be used as a substitute for a specific customer application benchmarking when critical capacity planning and/or product evaluation decisions are contemplated.

2.1 The TPC-H Business Environment

Figure 2 illustrates the TPC-H business environment and highlights the basic differences between TPC-H and other TPC benchmarks.



Other TPC benchmarks model the operational end of the business environment where transactions are executed on a real time basis. The TPC-H benchmark, however, models the analysis end of the business environment where trends are computed and refined data are produced to support the making of sound business decisions. In OLTP benchmarks the raw data flow into the OLTP database from various sources where it is maintained for some period of time. In TPC-H, periodic refresh functions are performed against a DSS database whose content is queried on behalf of or by various decision makers.

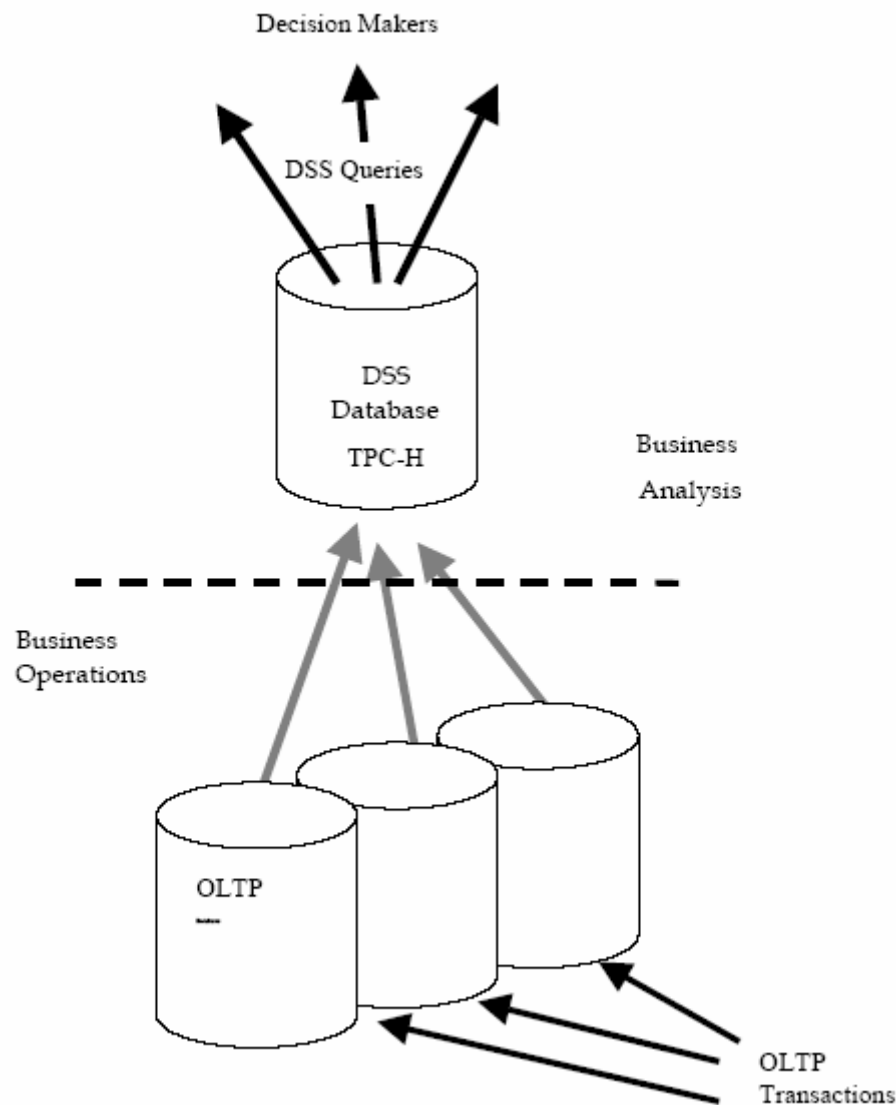


Figure 2: The TPC-H Business Environment



2.2 The TPC-H Schema

The components of the TPC-H database are defined to consist of eight separate and individual tables (the Base Tables). The relationships between columns of these tables are illustrated in Figure 3: The TPC-H Schema.

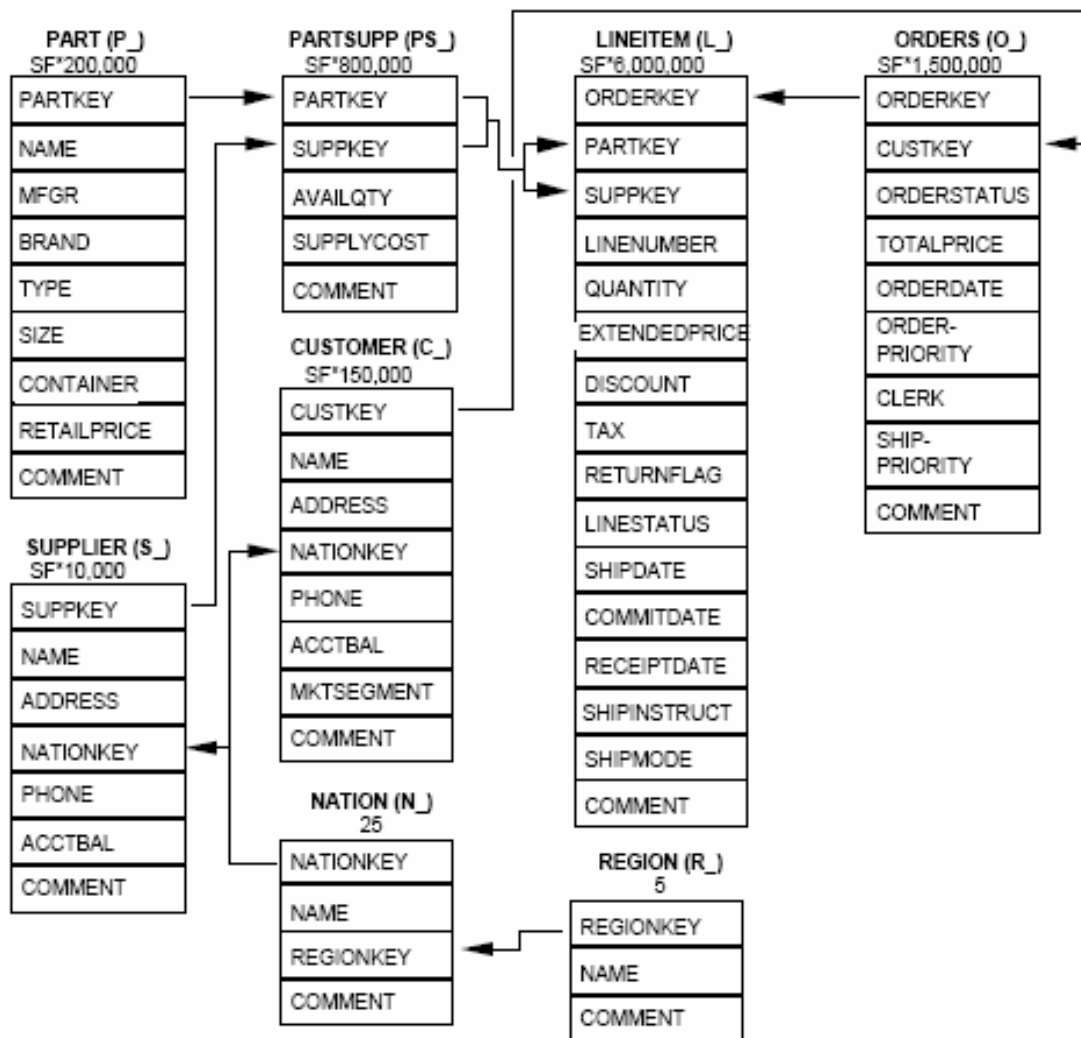


Figure 3: The TPC-H Schema



2.3 TPC-H Queries

There are 22 TPC-H 'Queries' plus 2 'Refresh Functions':

1. Pricing Summary Report Query (Q1)
This query reports the amount of business that was billed, shipped, and returned.
2. Minimum Cost Supplier Query (Q2)
This query finds which supplier should be selected to place an order for a given part in a given region.
3. Shipping Priority Query (Q3)
This query retrieves the 10 unshipped orders with the highest value.
4. Order Priority Checking Query (Q4)
This query determines how well the order priority system is working and gives an assessment of customer satisfaction.
5. Local Supplier Volume Query (Q5)
This query lists the revenue volume done through local suppliers.
6. Forecasting Revenue Change Query (Q6)
This query quantifies the amount of revenue increase that would have resulted from eliminating certain company-wide discounts in a given percentage range in a given year. Asking this type of "what if" query can be used to look for ways to increase revenues.
7. Volume Shipping Query (Q7)
This query determines the value of goods shipped between certain nations to help in the re-negotiation of shipping contracts.
8. National Market Share Query (Q8)
This query determines how the market share of a given nation within a given region has changed over two years for a given part type.
9. Product Type Profit Measure Query (Q9)
This query determines how much profit is made on a given line of parts, broken out by supplier nation and year.
10. Returned Item Reporting Query (Q10)
The query identifies customers who might be having problems with the parts that are shipped to them.
11. Important Stock Identification Query (Q11)
This query finds the most important subset of suppliers' stock in a given nation.



12. Shipping Modes and Order Priority Query (Q12)

This query determines whether selecting less expensive modes of shipping is negatively affecting the critical-prior-ity orders by causing more parts to be received by customers after the committed date.

13. Customer Distribution Query (Q13)

This query seeks relationships between customers and the size of their orders.

14. Promotion Effect Query (Q14)

This query monitors the market response to a promotion such as TV advertisements or a special campaign.

15. Parts/Supplier Relationship Query (Q16)

This query finds out how many suppliers can supply parts with given attributes. It might be used, for example, to determine whether there is a sufficient number of suppliers for heavily ordered parts.

16. Small-Quantity-Order Revenue Query (Q17)

This query determines how much average yearly revenue would be lost if orders were no longer filled for small quantities of certain parts. This may reduce overhead expenses by concentrating sales on larger shipments.

17. Large Volume Customer Query (Q18)

The Large Volume Customer Query ranks customers based on their having placed a large quantity order. Large quantity orders are defined as those orders whose total quantity is above a certain level.

18. Discounted Revenue Query (Q19)

The Discounted Revenue Query reports the gross discounted revenue attributed to the sale of selected parts handled in a particular manner. This query is an example of code such as might be produced programmatically by a data mining tool.

19. Potential Part Promotion Query (Q20)

The Potential Part Promotion Query identifies suppliers in a particular nation having selected parts that may be can-didates for a promotional offer.

20. Suppliers Who Kept Orders Waiting Query (Q21)

This query identifies certain suppliers who were not able to ship required parts in a timely manner.

21. Global Sales Opportunity Query (Q22)

The Global Sales Opportunity Query identifies geographies where there are customers who may be likely to make a purchase.

22. New Sales Refresh Function (RF1)

This refresh function adds new sales information to the database.



23. Old Sales Refresh Function (RF2)

This refresh function removes old sales information from the database.



3. TPC-H Methodology & Metrics

TPC-H is a benchmark that simulates a Decision Support System or Business Intelligence database environment. The performance of a system is measured when the system is tasked with providing answers for business analyses on a dataset. These analyses include:

- Pricing & Promotions Analysis
- Supply & Demand Management Analysis
- Profit & Revenue Management Analysis
- Customer Satisfaction Studies
- Market Share Studies
- Shipping Management Analysis

The server system runs a read-intensive Decision Support System (DSS) style database to provide the results for the business analyses. The DSS database is designed to mimic a repository of commercial order-processing Online Transaction Processing Databases. The analyses are performed on 100GB, 300GB, 1000GB, 10,000GB, 30,000GB or even 100,000GB scale factor (=size) datasets.

3.1 TPC-H Runs

The business analysis workload of the TPC-H consists of “ad-hoc” database queries with a high degree of complexity. The intention is that the exact queries are not known beforehand, which is often the case for DSS databases.

The TPC-H benchmark runs the business analyses in two different ways – it performs a Power Test and a Throughput Test.

3.1.1 TPC-H Power Test

The **TPC-H Power Test** workload performs significant sequential disk I/O as the single user’s queries perform index and table scans over the substantial datasets. The DSS database’s query optimizer typically splits the query into a number of parts and runs them in parallel, distributing the data retrieval across the disk arrays attached to the server system. The workload generally consists of read activity, though there are bursts of database inserts and updates performed during queries called “refresh functions”, which simulate OLTP databases periodically feeding data into the DSS system. There is typically a considerable amount of tempdb activity as the results of complex joins, aggregations and interim results are stored for the business analyses.

The TPC-H Power Test measures the query execution power of the system when connected with a single user. It runs the analyses in a serial manner – the queries and update functions



run one at a time and the elapsed time is measured. This test yields the **TPC-H Power @ Size metric**.

3.1.2 TPC-H Throughput Test

The **TPC-H Throughput Test** is essentially several instances of the Power Test being run simultaneously using different query parameters. This results in database buffer cache contention in system memory as the various streams compete for buffer space. The sequential read activity tends to become more random as the multiple streams access the disk drives simultaneously. The processors on the system also become busier from running more processes and switching tasks.

The TPC-H Throughput Test measures the ability of the system to process the most queries (and update functions) in the least amount of time in a multi-user environment. Each simulated user runs its own version of the TPC-H Power Test simultaneously. The number of simulated users for this test is selectable with a minimum number of users or “streams” detailed by the TPC-H specification for each scale factor. This test yields the **TPC-H Throughput @ Size metric**.

3.2 TPC-H Performance Metric

The primary performance metric for the TPC-H benchmark is the Composite Queries-Per-Hour (QphH). Composite Queries Per Hour (QphH) is calculated as a geometric mean from the results from the TPC-H Power Test and the TPC-H Throughput Test. The more QphH, the better the benchmark result. As per TPC rules, QphH is always represented as TPC-H **QphH @ Size**.

$$QphH@Size = \sqrt{Power@Size * Throughput@Size}$$

3.3 TPC-H Price/Performance Metric

TPC-H also has a pricing component, which is the total cost of the benchmark system (hardware and software) including 3 years worth of 24x7 4-hour response time support, divided by the Composite Queries Per Hour (ie price/performance). This is represented as **TPC-H \$(currency) / QphH @ Size**. The lower the price/performance, the more cost effective the system.



3.4 Comparing TPC-H Results

When comparing TPC-H results, be sure to compare results of the same scale factor. For example, comparing a system that ran 19323QphH@100GB with a system that ran 12,225QphH@300GB is not a fair comparison, whereas comparing a system that ran 102,375 QphH@1TB with a system that ran 81,515 QphH@1TB is.

- The overall TPC-H performance of a group of systems may be compared using the TPC-H Composite Queries-Per-Hour (QphH) metric.
- The price/performance of a group of systems may be compared using the \$/QphH metric.
- The system power of a single user may be compared using the TPC-H Power metric.
- The throughput and ability to process the most queries in the shortest amount of time may be compared using the TPC-H Throughput metric.

All published TPC-H results are stored on the TPC website at <http://www.tpc.org> . When making comparisons, it is useful to download the 3-4 page benchmark Executive Summary rather than try to wade through the more detailed Full Disclosure Report. The Executive Summary will include the price/performance, QphH, TPC-H Power, and TPC-H Throughput metrics, along with a full price sheet including all of the hardware and software used in the benchmark.



4. Hardware Configuration

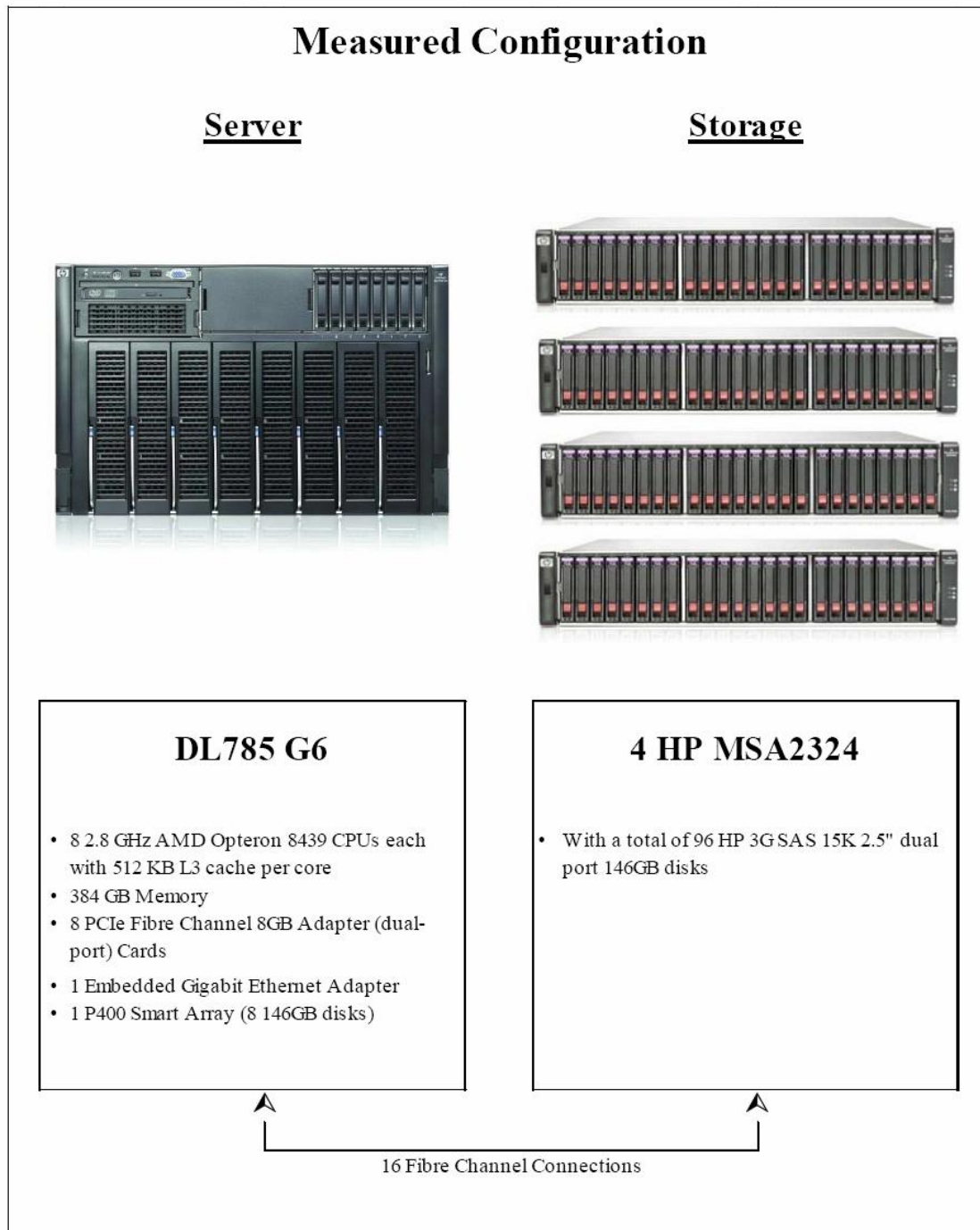


Figure 4: Hardware Configuration



Number of Nodes	1
Processors/Cores/Threads/Type	8/48/48/AMD Opteron 8439 2.8 GHz, 512 KB L3 cache per core
Memory	384 GB
Disk Drives	4 MSA2324 with a total of 96, 146GB disks 8, 146GB internal disks
Total Disk Storage	15184GB (In this number one GB is defined as 1024*1024*1024 bytes)
LAN Controllers	1 Embedded Gigabit Ethernet Adapter

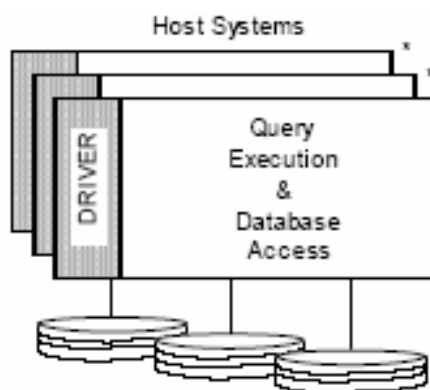


Figure 5: Internal Load Driver



5. Software Configuration

Database Manager	Sybase IQ Single Application Server Edition v15.1 ESD#1
Operating System	Red Hat Enterprise Linux 5.3

5.1 Parameter Settings

Settings must be provided for all customer-tunable parameters and options which 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 contains the Red Hat Enterprise Linux 5.3 and Sybase IQ 15.1 parameters used in this benchmark.



6. TPC-H Benchmark Results

The results of the TPC-H Benchmark are summarized below:

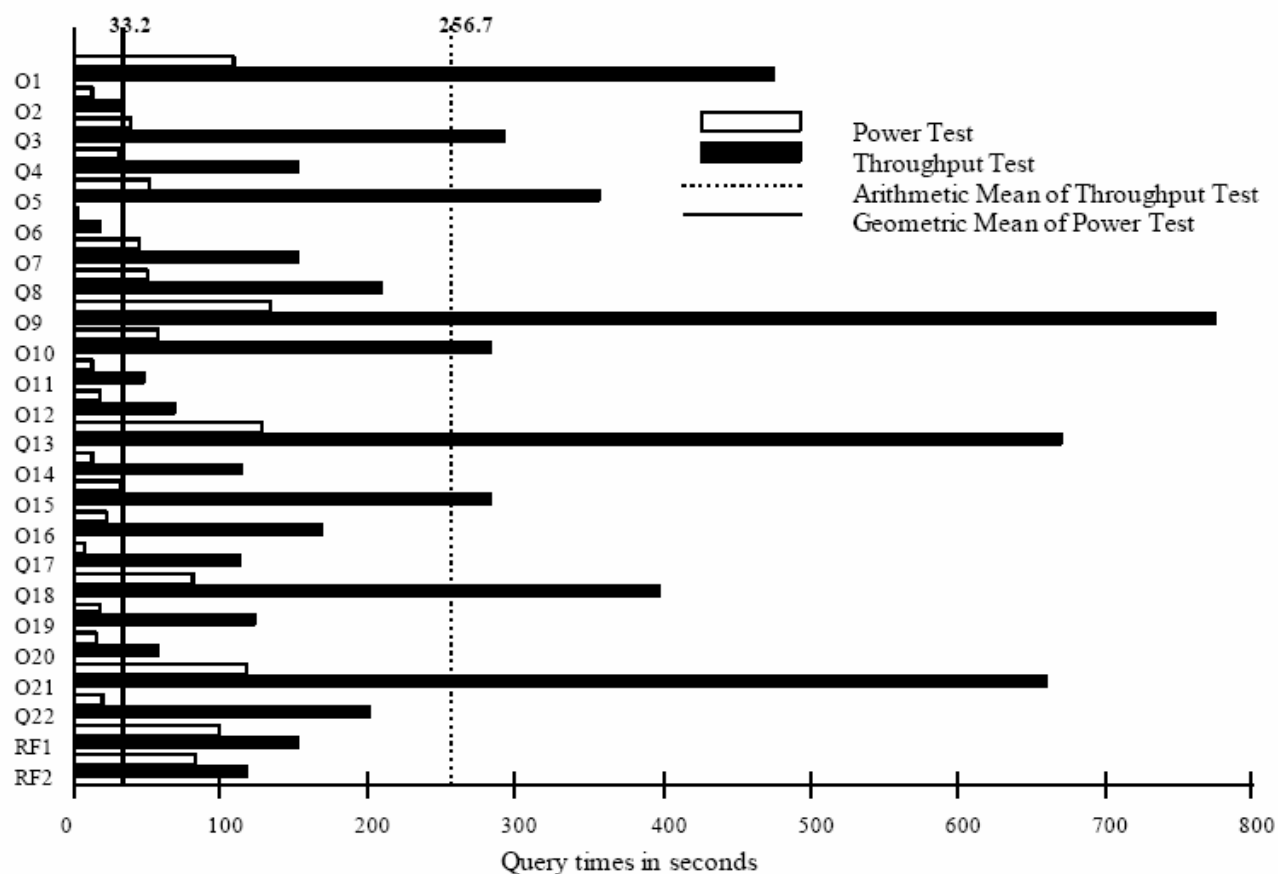


Figure 6: TPC-H Power & Throughput Test Results



Measurement Results:

Database Scaling (Scale Factor / Size)	= 1000GB
Total Data Storage/Database Size	= 15.18
Percentage Memory/Database Size	= 38.4%
Database Load Time	= 1:40:25
Query Streams for Throughput Test (S)	= 7
TPC-H Power @ 1000GB	= 108,436.8
TPC-H Throughput @ 1000GB	= 96,652.7
TPC-H Composite Query-per-Hour Metric (QphH@1000GB)	= 102,375.3
Total System Price Over 3 Years	= 371,721
TPC-H Price/Performance Metric (\$/QphH@1000GB)	= 3.63

Measurement Intervals:

Measurement Interval in Throughput Test (Ts) in seconds	= 5,736 seconds
---	-----------------



7. References

1. TPC BENCHMARKTM H (Decision Support) - Standard Specification Revision 2.9.0
<http://www.tpc.org/tpch/spec/tpch2.9.0.doc>
2. TPC-H Benchmark Executive Summary by HP.
http://www.tpc.org/results/individual_results/HP/HP_TPCH_020110_ES-DL785-Sybase-1TB.pdf
3. TPC-H Benchmark Full Disclosure Report by HP.
http://www.tpc.org/results/FDR/tpch/HP_TPCH_020110_FDR-DL785-Sybase-1TB.pdf



Appendix A Parameter Settings

A.1 *bm_setup*

```
#!/bin/ksh
# Sybase Product Environment variables
#
if [ -z "$1" ]
then
echo 'Usage: bm_setup <SCALE_FACTOR>'
return 1
fi
# set prompt and terminal type
export PS1="$HOSTNAME:\w $ "
export TERM=xterm
SCALE_FACTOR=$1; export SCALE_FACTOR
VERSION=15_0; export VERSION
DATA_DIR=/flat/scale_${SCALE_FACTOR}; export
DATA_DIR
SYBASE=/sybase/IQ_151ESD1; export SYBASE
TPCH_ROOT=/sybase/tpch_audit_kit; export TPCH_ROOT
#To seperate out generated files from static files
GENERATED_FILES=$TPCH_ROOT/generated_files; export
GENERATED_FILES
CONFIG_DIR=$TPCH_ROOT/config_dir; export
CONFIG_DIR
ODBCINI=$CONFIG_DIR/.odbc.ini; export ODBCINI
DEV_DIR=$TPCH_ROOT/devices; export DEV_DIR
QUERYGEN=$TPCH_ROOT/querygen; export QUERYGEN
#QUERYGEN=$TPCH_ROOT/querygen_debug; export
QUERYGEN
STREAMS=$TPCH_ROOT/streams_${SCALE_FACTOR};
export STREAMS
QUERIES=$TPCH_ROOT/queries_${SCALE_FACTOR};
export QUERIES
RESULTS=$TPCH_ROOT/Results; export RESULTS
TEMPLATES=$TPCH_ROOT/templates; export
TEMPLATES
TEMPFILES=$TPCH_ROOT/tempfiles; export TEMPFILES
SETUP=$TPCH_ROOT/setup; export SETUP
REFERENCE_FILES=$TPCH_ROOT/dbgen_src/reference;
export REFERENCE_FILES
REFERENCE_DATA=$TPCH_ROOT/reference_data/$SCALE_FACTOR; export REFERENCE_DATA
```



```
SQL=$SETUP/sql; export SQL
SCRIPTS=$SETUP/scripts; export SCRIPTS
SYBASE_OCS="OCS-15_0"; export SYBASE_OCS
SYBROOT="$SYBASE"; export SYBROOT
if [ $VERSION = 12_7 ];
then
SYBASE_IQ="ASIQ-12_7"; export SYBASE_IQ
TEST_DBFILE="asiqdemo.db"; export
TEST_DBFILE
START_SERVER='start_asiq'; export
START_SERVER
STOP_SERVER='stop_asiq'; export STOP_SERVER
else
SYBASE_IQ="IQ-15_1"; export SYBASE_IQ
TEST_DBFILE="iqdemo.db"; export TEST_DBFILE
START_SERVER='start_iq'; export
START_SERVER
STOP_SERVER='stop_iq'; export STOP_SERVER
STOP_SERVER_WITH_CHECKPOINT='stop_iq_ch
eckpoint'; export STOP_SERVER_WITH_CHECKPOINT
fi
#Run the corresponding IQ script.
. $SYBASE/$SYBASE_IQ/$SYBASE_IQ.sh
PATH="{TPCH_ROOT}:{TPCH_ROOT}/dbgen_src/{TP
CH_ROOT}/bin:${SCRIPTS}:{TPCH_ROOT}/Report_Gene
ration/{PATH}"
export PATH
INCLUDE="{SYBASE}/{SYBASE_OCS}/include:$INCL
UDE"
export INCLUDE
LIB="{SYBASE}/{SYBASE_OCS}/lib:$LIB"
export LIB
if [ $VERSION = 12_7 ];
then
alias ISQL="isql -UDBA -PSQL -Stpch_${VERSION} -I
${CONFIG_DIR}/interfaces"
ISQL="isql -UDBA -PSQL -Stpch_${VERSION} -I
${CONFIG_DIR}/interfaces"; export ISQL
else
alias ISQL="iqisql -UDBA -Psql -Stpch_${VERSION} -I
${CONFIG_DIR}/interfaces -w 500"
ISQL="iqisql -UDBA -Psql -Stpch_${VERSION} -I
${CONFIG_DIR}/interfaces -w 500"; export ISQL
fi
if [ -z "$STREAM_COUNT" ];
then
STREAM_COUNT=`sf_to_streamscount`
```



```
$SCALE_FACTOR`; export STREAM_COUNT  
fi
```

A.2 *sysctl.conf*

```
# Kernel sysctl configuration file for Red Hat Linux  
#  
# For binary values, 0 is disabled, 1 is enabled. See sysctl(8)  
and  
# sysctl.conf(5) for more details.  
# Controls IP packet forwarding  
net.ipv4.ip_forward = 0  
# Controls source route verification  
net.ipv4.conf.default.rp_filter = 1  
# Do not accept source routing  
net.ipv4.conf.default.accept_source_route = 0  
# Controls the System Request debugging functionality of the  
kernel  
kernel.sysrq = 0  
# Controls whether core dumps will append the PID to the core  
filename  
# Useful for debugging multi-threaded applications  
kernel.core_uses_pid = 1  
# Controls the use of TCP syncookies  
net.ipv4.tcp_syncookies = 1  
# Controls the maximum size of a message, in bytes  
kernel.msgmnb = 65536  
# Controls the default maximum size of a message queue  
kernel.msgmax = 65536  
# Controls the maximum shared segment size, in bytes  
kernel.shmmax = 68719476736  
# Controls the maximum number of shared memory segments,  
in pages  
kernel.shmall = 4294967296  
# SEMMSL - maximum number of System V IPC semaphores  
per identifier  
# SEMMNS - number of System V IPC system-wide  
semaphores  
# SEMOPM - maximum number of semaphore operations  
# SEMMNI - number of System V IPC system-wide  
semaphore identifier  
kernel.sem = 500 32000 100 128  
vm.swappiness = 0  
vm.pagecache = 1
```




A.3 *tpch.cfg*

```
# tpch.cfg
# -----
# Default startup parameters for the ASIQ demo database
# -----
-n tpch_15_0_sybase
-c 64M
-gd all
-gm 25
-gc 5000
-gr 5000
-gp 4096
-tl 0
-iqtss 240
-iqmt 1400
-iqmc 259000
-iqtc 119000
-iqpartition 64
-iqgovern 10
```

A.4 *options.sql*

```
SET OPTION "PUBLIC".STRING_RTRUNCATION='Off';
SET OPTION "PUBLIC".Allow_Nulls_By_Default='Off';
SET OPTION "PUBLIC".Append_Load='On';
SET OPTION "PUBLIC".Force_No_Scroll_Cursors='On';
SET OPTION "PUBLIC".Load_Memory_Mb=0;
SET OPTION "PUBLIC".Minimize_Storage='On';
SET OPTION "PUBLIC".Notify_Modulus=10000000;
SET OPTION "PUBLIC".Row_Counts='On';
SET OPTION
"public".SignificantDigitsForDoubleEquality=12;
SET OPTION "PUBLIC".QUERY_TEMP_SPACE_LIMIT=0;
SET OPTION "PUBLIC".Hash_Thrashing_Percent=100;
--Performance Options
SET OPTION "PUBLIC".Max_Hash_Rows=5000000;
SET OPTION
"PUBLIC".Default_Having_Selectivity_PPM=10;
SET OPTION "PUBLIC".subquery_flattening_preference=3;
SET OPTION
"PUBLIC".FP_PREDICATE_WORKUNIT_PAGES=50;
SET OPTION "PUBLIC".Garray_Fill_Factor_Percent=3;
SET OPTION
"PUBLIC".Max_IQ_Threads_Per_Connection=500;
```



```
SET OPTION "PUBLIC".Max_Query_Parallelism=64;  
SET OPTION "PUBLIC".Sweeper_Threads_Percent=8;  
SET OPTION "PUBLIC".Wash_Area_Buffers_Percent=30;  
SET OPTION "PUBLIC".Prefetch_Threads_Percent=25;
```