TPC Benchmark™ C

Full Disclosure Report

First Edition

30–Aug–2021

Using

Goldilocks v3.1 Standard Edition

on

LTechKorea L224S-D

First Edition: 30-Aug-2021

TTA, Telecommunications Technology Association, believes that all the information in this document is accurate as of the publication date. The information in this document is subject to change without notice. TTA, the sponsor of this benchmark test, assumes no responsibility for any errors that may appear in this document. The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, the sponsor provides no warranty of the pricing information in this document.

Benchmark 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, the TPC Benchmark™ C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtain
ned in other operating environments may vary significantly. No warranty of system performance or price/performance is expressed or implied in this report.

Trademarks

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

- *TPC Benchmark, TPC-C, and tpmC are trademarks of the Transaction Processing Performance Council*
- *TTA is a registered trademark of Telecommunications Technology Association*
- *Goldilocks is a registered trademark of SUNJESOFT, Inc.*
- *JBoss is a registered trademark of RedHat, Inc.*
- *Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation.*
- *All other trademarks and copyrights are properties of their respective owners.*

Table of Contents

TABLE OF CONTENTS **3**

ABSTRACT **5**

PREFACE **6**

GENERAL ITEMS **11**

0.1 APPLICATION CODE AND DEFINITION STATEMENTS 11
0.2 BENCHMARK SPONSOR 11
0.3 PARAMETER SETTINGS 11
0.4 CONFIGURATION DIAGRAMS 12

CLAUSE 1: LOGICAL DATABASE DESIGN **13**

1.1 TABLE DEFINITIONS 13
1.2 PHYSICAL ORGANIZATION OF DATABASE 13
1.3 INSERT AND DELETE OPERATIONS 13
1.4 HORIZONTAL OR VERTICAL PARTITIONING 13
1.5 REPLICATION OR DUPLICATION 13

CLAUSE 2: TRANSACTION AND TERMINAL PROFILES **14**

2.1 RANDOM NUMBER GENERATION 14
2.2 INPUT/OUTPUT SCREENS 14
2.3 PRICED TERMINAL FEATURE 14
2.4 PRESENTATION MANAGERS 14
2.5 TRANSACTION STATISTICS 15
2.6 QUEUING MECHANISM 15

CLAUSE 3: TRANSACTION AND SYSTEM PROPERTIES **16**

3.1 ATOMICITY 16
3.1.1 Atomicity of Completed Transactions *16*
3.1.2 Atomicity of Aborted Transactions *16*
3.2 CONSISTENCY 16
3.3 ISOLATION 17
3.4 DURABILITY 21
3.4.1 Durable Media Failure *21*
3.4.2 Instantaneous Interruption, Loss of Memory *22*

CLAUSE 4: SCALING AND DATABASE POPULATION **24**

4.1 CARDINALITY OF TABLES 24
4.2 DATABASE IMPLEMENTATION 24
4.3 DISTRIBUTION OF DATABASE FILES 24
4.4 60-DAY SPACE 26

CLAUSE 5: PERFORMANCE METRICS **27**

5.1 TPC BENCHMARK C METRICS 27
5.2 RESPONSE TIMES 27
5.3 KEYING AND THINK TIMES 27

5.4 Distribution and Performance Curves 27
5.4.1 Response Time frequency distribution curves 27
5.4.2 Response Time versus throughput 31
5.4.3 Think Time frequency distribution 32
5.4.4 Throughput versus elapsed time 33
5.5 Steady State Determination 34
5.6 Work Performed During Steady State 34
5.7 Measurement Period Duration 34
5.8 Transaction Statistics 34
5.9 Checkpoints 34
CLAUSE 6: SUT, DRIVER AND COMMUNICATION 36
6.1 Remote Terminal Emulator (RTE) 36
6.2 Emulated Components 36
6.3 Functional Diagrams 36
6.4 Networks 36
6.5 Operator Intervention 36
CLAUSE 7: PRICING 37
7.1 Hardware and Software Pricing 37
7.2 Three Year Price 37
7.3 Availability Dates 37
CLAUSE 8: REPORTING 38
8.1 Full Disclosure Report 38
CLAUSE 9: AUDITOR ATTESTATION 39
9.1 Auditor Information 39
9.2 Attestation Letter 39
APPENDIX A: SOURCE CODE 42
APPENDIX B: TUNABLE PARAMETERS 44
APPENDIX C: PRICE QUOTATIONS 46

Abstract

This report documents the methodology and results of the TPC Benchmark™ C (TPC-C) test conducted by TTA on the Goldilocks v3.1 Standard Edition on LTechKorea L224S-D

Goldilocks v3.1 Standard Edition on LTechKorea L224S-D

Company Name	System Name	Database Software	Operating System
Telecommunications Technology Association	LTechKorea L224S-D	Goldilocks v3.1 Standard Edition	RedHat Enterprise Linux 7.9

TPC Benchmark™ C Metrics

Total System Cost	TPC-C Throughput	Price/Performance	Availability Date
₩ 276,751,800 (KRW)	144,714 tpmC	1,913 KRW/tpmC	Available Now

Preface

The Transaction Processing Performance Council (TPC™) is a non-profit corporation founded to define transaction processing and database benchmarks and to disseminate objective, verifiable TPC performance data to the industry. The TPC Benchmark© C is an on-line transaction processing benchmark (OLTP) developed by the TPC.

TPC Benchmark™ C Overview

TPC Benchmark™ C (TPC-C) simulates a complete computing environment where a population of users executes transactions against a database. The benchmark is centered around the principal activities (transactions) of an order-entry environment. These transactions include entering and delivering orders, recording payments, checking the status of orders, and monitoring the level of stock at the warehouses. While the benchmark portrays the activity of a wholesale supplier, TPC-C is not limited to the activity of any particular business segment, but, rather represents any industry that must manage, sell, or distribute a product or service.

TPC-C consists of a mixture of read-only and update intensive transactions that simulate the activities found in complex OLTP application environments. It does so by exercising a breadth of system components associated with such environments, which are characterized by:

- *The simultaneous execution of multiple transaction types that span a breadth of complexity*
- *On-line and deferred transaction execution modes*
- *Multiple on-line terminal sessions*
- *Moderate system and application execution time*
- *Significant disk input/output*
- *Transaction integrity (ACID properties)*
- *Non-uniform distribution of data access through primary and secondary keys*
- *Databases consisting of many tables with a wide variety of sizes, attributes, and relationships*
- *Contention of data access and update*

The performance metric reported by TPC-C is a "business throughput" measuring the number of orders processed per minute. Multiple transactions are used to simulate the business activity of processing an order, and each transaction is subject to a response time constraint. The performance metric for this benchmark is expressed in transactions-per-minute-C (tpmC). To be compliant with the TPC-C standard, all references to tpmC results must include the tpmC rate, the associated price-per-tpmC, and the availability date of the priced configuration.

TPC-C uses terminology and metrics that are similar to other benchmarks, originated by the TPC or others. Such similarity in terminology does not in any way imply that TPC-C results are comparable to other benchmarks. The only benchmark results comparable to TPC-C are other TPC-C results conformant with the same revision.

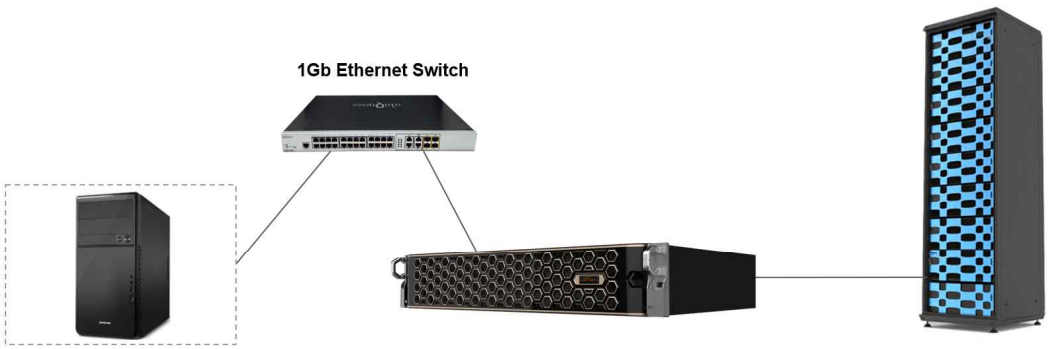
Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to other environments 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-C should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

Further information is available at www.tpc.org

	Goldilocks v3.1 Standard Edition on LTechkorea L224S-D			**TPC-C Version 5.11.0** **TPC Pricing 2.7.0** Report Date **30-Aug-2021**
Total System Cost	TPC-C Throughput	Price/Performance		Availability Date
₩ 276,751,800 (KRW)	**144,714 tpmC**	**1,913 KRW/tpmC**		**Available Now**
Server Processors/Cores/Threads	Database Manager	Operating System	Other Software	Number of Users
2/56/112	**Goldilocks v3.1 Standard Edition**	**RHEL 7.9**	**JBoss Web Server**	**114,000**

Priced Configuration (LTechKorea)



Web Application Server

3 x HANSUNG DT-S170G1IV0
- 1 x Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz
- 1 x 32GB Memory
- 1 x 2TB SATA HDD
- 1 x 512GB SATA SSD
- 1 x 2-port 10G Ethernet
- 1 x 1G Ethernet

Database Server

1 x LTechKorea L224S-D
- 2 x Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz
- 12 x 128GB (1,536GB) Memory
- 2 x 1TB SAS SSD (RAID-1, Write Through)
- 1 x 16Gb 2-Port Host Bus Adaptor
- 1 x 2-port 10G Ethernet
- 1 x 2-port 1G Ethernet

Storage

1 x UNIWIDE FCH2800
- 16 x 32GB (512GB) Cache Memory
- 8 x 1.6TB FMD Drive (Write Back)
- 4 x 8/16Gb 8-Port Host Bus Adaptor

System Components	DB Server		WAS Server	
	Quantity	**Description**	**Quantity**	**Description**
Processors/Cores/Threads	2/56/112	Intel(R) Xeon(R) Gold 6258R CPU @ 2.70GHz	1/8/16	Intel(R) Core(TM) i7-10700K CPU @ 3.80GHz
Memory	12	128GB	1	32GB
Storage Controller	1 2 2	- MegaRAID SAS-3 3108 - C620 Series SATA Controller - QLogic Corp. ISP8324-based 16Gb Fibre Channel to PCI Express Adapter		
Storage Device	2 8	1TB SAS HDD (Write Through) 1.6TB FMD SSD (External, Write back)	1 1	1TB SATA HDD 512GB SATA SSD
Total Storage Capacity		14.8TB		

	Goldilocks v3.1 Standard Edition on LTechkorea L224S-D	TPC-C Version 5.11.0 TPC Pricing 2.7.0 Report Date 30-Aug-2021 Available Now

Description	Part Number	Source	Unit Price	Qty	Price	3-Yr. Maint. Price
Server Hardware						
DB Server – L224S-D Server	L224S-D	1	207,494,400	1	**207,494,400**	
Intel® Xeon® Gold 6258R 28C/56Th 2.7Ghz (Up To 4Ghz)	-	1	(included)	2		
128GB fully buffered DDR4 Memory, LRDIMM	-	1	(included)	12		
SSD 1TB, Enterpeise	-	1	(included)	2		
RAID Controller - LSI 9361 8i	-	1	(included)	1		
Intel® Ethernet Server Adapter I350-T2V2	-	1	(included)	1		
Intel® Ethernet Converged Network Adapter X550-T2	-	1	(included)	1		
QLE 2692 PCI-e 3.0 - 16G Fc gbic	-	1	(included)	1		
224S-D Server(power 920watt) 2.5 inch disk bay * 24EA	-	1	(included)	1		
Maintenance - 7x24x4 Care Pack (3-yrs)	-	1	(included)	1		
WAS Servers (per server) - DT-S170G1IV0	DT-S170G1IV0	2	1,551,000	5[1]	**7,755,000**	
Intel® Core Processor i7-10700K, 3.8GHz		2	(included)	1		
DDR4 32GB, Samsung, UDIMM PC4-25600U		2	(included)	1		
Seagate Barracuda ST2000DM008 2 TB 3.5"		2	(included)	1		
HP SSD EX900 500GB		2	(included)	1		
Server Hardware Sub Total					**215,249,400**	
Storage Hardware						
All Flash Storage - FCH2800	FCH2800	3	72,250,000	1	**72,250,000**	
FCH2800 Controller Device	T0001-0117-00	3	(included)	1		
Back-end Bus Adapter 12G SAS	T0001-0117-01	3	(included)	1		
16G 8-Port Host Bus Adapter	T0001-0117-02	3	(included)	4		
Cache Interconnect Adapter	T0001-0117-03	3	(included)	1		
Cache Memory DDR-3 (32GB)	T0001-0117-04	3	(included)	16		
FCH2800 Flash Disk Drive Expantion Unit	T0001-0117-05	3	(included)	1		
FCH2800 controller cpu Board	T0001-0117-06	3	(included)	1		
Rack 600x1200x2010 mm (WxDxH) 42U	T0001-0117-07	3	(included)	1		
Storage Management SW	T0001-0117-08	3	(included)	1		
UTP CAT5e Ethernet Cable 1M	61001-0001-00	3	(included)	1		
Power Cord, NICETECH, 2.5M	42119-0005-00	3	(included)	2		
1.6TB Flash Memory Disk Drive	T22601-0117-03	3	3,900,000	8	31,200,000	
3-yrs 24x7x4hrs Onsite Support Service	-	3	26,350,000	1		26,350,000
Storage Hardware Sub Total					**103,450,000**	**26,350,000**

[1] The number of WAS Servers in SUT is 3, But priced condition includes 2 spares for replaceable items. (**TPC Pricing 2.7.0 - 4.3.2**)

Client/Server Software						
Red Hat Enterprise Linux Server Standard 3yrs	RH00004	4	4,098,000	4	16,392,000	
RHEL Server Standard Maintenance - 3yrs 24x7x4hrs	RP-CSP(OS)	4	5,000,000	4		20,000,000
Red Hat JBoss Web Server 4-Core Standard 3Year	MW00123F3	4	2,138,000	6	12,828,000	
JBoss Web Server per 4Core 3Year Maintenance	RP-CSP(WAS)	4	12,000,000	6		72,000,000
Goldilocks v3.1 Standard Edition	-	5	96,000,000	1	96,000,000	
Goldilocks v3.1 Standard Edition Technical Supports	-	5	10,000,000	3		30,000,000
Software Sub Total					**125,220,000**	**122,000,000**
Other Hardware						
UbiQuoss uSafe3010-24ps (10G, 24-port)(w/spares)	22917889	6	1,900,000	3	5,700,000	
Other Hardware Sub Total					**5,700,000**	
Discounts*						
HW Discount - DB Server(L224S-D)					-181,557,600	
Red Hat OS Discount					-6,792,000	-12,000,000
Red Hat JBoss Discount					-5,268,000	-36,000,000
SW Discount - Goldilocks					-64,000,000	-15,600,000
Discounts Sub Total					**-257,617,600**	**-63,600,000**
Total					**192,001,800**	**84,750,000**

Pricing Notes

1) LTechKorea Inc.
2) Hansung Corporation.
3) UNIWIDE Technologies Inc.
4) Rockplace Inc.
5) Sunjesoft Inc.
6) UbiQuoss Inc.

All of the prices are based on South Korea's currency, KRW (₩, Korean Won) and excluded VAT.

* All discounts are based on Korea list prices and for similar quantities and configurations. Discounts for similarly sized configurations will be similar to those quoted here, but may vary based on the components in the configuration.

Three year cost of ownership KRW(₩): 276,751,800

TPC-C throughput: 144,714 tpmC

Price/Performance: 1,913 ₩ / tpmC

Benchmark implementation and results independantly audited by Doug Johnson of InfoSizing (www.sizing.com)

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 pricing specifications. If you find that the stated prices are not available according to these terms, please inform the TPC at pricing@tpc.org. Thank you.

	Goldilocks v3.1 Standard Edition on LTechkorea L224S-D	TPC-C Version 5.11.0 TPC Pricing 2.7.0 Report Date 30-Aug-2021 Available Now

MQTh, computed Maximum Qualified Throughput	144,714 tpmC	

Response Times (seconds)	Min	Average	90th	Max
New-Order	0.102	0.104	0.104	0.912
Payment	0.102	0.103	0.103	0.715
Order-Status	0.102	0.103	0.102	0.712
Delivery (interactive portion)	0.101	0.101	0.101	0.397
Delivery (deferred portion)	0.002	0.006	0.006	0.674
Stock-Level	0.102	0.103	0.103	0.706
Menu	0.101	0.101	0.102	0.507

Emulated Display Delay: 0.1 sec.

Transaction Mix	Percent	Number
New-Order	44.979%	60,782,381
Payment	43.012%	58,123,521
Order-Status	4.003%	5,408,747
Delivery	4.003%	5,409,721
Stock-Level	4.003%	5,409,646

Keying Times (seconds)	Min	Average	Max
New-Order	18.001	18.001	18.004
Payment	3.001	3.001	3.006
Order-Status	2.001	2.001	2.002
Delivery	2.001	2.001	2.003
Stock-Level	2.001	2.001	2.002

Think Times (seconds)	Min	Average	Max
New-Order	0.001	12.044	120.501
Payment	0.001	12.045	120.501
Order-Status	0.001	10.052	100.501
Delivery	0.001	5.028	50.301
Stock-Level	0.001	5.030	50.301

Test Duration	
Ramp-up time	65 min
Measurement Interval (MI)	420 min
Checkpoints in MI	14
Checkpoint Interval (Average / Max)	29:22 min / 29:23 min
Number of Transactions in MI (all types)	135,134,016

General Items

0.1 Application Code and Definition Statements

The application program (as defined in clause 2.1.7) must be disclosed. This includes, but is not limited to, the code implementing the five transactions and the terminal input output functions.

Appendix A contains the application source code for the transactions.

0.2 Benchmark Sponsor

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

This benchmark was sponsored by TTA, Telecommunications Technology Association. The implementation was developed and engineered in partnership with SUNJESOFT Inc. and LTechKorea Inc.

0.3 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 by not limited to:

- *Database options*
- *Recover/commit options*
- *Consistency locking options*
- *Operating system and application configuration parameters*

This requirement can be satisfied by providing a full list of all parameters.

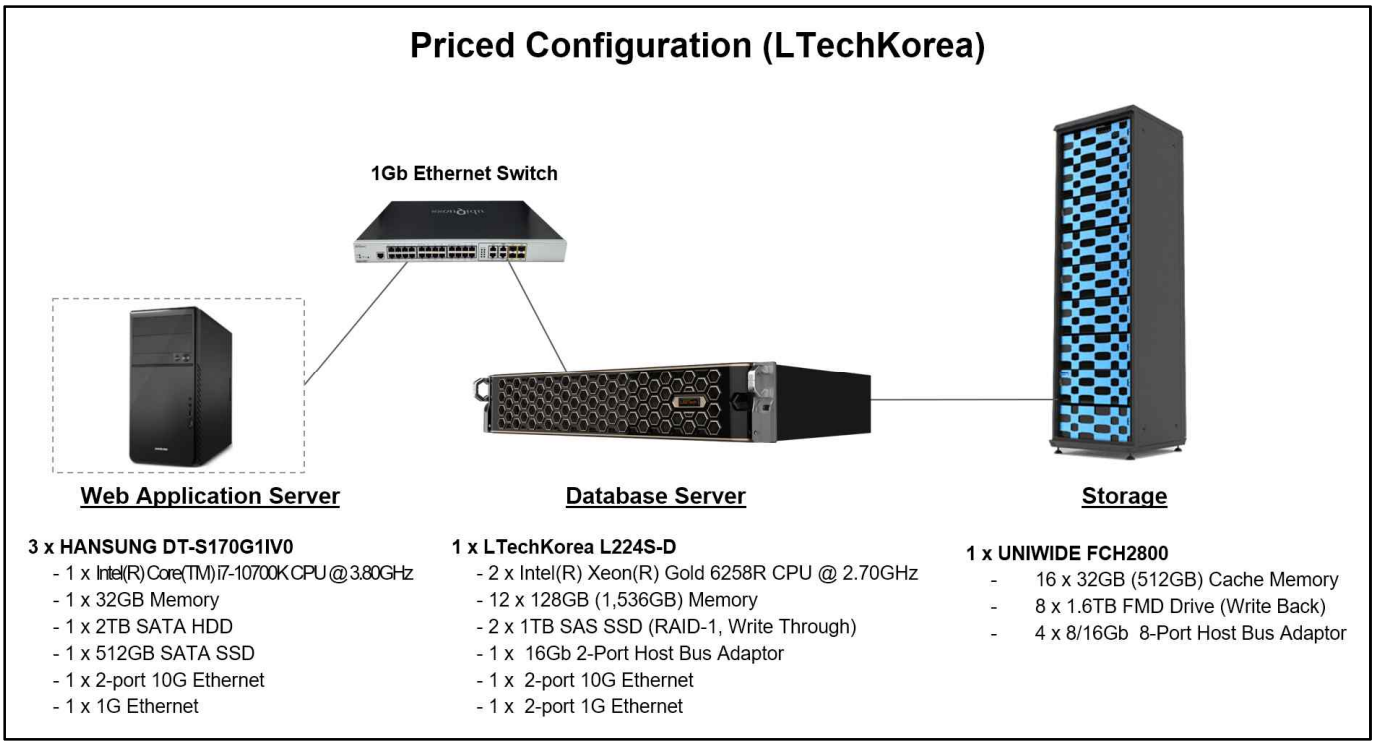
Appendix B contains the tunable parameters for the database, the operating system, and the transaction monitor.

0.4 Configuration Diagrams

Diagrams of both measured and priced configurations must be provided, accompanied by a description of the differences.

The configuration diagram for both the tested and priced system is depicted in Figure 0.1. There was no difference between the priced and tested configurations.

Figure 0.1: Benchmarked and Priced Configuration



Clause 1: Logical Database Design

1.1 Table Definitions

Listing must be provided for all table definition statements and all other statements used to set up the database.

Appendix A contains the code used to define and load the database tables.

1.2 Physical Organization of Database

The physical organization of tables and indices within the database must be disclosed.

The physical organization of the database is shown in Table 1.2.

Table 1.2: Physical Organization of the Database

Controller	Array	RAID Array	Drives	Content
MegaRAID SAS-3 3108	Internal	RAID 1	2 x SATA 1TB HDD	OS
Hitachi DKC810I Series	FCH2800 Array	RAID 1 (2D+2D)	4 x 1.6TB FMD	Database files
Hitachi DKC810I Series	FCH2800 Array	RAID 1 (2D+2D)	4 x 1.6TB FMD	Redo Logs

1.3 Insert and Delete Operations

It must be ascertained that insert and/or delete operations to any of the tables can occur concurrently with the TPC-C transaction mix. Furthermore, any restrictions in the SUT database implementation that precludes inserts beyond the limits defined in Clause 1.4.11 must be disclosed. This includes the maximum number of rows that can be inserted and the minimum key value for these new rows.

All insert and delete functions were verified to be fully operational during the entire benchmark.

1.4 Horizontal or Vertical Partitioning

While there are a few restrictions placed upon horizontal or vertical partitioning of tables and rows in the TPC-C benchmark, any such partitioning must be disclosed.

No horizontal or vertical partitioning was used in this benchmark.

1.5 Replication or Duplication

Replication of tables, if used, must be disclosed. Additional and/or duplicated attributes in any table must be disclosed along with a statement on the impact on performance.

No replications, duplications or additional attributes were used in this benchmark.

Clause 2: Transaction and Terminal Profiles

2.1 Random Number Generation

The method of verification for the random number generation must be described.

Random numbers were generated using ‘SysVr4 rand_r()’ call. The seed value for ‘rand_r()’ was collected and reviewed by the auditor.

2.2 Input/Output Screens

The actual layout of the terminal input/output screens must be disclosed.

All screen layouts were verified by the auditor to validate that they followed the requirements of the specifications.

2.3 Priced Terminal Feature

The method used to verify that the emulated terminals provide all the features described in Clause 2.2.2.4 must be explained. Although not specifically priced, the type and model of the terminals used for the demonstration in 8.1.3.3 must be disclosed and commercially available (including supporting software and maintenance).

The terminal attributes were manually verified by the auditor by verifying that each required feature was implemented.

2.4 Presentation Managers

Any usage of presentation managers or intelligent terminals must be explained.

Application code running on the client systems implemented the TPC-C user interface. No presentation manager software or intelligent terminal features were used. The source code for the user interface is listed in Appendix A.

2.5 Transaction Statistics

Table 2.1 lists the transaction statistics defined in Clauses 8.1.3.5 to 8.1.3.11 and observed during the Measurement Interval.

Table 2.1: Transaction Statistics

Statistic		Value
New Order	Home warehouse order lines Remote warehouse order lines Rolled back transactions Average items per order	99.001% 0.999% 0.999% 10.000
Payment	Home warehouse Remote warehouse Accessed by last name	85.006% 14.994% 60.001%
Order Status	Accessed by last name	60.008%
Delivery	Skipped transactions	0
Transaction Mix	New Order Payment Order status Delivery Stock level	44.979% 43.012% 4.003% 4.003% 4.003%

2.6 Queuing Mechanism

The queuing mechanism used to defer the execution of the Delivery transaction must be disclosed.

The queuing mechanism was implemented using ‘BlockingQueue’ provided by Java.

Clause 3: Transaction and System Properties

The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. This includes disclosing which case was followed for the execution of Isolation Test 7.

All ACID property tests were conducted according to the specification.

3.1 Atomicity

The system under test must guarantee that the 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.

3.1.1 Atomicity of Completed Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have been changed appropriately.

A row was randomly selected from the CUSTOMER, DISTRICT, and WAREHOUSE tables, and the balances noted. A payment transaction was started with the same Customer, District, and Warehouse identifiers and a known amount. The payment transaction was committed and the rows were verified to contain correctly updated balances.

3.1.2 Atomicity of Aborted Transactions

Perform the Payment transaction for a randomly selected warehouse, district, and customer (by customer number) and substitute a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the records in the CUSTOMER, DISTRICT, and WAREHOUSE tables have NOT been changed.

A row was randomly selected from the CUSTOMER, DISTRICT, and WAREHOUSE tables, and the balances noted. A payment transaction was started with the same Customer, District, and Warehouse identifiers and a known amount. The payment transaction was rolled back and the rows were verified to contain the original balances.

3.2 Consistency

Consistency is the property of the application that requires any execution of a data base transaction to take the database from one consistent state to another, assuming that the data base is initially in a consistent state.

Verify that the data base is initially consistent by verifying that it meets the consistency conditions defined in Clauses 3.3.2.1 to 3.3.2.4. Describe the steps used to do this in sufficient detail so that the steps are independently repeatable.

The specification defines 12 consistency conditions, of which Consistency conditions 1 through 4 were demonstrated as follows:

1. The sum of balances (d_ytd) for all Districts within a specific Warehouse is equal to the balance (w_ytd) of that Warehouse.
2. For each District within a Warehouse, the next available Order ID (d_next_o_id) minus one is equal to the most recent Order ID [max(o_id)] for the ORDER table associated with the preceding District and Warehouse. Additionally, that same relationship exists for the most recent Order ID [max(o_id)] for the NEW-ORDER table associated with the same District and Warehouse. Those relationships can be illustrated as:
   d_next_o_id – 1 = max(o_id) = max(no_o_id)
   where (d_w_id = o_w_id = no_w_id) and (d_id = o_d_id = no_d_id)

3. For each District within a Warehouse, the value of the most recent Order ID [max(no_o_id)] minus the first Order ID [min(no_o_id)] plus one, for the NEW-ORDER table associated with the District and Warehouse, equals the number of rows in that NEW-ORDER table.
   That relationship can be illustrated as:
   max(no_o_id) – min(no_o_id) + 1 = rows in NEW-ORDER
   where (o_w_id = no_w_id) and (o_d_id = no_d_id)
4. For each District within a Warehouse, the sum of Order-Line counts [sum(o_ol_cnt)] for the Orders associated with the District equals the number of rows in the ORDER-LINE table associated with the same District.
   That relationship can be illustrated as:
   sum(o_ol_cnt) = rows in the ORDER-LINE table for the Warehouse and District

To test consistency, the following steps were executed:

1. The consistency conditions 1 through 4 were tested by running queries against the database. All queries showed that the database was in a consistent state.
2. An RTE run was executed at full load for a duration sufficient to include at least one completed checkpoint.
3. The consistency conditions 1 through 4 were tested again. All queries showed that the database was still in a consistent state.

3.3 Isolation

Sufficient conditions must be enabled at either the system or application level to ensure the required isolation defined above (clause 3.4.1) is obtained.

The benchmark specification defines nine tests to demonstrate the property of transaction isolation. The tests, described in Clauses 3.4.2.1 – 3.4.2.9, were all successfully executed using a series of scripts. Each included timestamps to demonstrate the concurrency of operations. The results of the queries were logged. The captured logs were verified to demonstrate the required isolation had been met.

Isolation Test 1

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is committed.

The test proceeds as follows:

1. An Order-Status transaction T0 was executed and committed for a randomly selected Customer, and the Order returned was noted.
2. A New-Order transaction T1 was started for the same Customer used in T0. T1 was stopped prior to COMMIT.
3. An Order-Status transaction T2 was started for the same Customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same Order that T0 had returned.
4. T1 was allowed to complete and was committed.
5. An Order-Status transaction T3 was started for the same Customer used in T1. T3 returned the Order inserted by T1.

Isolation Test 2

This test demonstrates isolation for read-write conflicts of Order-Status and New-Order transactions when the New-Order transaction is rolled back.

The test proceeds as follows:

1. An Order-Status transaction T0 was executed and committed for a randomly selected Customer and the Order returned was noted.
2. A New-Order transaction T1 with an invalid item number was started for the same Customer used in T0. T1 was stopped immediately prior to ROLLBACK.
3. An Order-Status transaction T2 was started for the same Customer used in T1. T2 completed and was committed without being blocked by T1. T2 returned the same Order that T0 had returned.
4. T1 was allowed to ROLLBACK.
5. An Order-Status transaction T3 was started for the same Customer used in T1. T3 returned the same Order that T0 had returned.

Isolation Test 3

This test demonstrates isolation for write-write conflicts of two New-Order transactions when both transactions are committed.

The test proceeds as follows:

1. The D_NEXT_O_ID of a randomly selected district was retrieved.
2. A New-Order transaction T1 was started for a randomly selected customer within the District used in step 1. T1 was stopped immediately prior to COMMIT.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to complete. T2 completed and was committed.
5. The order number returned by T1 was the same as the D_NEXT_O_ID retrieved in step 1. The order number returned by T2 was one greater than the order number returned by T1.
6. The D_NEXT_O_ID of the same District was retrieved again. It had been incremented by two (i.e. it was one greater than the order number returned by T2).

Isolation Test 4

This test demonstrates isolation for write-write conflicts of two New-Order transactions when one transaction is rolled back.

The test proceeds as follows:

1. The D_NEXT_O_ID of a randomly selected District was retrieved.
2. A New-Order transaction T1, with an invalid item number, was started for a randomly selected customer within the district used in step 1. T1 was stopped immediately prior to ROLLBACK.
3. Another New-Order transaction T2 was started for the same customer used in T1. T2 waited.
4. T1 was allowed to roll back, and T2 completed and was committed.
5. The order number returned by T2 was the same as the D_NEXT_O_ID retrieved in step 1.

6. The D_NEXT_O_ID of the same District was retrieved again. It had been incremented by one (i.e. one greater than the order number returned by T2).

Isolation Test 5

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when Delivery transaction is committed.

The test proceeds as follows:

1. A query was executed to find out the Customer who is to be updated by the next Delivery transaction for a randomly selected Warehouse and District.
2. The C_BALANCE of the Customer found in step 1 was retrieved.
3. A Delivery transaction T1 was started for the same Warehouse used in step 1. T1 was stopped immediately prior to COMMIT.
4. A Payment transaction T2 was started for the same Customer found in step 1. T2 waited.
5. T1 was allowed to complete. T2 completed and was committed.
6. The C_BALANCE of the Customer found in step 1 was retrieved again. The C_BALANCE reflected the results of both T1 and T2.

Isolation Test 6

This test demonstrates isolation for write-write conflicts of Payment and Delivery transactions when the Delivery transaction is rolled back.

The test proceeds as follows:

1. A query was executed to find out the Customer who is to be updated by the next delivery transaction for a randomly selected Warehouse and District.
2. The C_BALANCE of the Customer found in step 1 was retrieved.
3. A Delivery transaction T1 was started for the same Warehouse used in step 1. T1 was stopped immediately prior to COMMIT.
4. A Payment transaction T2 was started for the same customer found in step 1. T2 waited.
5. T1 was forced to execute a ROLLBACK. T2 completed and was committed. The C_BALANCE of the Customer found in step 1 was retrieved again. The C_BALANCE reflected the results of only T2.

Isolation Test 7

This test demonstrates repeatable reads for the New-Order transaction while an interactive transaction updates the prices of some items.

The test proceeds as follows:

1. The I_PRICE of two randomly selected items X and Y were retrieved.
2. A New-Order transaction T1 with a group of Items including Items X and Y was started. T1 was stopped immediately after retrieving the prices of all items. The prices of Items X and Y retrieved matched those retrieved in step 1.

3. A transaction T2 was started to increase the price of Items X and Y by 10%.
4. T2 did not stall and was committed.
5. T1 was resumed, and the prices of all Items were retrieved again within T1. The prices of Items X and Y matched those retrieved in step 1.
6. T1 was committed.
7. The prices of Items X and Y were retrieved again. The values matched the values set by T2.

The Execution followed Case D, where T3 does not stall and no transaction is rolled back. Query T4 verifies the price change made by T3.

Isolation Test 8

This test demonstrates isolation for phantom protection between New-Order and Delivery transactions.

The test proceeds as follows:

1. The NO_D_ID of all NEW_ORDER rows for a randomly selected Warehouse and District was changed to 11. The changes were committed.
2. A Delivery transaction T1 was started for the selected Warehouse.
3. T1 was stopped immediately after reading the NEW_ORDER table for the selected Warehouse and District. No qualifying row was found.
4. A New-Order transaction T2 was started for the same Warehouse and District. T2 completed and was committed without being blocked by T1.
5. T1 was resumed and the NEW_ORDER table was read again. No qualifying row was found.
6. T1 completed and was committed.
7. The NO_D_ID of all NEW_ORDER rows for the selected Warehouse and District was restored to the original value. The changes were committed.

Isolation Test 9

This test demonstrates isolation for phantom protection between New-Order and Order-Status transactions.

The test proceeds as follows:

1. An Order-Status transaction T1 was started for a randomly selected Customer.
2. T1 was stopped immediately after reading the ORDER table for the selected Customer to find the most recent Order for that Customer.
3. A New-Order transaction T2 was started for the same Customer. T2 completed and was committed without being blocked by T1.
4. T1 was resumed and the ORDER table was read again to determine the most recent Order for the same Customer. The Order found was the same as the one found in step 2.
5. T1 completed and was committed.

3.4 Durability

The tested system must guarantee durability: the ability to preserve the effects of committed transactions and ensure data base consistency after recovery from any one of the failures listed in Clause 3.5.3

- *Permanent irrecoverable failure of any single durable medium containing TPC-C database tables or recovery log data (this test includes failure of all or part of memory)*
- *Instantaneous interruption (system crash/system hang) in processing that requires system reboot to recover*
- *Failure of all or part of memory (loss of contents)*

3.4.1 Durable Media Failure

3.4.1.1 Loss of Log Media and Data Media

This test was conducted on a fully scaled database. To demonstrate recovery from a permanent failure of durable medium containing TPC-C Log Media and Data Media, the following steps were executed:

1. The total number of Orders is determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table; giving count-1.
2. The consistency is verified.
3. The RTE is started with full user load.
4. The test is allowed to run for a minimum of 5 minutes after ramp-up.
5. A first checkpoint is initiated and completed.
6. The test is allowed to run for a minimum of 2 more minutes.
7. A second checkpoint is initiated.
8. Before the second checkpoint completes, one data disk is disabled by removing it physically. Since the data disks are configured with redundancy, the transactions continued to run without interruption.
9. The test is allowed to run until the completion of the second checkpoint and for at least 5 minutes
10. A third checkpoint is initiated.
11. Before the third checkpoint completes, one log device is disabled by removing it physically. Since the log devices are configured with redundancy, the transactions continued to run without interruption.
12. The test is allowed to run until the third checkpoint has completed, but no less than 5 more minutes.
13. The RTE run is completed.
14. The consistency is verified.
15. Step 1 is repeated, giving count-2.
16. The RTE result file is used to determine the number of New-Order transactions successfully completed during the full run.
17. The difference between the count-1 and count-2 is compared with the number of New-Order transactions successfully completed during the full run. The difference indicated that no committed transactions had been lost.
18. Data from the success file is used to query the database to demonstrate that the last 500 successful New-Orders have corresponding rows in the ORDER table.

3.4.1.2 Instantaneous Loss of Storage Controller Cache

This test was executed on a fully scaled database. The following steps were executed: To demonstrate recovery from a permanent failure of a controller cache, the following steps were executed:

1. The total number of Orders is determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table; giving count-1.
2. The consistency is verified.
3. The RTE is started with full user load.
4. The test is allowed to run for a minimum of 5 minutes at full load (after ramp-up)
5. A first checkpoint is initiated and completed.
6. The test is allowed to run for a minimum of 2 more minutes.
7. A second checkpoint is initiated.
8. Before the second checkpoint completes, one of the two caches in the storage subsystem was failed (removing it from the chassis)
9. The RTE run is completed.
10. Step 1 is repeated, giving count-2.
11. The consistency is verified.
12. The RTE result file is used to determine the number of New-Order transactions successfully completed during the full run.
13. The difference between the count-1 and count-2 is compared with the number of New-Order transactions successfully completed during the full run. The difference indicated that all committed transactions had been successfully recovered.
14. Data from the success file is used to query the database to demonstrate that the last 500 successful New-Orders have corresponding rows in the ORDER table.

3.4.2 Instantaneous Interruption, Loss of Memory

As the loss of power erases the contents of memory, the instantaneous interruption and the loss of memory tests were combined into a single test. This test was executed on a fully scaled database. The following steps were executed:

1. The total number of Orders is determined by the sum of D_NEXT_O_ID of all rows in the DISTRICT table; giving count-1.
2. The consistency is verified.
3. The RTE is started with full user load.
4. The test is allowed to run for a minimum of 5 minutes at full load (after ramp-up).
5. A first checkpoint is initiated and completed.
6. The test is allowed to run for a minimum of 2 more minutes.
7. A second checkpoint is initiated.
8. Before the second checkpoint completes, the primary power to the back-end server is shut off (removing both power cords).

9. The RTE is shutdown.
10. Power is restored to the database server and the system performs an automatic recovery.
11. GOLDILOCKS is restarted and performs an automatic recovery.
12. Step 1 is repeated, giving count-2.
13. The consistency is verified.
14. The RTE result file is used to determine the number of New-Order transactions successfully completed during the full run.
15. The difference between the count-1 and count-2 is compared with the number of New-Order transactions successfully completed during the full run. The difference indicated that all committed transactions had been successfully recovered.
16. Data from the success file is used to query the database to demonstrate that the last 500 successful New-Orders have corresponding rows in the ORDER table.

Clause 4: Scaling and Database Population

4.1 Cardinality of Tables

The cardinality (e.g. number of rows) of each table, as it existed at the start of the benchmark run, must be disclosed. If the database was over-scaled and inactive rows of the WAREHOUSE table were deleted, the cardinality of the WAREHOUSE table as initially configured and the number of rows

Table 4.1 shows that number of rows for each table as they were initially populated.

Table 4.1: Number of Rows for Server

Table	Cardinality
Warehouse	11,400
District	114,000
Customer	342,000,000
History	342,000,000
Order	102,600,000
New Order	3,419,370,542
Order Line	100,000
Stock	1,140,000,000
Item	342,000,000
Unused Warehouses	0

4.2 Database Implementation

A statement must be provided that describes: The data model implemented by DBMS used (e.g. relational, network, hierarchical). The database interfaces (e.g. embedded, call level) and access language (e.g. SQL, DL/1, COBOL read/write used to implement the TPC-C transaction. If more than one interface/access language is used to implement TPC-C, each interface/access language must be described and a list of which interface/access language is used with which transaction type must be disclosed.

Goldilocks v3.1 is an in-memory DBMS, implementing the relational model.

The transactions are implemented in SQL via JDBC calls to the database engine.

All application code and procedures are listed in Appendix A.

4.3 Distribution of Database Files

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

The database files are stored on a set of four 1.6TB disks configured as RAID1(2+2). The database log files are stored on four 1.6TB disks configured as RAID1(2+2).

Table 4.3: Database file locations

Name	Location	Description
system_XXX.dbf	/data/db/db1	System tables and dictionary
tpcc_data_XX.dbf	/data/db/db1 /data/db/db2 /data/db/db3 /data/db/db4 /data/db/db5	Database data files
redo_X_X.log	/wal	Database log files

The distribution of tables and logs across storage media is shown in Table 1.2.

4.4 60-Day Space

Details of the 60-day space computations along with proof that the database is configured to sustain 8 hours of growth for the dynamic tables (Order, Order-Line, and History) must be disclosed.

A test run of over 8 hours was executed to demonstrate that the configuration is capable of sustaining 8 hours of growth at the reported throughput. The computation of the 60-day storage requirements is shown in Table 4.4.

Table 4.4: 60-Day Space Calculations

Base Unit (KBytes)	1
tpmC	144,714.210

Table	Rows	Data	Index	Initial Population	5% Growth	8-Hour Growth	Required Runtime Space
WAREHOUSE	11,400	91,912	312	92,224	4,611	0	96,835
DISTRICT	114,000	14,472	4,280	18,752	938	0	19,690
CUSTOMER	342,000,000	220,090,728	28,064,992	248,155,720	12,407,786	0	260,563,506
NEW_ORDER	102,600,000	6,486,640	3,623,976	10,110,616	505,531	0	10,616,147
ITEM	100,000	10,816	2,736	13,552	678	0	14,230
STOCK	1,140,000,000	420,057,192	37,304,520	457,361,712	22,868,086	0	480,229,798
HISTORY	342,000,000	28,253,096	0	28,253,096	0	5,738,420	33,991,516
ORDERS	342,000,000	21,793,512	25,861,368	47,654,880	0	4,426,429	52,081,309
ORDER_LINE	3,419,370,542	321,586,048	133,872,832	455,458,880	0	65,316,591	520,775,471
Total		1,018,384,416	228,735,016	1,247,119,432	35,787,629	75,481,440	1,358,388,501

60-Day Requirements	
Dynamic-Space	371,632,656
Free-Space	1,266,568
Static-Space	875,486,776
Daily-Growth	75,481,440
Daily-Spread	0
60-Day Space	**5,404,373,195**

Memory Requirements	
Final Allocation	1,381,972,016
Non-Growing 5%	35,787,629
1-Day Memory	**1,417,759,645**

Storage Requirements	
Total Disk Space	6,549,825,127
Log space used	209,715,200
60-Day Space	5,404,373,195
Remaining Space	**935,736,732**

Clause 5: Performance Metrics

5.1 TPC Benchmark C Metrics

The TPC-C Metrics are reported in the front of this report as part of the executive summary.

5.2 Response Times

Ninetieth percentile, maximum and average response times must be reported for all transaction types as well as for the menu response time.

During the performance run transactions are submitted by the RTE in accordance with the required mix, Keying Times and Think Times of the benchmark Specification. Transactions are submitted by emulated users via HTTP. All timings are recorded by the RTE. The response time is measured from the submission of the transaction until the last byte of response is received by the RTE.

The details of the response times are reported in the front of this report as part of the Executive Summary.

5.3 Keying and Think Times

The minimum, the average, and the maximum keying and think times must be reported for each transaction type.

The details of the keying and think times are reported in the front of this report as part of the Executive Summary.

5.4 Distribution and Performance Curves

5.4.1 Response Time frequency distribution curves

Response Time frequency distribution curves must be reported for each transaction type.

Figure 5.4.1.1 shows the Response Time frequency distribution curves for the New-Order transaction.

Figure 5.4.1.2 shows the Response Time frequency distribution curves for the Payment transaction.

Figure 5.4.1.3 shows the Response Time frequency distribution curves for the Order-Status transaction.

Figure 5.4.1.4 shows the Response Time frequency distribution curves for the interactive portion of the Delivery transaction.

Figure 5.4.1.5 shows the Response Time frequency distribution curves for the Stock-Level transaction.

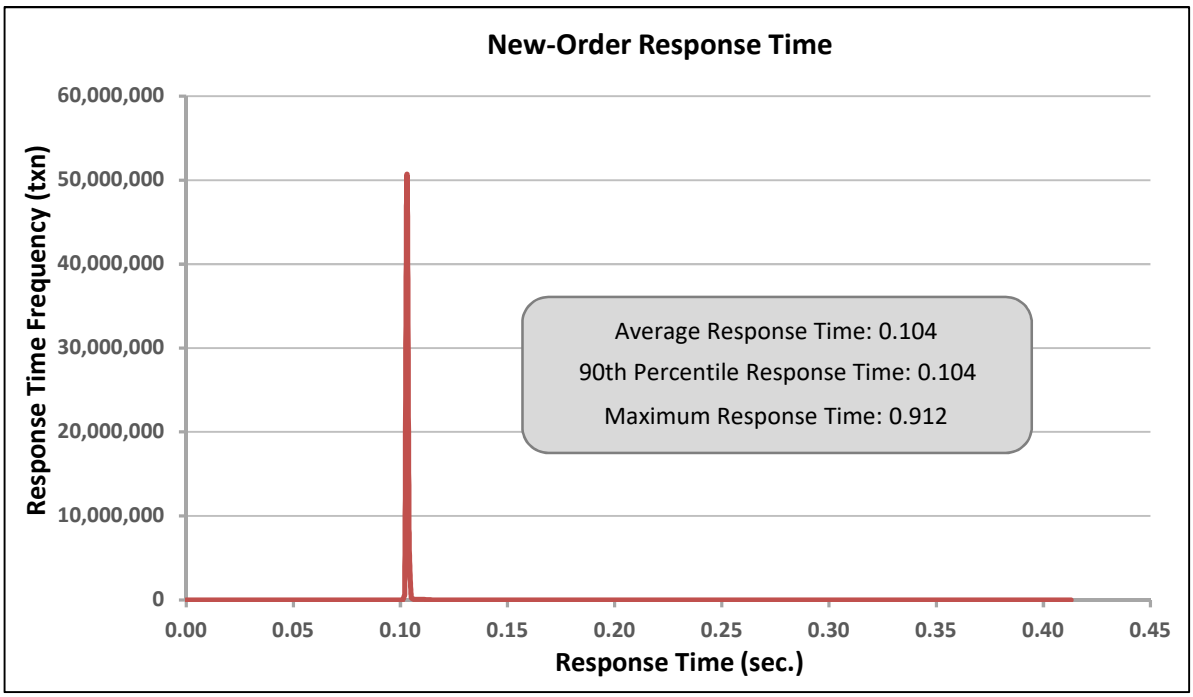


Figure 5.4.1.1: New-Order RT Frequency Distribution

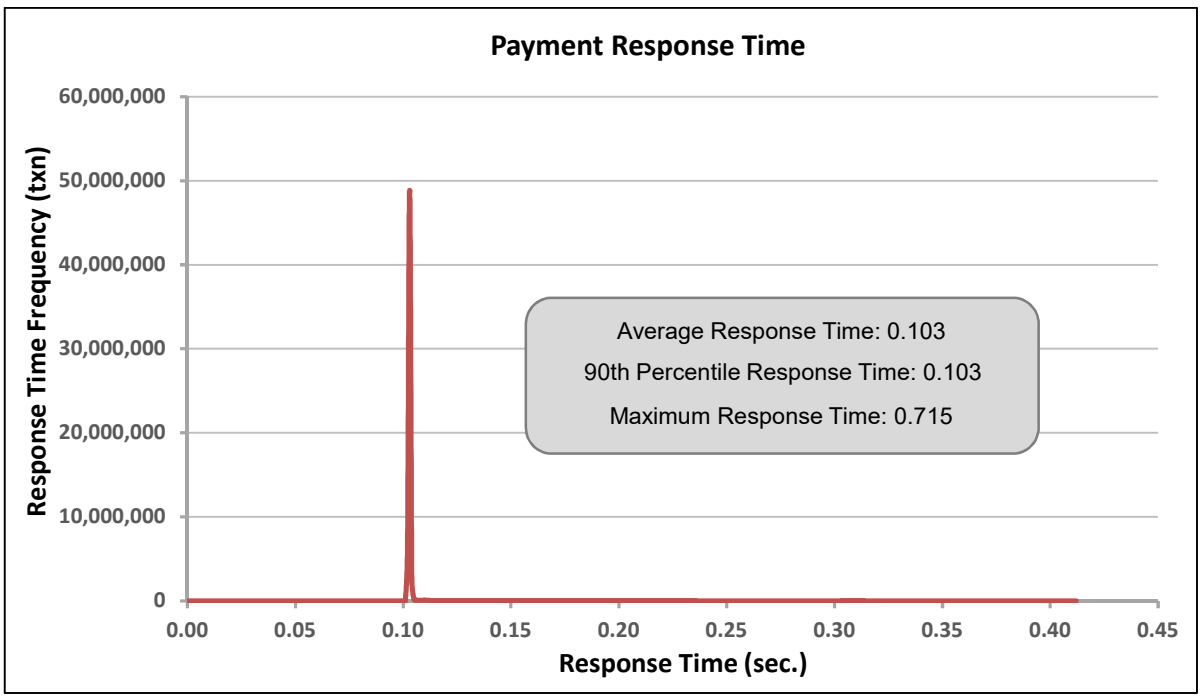


Figure 5.4.1.2: Payment RT Frequency Distribution

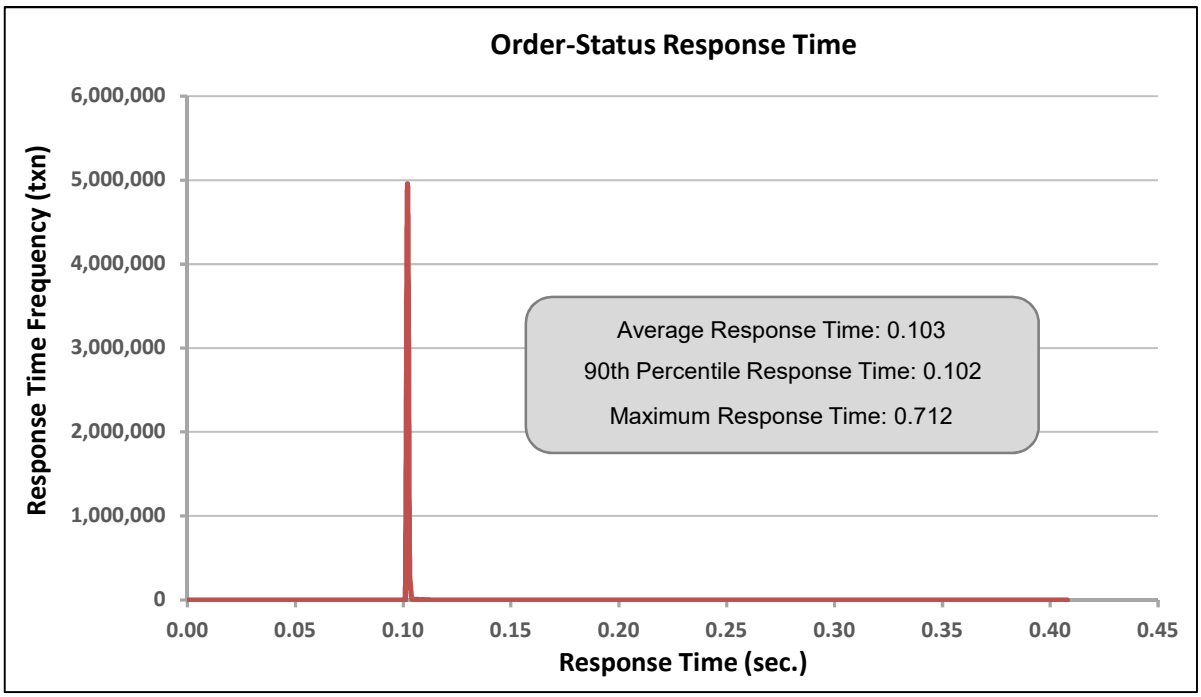


Figure 5.4.1.3: Order-Status RT Frequency Distribution

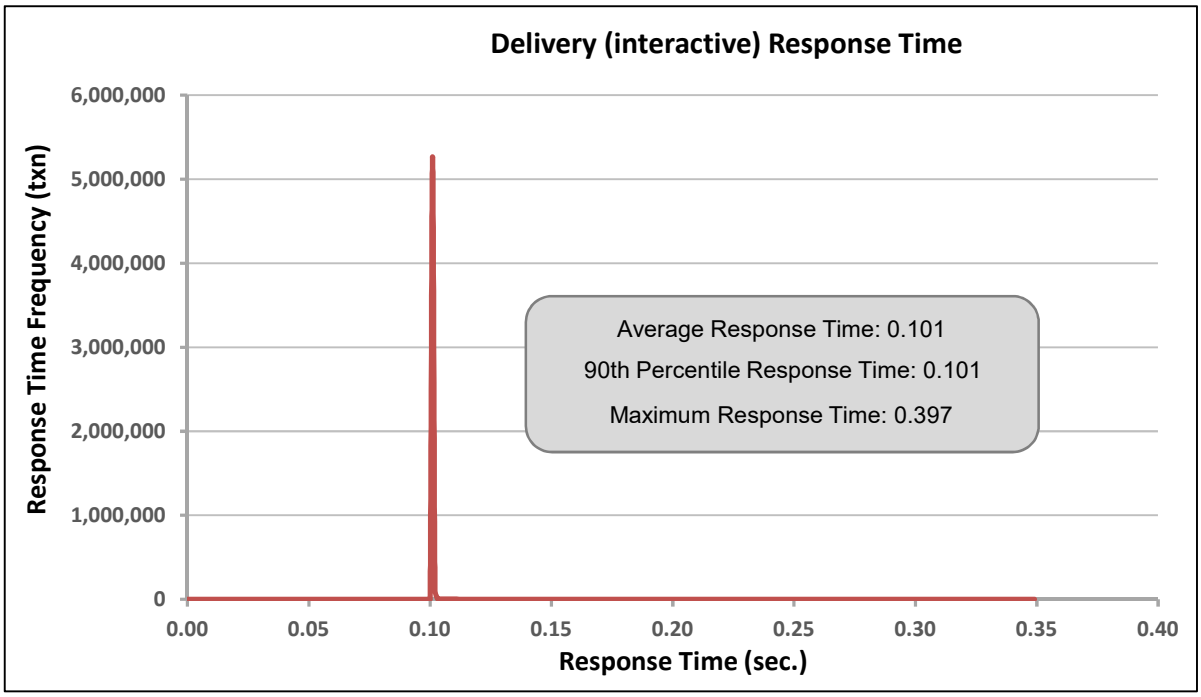


Figure 5.4.1.4: Delivery (Interactive) RT Frequency Distribution

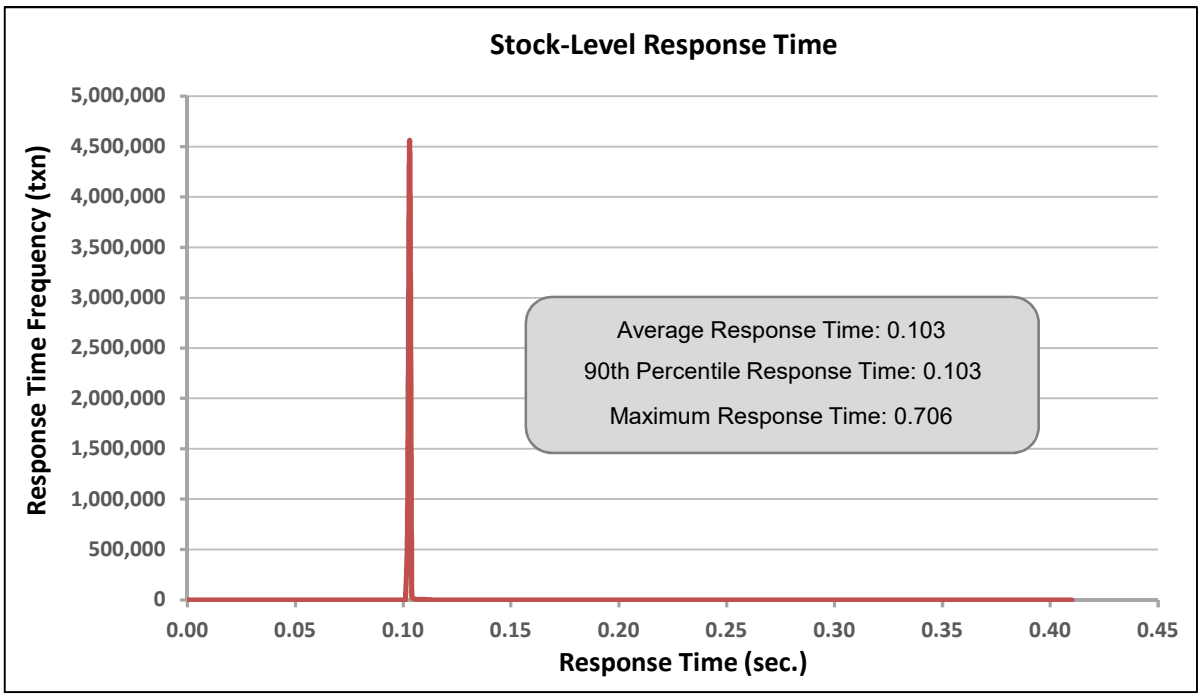


Figure 5.4.1.5: Stock-Level RT Frequency Distribution

5.4.2 Response Time versus throughput

The performance curve for response times versus throughput must be reported for the New-Order transaction.

Figure 5.4.2 shows the Response Time versus throughput curves for the New-Order transaction.

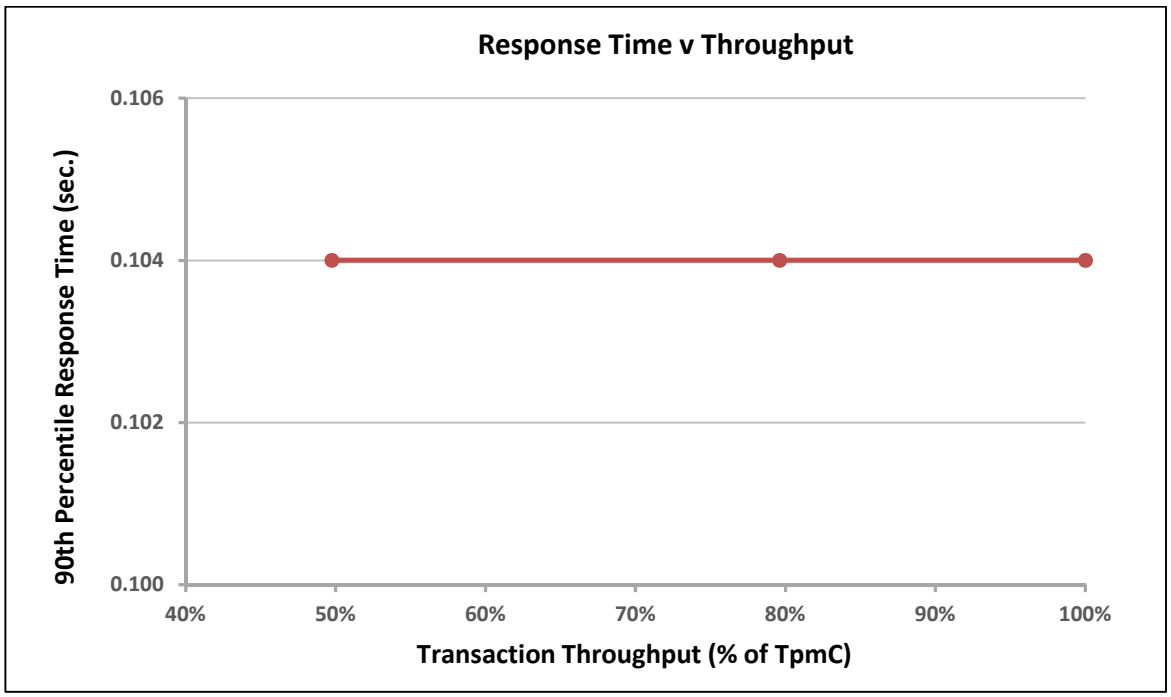


Figure 5.4.2: New-Order RT versus Throughput

5.4.3 Think Time frequency distribution

Think Time frequency distribution curves (see Clause 5.6.3) must be reported for the New-Order transaction.

Figure 5.4.3 shows the Think Time frequency distribution curves for the New-Order transaction.

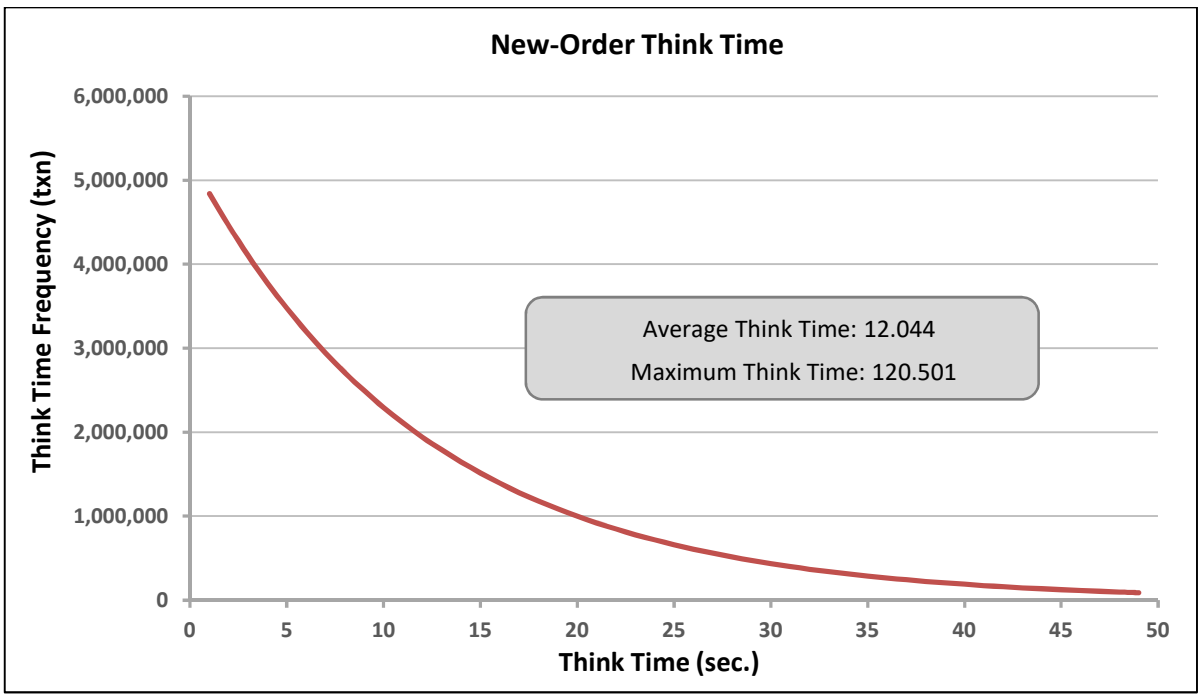


Figure 5.4.3: New-Order Think Time Frequency Distribution

5.4.4 Throughput versus elapsed time

A graph of throughput versus elapsed time must be reported for the New-Order transaction.

Figure 5.4.4 shows the throughput versus elapsed time for the New-Order transaction. The start and end of the Measurement Interval is included on the figure.

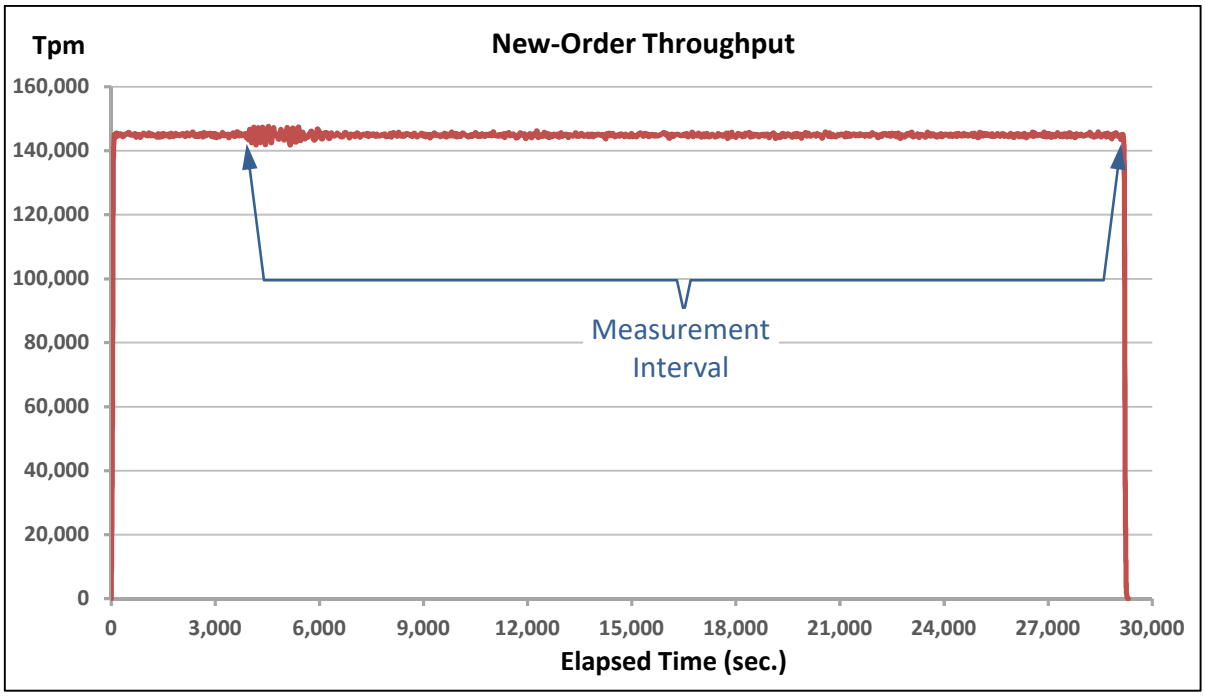


Figure 5.4.4: New-Order Throughput versus Elapsed Time

5.5 Steady State Determination

The method used to determine that the SUT had reached a steady state prior to commencing the measurement interval must be disclosed.

Steady state was determined using real time monitor utilities from the RTE. Steady state was further confirmed by a visual analysis of the throughput graph.

5.6 Work Performed During Steady State

A description of how the work normally performed during a sustained test (for example checkpointing, writing redo/undo log records, etc.) actually occurred during the measurement interval must be reported.

During the test, Goldilocks satisfied all of the ACID properties required by the benchmark specification. Committed transactions write a Redo record in the transaction log, to be used in case of system failure. The Redo records are used for roll-forward recovery during a re-start following a failure. This prevents the system from losing any committed transactions. Checkpoints periodically occurred about every 28 min. and are completed in about 7.5 min.

5.7 Measurement Period Duration

A statement of the duration of the measurement interval for the reported Maximum Qualified Throughput (tpmC) must be included.

The duration of the reported measured interval was 7 hours (7hr = 420min = 25,200sec).

5.8 Transaction Statistics

The percentage of the total mix for each transaction type must be disclosed. The percentage of New-Order transactions rolled back as a result of invalid item number must be disclosed. The average number of order-lines entered per New-Order transaction must be disclosed. The percentage of remote order lines per New-Order transaction must be disclosed. The percentage of remote Payment transactions must be disclosed. The percentage of customer selections by customer last name in the Payment and Order-Status transactions must be disclosed. The percentage of skipped Delivery transactions must be disclosed.

The details of the transaction statistics are reported in the front of this report as part of the Executive Summary.

5.9 Checkpoints

The number of checkpoints in the Measurement Interval, the time in seconds from the start of the Measurement Interval to the first checkpoint, and the Checkpoint Interval must be disclosed.

Two full checkpoints occurred before the Measurement Interval. 14 full checkpoints occurred during the Measurement Interval. The checkpoints' start and end times and durations during the Measurement Interval are listed in table 5.6.

Table 5.6: Checkpoints

Event	Event time	Execution time	Interval
Measurement Interval Begin	2021-06-21 14:50:06	-	-
Checkpoint3 Begin	2021-06-21 15:08:57		00:29:21
Checkpoint3 End	2021-06-21 15:23:15	00:14:18	
Checkpoint4 Begin	2021-06-21 15:38:17		00:29:20
Checkpoint4 End	2021-06-21 15:52:33	00:14:16	
Checkpoint5 Begin	2021-06-21 16:07:40		00:29:23
Checkpoint5 End	2021-06-21 16:22:37	00:14:57	
Checkpoint6 Begin	2021-06-21 16:37:01		00:29:21
Checkpoint6 End	2021-06-21 16:51:50	00:14:49	
Checkpoint7 Begin	2021-06-21 17:06:23		00:29:22
Checkpoint7 End	2021-06-21 17:20:43	00:14:20	
Checkpoint8 Begin	2021-06-21 17:35:45		00:29:22
Checkpoint8 End	2021-06-21 17:50:40	00:14:55	
Checkpoint9 Begin	2021-06-21 18:05:08		00:29:23
Checkpoint9 End	2021-06-21 18:19:36	00:14:28	
Checkpoint10 Begin	2021-06-21 18:34:30		00:29:22
Checkpoint10 End	2021-06-21 18:48:50	00:14:20	
Checkpoint11 Begin	2021-06-21 19:03:53		00:29:23
Checkpoint11 End	2021-06-21 19:18:06	00:14:13	
Checkpoint12 Begin	2021-06-21 19:33:15		00:29:22
Checkpoint12 End	2021-06-21 19:47:36	00:14:21	
Checkpoint13 Begin	2021-06-21 20:02:38		00:29:23
Checkpoint13 End	2021-06-21 20:16:59	00:14:21	
Checkpoint14 Begin	2021-06-21 20:32:00		00:29:22
Checkpoint14 End	2021-06-21 20:46:09	00:14:09	
Checkpoint15 Begin	2021-06-21 21:01:23		00:29:23
Checkpoint15 End	2021-06-21 21:16:07	00:14:44	
Checkpoint16 Begin	2021-06-21 21:30:46		00:29:23
Checkpoint16 End	2021-06-21 21:45:16	00:14:30	
Measurement Interval End	2021-06-21 21:50:07	-	-

Clause 6: SUT, Driver and Communication

6.1 Remote Terminal Emulator (RTE)

If the RTE is commercially available, then its inputs must be specified. Otherwise, a description must be supplied of what inputs (e.g., scripts) to the RTE had been used.

The RTE software used was internally developed. The RTE simulated web users. It generated random input data based on the benchmark requirements and recorded response times and other statistics for each transaction cycle.

6.2 Emulated Components

It must be demonstrated that the functionality and performance of the components being emulated in the Driver System are equivalent to the priced system. The results of the test described in Clause 6.6.3.4 must be disclosed.

No components were emulated by the driver system.

6.3 Functional Diagrams

A complete functional diagram of both the benchmark configuration and the configuration of the proposed (target) system must be disclosed. A detailed list of all hardware and software functionality being performed on the Driver System and its interface to the SUT must be disclosed.

The diagram in Figure 0.1 shows the tested and priced benchmark configurations.

6.4 Networks

The network configuration of both the tested services and proposed (target) services which are being represented and a thorough explanation of exactly which parts of the proposed configuration are being replaced with the Driver System must be disclosed.

The bandwidth of the networks used in the tested/priced configuration must be disclosed.

The diagram in Figure 0.1 shows the network configuration between the components of the tested configuration. The RTE and the SUT are connected through a 1Gbit switch.

The network bandwidths are listed in Figure 0.1.

6.5 Operator Intervention

If the configuration requires operator intervention (see Clause 6.6.6), the mechanism and the frequency of this intervention must be disclosed.

No operator intervention is required to sustain eight hours at the reported throughput.

Clause 7: Pricing

7.1 Hardware and Software Pricing

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

The details of the hardware and software are reported in the front of this report as part of the Executive Summary.

7.2 Three Year Price

The total 3-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 used must be disclosed.

The pricing details for this TPC-C result are reported in the front of this report as part of the Executive Summary.

7.3 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, the reported availability date for the priced system must be the date at which all components are committed to be available.

All components of the priced system are available as of the date of this publication.

Clause 8: Reporting

8.1 Full Disclosure Report

A Full Disclosure report is required in order for results to be considered compliant with the TPC-C benchmark specification

This document constitute the Full Disclosure Report for the TPC-C benchmark result describes within.

Clause 9: Auditor Attestation

9.1 Auditor Information

The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

This benchmark was audited by:

InfoSizing

Doug Johnson

63 Lourdes Drive

Leominster, MA, 01453 USA
Phone: +1 (978) 343-6562

www.sizing.com

9.2 Attestation Letter

The auditor's attestation letter is included in the following pages.

InfoSizing
The Right Metric For Sizing IT

TPC
Certified Auditor

Sejin Hwang
Senior Research Engineer
Telecommunications Technology Association (TTA)
Bundang-ro 47, Bundang-gu, Seongnam-city
Gyeonggi-do, 13591, Republic of Korea

September 7, 2021

I verified the TPC Benchmark™ C v5.11.0 performance of the following configuration:

Platform:	LTechKorea L224S-D
Operating System:	Red Hat Enterprise Linux 7.9
Database Manager:	Goldilocks v3.1 Standard Edition

The results were:

Performance Metric	**144,714 tpmC**
Number of Users	114,000

Server	**LTechKorea L224S-D**		
CPUs	2x Intel® Xeon® Gold 6258R (2.70 GHz, 28-core, 38.5 MB Cache)		
Memory	1,536 GB		
Storage	***Qty***	***Size***	***Type***
	2	1 TB	SAS SSD
	8	1.6 TB	FMD SSD (External)

In my opinion, these performance results were produced in compliance with the TPC requirements for the benchmark.

The following verification items were given special attention:

- The transactions were correctly implemented
- The database records were the proper size
- The database was properly scaled and populated
- The ACID properties were met
- Input data was generated according to the specified percentages
- The transaction cycle times included the required keying and think times
- The reported response times were correctly measured
- At least 90% of all delivery transactions met the 80 Second completion time limit
- All 90% response times were under the specified maximums

63 Lourdes Dr. | Leominster, MA 01453 | 978-343-6562 | www.sizing.com

- The measurement interval was representative of steady state conditions
- The reported measurement interval was over 120 minutes
- Checkpoint intervals were under 30 minutes
- The 60-day storage requirement was correctly computed
- The system pricing was verified for major components and maintenance

Additional Audit Notes:

None.

Respectfully Yours,

Doug Johnson, Certified TPC Auditor

63 Lourdes Dr. | Leominster, MA 01453 | 978-343-6562 | www.sizing.com

Appendix A: Source Code

The source code and scripts used to implement the benchmark is provided as a soft appendix. This soft appendix includes the following files:

```
\ACID
    \ACID\include
    \ACID\src
    \ACID\include\acid.h
    \ACID\src\atom.c
    \ACID\src\compare.c
    \ACID\src\consist.c
    \ACID\src\Delivery.c
    \ACID\src\isol1.c
    \ACID\src\isol2.c
    \ACID\src\isol3.c
    \ACID\src\isol4.c
    \ACID\src\isol5.c
    \ACID\src\isol6.c
    \ACID\src\isol7.c
    \ACID\src\isol8.c
    \ACID\src\isol9.c
    \ACID\src\Makefile
    \ACID\src\NewOrder.c
    \ACID\src\OrderStatus.c
    \ACID\src\Payment.c
    \ACID\src\support.c
\bin
    \bin\load.sh
\html
    \html\DeliveryInput.html
    \html\MainMenu.html
    \html\NewOrderInput.html
    \html\OrderStatusInput.html
    \html\PaymentInput.html
    \html\StockLevelInput.html
\include
    \include\spt_proc.h
    \include\support.h
\java
    \java\Common.java
    \java\Delivery.java
    \java\NewOrder.java
    \java\OrderStatus.java
    \java\Payment.java
    \java\StockLevel.java
\scripts
    \scripts\analyze_system.sql
    \scripts\analyze_table.sql
    \scripts\analyze_table_district.sql
    \scripts\analyze_table_item.sql
    \scripts\analyze_table_new_order.sql
    \scripts\analyze_table_orders.sql
    \scripts\analyze_table_order_line.sql
```

```
    \scripts\analyze_table_stock.sql
    \scripts\analyze_table_warehouse.sql
    \scripts\audit.sql
    \scripts\checkpoint.py
    \scripts\count.sql
    \scripts\create_audit_table.sql
    \scripts\create_index.sql
    \scripts\create_procedure.sql
    \scripts\create_table.sql
    \scripts\create_tablespace.sql
    \scripts\dbcheck.sql
    \scripts\dbtables.sql
    \scripts\runcheck.sql
    \scripts\sys
         \scripts\sys\be
             \scripts\sys\be\part_info.sh
             \scripts\sys\be\reboot_info.sh
             \scripts\sys\be\sw_info.sh
             \scripts\sys\be\sys_info.sh
\src
    \src\free_space.c
    \src\load.c
    \src\load_new.c
    \src\Makefile
    \src\support.c
```

Appendix B: Tunable Parameters

goldilocks.properties.conf

```
TRANSACTION_COMMIT_WRITE_MODE = 1
TRANSACTION_TABLE_SIZE = 1024
UNDO_RELATION_COUNT = 1024
LOG_BUFFER_SIZE = 3G
LOG_FILE_SIZE = 40G
LOG_GROUP_COUNT = 5
PENDING_LOG_BUFFER_COUNT = 8
SPIN_COUNT = 1
BUSY_WAIT_COUNT = 1000
SYSTEM_TABLESPACE_DIR = '/data/db/db1'
SYSTEM_MEMORY_UNDO_TABLESPACE_SIZE = 2G
SYSTEM_MEMORY_TEMP_TABLESPACE_SIZE = 1G
SHARED_MEMORY_STATIC_SIZE = 4G
PARALLEL_IO_FACTOR = 1
PARALLEL_IO_GROUP_1 = '/data/db/db1'
LOG_DIR = '/wal'
CLIENT_MAX_COUNT = 1024
PROCESS_MAX_COUNT = 1024
PARALLEL_LOAD_FACTOR = 16
SHARED_SESSION = NO
CONTROL_FILE_COUNT = 2
CONTROL_FILE_0 = '/wal/control_0.ctl'
CONTROL_FILE_1 = '/wal/control_1.ctl'
```

limit.conf

```
# /etc/security/limits.conf
#
#This file sets the resource limits for the users logged in via PAM.
#It does not affect resource limits of the system services.
#
#Also note that configuration files in /etc/security/limits.d
directory,
#which are read in alphabetical order, override the settings in this
#file in case the domain is the same or more specific.
#That means for example that setting a limit for wildcard domain here
#can be overriden with a wildcard setting in a config file in the
#subdirectory, but a user specific setting here can be overriden only
#with a user specific setting in the subdirectory.
#
#Each line describes a limit for a user in the form:
#
#<domain>        <type>  <item>  <value>
#
#Where:
#<domain> can be:
#        - a user name
#        - a group name, with @group syntax
#        - the wildcard *, for default entry
#        - the wildcard %, can be also used with %group syntax,
#                 for maxlogin limit
#
#<type> can have the two values:
#        - "soft" for enforcing the soft limits
#        - "hard" for enforcing hard limits
#
#<item> can be one of the following:
#        - core - limits the core file size (KB)
#        - data - max data size (KB)
#        - fsize - maximum filesize (KB)
#        - memlock - max locked-in-memory address space (KB)
#        - nofile - max number of open file descriptors
#        - rss - max resident set size (KB)
#        - stack - max stack size (KB)
#        - cpu - max CPU time (MIN)
#        - nproc - max number of processes
#        - as - address space limit (KB)
#        - maxlogins - max number of logins for this user
#        - maxsyslogins - max number of logins on the system
#        - priority - the priority to run user process with
#        - locks - max number of file locks the user can hold
#        - sigpending - max number of pending signals
#        - msgqueue - max memory used by POSIX message queues (bytes)
#        - nice - max nice priority allowed to raise to values: [-20,
19]
#        - rtprio - max realtime priority
#
#<domain>      <type>  <item>         <value>
#

#*               soft    core            0
#*               hard    rss             10000
#@student        hard    nproc           20
#@faculty        soft    nproc           20
#@faculty        hard    nproc           50
#ftp             hard    nproc           0
#@student        -       maxlogins       4

tpcc    soft    nofile  1000000
tpcc    hard    nofile  1000000
tpcc    soft    nproc   unlimited
tpcc    hard    nproc   unlimited

# End of file
```

server.xml

```
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed
with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version
2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at

      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- The contents of this file will be loaded for each web application
-->
<Context>

    <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>

    <!-- Uncomment this to disable session persistence across Tomcat
restarts -->
    <!--
    <Manager pathname="" />
    -->

    <!-- Uncomment this to enable Comet connection tacking (provides
events
         on session expiration as well as webapp lifecycle) -->
    <!--
    <Valve
className="org.apache.catalina.valves.CometConnectionManagerValve" />
    -->

<Resource
    name='jdbc/goldilocks'
    auth='Container'
    type='javax.sql.DataSource'
    driverClassName='sunje.goldilocks.jdbc.GoldilocksDriver'
    url='jdbc:goldilocks://10.100.250.81:22581/test'
    username='test'
    password='test'
    maxActive='10'
    maxIdle='10'
    maxWait='-1'
/>

</Context>
```

Sysctl fe.conf

```
<?xml version='1.0' encoding='utf-8'?>
<!--
  Licensed to the Apache Software Foundation (ASF) under one or more
  contributor license agreements.  See the NOTICE file distributed
with
  this work for additional information regarding copyright ownership.
  The ASF licenses this file to You under the Apache License, Version
2.0
  (the "License"); you may not use this file except in compliance with
  the License.  You may obtain a copy of the License at
```

```
      http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.
  See the License for the specific language governing permissions and
  limitations under the License.
-->
<!-- Note:  A "Server" is not itself a "Container", so you may not
     define subcomponents such as "Valves" at this level.
     Documentation at /docs/config/server.html
 -->
<Server port="8005" shutdown="SHUTDOWN">
  <Listener
className="org.apache.catalina.startup.VersionLoggerListener" />
  <!-- Security listener. Documentation at /docs/config/listeners.html
  <Listener className="org.apache.catalina.security.SecurityListener"
/>
  -->
  <!--APR library loader. Documentation at /docs/apr.html -->
  <!--Initialize Jasper prior to webapps are loaded. Documentation at
/docs/jasper-howto.html -->
  <Listener className="org.apache.catalina.core.JasperListener" />
  <!-- Prevent memory leaks due to use of particular java/javax APIs--
>
  <Listener
className="org.apache.catalina.core.JreMemoryLeakPreventionListener"
/>
  <Listener
className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener
" />
  <Listener
className="org.apache.catalina.core.ThreadLocalLeakPreventionListener"
/>

  <!-- Global JNDI resources
       Documentation at /docs/jndi-resources-howto.html
  -->
  <GlobalNamingResources>
    <!-- Editable user database that can also be used by
         UserDatabaseRealm to authenticate users
    -->
    <Resource name="UserDatabase" auth="Container"
              type="org.apache.catalina.UserDatabase"
              description="User database that can be updated and
saved"

factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
              pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>

  <!-- A "Service" is a collection of one or more "Connectors" that
share
       a single "Container" Note:  A "Service" is not itself a
"Container",
       so you may not define subcomponents such as "Valves" at this
level.
       Documentation at /docs/config/service.html
   -->
  <Service name="Catalina">

    <!--The connectors can use a shared executor, you can define one
or more named thread pools-->
    <!--
    <Executor name="tomcatThreadPool" namePrefix="catalina-exec-"
        maxThreads="150" minSpareThreads="4"/>
    -->


    <!-- A "Connector" represents an endpoint by which requests are
received
         and responses are returned. Documentation at :
         Java HTTP Connector: /docs/config/http.html (blocking & non-
blocking)
         Java AJP  Connector: /docs/config/ajp.html
         APR (HTTP/AJP) Connector: /docs/apr.html
         Define a non-SSL HTTP/1.1 Connector on port 8080
    -->
          <Connector port="8080"
                     acceptCount="150000"
                     maxConnections="141000"
                     connectionTimeout="20000000"
                     maxThreads="1024"
                     maxKeepAliveRequests="-1" keepAliveTimeout="-
1"

protocol="org.apache.coyote.http11.Http11NioProtocol"
redirectPort="8443"
          />

          <!--
    <Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
          -->
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
    <!-- Define a SSL HTTP/1.1 Connector on port 8443
         This connector uses the BIO implementation that requires the
JSSE
         style configuration. When using the APR/native
implementation, the
         OpenSSL style configuration is required as described in the
APR/native
         documentation -->
    <!--
    <Connector port="8443"
protocol="org.apache.coyote.http11.Http11Protocol"
               maxThreads="150" SSLEnabled="true" scheme="https"
secure="true"
               clientAuth="false" sslProtocol="TLS" />
    -->

    <!-- Define an AJP 1.3 Connector on port 8009 -->
    <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />


    <!-- An Engine represents the entry point (within Catalina) that
processes
         every request.  The Engine implementation for Tomcat stand
alone
         analyzes the HTTP headers included with the request, and
passes them
         on to the appropriate Host (virtual host).
         Documentation at /docs/config/engine.html -->

    <!-- You should set jvmRoute to support load-balancing via AJP
ie :
    <Engine name="Catalina" defaultHost="10.100.250.24"
jvmRoute="jvm1">
    -->
    <Engine name="Catalina" defaultHost="localhost">

      <!--For clustering, please take a look at documentation at:
          /docs/cluster-howto.html  (simple how to)
          /docs/config/cluster.html (reference documentation) -->
      <!--
      <Cluster
className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
      -->

      <!-- Use the LockOutRealm to prevent attempts to guess user
passwords
           via a brute-force attack -->
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <!-- This Realm uses the UserDatabase configured in the global
JNDI
             resources under the key "UserDatabase".  Any edits
             that are performed against this UserDatabase are
immediately
             available for use by the Realm.  -->
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
               resourceName="UserDatabase"/>
      </Realm>

      <Host name="localhost"  appBase="webapps"
            unpackWARs="true" autoDeploy="true">

        <!-- SingleSignOn valve, share authentication between web
applications
             Documentation at: /docs/config/valve.html -->
        <!--
        <Valve
className="org.apache.catalina.authenticator.SingleSignOn" />
        -->

        <!-- Access log processes all example.
             Documentation at: /docs/config/valve.html
             Note: The pattern used is equivalent to using
pattern="common"
       <Valve className="org.apache.catalina.valves.AccessLogValve"
directory="logs"
              prefix="localhost_access_log." suffix=".txt"
              pattern="%h %l %u %t &amp;quot;%r&amp;quot; %s %b" />-->

      </Host>
    </Engine>
  </Service>
</Server>
```

Appendix C: Price Quotations

DB Server

LTK엘텍코리아 www.ltechkorea.com

견 적 서

수 신 : TTA
참 조 : 황세진

서울 금천구 벚꽃로 234, 에이스하이엔드타워 6차 1904호
Tel : 1599-7230 Fax : 02)515-2263

엘텍코리아㈜

대표이사 김 세 호 / 이 풍 연

담당자 :
E-mail :
T e l :

1. 견적일자 : 2021-09-01
2. 견적내용 :
3. 납품일자 : 발주시 협의사항
4. 견적유효기간 : 발행후 3개월
5. 지불조건 : 협의
6. 제품보증기간 : 납품 당일후 3년 보증

(단위:대, 원, 부가세별도)

제 품 설 명		수량	표준단가	공급가	공급가 총액
L224S-D Server	CPU(Xeon Gold 6258R 28C*2ea), MEM(128GB*12ea), DISK(SSD 1TB*2ea)	1	207,494,400	25,936,800	25,936,800
CPU	인텔® 제온® 골드 6258R 프로세서 28C/56Th 2.7Ghz (Up To 4Ghz)	2			
MEMORY	128GB fully buffered DDR4 Memory, LRDIMM	12			
DISK_OS	SSD 1TB(960GB), Enterpeise	2			
RAID Controller	RAID Controller 0,1, 5, 6, 10, 50, 60 지원(1GB Memory, 8 port) LSI 9361 8i	1			
NIC	Intel® Ethernet Server Adapter I350-T2V2 Intel® Ethernet Converged Network Adapter X550-T2	1 1			
HBA	QLE 2692 PCI-e 3.0 8x 지원 - 16G Fc gbic	1			
CHASSIS	L224S-D Server(power 920watt) 2.5 inch disk bay * 24EA - 12 memory SLOT - Drive : 24bay 2.5" SSD Only - 24 Hot-Swap SAS/SATA Backplane - PCI-e 3.0 slot * 7 - RAID Controller 0,1, 5, 10 지원 - Dual-Giga Ethernet(on_board) - ATI ES1000 64MB On Board video - 920W Dual Power	1			
WARRANTY	3년 무상 분기별 유지보수 점검포함 -이하여백-				
				공급가 총액	25,936,800
				최종 견적가	25,930,000

1. 비 고 : 1) 위 구성 부품은 발주시점의 수급상황에 따라 딜리버리 및 가격변동 있을수 있음.

2. 견적서를 검토하신 후 제품 발주를 원하시는 경우 하단의 내용을 작성하여 날인/직인 후 스캔본을 회신 주시면 발주서의 효력을 갖습니다.
또는, 귀사 양식의 발주서를 팩스나 메일로 보내주시기 바랍니다. Fax. 02-515-2263

1) 발 주 담 당
2) 연 락 번 호
3) 결 제 일 자
4) 배 송 장 소

WAS Server

견 적 서

견적일	HSR 2021-09-01 -001	공급자	업체명	㈜한성
수신업체	한국정보통신기술협회		대표자	한 동 [illegible]
담당자	서병준 님		사업자등록번호	106-86-17[illegible]0
연락처	sbj8388@tta.or.kr		업태 / 종목	제조,서비스 / 컴퓨터 및 주변기기
유효기간	견적 후 7일		사업장 소재지	서울특별시 강서구 마곡중앙8로7길 35 (마곡동,한성타워)
납기 / 지급조건	협의 후 결정 / 현금(선입금)		담당자 / 연락처	김신영 이사 / 010-4358-0113

품명	상세스펙		수량	공급가 (VAT포함)
데스크톱컴퓨터	모델명	DT-S170G1IV0	5	1,551,000
	식별번호	24110096		
	CPU	Intel Core i7 10700K(3.8Ghz Up to 5.1Ghz / 16M / 8Core / 16Thread)		
	MainBoard	H410 칩셋		
	RAM	DDR4 32GB		
	HDD	2TB (7200RPM - 3.5형)		
	SSD	512GB (M.2)		
	VGA	Geforce GTX1660 OC 6GB		
	CASE	Middle 2 CASE (6861)		
	PSU	ATX 500W		
	기타	키보드, 마우스, 마우스패드		
옵션품목 (별도선택)	OS	N/A		-
	ODD	N/A		-
	Card Reader	N/A		-
	설치	미설치		-
총 합계	(VAT 포함)		₩	7,755,000
담당자 정보	* 담당자: 행망영업팀 김신영 이사 * 직통번호: 070-7595-7176 * 휴대폰번호: 010-4358-0113 * E-mail: young@hansungcom.co.kr			
특이사항	* 1년 무상보증			

Storage

UNIWIDE

Quotation

견 적 일 자: UWT20 - 1012 -100
수 신: 한국정보통신기술협회
납 품 일 자: 고객 요청 시,
견적유효기간: 견적 후 7일
결 제 조 건: 납품 후 즉시 현금
무상유지보수: 3년
기 타:

UNIWIDE Technologies, Inc.
서울시 구로구 디지털로26길5, 717~718호
(에이스하이엔드타워 1차)
대표이사 : 우 종 운
TEL : 070-7306-0500
FAX : 02-866-0037

Sales Manager
영 업 대 표 : 유통사업팀 / 김창환 팀장
전 화 번 호 : 070-7306-0550
휴 대 전 화 : 010-8335-1686
이 메 일 : chkim@uniwide.co.kr

공 급 금 액: **129,800,000** 원(부가세 별도

모 델: FCH2800

PART NUMBER	제품명	수량	도입수량	공급단가	공급합가
FCH2800	**All Flash Storage - FCH2800**	1			
T0001-0117-00	FCH2800 Controller Device	1			
T0001-0117-01	Back-end Bus Adapter 12G SAS	1			
T0001-0117-02	16G 8-Port Host Bus Adapter	4			
T0001-0117-03	Cache Interconnect Adapter	1			
T0001-0117-04	Cache Memory DDR-3 (32GB)	16	1	72,250,000	₩72,250,000
T0001-0117-05	FCH2800 Flash Disk Drive Expantion Unit	1			
T0001-0117-06	FCH2800 controller cpu Board	1			
T0001-0117-07	Rack 600x1200x2010 mm (WxDxH) 42U	1			
T0001-0117-08	Storage Management SW	1			
61001-0001-00	UTP CAT5e Ethernet Cable 1M	1			
42119-0005-00	Power Cord, NICETECH, 2.5M	2			
T22601-0117-03	1.6TB Flash Memory Disk Drive	1	8	3,900,000	₩31,200,000
	3-yrs 24x7x4hrs Onsite Support Service	1	1	26,350,000	₩26,350,000
제 안 가					**₩129,800,000**
부 가 세					**₩12,980,000**
부 가 세 포 함 가					**₩142,780,000**

본견적서를 발주를 진행하고 싶으시다면 명판 및 직인을 날인하시어 팩스를 송부하여 주시기 바랍니다.

명판	직인

RHEL/JWS

rockPLACE
(주)락플레이스 www.rockplace.co.kr

㈜락플레이스
135-120 서울시 강남구 신사동 634-10 윤당빌딩 3층 Tel.02)6251.7788 Fax.02)6251.6677

rockPLACE, Inc.
3F, Yundang bldg, 634-10, Shinsa-dong, Gangnam-gu,Seoul, Korea Tel : 822-6251-7788 Fax: 822-6251-6677

견 적 서

REF No.	: 2021RP08-2003	TERMS AND CONDITION	
DATE	: 2021. 08. 20.		
COMPANY	: TTA	납 기	: 발주후 4주이내
ATTN	: 서 병 준 선임 연구원님 귀하 TEL : 010-5110-5598	유지보수	: 2022.1.1 ~ 2024.12.31
Email	: sbj8388@tta.or.kr	결제조건	: 익월말 현금
FROM	: ㈜ 락플레이스 정 경환 차장 TEL : 010-4298-3447	유효기간	: 견적일로부터 4개월

下記와 같이 見積합니다.

㈜ 락플레이스
대표이사 서 동 식

ITEM DESCRIPTION (VAT 별도, 단위 : 원)

Part No.	Description	수량	소비자가	공급단가	공급합계
OS	Red Hat Enterprise Linux Operating System Platform				
RH00004	Red Hat Enterprise Linux Server, Standard (Physical or Virtual Nodes) 3Year support : Easy ISOs: OS, Source, Documentation ISO Images 가상화 Guest OS : 2guests Red Hat Network 서비스 : 3Year Phone,email Support : 09:00 ~ 17:00 Scope of Coverage : Standard Maximum Memory Support: Unlimited	4	4,098,000	2,400,000	9,600,000
	rockPLACE carePACK for Linux				
RP-CSP(OS)	rockPLACE Support Carepack - Linux Standard (3년) - Per Server 3 Year, 24x7, 4hr response 이메일, 전화, 원격지원, 현장지원 서비스 On Site Support - 년간 Total 10회 (아래 지원내역에 준함) - Installation & Startup Service Included - Problem tracking/Emergency assistance - Update, Patch 작업 지원 - 서비스, 시스템 환경, 네트워크 환경 설정 변경 지원 - 인수 시험, 성능 시험, 비상 복구 훈련 지원	4	5,000,000	2,000,000	8,000,000
소 계 금 액					17,600,000

Part No.	Description	수량	소비자가	공급단가	공급합계
WEB	Red Hat JBoss Web Server				
MW00123F3	Red Hat JBoss Web Server, 4-Core Standard 3Year - 전화/웹 지원 : 월-금, 9 a.m. - 5 p.m. 4시간내 응답 - unlimited incidents, - 4 business hour SLA	6	2,138,000	1,260,000	7,560,000
	rockPLACE carePACK for Linux				
RP-CSP(WAS)	rockPLACE Support Carepack - JBoss Standard (3년) - Per 4core 3 Year, 24x7, 4hr response 이메일, 전화, 원격지원, 현장지원 서비스 On Site Support - 년간 Total 10회 (아래 지원내역에 준함) - Installation & Startup Service Included - Problem tracking/Emergency assistance - Update, Patch 작업 지원 - 서비스, 시스템 환경, 네트워크 환경 설정 변경 지원 - 인수 시험, 성능 시험, 비상 복구 훈련 지원	6	12,000,000	6,000,000	36,000,000
소 계 금 액					43,560,000

합 계	61,160,000
부가세	6,116,000
합 계(부가세포함)	67,276,000

Remarks

1. Red Hat 제품은 년간 Subscription 제품이며, 기간이 만료되실 경우 Renewal을 하셔야합니다.
2. 발주 시에는 반드시 고객정보(엔드유저명, 담당자, 연락처, Email)가 있어야 합니다.
3. OnSite 방문지원이 필요하실 경우에는 케어팩을 구매하셔야 합니다.

Network Switch

상품상세정보

확대보기 수량 : 대

네트워크스위치

업체명 : 주식회사엔에스지[중소기업] 계약자/공급자 정보조회
계약방법 : 다수공급자계약
규격명 : 네트워크스위치, 유비쿼스, uSafe3010-24T, 24port (공급)
가격 : 1,900,000 원 다량납품할인율 확인 → Price
단위 : 대
원산지 : 대한민국
주요부품1[원산지] : Firmware[대한민국]
주요부품2[원산지] : Main Board[대한민국]
제조사 : (주)유비쿼스
납품장소 : 수요기관 지정장소
인도조건 : 현장설치도
공급지역 : 전지역
부가세여부 : 부가가치세포함
계약기간 : 2017/08/24 ~ 2022/08/23
납품기한 : 60일 (납품요구일로부터)
조달수수료여부 : 조달수수료 별도 조달수수료 안내·계산
첨부파일 : 2018/02/01_00176118204-(계약예규)물품구매(제조)계약일반조건(기재부예규제328호20161230).hwp
2018/02/02_00176118204-다수공급자계약특수조건(조달청공고2017-73호,2017.7.14).hwp
2018/02/03_00176118204-물품구매계약품질관리특수조건(2017.8.28.).hwp
2018/02/04_00176118204-엔에스지_규격서(2017.08).zip
대분류 : 09 - 전자/정보/통신/영상
중분류 : 07 - 음향장비 및 신호장치
물품분류번호 : 43222612
세부품명번호 : 4322261201
물품식별번호 : 22917889
계약번호 : 00176118204-9
징수구분 : 후징수

DBMS

Quotation

㈜TTA 貴中

Title : TPC-C Performance&Quality Authentication

참 조 : 황 세진 선임연구원
견적일자 : 2021년 8월 13일
유효기간 : 견적일로부터 4개월

SUNJE SOFT
(주)선재소프트

대표이사 : 김 기 완 (인)

주 소 : 서울시 영등포구 당산로
171 금강펜테리움IT타워 604호
영업대표 : 사업본부 최승렬 이사
전화번호 : 010-9312-0188
e-mail : slchoi@sunjesoft.com

* Goldilocks Standard Edition for LINUX 1식 (단위 : 원)

No.	Description	Unit List Price	Q'ty	Total Amount Price	Offer Price
1	Goldilocks Ver 3.1 DBMS Standard Edition	₩96,000,000	1 Set(s)	₩96,000,000	₩32,000,000
	- Query Processes Module				
	- Storage Management Module				
	Goldilocks DBMS License Fee	License Proposal Price			₩32,000,000
2	DBMS Implementaion & Supports	₩10,000,000	3 Set(s)	₩30,000,000	₩14,400,000
	Goldilocks Technical Supports Fee(3yr)	Support Proposal Price			₩14,400,000
Total Amount(VAT Exclude)				₩126,000,000	₩46,400,000

Goldilocks Total Amount (Offer Price)	₩46,400,000

* For Technical supports, it indicates 24 x 7 x 4 hours of support