# TPC Benchmark™ C

# Full Disclosure Report

First Edition

30–Oct–2020

*Using*

## Goldilocks v3.1 Standard Edition

*on*

## UNIWIDE RC2212

**First Edition: 2-Nov-2020**

*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 obtained 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

# 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 UNIWIDE RC2212

**Goldilocks v3.1 Standard Edition on UNIWIDE RC2212**

| Company Name | System Name | Database Software | Operating System |
|---|---|---|---|
| Telecommunications Technology Association | UNIWIDE RC2212 | Goldilocks v3.1 Standard Edition | RedHat Enterprise Linux 7.8 |

**TPC Benchmark™ C Metrics**

| Total System Cost | TPC-C Throughput | Price/Performance | Availability Date |
|---|---|---|---|
| ₩ 222,550,000 (KRW) | 76,174 tpmC | 2,922 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.*

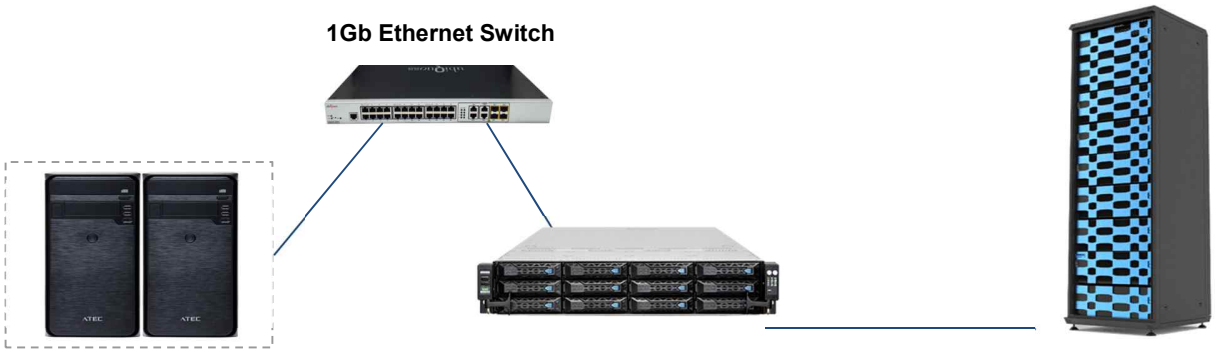
*Despite the fact that this benchmark offers a rich environment that emulates many OLTP applications, this benchmark does not reflect the entire range of OLTP requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-C 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](http://www.tpc.org)*

| | | | | |
|---|---|---|---|---|
| | **Goldilocks v3.1 Standard Edition on UNIWIDE RC2212** | | | **TPC-C Version 5.11.0**<br>**TPC Pricing 2.6.0**<br>Report Date<br>**2-Nov-2020** |
| Total System Cost | TPC-C Throughput | Price/Performance | | Availability Date |
| **₩ 222,550,000 (KRW)** | **76,174 tpmC** | **2,922 KRW/tpmC** | | **Available Now** |
| Server Processors/Cores/Threads | Database Manager | Operating System | Other Software | Number of Users |
| **2/20/40** | **Goldilocks v3.1 Standard Edition** | **RHEL 7.8** | **JBoss Web Server** | **60,000** |

**1Gb Ethernet Switch**



**Web Application Server**

**2 x ATEC A6HGBCP**
- 1 x Intel Xeon E3-1270 V5 3.60GHz
- 2 x 8GB Memory
- 1 x 1TB SATA HDD
- 1 x 250GB SATA SSD
- 1 x 1Gb Ethernet

**Database Server**

**1 x UNIWIDE RC2212**
- 2 x Intel Xeon Silver 4210 2.20GHz
- 24 x 32GB (768GB) Memory
- 2 x 1.2TB SAS HDD
- 1 x 16Gb 2-Port Host Bus Adaptor
- 1 x 2-port 10Gb Ethernet

**Storage**

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

| System Components | DB Server | | WAS Server | |
|---|---|---|---|---|
| | **Quantity** | **Description** | **Quantity** | **Description** |
| Processors/Cores/Threads | 2/20/40 | Intel Xeon Silver 4210 2.20GHz | 1/4/8 | Intel Xeon E3-1270 V5 3.60GHz |
| Memory | 24 | 32GB | 2 | 8GB |
| Network | 1<br>1 | Intel® Ethernet Server Adapter I350<br>Intel® Ethernet Controller 10G X550T | 1 | Intel® Ethernet Connection I219-LM |
| Storage Controller | 1<br>1<br>1 | MegaRAID SAS-3 3108<br>C620 Series SATA Controller<br>QLogic QLE2692 16Gb HBA | 1 | Intel® Sunrise Point-H SATA controller |
| Storage Device | 2<br>8 | 1.2TB SAS HDD<br>1.6TB FMD SSD (External) | 1<br>1 | 1TB SATA HDD<br>250GB SATA SSD |
| Total Storage Capacity | | 15.2TB | | 1.25 TB |

| | Goldilocks v3.1 Standard Edition on UNIWIDE RC2212 | TPC-C Version 5.11.0<br>TPC Pricing 2.6.0<br>Report Date<br>2-Nov-2020<br>Available Now |
|---|---|---|

| Description | Part Number | Source | Unit Price | Qty | Price | 3-Yr. Maint. Price |
|---|---|---|---|---|---|---|
| **Server Hardware** | | | | | | |
| **1 x DB Server – Uniwide RC2212** | SR285 K2 | 1 | 9,230,000 | 1 | **9,230,000** | |
| Uni RC2212(800Wx2)_3.5x12_RAID 2GB | SVR-UNI-RC2212-0010 | 1 | (included) | 1 | | |
| Intel Xeon Silver 4210 10C/20T 2.2GHz 14MB 85W, SRFBL | CPU-S4200-IN-0030 | 1 | (included) | 2 | | |
| 32GB DDR4-2666 R ECC | MEM-D4R26-SA-0030 | 1 | (included) | 24 | | |
| HDD SAS 12G 1.2TB 10K 2.5 인치 512e | HDD-SAS0-RP-0050 | 1 | (included) | 2 | | |
| FC Ctrl 16Gb/s 2CH QLE2692 | HBA-16G2P-RP-0010 | 1 | (included) | 1 | | |
| CARD LAN Intel 10G 2port UTP | LAN-10G2PR-IN-0010 | 1 | (included) | 1 | | |
| Maintenance - 7x24x4 Care Pack (3-yrs) | | 1 | (included) | 1 | | |
| | | | | | | |
| | | | | | | |
| **2 x WAS Servers (per server) - A6HGBDPNN** | A6HGBDPNN | 2 | 1,925,000 | | **3,850,000** | |
| Intel® Xeon® Processor XEON E3-1270V5, 3.6GHz | | 2 | (included) | 1 | | |
| DDR4 8GB ( 2400MHz / PC4 19000 / ECC / REG ) | | 2 | (included) | 2 | | |
| Samsung 850EVO, 250GB, SATA3, TLC, MEX | | 2 | (included) | 1 | | |
| 1TB, 7200RPM, 64M, SATAIII, ST1000DM010 | | 2 | (included) | 1 | | |
| KUB-1407, USB, Black | | 2 | (included) | 1 | | |
| MUB-1407, USB, 1000DPI, Black | | 2 | (included) | 1 | | |
| Tower, 500W, ATX | | 2 | (included) | 1 | | |
| UTP CAT5e Ethernet Cable 1M | | 2 | (included) | 1 | | |
| Power Cord, NICETECH, 2.5M | | 2 | (included) | 1 | | |
| A6HGBDPNN 7x24x4 Care Pack (3-yrs) | | 2 | (included) | 1 | | |
| **Server Hardware Sub Total** | | | | | **13,080,000** | |
| | | | | | | |
| **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** |
| | | | | | | |
| | | | | | | |
| **Client/Server Software** | | | | | | |
| Red Hat Enterprise Linux Server Standard 3yrs | RH00004F3 | 4 | 4,098,000 | 3 | 12,294,000 | |
| RHEL Server Standard Maintenance - 3yrs 24x7x4hrs | RSC-LSF3 | 4 | 6,000,000 | 3 | | 18,000,000 |
| Red Hat JBoss Web Server 4-Core Standard 3Year | MW00123F3 | 4 | 2,144,000 | 2 | 4,288,000 | |
| JBoss Web Server per 16Core 3Year Maintenance | RSC-JSF3 | 4 | 12,000,000 | 2 | | 24,000,000 |
| 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** | | | | | **112,582,000** | **72,000,000** |
| | | | | | | |
| **Other Hardware** | | | | | | |
| UbiQuoss uSafe3010-24T (10G, 24-port)(w/spares) | 22917889 | 6 | 1,850,000 | 3 | 5,550,000 | |
| **Other Hardware Sub Total** | | | | | **5,550,000** | |
| | | | | | | |
| **Discounts*** | | | | | | |
| SW Discount - Goldilocks | | | | | -64,000,000 | -15,600,000 |
| Red Hat OS Discount | | | | | -5,094,000 | -12,000,000 |
| Red Hat JBoss Discount | | | | | -1,768,000 | -12,000,000 |
| **Discounts Sub Total** | | | | | **-70,862,000** | **-39,600,000** |
| | | | | | | |
| **Total** | | | | | **163,800,000** | **58,750,000** |

**Pricing Notes**

1) UNIWIDE Technologies Inc. 4) Rockplace Inc.
2) ATECSYSTEM Co., Ltd. 5) Sunjesoft Inc.
3) UNIWIDE Technologies 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(₩): 222,550,000**

**TPC-C throughput: 76,174 tpmC**

**Price/Performance: 2,922 ₩ / 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 UNIWIDE RC2212 | TPC-C Version 5.11.0<br>TPC Pricing 2.6.0<br>Report Date<br>2-Nov-2020<br>Available Now |
|---|---|---|

| MQTh, computed Maximum Qualified Throughput | 76,174 tpmC | |
|---|---|---|

| Response Times (seconds) | Min | Average | 90th | Max |
|---|---|---|---|---|
| New-Order | 0.102 | 0.104 | 0.105 | 0.495 |
| Payment | 0.102 | 0.104 | 0.105 | 0.484 |
| Order-Status | 0.102 | 0.103 | 0.103 | 0.435 |
| Delivery (interactive portion) | 0.101 | 0.101 | 0.101 | 0.354 |
| Delivery (deferred portion) | 0.002 | 0.006 | 0.009 | 0.325 |
| Stock-Level | 0.102 | 0.103 | 0.104 | 0.499 |
| Menu | 0.101 | 0.102 | 0.102 | 0.503 |

Emulated Display Delay: 0.1 sec.

| Transaction Mix | Percent | Number |
|---|---|---|
| New-Order | 44.980% | 31,993,327 |
| Payment | 43.011% | 30,592,647 |
| Order-Status | 4.003% | 2,846,997 |
| Delivery | 4.003% | 2,847,317 |
| Stock-Level | 4.003% | 2,847,380 |

| Keying Times (seconds) | Min | Average | Max |
|---|---|---|---|
| New-Order | 18.001 | 18.001 | 18.001 |
| Payment | 3.001 | 3.001 | 3.001 |
| Order-Status | 2.001 | 2.001 | 2.001 |
| Delivery | 2.001 | 2.001 | 2.001 |
| Stock-Level | 2.001 | 2.001 | 2.001 |

| Think Times (seconds) | Min | Average | Max |
|---|---|---|---|
| New-Order | 0.001 | 12.043 | 120.501 |
| Payment | 0.001 | 12.043 | 120.501 |
| Order-Status | 0.001 | 10.045 | 100.501 |
| Delivery | 0.001 | 5.026 | 50.301 |
| Stock-Level | 0.001 | 5.029 | 50.301 |

| Test Duration | |
|---|---|
| Ramp-up time | 65 min |
| Measurement Interval (MI) | 420 min |
| Checkpoints in MI | 15 |
| Checkpoint Interval (Average / Max) | 27.58 min / 27.59 min |
| Number of Transactions in MI (all types) | 71,127,668 |

# 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 UNIWIDE Technologies 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.*

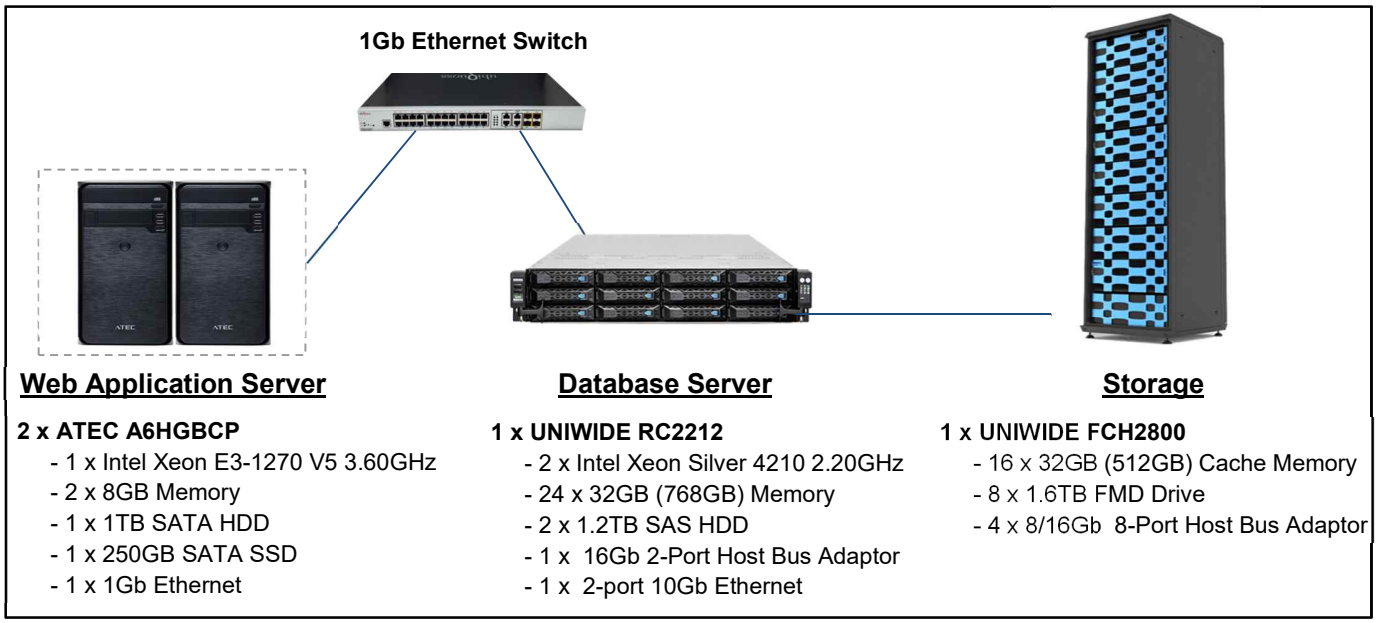
Appendix B contains the tunable parameters for the database, the operating system, and the transaction monitor.

## 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<br>Remote warehouse order lines<br>Rolled back transactions<br>Average items per order | 99.001%<br>0.999%<br>1.001%<br>9.999 |
| Payment | Home warehouse<br>Remote warehouse<br>Accessed by last name | 85.007%<br>14.993%<br>59.994% |
| Order Status | Accessed by last name | 59.986% |
| Delivery | Skipped transactions | 0 |
| Transaction Mix | New Order<br>Payment<br>Order status<br>Delivery<br>Stock level | 44.980%<br>43.011%<br>4.003%<br>4.003%<br>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 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.

### Isolation Test 9

*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.

## 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 indicates whether SUT has media cache failure durability.
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 indicates 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 | 6,000 |
| District | 60,000 |
| Customer | 180,000,000 |
| History | 180,000,000 |
| Order | 180,000,000 |
| New Order | 54,000,000 |
| Order Line | 1,799,616,364 |
| Stock | 600,000,000 |
| Item | 100,000 |
| 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<br>/data/db/db2<br>/data/db/db3<br>/data/db/db4<br>/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 | 76,174.588 |

| Table | Rows | Data | Index | Initial Population | 5% Growth | 8-Hour Growth | Required Runtime Space |
|---|---|---|---|---|---|---|---|
| WAREHOUSE | 6,000 | 48,368 | 168 | 48,536 | 2,427 | 0 | 50,963 |
| DISTRICT | 60,000 | 7,688 | 1,816 | 9,504 | 475 | 0 | 9,979 |
| CUSTOMER | 180,000,000 | 115,734,608 | 14,721,816 | 130,456,424 | 6,522,821 | 0 | 136,979,245 |
| NEW_ORDER | 54,000,000 | 3,408,912 | 1,901,344 | 5,310,256 | 265,513 | 0 | 5,575,769 |
| ITEM | 100,000 | 10,800 | 2,792 | 13,592 | 680 | 0 | 14,272 |
| STOCK | 600,000,000 | 220,762,544 | 19,540,384 | 240,302,928 | 12,015,146 | 0 | 252,318,074 |
| HISTORY | 180,000,000 | 14,759,688 | 0 | 14,759,688 | 0 | 2,998,168 | 17,757,856 |
| ORDERS | 180,000,000 | 11,413,848 | 13,563,488 | 24,977,336 | 0 | 2,318,520 | 27,295,856 |
| ORDER_LINE | 1,799,616,364 | 168,731,480 | 70,179,504 | 238,910,984 | 0 | 34,274,803 | 273,185,787 |
| Total | | 534,877,936 | 119,911,312 | 654,789,248 | 18,807,062 | 39,591,491 | 713,187,801 |

| 60-Day Requirements | |
|---|---|
| Dynamic-Space | 194,905,016 |
| Free-Space | 592,272 |
| Static-Space | 459,884,232 |
| Daily-Growth | 39,591,491 |
| Daily-Spread | 0 |
| 60-Day Space | **2,835,373,719** |

| Memory Requirements | |
|---|---|
| Final Allocation | 725,426,752 |
| Non-Growing 5% | 18,807,062 |
| 1-Day Memory | **744,233,814** |

| Storage Requirements | |
|---|---|
| Total Disk Space | 6,754,527,960 |
| Log space used | 104,857,600 |
| 60-Day Space | 2,835,373,719 |
| Remaining Space | **3,814,296,641** |

# 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.

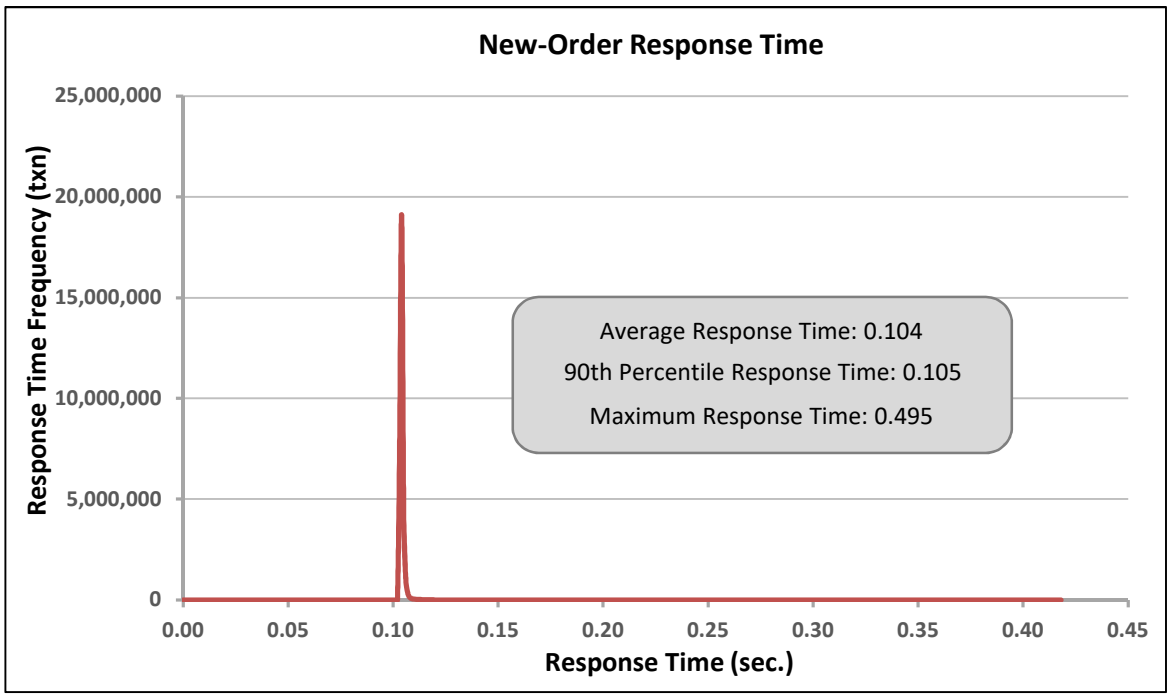



**Figure 5.4.1.1: New-Order RT Frequency Distribution**

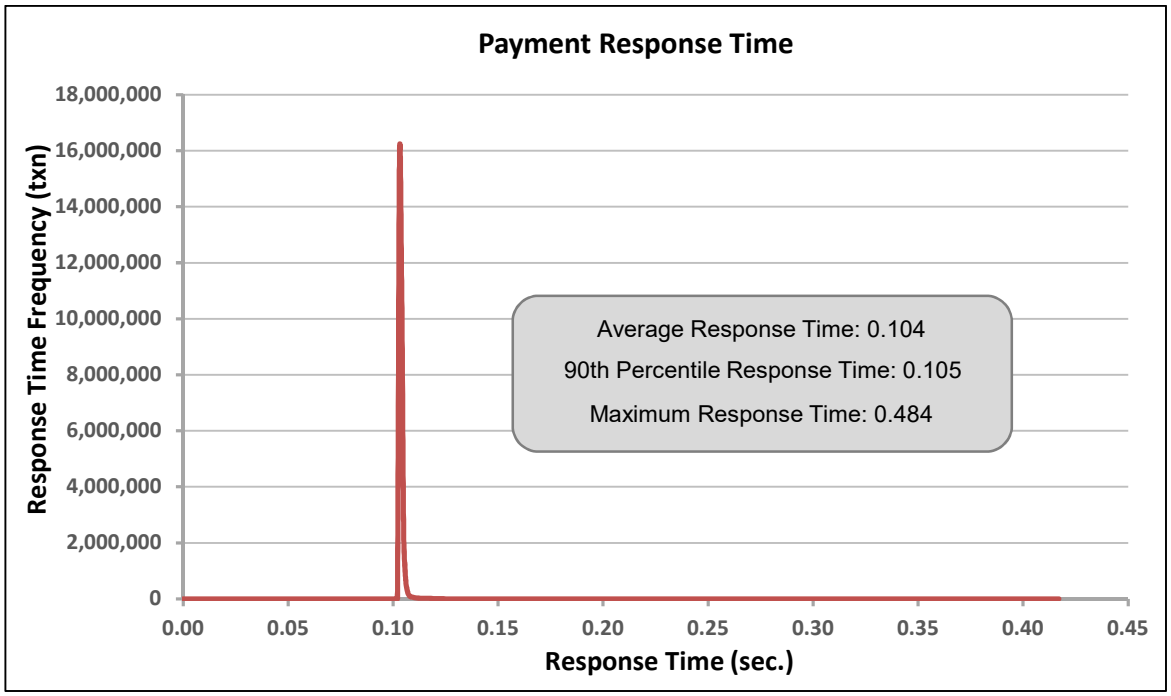



**Figure 5.4.1.2: Payment RT Frequency Distribution**

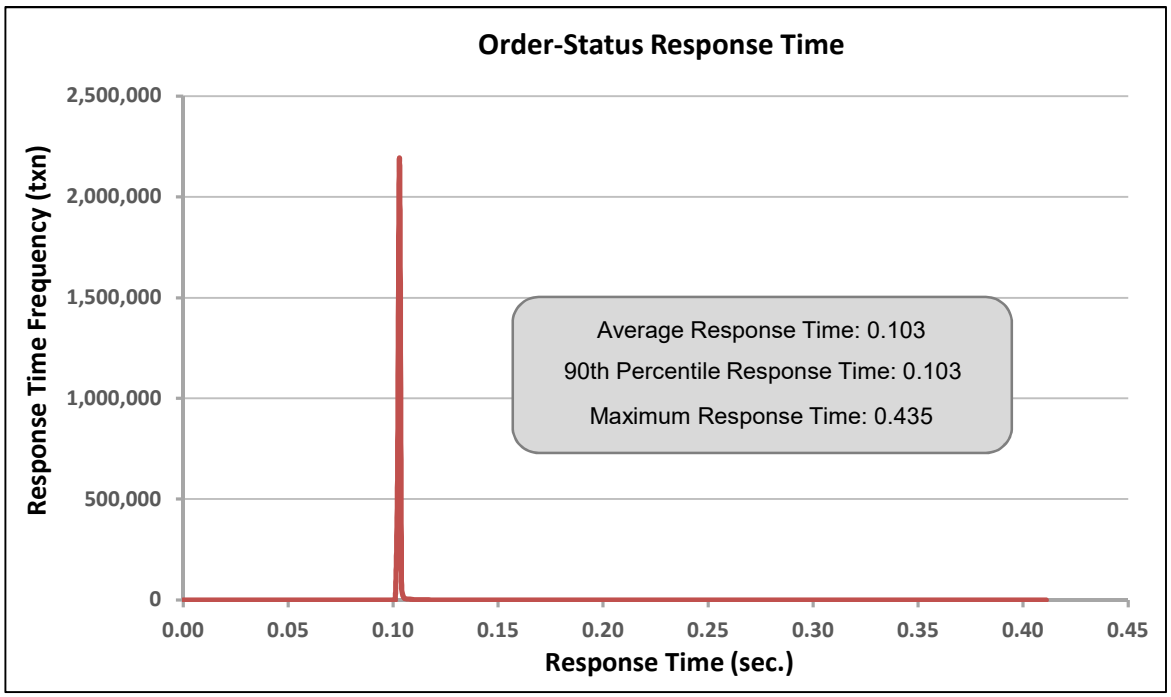



**Figure 5.4.1.3: Order-Status RT Frequency Distribution**

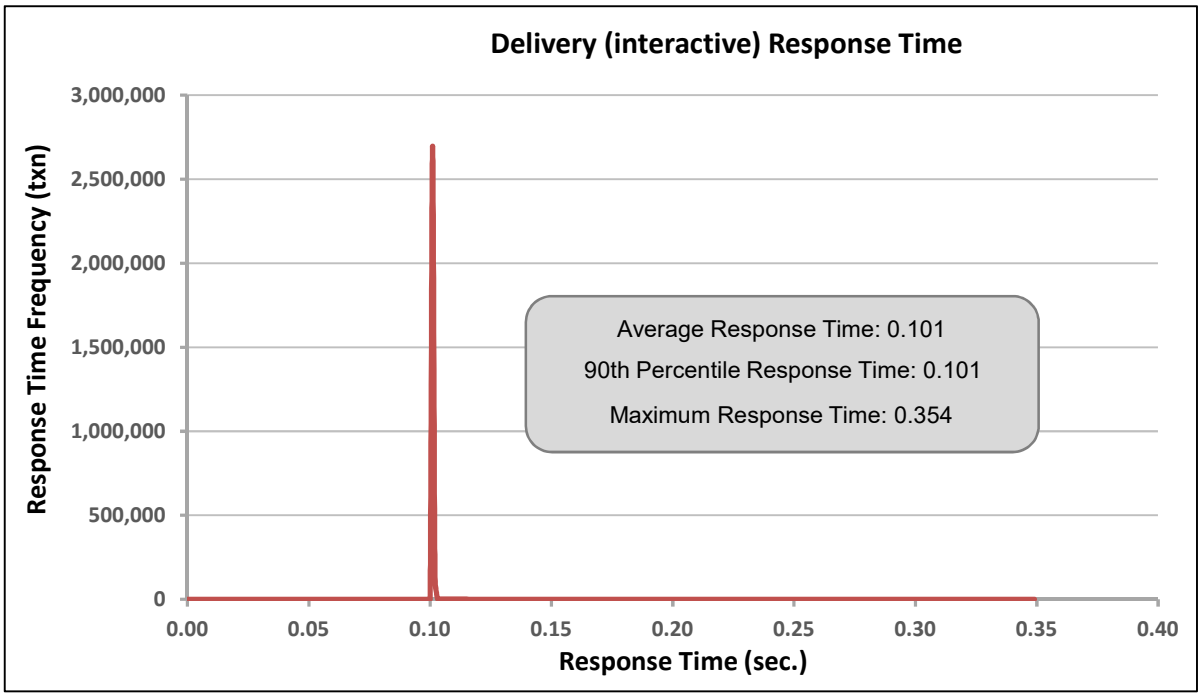



**Figure 5.4.1.4: Delivery (Interactive) RT Frequency Distribution**

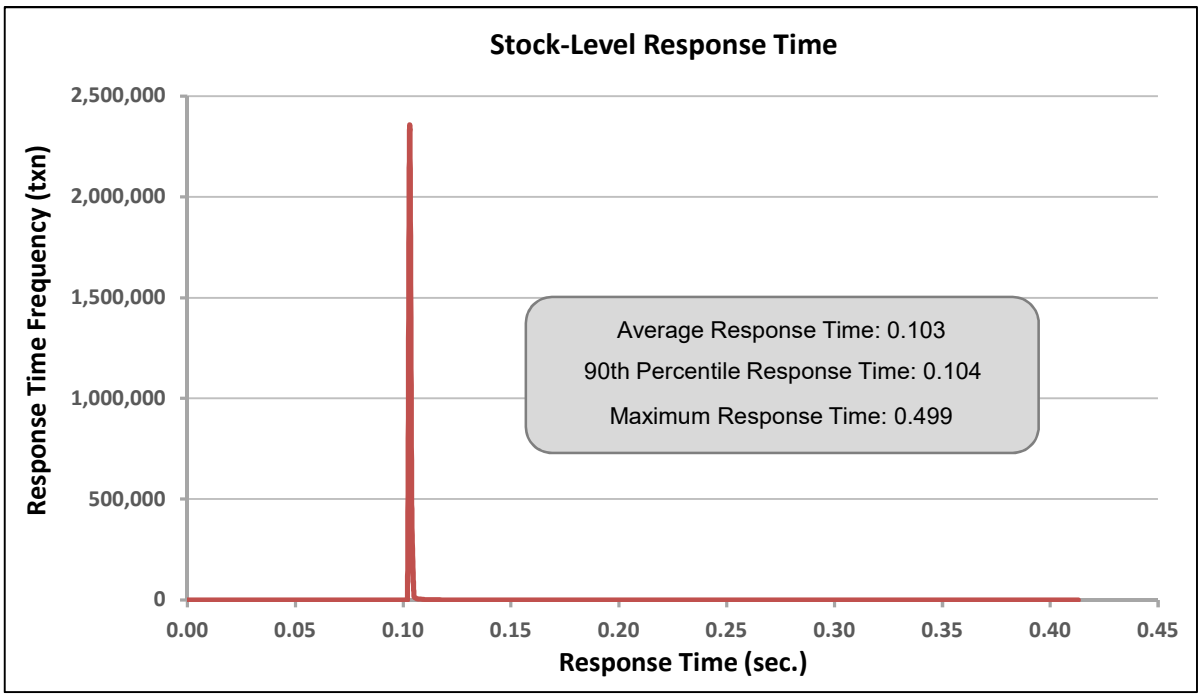



**Figure 5.4.1.5: Stock-Level RT Frequency Distribution**

### 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.

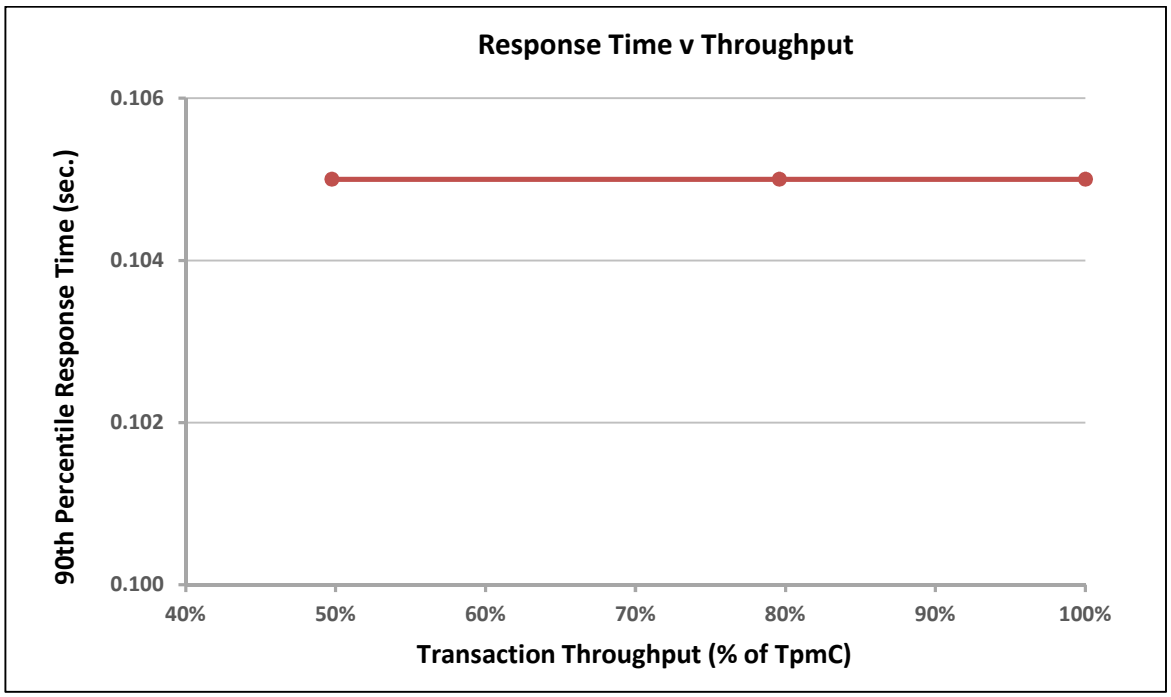



**Figure 5.4.2: New-Order RT versus Throughput**

### 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.

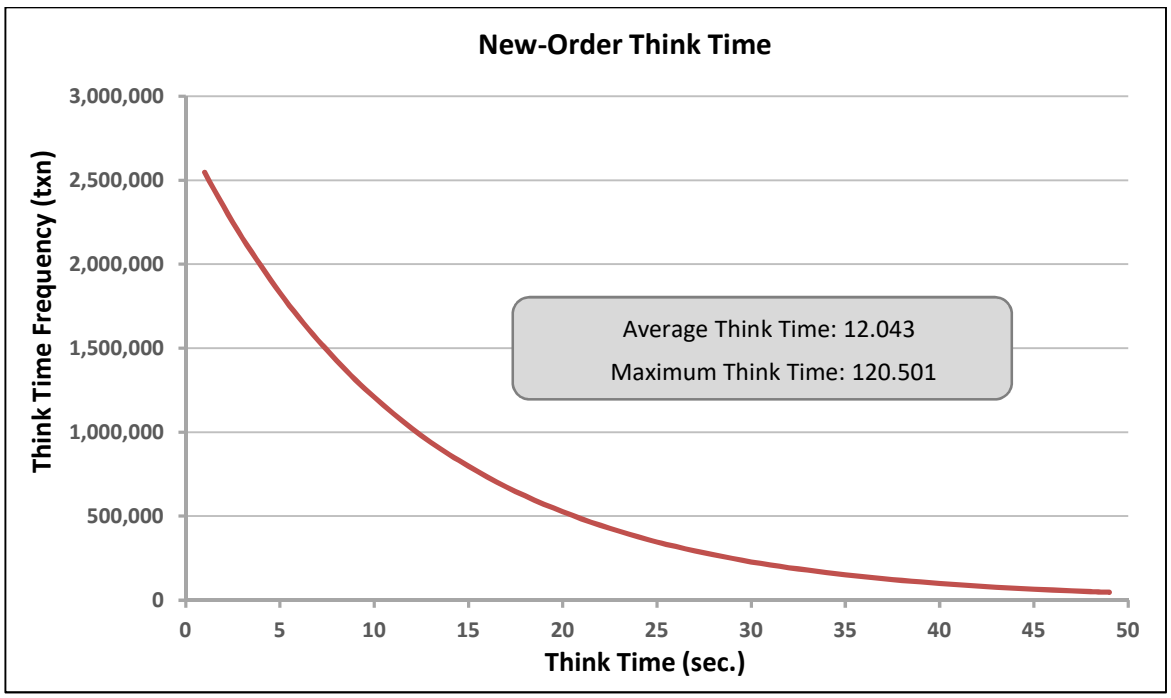



**Figure 5.4.3: New-Order Think Time Frequency Distribution**

### 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.

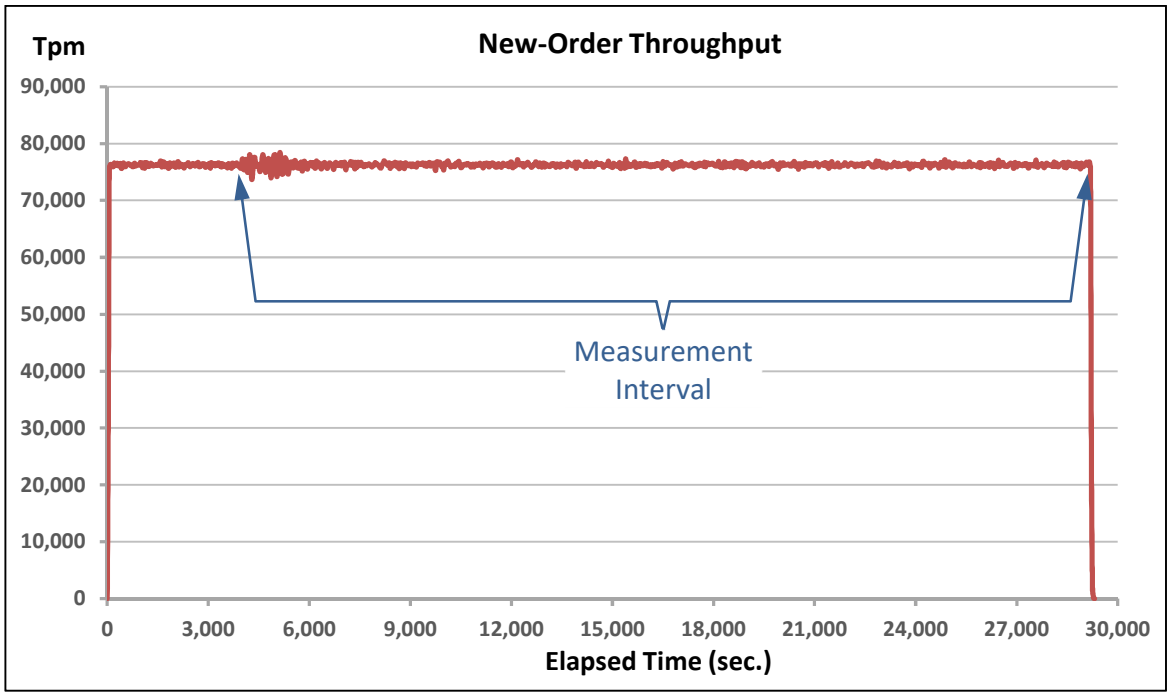



**Figure 5.4.4: New-Order Throughput versus Elapsed Time**

## 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 8 min(8:04).

## 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. 15 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 | 2020-09-14 12:42:21 | - | - |
| Checkpoint3 Begin | 2020-09-14 12:53:18 | | 00:27:59 |
| Checkpoint3 End | 2020-09-14 13:01:22 | 00:08:04 | |
| Checkpoint4 Begin | 2020-09-14 13:21:13 | | 00:27:55 |
| Checkpoint4 End | 2020-09-14 13:29:17 | 00:08:04 | |
| Checkpoint5 Begin | 2020-09-14 13:49:12 | | 00:27:59 |
| Checkpoint5 End | 2020-09-14 13:57:17 | 00:08:04 | |
| Checkpoint6 Begin | 2020-09-14 14:17:07 | | 00:27:55 |
| Checkpoint6 End | 2020-09-14 14:25:11 | 00:08:04 | |
| Checkpoint7 Begin | 2020-09-14 14:45:04 | | 00:27:56 |
| Checkpoint7 End | 2020-09-14 14:53:08 | 00:08:04 | |
| Checkpoint8 Begin | 2020-09-14 15:13:00 | | 00:27:57 |
| Checkpoint8 End | 2020-09-14 15:21:05 | 00:08:05 | |
| Checkpoint9 Begin | 2020-09-14 15:40:59 | | 00:27:59 |
| Checkpoint9 End | 2020-09-14 15:49:03 | 00:08:04 | |
| Checkpoint10 Begin | 2020-09-14 16:08:57 | | 00:27:58 |
| Checkpoint10 End | 2020-09-14 16:17:01 | 00:08:03 | |
| Checkpoint11 Begin | 2020-09-14 16:36:55 | | 00:27:57 |
| Checkpoint11 End | 2020-09-14 16:45:00 | 00:08:05 | |
| Checkpoint12 Begin | 2020-09-14 17:04:52 | | 00:27:58 |
| Checkpoint12 End | 2020-09-14 17:12:56 | 00:08:04 | |
| Checkpoint13 Begin | 2020-09-14 17:32:51 | | 00:27:58 |
| Checkpoint13 End | 2020-09-14 17:40:54 | 00:08:03 | |
| Checkpoint14 Begin | 2020-09-14 18:00:49 | | 00:27:58 |
| Checkpoint14 End | 2020-09-14 18:08:54 | 00:08:05 | |
| Checkpoint15 Begin | 2020-09-14 18:28:47 | | 00:27:58 |
| Checkpoint15 End | 2020-09-14 18:36:51 | 00:08:04 | |
| Checkpoint16 Begin | 2020-09-14 18:56:45 | | 00:27:58 |
| Checkpoint16 End | 2020-09-14 19:04:47 | 00:08:03 | |
| Checkpoint17 Begin | 2020-09-14 19:24:44 | | 00:27:59 |
| Checkpoint17 End | 2020-09-14 19:32:49 | 00:08:05 | |
| Measurement Interval End | 2020-09-14 19:45:16 | - | - |

# 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 (719) 473-7555

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

November 10, 2020

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

| | |
|---|---|
| Platform: | UNIWIDE RC2212 |
| Operating System: | Red Hat Enterprise Linux 7.8 |
| Database Manager: | Goldilocks v3.1 Standard Edition |

The results were:

| | |
|---|---|
| **Performance Metric** | **76,174 tpmC** |
| Number of Users | 60,000 |

| **Server** | **UNIWIDE RC2212** | | |
|---|---|---|---|
| CPUs | 2x Intel® Xeon® Silver 4210 (2.2 GHz, 10-core, 13.75 Cache) | | |
| Memory | 768 GB | | |
| Storage | ***Qty*** | ***Size*** | ***Type*** |
| | 2 | 1.2 TB | 12 G SAS 10K RPM HDD |
| | 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 = 20G
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
```

## 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.
-->
<!-- 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.mbeans.JmxRemoteLifecycleListener"
        rmiRegistryPortPlatform="9840" rmiServerPortPlatform="9841"/>
  <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 -->
  <Listener className="org.apache.catalina.core.AprLifecycleListener"
SSLEngine="on" />
  <!--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="localhost" 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>
```

## Sysctl fe.conf

```
# sysctl settings are defined through files in
# /usr/lib/sysctl.d/, /run/sysctl.d/, and /etc/sysctl.d/.
#
# Vendors settings live in /usr/lib/sysctl.d/.
# To override a whole file, create a new file with the same in
# /etc/sysctl.d/ and put new settings there. To override
# only specific settings, add a file with a lexically later
# name in /etc/sysctl.d/ and put new settings there.
#
# For more information, see sysctl.conf(5) and sysctl.d(5).


net.core.netdev_max_backlog=65535
net.core.somaxconn=65535
net.ipv4.tcp_tw_reuse=1
net.ipv6.conf.all.disable_ipv6=1
net.ipv6.conf.default.disable_ipv6=1
```

# Appendix C: Price Quotations

**DB Server**

UNIWIDE

## Quotation

견 적 일 자: UWT20 - 1012 - 1925
수 신: 한국정보통신기술협회 귀중
납 품 일 자: 협의
견적유효기간: 견적일로 부터 2주
결 제 조 건: 검수 후 현금결제
무상유지보수: 3Year → 3 year maintenance
기 타:

Amount : ₩9,230,000
Won(VAT Excluded)
Model : Uni RC2212(800Wx2)_3.5x12_RAID 2GB

* 견적유효기간 도과시 가격 변동이 있을 수 있습니다.


㈜유니와이드테크놀러지
서울시 구로구 디지털로26길5, 1404호
(에이스하이엔드타워 1차)
대표이사 : 우 종 운
TEL : 070-7306-0500
FAX : 02-866-0037

Sales Manager
담 당 자 : 김영철 기술연구소장
전 화 번 호 : 070-7306-0500
휴 대 전 화 : 010-8770-5483
이 메 일 : kkychel@uniwide.co.kr


| 품번 | 상세 내역 | 수 량 | 소비자단가 | 공급 단가 | 공급 금액 |
|---|---|---|---|---|---|
| SVR-UNI-RC2212-0010 | Uni RC2212(800Wx2)_3.5x12_RAID 2GB | 1 | | 9,230,000 | 9,230,000 |
| CPU-S4200-IN-0030 | Intel Xeon Silver 4210 10C/20T 2.2GHz 14MB 85W, SRFBL | 2 | | | |
| MEM-D4R26-SA-0030 | 32GB DDR4-2666 R ECC | 24 | | | |
| HDD-SAS0-RP-0050 | HDD SAS 12G 1.2TB 10K 2.5인치 512e | 2 | | | |
| HBA-16G2P-RP-0010 | FC Ctrl 16Gb/s 2CH QLE2692 | 1 | | | |
| LAN-10G2PR-IN-0010 | CARD LAN Intel 10G 2port UTP | 1 | | | |
| **Amount** | | | | | **₩9,230,000** |
| **V.A.T.** | | | | | **₩923,000** |
| **Amount(VAT Included)** | | | | | **₩10,153,000** |

| 명판 | 직인 |
|---|---|
| | |

**WAS Server**

# 견 적 서

(20201023-E047-Q001)

ATEC
에이텍시스템

2020년 10월 23일

주 식 회 사 에 이 텍 시 스 템
대 표 이 사 권 순 환
경기도 성남시 분당구 판교로 289

**한국정보통신기술협회 귀중**

폐사와의 거래에 감사드리오며, 아래와 같이 견적합니다.

| 품 명<br>DESCRIPTION | 규 격<br>SPECIFICATION | 단위<br>UNIT | 수량<br>Q'TY | 공 급 단 가<br>UNIT PRICE | 공 급 금 액<br>AMOUNT |
|---|---|---|---|---|---|
| 데스크톱 PC<br>A6HGBDPNN | 타입 : 타워형<br>CPU : XEON E3-1270V5, 3.6GHz<br>운영체제 : Windows 10 pro Workstations<br>메인메모리 : 16GB<br>하드디스크용량 : 1TB + 250GB(SDD)<br>그래픽 : Intel HD Graphics 530<br>메인보드 : Gigabyte X150M<br>USB포트 : USB 3.0 2EA / USB 2.0 2EA<br>LAN 규격 : 10/100/1000(Mbps)<br>제조사 무상유지보수 : 3년 / 7 x 24 x 4 Care Pack (3-yrs) | EA | 2 | 1,925,000 | 3,850,000 |
| 합 계 | | | | | ₩3,850,000 |

(단위 : 부가가치세 포함)

- 영 업 담 당 : 서버사업부 / 김상수 상무 / TEL:031) 698-8525 / FAX:031)698-8549 / HP:010-3009-7528 / sskim@atecsystem.kr

## 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 | 1 | 72,250,000 | ₩72,250,000 |
| 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 | | | |
| 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. | : 2020RP10-1303 | TERMS AND CONDITION | |
| DATE | : 2020. 10. 13. | | |
| COMPANY | : TTA | 납 기 | : 발주후 4주이내 |
| ATTN | : 최기한 선임연구원님 귀하 TEL : 031-780-9256 | 유지보수 | : 납품일로부터 1년 |
| Email | : kihanc@tta.or.kr | 결제조건 | : 익월말 현금 |
| FROM | : ㈜락플레이스 이왕모 과장 TEL : 010-9116-4680 | 유효기간 | : 견적일로부터 3개월 |

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 | | | | |
| RH00004F3 | Red Hat Enterprise Linux Server, Standard (Physical or Virtual Nodes) 3Year support :<br>Easy ISOs: OS, Source, Documentation ISO Images<br>가상화 Guest OS : 2guests<br>Red Hat Network 서비스 : 3년<br>Phone,email Support : 09:00 ~ 17:00<br>Scope of Coverage : Standard<br>Maximum Memory Support: Unlimited | 3 | 4,098,000 | 2,400,000 | 7,200,000 |
| 연간기술지원 | 연간 방문 기술지원 (옵션) | | | | |
| RP-CPS(OS) | rockPLACE Support Carepack - Linux Standard (3년) per Server<br>3 Year, 24x7, 4hr response<br>이메일, 전화, 원격지원, 현장지원 서비스<br>On Site Support - Total 연간 10회 Support (아래 지원내역에 준함)<br>- Installation & Startup Service Included<br>- Problem tracking/Emergency assistance<br>- Update, Patch 작업 지원<br>- 서비스, 시스템 환경, 네트워크 환경 설정 변경 지원<br>- 인수 시험, 성능 시험, 비상 복구 훈련 지원 | 3 | 6,000,000 | 2,000,000 | 6,000,000 |
| 소 계 금 액 | | | | | 13,200,000 |

| Part No. | Description | 수량 | 소비자가 | 공급단가 | 공급합계 |
|---|---|---|---|---|---|
| WEB | Red Hat JBoss Web Server | | | | |
| MW00123F3 | Red Hat JBoss Web Server, 4-Core Standard 3Year<br>- 전화/웹 지원 : 월-금, 9 a.m. - 5 p.m. 4시간내 응답<br>- unlimited incidents, | 2 | 2,144,000 | 1,260,000 | 2,520,000 |
| 연간기술지원 | 연간 방문 기술지원 (옵션) | | | | |
| RP-CPS(WAS) | rockPLACE Support Carepack - JBoss Standard (3년) per 4Core<br>3 Year, 24x7, 4hr response<br>이메일, 전화, 원격지원, 현장지원 서비스<br>On Site Support - Total 10회 Support (아래 지원내역에 준함)<br>- Installation & Startup Service Included<br>- Problem tracking/Emergency assistance<br>- Update, Patch 작업 지원<br>- 서비스, 시스템 환경, 네트워크 환경 설정 변경 지원<br>- 인수 시험, 성능 시험, 비상 복구 훈련 지원 | 2 | 12,000,000 | 6,000,000 | 12,000,000 |
| 소 계 금 액 | | | | | 14,520,000 |

| | |
|---|---|
| 합 계 | 27,720,000 |
| 부가세 | 2,772,000 |
| 합 계(부가세포함) | 30,492,000 |

*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 : 1,850,000 원 다량납품할인을 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 ~ 2023/08/23
납품기한 : 60일 (납품요구일로부터)
조달수수료여부 : 조달수수료 별도 조달수수료 안내·계산
첨부파일 : 2020/04/01_00176118214-물품구매(제조)계약일반조건(기재부계약예규415호20181231).hwp
2020/04/02_00176118214-다수공급자계약특수조건(조달청공고2018-137호,2018.12.12).hwp
2020/04/03_00176118214-규격서.zip
2020/04/04_00176118214-네트워크장비구축·운영사업추가특수조건.hwp
2020/04/05_00176118214-2019_08_03_00166171804-물품구매계약품질관리특수조건(20180524).hwp
대분류 : 03 - 전자.정보.통신
중분류 : 07 - 전산 및 통신용품
물품분류번호 : 43222612
세부품명번호 : 4322261201
물품식별번호 : 22917889
계약번호 : 00176118216-9
징수구분 : 후징수

**DBMS**

# Quotation

㈜TTA 貴中

Title : TPC-C Performance&Quality Authentication

참 조 : 황 세진 선임연구원
견적일자 : 2020년 10월 12일
유효기간 : 견적일로부터 4개월


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