TPC Benchmark™ H Full Disclosure Report

# Alibaba Cloud AnalyticDB

(with 576 Compute Instances)

using

## AnalyticDB for PostgreSQL 6.0

and

## Alibaba Group Enterprise Linux Server release 7.2 (Paladin)

**First Edition**
**May 20, 2020**

# First Edition – May 20, 2020

*Alibaba Cloud Elastic Compute Units*
*using*
*AnalyticDB for PostgreSQL 6.0*

# Abstract

## Overview

This report documents the methodology and results of the TPC Benchmark™ H test conducted on Alibaba Cloud ECU using AnalyticDB for PostgreSQL 6.0 in conformance with the requirements of the TPC Benchmark™ H Standard Specification, Revision 2.18.0. The operating system used for the benchmark was Alibaba Group Enterprise Linux Server release 7.2 (Paladin).

The TPC Benchmark™ H was developed by the Transaction Processing Performance Council (TPC). The TPC was founded to define transaction processing benchmarks and to disseminate objective, verifiable performance data to the industry.

TPC Benchmark™ H Full Disclosure Report and other information can be downloaded from the Transaction Processing Performance Council web site at www.tpc.org.

## Executive Summary Section

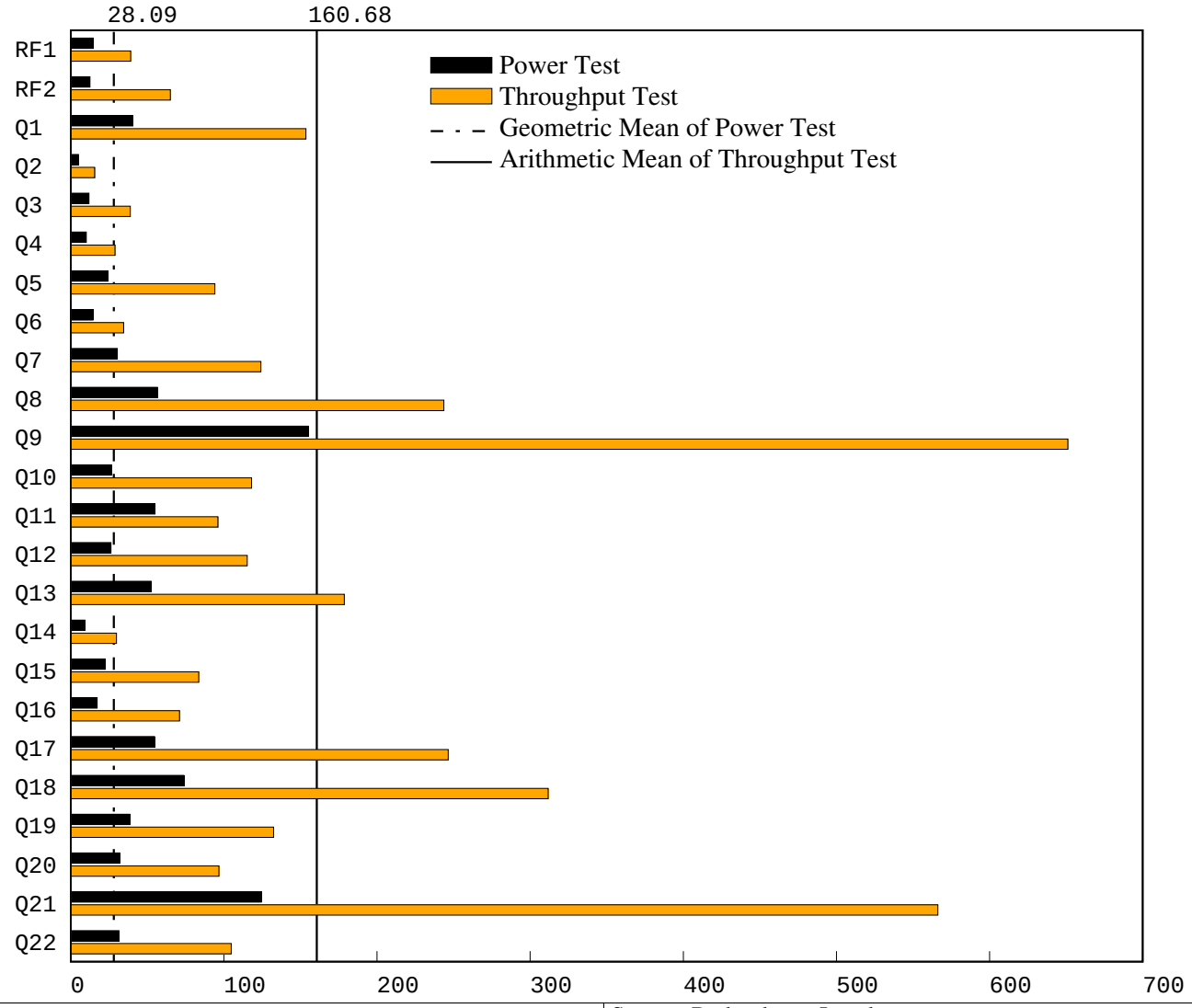
The first section of this report contains the Executive Summary and Numerical Quantities Summary of the benchmark results.

## Auditor

The benchmark configuration, environment and methodology used to produce and validate the test results, and the pricing model used to calculate the cost per QphH were audited by Francois Raab of InfoSizing to verify compliance with the relevant TPC specifications.

The auditor's information is available in Section 9.1.

| Alibaba Cloud | **Alibaba Cloud AnalyticDB** | | TPC-H Rev. 2.18.0<br>TPC-Pricing Rev. 2.5.0<br>Report Date<br>May 20, 2020 | |
|---|---|---|---|---|
| Total System Cost | Composite Query per Hour Metric | | Price / Performance | |
| **¥7,343,841 RMB** | **5,057,263**<br>QphH@30000GB | | **¥1.46 RMB**<br>RMB/QphH@30000GB | |
| Database Size | DB Manager | OS | Other Software | Available Date |
| 30,000GB | AnalyticDB for PostgreSQL 6.0 | AliOS 7.2 | None | As of publication |




| | | | |
|---|---|---|---|
| Database Load Time | 09h 42m 43s | Storage Redundancy Levels: | |
| Load Includes Backup: | N | Base Tables | Level One |
| Total Data Storage/Database Size | 1.86 | Auxiliary Data Structures | Level One |
| Memory to Database Size Percentage | 93.4% | DBMS Temporary Space | Level One |
| | | OS and DBMS Software | Level One |

**System Configuration**

| System Component | Per Node (Master) | System Total (Master) | Per Node (Compute) | System Total (Compute) |
|---|---|---|---|---|
| Node: | 1 | 2 | 1 | 576 |
| vCPU: | 32 vCPUs | 64 vCPUs | 6 vCPUs | 3,456 vCPUs |
| Memory: | 192 GB | 384 GB | 48 GB | 27,648 GB |
| Storage: | 320 GB | 640 GB | 96 GB | 55,296 GB |

**Total Storage:** 55,936 GB

# Alibaba Cloud AnalyticDB

TPC-H Rev. 2.18.0
TPC-Pricing Rev. 2.5.0

Report Date
May 20, 2020

| Desciption | Part Num. | Src | Unit Price (RMB) | Qty | Ext.Price (RMB) | 3-Year Maint. (RMB) |
|---|---|---|---|---|---|---|
| **Licence Computer and Software Services** | | | | | | |
| AnalyticDB for PostgreSQL 6.0 (Standard Edition, 576 nodes, 6 cores per node, 3-years pre-pay) | North China 5 | 1 | 7,340,544.00 | 1 | 7,340,544.00 | included |
| Compute Node (each with) | | | included | 576 | | |
| - 6 vCPUs | | | | | | |
| - 48 GB memory | | | | | | |
| - 96 GB SSD storage | | | | | | |
| Master Instance (each with) | | | included | 2 | | |
| - 32 vCPUs | | | | | | |
| - 192GB memory | | | | | | |
| - 320GB SSD storage | | | | | | |
| | **Licence Computer and Software Services Sub-Total** | | | | 7,340,544.00 | 0.00 |
| **Other Components** | | | | | | |
| Lenovo MIIX 210 Laptop (Includes spares) | | 2 | 1,099.00 | 3 | 3,297.00 | |
| | **Other Components Sub-Total** | | | | **3,297.00** | 0.00 |

| | | |
|---|---|---|
| 1 = Alibaba Cloud, 2 = Tmall.com | **3-Year Cost Of Ownership** | **7,343,841.00** |
| All prices are based on 3-year pre-paid subscriptions. | **QphH@30000GB** | **5,057,263** |
| **Audited by Francois Raab, InfoSizing** | **RMB/QphH@30000GB** | **1.46** |

*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 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.*

# Numeric Quantities

## Measurement Results

| | |
|---|---|
| Database Scale Factor | 30,000 GB |
| Total Data Storage / Database Size | 1.86 |
| Percentage Memory / Database Size | 93.4% |
| Start of Database Load | 2020-05-15 18:37:46 |
| End of Database Load | 2020-05-16 04:20:29 |
| Database Load Time | 00d 09h 42m 43s |
| Query Streams for Throughput Test | 10 |
| TPC-H Power | 3,844,697.0 |
| TPC-H Throughput | 6,652,257.2 |
| TPC-H Composite Query-per-Hour Metric (QphH@30000GB) | 5,057,263.4 |
| Total System Price Over 3 Years | ¥7,343,841.00 (RMB) |
| TPC-H Price/ Performance Metric ($/QphH@30000GB) | ¥1.46 (RMB) |

## Measurement Interval

| | |
|---|---|
| Measurement Interval in Throughput Test | 3,571.72 seconds |

## Duration of Stream Execution

| Power Run | Seed | Query Start Time<br>Query End Time | Total Time<br>(hh:mm:ss) | RF1 Start Time<br>RF1 End Time | RF2 Start Time<br>RF2 End Time |
|---|---|---|---|---|---|
| | 516042029 | 2020-05-16 05:36:04<br>2020-05-16 05:51:17 | 00:15:13 | 2020-05-16 05:35:50<br>2020-05-16 05:36:04 | 2020-05-16 05:51:17<br>2020-05-16 05:51:30 |
| **Throughput Stream** | **Seed** | **Query Start Time<br>Query End Time** | **Total Time<br>(hh:mm:ss)** | **RF1 Start Time<br>RF1 End Time** | **RF2 Start Time<br>RF2 End Time** |
| **1** | 516042030 | 2020-05-16 05:51:29<br>2020-05-16 06:50:29 | 00:59:00 | 2020-05-16 05:51:29<br>2020-05-16 05:52:15 | 2020-05-16 05:52:15<br>2020-05-16 05:53:16 |
| **2** | 516042031 | 2020-05-16 05:51:29<br>2020-05-16 06:50:01 | 00:58:32 | 2020-05-16 05:53:16<br>2020-05-16 05:53:55 | 2020-05-16 05:53:55<br>2020-05-16 05:54:47 |
| **3** | 516042032 | 2020-05-16 05:51:29<br>2020-05-16 06:50:19 | 00:58:50 | 2020-05-16 05:54:47<br>2020-05-16 05:55:22 | 2020-05-16 05:55:22<br>2020-05-16 05:56:15 |
| **4** | 516042033 | 2020-05-16 05:51:29<br>2020-05-16 06:50:27 | 00:58:58 | 2020-05-16 05:56:15<br>2020-05-16 05:56:55 | 2020-05-16 05:56:55<br>2020-05-16 05:57:52 |
| **5** | 516042034 | 2020-05-16 05:51:29<br>2020-05-16 06:50:17 | 00:58:48 | 2020-05-16 05:57:52<br>2020-05-16 05:58:34 | 2020-05-16 05:58:34<br>2020-05-16 05:59:39 |
| **6** | 516042035 | 2020-05-16 05:51:29<br>2020-05-16 06:50:13 | 00:58:44 | 2020-05-16 05:59:39<br>2020-05-16 06:00:13 | 2020-05-16 06:00:13<br>2020-05-16 06:01:29 |
| **7** | 516042036 | 2020-05-16 05:51:29<br>2020-05-16 06:50:21 | 00:58:52 | 2020-05-16 06:01:29<br>2020-05-16 06:02:00 | 2020-05-16 06:02:00<br>2020-05-16 06:03:12 |
| **8** | 516042037 | 2020-05-16 05:51:29<br>2020-05-16 06:51:02 | 00:59:33 | 2020-05-16 06:03:12<br>2020-05-16 06:03:48 | 2020-05-16 06:03:48<br>2020-05-16 06:04:55 |
| **9** | 516042038 | 2020-05-16 05:51:29<br>2020-05-16 06:50:41 | 00:59:12 | 2020-05-16 06:04:55<br>2020-05-16 06:05:35 | 2020-05-16 06:05:35<br>2020-05-16 06:06:42 |
| **10** | 516042039 | 2020-05-16 05:51:29<br>2020-05-16 06:50:23 | 00:58:54 | 2020-05-16 06:06:42<br>2020-05-16 06:07:30 | 2020-05-16 06:07:30<br>2020-05-16 06:08:51 |

## TPC-H Timing Intervals (in seconds)

| Stream ID | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 | Q11 | Q12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 40.57 | 5.00 | 11.89 | 10.10 | 24.16 | 14.49 | 30.38 | 56.57 | 155.04 | 26.41 | 55.05 | 26.30 |
| **1** | 97.79 | 19.30 | 50.35 | 8.39 | 126.32 | 41.05 | 106.92 | 255.26 | 645.43 | 134.43 | 104.79 | 155.52 |
| **2** | 135.68 | 16.50 | 46.13 | 32.76 | 102.31 | 36.96 | 112.26 | 297.40 | 662.80 | 103.67 | 93.79 | 81.80 |
| **3** | 176.77 | 14.89 | 20.42 | 25.17 | 86.56 | 35.74 | 133.04 | 209.17 | 627.32 | 121.03 | 98.78 | 73.39 |
| **4** | 182.32 | 14.90 | 26.44 | 43.10 | 102.31 | 41.44 | 124.12 | 275.29 | 637.87 | 96.25 | 101.93 | 93.72 |
| **5** | 120.10 | 16.20 | 46.70 | 23.79 | 102.88 | 39.05 | 132.52 | 294.11 | 599.96 | 133.88 | 95.46 | 92.87 |
| **6** | 188.03 | 13.42 | 38.17 | 41.38 | 106.30 | 32.42 | 164.61 | 234.11 | 677.29 | 109.76 | 101.32 | 121.43 |
| **7** | 167.83 | 15.55 | 35.33 | 22.44 | 112.18 | 30.87 | 139.35 | 211.73 | 672.42 | 116.79 | 96.79 | 59.01 |
| **8** | 135.05 | 12.93 | 36.13 | 30.27 | 104.53 | 36.58 | 91.79 | 249.12 | 726.93 | 117.72 | 71.84 | 160.36 |
| **9** | 143.31 | 16.14 | 47.66 | 24.85 | 51.11 | 26.78 | 95.07 | 207.61 | 646.39 | 121.29 | 94.84 | 137.47 |
| **10** | 188.66 | 16.37 | 41.00 | 36.86 | 46.86 | 24.93 | 141.81 | 202.70 | 614.52 | 125.23 | 101.20 | 177.16 |
| **Qi Min** | 40.57 | 5.00 | 11.89 | 8.39 | 24.16 | 14.49 | 30.38 | 56.57 | 155.04 | 26.41 | 55.05 | 26.30 |
| **Qi Avg** | 143.28 | 14.65 | 36.38 | 27.19 | 87.77 | 32.76 | 115.62 | 226.64 | 606.00 | 109.68 | 92.34 | 107.18 |
| **Qi Max** | 188.66 | 19.30 | 50.35 | 43.10 | 126.32 | 41.44 | 164.61 | 297.40 | 726.93 | 134.43 | 104.79 | 177.16 |

| Stream ID | Q13 | Q14 | Q15 | Q16 | Q17 | Q18 | Q19 | Q20 | Q21 | Q22 | RF1 | RF2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 52.51 | 9.27 | 22.58 | 17.20 | 54.64 | 73.88 | 38.58 | 31.75 | 124.35 | 31.53 | 14.57 | 12.56 |
| **1** | 176.43 | 24.70 | 72.42 | 53.47 | 264.81 | 356.68 | 107.56 | 120.55 | 526.59 | 89.80 | 45.13 | 61.15 |
| **2** | 197.02 | 27.12 | 87.94 | 71.46 | 203.31 | 329.10 | 126.02 | 91.93 | 553.46 | 101.55 | 38.72 | 52.18 |
| **3** | 185.20 | 36.77 | 110.95 | 68.08 | 250.36 | 349.95 | 154.13 | 112.55 | 535.63 | 103.33 | 35.14 | 52.83 |
| **4** | 203.70 | 30.78 | 86.54 | 91.68 | 278.05 | 286.13 | 146.48 | 59.51 | 538.59 | 75.21 | 39.41 | 57.14 |
| **5** | 178.67 | 32.46 | 99.61 | 82.43 | 195.50 | 337.02 | 161.32 | 97.38 | 516.78 | 128.48 | 42.62 | 64.26 |
| **6** | 161.16 | 18.96 | 66.10 | 66.99 | 251.34 | 284.06 | 114.79 | 80.89 | 542.78 | 107.52 | 34.55 | 75.61 |
| **7** | 180.97 | 22.16 | 54.80 | 81.26 | 278.75 | 284.41 | 104.93 | 100.00 | 636.71 | 106.73 | 31.30 | 71.41 |
| **8** | 149.20 | 42.63 | 70.74 | 91.96 | 235.19 | 272.68 | 125.89 | 97.71 | 587.71 | 124.62 | 36.61 | 67.03 |
| **9** | 160.89 | 23.25 | 109.49 | 18.35 | 260.42 | 343.88 | 129.92 | 104.86 | 659.69 | 127.18 | 39.71 | 66.97 |
| **10** | 192.98 | 39.10 | 76.27 | 83.93 | 246.97 | 273.34 | 153.04 | 101.53 | 564.78 | 83.38 | 47.57 | 81.62 |
| **Qi Min** | 52.51 | 9.27 | 22.58 | 17.20 | 54.64 | 73.88 | 38.58 | 31.75 | 124.35 | 31.53 | 14.57 | 12.56 |
| **Qi Avg** | 167.16 | 27.93 | 77.95 | 66.07 | 229.03 | 290.10 | 123.88 | 90.79 | 526.10 | 98.12 | 36.85 | 60.25 |
| **Qi Max** | 203.70 | 42.63 | 110.95 | 91.96 | 278.75 | 356.68 | 161.32 | 120.55 | 659.69 | 128.48 | 47.57 | 81.62 |

# Table of Contents


**TPC BENCHMARK H OVERVIEW** **1**

**0 General Items** **3**
- 0.1 Benchmark Sponsor 3
- 0.2 Parameter Settings 3
- 0.3 Configuration Diagram 3

**1 Clause 1: Logical Database Design** **5**
- 1.1 Database Definition Statements 5
- 1.2 Physical Organization 5
- 1.3 Horizontal Partitioning 5
- 1.4 Replication 5

**2 Clause 2: Queries and Refresh Functions Related Items** **6**
- 2.1 Query Language 6
- 2.2 Verifying Method of Random Number Generation 6
- 2.3 Query Text and Output Data from Qualification Database 6
- 2.4 Query Substitution Parameters and Seeds Used 6
- 2.5 Isolation Level 6
- 2.6 Source Code of Refresh Functions 6

**3 Clause 3: Database System Properties** **7**
- 3.1 ACID Properties 7
- 3.2 Atomicity Requirements 7
  - 3.2.1 Atomicity of the Completed Transactions 7
  - 3.2.2 Atomicity of Aborted Transactions 7
- 3.3 Consistency Requirements 7
  - 3.3.1 Consistency Test 8
- 3.4 Isolation Requirements 8
  - 3.4.1 Isolation Test 1 – Read-Write Conflict with Commit 8
  - 3.4.2 Isolation Test 2 – Read-Write Conflict with Rollback 8
  - 3.4.3 Isolation Test 3 – Write-Write Conflict with Commit 9
  - 3.4.4 Isolation Test 4 – Write-Write Conflict with Rollback 9
  - 3.4.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables 9
  - 3.4.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream 9
- 3.5 Durability Requirements 10
  - 3.5.1 Permanent Unrecoverable Failure of Any Durable Medium 10
  - 3.5.2 System Crash 10
  - 3.5.3 Memory Failure 10
  - 3.5.4 Loss of External Power 10
  - 3.5.5 Node or Controller Failure 11

**4 Clause 4: Scaling and Database Population** **12**
- 4.1 Initial Cardinality of Tables 12
- 4.2 Distribution of Tables and Logs Across Media 12
- 4.3 Mapping of Database Partitions/Replication 12
- 4.4 Implementation of Data Redundancy 13
- 4.5 DBGen Modifications 13
- 4.6 Database Load Time 13
- 4.7 Data Storage Ratio 13
- 4.8 Database Load Mechanism Details and Illustration 14
- 4.9 Qualification Database Configuration 14
- 4.10 Memory to Database Size Percentage 14

**5 Clause 5: Performance Metrics and Execution Rules Related Items** **15**
- 5.1 System Activity between Load and Performance Tests 15
- 5.2 Steps in the Power Test 15
- 5.3 Timing Interval for Each Query and Refresh Functions 15

5.4 Number of Streams for the Throughput Test 15
5.5 Start and End Date/Time of Each Query Stream 15
5.6 Total Elapsed Time of the Measurement Interval 15
5.7 Refresh Function Start Date/Time and Finish Date/Time 15
5.8 Timing Metric Precision 16
5.9 Performance Metrics 16
5.10 The Performance Metric and Numerical Quantities from Both Runs 16
5.11 System Activity between Performance Tests 16
5.12 Documentation to satisfy Clause 5.2.7 16
5.13 Query Output Validation 16

**6 Clause 6: SUT and Driver Implementation Related Items 17**
6.1 Driver 17
6.2 Implementation Specific Layer (ISL) 17
6.3 Profile-Directed Optimization 17

**7 Clause 7: Pricing 18**
7.1 Priced Configuration 18
7.2 Availability Date and Orderability Date 18
7.3 Country-Specific Pricing 18

**8 Clause 8: Full Disclosure 19**
8.1 Supporting Files Index Table 19

**9 Clause 9: Audit Related Items 20**
9.1 Auditor's Report 20

**Appendix A: Price Quotes 23**

# TPC BENCHMARK H OVERVIEW

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

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

TPC-H evaluates the performance of various decision support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions;
- Simulate generated ad-hoc queries (e.g., via a point and click GUI interface);
- Are far more complex than most OLTP transactions;
- Include a rich breadth of operators and selectivity constraints;
- Generate intensive activity on the part of the database server component of the system under test;
- Are executed against a database complying to specific population and scaling requirements;
- Are implemented with constraints derived from staying closely synchronized with an on-line production database.

The TPC-H operations are modeled as follows:

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

The performance metric reported by TPC-H is called the TPC-H Composite Query-per-Hour Performance Metric (QphH@Size), and reflects multiple aspects of the capability of the system to process queries. These aspects include the selected database size against which the queries are executed, the query processing power when queries are submitted by a single stream and the query throughput when queries are submitted by multiple concurrent users. The TPC-H Price/Performance metric is expressed as $/QphH@Size. To be compliant with the TPC-H standard, all references to TPC-H results for a given configuration must include all required reporting components. The TPC believes that comparisons of TPC-H results measured against different database sizes are misleading and discourages such comparisons.

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

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

Despite the fact that this benchmark offers a rich environment representative of many decision support systems, this benchmark does not reflect the entire range of decision support requirements. In addition, the extent to which a customer can achieve the results reported by a vendor is highly dependent on how closely TPC-H approximates the customer application. The relative performance of systems derived from this benchmark does not necessarily hold for other workloads or environments. Extrapolations to any other environment are not recommended.

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

Further information is available at www.tpc.org.

# 0. General Items

## 0.1 Benchmark Sponsor

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

This benchmark is sponsored by Alibaba Cloud Computing Ltd..

## 0.2 Parameter Settings

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

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

*In the event that some parameters and options are set multiple times, it must be easily discernible by an interested reader when the parameter or option was modified and what new value it received each time.*

*This requirement can be satisfied by providing a full list of all parameters and options, as long as all those that have been modified from their default values have been clearly identified and these parameters and options are only set once.*

The Supporting File Archive contains the Operating System and DBMS parameters used in this benchmark.

## 0.3 Configuration Diagram

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

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

The system diagram of the measured system is depicted in Figure 4 and the configuration of the measured systems is shown in Table 1. The system is composed of 2 master instances and 576 compute instances. There is no difference between the priced and measured configurations.

**Table 1.** Configuration of the measured system.

| **Configuration** | **Master Instance** | **Compute Instance** |
|---|---|---|
| **#Instances** | 2 | 576 |
| **Per Instance Config** | 32 vCPUs | 6 vCPUs |
| | 192 GB RAM | 48GB RAM |
| | 320 GB SSD storage | 96 GB SSD storage |
| **Network** | 25 Gbit/s | 25 Gbit/s |

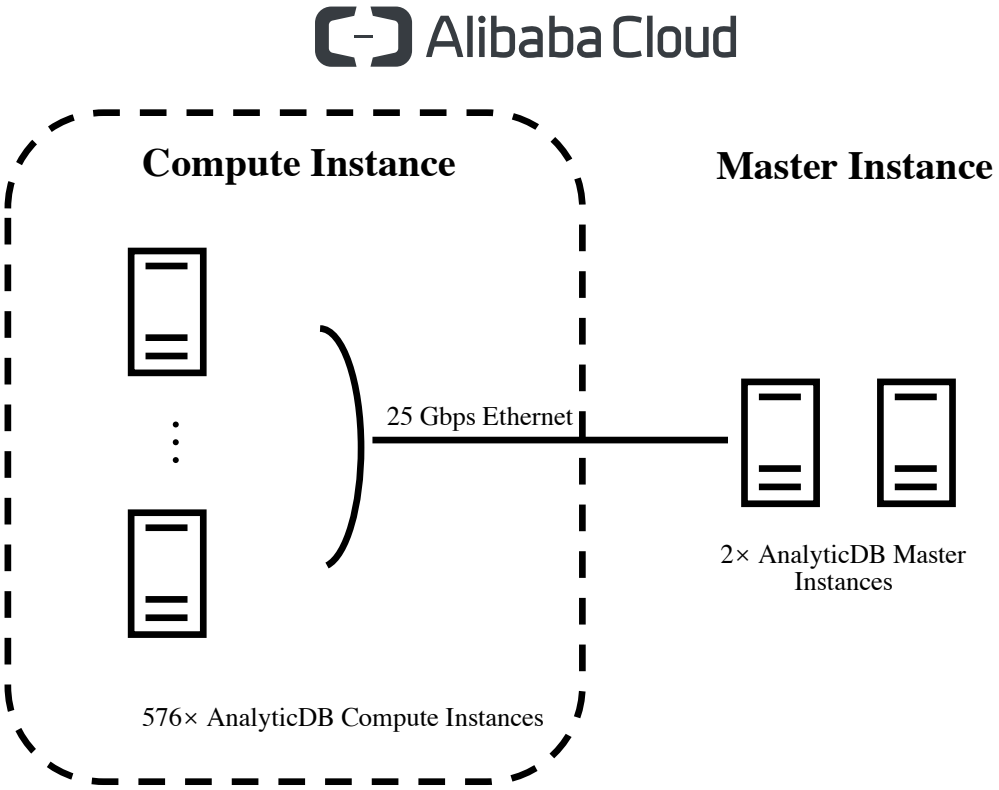



**Fig. 1.** System diagram of the measured configuration.

# 1. Clause 1: Logical Database Design

## 1.1 Database Definition Statements

*Listings must be provided for all table definition statements and all other statements used to set-up the test and qualification databases. All listings must be reported in the supporting files archive.*

The Supporting File Archive contains the table definitions and all other statements to set up the tables used in this benchmarking.

## 1.2 Physical Organization

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

*The concept of physical organization includes, but is not limited to: record clustering (i.e., rows from different logical tables are co-located on the same physical data page), index clustering (i.e., rows and leaf nodes of an index to these rows are co-located on the same physical data page), and partial fill-factors (i.e., physical data pages are left partially empty even though additional rows are available to fill them).*

Physical organization requires no user input. All the database data is placed on the same partition.

## 1.3 Horizontal Partitioning

*Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed. Scripts to perform horizontal partitioning must be reported in the supporting files archive.*

The tested database consists of 576 data segments. The mapping between data segments and compute instances is managed by the DBMS automatically. Tables are horizontally partitioned and distributed on data segments using a hash algorithm. The columns used for the hashing (e.g., distribution key) are controlled by DDL statements (see Supporting Files Archive, keyword: *distributed by*).

## 1.4 Replication

*Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.7. Scripts to perform any replication must be reported in the supporting files archive.*

Each data segment is configured with one primary segment and one mirror segment. Segments are evenly distributed over the compute instances. One primary segment and one mirror segment share the cores, memory, and network resources of one compute instance in an on-demand manner, while the storage resources are evenly partitioned. It is guaranteed that primary segments do not reside on the same physical machine with their mirror segments. A mirror segment is switched to primary when the corresponding primary segment fails. The data replication mechanism is managed by the DBMS automatically and is transparent to all data manipulation operation. All ACID properties are maintained and updates are reflected in all mirror segments by the time the updating transaction is committed.

# 2. Clause 2: Queries and Refresh Functions Related Items

## 2.1 Query Language

*The query language used to implement the queries must be identified.*

SQL was the query language used to implement the queries.

## 2.2 Verifying Method of Random Number Generation

*The version number, release number, modification number, and patch level of QGen must be disclosed. Any modifications to the QGen source code must be reported in the supporting files archive.*

TPC-supplied DBGen version 2.18.0 and QGen version 2.18.0 were used.

## 2.3 Query Text and Output Data from Qualification Database

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

The actual query text and query output are included in the Supporting Files Archive. The standard queries were used throughout with the following modifications:

- LIMIT syntax used to restrict the number of output rows (Q2, Q3, Q10, Q18, Q21).

## 2.4 Query Substitution Parameters and Seeds Used

*All the query substitution parameters used during the performance test must be disclosed in tabular format, along with the seeds used to generate these parameters.*

The Supporting Files Archive contains the seed and query substitution parameters.

## 2.5 Isolation Level

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

The queries and transactions were run with isolation level 3.

## 2.6 Source Code of Refresh Functions

*The details of how the refresh functions were implemented must be disclosed (including source code of any non-commercial program used).*

Supporting Files Archive contains the Source Code of the refresh functions.

# 3. Clause 3: Database System Properties

*The results of the ACID tests must be disclosed along with a description of how the ACID requirements were met. All code (including queries, stored procedures etc.) used to test the ACID requirements and their entire output must be reported in the supporting files archive.*

The results of the ACID tests are disclosed as required.

## 3.1 ACID Properties

*The ACID (Atomicity, Consistency, Isolation, and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.*

All ACID tests were conducted according to the specifications. The Supporting Files Archive contains the source code of the ACID test scripts.

## 3.2 Atomicity Requirements

*The system under test must guarantee that 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.*

### 3.2.1 Atomicity of the Completed Transactions

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

The following operations were performed to verify the atomicity of the completed transactions:

1. Randomly selected one order key, and then retrieved the total price from table ORDERS and the extended price from table LINEITEM.
2. One transaction was performed using the selected order key.
3. The transaction was committed.
4. Retrieved the total price from the table ORDERS and the extended price from the table LINEITEM for the order key.
5. It was verified that the rows had been changed.

### 3.2.2 Atomicity of Aborted Transactions

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

The following operations were performed to verify the atomicity of the aborted transactions:

1. Randomly selected one order key, and then retrieved the total price from table ORDERS and the extended price from table LINEITEM.
2. One transaction was performed using the selected order key and the transaction was stopped before commit.
3. The transaction was rolled back.
4. Retrieved the total price from the table ORDERS and the extended price from the table LINEITEM for the order key.
5. It was verified that the rows had not been changed.

## 3.3 Consistency Requirements

*Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another. A consistent state for the TPC-H database is defined to exist when:*

```
O_TOTALPRICE = SUM(trunc(trunc(L_EXTENDEDPRICE*(1 - L_DISCOUNT), 2) * (1 + L_TAX), 2))
For each ORDER and LINEITEM defined by (O_ORDERKEY = L_ORDERKEY).
```

### 3.3.1 Consistency Test

*Verify that ORDERS and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDERS and LINEITEM.*

The following query was executed before and after the consistency tests to demonstrate that the database kept staying in a consistent state both initially and after submitting transactions:

```
SELECT *
FROM (SELECT o_orderkey,
             (o_totalprice -
             SUM(trunc ((trunc ((l_extendedprice*(1 - l_discount))::text::NUMERIC,2)
             *(1 + l_tax))::text::NUMERIC,2)))::text::NUMERIC part_res
      FROM orders,
           lineitem
      WHERE o_orderkey = l_orderkey
      GROUP BY o_orderkey,
               o_totalprice) AS temp
WHERE NOT part_res = 0
limit 10;
```

The following operations were performed to verify the consistency of ACID transactions:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. For each of the 10 execution streams, 100 transactions were prepared.
3. For all 10 execution streams, the prepared ACID transactions were executed.
4. The consistency of the ORDERS and LINEITEM tables was verified again.

## 3.4 Isolation Requirements

*Operations of concurrent transactions must yield results, which are indistinguishable from the results, which would be obtained by forcing each transaction to be serially executed to completion in some order.*

The steps of the isolation tests were adapted to the AnalyticDB for PostgreSQL isolation environment.

### 3.4.1 Isolation Test 1 – Read-Write Conflict with Commit

*Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the readwrite transaction is committed*

The following operations were performed to verify the isolation for a read-only and a read-write committed transaction:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.
3. Started the same query and verified that the row has not changed.
4. Committed the update transaction.
5. Started a query and verified that the new row was retrieved.

### 3.4.2 Isolation Test 2 – Read-Write Conflict with Rollback

*Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the readwrite transaction is rolled back.*

The following operations were performed to verify the isolation for a read-only and a rolled back read-write transaction:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.
3. Started the same query and verified that the row has not changed.
4. Rolled back the update transaction.
5. Started a query and verified that the old row was retrieved.

#### 3.4.3 Isolation Test 3 – Write-Write Conflict with Commit

*Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.*

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

1. Started an update transaction T1, and stopped T1 immediately prior to COMMIT.
2. Started an update transaction T2. T2 was blocked.
3. Committted T1. T2 should now complete and commit too.
4. Compared extendedprice after update in T1 and extendedprice before update in T2. They should be same.

#### 3.4.4 Isolation Test 4 – Write-Write Conflict with Rollback

*Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.*

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

1. Started an update transaction T1, and stopped T1 immediately prior to COMMIT.
2. Started an update transaction T2. T2 was blocked.
3. Rollbacked T1. T2 should now complete and commit, too.
4. Compared extendedprice before update in T1 and extendedprice before update in T2. They should be same.

#### 3.4.5 Isolation Test 5 – Concurrent Read and Write Transactions on Different Tables

*Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.*

The following operations were performed to demonstrate the ability of read and write transactions involving different tables to make progress concurrently:

1. Started a query and verified that the row was retrieved.
2. Started an update transaction, read and updated the same row. Stalled before commit.
3. Started another transaction that performed the following operation: Select random values of PS_PARTKEY AND PS_SUPPKEY. Return all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal to the selected values.
4. Verified that the read transaction completed.
5. Committed the update transaction.
6. Started the same query and verified that the new row was retrieved.

#### 3.4.6 Isolation Test 6 – Update Transactions during Continuous Read-Only Query Stream

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

The following query was used to ensure sufficient execution to perform the test:

1. A transaction, T1, which executed the above query on the qualification database, was started using a random DELTA.
2. A transaction, T2, was started for a randomly selected O_KEY, L_KEY, and DELTA.
3. T2 completed and appropriate rows in the tables ORDERS, LINEITEM, and HISTORY had been changed.
4. T1 was still execution.
5. Transaction T1 completed executing the query.

```
SELECT l1.l_quantity,
       SUM(l2.l_extendedprice),
       SUM(l3.l_extendedprice),
       SUM(l3.l_quantity)
FROM   lineitem l1, lineitem l2, lineitem l3, lineitem l4, lineitem l5
WHERE  l1.l_shipdate <= DATE '1998-12-01' -0
       AND l1.l_orderkey = l2.l_orderkey
       AND l1.l_linenumber = l2.l_linenumber
       AND l1.l_extendedprice = l3.l_extendedprice AND l3.l_quantity < 30
       AND l4.l_quantity = l1.l_quantity AND l4.l_orderkey < 150
       AND l5.l_receiptdate = l1.l_receiptdate AND l5.l_partkey <140
GROUP BY l1.l_quantity;
COMMIT;
```

## 3.5 Durability Requirements

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

The following steps were performed for the durability test:

1. The consistency of the ORDERS and LINEITEM tables was verified.
2. 400 transactions for each of the 11 executions streams were prepared.
3. After at least 100 ACID transactions were completed by each of the 11 execution streams.
4. A durability failure was induced (see details for each failure shown below).
5. The consistency of the ORDERS and LINEITEM tables was verified again.
6. The durability success files were compared with the HISTORY table.

All durability tests were performed on the cluster shown in Section 0.3.

### 3.5.1 Permanent Unrecoverable Failure of Any Durable Medium

*Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-H database tables or recovery log tables.*

Disk, node, and controller failure tests were performed together as shown in Section 3.5.5.

### 3.5.2 System Crash

*Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.*

The system crash, memory failure, and loss of external power tests were performed together as shown in Section 3.5.4.

### 3.5.3 Memory Failure

*Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents). See the previous section.*

The system crash, memory failure, and loss of external power tests were performed together as shown in Section 3.5.4.

### 3.5.4 Loss of External Power

*Loss of External Power: Guarantee the database and the effects of committed updates are preserved during the loss of all external power to the SUT for an indefinite time period.*

The tested cluster consists of 2 master instances and 576 computing instances. Two tests were performed to test the durability:

1. Shutdown the host machine of the primary master instance. After completion of the durability test, the full cluster was restored automatically after the master node was started manually.
2. Shutdown the host machine of the mirror master instance. Note that the mirror master instance shares the host machine with compute instances. After completion of the durability test, the full cluster was restored automatically after the compute node was started manually.

#### 3.5.5 Node or Controller Failure

*Guarantee the database and committed updates are preserved across failure of the controller or the whole node.*

The node or controller failure test was performed together with the loss of external power test as shown in Section 3.5.4.

# 4. Clause 4: Scaling and Database Population

## 4.1 Initial Cardinality of Tables

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

Table 2 lists the TPC Benchmark™ H defined tables and the row count for each table as they existed upon completion of the test database build.

**Table 2.** Table Cardinalities.

| Table | #Rows |
|---|---|
| Lineitem | 179,999,978,268 |
| Order | 45,000,000,000 |
| Partsupp | 24,000,000,000 |
| Part | 6,000,000,000 |
| Customer | 4,500,000,000 |
| Supplier | 300,000,000 |
| Nation | 25 |
| Region | 5 |

## 4.2 Distribution of Tables and Logs Across Media

*The distribution of tables and logs across all media must be explicitly described using a format similar to that shown in the following example for both the measured and priced configurations.*

**Table 3.** Distribution of tables and logs.

| Server Node | Disk Type | Disk Drive | Description of Content |
|---|---|---|---|
| Master Instance | Local SSD Disk | /dev/mapper/vgdata-volume1 | Metadata of the cluster, event log and transaction log. |
| Compute Instance | Local SSD Disk | /dev/mapper/vgdata-volume1 | Event log, transaction log, temp files, and table data. |

All the base tables were stored on the local storage, the sizes of tables are shown in Table 4.

**Table 4.** Table Size on local storage.

| Table Name | Lineitem | Order | Partsupp | Part | Customer | Supplier | Nation | Region |
|---|---|---|---|---|---|---|---|---|
| Table Size | 13,318 GB | 3,450 GB | 2,174 GB | 434 GB | 634 GB | 40 GB | 74 MB | 72 MB |

## 4.3 Mapping of Database Partitions/Replication

*The mapping of database partitions/replications must be explicitly described.*
*The intent is to provide sufficient detail about partitioning and replication to allow independent reconstruction of the test database.*

Tables are horizontally partitioned using a hash algorithm. The columns used for the hashing (e.g., distribution key) are controlled by DDL statements (see Supporting Files Archive, keyword: distributed by).

In this benchmarking, the data volume is partitioned into 576 logical segments. Each logical segment corresponds to one primary physical segment and one mirror physical segment. AnalyticDB for PostgreSQL automatically manages the physical mapping between segments and compute nodes. Segments are evenly distributed over the compute nodes. It is guaranteed that primary segments do not reside on the same compute node with their mirror segment. It is also guaranteed that the data on one mirror segment is identical with that on the corresponding primary segment. A mirror segment is switched to primary when the corresponding primary segment fails.

## 4.4 Implementation of Data Redundancy

*Implementations may use data redundancy mechanism(s). The type of data redundancy mechanism(s) and any configuration parameters (e.g., RAID level used must be disclosed for each device). If data redundancy mechanism(s) are used in an implementation, the logical intent of their use must be disclosed.*

RAID is not used. The redundancy level is shown in the execution summary. AnalyticDB for PostgreSQL manages the data redundancy at the software level. The data is partitioned into multiple segments and all segments are mirrored to achieve data redundancy. Please refer Section 4.3 for details.

## 4.5 DBGen Modifications

*The version number, release number, modification number, and patch level of DBGen must be disclosed. Any modifications to the DBGen (see Clause 4.2.1) source code (see Appendix D) must be reported in the supporting files archive.*

The supplied DBGen version 2.18.0 was used, no modifications were made.

## 4.6 Database Load Time

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

See Numerical Quantities Summary in the Executive Summary.

## 4.7 Data Storage Ratio

*The data storage ratio must be disclosed. It is computed by dividing the total data storage of the priced configuration (expressed in GB) by the size chosen for the test database as defined in Clause 4.1.3.1. Let r be the ratio. The reported value for r must be rounded to the nearest 0.01. That is, reported value=round(r,2). For example, a system configured with 96 disks of 2.1 GB capacity for a 100GB test database has a data storage ratio of 2.02.*

*For the reporting of configured disk capacity, gigabyte (GB) is defined to be $2^{30}$ bytes. Since disk manufacturers typically report disk size.*

| Server Node | Disk Size/Node (GB) | Node Count | Total(GB) |
|---|---|---|---|
| Master Instance | 320 | 2 | 640 |
| Compute Instance | 96 | 576 | 55,296 |
| | | Total | 55,936 |

Total disk capacity: 55,936 GB
Scale factor: 30,000
The database storage ratio is 1.86.

## 4.8 Database Load Mechanism Details and Illustration

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

Figure 2 shows the database build procedure. The raw data flat files were created using DBGen and stored on Alibaba Cloud OSS (https://www.alibabacloud.com/product/oss). The configuration for loading data from Alibaba Cloud OSS is disclosed in the Supporting Files Archive. After the database build process, the connection between the measured configuration and the Alibaba Cloud OSS was disabled.

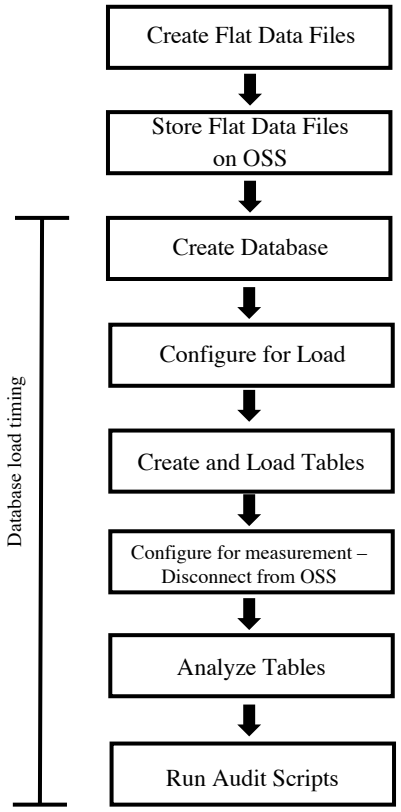



**Fig. 2.** Block Diagram of Database Loading Procedure.

## 4.9 Qualification Database Configuration

*Any differences between the configuration of the qualification database and the test database must be disclosed.*

The qualification database used identical scripts to create and load the data with changes to adjust for the database scale factor.

## 4.10 Memory to Database Size Percentage

*The memory to database size percentage must be disclosed. It is computed by multiplying by 100 the total memory size priced on the SUT (see clause 6.2.1 ) and dividing this number by the size chosen for the test database as defined in Clause 4.1.3.1. Let r be this ratio. The reported ratio must be rounded to the nearest 0.1. That is, reported value=round(r,1). For example, a system configured with 256GB of memory for a 1000GB test database has a memory/database size percentage of 25.6.*

Available Memory: 28,032 GB
Scale Factor: 30,000
The memory to database size percentage is 93.4%.

# 5. Clause 5: Performance Metrics and Execution Rules Related Items

## 5.1 System Activity between Load and Performance Tests

*Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully reported in the supporting files archive including listings of scripts, command logs and system activity.*

The only activity between the Load and the Performance Tests was the generation of queries using QGen.

## 5.2 Steps in the Power Test

*The details of the steps followed to implement the power test (e.g., system boot, database restart, etc.) must be reported in the supporting files archive.*

The following steps were used to implement the power test:

- RF1 Refresh Function from the refresh stream.
- Query Execution from query stream 0.
- RF2 Refresh Function from the refresh stream.

## 5.3 Timing Interval for Each Query and Refresh Functions

*The timing intervals (see Clause 5.3.7) for each query and for both refresh functions must be reported for the power test. The output for each query and for both refresh functions must be reported in the supporting files archive.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.4 Number of Streams for the Throughput Test

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

10 query streams were used for the throughput test.

## 5.5 Start and End Date/Time of Each Query Stream

*The start time and finish time for each query stream for the throughput test must be disclosed. The output for each query stream for the throughput test must be reported in the supporting files archive.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.6 Total Elapsed Time of the Measurement Interval

*The total elapsed time of the measurement interval (see Clause 5.3.6) must be disclosed for the throughput test.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.7 Refresh Function Start Date/Time and Finish Date/Time

*The start time and, finish time for each refresh function in the refresh stream for the throughput test must be disclosed. The output of each refresh function in the refresh stream for the throughput test must be reported in the supporting files archive.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.8 Timing Metric Precision

*The start time and finish time for each query and refresh stream shall be reported to the hundredth of a second. If times are measured with the precision greater than one hundredth of a second, the reported times shall be truncated to the hundredth of a second.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.9 Performance Metrics

*The computed performance metric, related numerical quantities and the price/performance metric must be disclosed.*

See the Numerical Quantities Summary in the Executive Summary at the beginning of this report.

## 5.10 The Performance Metric and Numerical Quantities from Both Runs

*The performance metric (QphH@Size) and the numerical quantities (TPC-H Power@Size and TPC-H Through-put@Size) from both of the runs must be disclosed.*

| Run ID | QphH@30,000G | QppH@30,000G | QthH@30,000G |
|---|---|---|---|
| Run 1 | 5,147,629.1 | 3,996,716.4 | 6,629,963.8 |
| Run 2 | 5,057,263.4 | 3,844,697.0 | 6,652,257.2 |

## 5.11 System Activity between Performance Tests

*Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be fully disclosed including system activity, listings of scripts or command logs along with any system reboots or database restarts.*

There was no activity between Run 1 and Run 2.

## 5.12 Documentation to satisfy Clause 5.2.7

*All documentation necessary to satisfy Clause 5.2.7 must be made available upon request.*

The documentations of AnalyticDB for PostgreSQL is publicly available at https://www.alibabacloud.com/product/hybriddb-postgresql.

## 5.13 Query Output Validation

*The output of the Query Output Validation Test must be reported in the supporting files archive.*

The output of the validation test is available in the Supporting Files Archive.

# 6. Clause 6: SUT and Driver Implementation Related Items

## 6.1 Driver

*A detailed textual description of how the driver performs its functions, how its various components interact and any product functionalities or environmental settings on which it relies and all related source code, scripts and configuration files must be reported in the supporting files archive. The information provided should be sufficient for an independent reconstruction of the driver.*

Scripts are used to perform the tests and QGen is used to generate query text.

For each power-test run, two processes, Process A and B, are used:

- Process A executes RF1 and then waits Process B to complete.
- Process B executes the 22 queries in the required order for stream 0. After it finishes all queries, it signals the Process A to continue.
- Process A continues to execute RF2.

For each throughput-test run:

- Multiple query streams and one update stream are used to perform the throughput-test run. All streams run in parallel.
- The number of query streams is shown in the Numerical Quantities of the execution summary. Queries that are generated by QGen are submitted in the order defined in Clause 5.3.5.4.
- The update stream runs concurrently with the query streams. Pairs of RF1 and RF2 are executed repeatedly in the update stream.

The source code of all above scripts is disclosed in the Supporting Files Archive.

## 6.2 Implementation Specific Layer (ISL)

*If an implementation specific layer is used, then a detailed description of how it performs its functions, how its various components interact and any product functionalities or environmental setting on which it relies must be disclosed. All related source code, scripts and configuration files must be reported in the supporting files archive. The information provided should be sufficient for an independent reconstruction of the implementation specific layer.*

The scripts used to implement the ISL are available in the Supporting Files Archive.

## 6.3 Profile-Directed Optimization

*If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed. In particular, the procedure and any scripts used to perform the optimization must be reported in the supporting files archive.*

Profile-directed optimization was not used.

# 7. Clause 7: Pricing

## 7.1 Priced Configuration

*The pricing methodology used for pricing the Priced Configuration is the "Default 3-Year Pricing Methodology", as defined in the current revision of the TPC Pricing specification.*

*The system to be priced shall include the hardware, Licensed Compute Services and software components present in the System Under Test (SUT), a communication interface that can support user interface devices, additional operational components configured on the test system, and maintenance on all of the above:*

- *System Under Test*.
- *User Interface Devices and Communications*.
- *Database Storage and Recovery Log*.
- *Additional Operational Components*.
- *Software*.

A *3-Year Pricing Methodology* is used for this benchmark. A detailed list of hardware and software used in the priced system is included in the pricing sheet in the Executive Summary at the beginning of this report. The price quotations are included in Appendix A.

## 7.2 Availability Date and Orderability Date

*The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the availability date reported on the executive summary must be the date by which all components are committed to being available. The full disclosure report must report availability dates individually for at least each of the categories for which a pricing subtotal must be provided.*

All components of the priced configuration are available at the time of this publication.

*For each of the components that are not orderable on the report date of the FDR, the following information must be included in the FDR:*

- *Name and part number of the item that is not orderable*
- *The date when the component can be ordered (on or before the Availability Date)*
- *The method to be used to order the component (at or below the quoted price) when that date arrives*
- *The method for verifying the price*

All priced components are orderable at the time of this publication date.

## 7.3 Country-Specific Pricing

*The Priced Locale and Priced Currency of the Priced Configuration must be disclosed.*

The configuration is priced in RMB for the China market.

# 8. Clause 8: Full Disclosure

## 8.1 Supporting Files Index Table

*An index for all files and/or directories included in the Supporting Files Archive as required by Clauses 8.3.2 through 8.3.8 must be provided in the report.*

| Clause | Description | Archive Files | Pathname |
|---|---|---|---|
| 1 | Parameter Settings | Analyticdb_postgres_tpc_h_30tb.zip | qualification/qual_bb/params.log |
| | DB Creation Scripts | | adb_pg_kit/sql/create_user.sql |
| | | | adb_pg_kit/sql/create_schema.sql |
| | | | adb_pg_kit/sql/create_indices.sql |
| | | | adb_pg_kit/sql/table_sort.sql |
| | | | adb_pg_kit/sql/analyze_database.sql |
| | System Verification | | adb_pg_kit/main/tools/hwinfo.sh |
| | Toolkit Common Scripts | | adb_pg_kit/main |
| 2 | Minor query modifications | Analyticdb_postgres_tpc_h_30tb.zip | adb_pg_kit/tpch_archives/tpch_2_18_0.zip.patch |
| 3 | ACID Test Scripts | Analyticdb_postgres_tpc_h_30tb.zip | adb_pg_kit/ACID |
| | ACID Test Results | | ACID |
| 4 | Database Load Scripts | Analyticdb_postgres_tpc_h_30tb.zip | adb_pg_kit/main/load_init.sh |
| | Qualification Test Results | | qualification |
| 5 | Query Output Results | Analyticdb_postgres_tpc_h_30tb.zip | full_test/run1 |
| | | | full_test/run2 |
| 6 | Source Codes and Scripts of Driver | Analyticdb_postgres_tpc_h_30tb.zip | adb_pg_kit/main/query_streams |
| 7 | There are no files to be included for Clause 7. | N/A | N/A |
| 8 | Query Parameters & Seeds | Analyticdb_postgres_tpc_h_30tb.zip | full_test/run1/substitution_parameters.txt |
| | Executable Query Text | | full_test/run1/stream*.sql |
| | RF function source code | | adb_pg_kit/main/tpc_h_full.sh |

# 9. Clause 9: Audit Related Items

## 9.1 Auditor's Report

*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 implementation of the TPC Benchmark™ H was audited by Francois Raab of InfoSizing, a certified TPC-H auditor. Further information regarding the audit process may be obtained from:

**Francois Raab**
InfoSizing (www.sizing.com)
20 Kreg Ln.
Manitou Springs, CO 80829
(719) 473-7555

TPC Benchmark™ H Full Disclosure Report and other information can be downloaded from the Transaction Processing Performance Council web site at www.tpc.org.

InfoSizing

TPC Certified Auditor

Benchmark sponsor: Xiaoqiang Peng
Alibaba Cloud Intelligence Business Group
969 West Wen Yi Road
Yu Hang District, Hangzhou
Zhejiang, China

May 20, 2020

I verified the TPC Benchmark H (TPC-H™ v2.18.0) performance of the following configuration:

| | |
|---|---|
| Platform: | Alibaba Cloud AnalyticDB |
| Operating System: | Alibaba Group Enterprise Linux Server release 7.2 (Paladin) |
| Database Manager: | AnalyticDB for PostgreSQL 6.0 |
| Other Software: | n/a |

The results were:

| **Performance Metric** | **5,057,263.4 QphH@30000GB** |
|---|---|
| TPC-H Power | 6,652,257.2 |
| TPC-H Throughput | 5,057,263.4 |
| Database Load Time | 9h 42m 43s |

| **Server** | **Alibaba Cloud AnalyticDB Compute Service, with:** | | |
|---|---|---|---|
| **2 Master Nodes, with:** | | | |
| CPUs | 32 x vCPU | | |
| Memory | 192 GB | | |
| Disks | ***Qty*** | ***Size*** | ***Type*** |
| | 1 | 320 GB | SSD Cloud Disk (boot + metadata) |
| **576 Worker Nodes, with:** | | | |
| CPUs | 6 x vPCU | | |
| Memory | 48 GB | | |
| Disks | ***Qty*** | ***Size*** | ***Type*** |
| | 1 | 96 GB | SSD Cloud Disk (boot + data) |

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 database records were defined with the proper layout and size

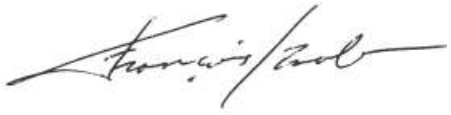
20 Kreg Lane • Manitou Springs, CO 80829 • 719-473-7555 • www.sizing.com

- The database population was generated using DBGen
- The database was properly scaled to 30,000GB and populated accordingly
- The compliance of the database auxiliary data structures was verified
- The database load time was correctly measured and reported
- The required ACID properties were verified and met
- The query input variables were generated by QGen
- The query text was produced using minor modifications and no query variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 10 query streams
- The ratio between the longest and the shortest query was such that no query timings were adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

In the course of the audit, an issue was uncovered with the dbgen tool. In my opinion, this issue did not interfere with the validity of the testing or the accuracy of the reported performance.

Respectfully Yours,



François Raab, TPC Certified Auditor

20 Kreg Lane • Manitou Springs, CO 80829 • 719-473-7555 • www.sizing.com

# Appendix A: Price Quotes

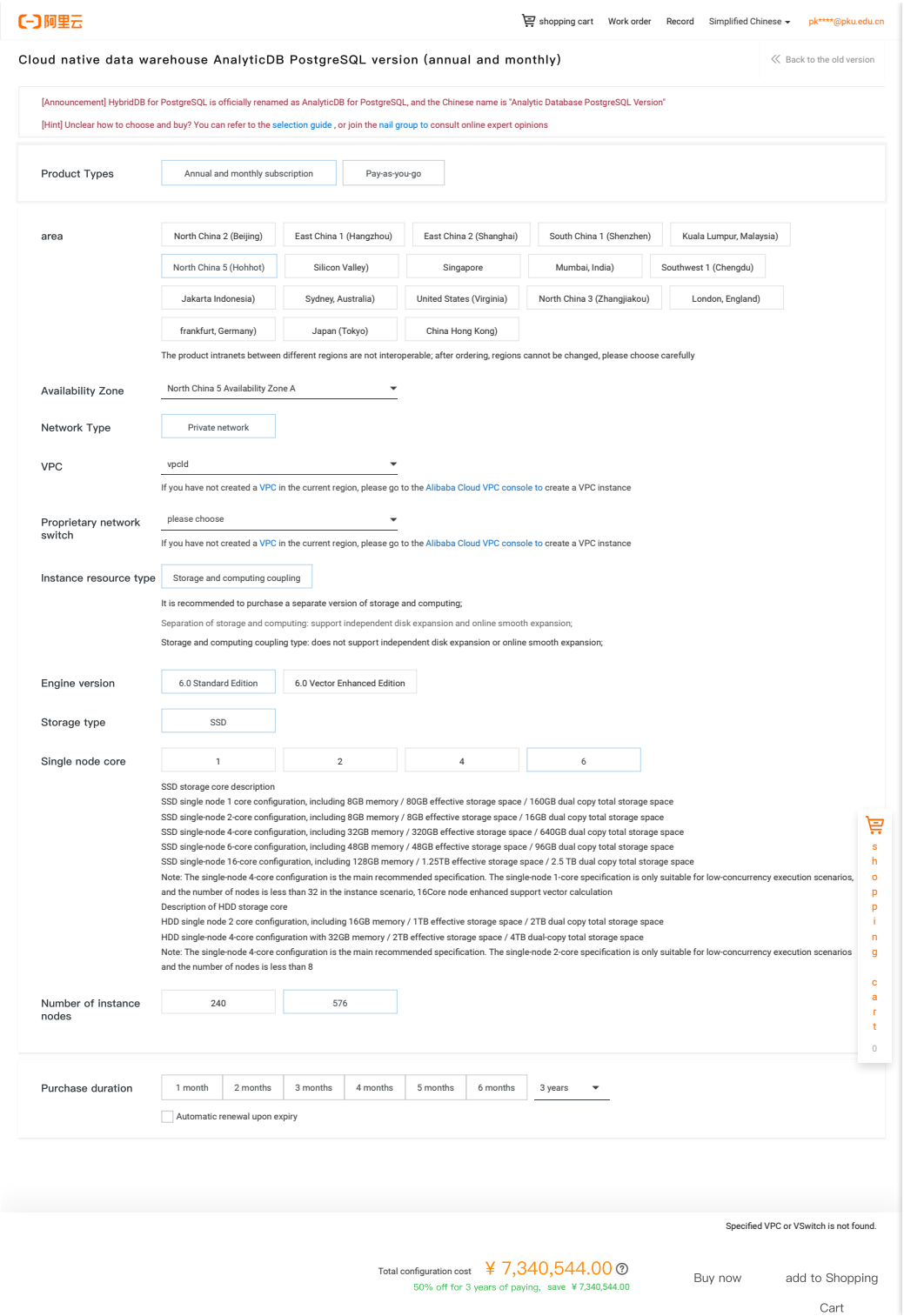



**Fig. 3.** Purchase Page for provisioning the tested Alibaba Cloud AnalyticDB for PostgreSQL 6.0 with 3-Year Subscription.

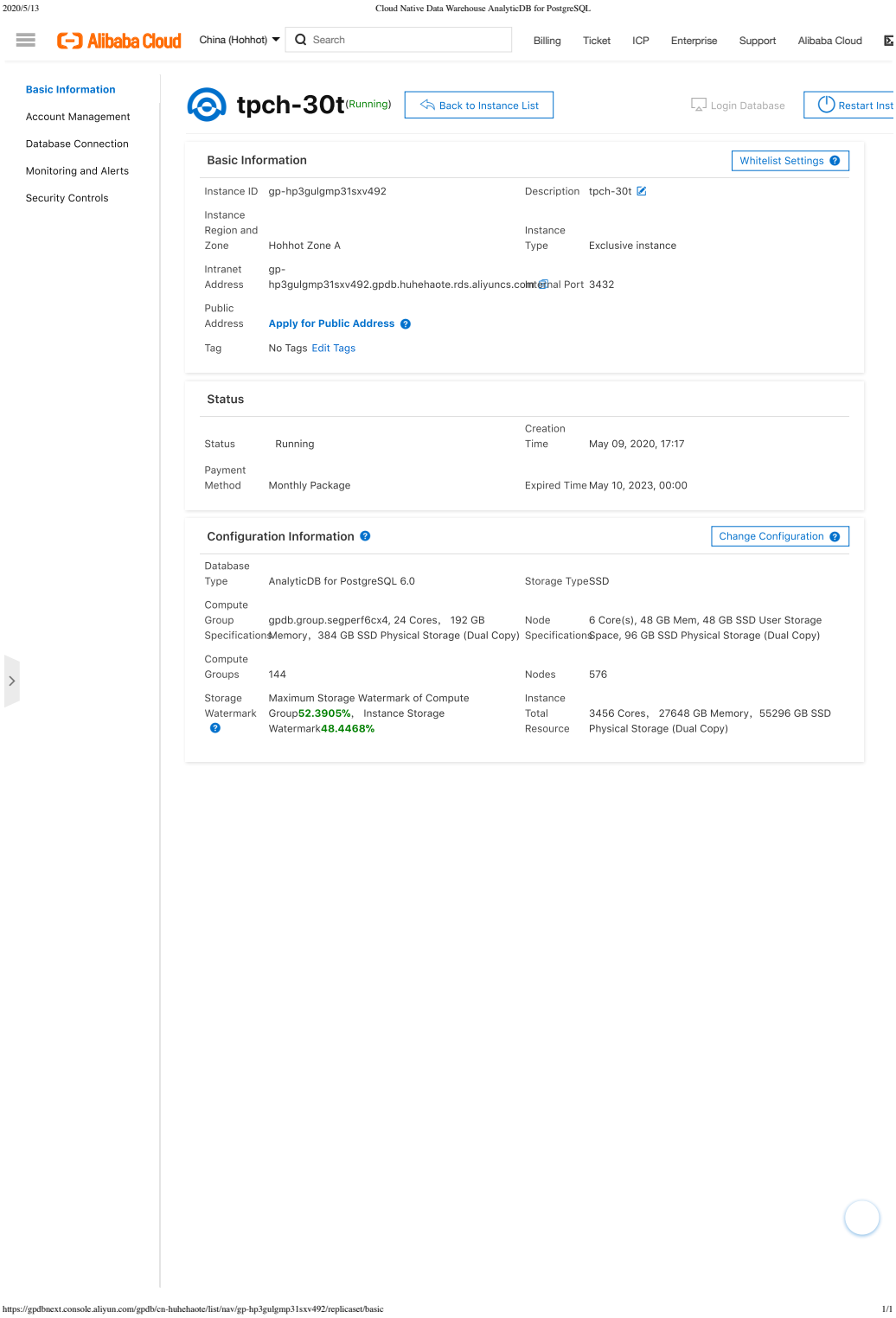



**Fig. 4.** Configuration page on the Alibaba Cloud panel.

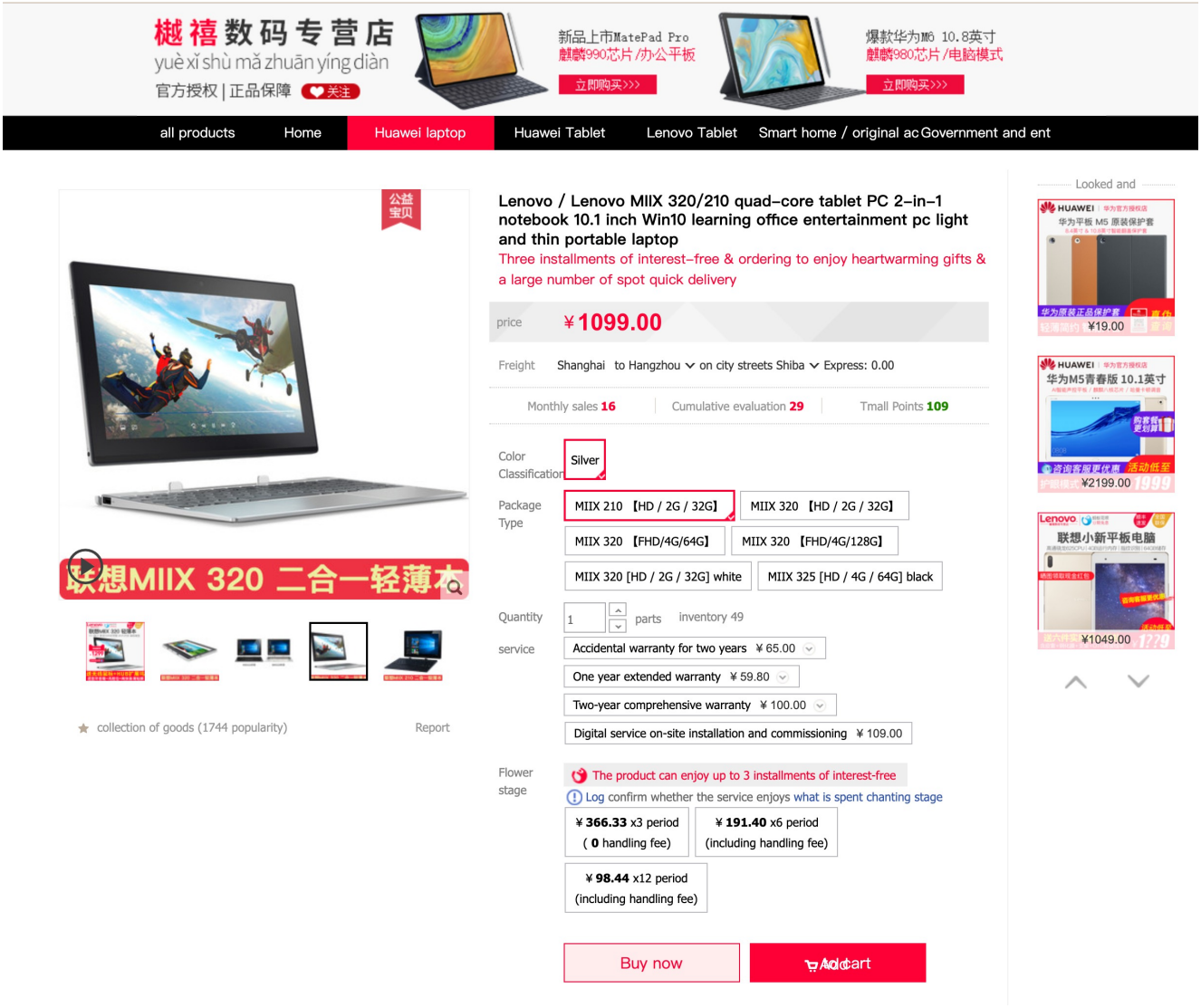



**Fig. 5.** Lenovo MIIX 210 tablet purchase page (Google translated English version).