



















表4 种子数据

序号	数据表	数据量（行）	文件大小	说明
1	订单表	3000 万	3.375G	数据量随交易日动态增长
2	成交表	4500 万	5.25G	数据量随交易日动态增长
3	逐笔行情表	300 万	825M	数据量随交易日动态增长
4	股份持有表	3000 万	1.8G	数据量随交易日动态增长
5	证券信息表	1 万	3M	资料表，数据量固定
6	投资者资料表	10 亿	145G	资料表，数据量固定
7	营业部资料表	10 万	19M	资料表，数据量固定
8	会员资料表	1000 个	0.5M	资料表，数据量固定
9	交易单元资料表	10 万	15M	资料表，数据量固定

假设以一年历史数据作为存量数据，按业务规则对种子数据的4个动态表进行数据翻倍（按一年240个交易日计算），得到存量数据，见表5。

表5 翻倍后的数据

序号	数据表	数据量（行）	文件大小
1	订单表	72 亿	810G
2	成交表	108 亿	1260G
3	逐笔行情表	7.2 亿	198G
4	股份持有表	72 亿	432G
5	证券信息表	1 万	3M
6	投资者资料表	10 亿	145G
7	营业部资料表	10 万	19M
8	会员资料表	1000 个	0.5M
9	交易单元资料表	10 万	15M

增量数据同样是在种子数据基础上按业务规则对动态表进行数据量翻倍以及数据倾斜改造。以订单表为例，按四档数据规模划分，见表6。

表6 数据规模划分

序号	订单表数据规模	数据量（行）	文件大小
1	一般	3000 万	3.375G
2	平峰	8000 万	9G
3	高峰	1.2 亿	13.5G
4	极限	4 亿	45G

每一档数据再根据数据倾斜程度准备“无倾斜”、“一般倾斜”、“极端倾斜”三份数据，则订单表总计需要准备12份数据待用。

成交表、逐笔行情表和股份持有表采用同样的方法各生成12份数据待用，最终完成所有组合的增量测试数据准备。

## 7 测试内容和方法

### 7.1 测试概述

大数据平台性能测试主要包括平台基准测试和基于业务场景的性能测试：

- a) 平台的基准测试，该项测试采用自带的业务模型，对大数据平台进行指标定量、可对比的性能测试；
- b) 基于业务场景的性能测试，采用已准备的代表不同数据规模和数据特征的测试数据组合开展基于业务场景的性能测试，包括以下测试内容：
  - 1) 数据加载性能测试：数据加载分为全量加载、增量加载和库内加载。全量加载模拟的是业务存量数据迁移到被测系统的场景，属于一次性加载场景，主要考察加载效率；增量加载模拟的是外部业务数据每日增量加载到被测系统的场景，属于实时加载场景，主要考察吞吐量和时延；库内加载模拟的是增量业务数据归档（从当前表存入历史表）的场景，同样考察加载效率。出于对场景通用性的考虑，全量加载的数据源采用文本数据源，增量加载的数据源采用 Kafka（名称解释参见附录 A 的 A.1）数据源；
  - 2) 数据查询性能测试：数据查询性能测试模拟的是业务应用程序向被测系统发起查询请求的场景，通过调整查询语句的复杂程度和查询的并发量，渐进式的评估被测系统处理查询任务时的性能表现。测试内容分为单项查询性能测试和并发查询性能测试；
  - 3) 混合负载性能测试：混合负载性能测试是数据增量加载性能测试与并发查询性能测试的一种测试组合，模拟的是被测系统在实时加载数据的过程中，同时处理业务查询请求的场景。测试目的是观察加载与查询同时开展时，相比于加载与查询独立开展时性能指标发生了多大的变化。
  - 4) 数据操作性能测试：数据操作指增加、更新和删除数据的操作。因数据加载实际上已覆盖到增加数据的操作，所以本项测试只针对更新和删除数据进行测试，测试模拟的场景主要是采用 SQL（update 和 delete）语句更新和删除指定表指定范围的数据；
  - 5) 数据导出性能测试：数据导出常见的两种场景为全表导出和查询结果集导出，针对这两种场景分别开展性能测试。

## 7.2 基准测试

基准测试是指通过设计科学的测试方法、测试工具和测试系统，实现对一类测试对象的某项性能指标进行定量的和可对比的测试。企业可根据自身系统设计目标和业务特点选择适合的 TPC 测试基准测试套，常见的基准测试套参见附录 A 的 A.2。

注：TPC（Transaction Processing Performance Council 事务处理性能委员会）是行业内权威的基准测试组织，针对大数据平台 TPC 提供了若干基准测试套，根据给定系统和数据库每单位时间处理事务数来生成测量被测系统的性能基准。

## 7.3 数据加载性能测试

### 7.3.1 全量加载性能测试

#### 7.3.1.1 指标说明

测试输出的指标如下：

**文本数据加载速率**，是指在给定的约束条件下，ETL 服务器一次性加载文本数据到被测系统，平均每秒加载的记录数，单位：万行/秒。计算公式：

文本加载速率=总记录数/(加载完成时间-加载开始时间)

#### 7.3.1.2 约束条件

全量加载性能测试的约束条件见表 7。

表 7 约束条件

序号	名称	内容描述
1	加载方式	采用被测系统配套的 ETL 工具和加载方式,用于计算性能指标的加载开始时间和完成时间也以该配套工具记录的时间为准。
2	文本数据存储位置	文本数据统一存放于 ETL 服务器的本地磁盘的一个独立分区文件系统。
3	文本文件数量	每张表对应一个文本文件,文本文件数量等于测试数据表的数量。

### 7.3.1.3 测试准备

测试准备包括以下方面:

- a) 存量测试数据使用文本文件形式存储于 ETL 服务器本地磁盘;
- b) ETL 工具准备就绪。

### 7.3.1.4 测试用例

全量加载性能测试的测试用例见表 8。

表 8 测试用例描述

序号	测试用例描述
1	存量数据一次性加载到被测系统。

### 7.3.1.5 测试执行

表8中序号1测试用例的测试执行方法如下:

- a) 使用被测系统配套的 ETL 工具在 ETL 服务器上将所有测试数据加载到被测系统;
- b) 记录加载开始到加载完成的时间,观察被测系统的系统资源占用率;
- c) 测试结束后计算文本数据加载速率。

### 7.3.1.6 结果评估

当测试结果满足以下条件时,视为有效测试:

- a) 测试时间内被测系统的资源消耗情况在预期范围;
- b) 加载到被测系统的数据记录数与源数据记录数一致。

测试结果中文本加载速率越高,可认为被测系统在此用例场景下性能越优。

## 7.3.2 增量加载性能测试

### 7.3.2.1 指标说明

测试输出以下两项指标:

- a) **Kafka 数据加载吞吐速率**,是指在给定的约束条件下,业务数据由数据生产者(Producer)程序实时写入 Kafka 集群,ETL 服务器实时从 Kafka 集群加载数据到被测系统,每秒可加载的最大记录数,单位:万行/秒;
- b) **Kafka 数据加载时延**,是指在给定的 Kafka 数据加载吞吐速率下,对于每一行待加载的数据,T1 是 ETL 服务器开始读取这行数据的时间,T2 是这行数据最终写入被测系统文件系统的时间,T2-T1 为这行数据的加载时延,单位:ms(毫秒)。统计描述宜采用平均值,用以衡量时延的平均值大小。

## 7.3.2.2 约束条件

增量加载性能测试的约束条件见表 9。

表 9 约束条件

序号	名称	内容描述
1	Producer 程序实现	Producer 程序的作用是模拟业务程序按特定速率（可变）向 Kafka 持续写入数据，此程序需要自行编码实现，具体实现方式不限，写入到 Kafka 的数据可被 ETL 程序读取。Producer 程序运行在被测系统和 ETL 服务器之外的独立服务器上。
2	数据加载时间戳记录	可通过 ETL 程序记录被加载的每一行数据的 T1 和 T2 时间戳，以便用于后续的加载时延计算和统计时延平均值。

## 7.3.2.3 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统；
- b) Producer 程序与增量数据已准备就绪。

## 7.3.2.4 测试用例

增量加载性能测试的测试用例见表 10。

表 10 测试用例描述

序号	测试用例描述
1	增量数据实时加载到被测系统。

## 7.3.2.5 测试执行

表10中的序号1的测试用例的测试执行方法如下：

- a) 采用 Producer 程序分批向 Kafka 写入增量测试数据，ETL 服务器实时从 Kafka 读取数据加载到被测系统。Producer 程序按批次调整写入速率，每批持续一定的时间（例如 10 分钟），找到被测系统可承受的最大持续压力速率，此时观察到的被测系统数据加载速率即为吞吐速率；
- b) 将测试环境恢复到 7.3.2.5 a) 开始执行前的状态，根据需要选取其它测试数据组合，重复 7.3.2.5 a) 的测试，以观察不同数据规模和不同数据倾斜程度对性能指标的影响。

## 7.3.2.6 结果评估

结果评估在测试有效的情况下开展，具体如下：

- a) 当测试结果满足以下条件时，视为有效测试：
  - 1) 测试时间内被测系统的资源消耗情况在预期范围；
  - 2) 设定加载时延目标值  $t$ （例如 1 秒），计算达到吞吐速率的加载批次中，加载时延超过目标值  $t$  的记录数占比  $y$ ， $y = \text{加载时延超过 } t \text{ 的记录数} / \text{该批次加载的总记录数}$ ，当  $y$  不超过预期值（例如 3%）时视为有效测试。
- b) 测试结果评估如下：
  - 1) 吞吐速率指标，采用相同测试数据测试得到的吞吐速率越高，可认为被测系统在此用例场景下性能越优；
  - 2) 加载时延指标，采用相同测试数据测试得到的平均加载时延越低，可认为被测系统在此用例场景下性能越优；

- 3) 上述两项指标,采用相同数据规模不同数据倾斜程度的测试数据进行测试,得到的指标结果受数据倾斜影响越小,可认为被测系统在此用例场景下性能越优。

### 7.3.3 库内加载性能测试

#### 7.3.3.1 指标说明

测试输出的指标如下:

**表间数据插入速率**,是指在给定的约束条件下,在被测系统的同一个数据库内将一张表的数据插入到另一张表,平均每秒插入的记录数,单位:万行/秒。计算公式:

表间数据插入速率=插入的总记录数/(数据插入完成时间-数据插入开始时间)

#### 7.3.3.2 约束条件

库内加载性能测试的约束条件见表11。

表 11 约束条件

序号	名称	内容描述
1	表间数据插入方式	采用执行 SQL 的方式完成表间数据插入,SQL 语句形式为: insert into 目标表 select * from 源表。
2	源表与目标表存放位置	源表与目标表分别位于同一个库的不同 schema 下。
3	源表与目标表结构	源表与目标表结构一致,索引、主键、分布键和分区键等所有特性相同。

#### 7.3.3.3 测试准备

测试准备包括以下方面:

- 存量测试数据已全部加载到被测系统;
- 选取存量测试数据中所有记录数超过 10 亿行的表作为目标表,按照约束条件 2 和 3 为每一张目标表创建对应的源表;
- 为每一张源表预置 1 亿行数据,这些数据与目标表不存在唯一性约束冲突。

#### 7.3.3.4 测试用例

库内加载性能测试的测试用例见表 12。

表 12 测试用例描述

序号	测试用例描述
1	源表数据一次性插入到目标表

#### 7.3.3.5 测试执行

表12中序号1测试用例的测试执行方法如下:

- 选择一张源表,按照约束条件 1 的方式执行 SQL,将数据插入到目标表;
- 记录 SQL 开始执行到结束执行的时间,观察被测系统的系统资源占用率;
- 重复 7.3.3.5 a) 和 b),遍历执行每一张源表插入数据到目标表的测试;
- 测试结束后计算每一次测试得到的表间数据插入速率。

#### 7.3.3.6 结果评估

当测试结果满足以下条件时,视为有效测试:

- a) 测试时间内被测系统的资源消耗情况在预期范围；
- b) 插入到每一张目标表的记录数与其对应的源数据记录数一致。

测试结果中同一对表的表间数据插入速率越高，可认为被测系统对该类表结构的库内加载性能越优。

## 7.4 数据查询性能测试

### 7.4.1 查询 SQL 说明

#### 7.4.1.1 SQL 复杂度划分

从查询数据范围、查询过滤条件、查询过程运算和返回结果集等维度，将查询测试用到的SQL语句按复杂程度归类为简单查询、中等查询和复杂查询三类。用于判断SQL复杂程度的维度说明如下：

- a) 查询数据范围，包括不限于：目标表数据规模、单表或多表关联、嵌套（子查询）、集合查询等；
- b) 查询过滤条件，包括不限于：过滤条件数量、过滤数据范围、数据分组形式等；
- c) 查询过程运算，包括不限于：聚合运算、条件判断（含非空判断）、四则运算、和数据类型转换等；
- d) 返回结果集，包括不限于：结果集规模，结果集排序方式等。

查询SQL中，涉及到上述维度的内容越多或程度越深，视为越复杂的查询。

注：上述SQL复杂程度划分并非定义严谨的划分，主要目的是提供一种思路，方便后续的测试比较与观察。一种可行的操作方式是，从生产环境选取典型的业务查询SQL语句，进行适当的抽象改造，按上述维度进行复杂程度划分，用于后续的测试。

#### 7.4.1.2 简单查询 SQL

##### 7.4.1.2.1 简单查询 1-小表简单过滤

小表简单过滤的查询特征见表13。

表 13 查询特征

序号	维度	特征	内容描述
1	查询数据范围	小表查询	目标表为小表，数据量 10 万行以内。
2	查询过滤条件	简单过滤	仅采用一个过滤条件。
3	查询过程运算	无过程运算	取字段原值，无任何运算。
4	返回结果集	单个结果	返回一行结果数据。

查询场景示例：

从证券信息表查询指定证券代码的一条证券信息。

##### 7.4.1.2.2 简单查询 2-大表多条件过滤并排序

大表多条件过滤并排序的查询特征见表14。

表 14 查询特征

序号	维度	特征	内容描述
1	查询数据范围	大表查询	目标表为大表，数据量 10 亿行以上。
2	查询过滤条件	多过滤条件	采用 3 到 5 个过滤条件。
3	查询过程运算	无过程运算	取字段原值，无任何运算。
4	返回结果集	大结果集	返回结果集在 10 万行以上。
5	返回结果集	按单列排序	返回结果按其中 1 个字段进行排序。

**查询场景示例：**

从成交表按指定日期范围、指定时间段、指定证券代码和指定成交量范围查询证券的成交明细信息，查询结果按成交日期由远及近进行排序。

**7.4.1.2.3 简单查询 3-大表分组集合查询**

大表分组集合查询的查询特征见表15。

表 15 查询特征

序号	维度	特征	内容描述
1	查询数据范围	多表集合查询	使用 union all 关键字集合 2 到 3 个子查询。
2	查询数据范围	大表查询	目标表至少有 2 个表是大表，大表数据规模 10 亿行以上。
3	查询过滤条件	多过滤条件	每个子查询采用 3 到 5 个过滤条件。
4	查询过滤条件	单列分组	每个子查询对其中一列进行 group by 分组。
5	查询过程运算	少量聚合运算	每个子查询涉及 3 项以内聚合运算，例如 sum、count 等。
6	返回结果集	小结果集	返回结果集在万行以内。

**查询场景示例：**

子查询1（

查询成交表，按指定日期范围、指定时间段和指定证券代码作为过滤条件，并按证券代码分组计算，列出证券代码、证券的总成交数量和成交笔数

) union all

子查询2（

查询订单表，按指定日期范围、指定时间段和指定证券代码作为过滤条件，并按证券代码分组计算，列出证券代码、证券的总订单数量和订单笔数

) union all

子查询3（

查询证券表，按指定证券代码作为过滤条件，并按证券代码分组，列出证券代码、交易数量单位和证券信息条数

)

**7.4.1.3 中等查询 SQL****7.4.1.3.1 中等查询 1-大表关联多条件过滤并分组排序**

大表关联多条件过滤并分组排序的查询特征见表16。

表 16 查询特征

序号	维度	特征	内容描述
1	查询数据范围	大表关联	两个大表采用左连接关联，每个表数据量 10 亿行以上，关联条件 2 个以内。
2	查询过滤条件	多过滤条件	采用 5 个或以上数量的过滤条件。
3	查询过滤条件	多列分组	按 2 到 3 列进行 group by 分组。
4	查询过程运算	少量聚合运算	每个子查询涉及 3 项以内聚合运算，例如 sum、max 等。
5	返回结果集	大结果集	返回结果集在 10 万行以上。
6	返回结果集	多列排序	返回结果按 2 到 3 列进行排序。

**查询场景示例：**

成交表与投资者资料表按证券账户代码与投资者类别两列作为关联条件进行左连接关联，以指定日期范围、指定时间段、指定成交量和指定成交类别作为过滤条件，按成交日期和证券代码两列进行分组计算，列出成交日期、证券代码、总成交数量和最大成交价格，返回结果按成交日期和证券代码进行升序排序。

## 7.4.1.3.2 中等查询 2-多表关联多列分组按聚合运算结果排序

多表关联多列分组按聚合运算结果排序的查询特征见表17。

表 17 查询特征

序号	维度	特征	内容描述
1	查询数据范围	多表关联	5 个或以上的表关联，其中 3 个大表（数据量 10 亿行以上），其余小表（数据量 10 万行以内），关联条件总计 10 个或以上。
2	查询过滤条件	多过滤条件	采用 5 个或以上数量的过滤条件。
3	查询过滤条件	多列分组	按 5 列或以上进行 group by 分组。
4	查询过程运算	少量聚合运算	查询涉及 3 项以内聚合运算，例如 sum、min 等。
5	返回结果集	结果集数量限定	使用 limit 关键字限定返回结果集为一万行。
6	返回结果集	按聚合运算结果排序	按其中一列 sum 运算结果倒序排序。

**查询场景示例：**

成交表与股份持有表、证券信息表、会员资料表、营业部资料表、交易单元资料表、投资者资料表6个表进行关联，以指定日期范围、指定时间段、指定成交量、指定证券代码范围和指定成交类别作为过滤条件，按证券代码、证券账户代码、投资者代码、投资者名称、投资者类别、交易单元名称6列进行分组计算，列出证券代码和证券账户代码维度对应的投资者信息、交易单元信息以及买入成交总量，返回结果按买入成交总量进行倒序排序，并限定结果集数量为1万行。

## 7.4.1.4 复杂查询 SQL

多表关联子查询辅以条件判断及四则运算的查询特征见表18。



表 18 查询特征

序号	维度	特征	内容描述
1	查询数据范围	嵌套子查询	使用 with 关键字定义子查询，子查询作为父查询的目标表之一，子查询结果集千行以内。
2	查询数据范围	多表关联	子查询的结果集作为临时表，与其它 3 个或以上的表进行关联，至少有一个大表（数据量 10 亿行以上）。
3	查询过滤条件	单列分组	在嵌套的子查询内对其中一列进行 group by 分组。
4	查询过程运算	多项聚合运算	子查询和父查询都使用了聚合运算，其中父查询使用了 5 项或以上的聚合运算，包括但不限于 sum、max、min、count 等。
5	查询过程运算	多项四则运算	父查询使用了 5 项或以上的四则运算，包含了加减乘除法运算。
6	查询过程运算	少量条件判断	父查询使用了 2 到 3 项条件判断，包含 case when 判断和非空判断。
7	查询过程运算	数据类型转换	父查询进行了 2 到 3 项数据类型转换，例如 date 转 char、char 转 decimal、integer 转 char 等。
8	返回结果集	多列排序	返回结果按 2 到 3 列进行排序。
9	返回结果集	小结果集	返回结果集在千行以内。

**查询场景示例：**

子查询：查询成交表，按证券账户分组计算，列出净买卖金额排名前 200 名的证券账户代码和净买卖金额，查询结果按净买卖金额进行排序；

父查询：子查询结果集作为临时表，与投资者资料表、营业部资料表和会议资料表 3 个表分别按内连接进行关联，查询过程采用了条件判断、聚合运算、四则运算等，通过计算最终列出净买卖金额前 200 名的证券账户代码、投资者名称、营业部代码、营业部名称、会员代码、会员名称、买入总金额、卖出总金额、净买卖金额、买卖股票支数、主动买入总金额、主动卖出总金额，返回结果按净买卖金额的倒序和证券代码的升序进行排序。

**7.4.2 查询参数化**

查询参数化指在设计 SQL 语句时，在需要填入数值或数据的地方，使用参数（parameter）赋值。

**示例：**

```
select * from table_name where col_1 = @col_value_1 and col_2 = @col_value_2
```

对上述查询传入不同的 @col\_value\_1 和 @col\_value\_2 可能获得不同的查询结果。在开展查询性能测试之前，对于已定义的每一项查询 SQL，需各准备一个参数值集合，作为该查询过滤条件的取值输入集合。后续测试过程中的每一次查询，都从该查询的参数值集合中随机取一组值作为查询过滤条件输入。参数值集合应满足以下条件：

- a) 集合中任意一组值作为输入，查询返回结果非空；
- b) 集合中任意一组值作为输入，查询返回的结果集在数据量上相当。

**7.4.3 指标说明**

数据查询性能测试输出的主要指标是查询响应时间。

**查询响应时间**，是指对于一次查询，查询服务器向被测系统发出查询请求的时间为 T1，收到返回查询结果的时间是 T2，T2-T1 为查询响应时间，单位：ms（毫秒）。

查询性能统计指标，见表 19。

表 19 查询性能统计指标

指标	说明
平均响应时间	统计范围内，所有已完成的查询响应时间平均值，单位：ms（毫秒）。
中位响应时间	统计范围内，所有已完成的查询响应时间按从小到大排序，位于中间位置的值，单位：ms（毫秒）。
90%响应时间	统计范围内，所有已完成的查询中，有 90%的查询响应时间小于或等于该值，ms（毫秒）。
99%响应时间	统计范围内，所有已完成的查询中，有 99%的查询响应时间小于或等于该值，ms（毫秒）。
最小响应时间	统计范围内，所有已完成的查询中，响应时间最小值，单位：ms（毫秒）。
最大响应时间	统计范围内，所有已完成的查询中，响应时间最大值，单位：ms（毫秒）。
错误率	统计范围内，查询报错的次数在查询总次数中的占比（百分比）。

#### 7.4.4 单项查询性能测试

##### 7.4.4.1 测试准备

全量测试数据已加载到被测系统。

##### 7.4.4.2 测试用例

单项查询性能测试的测试用例见表20。

表 20 测试用例描述

序号	测试用例描述
1	简单查询 1 性能测试
2	简单查询 2 性能测试
3	简单查询 3 性能测试
4	中等查询 1 性能测试
5	中等查询 2 性能测试
6	复杂查询 1 性能测试

##### 7.4.4.3 测试执行

表20中序号1~6测试用例采用相同的测试执行方法，如下：

- a) 根据测试用例，选取对应的查询 SQL，通过查询服务器向被测系统连续发出一定数量的查询请求；
- b) 查询服务器连续发出查询请求的频率为，每当接收到一个查询的返回结果后，发起一个新的查询请求，即同一时间只会有一个查询在被测系统中执行，每个查询从对应的参数集合随机取一组值作为查询过滤条件输入；
- c) 测试过程中记录每一个查询的响应时间，观察被测系统的资源占用率；
- d) 测试结束后，计算查询性能统计指标值；
- e) 根据需要选取其它测试数据组合，替换被测系统中的测试数据。重复上述测试步骤，以观察不同数据规模和不同数据倾斜程度对性能指标的影响。

##### 7.4.4.4 结果评估

结果评估在测试有效的情况下开展，具体如下：

- a) 当测试结果满足以下条件时，视为有效测试：
  - 1) 测试时间内被测系统的资源消耗情况在预期范围；

- 2) 查询性能统计指标值在预期范围;
  - 3) 连续的查询次数达到足够多的数量, 例如 100 次;
  - 4) 查询错误率为 0。
- b) 测试结果评估如下:
- 1) 同一用例, 采用相同测试数据进行测试, 得到的查询平均响应时间越小, 可认为被测系统在该用例场景下查询性能越优;
  - 2) 同一用例, 采用相同数据规模不同倾斜程度的测试数据进行测试, 得到的查询平均响应时间受数据倾斜影响越小, 可认为被测系统在该用例场景下查询性能越优。

#### 7.4.5 并发查询性能测试

##### 7.4.5.1 测试准备

全量测试数据已加载到被测系统。

##### 7.4.5.2 测试用例

并发查询性能测试的测试用例见表 21。

表 21 测试用例描述

序号	测试用例描述
1	简单查询 1 并发性能测试
2	简单查询 2 并发性能测试
3	简单查询 3 并发性能测试
4	中等查询 1 并发性能测试
5	中等查询 2 并发性能测试
6	复杂查询 1 并发性能测试
7	混合 (简单、中等、复杂) 查询并发性能测试

##### 7.4.5.3 测试执行

表21中序号1~6测试用例采用相同的测试执行方法, 如下:

- a) 根据测试用例选取对应的查询 SQL, 结合查询的复杂程度设定合理的并发数量, 记为  $c$ ;
- b) 通过查询服务器向被测系统同时发出  $c$  个查询请求, 需特别说明的是, 每个查询皆从对应的参数集合随机取一组值作为本查询的过滤条件输入;
- c) 测试过程中记录每一个查询的响应时间, 观察被测系统的资源占用率;
- d) 测试结束后, 计算查询性能统计指标值;
- e) 根据需要选取其它测试数据组合, 替换被测系统中的测试数据。重复上述测试步骤, 以观察不同数据规模和不同数据倾斜程度对性能指标的影响。

表21中的序号7的测试用例的测试执行方法在7.4.5.3 a) 上有所区别, 具体为: 从简单、中等、复杂三类查询中各选取1项查询SQL形成混合查询 (例如简单查询1、中等查询2、复杂查询1的组合), 分别设定每项查询合理的并发数量, 记为 $c_1$ 、 $c_2$ 、 $c_3$ , 并发总数记为 $c=c_1+c_2+c_3$ 。需特别说明的是, 三个并发数量的比例 $c_1:c_2:c_3$ , 可根据生产实际查询业务中简单、中等、复杂查询的真实比例进行设定。其它测试执行步骤相同。

##### 7.4.5.4 结果评估

结果评估在测试有效的情况下开展，具体如下：

- a) 当测试结果满足以下条件时，视为有效测试：
  - 1) 测试时间内被测系统的资源消耗情况在预期范围；
  - 2) 查询性能统计指标值在预期范围；
  - 3) 所有查询全部完成，错误率为 0。
- b) 测试结果评估如下：
  - 1) 同一用例，采用相同测试数据进行测试，得到的查询平均响应时间越小，可认为被测系统在该用例场景下查询性能越优；
  - 2) 同一用例，采用相同数据规模不同倾斜程度的测试数据进行测试，得到的查询平均响应时间受数据倾斜影响越小，可认为被测系统在该用例场景下查询性能越优。

## 7.5 混合负载性能测试

### 7.5.1.1 指标说明

采用与增量加载性能测试、并发查询性能测试相同的指标。

### 7.5.1.2 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统；
- b) Producer 程序与增量数据已准备就绪。

### 7.5.1.3 测试用例

混合负载性能测试的测试用例见表 22。

表 22 测试用例描述

序号	测试用例描述
1	增量加载的同时并发执行简单查询
2	增量加载的同时并发执行中等查询
3	增量加载的同时并发执行复杂查询
4	增量加载的同时并发执行混合（简单、中等、复杂）查询

注：序号1、2、4的测试用例选取的查询可根据需要选择对应复杂度里面的任意查询或增加用例分别遍历所有的查询。

### 7.5.1.4 测试执行

表22中序号1~4测试用例采用相同的测试执行方法，如下：

- a) 根据测试用例选取对应的查询 SQL，参考增量加载性能测试执行和并发查询性能测试执行的方法，同时启动加载和并发查询；
- b) 测试过程中收集所有用于计算性能指标的数据，同时观察被测系统的资源占用率；
- c) 测试结束后，计算性能指标值，同样根据 7.5.1.4 a) 参考前文判断测试有效性；
- d) 根据需要选取其它测试数据组合，替换被测系统中的测试数据，重复上述测试步骤，以观察不同数据规模和不同数据倾斜程度对性能指标的影响。

### 7.5.1.5 结果评估

混合负载性能测试是数据增量加载性能测试与并发查询性能测试的组合测试，测试结果的有效性需要同时满足这两项测试的有效性判断条件，即：

- a) 满足增量加载性能测试的有效测试条件；
- b) 满足并发查询性能测试的有效测试条件。

测试结果分别从横向和纵向两个维度进行评估。横向评估是指比较同一性能指标项在不同被测系统中测试得到的结果，方法分别参考增量加载性能测试和并发查询性能测试的结果评估方法。纵向评估是比较混载场景相比于单独进行加载或单独进行查询时造成的性能损耗，性能损耗越小可认为被测系统混载性能越优。

## 7.6 数据操作性能测试

### 7.6.1 数据更新性能测试

#### 7.6.1.1 数据更新 SQL

选取测试数据中数据量较大的表作为更新对象，使用至少3个过滤条件作为数据更新范围限定，更新该表的至少3列数据。

**更新示例：**更新订单表的数据，以指定证券代码、交易日期范围和交易时间范围作为限定条件，更新订单表的委托价格、委托数量和买卖方向。其中委托价格更新为原值乘0.9，委托数据更新为原值5倍，买卖方向取原值的相反方向。

为了便于测试观察对比，对上述数据更新语句分别给定3组过滤条件输入，形成3个update语句，使得被更新的数据量分别为十万行、百万行和千万行。

#### 7.6.1.2 指标说明

测试输出的指标如下：

**数据更新响应时间**，是指采用指定的update语句对指定的表进行更新操作，T1是向被测系统提交update语句请求的时间，T2是被测系统执行update语句结束的时间，T2-T1为被测系统对本次数据更新操作的响应时间，单位：ms（毫秒）。

#### 7.6.1.3 约束条件

数据更新性能测试的约束条件见表23。

表 23 约束条件

序号	名称	内容描述
1	update 语句执行终端	采用被测系统自带的 SQL 执行终端或 SQL 命令行接口执行 update 语句。

#### 7.6.1.4 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统；
- b) update 语句执行终端已准备就绪。

#### 7.6.1.5 测试用例

数据更新性能测试的测试用例见表24。

表 24 测试用例描述

序号	测试用例描述
1	更新十万行数据
2	更新百万行数据
3	更新千万行数据

#### 7.6.1.6 测试执行

表24中序号1~3测试用例采用相同的测试执行方法，如下：

- a) 根据测试用例，选取对应的数据更新 update 语句，向被测系统发出执行语句请求；
- b) 记录请求发起时间和执行完毕时间，观察期间被测系统的资源占用率；
- c) 测试结束后，计算数据更新响应时间。

#### 7.6.1.7 结果评估

当测试结果满足以下条件时，视为有效测试：

- a) update 语句正常执行完毕；
- b) 测试时间内被测系统的资源消耗情况在预期范围。

测试得到的数据更新响应时间越小，可认为被测系统在该用例场景下数据更新性能越优。

### 7.6.2 数据删除性能测试

#### 7.6.2.1 数据删除 SQL

数据删除SQL与数据更新SQL采用类似的定义策略，选取测试数据中数据量最大的表作为数据删除对象，使用至少3个过滤条件作为数据删除范围限定。

**删除示例：**删除订单表中指定证券代码、指定交易日期范围和指定交易时间范围内的数据。

同样为了便于测试观察对比，对上述数据删除语句分别给定三组过滤条件输入，形成三个delete语句，使得被删除的数据量分别为十万行、百万行和千万行。

#### 7.6.2.2 指标说明

测试输出的指标如下：

**数据删除响应时间**，是指采用指定的delete语句删除指定表的数据，T1是向被测系统提交delete语句请求的时间，T2是被测系统执行delete语句结束的时间，T2-T1为被测系统对本次数据删除操作的响应时间，单位：ms（毫秒）。

#### 7.6.2.3 约束条件

数据删除性能测试的约束条件见表25。

表 25 约束条件

序号	名称	内容描述
1	delete 语句执行终端	采用被测系统自带的 SQL 执行终端或 SQL 命令行接口执行 delete 语句。

#### 7.6.2.4 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统；

b) delete 语句执行终端已准备就绪。

### 7.6.2.5 测试用例

数据删除性能测试的测试用例描述见表26。

表 26 测试用例描述

序号	测试用例描述
1	删除十万行数据
2	删除百万行数据
3	删除千万行数据

### 7.6.2.6 测试执行

表26中序号1~3测试用例采用相同的测试执行方法，如下：

- 根据测试用例，选取对应的数据删除 delete 语句，向被测系统发出执行语句请求；
- 记录请求发起时间和执行完毕时间，观察期间被测系统的资源占用率；
- 测试结束后，计算数据删除响应时间。

### 7.6.2.7 结果评估

当测试结果满足以下条件时，视为有效测试：

- delete 语句正常执行完毕；
- 测试时间内被测系统的资源消耗情况在预期范围。

测试得到的数据删除响应时间越小，可认为被测系统在该用例场景下数据删除性能越优。

## 7.7 数据导出性能测试

### 7.7.1 全表数据导出性能测试

#### 7.7.1.1 指标说明

测试输出的指标如下：

**数据导出速率**，是指在给定的约束条件下，源数据从被测系统导出到目标存储路径，平均每秒可导出的记录数，单位：万行/秒。计算公式：

数据导出速率=总记录数/(导出完成时间-导出开始时间)

#### 7.7.1.2 约束条件

全表数据导出性能测试的约束条件见表27。

表 27 约束条件

序号	名称	内容描述
1	数据导出方式	采用被测系统配套的数据导出工具或数据导出命令行接口进行导出。
2	目标存储路径	数据导出到被测系统的本地磁盘中的一个独立分区文件系统。
3	数据导出后的保存格式	数据导出后以文本文件的方式保存。

#### 7.7.1.3 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统，从中选一张数据量较大的表作为源表，例如成交表，用于后续导出测试；
- b) 数据导出工具或数据导出命令行接口已准备就绪。

#### 7.7.1.4 测试用例

数据导出性能测试的测试用例描述如表28所示：

表 28 测试用例描述

序号	测试用例描述
1	全表数据一次性导出

#### 7.7.1.5 测试执行

表28中序号1测试用例的测试执行方法如下：

- a) 使用被测系统配套的数据导出工具或命令行接口将源表一次性从被测系统导出；
- b) 记录数据导出开始到完成的时间，观察被测系统的系统资源占用率；
- c) 测试结束后，计算数据导出速率；
- d) 根据需要更换 7.7.1.5 a) 的数据导出方式，重复上述步骤进行测试，以得到不同导出方式的数据导出速率。

#### 7.7.1.6 结果评估

当测试结果满足以下条件时，视为有效测试：

- a) 导出的文本文件数据记录数与源表记录数一致；
- b) 测试时间内被测系统的资源消耗情况在预期范围。

测试得到的数据导出速率越高，可认为被测系统在该用例场景下数据导出性能越优。

### 7.7.2 查询结果数据导出性能测试

#### 7.7.2.1 查询 SQL 说明

选取测试数据中数据量较大的一张表作为源表，查询该表指定日期范围内的数据，查询结果按指定字段进行排序。

示例：查询成交表最近半年的数据，查询结果按成交日期和成交序号升序排序。

#### 7.7.2.2 指标说明

与全表导出性能测试采用相同的指标。

#### 7.7.2.3 约束条件

查询结果数据导出性能测试的约束条件如表29所示：

表 29 约束条件

序号	名称	内容描述
1	数据导出方式	采用自定义脚本调用被测系统 SQL 命令行接口的方式导出数据。
2	目标存储路径	数据导出到被测系统的本地磁盘中的一个独立分区文件系统。
3	数据导出后的保存格式	数据导出后以文本文件的方式保存。



#### 7.7.2.4 测试准备

测试准备包括以下方面：

- a) 存量数据已加载到被测系统；
- b) 编制脚本，该脚本可使用定义好的查询 SQL 语句，通过调用被测系统 SQL 命令行接口，同时执行 n 个查询结果数据导出作业，并发数量 n 可配置，最终导出的数据分别存储在 n 个文本文件中。

#### 7.7.2.5 测试用例

查询结果数据导出性能测试的测试用例描述如表30所示：

表 30 测试用例描述

序号	测试用例描述
1	单个查询结果数据导出
2	并发查询结果数据导出

#### 7.7.2.6 测试执行

表30中序号1~2测试用例的测试执行方法如下：

- a) 序号 1 测试用例的测试执行：
  - 1) 使用已准备好的查询 SQL 语句，通过脚本执行 1 个查询结果数据导出作业；
  - 2) 记录数据导出开始到完成的时间，观察被测系统的系统资源占用率；
  - 3) 测试结束后，计算数据导出速率。
- b) 序号 2 测试用例的测试执行：
  - 1) 使用已准备好的查询 SQL 语句，设置合理的并发数量 n（例如 30），通过脚本同时执行 n 个查询结果数据导出作业；
  - 2) 记录每个作业数据导出开始到完成的时间，观察被测系统的系统资源占用率；
  - 3) 测试结束后，计算每个作业的数据导出速率，最后取平均值作为最终的数据导出速率。

#### 7.7.2.7 结果评估

当测试结果满足以下条件时，视为有效测试：

- a) 测试时间内被测系统的资源消耗情况在预期范围；
- b) 导出到每个文件的数据记录数与对应的查询结果数据记录数一致。

测试得到的数据导出速率越高，可认为被测系统在该用例场景下的数据导出性能越优。

## 8 扩展测试

### 8.1 扩展测试概述

大数据平台的性能表现与大数据平台的一些技术特性的设计有关，例如表的存储方式（行存、列存），数据模型的索引、分区键和分布键等特性的设计，被测系统资源队列的设计等对关键设计的性能进行扩展测试测评不同的设计方式对性能的影响。

### 8.2 索引测试

合理创建和利用索引以提高系统查询性能，同时带来了额外的系统开销，执行表31中的测试内容评估索引对被测系统性能的影响。

表 31 测试内容

序号	测试内容
1	测试使用和不使用索引，对数据加载、查询和操作的性能影响。
2	测试索引列数量不同时，对数据加载、查询和操作的性能影响。
3	测试不同类别的索引对查询性能的影响。

### 8.3 分区键测试

合理设计分区键提升查询和加载性能，执行表32中的测试内容评估分区键对被测系统性能的影响。

表 32 测试内容

序号	测试内容
1	测试使用和不使用分区键，对数据加载、查询和操作的性能影响。
2	测试多级分区键对数据加载、查询和操作的性能影响。

### 8.4 分布键测试

分布键作为分布式数据系统的数据分布策略依据，结合业务数据的倾斜分布特征，合理设计分布键提升查询和加载性能，执行表33中的测试内容评估分布键对被测系统性能的影响。

表 33 测试内容

序号	测试内容
1	测试采用默认分布键和自定义分布键，对数据加载、查询和操作的性能影响。
2	测试不同类别的分布键（例如 Hash 分布和随机分布）对数据加载、查询和操作的性能影响。
3	测试分布键包含的列数量不同时，对数据加载、查询和操作的性能影响。

### 8.5 列存储测试

列存储已于压缩数据节省空间并提升OLAP类业务性能，但会导致数据表局部增删改性能下降，执行表34中的测试内容评估列存储对被测系统性能的影响。

表 34 测试内容

序号	测试内容
1	测试使用列存储对数据加载、查询和操作产生的性能影响。
2	测试使用压缩列存储对数据加载、查询和操作产生的性能影响。

### 8.6 资源队列测试

有些大数据平台提供了资源队列或其它类似的特性，限制特定用户或特定作业的资源消耗，合理分配资源、提升整体系统性能。结合业务特点合理设计资源队列，执行表35中的测试内容评估资源队列对被测系统性能的影响。

表 35 测试内容

序号	测试内容
1	根据功能域划分资源队列，例如对数据加载、查询和操作分别设置资源队列，测试该设计对各项性能指标的影响。
2	根据查询的复杂程度和重要程度设置资源队列，例如将高复杂度（高资源消耗）低优先级的查询限制在独立的资源队列中执行，测试该设计对各项性能指标的影响。
3	针对 ad-hoc 查询设置独立的资源队列，测试该设计对各项性能指标的影响。

## 附录 A

(资料性)

## 平台、术语、工具名称解释及 TCP 基准测试套说明

## A.1 平台、术语和工具名称解释

本文件中涉及的平台、术语和工具名称的解释见表A.1。

表 A.1 平台、术语和工具名称解释

名称	内容
Hadoop平台	Hadoop平台是Apache软件基金会管理的开源软件平台，是在分布式服务器集群上存储海量数据并运行分布式分析应用的一个平台。其核心部件是HDFS（分布式文件系统）和MapReduce（分布式计算模型）。
MPP数据库	MPP数据库是指采用了Massively Parallel Processor大规模并行处理（缩写为MPP）架构的数据库，在MPP数据库非共享集群中，每个节点都有独立的磁盘存储系统和内存系统，业务数据根据数据库模型和应用特点划分到各个节点上，每台数据节点通过专用网络或者商业通用网络互相连接，彼此协同计算，作为整体提供数据库服务，具有可伸缩性、高可用、高性能、优秀的性价比、资源共享等优势。
管理节点	管理节点，分布式系统中对集群进行管控和分派作业或计算任务的服务单元。通常管理节点上只存储元数据，不存储业务数据。
数据节点	数据节点相对于管理节点而言，是分布式系统中参与执行作业或具体计算任务、存储业务数据的服务单元，也称为计算节点。
分布键	分布键是一张数据表中用于确定特定数据行在数据库中的存储位置的列（或列的组合）。
分区键	分区键是一张数据表中用于确定特定数据行在数据库中所属数据分区的一个或多个列的有序集合。
数据倾斜	数据倾斜是指在分布式数据存储系统中同一张表的数据非均匀的分布到不同数据节点进行存储的形态。
Kafka	Kafka是Apache软件基金会开发的一个分布式消息订阅发布系统，具有高吞吐量、高可用和实时性等特点。
Nmon	Nmon是一个分析 AIX 和 Linux 性能的免费工具，提供监视和分析性能数据的功能。
Atop	Atop是一款用于监控Linux系统资源与进程的工具，它以一定的频率记录系统的运行状态，采集系统资源使用情况和进程运行情况。
Sysbench	Sysbench是一款开源的多线程性能测试工具，可以执行CPU/内存/线程/IO/数据库等方面的性能测试。
Netperf	Netperf是一款网络性能的测量工具，主要针对基于TCP或UDP的传输。
IOPS	每秒钟进行的输入、输出操作
IOWait	CPU因等待输入、输出而产生的时间消耗

## A.2 TPC基准测试套

与大数据相关的常见的 TPC 基准测试套见表 A.2，引用自 TPC 官网 <http://www.tpc.org/tpcc/>。

表 A.2 TPC 基准测试套说明

名称	内容
TPC-DI	基于ETL场景的基准测试，它将从联机事务处理 (OLTP) 系统中提取的数据与其他数据源进行组合和转换，并将其加载到数据仓库中。源和目标的数据模型、数据转换和实现规则被设计为广泛地代表现代数据集成需求。
TPC-DS	基于决策支持场景的基准测试，它对决策支持系统的几个普遍适用的方面进行建模，包括查询和数据维护，测量单用户模式下的查询响应时间、多用户模式下的查询吞吐量 and 数据维护性能等。
TPC-H	基于决策支持场景的基准测试，它由一套面向业务的特别查询和并发数据修改操作组成。测试过程模拟了一个执行大量数据、高复杂度的查询并回答关键业务问题的决策支持系统，测量在不同数据规模下系统对单个查询的处理能力以及多并发查询的吞吐量。
TPCx-BB	衡量基于Hadoop的大数据系统的性能。它通过建立一个具有实体和在线商店的零售商业模式，执行30个高频度分析性查询来测量系统性能。这些查询包括以结构化数据为查询对象的SQL，以半结构和非结构化数据为处理对象的机器学习算法。
TPCx-HS (V2)	衡量基于Hadoop的大数据系统的性能，它以供应商中立、技术严谨、直接可比的方式，测试MapReduce (MR2) 的批处理性能，以及基于Apache Spark流计算平台的Hadoop实时分析性能。

### 参 考 文 献

- [1] GB/T 15532-2008 计算机软件测试规范
  - [2] GB/T 35589-2017 大数据 技术参考模型
  - [3] GB/T 35295-2017 大数据 术语
  - [4] JR/T 0145-2016 资本市场交易结算系统核心技术指标
-