



QianBase 技术白皮书 1.6.6

(含附录)

2020 年 12 月

© Copyright 2019-2020 贵州易鲸捷信息技术有限公司

公告

本文档包含的信息如有更改，恕不另行通知。

保留所有权利。除非版权法允许，否则在未经易鲸捷预先书面许可的情况下，严禁改编或翻译本手册的内容。易鲸捷对于本文中所包含的技术或编辑错误、遗漏概不负责。

易鲸捷产品和服务附带的正式担保声明中规定的担保是该产品和服务享有的唯一担保。本文中的任何信息均不构成额外的保修条款。

声明

Microsoft® 和Windows® 是美国微软公司的注册商标。Java® 和MySQL® 是Oracle及其子公司的注册商标。Bosun是Stack Exchange的商标。Apache®、Hadoop®、HBase®、Hive®、openTSDB®、Sqoop® 和Trafodion® 是Apache软件基金会的商标。Esgyn和QianBase是易鲸捷的商标。

目录

目录.....	1
1 QianBase 的起源.....	1
2 NoSQL 变革和挑战.....	3
3 传统 RDBMS 和新型 NoSQL 的完美结合 - QianBase.....	5
4 QianBase 如何满足这些需求?	6
4.1 流式处理.....	6
4.2 运营型工作负载.....	8
4.2.1 与 HBase 深度集成.....	9
5 满足企业当前和未来需求.....	11
6 总结.....	13
7 关于易鲸捷公司和 QianBase.....	14
8 附录.....	15
8.1 QianBase 软件架构概览.....	18
8.2 流程概览和 SQL 执行流.....	20
8.3 优化器技术.....	21
8.3.1 可扩展的优化器技术.....	21
8.3.2 基于统计信息的优化执行计划.....	22
8.3.3 优化器差异化功能.....	23
8.4 并行数据流 SQL 执行器技术.....	26
8.5 支持事务型/运营型.....	29
8.6 数据连接服务.....	31
8.6.1 抽象层.....	32

8.6.2	负载均衡.....	32
8.6.3	扩展性.....	33
8.6.4	高可用.....	34
8.6.5	数据本地化.....	34
8.7	功能概览.....	36
8.7.1	全面支持 ANSI SQL.....	36
8.7.2	存储过程.....	37
8.7.3	SQL 函数.....	38
8.7.4	触发器.....	38
8.7.5	用户自定义函数.....	38
8.7.6	Table UDF.....	39
8.7.7	加载工具.....	39
8.8	管理性.....	40
8.8.1	工作负载可见性和控制.....	40
8.8.2	系统检测和健康检查.....	40
8.8.3	工作负载管理系统.....	41
8.8.4	多租户.....	42
8.8.5	其它功能.....	43
8.9	与 HBase 深度集成.....	45
8.9.1	行键 salting.....	47
8.9.2	Divisioning.....	48
8.10	多维访问方法 (Multidimensional Access Method)	49
8.11	分布式事务管理 (Distributed Transaction Management)	50
8.11.1	行锁.....	52
8.12	高可用和数据完整性功能.....	53

8.12.1	跨数据中心双活零丢失事务.....	54
8.12.2	增量备份和恢复.....	54
8.12.3	弹性扩展.....	54
8.13	安全性.....	56
8.13.1	授权.....	57
8.13.2	验证.....	58
8.13.3	账户.....	60
8.13.4	数据保护.....	61
8.14	运营性能.....	64
8.14.1	YCSB 基准.....	64
8.14.2	Order Entry 基准.....	65

1 QianBase 的起源

数据库引擎从起始到成熟需要很长时间，各式各样的、来自真实客户现场的工作负载能在不同程度上促进数据库引擎不断成熟，并在以下核心技术上持续进步：查询优化技术、事务和查询处理、支持的函数和功能。这些技术进步最终体现于满足用户 SLA 要求（规模、并发性和性能）的能力。同时，为了达到这些要求需要管理数据和工作负载、具备多租户功能，并满足安全性、一致性和可用性的要求。即使 Oracle、IBM DB2、Microsoft SQL Server 和 Teradata 已在数据库引擎领域研究了几十年，也难以满足所有用户的工作负载需求。然而，QianBase 架构的高灵活性和强扩展性可以最大化地满足用户的所有需求。QianBase 汇集了大量的专利创新，在它的发展成熟过程中，众多优秀的工程师贡献了毕生的心血。

Apache Trafodion 的鼻祖是 Tandem 的 NonStop SQL/MX (1987 年)，它是一个 ANSI SQL 数据库，建立在可扩展和高可用的 MPP (Massively Parallel Processing) 架构上。MX 是 Massively Extensible，代表可扩展架构。惠普公司在 NonStop SQL/MX 的基础上开发了一个旨在处理高强度企业级数据仓库负载的 MPP 数据库，这是设计 MPP 构架的初衷，而 Trafodion 直接继承该系统。因此，Apache Trafodion 能处理 OLTP、ODS、BI/分析应用程序的工作负载。

Trafodion 继承了 SQL/MX 的所有特性与核心代码，还直接继承了 NonStop SQL MX 的分支 Neoview 和其后续产品 SeaQuest（该产品线拥有数百项专利、超过 3 亿美元的投资以及超过 20 年的研发投入），因此还能处理 ODS 和 BI/分析应用程序的工作负载。2014 年 6 月，惠普公司将 Trafodion 开源。2015 年 5 月，Trafodion 成为 Apache 家族的新成员。2015 年 7 月，易鲸捷从惠普公司中剥离出来，成立了易鲸捷公司并拥有 30 名原 Neoview/SeaQuest 项目组成员，在数据库领域继承了 30 余年的

丰富的研发经验和深厚的研发实力。QianBase 架构在 Apache Trafodion 上，是企业服务的商业版数据库。

2 NoSQL 变革和挑战

NoSQL 为 Big Data 计算带来了以下前沿变革：

- **降低成本**：开源软件和标准通用硬件。
- **Schema 灵活性和 schema on read**：对于面临动态变化的成长型公司很重要。
- **增强弹性扩展**：满足大数据的高容量和速度。
- **支持半结构化和非结构化数据**：处理各类数据。
- **提高数据模型的灵活性**：处理各类工作负载。
- **并行处理用户代码**：利用 MapReduce 实现。
- **高可用模式**：适用于低廉且易出故障的通用商业服务器。

这些变革催生了几类典型的 NoSQL 解决方案，每一类解决方案都设计了自己独特的 API 接口，并仅用于解决部分特定的大数据应用场景。典型用例包括实时流和分析机器学习应用程序。更多关于这些用例的信息，请参阅《数据管理的未来》。

随着 NoSQL 和 Hadoop 越来越受欢迎，越来越多的应用程序开始迁移到这些环境。但是，当新型初创公司走向成熟阶段后，其运营型工作负载的需求不断增加，更加需要那些传统 RDBMS 所能提供的能力。虽然大型企业未面临和初创公司一样的挑战，但也想利用这些新技术的优势，并希望继续使用 SQL，原因如下：

- 企业普遍使用 SQL，这能简化开发。
- 企业已经使用了大量 SQL 相关的工具和应用程序生态系统。
- 尽管事务处理有一定开销，但很多情况下仍需要事务支持。
- SQL 引擎能高效处理 join 操作，很多情况下仍需要 join 的支持。
- 企业开发人员需要进行相对困难的应用程序开发，或者编写 MapReduce/Spark 作业，而使用 SQL 更简单。

- 预定义数据表列有很多优势，因此用户可以使用数据类型检查和约束来保证数据质量，而且无需牺牲 schema 灵活性。
- 促进了应用程序之间的元数据统一管理和规范。

3 传统 RDBMS 和新型 NoSQL 的完美结合 - QianBase

虽然 NoSQL 运动是革命性的技术创新，但现在对传统 SQL 和 RDBMS 的需求在复苏，两者结合的新产品可以最好地满足需求。这催生了新型的强大数据库引擎 QianBase，它整合了 Apache HBase 原生的 NoSQL 能力。

QianBase 在 HBase 之上支持结构化的关系型抽象、Google Big Table 模型（使用原生 HBase）。

4 QianBase 如何满足这些需求?

4.1 流式处理

Zero Latency 公司实现了梦寐以求的愿景：实时处理高速捕捉的数据。数据批处理导致的延迟已不能被接受，许多应用程序使用流式编程模型（而不是批处理）处理注入的数据。流式处理包括两方面：一方面，保存注入的数据；另一方面，如果数据注入速度很快，处理前无法保存数据，必须进行实时处理。

HBase 擅长高速数据写入，很适合被 QianBase 使用，用于处理运营型工作负载。因此，QianBase 能满足高速注入 Big Data 的需求，并提供全面的 ACID 支持。

Kafka 作为一种缓冲，能避免大数据系统因为数据注入的瞬时增大而导致系统崩溃，这在数据实时入库保存或实时分析的场景下均可能出现。为了更好地保存并简化数据注入，QianBase 与 Kafka 进行集成。该集成采用了 QianBase 的 Table UDF，能流式地处理复杂场景或进行分析，同时，它还能利用引擎的所有 SQL 能力。

数据导入进 HBase memstore 或内存，QianBase 数据注入速度约为 74 μ s/行。任何需要访问数据的查询均能实现立即访问。只有那些能在 100 μ s 或 0.1ms 内处理数据的应用程序将先处理数据，再保存数据（如需要）。另外，Netflix 每天通过 3000 多个代理的 24 个集群获取 7000 亿条信息（=370 μ s/信息）；LinkedIn 通过 1100 个代理传输 1300 万条信息/秒（=85 μ s/信息）；Facebook Messenger 和 WhatsApp 每天处理 600 亿条信息。

如果您的需求大于 Netflix、LinkedIn 或 Facebook，在不使用 Kafka 的情况下，QianBase 也能满足您高速注入数据和存储数据的需求。另外，QianBase 支持事务插入和更新

(Kafka 不支持 SQL) , 还能为数据注入速度更快的场景使用 [bulk load](#) 进行批量加载或 mini-batch 加载。

作为通用参考, QianBase 的流式数据注入速度约为 50,000 行/秒/节点, bulk load 数据注入速度约为 100,000 行/秒/节点。

4.2 运营型工作负载

传统关系型数据库处理传统的运营型工作负载，但由于 Hadoop 能大幅降低成本、减少供应商锁定和无缝扩展以处理更多工作负载、存储更多数据，因此，使用 Hadoop 来处理工作负载的需求越来越大。

一般而言，受事务保护的运营型工作负载创造收益、接触客户、预测前景或经营业务，所以对公司而言，它是非常重要的部分。通常情况下，该工作负载在响应时间（次秒级）、事务数据完整性、用户数量、并发性、可用性和数据量方面都有严格要求。随着物联网的兴起，设备数量和种类的不断增长促进了事务和数据的增长，同时还改变了数据的类型，这些数据需要被获取并作为事务的一部分被使用。运营数据发展迅速，包括各式各样的数据格式和类型，因此，下一代运营应用程序需要使用结构化、半结构化和非结构化数据类型。例如，半结构化的日志数据和结构化的事务数据、从设备获取的物联网数据、文字信息与情感和其它分析评论、需要被搜索的文件或文本信息以及与结构化数据/元数据相关的视觉图像等。

然而，Hadoop 无法满足前述所有需求，例如，事务支持、灾备方案、强数据完整性、次秒级响应时间、运营查询优化和管理工作负载（根据优先级分配任务且并发地执行）。QianBase 能解决以上所有问题，它强大的 DBMS 支持应用程序和数据，减少架构在 Hadoop 上软件之间的数据延迟和冗余。

QianBase 提供全面的 SQL 数据库功能，支持公司利用现有 SQL 技能，提高开发人员产出和处理工作负载的效率。在处理运营型工作负载时，QianBase 具备以下优势：

- 与 HBase 深度集成，充分发挥 HBase 优势。
- 低延迟读写事务优化，支持事务 SQL 工作负载的高并发和快速响应。
- JNI 客户端（Type 2 JDBC 驱动）直接访问 HBase 服务，减少服务时间。

- 编译 SQL 计划缓存, 减少重新编译的开销。缓存命中策略不仅是简单文字匹配, 它在编译过程的多个阶段进行。如果谓词的代入值不同, 导致基数有很大差异, 则 QianBase 将为相似查询缓存出不同计划。
- 多维访问方法 (Multidimensional Access Method) 专利技术:
 - 即使索引的前导列没有谓词, 也能高效地使用索引。
 - 大幅减少使用二级索引的需求。
 - 帮助用户自动实现 salting 和 divisioning, 实现数据分区。
- QianBase 扩展 HBase 时为它添加了 ACID (原子性、一致性、隔离性和持久性) 事务保护, 从而确保数据跨多行、表和 SQL 语句 (使用完全分布式事务管理架构) 的一致性。QianBase 与 HBase 深度集成, 能在高并发场景时提供高扩展性、高效率的执行和事务一致性。
- 提供高可用和数据完整性功能, 对运营型应用程序至关重要。
- 跨数据中心的全面双活分布式事务支持, 扩展读/写工作负载并提供灾难恢复零事务丢失机制。
- 弹性扩展。在不宕机和同时处理事务和查询的情况下, 跨集群的节点和存储可以大幅增加或减少。下一个事务或查询将利用重新配置的计算机或存储资源

4.2.1 与 HBase 深度集成

Trafodion 宣布开源时, 当时已存在基于 Big Data Hadoop 平台的用于满足 BI 和分析需求的 SQL 解决方案, 但未有像 QianBase 一样能同时处理事务型和运营型工作负载的解决方案。为了处理这些工作负载, QianBase 专注于使用强大的数据库引擎、与 HBase 深度集成、提供完整可扩展的分布式事务管理基础架构并尽可能减少开销。HBase 能处理低延迟工作负载、支持高速数据注入、有利于运营型查询的高效的键

结构以及具备 schema 灵活性和从 HDFS 继承的 NoSQL 的优势。HBase 还拥有压缩、加密、高可用复制和弹性扩展等功能。

HBase 不仅为 QianBase 提供了用于保存结构化关系抽象的存储引擎，还提供了 Big Table 数据模型（键-列-值，即宽-列存储）处理半结构化数据（无法使用关系模型表示）。对运营型工作负载而言，QianBase 是 SQL 和 NoSQL 能力的完美结合。QianBase 不仅能在相同查询中对 Trafodion 结构化表和原生 HBase 表进行查询，还能在相同 ACID 事务中对它们进行更新。HBase 表可以定义为 QianBase 外表，您能通过这些表创建视图。

最后，选择 HBase 的重要原因在于 Hadoop 已经是大数据行业标准，HBase 是 Hadoop 大型生态系统的一部分，用户能利用 Hadoop 生态系统来满足各类需求。

5 满足企业当前和未来需求

现在我们来讨论企业部署 Big Data 技术的其它重要方面：

- 熟悉 Big Data 的 4V——Volume（容量）、Velocity（速度）、Variety（种类）与 Veracity（精确）：
 - **容量：** QianBase 架构在 Hadoop 上，拥有大规模并发处理能力并具备高可扩展性。因此，QianBase 擅长处理海量数据。更多关于[架构](#)和[性能](#)的信息，请参见[附录](#)。
 - **种类：** QianBase 支持多数据模型，并在将来支持更多数据类型。QianBase 不仅支持多种数据结构和模型，还支持 OLTP 和运营负载。
 - **精确：** 精确与数据治理、transparency 和 lineage 息息相关，QianBase 能使这些流程的处理变得更简单。数据分析有利于 QianBase 处理半结构化和非结构化数据。分布式 ACID 事务处理、强大的灾难恢复能力与强制数据类型和检查约束使 QianBase 能保证数据一致性和数据质量，并在保证数据精确方面发挥了至关重要的作用。
 - **速度：** 数据速度是 Big Data 平台为 IoT 相关应用处理数据的速度。数据速度有两方面，QianBase 在处理高并发时，性能可以保证达到 SLA 的要求。
- 大幅降低总成本。
 - 开源软件和定价模式。
 - 在标准通用硬件而不是在专有硬件上运行。
 - 使用现有 SQL 技能和工具。

- 企业级**管理**和运营基础架构：企业能成熟地管理和运营自身的应用程序和数据部署，并期望使用相同的技能管理大数据工具。为了解决这些需求，QianBase 提供企业级**管理**和**可用**能力。

6 总结

QianBase 是一个能运行所有事务和运营负载的并行数据库引擎，其集成了 NoSQL 和 SQL 技术，为您的下一代应用程序提供平台，并提供[最佳性能](#)。

更多关于本文讨论的详细信息，请参阅[附录](#)。

7 关于易鲸捷公司和 QianBase

易鲸捷公司的使命是采用颠覆性的技术和产品引领大数据市场的未来， QianBase 能处理 OLTP 和各类运营型工作负载。

易鲸捷公司的顶级产品——QianBase 企业版，是一款专门处理 Big Data 的安全可靠的企业级 HTAP SQL 引擎。易鲸捷在美国硅谷、中国上海、北京和贵阳设立了分公司，为 QianBase 提供支持、服务和培训，使 QianBase 能满足企业的各类需求。

更多信息，请访问 www.esgyn.com 或发邮件至 info@esgyn.com。

© 2015-2019 贵州易鲸捷信息技术有限公司

8 附录

该附录为正文的参考资料，包括 QianBase 的技术细节、架构和关键功能。

架构

- QianBase 软件架构概览
- 流程概览和 SQL 执行流
- 优化器技术
 - 可扩展的优化器技术
 - 基于统计信息的优化执行计划
 - 优化器差异化功能
 - Large Scope Rules
 - Skew Buster
 - Adaptive Segmentation
- 并行数据流 SQL 执行器技术
- 支持事务型/运营型和/BI/分析型查询
- 数据连接服务

功能

- 功能概览
 - 全面 ANSI SQL 语言支持
 - 存储过程
 - SQL 函数
 - 触发器
 - 用户自定义函数

- Table UDF
- 加载工具

管理

- 工作负载可见性和控制
- 系统检测和健康检查
- 工作负载管理系统
- 多租户
- 其它功能

运营功能

- 与 HBase 深度集成
 - 行键 salting
 - Divisioning
- 多维访问方法 (Multidimensional Access Method)
- 分布式事务管理
 - 行锁
- 高可用和数据完整性功能
 - 跨数据中心双活零丢失事务
 - 弹性扩展
- 安全性

运营性能

- YCSB 基准
- Order Entry 基准

正如前文所述，QianBase 的架构为了满足 Big Data 的 4V 需求。如图 9-1 所示，QianBase 能处理各类工作负载和数据流。根据不同用户的需求，QianBase 将进行有针对性的部署，并愿意与您一起探讨制定适合您 Big Data 需求的解决方案。

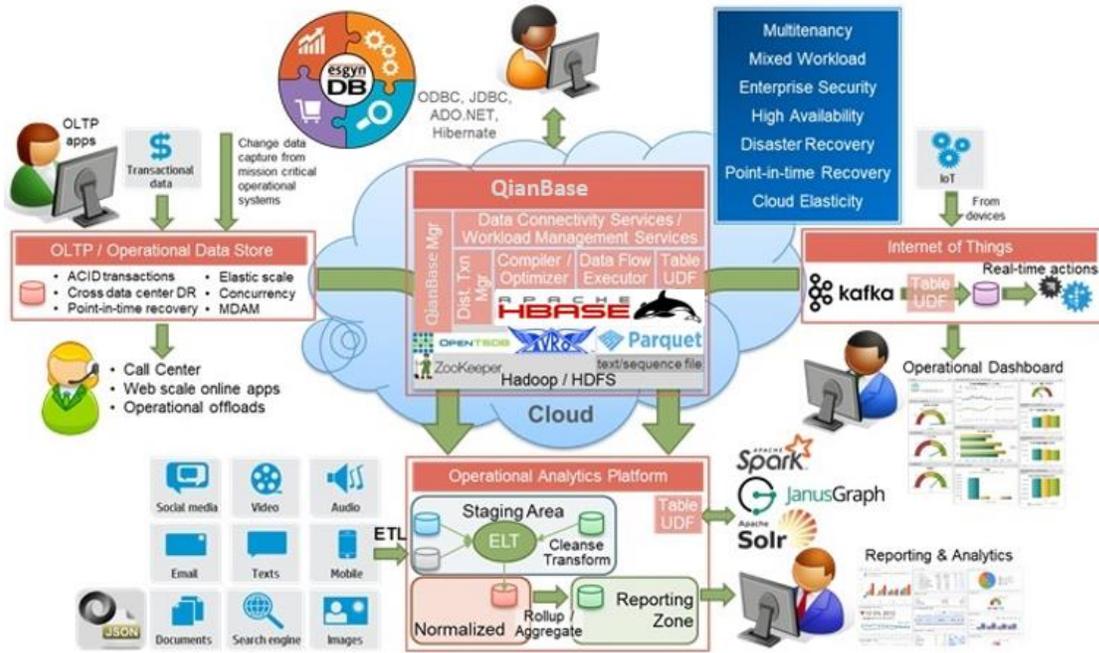


图 8-1 QianBase 处理各类工作负载和数据流

8.1 QianBase 软件架构概览

QianBase 软件架构包括 3 层：客户端服务层、SQL 数据库服务层和存储引擎层。

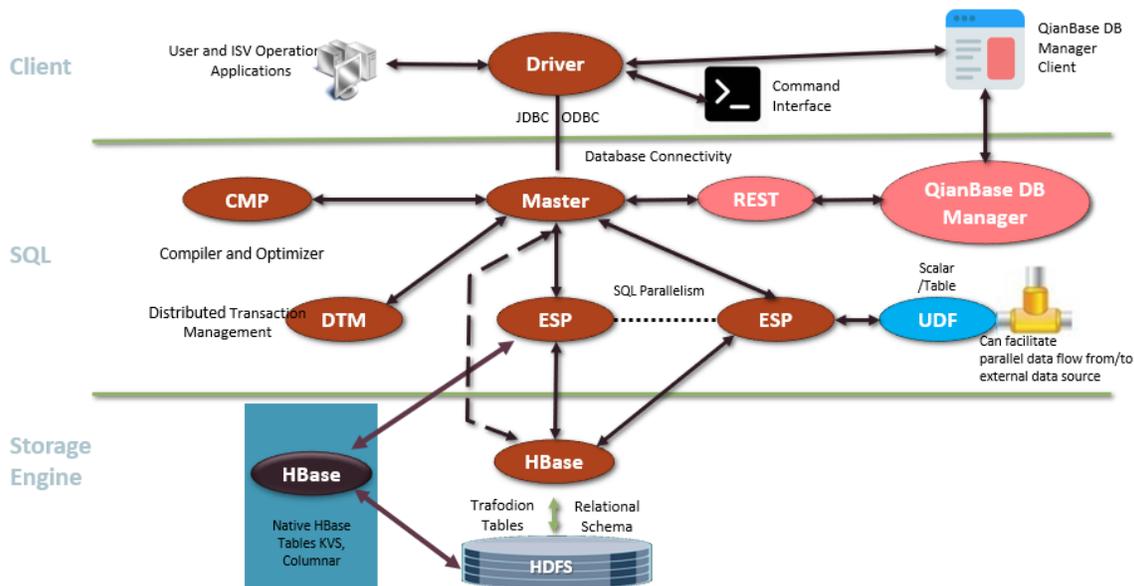


图 8-2 QianBase 软件架构

第一层是应用程序所在的客户端服务层。应用程序可以由用户编写或通过第三方 ISV 工具/解决方案实现。您能使用 QianBase 提供的 Windows 或 Linux 客户端驱动、通过标准 ODBC/JDBC 接口访问 QianBase 数据库服务层。QianBase 支持 type 2 JDBC、type 4 JDBC 和 ADO.NET 驱动程序。根据特定的需求（响应时间、连接数量、安全要求和其它因素），您可以选择合适的驱动程序类型。

第二层是 SQL 数据库引擎层。该层包括所有 QianBase 服务，封装了所有管理 QianBase 对象和高效执行 SQL 数据库请求的服务。服务包括连接管理、SQL 语句编译和创建最优执行计划、SQL 执行（串行和并行）、事务管理和工作负载管理。

第三层是存储引擎层，包括 QianBase 使用的标准 Hadoop 服务(HDFS 和 Zookeeper)。

QianBase 对象存储在原生 Hadoop 数据库结构中，包括以下格式：

- HBase，提供 Big Table 或 wide-column key-value，数据模型。

- 缓存文本文件，例如，字符分隔值 (csv) 或日志数据。
- 键值序列文件。

QianBase 处理从应用程序传来的 SQL 请求，并将这些请求透明地转换成底层数据格式所需的原生接口调用。QianBase 在 HBase 之上提供了关系型 schema 抽象，所以，QianBase 能通过使用熟悉的 DDL/DML 语法（对象命名、列定义和数据类型支持）支持传统关系型数据库对象（表、视图、二级索引）。另外，QianBase 还支持将 HBase 和 Hive 的原生表作为 QianBase 的外表。

8.2 流程概览和SQL执行流

QianBaseSQL 层由几种服务或处理连接请求和 SQL 执行的流程组成。

- 流程从应用程序或第三方客户端软件开始。Windows 或 Linux 客户端通过 ODBC/JDBC/ADO.NET 驱动程序访问 QianBase。
- 当客户端请求一个连接时，QianBase 的 DCS (Database Connection Services) 处理请求并向 QianBase Master SQL 进程分配连接。为了平衡负载，QianBase 使用 Zookeeper 协调并分配跨集群的连接服务，并确保一旦出现 Master 进程失败，其它客户端能立刻重新连接。
- Master 进程协调执行来自客户端应用程序的 SQL 语句。
- Master 调用 COP (Compiler and Optimizer Process) 解析和编译 SQL 语句，并生成最优执行计划。
- 如果最优执行计划调用并行执行，则 Master 将工作平均地分配给 ESP (Executor Server Processes)，此时，ESP 代表 Master 进程，并行地处理工作，处理的结果将传回至 Master 进行合并。复杂查询（例如，大型 n-way 联合或聚合）可能会调用多个 ESP 层。如果生成串行计划，为了提高性能，Master 将直接调用存储引擎服务。
- 对于分布式事务保护而言，QianBase 将调用 DTM 服务，确保 Hadoop 集群事务的 ACID 一致性。通过使用 HBase 的协处理器管理事务上下文、检测事务冲突、写入事务数据和执行事务恢复与 HBase 进行深度集成，DTM 管理分布在多个 Region Server 上的事务。

- 最后，为了完成 I/O 请求，Master 或 ESP 进程使用原生 API 调用 vanilla 存储引擎和 HDFS 服务。QianBase 使用 HBase 的过滤器或协处理器，尽可能将 SQL 执行下推至存储引擎层。

8.3 优化器技术

与其它 SQL-on-Hadoop 项目或产品相比,优化器技术是 QianBase 的最大优势之一。优化器主要有两方面优势:一方面, QianBase 能扩展优化器,以适应更改和新增的改进;另一方面, QianBase 拥有复杂和成熟的优化器,能选择最优执行计划。

8.3.1 可扩展的优化器技术

如前文所述, QianBase 的最大优势是优化器技术,该技术的首要方面是 QianBase 能扩展优化器。QianBase 优化器基于 Cascades 优化框架, Cascades 是最高级和可扩展的优化器框架。Cascades 框架是混合优化引擎,它结合了逻辑和物理运算符转换规则与成本模型,生成 QianBase 优化器。



图 8-3 QianBase 优化器

QianBase 能轻松添加或更改新规则或新成本模型,生成改进的优化器。因此,优化器能被快速发展提高,新运算符能被快速添加或更改,从而生成最佳的 SQL 优化计划。

8.3.2 基于统计信息的优化执行计划

如前文所述, QianBase 的最大优势是优化器技术, 该技术的另一方面是 QianBase 拥有复杂和成熟的优化器。因此首先, 您需了解优化器的各类组件:

- **SQL 规范器** —— 已解析的 SQL 语句将被传送至规范器, 它将 SQL 无条件转换至标准格式 (包括子查询转换), 并将 SQL 翻译成能被内部优化器优化的格式。
- **SQL 分析器** —— 分析优化器规则使用的等价 join 连接模式、table access paths 和分区匹配信息等。结果将传送至计划生成器, 以计算不同计划的开销。
- **表统计信息** —— 等高直方图表示列数据和列之间关联的数据分布。大表使用样本减少生成统计信息的开销。
- **基数估计器** —— 计算运行过程中的中间结果的基数、数据倾斜和直方图。
- **成本预估器** —— 在考虑数据倾斜的前提下, 为每个运算符预估节点、I/O 和消息的成本。
- **计划生成器** —— 使用成本预估优化器评估等价计划, 并选择成本最低的计划。优化器将尽可能选择 SQL 下推、排序删除、使用内存存储而避免溢出到磁盘。另外, 它还确定优化的并发度和串行计划。

为了计算出最优并行度, 优化器从查询的历史执行记录中, 获取运算符级别的统计信息。由于减少了进程开销, 降低了 27% 的查询平均用时, 因此提高了 50% 的并发度。

总之，优化器选择执行计划，在占用系统最少资源的情况下做出最快响应，并优化运营型和报表型工作负载。

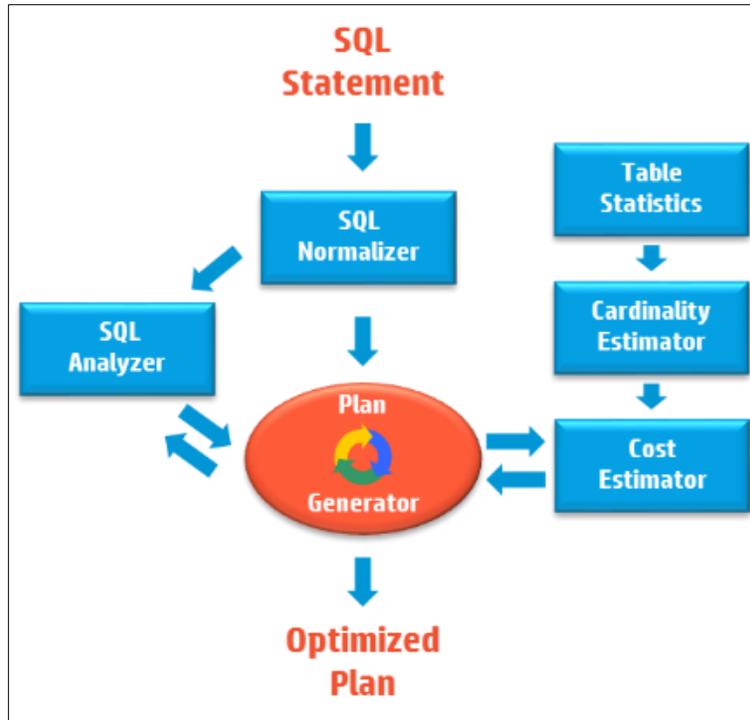


图 8-4 QianBase 优化器各类组件

8.3.3 优化器差异化功能

8.3.3.1 Large Scope Rules

其它数据库使用 Cascades 框架，但由于复杂查询的优化计划的搜索空间可能很大，这将造成编译时间过长。在这种情况下，优化器使用规则或经验规则来减少搜索空间。对此，QianBase 添加了 Large Scope Rules 处理该问题。Large Scope Rules 检测查询模式并能使用它们大幅减少搜索空间。例如，如果存在 join 一个事实表的几个小维度表，大部分优化器将执行全量事实表扫描，再将扫描的结果和这些维度表进

行 hash join。然而，QianBase 先对维度表应用谓词，对过滤结果计算笛卡尔积，再对笛卡尔积与事实表执行 nested join。这能快速生成计划，并高效访问事实表。

QianBase 使用这些技术和分支限定 (Branch and Bound) 策略，在查询开始时，QianBase 尽力获取最佳计划 (如前文所述)，并在不能找到更优计划 (达到收益递减点) 时，立即停止搜索更多计划。

8.3.3.2 Skew Buster

查询执行的一个难题是如何处理数据中的倾斜，尤其是当大量节点横向扩展时。数据倾斜问题不仅会在存储层出现 (QianBase 通过分区来处理该问题)，还可能在查询操作层出现，例如，执行 join 或聚合操作时。在这种情况下，您能在 100 个节点上运行一个查询，但由于倾斜，大部分数据只由单个或少数几个节点处理。这可能导致查询执行时间过长和造成集群资源倾斜，并影响其它所有并行运行的查询。

通过使用等高直方图 (擅长检测倾斜) 计算在执行树所有层的基数，QianBase 能检测出执行树所有层的倾斜。与处理其它值的方法不同，QianBase 对内表和外表混合使用 repartition 和 broadcasting 方法处理倾斜值并消除倾斜，从而在最短的时间内执行查询、利用最小系统资源且不影响其它并发的工作负载。

8.3.3.3 Adaptive Segmentation

QianBase 使用不同并发度处理查询的不同阶段，通过直方图统计数据预估行 (执行的每一步需要处理的行。例如，join 和 aggregation 等) 的基数，再根据该基数确定并发度。根据所有这些信息，QianBase 确定整个查询的并发度。例如，根据预估处理该查询所需的行数，QianBase 可以智能地决定在 5 个节点上执行查询，而不是使用全部 100 个节点，这样不仅能使用合适的系统资源 (例如，内存、计算和消息)，还有利于处理更多并行执行的查询。如果查询中存在倾斜，降低并发度能提高整体

资源的利用率。与所有节点用于执行所有查询的数据库相比，如果 QianBase 一个节点失败了，它不会影响所有查询。

图 8-5 为 7 个在 128 节点集群上运行的查询，它们分别以 32、64 和 128 并发度执行。为了充分利用所有 segment，QianBase 限制 adaptive segmentation 的 segment 的数量，以简化负载均衡。由于集群能通过横向扩展（增加新节点）获取更快计算、更多内存和 I/O 带宽，因此该功能非常重要，使用它能比在扩展集群前运行更多查询。

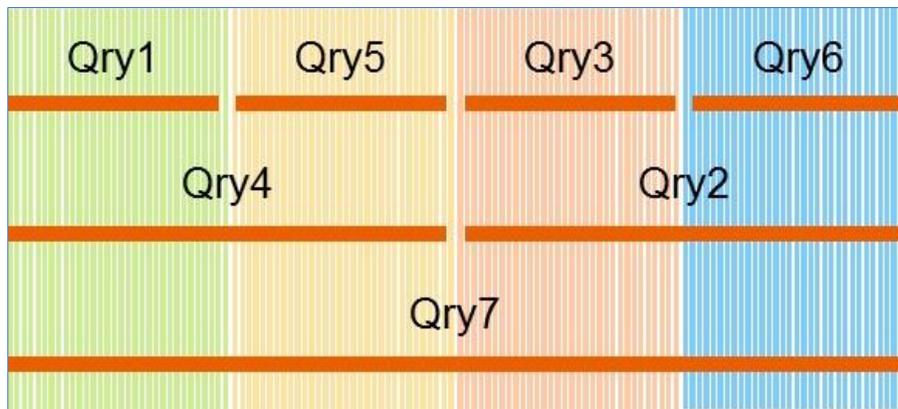


图 8-5 Adaptive Segmentation

8.4 并行数据流SQL执行器技术

QianBase 的 SQL 执行器使用数据流和调度器驱动任务模式执行优化的查询计划。该计划的每个运算符都是一个独立的任务，数据通过内存队列（上和下）或进程间通信流经运算符。任务间的队列允许运算符一次交换多个请求或结果行。调度器协调任务执行，并运行任务（一旦某一输入队列中存在数据）。针对大型 hash join 或 sort，QianBase 检测并释放内存压力至磁盘；而大部分情况下，QianBase 执行器处理查询时，所有数据都完全在内存中，从而提高性能并降低对磁盘空间和 I/O 带宽的依赖。

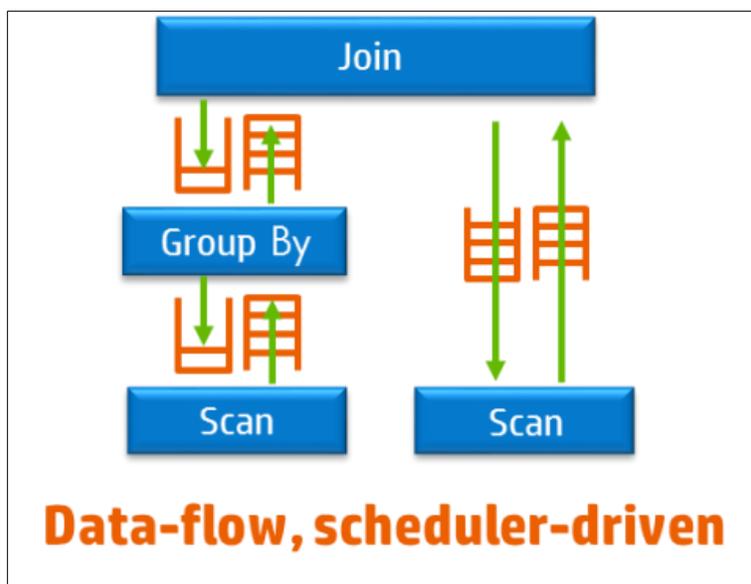


图 8-6 QianBase SQL 执行器的数据流和调度器驱动任务模式

执行器采用几种并行策略，例如：

- 分区并行

并行地处理多数据分区。在分区并行计划中，多个运算符为相同的计划工作。使用多队列或管道合并结果，再保存输入分区的 sort 顺序。由于数据被划分成多个独立执行的单元，所以分区也被称为“数据并行”。优化器根据数据如何在集群上分区、查询中每个步骤预估的行基数、集群中节点数量以及每个节点的核心

数量这四个因素决定并发度，进而决定每个 ESP 访问单个/多个 Region，或多个 ESP 访问相同 Region 的某些部分/salt 分区。

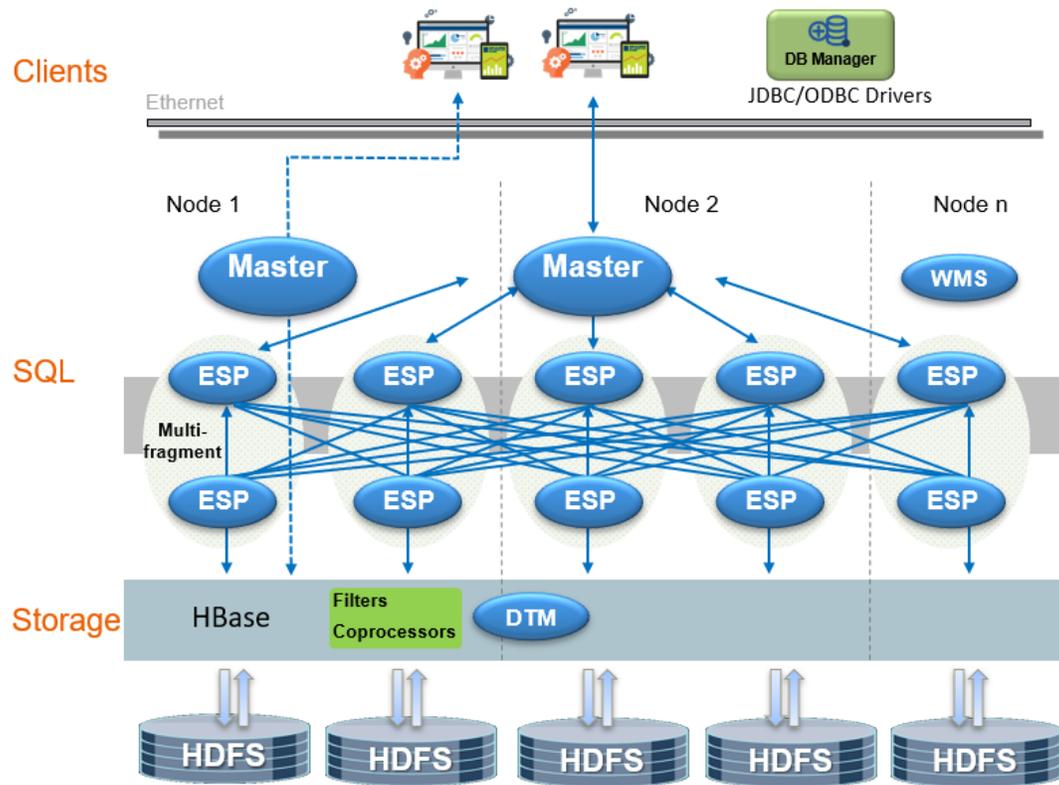


图 8-7 QianBase 分区并行

• 管道并行

管道并行是数据流架构的自带特性。数据流架构通过队列连接所有运算符，队列中前一个运算符的输出作为后一个运算符的输入，每个运算符独立工作，不依赖于其它运算符，并在输入可用时立刻生成输出。管道并行适用于大部分查询计划，唯一例外是 sort 操作。由于 sort 需要下游运算符处理完所有数据，所以它不适用于管道并行。sort 在该数据流中是唯一阻塞运算符。

• 运算符并行性

运算符并行性允许两个或多个运算符并行地工作。除了在某些同步 (synchronization) 条件下, 一般而言, 运算符独立运行。

QianBase 无需像使用 Hadoop MapReduce 或 Spark 一样通过特殊代码处理并发, 而是自动处理并发。优化器产生的单个查询计划能包含分区、管道或运算符并行这三种组合。处理行所需的基数和优化器经验规则共同决定查询计划任一阶段的并发度。

8.5 支持事务型/运营型

QianBase 为 OLTP 单行访问（事务）提供多种编译和运行时优化。

- 对于 OLTP 或运营型查询而言，Master 向 HBase 发出“直接”键访问请求，无需中间 ESP 进程。
- 对于查询（包含复杂 SQL 语句，例如，需要对数据进行 rebroadcasting 或 redistribution 的 n-way join 或 aggregation）而言，包含 ESP 的一层或多层的并行计划可以大幅缩短响应时间。

其它优化包括：

- 断开连接后，依然保留 Master 和 ESP 且可以重用，这能减少启动和关闭进程的开销。
- 一个节点上的多个 ESP 组合成单个多 segment 进程，用于减少进程数量。

根据基数和其它因素，QianBase 优化器为工作负载选择合适的连接策略。通常而言，QianBase 为运营型工作负载选择 nested join 或者 probe cache，为 BI/分析型查询选择 merge 连接或者 hash join。但正如在 [Large Scope Rules](#) 中的讨论，QianBase 对 BI 工作负载的星型连接也会选择 nested join 方法，但会非常谨慎，以避免生成糟糕的执行计划。如果预估基数和实际行数大相径庭，那么 nested join 和 merge join 或串行计划可能是毁灭性的。

QianBase 优化器对这些计划应用风险溢价规则，如果计划的成本低于潜在安全计划的成本（低于一定比例，该比例与风险溢价相关），则优化器会选择该计划。预估行数和实际行数大相径庭时，根据基数预估的准确度和导致查询低性能的可能性，优化器会适当应用风险溢价规则。

以下总结是 QianBase 的多种优化方式, QianBase 能优化运营型/事务型工作负载(次秒级响应时间)。

表 8-1 QianBase 工作负载优化

运营型/事务型

- 主键访问优化。
 - 下推技术。例如, 过滤器, 协处理器。
 - 通过串行和并行计划进行有效访问。
 - 使用原生 SQL 表达式提高编译器和执行速度。
 - 查询计划缓存减少不必要的再编译。
 - 支持二级索引的并行访问和维护。
 - 多种 ODBC/JDBC 驱动器支持不同配置和性能要求。
 - 透明使用 HBase API 优化。
 - Adaptive segmentation 为查询选择合适的并发度, 在利用最少资源的情况下能实现高并发性。
-

8.6 数据连接服务

分布式应用程序客户端连接至高可扩展集群数据库引擎时，数据连接服务（Data Connectivity Services）提供了非常重要的服务：

- **抽象层**：允许 QianBaseSQL 引擎透明地与任何使用 JDBC、ODBC 或 ADO.NET，甚至 REST API 的客户端协作。
- **负载均衡**：在集群中，客户端连接和与连接相关的工作负载的高效负载均衡。
- **扩展性**：在最小开销的情况下，利用纵向/横向扩展，增加并发客户端连接数和为这些连接服务的 QianBase 组件的数量。
- **高可用**：进程失败时，使用 persistent 和 backup 进程自动恢复客户端连接。
- **数据本地化**：（该功能可在将来实现）将请求传送至某些节点，这些节点存储了该请求所需的数据，这能减少节点间通信。

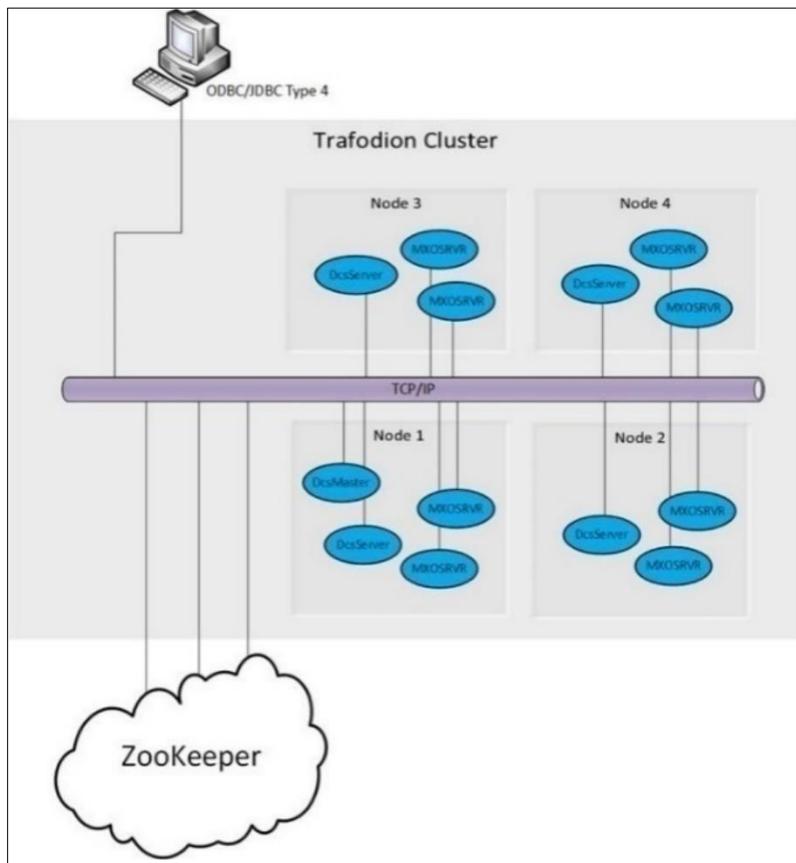


图 8-8 QianBase 数据连接服务

8.6.1 抽象层

QianBase 支持驱动程序的 Linux (JDBC 或 ODBC) 和 Windows (JDBC、ODBC 或 ADO.NET) 版本。客户端通过特定端口连接至 DCS Master。与 QianBase SQL 引擎连接时，DCS 组件将这些不同 API 映射到通用 CLI，为 SQL 引擎提供了一个通用抽象层。

8.6.2 负载均衡

DCS Master 管理多个 MXOSRVRs (即, 在之前架构图中的 Master)。为了不与 DCS Master 混淆, 本章使用 MXOSRVR 指代 MXOSRVR Master。MXOSRVR 编译和处理查询, 再发送结果至客户端。在多种因素的影响下, DCS Master 将客户端连接至

某一 MXOSRVR。其中一个因素是客户端与 MXOSRVR 的关联性，如果客户端过去已连接至某个可用的 MXOSRVR，那么这不仅能节约打开文件的时间（该文件与被访问的表、安全检查、缓存元数据和缓存计划等相关），还能节约编译和开始执行查询的时间。

另一个因素是平衡集群负载的需求。DCS Master 尽量平衡节点上客户端的连接。根据配置中 MXOSRVR 的运行数量、可用性和每个节点活跃连接的数量，DCS Master 为客户端选择合适的 MXOSRVR。

8.6.3 扩展性

DCS 架构中存在一个 DCS Master。在大部分情况下，由于 DCS Master 功能简单且运行高效，因此，单个 DCS Master 可以轻松处理大量并发客户端，不会遇到任何性能瓶颈。但是，QianBase 强大的横向扩展力允许您能跨集群配置多个 DCS Master。

针对每个 DCS Master，您可以配置集群中开始运行 MXOSRVR 的节点。根据应用程序所需的并发度，确定 MXOSRVR 的数量。这使节点能纵向扩展，为您的应用程序带来最佳并发度，并能最大化地利用节点上的资源，且性能不会低于您 SLA 的要求。

您能灵活地选择工作的节点，例如，您能为所有应用程序使用整个集群，或限制某些应用程序仅能在某些节点上运行查询。DCS Master 支持横向扩展和纵向扩展的组合，并能根据应用程序的需求均衡负载，平均分布应用程序将访问的数据。

这也是跨多租户管理 SLO（Service Level Objective）的关键配置因素，这些租户可以是单个或多个具备不同 SLA 要求的应用程序：

- 独立用户，拥有各自的 schema。
- 分享相同应用程序 schema/表的 用户或企业部门。

- 共享或独立 schema 的组合。

8.6.4 高可用

DCS 支持高可用。DCS Master 在每个节点上开启 DCS 服务器守护进程，该守护进程是一个保持所有 MXOSRVR 运行的持续进程。所有守护进程在 Zookeeper 中注册，因此，失败后启动守护进程时，它将知晓其它与自己相关的进程。

DCS Master 采用主备进程的方式实现故障转移，如果主 DCS Master 节点失效，它会立刻连接到备 DCS Master（接管所有工作），从而保证客户端不会失效。另外，QianBase 还使用 keepalived 浮动 IP 方案实现故障转移，即客户端可以配置多个 IP 地址，如果连接某个 IP 地址失败，可以连接其它 IP 地址。

如果 MXOSRVR 失败，则 DCS Server 将重启 MXOSRVR。如果 MXOSRVR 正在处理客户端发出的查询，则客户端会收到连接和查询的失败消息。如果已发起一个事务，则该事务将自动被 DTM (Distributed Transaction Manager) 终止，该 DTM 在每个节点上有一个 TM 进程，管理节点上发起的事务。如果 DCS 服务器失败，则 DCS Master 重启 DCS Server。

DCS 架构是企业级高可用需求的特殊设计，它能最小化对客户端的干扰，提高数据库的可用性。

8.6.5 数据本地化

虽然目前还未实现该功能，但可根据用户需求在今后实现。根据处理客户端请求的所需数据所在的节点，该功能将客户端请求传递至这些节点。根据跨 partition/region 的数据分布信息、salting 和计算 salt 值的方法，客户端驱动能推算出请求所需访问的节点，并直接发送请求至该节点，而不需要 DCS 的分配，从而减少为该请求服务

的节点间通信、降低不必要的开销、避免集群资源利用的不平衡，并有利于线性横向扩展。

8.7 功能概览

QianBase 的主要功能包括：

- QianBase 是企业级 SQL DBMS，提供关系型数据库的所有功能。
- 全面支持 ANSI SQL 语言，包括数据定义、数据操作、事务控制和数据库工具；还具备其它功能，例如，根据时间聚集数据、引用完整性、存储过程和用户自定义函数等。
- 支持标准 JDBC/ODBC/ADO.NET 驱动程序。
- 与 Hibernate 集成，为需要在应用程序上使用对象模型的用户提供 ORM 应用程序发展支持。ORM 模型不是基于数据库结构，而是基于真实业务概念，能简化应用程序的开发，并提高可移植性。

8.7.1 全面支持 ANSI SQL

QianBase 提供全面的 ANSI SQL 语言支持，包括全功能 DDL (Data Definition Language)、DML (Data Manipulation Language)、TCL (Transaction Control Language) 和数据库工具支持。

- QianBase 为创建和管理传统关系型数据库对象（例如，表、视图、二级索引、CHECK 约束、UNIQUE 约束和参照完整性约束）提供支持，包括自引用外键、支持层级关系的参照完整性（例如，确保员工的经理也作为“员工”身份，存储在数据库中）。
- QianBase 支持多种数据类型，包括数值 (NUMERIC)、整数 (INTEGER)、布尔类型 (BOOLEAN)、固定长度字符 (CHARACTER)、可变长度字符 (VARCHAR)、日期 (DATE)、时间 (TIME) 和时间间隔 (INTERVAL) 等。QianBase 支持多

种 Unicode 编码，包括 UTF-8、UCS2 和 ISO 8859-1，并使用隐式转换和翻译透明地处理不同数据编码之间的比较和数据操作。

- 同时，QianBase 支持大对象 (Large Object) 数据类型，例如，BLOB (二进制大型对象, Binary Large Object) 或 CLOB (字符型大对象, Character Large Object)，用户能直接在 HDFS 中管理 BLOB 和 CLOB，HDFS 文件路径与存储大对象的 Trafodion 列相关联。
- QianBase 全面支持标准 SQL 操作 (包括 SELECT、INSERT、UPDATE、DELETE 和 UPSERT/MERGE) 和语法选项 (包括 JOIN VARIANTS、UNIONS、WHERE 谓词、聚合[GROUP BY 和 HAVING]、SORT ORDERING、取样、相关和嵌套子查询、游标和多种 SQL 函数)。
- QianBase 提供其它实用功能，例如，更新表的统计信息，优化器使用这些统计信息计算各种替代方案的成本 (选择率/基数预估)；显示 SQL 执行计划；提供与数据库交互的命令行工具。
- QianBase 的显式控制语句允许应用程序定义事务边界，并能在确保安全的情况下中断事务。
- QianBase 支持 ANSI 定义的 GRANT/REVOKE 语法，在管理和访问数据库对象时，支持列级别的用户和角色权限。

8.7.2 存储过程

QianBase 支持 SPJ (Stored Procedures in Java)，您能使用 Java 编写运营或业务程序。客户端应用程序调用 SPJ 时，您能在客户端执行这些程序。如果一个存储过程有多条 SQL 语句，每条语句都需要在客户端和服务端之间交换数据时，此时使用存

储过程将 SQL 语句执行过程下推至服务端会更高效。Oracle PL/SQL 和 ANSI SQL 存储过程可以转换为 SPJ。您也能使用 QianBase 提供的工具，无需任何改动即能在 QianBase 中运行 PL/SQL 存储过程。

8.7.3 SQL 函数

QianBase 支持超过 150 种内置函数，例如，聚合、字符、日期时间、数学和 OLAP 函数，这些包括全部 ANSI 函数和其它数据库最常使用的函数，例如，Oracle 的 DECODE 函数，CONNECT_BY 函数和 TO_CHAR 函数。

为了处理 BI 和分析型工作负载，QianBase 支持更多 OLAP 函数（例如，LEAD 函数和 LAG 函数）。在同一个 OLAP 查询中，QianBase 支持多个窗口。此外，还支持其它常用分析功能（例如，ROLLUP 函数和 GROUPING 函数），并增加了 INTERSECT 和 EXCEPT 操作。

WITH 子句能简化复杂查询语句。在查询语句中，您能使用 WITH 子句定义重复多次的子表达式。

8.7.4 触发器

当特定的表或者视图发生特定的事件时，触发器会自动执行对应的存储过程。触发器主要用于维护数据库中信息的完整性。QianBase 在行或 SQL 语句上支持 BEFORE 和 AFTER 触发器。

8.7.5 用户自定义函数

QianBase 支持 C++ 或 Java 编写的用户自定义函数（User Defined Functions）。UDF 能部署数据库不支持的简单函数、复杂事件处理逻辑或机器学习算法库，例如，Continuum 开发的 Anaconda。

8.7.6 Table UDF

QianBase 支持 Table UDF，允许用户采用 MapReduce 模型进行开发。Hive UDF 和 Teradata Aster SQL MapReduce 的实现方式相似。与仅返回标量值的常规 UDF 函数不同，这些非常强大的 UDF 不仅能读取表或并行数据流，返回数组，还能在首次调用时动态地返回一组列。在一个查询中，这些 UDF 的返回结果类似于表或子查询。

QianBase 为开发人员提供了相关接口，因此，开发人员可以为优化器提供 UDF 的统计信息（例如，返回的行基数）。调用这些 UDF 时，优化器能生成更多高效计划。

这些函数能与在 Hadoop 系统中的其它应用程序集成：

- 处理来自 Apache Kafka 的流式数据。
- 在 Apache Spark DataFrames 中创建或访问内存数据。
- 对存储在 QianBase 表中的大型文本数据，使用 Apache Lucene 执行复杂文本搜索。
- 连接至任何 JDBC 数据源。
- 根据网络搜索，从文本中集成数据。
- 在 MongoDB 中创建、删除、搜索和读取 JSON 文件。

8.7.7 加载工具

QianBase 利用 HBase 的批量加载能力，直接将 HDFS 文本文件高速加载至 QianBase 表。该工具能处理带索引的表，并提供错误处理能力。另一个高速并行加载工具 ODB，能并行地将数据直接从其它 RDBMS 加载至 QianBase。QianBase 能使用大部分通过 JDBC 和 ODBC 连接至数据库的 ETL 工具。

8.8 管理性

QianBase Manager，以下是 QianBase Manager 的部分功能。

8.8.1 工作负载可见性和控制

- 查看活跃查询和查询统计信息，了解影响工作负载性能的查询。
- 根据 CPU 用时、内存使用、磁盘 IO 和运行时间显示排名前五的查询。
- 如果查询可能影响在系统中运行的其它工作负载，您能取消该查询。
- 通过历史查询统计信息和查询计划，您能了解性能问题、事后分析问题并采取纠正措施；或对硬件资源进行容量规划，以满足未来不断增长的 SLA 的要求。
- 查询工作台执行即席查询、分析查询计划，并在将来优化这些查询和计划；支持从 JSON 文本文件中导出和导入查询结果和解释计划，或导出和导入查询结果和解释计划至 JSON 文本文件。
- SQL 转换工具，将其它数据库的 SQL DDL 和 DML 语句迁移至 QianBase。
- 执行多语句 SQL 脚本的脚本执行工具。

8.8.2 系统检测和健康检查

- 通过和数据库集成或独立运行方式监控单个或多个 QianBase 实例。
- 仪表盘显示核心子系统的状态（例如，运行中或非运行中的配置进程，启动或关闭的节点），从而在出现故障时能立刻采取措施。
- 根据时间进行事务计数（终止/提交/开始），了解事物性能和事务计数变化，或减少事务终止。

- 系统运行时，关键系统指标分析系统性能。例如，IO 等待、HBase Region Server 内存使用、磁盘空间使用、垃圾回收时间等指标的时间序列图。您还能通过关键系统指标的历史记录，检测问题并采取措施，或了解资源使用情况以进行容量规划，特别是部署的新应用程序。
- 当序列值超过阈值时，系统将发出邮件提醒或 HTTP 警报，相关人员能立刻处理问题。
- Canary 查询响应时间(显示为时间序列值)：每隔五分钟运行一次 Canary 查询，计算扫描/写入的次数，评估系统性能。如果 Canary 查询的响应时间超过阈值(与正常响应时间相差甚远)，则代表系统性能下降，此时系统将向用户发出警报，用户需立刻采取措施。
- 查看所有可用连接服务器和已连接的会话，监控目前已连接至 QianBase 的应用程序和用户，评估系统健康，了解系统对用户和应用程序的可用性。
- 查看所有 QianBase 组件的事件/日志以进行事后分析，了解问题区域或监控违反安全规定的操作。

8.8.3 工作负载管理系统

QianBase 已拥有服务等级基础架构，用户能：

- 根据 ODBC/JDBC 连接信息，设置工作负载配置项。
- 将配置项映射到服务等级。
- 提供会话级设置或提示，影响与服务等级相关的工作负载的行为。
- 为特定工作负载分配特定节点，这些特定工作负载与服务等级相关。

8.8.4 多租户

QianBase 的多租户功能将 Linux cgroup 集成至工作负载管理架构，用于管理跨租户的工作负载。QianBase 与 HBase 命名空间集成，使租户能隔离数据库对象及其安全和管理。

QianBase 的多租户有两层含义：QianBase 本身作为一个系统的租户，和其它系统（例如，Spark 和 Hive 等）分享不同比例的系统资源，QianBase 只使用被分配的资源；在 QianBase 内部，不同租户根据配置，使用不同比例的系统资源。

8.8.4.1 分配给 QianBase 租户的资源

理想状况下，QianBase 不与其它系统分享资源，它独享所有资源，但这种情况不太现实。为了向 QianBase 提供资源，用户使用 cgroup 分配的 CPU、内存、I/O 和实例 Cloudera Manager 静态服务池。QianBase 并不是独立工作，它与 Hadoop 生态圈组件协同工作，例如，QianBase 使用 HBase 作为存储引擎（即使用了 HDFS）；它使用 Zookeeper 协调资源；在内存不够的情况下执行复杂查询时，它溢出到磁盘并使用 Linux 文件系统。根据不同的工作负载，QianBase 使用不同的 Hadoop 组件和资源。但由于不可预测性和流动性，QianBase 无法获取充分的资源和适当的组件。

8.8.4.2 在 QianBase 中向租户分配资源

用户可以跨租户分享一个 QianBase 实例。如果租户运行完全独立的工作负载且不分享数据，那么建议每个租户使用单独的 QianBase 实例，但这种情况不适合所有用户。无论租户是否分享数据，QianBase 允许用户向每位租户分配不同的 CPU 和内存。QianBase 建议向每位租户分配至少 16 核和 128G 内存。租户有父租户和子租户两个

层级，父租户管理子租户。QianBaseManager 配置和管理租户，多租户集成至 QianBase 工作负载管理系统。

8.8.4.3 租户安全

QianBase 支持角色和用户（授权 ID）功能，以及 Grant 和 Revoke 操作（schema、对象和列级别），但跨租户管理授权 ID 和 schema 可能会非常复杂。另外，每个租户可能希望管理自身的授权 ID 和数据库权限。因此，创建租户后，您能为该租户指定管理员角色和默认 schema。租户管理员可以创建角色、用户和 schema，并为租户的特定用户或角色设置安全级别。

QianBase 使用 HBase 的 namespace 特性来隔离不同租户的对象，所有对象被创建在租户自身的 namespace 中。根据租赁期，您能通过合适的动态目录/LDAP 资源对租户进行验证，租户能设置不同的目录服务器用于验证用户。

QianBase 通过租户将 AD/LDAP 的用户组与 QianBase 的角色相互映射。用户登录时，QianBase 进行身份验证，从 AD/LDAP 中获取当前用户所属的用户组列表，并根据当前用户连接时指定的租户，确认用户所属的用户组是否为当前租户的合法用户组，再根据符合的合法用户组所映射的角色，确定当前用户的权限。

您能为不同租户设置不同并发度，租户能扩展或收缩。

QianBase 支持多层次的多租户功能，能为不同租户分配不同资源。同时，QianBase 还能通过隔离不同租户的对象，提高管理和安全水平。

8.8.5 其它功能

- 数据库浏览，包括获取表的 Region 统计信息。
- 与 LDAP 协作，负责所有 SQL 管理功能的认证，确保不同用户能使用不同管理功能，支持多域活动目录或 LDAP。

- 与 Hadoop 的安全认证机制 Kerberos 集成，解决[安全认证](#)问题。
- 用户自定义函数和存储过程 jar 文件管理。
- 管理架构的持久性/高可用，因此，当系统压力过大或出现系统故障时，管理架构仍然能分析并解决问题。
- REST 服务器提供公开 API，这些 API 能完成任何所需的自动化操作，因此，用户能将 QianBase 管理环境集成至自身管理环境中，或自动化管理任务，提高灵活性。
- 与 Replicant 集成，该工具能将 Oracle 和 MySQL 的数据捕获（一次性、递增或实时）至 QianBase。

8.9 与HBase深度集成

对于运营型工作负载而言，QianBase 将数据库对象存储至 HBase/HDFS 存储结构，与 Vanilla HBase 相比，QianBase 拥有更多附加价值，例如：

- QianBase 在 HBase 之上提供了关系型 Schema 抽象，允许用户使用知名的、并经过充分测试的关系型设计方法论和 SQL 编程技能。
- 从数据的物理存储角度看，QianBase 使用标准 HBase 存储机制（列族存储使用键值对）存储和访问对象。它使用 HBase 多列族支持，将更新或访问频率高和频率低的列分别存储至不同列族，或将访问频率低的 Large Column 存储至单独的列族。QianBase 对列名进行编码，从而节约磁盘空间，减少消息通信开销，提高 SQL 性能。
- 增加数据或更新数据时，QianBase 的列必须定义成某种数据类型。这不仅能大幅提高数据的质量和完整性，还能减少为解析和翻译数据而开发应用程序逻辑的需求；而 Vanilla HBase 存储的数据是无数据类型的字节数组。
- QianBase 提供 Aligned Format，该行存储格式将逻辑关系元组所有列值存储在一个 HBase 列值中，从而将一个逻辑行映射至一个 HBase 行。对于更新较少、查询频繁且需要访问多列的工作负载而言，该功能可以大幅提高性能。
- QianBase 扩展 ACID 保护至应用程序中定义的事务，该事务能涵盖多条 SQL 语句、多个表和行。通过保护数据库完整地执行事务（例如，确保数据库中事务完整地执行或完全不执行），QianBase 能大幅提高数据库的完整性；而 Vanilla HBase 仅提供行级 ACID 事务保护。

- QianBase 允许用户定义表能被复制（跨指定集群/数据中心）。跨数据中心的分布式事务管理确保数据并行地同步，并作为事务的一部分写入到其它数据中心。发生灾难时，在最小影响性能的前提下，确保对已提交的数据实现零丢失；而 HBase 只能提供行级保证，对已提交数据的预写式日志执行异步复制。
- QianBase 的 API 符合 ANSI SQL 标准，该 API 是业界熟悉且被广泛使用的编程接口，因此企业能使用现有 SQL 知识、技能和工具；而 HBase 原生 API 是底层 API，未被企业开发人员广泛使用。
- QianBase 允许主键成为由多列组成的组合键；而 HBase 的键结构由单个无数据类型的字节组组成。
- QianBase 支持创建二级索引，根据非主键列值访问行数据时，它能提高处理事务的性能。ACID 事务支持保证索引和基表的完整性；而 Vanilla HBase 不具备该功能。
- SQL 下推使用标准 HBase 服务，例如，Filter（Start-stop Key 谓词和 Non-key 谓词）和 Coprocessor（Count Aggregate），Coprocessor 也处理 Update Statistics 和使用样本的快速行数预估。
- QianBase 使用 HBase Bulk Load 功能，能快速地加载文本文件至 QianBase 表。
- 通过 QianBase，您能使用 HBase 快照功能执行更快速的扫描。QianBase 也使用该功能进行备份和恢复。
- QianBase 使用 HBase 命名空间为用户隔离数据库对象（例如，Schema 和表），这能避免与宿主在 QianBase 上的租户同名。另外，QianBase 也可以通过 HBase 的 Namespace 为多租户指定配额（例如，每个 Namespace 的表或 Region 的数量）。

- 您能通过 QianBase 设置多种 HBase 属性，例如，文件和块压缩、复制、使用版本控制/时间戳获取先前的行并以时间点和标签等形式提供数据库快照。

8.9.1 行键 salting

众所周知，HBase 在支持事务工作负载时存在一个缺陷：数据以行键顺序插入进表，此时，所有 I/O 集中在单个 HBase Region，这会引起服务器和磁盘热点，并导致性能瓶颈。为了解决这个问题，QianBase 通过行键列计算 salting。

为了启用该功能，创建表时，DBA 指定表的分区数量（例如，SALT USING 4 PARTITIONS）。QianBase 创建表预分区，用户能决定 salt 值映射到 HBase Region 的数量，允许 Region 分区的数量随表的数量增长。QianBase 为行键添加前缀列“_SALT_”，它是一个内部 hash 值列。Salting 自动被 QianBase 处理，它对应用程序编写的 SQL 语句是透明的。随着数据插入进表，QianBase 计算出 salt 值并将它们插入至合适的 Region。同样，从表中获取数据时，QianBase 计算 salt 值并尽可能自动生成谓词。Multi-dimensional Access Method (MDAM) 技术大幅提高了处理效率（稍后讨论），它是轻量级操作，几乎没有开销，也不会影响键访问操作。

Salting 能使您跨 Region 获取更多数据分区，并通过减少热点提高性能。

QianBase 允许多个 salt 键映射至单个 Region，随着集群数量增加，HBase 的 Region 可以在 salt 值边界进行分裂，有助于再平衡整个集群的节点间 I/O。您可以映射大量值到 Region，以满足集群多种未来扩展的需求。

另外，您能使用 SPLIT BY 的分区列显式地指定分区范围，并将分区中所有值分布至 HBase Region 中，这等价于范围分区，而 SALT 支持 hash 分区。

8.9.2 Divisioning

QianBase 也支持 divisioning 技术，它类似于分离冷热数据 [Teradata's Partitioned Primary Index \(PPI\)](#)。divisioning 允许基于一个表达式（例如，从日期或时间戳列值里获取的月份）创建列，因此，与特定时间段相关的数据，例如，某个月的数据，可以在一个 Region 内或 salt 分区内聚集。该列排列在 salt 键列后，这不仅有利于数据基于通常被查询访问的某一时间段进行聚集，提供更快速的访问，还有利于轻松删除老化数据。MDAM 保证高效访问（稍后讨论）位于 salt 和 divisioning 列之后的列。

总之，与 Vanilla HBase 相比，QianBase 增加了多种改进。例如，提高了事务性能、数据完整性和 DBA/开发人员产出，同时减少了应用程序复杂性，使用标准和常用的关系型数据库的经验技能和 API。

8.10 多维访问方法 (Multidimensional Access Method)

MDAM 是一项专利技术，它使用“多维”谓词提高行检索性能。例如，假设您有一张表，行键是 Week 列、Item 列和 Store 列的组合值，但应用程序仅提供 Item 列和 Store 列谓词。如果没有 MDAM，DBMS 必须执行全表扫描，或在 Item 列和 Store 列上创建二级组合索引。

然而，MDAM 使用原生 HBase 行键排序功能，扫描数据并跳过不符合条件的行，仅读取处理 SQL 语句所需的最少量的一组行。MDAM 也可用于处理数据检索请求（例如，在组合索引列上的 IN 列表、NOT equal[<>]谓词和多值谓词等），降低响应时间，并减少额外二级索引的需求。另外，它还能高效访问具备 salted 列和 divisioning 列的表。

Where is item = 1, Stores 2 through 5?

Week	Item	Store	...
1/7/90	1	1	...
1/7/90	1	3	...
1/7/90	1	5	...
1/7/90	2	34	...
1/7/90	3	13	...
1/7/90	3	3	...
1/7/90	4	2	...
1/7/90	4	4	...
1/14/90	1	2	...
1/14/90	1	4	...
1/14/90	1	5	...
1/14/90	1	35	...
1/14/90	3	1	...
1/14/90	3	20	...
1/14/90	4	11	...
1/14/90	4	12	...

图 8-9 多维访问方法

8.11 分布式事务管理 (Distributed Transaction Management)

Vanilla HBase 仅提供单表、单行级 ACID 保护，而 QianBase 的 DTM 将事务保护扩展至包含多条 SQL 语句、多张表或单张表的多行。另外，QianBase DTM 使用 2 阶段提交协议处理多个 Region 事务，为分布式集群配置提供保护。事务保护在 QianBase 的组件和进程间自动传播。

DTM 为隐式 (auto-commit) 和显式 (BEGIN, COMMIT, ROLLBACK WORK) 事务控制提供支持。使用 HBase 多版本并发控制 (Multi-Version Concurrency Control) 算法，QianBase 允许多个事务并发地访问相同行 (不加锁)。首个事务更新成功后，其它更改相同行的事务将在提交时收到操作失败的消息 (更新冲突)。

与 HBase 单行原子操作相比，QianBase 使用两阶段提交协议机制，这会增加通信开销。对于多语句操作而言，通信开销不会对更新、删除、插入和扫描操作造成太大影响。但对于单行操作而言，通信开销占整个查询执行流程的较大部分，并对性能有较大影响。为了避免发生这种情况，QianBase 优化单行操作，确保它们与 HBase 单行操作的性能相同。在 region 中，单行操作直接在 SQL 引擎和 HBase 协处理器中处理，而无需使用 DTM。单行操作在 region 内执行冲突检查，即使发生冲突的可能性极小 (运行时间很短)，但它仍然支持冲突解决。

QianBase 的分布式事务管理架构具有可扩展性，每个节点有一个事务管理器 (Transaction Manager)，负责协调分配该节点发起的事务。DCS 负责平衡节点之间的事务工作负载。HBase coprocessor 管理事务的上下文 (Region 级)。每个 Region 管理自身的事务更新和数据冲突解决。所有事务更新写入 HBase 预写式日志 (Write Ahead Log) 文件，因此，您能使用 HBase 备份和恢复功能 (以并行方式高效运行)。

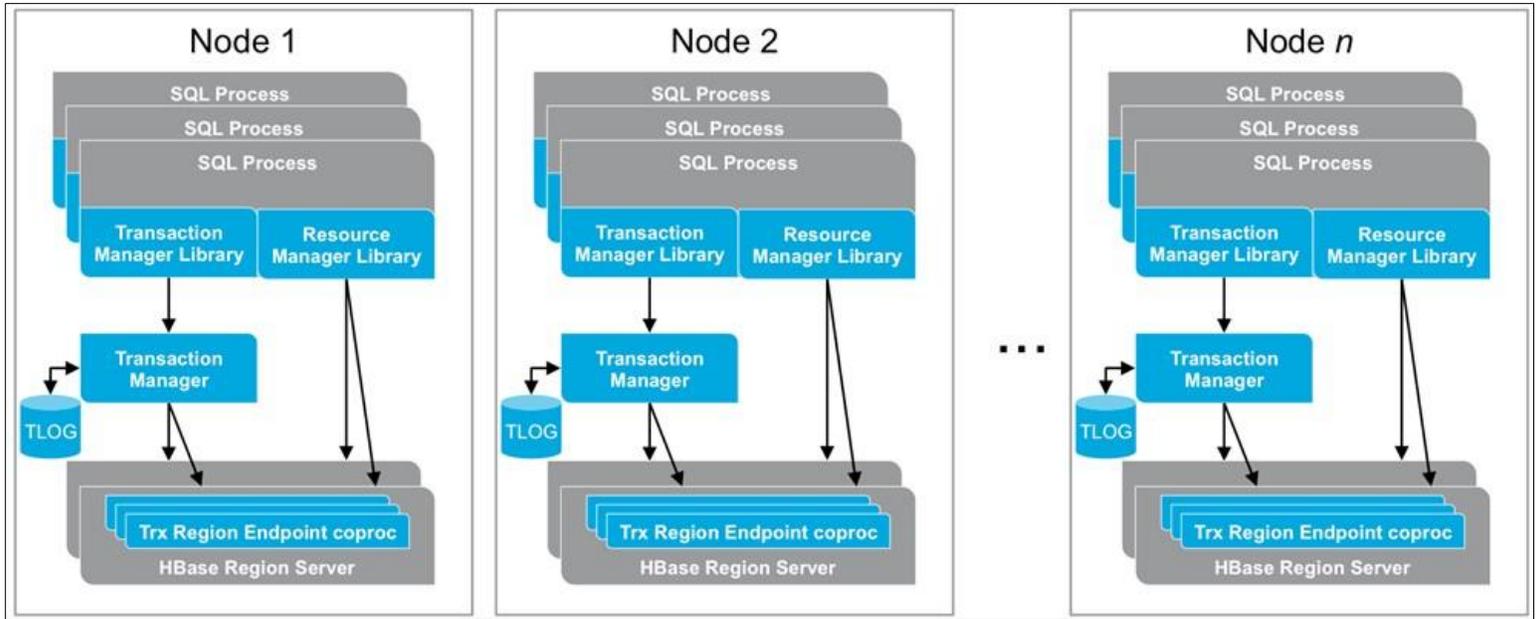


图 8-70 分布式管理架构

与该分布式事务管理架构相比，HBase 的其它事务管理实现具有以下劣势：

- 集群大小增加时，事务管理器不能扩展，这将对托管在该事务管理器上的节点造成性能瓶颈和巨大消息流量。
- 必须在事务管理器内解决事务冲突。因此，事务管理器管理所有事务的上下文和输入。由于扩展可能引起事务管理器节点巨大消息流量，所以不能扩展。由于节点没有足够内存来维护所有上下文和处理事务的巨大消息流量，因此，无法处理大型事务。
- 事务管理器使用 HBase 的版本控制/时间戳列管理事务，这将导致某些快照和基于时间的查询无法使用该版本控制/时间戳列。

8.11.1 行锁

在 OLTP 环境中，访问数据库的并发用户数可能非常高，同时事务响应需要在几十到几百毫秒的范围内。为了获得更好的性能，事务管理系统支持行级锁的能力非常重要。与任何传统的 RDBMS 一样，QianBase 也支持此功能。

事务可以通过锁定行或者表来阻止其他事务对行的修改，以确保 ACID 属性（防止读取未提交的数据，防止多个用户尝试同时更改相同的数据）。当事务不再需要对行或表数据操作时，事务将解除锁。

行锁由 Lock Manager 通过 HBase 处理器统一协调管理，Lock Manager 托管在每个 HBase Region Server 中。Lock Manager 支持表和行锁以及其他多种锁模式- S, U, X, IS, IX, NO。

表格 8-2 行锁多种模式对照表

冲突对照						兼容对照					
	X	U	IX	S	IS		X	U	IX	S	IS
IS	1	0	0	0	0	IS	0	0	0	0	1
S	1	0	1	0	0	S	0	0	0	1	1
IX	1	1	0	1	0	IX	0	0	1	0	1
U	1	1	1	0	0	U	0	1	0	1	1
X	1	1	1	1	1	X	1	1	1	1	1

8.12 高可用和数据完整性功能

QianBase 使用 HBase 和 HDFS 的内在可用和数据完整性功能。

Hadoop/HDFS/HBase	HA 优势
Name Node 冗余	保护系统，防止 Name Node 失效。
HBase 复制 (异步)	在 HBase 部署之间复制数据，以提供灾难恢复解决方案。
HDFS 复制 (数据块复制)	提供数据保护，防止节点失效、磁盘故障或数据损坏。
HBase 快照	捕获某一时间点表的快照，能将表恢复至该时间点。
Zookeeper	为托管在 Hadoop 上的服务提供高可靠的协调管理。

另外, QianBase 使用不同 Hadoop 发行版, 这些发行版提供企业级可用性扩展特性, 在 HBase 和 HDFS 的基础上, QianBase 提供高可用功能, 包括:

- 持续连接服务, 在数据连接服务失败时, 确保客户端能再次建立连接。
- 自动查询再提交 (Automatic Query Resubmission), 在某些情况下, 重新提交失败的 SQL 语句。
- 由于数据跨 Region 被分裂和再分配, 为了提高性能和扩展性, 在不影响事务管理器的情况下, HBase Region Server 能对数据进行分裂和再分配。因此, 您的应用程序能一直在线, 不被中断。
- 大幅减少备份窗口 (从小时减少到分钟), 支持全量在线备份。

8.12.1 跨数据中心双活零丢失事务

QianBase 能跨多个集群（或多个数据中心）同步复制完整的 SQL ACID 事务更新，这能确保在发生灾难时不丢失任何事务更新。QianBase 支持主-主模式（Active - Active），即能跨数据中心在多个集群上读取和写入数据。这有利于跨集群和数据中心扩展工作负载，提供数据本地访问，提高安全性能。

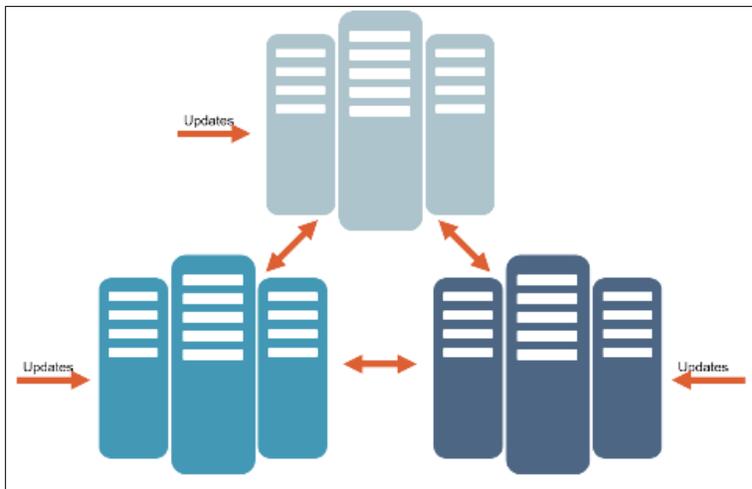


图 8-81 QianBase 跨数据中心双活零丢失事务

8.12.2 增量备份和恢复

QianBase 支持增量备份和恢复单个数据库对象。

8.12.3 弹性扩展

大部分 Big Data 部署在公有云或私有云中，即使数据库未在云中部署，随着服务增加，应用程序也有按需扩展的需求。这不仅对前景明朗、飞速发展的小企业而言十分重要，对大型企业而言，业务急剧增长导致宕机的代价也是巨大的。许多大型业务呈季节性增长，例如，节假日的销售量。在节假日时，计算和存储资源的需求增加。节假日结束后，计算和存储资源的需求恢复到日常水平。通常而言，扩展集群

前必须关闭数据库，再重新配置集群，重新分配数据，最后恢复数据库。对于专有数据库而言（例如，Oracle），扩展意味着更大型、更昂贵的硬件集群，或进行 RAC 扩展（也需要大量宕机时间和导致高昂成本），且每次扩展的增量较大（无法实现小幅扩展），因此不具备真正的弹性。

QianBase 提供完全的弹性扩展功能。在新增节点上安装软件的同时，可以在线扩展集群，且其余集群仍能运行。一旦集群配置更新节点信息，QianBase 将立刻为查询和事务使用计算节点。QianBase 能立刻增加节点，无需关闭数据库，没有宕机。HBase 配置添加 Region Server 后，HBase 将自动再分配数据。无论计算服务器与存储服务器增加数量相同与否，由于数据能跨新的存储服务器平均分配，通过将多个 salts 映射至一个 Region（已在 salting 章节中讨论），都可以在 salt 边界实现。

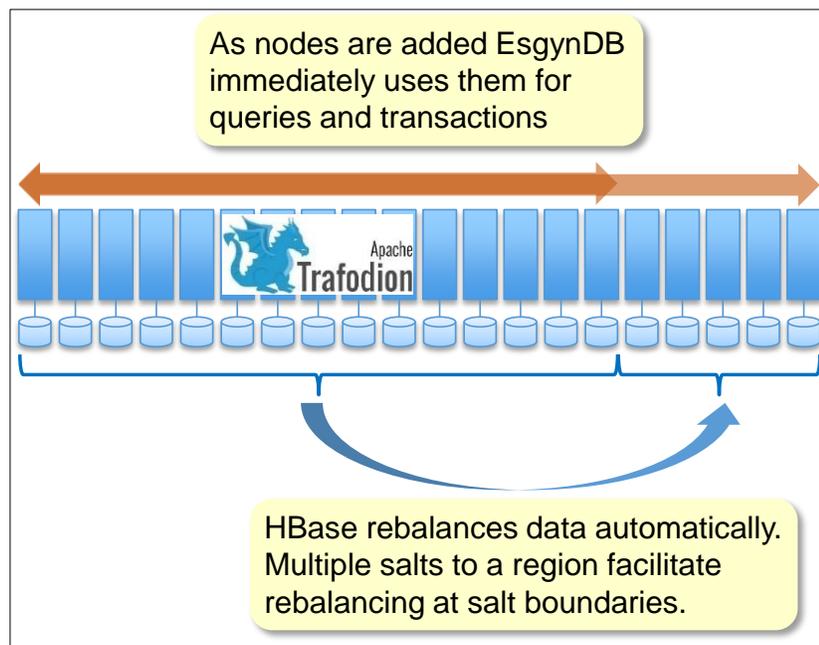


图 8-12 QianBase 弹性扩展

8.13 安全性

QianBase 提供企业级安全功能。通过标准集成接口，QianBase 集成了 Kerberos。

QianBase 支持的安全功能包括：

- 可靠的安全验证。
- 符合 ANSISQL 标准的权限功能，支持 schema 级、组件级、对象级和列级权限。
- 支持角色级和用户级权限。
- 集成的用户管理和对象所有权。
- Kerberos 与 Hadoop 集成。
- Repository 跟踪已执行的 SQL 语句和用户修改权限的行为。
- 日志记录任何违反安全规定的操作。
- 日志记录审核

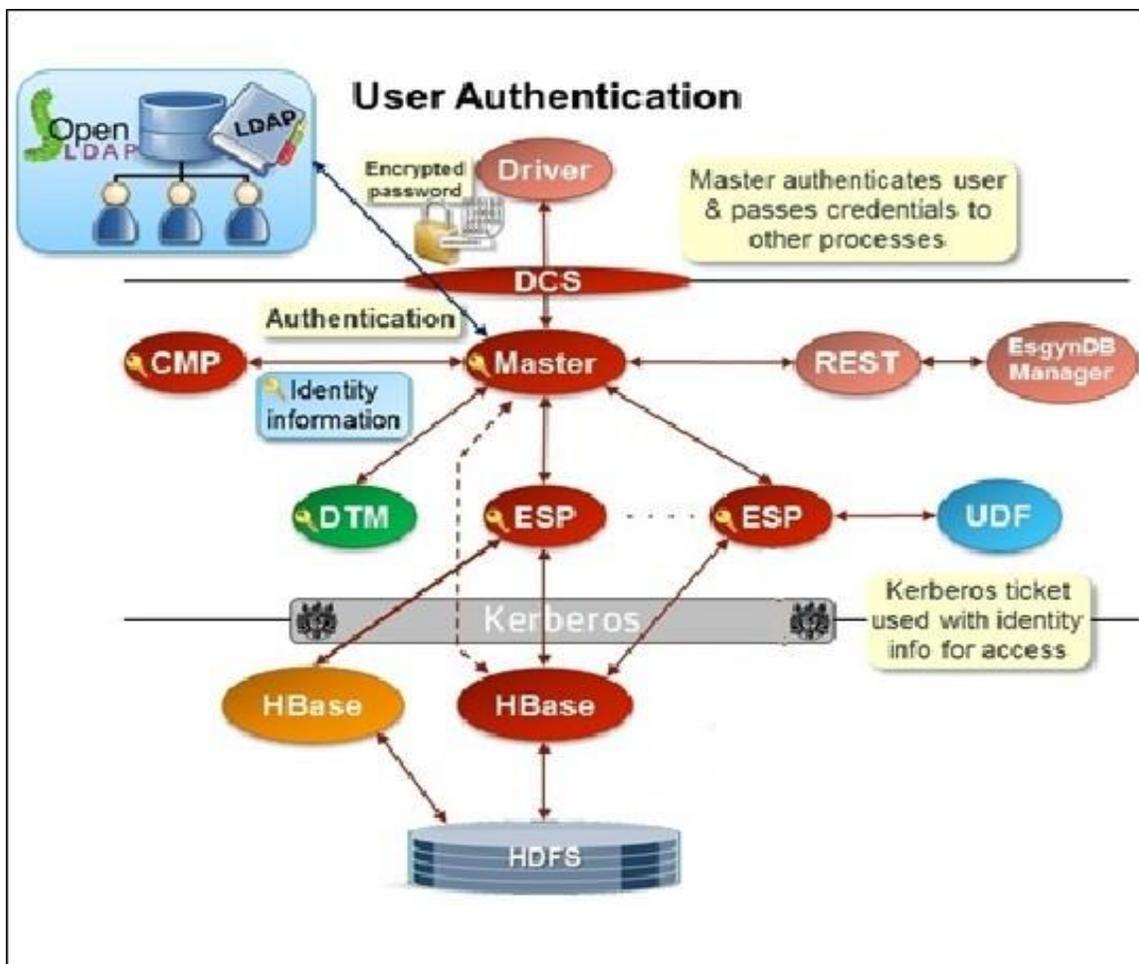


图 8-93 QianBase 安全性

8.13.1 授权

授权功能是验证连接 QianBase 用户身份的过程。授权框架验证用户具备访问系统的权限。QianBase 将用户名和加密密码与指定的身份存储库中的信息相比较。QianBase 支持多域活动目录或 LDAP 身份存储。ODBC 驱动程序将用户 ID 和加密密码保存在 DSN 中。

8.13.2 验证

验证功能确保用户具备所请求的 SQL 操作的权限，该过程由 QianBase 权限管理器管理。

权限管理器提供安全服务，包括向用户获取和显示权限、授予和撤回权限。为了获取权限信息，权限管理器使用 QianBase 存储权限相关的元数据。

验证用户并将认证信息存储至 QianBase 后，用户能执行查询。编译查询时，您能从元数据库中检索当前用户和当前用户角色，包括特殊 PUBLIC 角色。您能检索系统（组件）级、对象级和列级权限，并将它们组合成一个简单列表。权限管理器将执行查询必要的权限与用户已有权限和请求成功或失败的信息进行对比。

在对象的生命周期内，数据库安全管理员授予和撤回权限。撤回权限后，权限管理器向所有其它 QianBase 进程发送消息，通知它们已变更权限。如果执行受变更权限影响的查询，QianBase 将重新编译并使新的更改生效。

权限功能使您能针对特定对象或组件执行特定操作。您能通过以下方式授予或撤回用户或角色的权限：

- 创建对象或组件时，隐式权限授予给对象或组件的所有者。在对象的整个生命周期，所有者拥有隐式权限。
- 数据库用户管理员、对象所有者或在被创建时使用 WITH GRANT OPTION 语法的用户能授予或撤回用户或角色的显式权限。
- 您能通过多种方式向用户授予权限。权限能直接被授予，也能通过角色继承。例如，用户能通过两个不同的角色获得同一张表的 SELECT 权限，如果用户撤回了其中一个角色的权限，用户仍能通过另一角色获得该表的 SELECT 权限。

- 被授予某一角色的用户也拥有该角色所有的权限。撤回权限的方法是撤回用户的角色，或撤回角色的权限。

您能授予和撤回以下权限：

- 组件的系统级权限，例如，SQL OPERATIONS，包括：
 - 允许用户 CREATE、ALTER、DROP、MANAGE 和 SHOW 对象。
 - MANAGE 权限允许用户管理数据库 STATISTICS、COMPONENTS、ROLES 和 USERS 等。
- 对象和对象列。

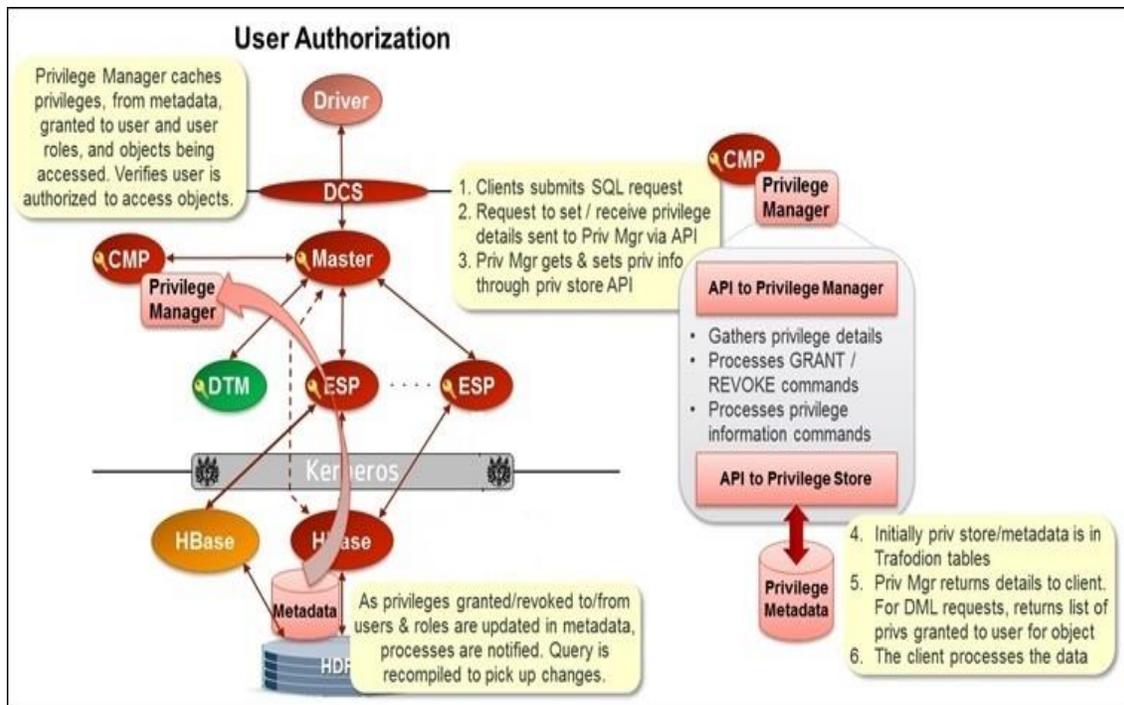


图 8-104 QianBase 用户验证

8.13.3 账户

每个人都担心自身的个人信息被泄漏，例如，社会安全号丢失、被盗或错放。因此，保护所有敏感信息至关重要。但如果泄漏了敏感信息，最重要的是找出哪些信息被泄漏、泄漏的原因和防止再次发生。您需要采取预防措施来修复泄漏造成的问题。记录活动是提前检测问题和事后分析问题的方法之一。您能掌握活动的全部细节并使用正确的工具了解泄漏的影响，从而降低未来发生安全泄漏的概率。

有效日志包括日志生成、日志存储和日志检测。日志生成决定记录的内容和创建合适的事件。日志存储管理时间存储的位置和存储时间的时长等。日志监控允许用户浏览当前活动和事后分析的事件。

SIEM（安全信息和事件管理）解决方案提供日志存储和监控方法，它跨系统从各类日志中收集信息，将事件集中到同一位置，为用户提供报表工具。使用 QianBase 时，建议您使用任意 SEIM 解决方案。

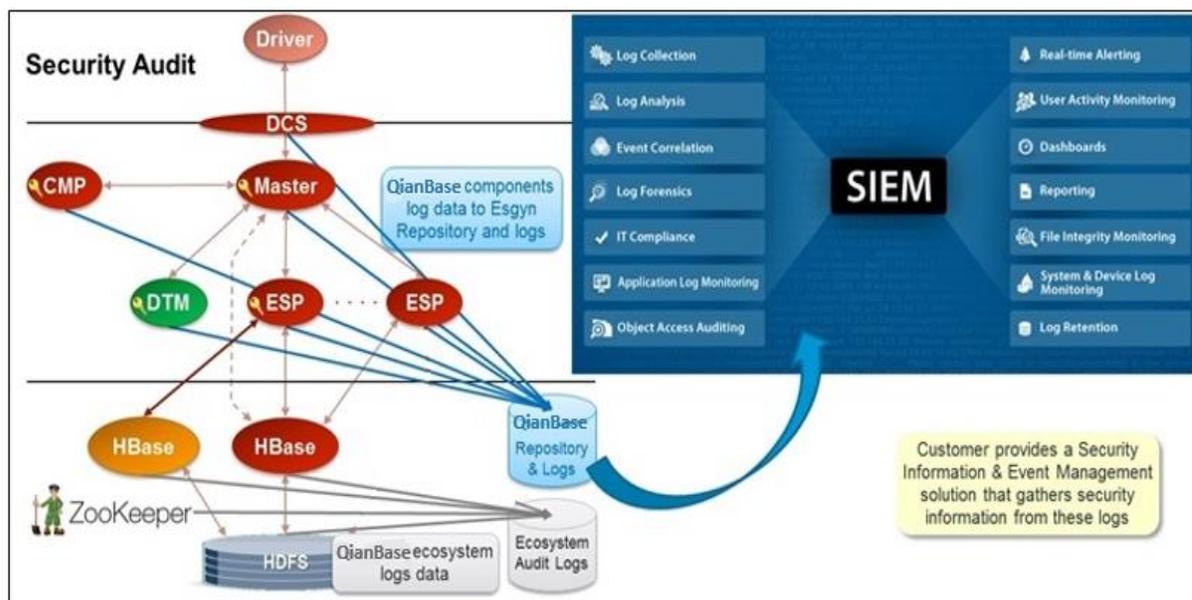


图 8-15 SIEM 日志存储和监控

8.13.4 数据保护

为了保证数据的一致性，QianBase 运行在安全平台上，数据存储在全生态系统中，该生态系统仅能被有权限的用户访问。

以下为某些领域存在的风险。

8.13.4.1 外部威胁-通过网络接口访问 QianBase

安全的第一层是防火墙，限制访问所有连接至集群外部网络的节点，访问外网的节点可能是头节点和故障转移节点、登录节点和 I/O 节点。为了提高安全性，您需要设置防火墙并开放正常使用 QianBase 的端口。该端口列表和 QianBase 授权都能减少外部访问威胁的风险。

8.13.4.2 集群级访问-非授权访问 QianBase 组件

网络非法分子能通过欺骗 (spoofing) 或其它方法访问文件和进程，因此，您应正确分配文件权限、限制访问文件和进程所在的集群、使用工具 (例如，sudo) 或其它产品 (例如，PowerBroker) 限制具备访问集群权限的用户的操作，以减少未授权访问并减少欺骗。

集群提供安装过程管理和集群内实例配置。每个 QianBase 实例负责各自的配置和管理。

QianBase 需要一个名为 *trafodion* 的 Linux ID 和一个名为 *trafodion* 的对应组。用户和组在安装 QianBase 时被创建。QianBase 进程作为 *trafodion* 运行。*trafodion* 拥有 HDFS 和 Linux 中文件的权限。

您应特别注意以 *trafodion* ID 身份连接至集群的情况。*trafodion* ID 是一个匿名用户，任何以该 ID 身份运行的操作都应被记录。通过 DB Manager 和 REST 接口, QianBase 提供多种操作。以 *trafodion* ID 的身份登录只能执行与 *trafodion* 相关的操作。

8.13.4.3 应用程序/SQL 命令访问-使用标准 SQL 接口对数据进行非授权访问

您还应考虑用户可能越权访问。使用数据库授权检查和对数据进行加密能限制应用程序和 SQL 命令访问。

8.13.4.4 加密

您能使用多种方法对未被使用的数据和使用中的数据进行加密。当数据未被使用时，您能利用硬件和软件算法对数据进行加密；当数据被使用时，您可以对数据进行解密。如果加密数据的密钥被网络非法分子破解，那么数据可能遭受损坏。

为了提高操作的安全性，通常使用中的数据被解密，并在应用软件中使用。数据可能被伪装成信任用户的欺骗软件损坏。

数据传输是数据流经网络边界。中间人攻击 (Man-in-the-middle Attack) 可能拦截并损坏数据。

您需妥善处理不再需要的数据。如果数据信息仍以某种格式存在数字设备中，则数据可能被损坏。

QianBase 使用 Hadoop 和不同 Hadoop 供应商的加密解决方案：

- HBase 提供透明加密保护。
- HDFS 提供 HDFS 级透明加密保护、密钥管理解决方案 (Key Management Solution) 和加密区域。

QianBase 使用 AES 加密算法对数据进行加密和解密。

8.14 运营性能

8.14.1 YCSB 基准

QianBase 是关系型数据库引擎，使用 YCSB 测试能将 QianBase 处理运营型读写操作与在 HBase 上执行 NoSQL 操作进行对比。与直接使用 HBase 的应用程序相比，QianBase 使用原生 HBase。理想情况下，随着开销大幅上升，QianBase 的性能不会下降。

YCSB 是一个开源 NoSQL 基准，包括[六种工作负载](#)，覆盖混合读写操作。无论使用流式数据或 OLTP 应用程序，使用 HBase 处理运营型工作负载时，数据注入速度和读写等比混合操作（模仿不间断的、经常需要更新数据的运营型工作负载）至关重要。

如图 8-16 所示，在更新性能方面，QianBase 与 HBase 的表现持平。

图 8-17 为 QianBase 和 HBase 在读写各占一半的混合场景的性能变化趋势。HBase 测试使用运行在相同集群的 T2 JDBC 驱动；为了模拟更真实的网页应用程序和存储引擎服务器环境，QianBase 使用 T4 JDBC 驱动。



图 8-16 YCSB 基准：
QianBase 与 HBase 的更新性能对比



图 8-17 YCSB 基准：
QianBase 与 HBase 读写各占一半的混合场景的性能对比

8.14.2 Order Entry 基准

Order Entry (OE) 基准从 TPC-C 基准衍生发展而来，它的表集和事务集与 TPC-C 基准非常相似。以下是 OE 基准与 TPC-C 基准的主要区别：

- TPC-C 基准模拟用户的想法和键盘输入时间，OE 基准持续不断地发起事务，产生大量负载压力，模拟不间断的事务流。
- TPC-C 测量 tpmC 速度，该速度为 New Order 事务的速度。OE 每分钟测量的事务包括所有 TPC-C 事务的性能，即 New Order、Payment、Order-Status、Delivery 和 Stock-Level。Order-Status 和 Stock-Level 模拟查询工作负载，New Order、Payment 和 Delivery 模拟混合读取和更新操作的工作负载。

如图 8-18 所示，当并发度增加至 384 时，QianBase 每分钟能执行 191916 个事务（最大值）。黄线表示使用 auto-commit 时事务的速度，如图 8-19 所示，YCSB 和 OE 测试在 8 个节点的配置上完成。

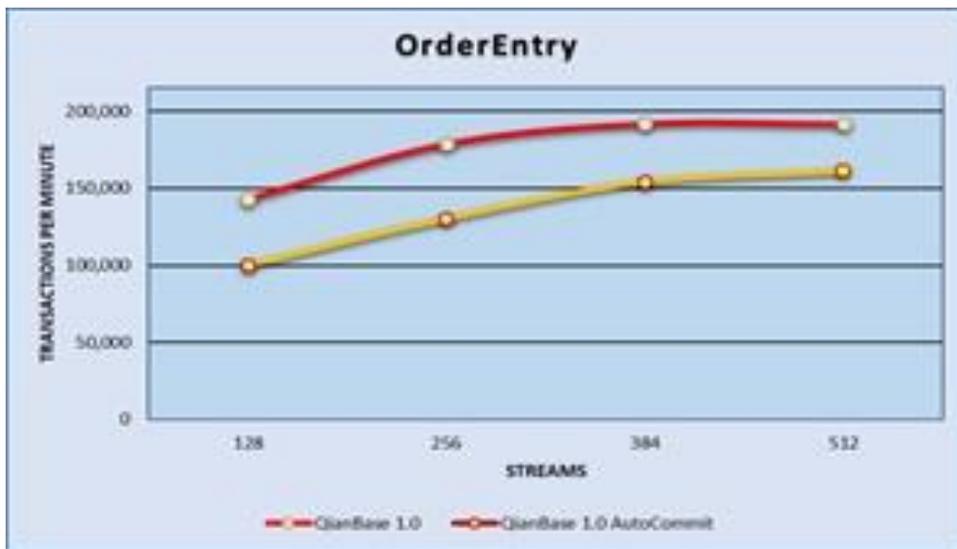


图 8-18 OrderEntry 基准测试 1：
每分钟执行事务对比：QianBase 1.0 与 QianBase 1.0 (AutoCommit)

System under Test	Performance Instance
Nodes	8 x Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz
Memory	256 GiB per node
Cores	10C/20T x 2 with HyperThreading turned on
Data Drives	6 per node (2.0TB 7200 RPM SAS 6 Gbps)
OS	Red Hat Enterprise Linux Server release 7.4 (Maipo)
Software	CDH 5.13.3 Parcels

图 8-19 OrderEntry 基准测试 1 配置



图 8-20 OrderEntry 基准测试 2:
QianBase 1.0 每分钟执行事务