

昆仑分布式数据库产品技术介绍

泽拓科技 ZettaDB.com

赵伟

Agenda

- 产品定位&业务模式
- 产品技术概况
- 架构和技术特点
- 产品技术优势

泽拓科技 ZettaDB 昆仑分布式数据库

- 核心技术团队简介
 - Oracle 从事MySQL, Berkeley DB等数据库内核研发多年
 - 腾讯TDSQL团队主力开发者, 主导开发TDSQL2.0, 推动TDSQL由分库分表中间件升级为分布式数据库系统;
 - 华为2012实验室技术专家;
 - 2019年8月启动昆仑分布式数据库项目, 目前完成所有核心功能
 - 目前有十多位核心产品开发者
- 泽拓科技
 - 2020年底成立 泽拓科技 ZettaDB www.zettadb.com
 - 面向新时代海量数据存储管理利用需求
 - NewSQL OLTP分布式数据库产品和DBaaS云服务
 - 2021年上半年完成数千万元人民币天使轮融资

昆仑分布式数据库产品定位

- 具备完备的NewSQL能力并且全面支持SQL标准的MySQL分布式数据库管理系统
- 面向高性能、高并发、高可扩展、高可靠的海量数据存储管理利用的需求
 - 降低各行业用户的应用系统研发复杂度，提升开发效率，加快其迭代和上线周期
 - 提升其业务系统的可靠性和业务处理能力，助力客户的业务系统为其终端用户带来完美的使用体验
- 帮助使用应用层分表 & 中间件分表的用户简化应用系统设计研发
 - 原架构下，业务侧开发者承担部分数据管理功能开发任务
 - 技术复杂易错，开发周期长、产品质量无法保障，以及上线和迭代进度不可测，人力成本高
 - case by case的解决数据管理和容灾问题，可靠性低，软件复用度低，开发任务量巨大，变更困难
- 兼容微服务架构
 - 所有微服务共用同一个分布式数据库集群，协作更加简单，省去了很多消息队列
 - 服务扩容更加简单，服务节点完全无状态
 - 简洁的事务处理架构：服务节点总在**当前事务**中执行DML

昆仑分布式数据库产品定位

- 公有云DBaaS
 - 与各公有云平台合作
 - 融入公有云平台基础设施（存储，数据库，数据分析，调度管理，监控）
 - 例如：Kunlun + Aurora, Kunlun节点容器 + K8S
- 面向全球各行业的PostgreSQL和MySQL用户群，借力两大社区的人力资源和技術资源
 - 人力资源：海量的经验丰富的DBA和应用开发者
 - 技术资源：海量的周边工具和技术经验知识积累
- 一套集群原地做数据分析，免除数据灌入，支持Spark所有分析功能
 - Spark：大数据量导致OOM等问题
 - OLTP与OLAP使用不同的计算节点和存储节点，互不干扰
 - 并行查询处理实现高性能
 - 行存：能够高性能流式灌入

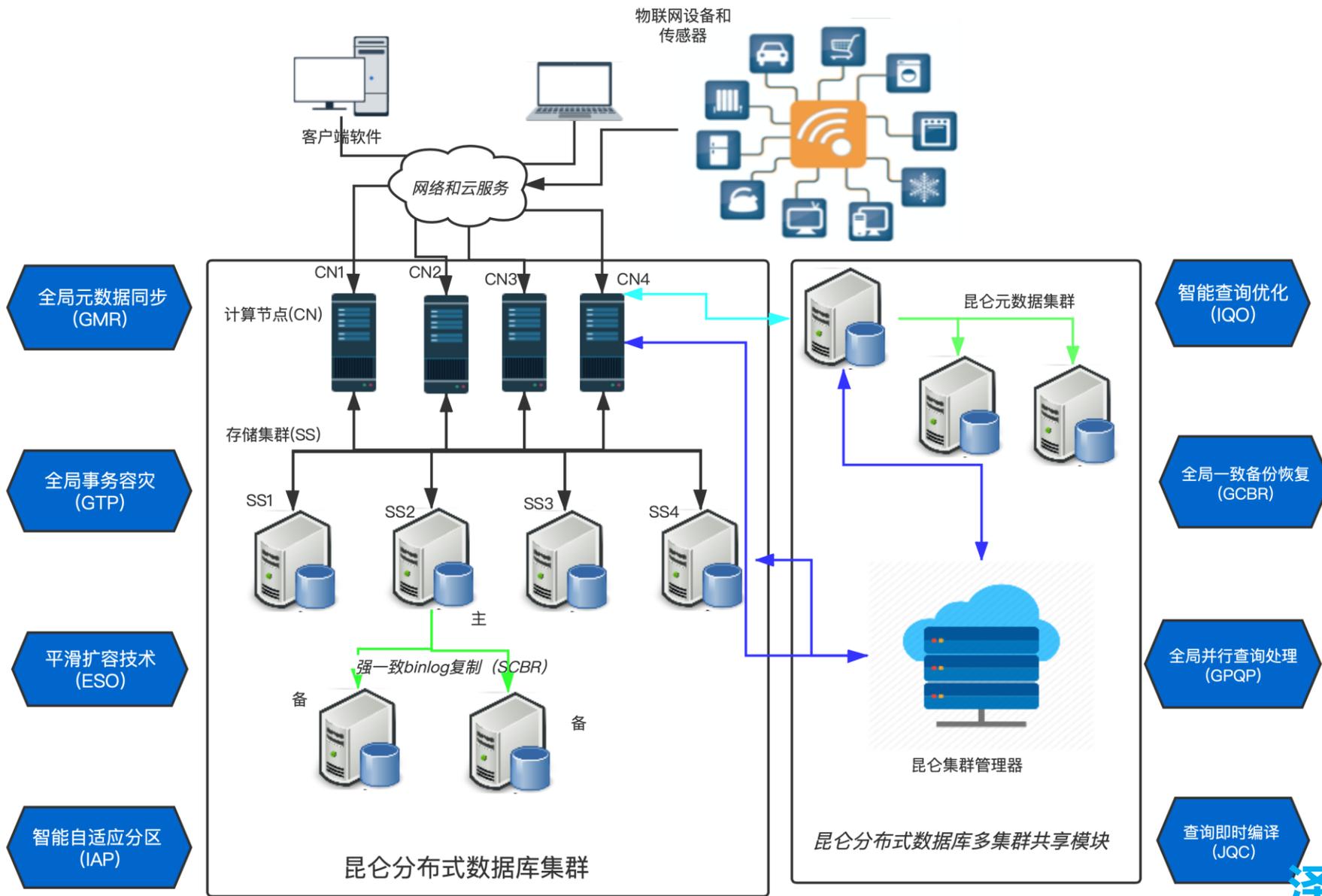
昆仑分布式数据库概况

- 高性能高效率
 - sysbench性能数据行业领先 <http://www.zettadb.com/blogs/perf-cmp1>
- 强大的NewSQL能力：大大降低应用开发者管理海量数据的技术难度，像使用单机数据库一样简单
 - 容灾（分布式事务处理），高可用，强一致性
 - 数据自动分区，水平弹性扩容
 - 完备的SQL支持（分布式查询处理）
- OLTP & OLAP：同时支持，互不干扰
 - 数据按行存储，面向OLTP 高性能高并发高吞吐度量场景；
 - 具备所有OLAP SQL功能，适合原地做OLAP分析
- 基于开源，大量内核研发成果，已经开源
 - 10万多行新增的数据库内核代码
 - 将PostgreSQL和MySQL融合为一个新的产品，产生1+1>>2的价值
- 融入业界数据处理生态
 - SQL生态：各种数据处理和机器学习算法，OR映射中间件
 - MySQL binlog生态：基于消费binlog事件的数据处理机制，例如flink，kafka等

昆仑分布式数据库业务和交付模式

- 公有云服务
 - 使用企业版本登录各公有云平台提供DBaaS
 - 可以与Aurora/PolarDB等MySQL数据库分支协作
 - 接入各公有云的云服务体系和/或调用公有云接口完成DBaaS服务部署
 - 公有云的Mall模式 VS 百货公司模式：国内3大云平台应该向AWS看齐
- 私有云和传统商业软件模式
 - 企业版产品销售和技术支持服务订阅
- 企业版本
 - 独有的性能增强&面向企业场景的周边辅助工具
 - 高优先级的需求开发和bug修复以及专家技术支持服务
- 开源版本
 - 功能与企业版始终保持完全相同
 - 与企业版本可以相互替换，相互兼容数据文件格式

昆仑分布式数据库架构



昆仑分布式数据库架构

- 计算节点 (Computing Nodes, CN)
 - 接受和验证用户连接请求
 - 接收和处理来自用户连接中的DDL/DML SQL语句
 - GTP/GPQP: 全局事务处理和全局查询处理
 - GMR: DDL事务并发执行和并发复制
 - 数量按需增减, 互相独立, 本地元数据相同
 - 不存储用户数据, 只存储元数据
 - 占用微量存储空间 (若干MB)
 - 本地存储会话/连接临时数据 (临时表和物化视图)
 - 用户数据存储存储在存储集群中
 - 异步读写存储集群访问用户数据
 - 本地状态可用集群的元数据重建
 - 无需为计算节点本地数据做容灾
 - 不会给DBA和运维带来额外管理工作

昆仑分布式数据库架构

- 存储集群 (Storage Shards, SS)
 - 存储应用 (用户) 数据
 - 执行计算节点发起的XA事务分支
 - 目前使用MGR单主模式做集群高可用
 - 自动主选举
 - 健壮的一致性保障
 - 目前只支持innodb引擎
 - 不使用非事务引擎(myisam, etc)
 - 未来可能支持myrocks
 - 必须使用 Kunlun-Storage
 - 基于 Percona-MySQL-8.0.18-9 开发
 - 含有关键的社区版XA事务容灾bug修复和支持功能, 以及性能巨大提升
 - 会随上游版本升级

昆仑分布式数据库架构

- 元数据集群
 - 存储着若干个昆仑分布式数据库集群的元数据
 - commit log & ddl log
 - 所有节点连接信息以及存储集群信息
- Cluster_mgr
 - 维持每一个存储集群及其节点的replication状态
 - 集群计算节点与存储节点之间的元数据和状态同步
 - 分布式事务特定故障处理
- 周边工具
 - 集群安装、部署、监控
 - 备份、恢复、迁移
- 云服务化

昆仑分布式数据库技术特点 --- 高可用性 (HA)

- 兼容多种强一致性，高可用性方案 (strong consistency, high availability)
 - 存储shard: 默认使用MySQL Group Replication 保障数据库服务高可用
 - 2*N+1个节点的shard, 每个已提交的事务都复制到了至少N个备机

shard同时宕机或失联节点数x	shard仍然具备的能力	shard数据一致性保障
$x \leq N$	可读可写	可以保障
$N < x \leq 2*N$	可读, 未必最新数据	不可以保障

- 兼容基于mysql row based replication的半同步/强同步的高可用技术 (*)
- 兼容自带高可用机制的mysql分支, 比如Aurora, PolarDB, etc

昆仑分布式数据库技术特点 ---- 完备的容灾能力

- 完备的容灾能力 (crash safety&fault tolerance)
 - 存储集群主备强一致
 - GTP: 全局事务容灾能力
 - 分布式事务两阶段提交
 - 节点/网络故障时可以保障分布式事务ACID
 - GMR: DDL与集群全局元数据一致性
 - DDL事务涉及 计算节点, 元数据集群, 存储集群
 - GMR: 计算节点之间的元数据语句复制及其一致性保障
 - 复制过程随时可能中断
 - GTP: 计算节点自动切换存储集群主节点 (auto failover)
 - 元数据集群, 存储集群
 - GTP: cluster_mgr:自动维护各shard的存储集群复制状态

昆仑分布式数据库的技术特点 --- 水平扩展能力

- 高可扩展性 (high scalability) : 计算能力, 存储空间, 资源利用率
 - IAP: 灵活的sharding方式
 - 用户对sharding 方式拥有全部控制
 - sharding方式: hash/range/list, 未来mirror, UDF等*
 - 选择sharding列: 任意若干个列
 - 用户最理解自己的数据及其访问模式
 - 定制分区选项达到最优性能
 - **专业模式** VS **傻瓜模式**
 - 根据数据表的规模定制sharding方案
 - 不需要预估全局的固定的分区数目, 也不把所有表等分为固定数量的分区
 - 可以为每个表按需增加分区, 各表分区数量各异
 - 最少化两阶段提交的事务数量
 - 计算节点自动为每个分片选择合适的存储集群

昆仑分布式数据库的技术特点 --- 水平扩展能力

- 高可扩展性 (high scalability)
 - ESO: 自动按需扩展 (*)
 - 自动透明地分布数据表到新加入的存储集群
 - 业务和最终用户无感知
 - GPQP: 全局并行查询处理: 后摩尔时代, 利用更多的计算资源
 - 计算节点层的并行
 - 计算节点与存储节点之间的并行
 - 存储节点层的并行 (*)
 - 多点读写, 没有写入瓶颈
 - 按需增加/减少计算节点
 - 按需增加/减少存储shard (*)
 - 存储集群支持多点写入 (*)
 - 备机读 (*)

昆仑分布式数据库的技术特点 -- 查询处理

- 充分理解用户数据
 - 本地存储完备的元数据和数据字典
 - 本地存储完备的全局数据统计信息
 - 有条件产生最优的分布式查询计划和查询执行性能
- 完整的查询处理过程
 - parser->resolver->optimizer->executer
 - 可以处理任意SQL查询
 - 支持多表连接, 子查询, 聚集查询, CTE, window function, 存储过程, 视图, 物化视图
 - 完整的查询处理功能: 真prepared statement
- 完美支持OLAP查询
 - 查询处理能力完美支持大数据分析任务
 - 直接使用本地数据, 无需数据搬迁, 无需spark/hadoop生态
- 可以调用MySQL系统函数和用户定义的存储过程/函数

昆仑分布式数据库的优势 --- 扬众长避众短

- 集3大主要数据库Oracle, MySQL, PostgreSQL的强项于一身并产生1+1>>2的放大效应
 - Oracle: 存储引擎, 查询处理
 - innodb完全遵从 Oracle的存储引擎的设计
 - MySQL: innodb存储引擎和 binlog 复制 (RBR)
 - PostgreSQL: 查询处理能力在所有开源RDBMS中最强
 - MySQL&PostgreSQL开源社区的人力资源和技術资源
- 避免它们的弱项
 - Oracle: 硬件和软件昂贵, 成本过高; 无法做到安全可控, 政策合规
 - MySQL: 查询处理的性能和功能有限; 单机数据库, 无法水平弹性扩容
 - PostgreSQL: 存储引擎不适合重负载OLTP负载

昆仑分布式数据库的优势 --- 完备的查询处理功能

- 计算节点支持PostgreSQL的所有主要查询处理功能
 - 绝大多数DDL和所有DML语法和功能
 - 例外：外键和触发器， tablespace和存储相关功能， WAL replication
 - 所有基本数据类型
 - 数值， 字符串， text/blob， 时间/日期/时间戳/money/enum， 序列 等等
 - 高级查询处理功能
 - 任意跨shard的多表连接， 子查询， 存储过程
 - OLAP分析能力： 聚集函数， window函数， grouping sets， cube， rollup
 - CTE， 视图， 物化视图， 真prepared stmt， jit
- 计算节点兼容MySQL和Oracle的常用SQL语法(*)
 - 支持MySQL客户端协议 (*)
 - 去O的迁移工作量较小， 技术人员技能可平移， 学习曲线平缓

昆仑分布式数据库的优势 --- 全方位数据安全保障

- 在数据源头控制数据访问更加安全可靠
 - 统一/多层次/灵活动态 配置访问控制规则
 - 应用层面访问控制的多种缺陷
 - 不统一：多种应用访问同一个数据库，每个应用都需要规则配置甚至编码实现
 - 不灵活动态：硬编码的访问控制规则，不容易修改
 - 不安全：控制策略和规则本身会泄露信息
- 多层次细粒度的访问控制
 - 多层次的用户/角色
 - 多层次的数据库对象：数据库/schema/表/视图/列
 - 多层次管理各种数据库对象的访问控制规则
- 全链路SSL连接，安全传输数据
- 用户数据全系统加密：数据文件和binlog文件以及 WAL日志文件加密
- Prepared Statement防止SQL注入

昆仑分布式数据库的优势 --- 兼容并蓄

- 计算节点开放架构
 - Extension: 无缝兼容 PostgreSQL生态, PostGIS 等
 - 多语言 (python, java, lua, js, PLSQL) 编写存储过程实现In-Database-ML/Data Analysis
 - FDW (foreign data wrapper): 可以实现接口来读取所有主流数据源
 - hadoop生态: hbase, hive等
 - 主流数据库: Oracle, MS SQL Server, DB2, MySQL, PostgreSQL等等
 - 列存储 (OLAP) 和时序数据库
- 计算节点其他优势
 - 完善的i18n/globalization/localization支持
 - 时区, 字符集和collation
 - 多语言能力

昆仑分布式数据库的优势 --- 多层次多方面的可扩展性

- 按需弹性水平扩展能力
 - 多个读写节点，处理读写负载都可以按需扩展处理能力
 - 无共享 (share nothing) ，无单点依赖
 - 无性能瓶颈，无计算/存储能力瓶颈
 - 按需增减计算节点和存储集群/节点
 - 透明的按需扩展，业务系统和最终用户无感知
 - 存储集群扩容速度可调，对数据源节点的计算/存储/网络资源消耗可控
- 全系统并行计算能力
 - 充分发挥服务器的并行工作能力
 - 多核并行
 - 存储系统并行
 - 网络系统并行

昆仑分布式数据库的优势 --- 其他

- 存储集群
 - 性能领先：分布式事务处理性能大大高于社区版本 <http://www.zettadb.com/blogs/perf-cmp-mysql>
 - 完备的容灾能力：填补社区版MySQL 8.0的分布式事务处理的容灾能力空白
 - https://fosdem.org/2021/schedule/event/mysql_xa/
 - 本技术分享视频的国内地址：<https://www.bilibili.com/video/BV1zo4y1d7pu>
 - MySQL 8.0数据文件加密，binlog/事务日志文件加密
 - 原生的online DDL功能：快速加列
- 昆仑数据库 VS MySQL：使用昆仑数据库管理小规模数据的优势
 - 放大了单一MySQL集群的能力，按需水平扩展能力和更强大的数据分析能力
 - 更简单方便地使用MySQL集群：自动切主并维护mysql集群状态
 - 并行查询处理，备机读
- 集群结构简单，不依赖第三方模块和软件（etcd/zookeeper等）
 - 产品质量可控
 - 人力负担小

Q&A

Thank You

Zhao Wei