



# 云栖干货回顾 | 行业顶级NoSQL成员坐阵，NoSQL数据库专场重点解析！

转载

Taoo  于 2019-10-30 16:26:21 发布  256  收藏 1

原文链接: [https://yq.aliyun.com/articles/721071?utm\\_content=g\\_1000084194](https://yq.aliyun.com/articles/721071?utm_content=g_1000084194)

版权

NoSQL数据库作为数据库市场最重要的组成之一，它的一举一动都影响着成千上万的企业。本专场邀请了行业顶级的NoSQL核心成员与大家共同展望NoSQL数据库的未来，阿里巴巴、MongoDB、Redisson、斗鱼等公司的技术大咖与大家共同分享了阿里云NoSQL数据库的企业级特性及行业解决方案。

## Redis & MongoDB云数据库技术剖析

阿里云智能事业群数据库产品事业部技术总监，MongoDB中国用户组杭州用户会主席杨成虎(叶翔)为大家深度剖析了Redis和MongoDB云数据库的技术。

### Redis企业级数据库服务

Redis作为企业级数据库需要关注四个方面：

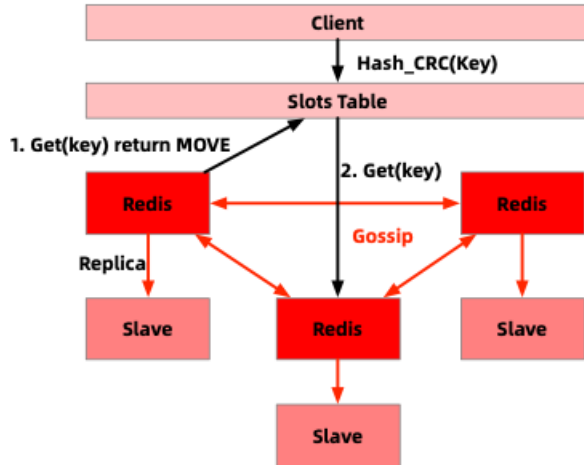
- 分布式：需要满足企业快速成长和降低成本的需要，实现弹性扩容，以及从主从模式变为集群模式。
- 兼容性：兼容性是永恒的话题，即使无法做到100%一致，但需要无限接近。
- 安全审计：安全在云环境中越来越重要，Redis开源版的安全审计能力比较薄弱，阿里云Redis对于这一点进行了加强。
- 数据同步：需要能够支持混合云部署，使得第三方云厂商、IDC与阿里云实现互通，以及数据迁移和转换，满足客户上云或者下云的灵活决策。

Redis原生的Cluster架构采用了Gossip协议实现路由表的同步，但这种架构在社区以及企业中并没有快速流行起来。虽然其有无中心架构、组件依赖少等优点，但也存在很多问题，如运维困难，路由存在不确定性，需要依赖Smart Client，并且不支持Multi-Key以及从主从模式迁移到集群模式，进而造成升级困难。

Alibaba Group | APSARA 阿里巴巴集团 | 云栖大会

# 分布式：Redis Cluster原生架构

New Future on Cloud



优势：

- 无中心架构
- 组件依赖少

缺点：

- 运维困难，路由不确定性
- Smart Client依赖
- Multi-Key命令不兼容
- 无法从Master-Slave迁移到Cluster

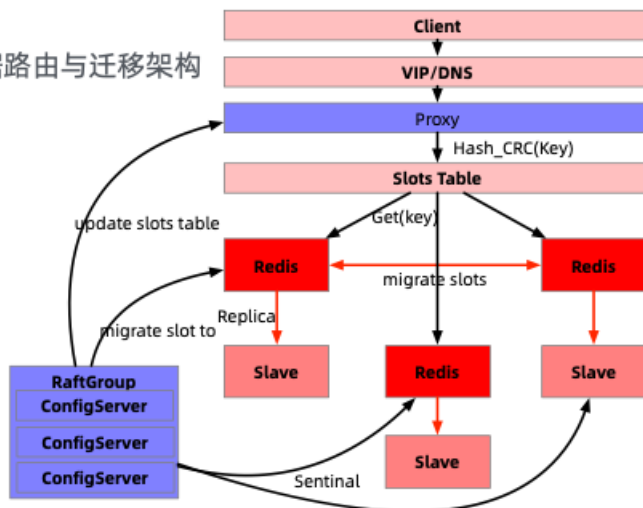
为了解决上述问题，阿里云Redis数据库没有采用Gossip协议，而是引入了新的两个组件：Proxy和Config Server。阿里云Redis采用了配置中心对于路由表信息进行管理，可以通过Config Server进行智能化调度，Proxy则能够兼容非Smart Client，支持Multi-Key，并能够实现流量管理以及读写分离等。Proxy和Config Server虽然带来了架构的复杂性，但管理大规模复杂架构正是云厂商所擅长的。此外，这两个新组件所造成的额外成本也会被削平。通过这样的云服务架构使得用户能够将Redis从主从架构无缝迁移到集群版本。

Alibaba Group | APSARA 阿里巴巴集团 | 云栖大会

# 分布式：Redis Cluster云服务架构

New Future on Cloud

复用Slots数据路由与迁移架构



ConfigServer：

- 中心化管理Slots路由
- Slots Balance调度

Proxy：

- 兼容非Smart Client，支持Multi-Key等
- 流量管理，读写分离，Retry等
- 后端DB分层，深度优化定制

服务价值：

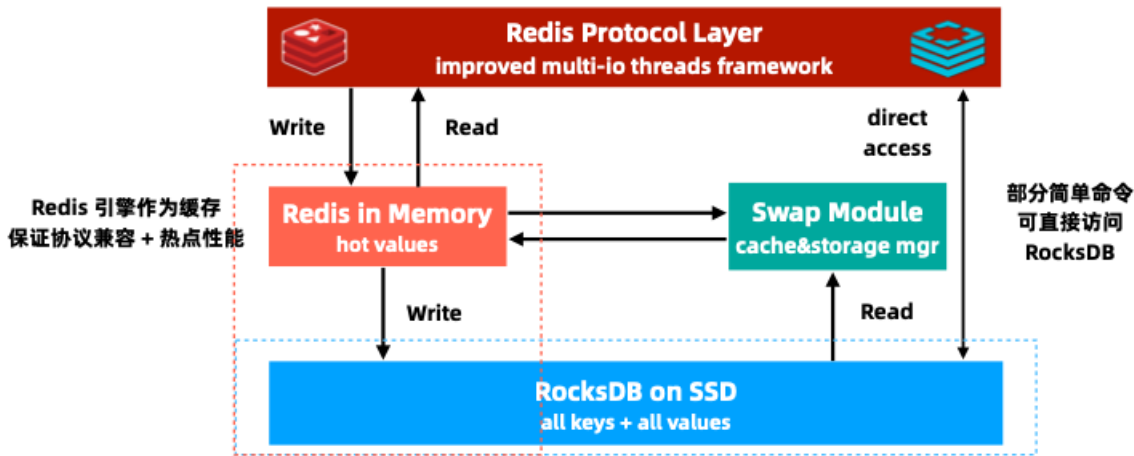
- Master-Slave无缝升级Cluster

随着Redis Cluster云服务架构的延伸，出现了一个新概念——Redis云数据库企业分布式矩阵。这个矩阵能从纵向和横向进行扩展，纵向能够随着Shard的添加进行分片，从而实现弹性扩展；横向则能够实现读写分离，并且做了Group分组隔离。全局来看，还支持Memcache和Redis双协议，并且能实现集群、主备之间的平滑切换。

阿里云Redis的Proxy引入了Connection Session的概念，能够对于Connection实现更细粒度的管理，并且通过连接池实现了长连接复用，不仅能够兼容多种协议，并通过C语言高性能代码也提升了短连接的性能。阿里云Redis的Proxy还具有热升级能力，能保证在服务不间断的情况下升级版本。

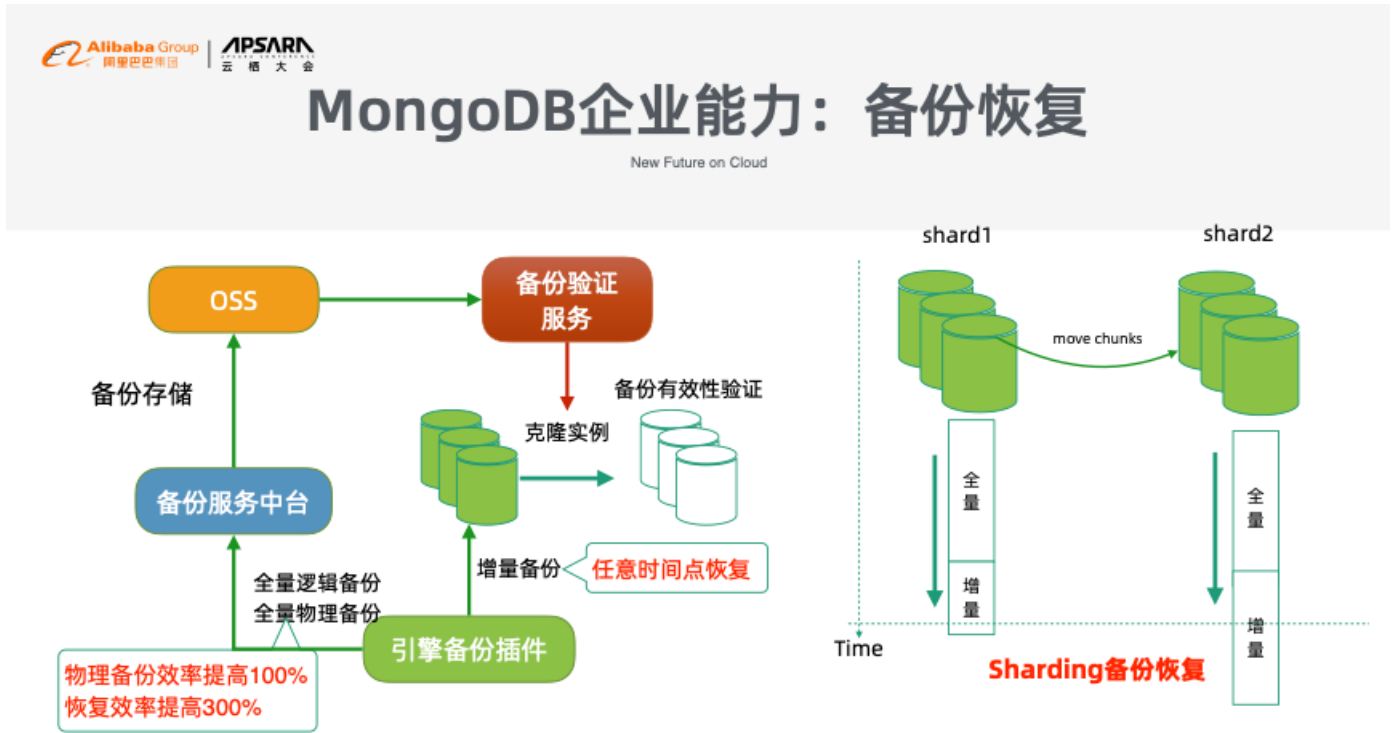
阿里云Redis在整个数据链路上进行了逐层加密处理，支持了SSL、白名单、权限管理以及关键命令的禁用和审计等，增强了Redis的安全审计能力。Redis还提供了一些免费的开源工具，如同步工具RedisShake以及数据校验工具RedisFullCheck等。

而Redis作为内存型的缓存服务也存在很多挑战，比如容量受限，成本较高以及持久化能力弱等。基于以上问题，阿里云提供了混合存储的Redis版本，其目的在于为用户提供持久化、可安全存储的Redis服务。其实现依赖于底层的RocksDB，通过不断同步冷热Key，使得内存处于可控范围之内。



### MongoDB企业级数据库服务

MongoDB作为企业级数据库需要关注四个方面，即安全审计、备份恢复、数据同步以及弹性伸缩。MongoDB的安全审计与Redis基本一致，进一步增加了TDE加密。MongoDB增加了物理备份，使得备份和恢复效率都有了大幅度提升，并且通过增量备份能力使得数据能够恢复到任意时间点。此外，在备份的基础之上，阿里云MongoDB还提供了备份验证能力。



阿里云MongoDB还提供了诊断分析能力，并提供了MongoShake工具对数据进行同步。阿里云MongoDB基于RocksDB引擎实现了共享存储解决方案，可以实现存储弹性伸缩，秒级添加只读节点，并解决了oplog全局锁问题。当然，这样的方案也面对着几点挑战，如与WiredTiger的兼容性问题；Compaction带来的性能抖动；以及共享存储延迟稳定性。

## 基于MongoDB的数据中台技术实现

MongoDB大中国区首席架构师唐建法为大家介绍了基于MongoDB的数据中台技术实现。

如果企业业务需要对接不同的客户数据，而这些数据的结构、类型各不相同，可能需要花费数周甚至数月。很多已有的解决方案就是实现数据统一平台，将所有数据通过ETL抽取到数据平台上，这种方式的共性是“T+1”的方式批量采集汇总，做成数据集，以交互方式提供下载。但这种方式存在着平台数据滞后、响应速度慢、交互方式粗糙等问题。

数据中台从技术的角度进行定义就是“数据统一平台+数据即服务能力”。数据来源于业务，需要按照“T+0”方式采集，提供及时的数据。数据需要以API的服务化方式交付出去，而非打包。这使得数据中台能够对企业赖以生存的操作型系统提供支持，相比于分析型业务，操作型业务更加核心，更加能够提高企业竞争力，这也是数据中台火爆的原因。

数据中台的定义就是包含企业实时全域数据的，主要面向操作型业务应用为主的数据服务技术平台。其概念起源于国内，存在众多流派，众说纷纭。咨询公司说数据中台是一种组织架构的转变，方案提供商则说数据中台是像Hadoop一样的技术平台产品，不同的组织有不同的出发点。

中国97%小微企业与数据中台基本不相关。腰部占3%的120万家大中型企业，可能有很多的开发人员但没有数据专家，另外还有少部分头部企业。对于腰部的大中企业而言，系统可能不多，而数据团队基本没有，无法快速构建完善的数据中台，但是数据孤岛的痛点、数据打通以及快速开发的需求却是真实存在的。这些企业可以选用技术型架构，具体需要考虑的能力包括数据汇聚、数据治理以及建模、数据API服务，以及最关键的存储、海量、多模和高性能。

RDBMS、MPP、Hadoop、NoSQL以及NewSQL数据技术各有长短板，在构建中台时也可以做一些参考，企业需要根据自身实际情况进行考量。

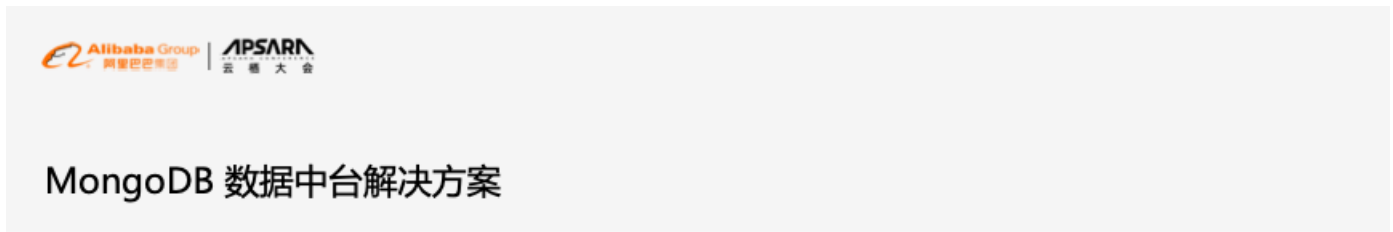
## 数据技术长短板

	海量数据 VOLUME	速度与并发 VELOCITY	多结构数据 VARIETY	事务支持 ACID
<b>RDBMS</b>	■■■■■	■■■■■	■■■■■	■■■■■
<b>MPP</b>	■■■■■	■■■■■	■■■■■	■■■■■
<b>Hadoop</b>	■■■■■	■■■■■	■■■■■	■■■■■
<b>MongoDB</b>	■■■■■	■■■■■	■■■■■	■■■■■
<b>NewSQL</b>	■■■■■	■■■■■	■■■■■	■■■■■

之前，MongoDB用于大数据离线分析并不是很好的选择，更多地是将其用于业务场景。而数据中台面向的就是业务应用场景，因此MongoDB成为了一个不错的选择，其具有较强的横向自动扩展能力，支持多模多态，并且API友好。此外，基于MongoDB实现建模所需要的工作远少于传统方式，能够降低成本。

此外，MongoDB还具有数据采集、可视化建模、无代码化API、数据可视化等数据中台构建所必须的能力。

如下图所示的是较为完整的MongoDB数据中台解决方案参考架构，从下到上依次是采集、存储处理以及数据服务三层。



基于MongoDB构建数据中台具有这样几个核心优势，即无缝横向扩展能力、多类型结构数据模型、逻辑模型即存储模型、异构实时数据库同步能力、无代码快速API发布能力以及简单、轻量和快速。

## 图数据库GDB的设计与实践

阿里巴巴资深技术专家朱国云(宗岱)为大家分享了阿里巴巴图数据库GDB的设计与实践。

### 什么是图数据库

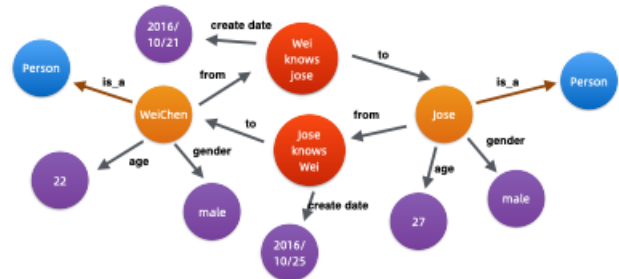
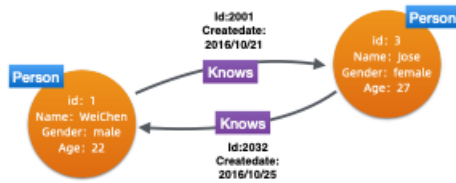
图数据库是针对于图结构设计数据库，而非图片数据库。什么是图结构呢？这是以社交网络模型为例介绍，该模型中存在人与人、人与论坛、人与帖子、帖子与论坛之间的关系，人、论坛、帖子就属于图中的点（即Vertex），点之间的关系就称之为边（即Edge），在点和边上会有一些属性（即Property）。

如今，一些优秀的社交应用会将多维数据存储到统一的图空间中来，进行存储、查询和分析，为用户带来更好的体验。近年来，数据量越来越大的同时，数据维度也逐渐增多，图数据库就是在这种背景下诞生的。

图数据库作为近年来数据库领域中发展最快的一类，与关系型数据库存在哪些差别呢？通常情况下，关系型数据库中需要通过建七八张表才能做到的模型，图数据库能够更加直观、自然地表现出来。此外，图数据库做关联查询的速度更快，还能够提供更多探索发现的能力。

前面提到的是属性图模型，在图数据领域还有一种RDF模型。两者的主要区别在于RDF的点和边上不可以有属性。

## 属性图和RDF



图数据库发展速度很快，因此种类也是特别多，主要可以分成四类，即知识图/RDF、分析图、图数据库、多模型图数据库。这些图数据库系统使用的主流查询语言大致有三类，即Neo4j主推的最早使用类SQL查询语言的Cypher、用于RDF上的描述语言SPARQL和目前支持最广泛的基于属性图的查询语言Gremlin。

### 什么是图数据库GDB

GDB是一种图数据库，其主要处理高度连接数据的存储和查询，其支持了属性图模型和开源的TinkerPop Gremlin查询语言。与其他数据库不同的是：GDB是云原生数据库，从一开始就建设在阿里云基础设施之上，因此能够做到弹性、实时和高可靠。

GDB脱胎于Tair Service中的TairGraph子系统，后来其孵化出来，并放到阿里云上来专注地解决高度连接数据场景中的问题。基于Tair 10年的技术基础，GDB实现了高度优化的自研引擎，能够实现实时更新和秒级查询，并且完整地支持ACID事务，并通过多副本保障高可靠。此外，还做到了服务高可用，能够实现节点故障迅速转移；易运维，提供了开箱即用的能力；可视化，更利于分析数据的内在关系。

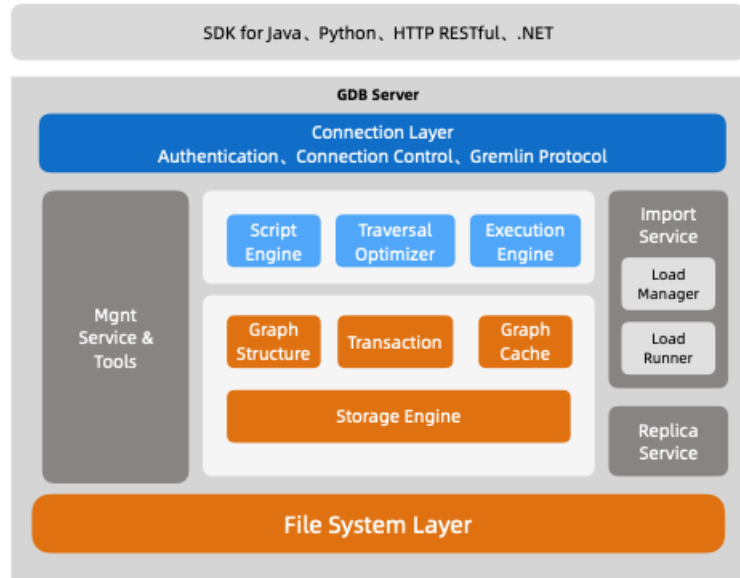


在架构层面，GDB为客户提供的独享专属实例，这意味着资源独立，无须担心抢占问题。HA方面采用了最经典主备架构，并提供只读节点来提升实时查询能力。GDB支持了Gremlin开源的TinkerPop SDK，为了实现每秒百万级点边过滤，GDB定制了专属的图友好数据库引擎，并在查询优化和并行执行等方面做了大量优化，还支持了事务和自动索引。在数据通道部分，GDB还提供了多种数据源的高效导入支持。



## 系统架构

- 支持属性图
  - Gremlin规范，基于开源TinkerPop
- 自研图数据库引擎
  - 查询优化，并行执行
  - 支持ACID事务
  - 自动索引
  - 定制Graph-Friendly存储引擎
  - HTAP
- 数据通道
  - 高效导入
  - 数据同步



### GDB的场景和案例

如今，GDB在社交网络、金融欺诈检测、实时推荐引擎、知识图谱以及网络/IT运营等场景中得到广泛应用，而且这些场景往往交织在一起。使用GDB能够将之前偏离线的场景做到实时或者准实时。

总结而言，在数据维度越来越多、数据相互关联越来越紧密的今天，GDB提供了一种有效的图存储方式，能够将多维数据很好地连接起来，并通过图查询、图算法把数据隐藏的价值实时地、智能地挖掘出来。

### 从Java走向云原生，Redisson不停地探索

Redisson联合创始人顾睿为大家分享了Redisson从Java走向云原生的探索之路。

Redisson是架设在Redis基础上的一个Java驻内存数据网格。Redisson以Java接口方式而非命令的形式提供给大家，使用非常简单。其优势在于上手容易，只要能够使用Java基本就能够使用Redisson。此外，Redisson在设计时规避了多线程的问题，采用了线程安全的设计，同时引入了线程池和连接池的管理，在同步和异步的场景中都能选到适合的方式。

除了使用简单外，Redisson在功能上也提供了多种选择，能够支持31种分布式集合、14种分布式对象、8种分布式锁和分布式同步器以及5种分布式服务。

Redisson的架构主要分为两大块，包含Redisson客户端的连接管理、协议解析在内的基本功能和包括分布式结构、分布式中间件以及第三方功能支持在内的高级功能。

从Redisson架构角度来看，似乎和Redis的理念相冲突。Redis设计理念强调简单，而Redisson设计却比较复杂；Redis提供了9种数据结构，界限清晰，而Redisson提供了约60种，界限比较模糊；Redis以命令形式面向用户，而Redisson却以Java API形式面向用户。看似分道扬镳，实则殊途同归，都是为了将复杂隐藏起来，将简单的使用方式提供给用户。

只支持Java是Redisson的优点，也是缺点。Java是Redisson的一个牢笼，这对于应用程序开发者而言是优势，而对于程序库开发者而言就是劣势。因此，Redisson一直在思考如何走出困境，拥抱其他的生态。

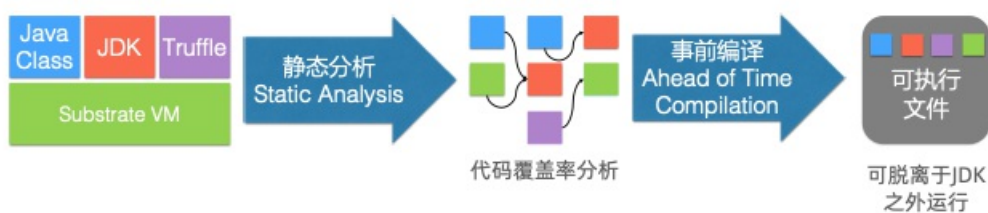


2016年，Redisson首先尝试了使用Vert.x框架，Vert.x的特点是集群运行环境、多语言交互性和基于成熟技术，并且Vert.x对开发者的限制比较少。因此，Redisson做了相关的实验，实现了Redisson在其他语言中的运行。但是这种方案学习成本非常大，而实际收益却不高。

2018年，Redisson注意到ORACLE Labs推出的GraalVM，GraalVM的底层是Java运行层，包括GraalVM和SubstrateVM，可以让其他语言都能够编译融合并放入JVM中执行，同时保证相互沟通的桥梁。SubstrateVM是最吸引Redisson的点，它可以理解为由Java编写的嵌入式虚拟机，使得真正的跨平台和跨语言成为可能。

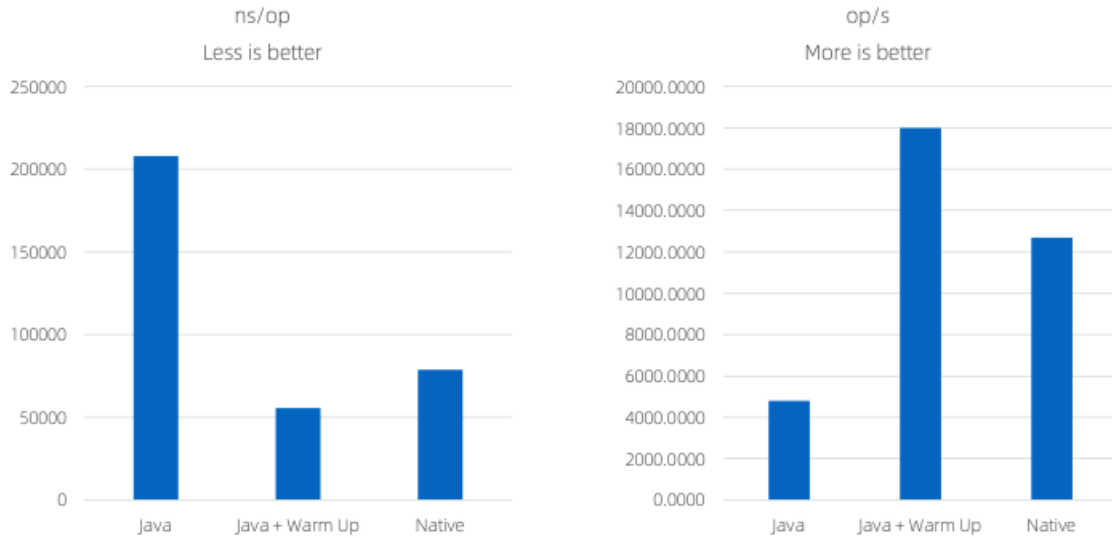


## SubstrateVM介绍



于是，Redisson开始了“逃跑之路”，实现了redisson-native。对于Java、Java+Warm UP以及Native三种方式的性能进行对比，能够看出redisson-native的性能具有明显的优势。

## redisson-native性能



因此，这说明借助SubstrateVM逃离Java是非常好的解决方案，无需考虑JNI等相关问题，大部分操作只需要Java即可完成，学习成本较低，并且无需安装独立的JVM，生成文件也较小，云原生情况下性能较高，并且C调用非常简单。延伸开去，可以将Redisson带入到原生的二进制状态并进行二次封装，实现遍地开花。

### 基于企业级HBase的大数据存储与处理

阿里云智能事业群数据库产品事业部技术总监，Apache HBase PMC沈春辉(天梧)为大家分享了基于企业级HBase的大数据存储与处理。

进入大数据时代，数据量越来越多，数据种类也越来越丰富。数据量多这一点容易理解，而数据种类丰富则可以从三个维度来看：从静态维度，如今能够用数字化设备越来越多；从动态维度，设备、服务的运行状态越来越多；此外，还有数据再加工又产生了新数据，使得数据变得无穷无尽。面对这么多数量和种类的数据，如果没有价值就都是废墟。回顾这十年，大家对数据价值层面的认知越来越强烈，数据也越来越多地应用到生活中的各个场景。

随着对数据的应用，系统会面临很多挑战。大数据提出了“4V”，具体对于开发者而言，数据体量非常大意味着系统需要高扩展性；数据种类非常丰富意味着系统需要具有高灵活性，能够很好承载随时随地产生的新数据种类；数据时效性意味着系统具有高实时性，具有数据在线化能力；数据价值则意味着需要能够商业化，系统需要降低数据的存储和计算成本。

十多年前，Google首先遇到大数据问题，因此发表了Big Table论文。而HBase则是基于该论文设计的高可靠、高性能、可伸缩的开源大数据NoSQL系统。HBase放弃了对于关系型数据库事务的支持，重点构建扩展性能力、灵活性能力、实时响应能力以及对大体量数据存储低成本的能力。



# Apache HBase

一个基于Google BigTable论文设计的高可靠、高性能、可伸缩的开源大数据NoSQL系统

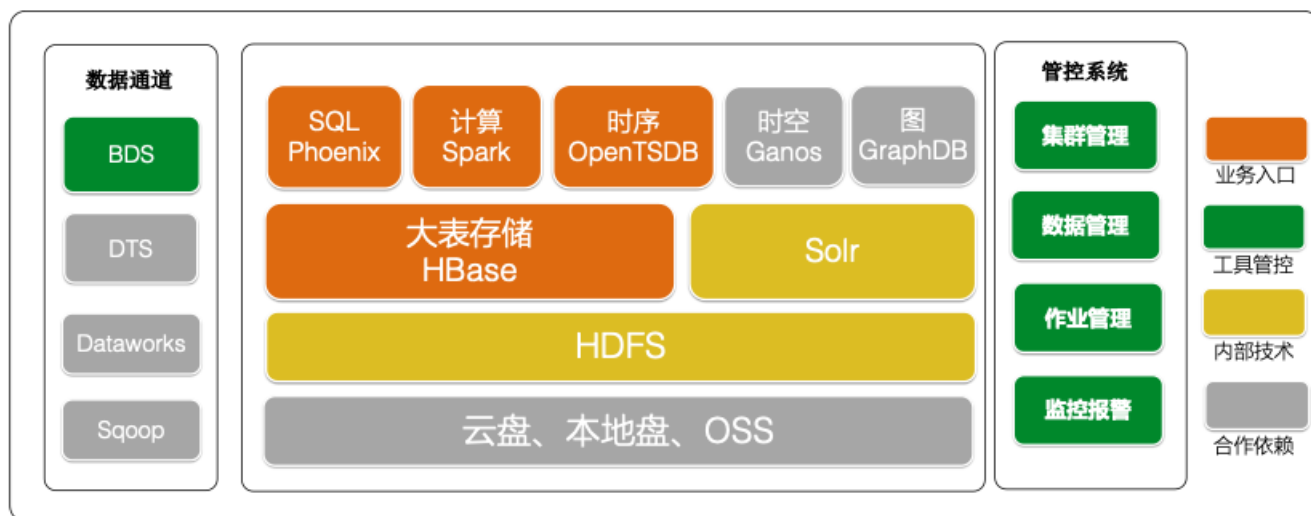


阿里巴巴从2010年开始调研HBase，如今已经走过了近十个年头。随着这十年的逐步探索，阿里巴巴也丰富了HBase的使用场景，如消息，订单，Feed流，监控，大屏，轨迹，设备状态，AI存储，推荐，搜索，BI报表等。阿里巴巴自己使用HBase已经达到了非常大的体量和规模，也在产品上有了很多积累和沉淀，形成了如今云HBase+X-Pack的架构。单独依靠HBase数据库无法解决业务场景下的复杂问题，因此X-Pack基于云HBase在计算、检索、多模型上进行了扩展，包含了Spark、Phoenix、Solr以及OpenTSDB等，形成了稳定、易用、低成本的一站式大数据NoSQL平台。



## 云HBase + X-Pack整体架构

稳定、易用、低成本的一站式大数据NoSQL平台



一站式满足企业的大数据存储与处理

X-Pack: 云HBase在计算、检索、多模型上的扩展能力集，包括Spark、Phoenix、Solr、OpenTSDB等

云HBase+X-Pack架构实现了低成本的数据存储，将HBase运行在OSS上面，并且让整体接口模型复用HDFS能力。并且同时克服了OSS在面向文件场景下的问题，将原本面向对象的存储系统当做类似云盘来使用，使得存储成本降低3到7倍。此外，还基于HBase做到了一体化冷热分离，并使得业务无感知。

除了低成本存储之外，阿里云HBase还投入了大量的精力来优化性能。相比开源版本，阿里云HBase在各个性能指标上都有较大的提升。在这背后是不断的优化，如原本将基于HDFS Pipeline日志三副本转变向LLC机制，并将串行改为并行；将原本串行获取锁的方式变为并行；并且实现了10倍的Java GC优化。

最后一点，HBase属于大数据领域，必须结合很多组件，因此易用性也是大家最为迫切需要的。阿里云HBase实现了HBase和Spark的数据联动以及在线和离线的高效融合。此外，阿里内部也提供了一套易用的数据迁移系统，能够实现平滑在线搬迁。

无论是从稳定性、易用性还是性能和成本上来说，阿里云HBase都有很大的提升。未来，阿里云HBase还会通过共享块存储等技术进一步降低成本，也将会推出Serverless能力，并且会通过新硬件来加速计算，降低成本。

## 斗鱼直播从0到1混合云架构演进

斗鱼技术总监马勇为大家分享了斗鱼直播的混合云架构演进之路。

斗鱼直播成立于2014年，是以游戏赛事为主的直播平台，平台签约国内Top100主播约50位，覆盖游戏主播Top10中8位，月活达1.5亿，2019年Q1付费用户约600万。斗鱼主要有三条业务特点：头部主播热点效应，流量水位波动较大，以及在线互动场景较多。目前的技术现状是每天业务调用量在千亿左右，Redis实例集群达2000以上，单个接口QPS达20万以上。

斗鱼直播从2016年开始保持每年25%以上的月活增长，目前面对的技术困境主要有三点：(1)“炸鱼”，头部流量拖死全站房间；(2)服务器资源利用率低，日常水位大量服务器闲置；(3)Redis维护和容灾成本高。

斗鱼混合云架构历程主要分为三个阶段，在探索期做了独立业务上云的尝试；在成长期通过IDC+云的方式实现了横向流量扩容；在成熟期完成横向扩容之后实现对资源的最大化利用。



探索期的主要背景是IDC硬件资源呈现为长期紧缺状态，研发支撑跟不上业务发展，而公有云逐步成熟。因此在这一阶段，斗鱼尝试性选取了广告业务作为上云试点，并且取得了较大收益，系统的吞吐量直线上升，依赖稳定性显著提升，计算成本也大幅下降。但是这一模式的适用范围较窄，无法直接复制到其他业务场景，而且这一模式只适用于单一数据中心的情况，于是就进入了成长期。

成长期的背景是需要解决IDC到公有云的数据通道构建问题。针对这一问题，斗鱼和阿里云共同构建了RedisShake数据同步工具，支持了Redis全量、增量数据同步、支持云上、云下不同数据中心的同步，还支持秒级数据监控。通过RedisFullCheck实现了数据多维度对比，基本能保证数据通路的数据一致性问题。这一阶段的收益在于实现了单一机房到多机房的数据扩展过程。这个阶段存在存在两点有待改进，即资源调度成本比较高和资源缺乏精细化运营。

成熟期的主要优化方向是职责分离和弹性伸缩，优化方案包括四个方面，即流量分级、数据冷热分离、弹性伸缩和流量调度。其中调度策略包括了手动调度、定时调度、资源消耗调度和Hook调度。

对于混合云架构而言，斗鱼也总结了三点经验：

- 充分合理评估：云上计算网络与IDC差异较大，需要结合业务实际情况进行测试，避免产生影响。
- 投入产出比：混合云架构对资源冗余存在一定要求或者带来一定负面影响。
- 延时问题：企业应通过评估业务的重要性决定是否做混合云，虽然从数据中心到云有专线，但也存在一定延时。

## Cassandra&X-Pack Spark云数据库技术剖析

阿里云智能高级技术专家曹龙(封神)为大家剖析了Cassandra与X-Pack Spark云数据库技术。

为什么选择Cassandra呢？Cassandra是一种完全没有中心的数据库，其每个节点都是主节点，如果Kill掉其中任何一个节点都不会影响集群的QPS以及延时。除了Cassandra使用的P2P-QUORUM机制之外，还有HA机制、Raft以及单内存副本+共享存储等机制，而只有Cassandra能够做到几乎没有感知时间，因此Cassandra的Slogan就是“Always Online”。



**极致可用**

	P2P-QUORUM	HA机制	Raft	单内存副本+共享存储
故障恢复机制	大部分Coordinator重试 少部分情况客户端重试 无等待超时时间 + 无log回放	客户端重试，超时时间、log回放	客户端重试，超时时间、log回放	客户端重试，超时时间、log回放
故障恢复时间	无感知时间	30s~10min (超时心跳+日志回放)	30s~10min (超时心跳+日志回放)	30s~10min (超时心跳+日志回放)
代表系统	Cassandra	MySQL、MongoDB、Redis	TIDB	HBase



Cassandra能够实现平滑扩展，一方面可以增加节点数据量，甚至扩展多个DC。另一方面在云上还可以增加内存等。平滑扩展是Cassandra的重要特性，其他数据库往往难以做到。Cassandra还可以实现全球多DC，架构师可以根据业务自由适配。

对于学习成本而言，Cassandra提供类似于SQL语句的CQL，会MySQL的DBA或者开发人员基本上一天之内就能学会Cassandra。在安全方面，Cassandra和主流数据库一样提供了完善的认证以及鉴权体系。在多语言方面，Cassandra采用了非Thrift方式，采用客户端和服务端直连方式，并且支持主流的语言，并且具有良好的性能。最后一点，就是运维简单，Cassandra整体只有一个进程，没有Proxy、HA以ZK等角色节点。



Cassandra具有很多功能，比较特别的就是其索引支持物化视图、还支持SASI全文索引，并且集成了Lucene做更强的全文索引，以及支持CDC对接流式系统。



The image shows a slide titled "Cassandra功能" (Cassandra Features) from the Alibaba Group APSARA Cloud Summit. It contains a table with 15 rows, each listing a feature category and its description. The categories include basic capabilities, NoSQL features, CQL, complex types, UDF, indexing, consistency, CAS, multi-region, sync, CDC, security, maintenance, and integrations with Spark, Kafka, ES, Solr, and RocksDB.

	功能描述
基本能力	PB级别、千万QPS、平均延迟ms、999延迟<40ms
NoSQL基本能力	schema动态列，列数支持百万列，行数无限制，TTL，ZSTD等高压压缩等
易用CQL	支持类似SQL的语法
复杂类型	支持Map、Set、List、JSON等复杂类型
UDF	自定义函数
索引	支持原生localindex、MV全局索引、SASI全文索引、集成Lucene作为更强的全文索引
可调一致性	支持强一致性、最终一致性、多机房一致性等多种一致性级别
CAS	支持行级别事务
多活	原生支持多DC、同城多活、异地多活等
同步	Datax支持Cassandra & C*多活同步 & BulkLoad (快速加载数据) 等
CDC	数据可以增量流出，对接流式系统
安全	支持 账户名密码、TLS/SSL链路加密
运维	单进程运维简单、支持审计日志等、监控与grafana&prometheus集成
与Spark、Kafka、ES、Solr集成	支持与Apache Spark、Kafka、ES、Solr集成
RocksDB	<a href="https://github.com/Instagram/cassandra">https://github.com/Instagram/cassandra</a>

Cassandra的功能和生态比较丰富，其可以和其他组件进行搭配，比如Spark、Kafka、ES、Lucene、RocksDB等。

Cassandra在宽表领域排名全球第一，即使在国内缺乏宣传的情况下排名也是靠前的。Cassandra的发展目前已经经过了十年，其将AWS的DynamoDB和Google的BigTable两者的长处融合在一起形成的。阿里巴巴也在2019年公测并发布了阿里云Cassandra数据库服务，并且对原生的Cassandra进行了多方面提升，比如实现了自动化运维、兼容DynamoDB、全链路优化性能提升100%等。

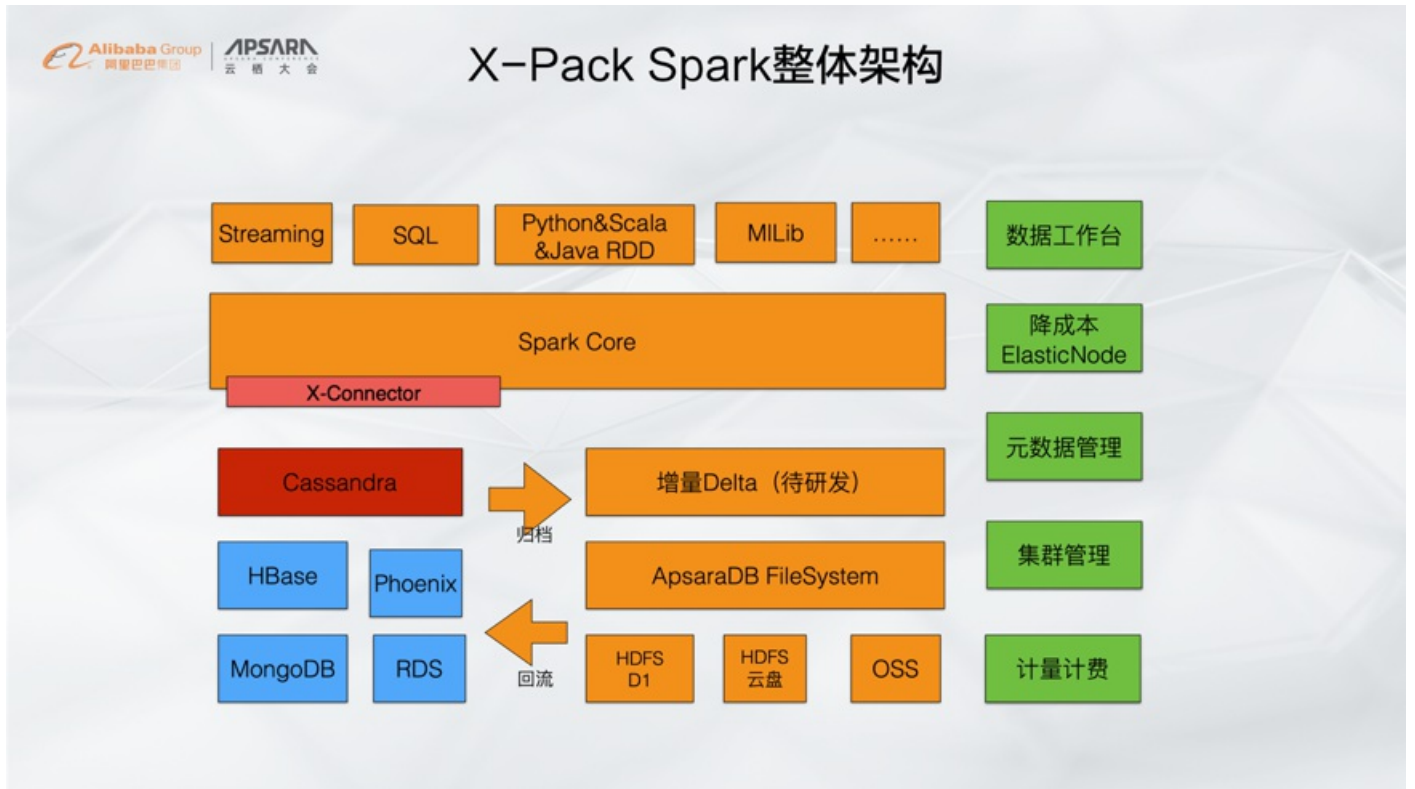
总结而言，云数据库Cassandra版是在线可靠的NoSQL可调一致性的分布式数据库服务，支持类SQL语法CQL，提供强大的分布式索引能力，提供安全、多活容灾、监控、备份恢复等企业级能力，兼容DynamoDB协议。

X-Pack Spark不仅仅支持Cassandra，还能够支持HBase、Phoenix、RDS和MongoDB。X-Pack Spark不仅具有强大的连接能力和归档能力，还能够通过ElasticNode降低计算和存储成本。

Cassandra+Spark能够应用于非常广泛的业务场景中，比如用户画像、Feed、小对象存储以及推荐平台等。



总而言之，将Spark与Cassandra的优点结合在一起能够满足多种业务场景的需求，能够实现Always Online、扩展性强、好用、功能和生态丰富以及Spark数据闭环。



[原文链接](#)

本文为云栖社区原创内容，未经允许不得转载。