非关系型数据库技术技术报告

研究背景

互联网普及、“关系型”数据库难以处理海量数据

截止2016年中，中国网民规模达突破7亿，互联网普及率51.7%，互联网已基本完成对大中型城市的覆盖开始向农村延伸。随之而来的是海量的数据，互联网行业在数据处理上遇到了挑战。

开源数据库产品MySQL在数据量级达到千万之后会显著下降，大型数据库例如Oracle在处理亿级数据处理上也要以秒、分钟为单位。而互联网行业对数据的处理需求一般以GB甚至TB为单位，并且要求较高的实时性，传统“关系型”数据库已难当重任。

分布式系统发展

随着计算机技术的发展，受限于“摩尔定律”，单机性能的提高遇到了瓶颈。业界转而寻求更廉价、易用的解决方案，能否让多台计算机联合起来求解一个大问题？由此引发了对集群、分布式系统的研究。2000年后，Google公司先后发布《Google File System》、《MapReduce》、《BitTable》三篇论文，引发了分布式计算的热潮，如今分布式系统已在产业界成为“标配”。

数据更加多样性

在早期，数据库系统多应用于大型企业与组织内部，数据来源单一，范围可控，结构固定，一般具备明显的“关系型”特征。社会信息化的发展带来了多样化的数据处理需求，比如社交系统中的聊天记录、媒体网站中的新闻、移动软件采集的地理位置信息、以及其它多媒体信息，数据逐渐变得半结构化、非结构化。“关系型”数据库提供了部分针对性的处理方式，但是需要耗费较多的空间和时间，数据处理需要跳出“关系型”的圈子寻求方案。

新型计算机存储技术普及

现代计算机以冯·诺依曼计算机体系结构为基础，主要有运算器、控制器、存储器、IO设备组成，程序以二进制码存放在存储器中。作为数据服务器，数据多数被持久化在硬盘上，传统硬盘以扇区的方式存储数据，扇区一般为512B或4KB，为了降低寻道时间数据库及其索引在设计中会尽量访问整数个扇区数据。磁盘的IO性能多年来一直制约着计算机性能的提高。SSD技术放弃了磁盘方式采用闪存芯片，可以适应高密集度的读写需求，SSD的普及也让非关系型数据库得到了更好的应用。

研究方向

文章以一个互联网应用用户平台为案例，围绕如下x个真实业务场景探讨NoSQL技术的选型、应用、性能、前景问题。

- 存储亿级用户，支持5000万DAU，用户相互关注关系管理。
- 用户发表状态、评论、回复评论的管理
- 亿级用户认证与亿级终端采集数据数据存储与查询
- 十亿日PV的广告接入
- 用户推荐系统
- 举办一次用户邀请注册活动，根据邀请人数计算排名

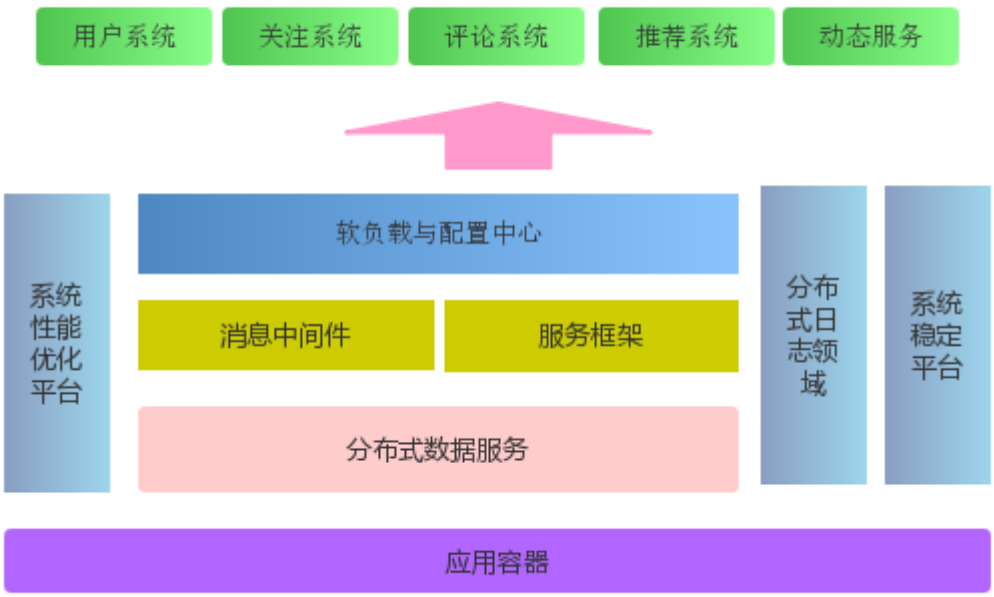
先验技术

先验知识表明NoSQL一般应用于海量数据、高并发、高性能要求的场景中，相对于传统SQL门槛较高，需要有一定的技术基础。

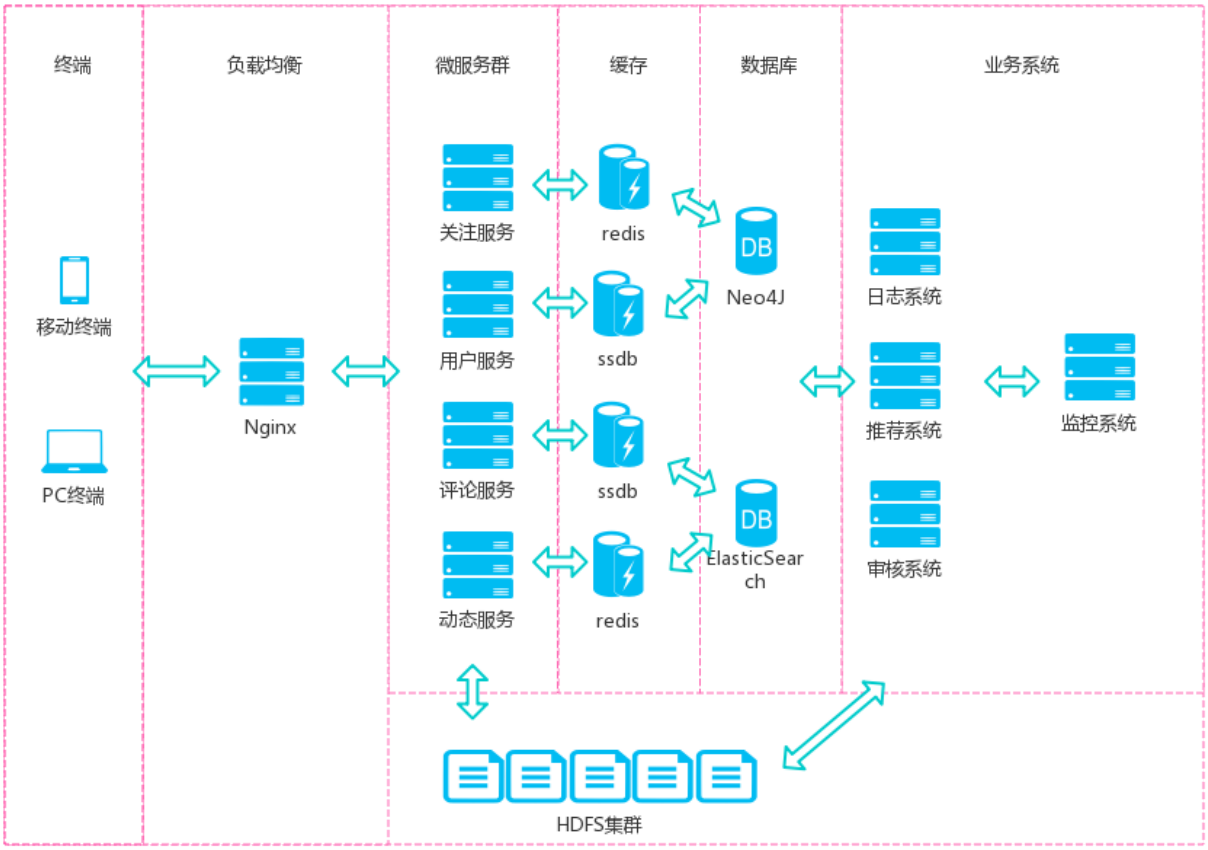
- 熟悉关系型数据库常用操作，理解ACID特性和事务隔离级别
- 熟悉分布式系统设计，理解CPA理论
- 掌握常用数据结构
- 了解PV、VV等常用性能参数
- 知道消息队列、异步通信的概念

设计

功能设计



架构设计



服务拆分

关注服务

存储、计算用户关注关系，使用Neo4J做持久化存储，Redis做查询缓存。

用户服务

存储用户基础信息，与关注系统共用Neo4J做持久化数据库，使用ssdb做查询缓存。

评论服务

存储、维护用户发布信息，使用ES充当持久化存储和检索引擎，使用ssdb做查询缓存。

动态服务

接收、存储、发布用户动态信息，使用ES做持久化存储，使用redis做查询缓存。

日志系统

收集各主机日志统一处理，由消息队列灌入ES。

推荐系统

统计用户访问信息，通过推荐算法为用户推荐好友、信息等。

审核系统

审核用户基础和发布信息是否违规。

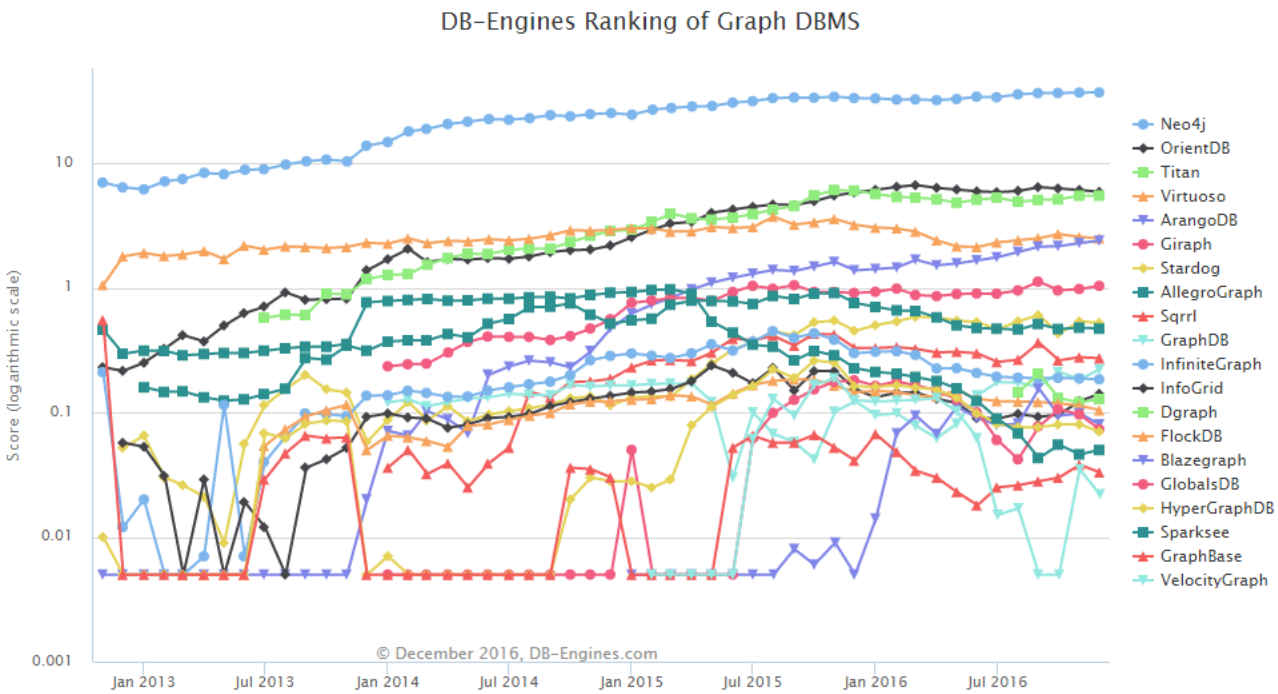
监控系统

监控各服务健康状态和调用量等信息。

NoSQL技术选型

上述方案在选型中利用多种NoSQL技术，其中

- Redis:主要充当查询缓存，内存存储，高性能，支持KV、Set、SortedSet、Hash、List等多种数据类型，支持Master/Slave模式，支持Cluster，支持RDB、AOF两种方式的持久化。Redis已在互联网行业广泛应用，官方着力挖掘其在消息中间件方面的潜力。
- SSDB:360公司基于Google的Leveldb开发的一款类KV数据库。与Redis相比其支持数据类型较少比如不支持Set类型，但降低了Redis对内存的依赖，使用硬盘存储，在SSD环境下单机可达到8w读或4w写的QPS。同时SSDB目前在集群方面还不成熟，一般使用单机或者主从模式。
- Neo4J:图数据库中知名度较高的一款，使用图概念来描述数据模型。以节点、关系以及其属性来保存数据。对于社交网络、知识网络等以图为主要模型结构的数据的处理有其天然的优势。Neo4J开源社区版，不支持集群。



- ElasticSearch:严格来说这并不是一款NoSQL数据库，ES是一个基于Lucence的全文搜索引擎，提供RESTful web API。ES使用Json的格式分布式存储数据，通过构建索引实现大数据处理和实时检索，在处理以文本为主的半结构化数据方便具有优势。
- HBase:一般运行于HDFS之上的分布式存储列式数据库，支持MR。使用行键、列簇、列的方式存储数据，支持横向扩展。在原生HDFS结构中，文件一般被分为64MB的块存储，这样在处理大量小文件的时候性能会急剧下降，HBase中没有数据类型的概念，全部数据都以字节码的形式存放在Hfile的大文件中，不存在存储碎片的问题。

NoSQL技术应用

查询缓存

对于目前的常规互联网技术架构来讲，MySQL等关系型DB往往是性能木桶中的短板。基于82原则一个系统中读请求可以占到80%，实际中譬如12306的优化过程中统计结论查询请求在90%以上，一个成熟的解决方案既是应用NoSQL技术设置查询缓存，减少请求直接访问DB的可能，降低DB压力。以Redis的应用为例，查询缓存的设计有几种方式。

1、冷热分离，写的时候双写，读的时候先读缓存，未命中再查询数据库，同时回原缓存。利用NoSQL数据库的数据过期与淘汰机制实现热数据主要存储在缓存中，冷数据持续沉淀到DB。

2、高并发缓冲，每次写的时候使用getset方法取出原缓存数据，如果原缓存未命中或数据更新时间早于一个阈值则写入DB。有效降低访问DB的访问量。

3、定时更新缓存，采用定时策略每隔一个时间段T从数据库取出一批数据放入缓存，达到大部分请求都可命中缓存的目的，适用于读写比值特别大的场景。

实时计算

MySQL等关系型数据库基于SQL语言提供了基本的数据计算功能，但是一般计算函数都需要耗费较大的性能在高并发场景往往被禁止使用。NoSQL尤其是kv数据库在存储时既是按照key的顺序排列，因此计算更加高效。以一个实时计算千万级用户积分排名的场景为例，如果使用关系型数据库，每个用户的积分变动都需要重置整张表的排名数据。在一个QPS50000的系统中这代表这这张表最少在1秒内执行50000次全表更新操作，是不可能实现的任务。使用ssdb中的SortedSet可以很容易实现，每次修改值后使用zrrank key member来获取当前排名(从大到小)。另一个场景需要记录当前用户的id地址归属地，ipv4地址使用32位二进制数组成，在Mysql中会构建省市与ip段的对应关系表，每次请求使用between查询。between查询在索引的优化下可以实现较高的性能然而在大并发量的情况下还是要尽量避免。将城市ID和起始地址IP（十进制long型）值作为key，value对存入Sorted Set，然后使用ZREVRANGEBYSCORE命令取出0-当前值中最大的那条记录既是匹配的城市ID。

分布式系统中的应用

在分布式系统中多节点的数据共享与同步机制处理起来比较繁琐。NoSQL数据库产品提供了原子操作和队列机制可以在一定程度上解决这个问题。比如，上面的定时更新查询缓存，需要使用定时任务每天更新数据到缓存中，如果没有分布式锁的机制会导致N个节点各自执行一次浪费性能。传统使用DB标识的方法因DB的事物隔离级别问题仍然不够安全，使用NoSQL中提供的getset、incr、setnx等原子操作设置竞争锁，一次更新后延迟释放，可以保证只执行一次更新。

Redis和SSDB两个kv数据库中同样提供了List数据结构，可以实现分布式的队列/栈，也可以作为分布式系统消息通信的一个选择。

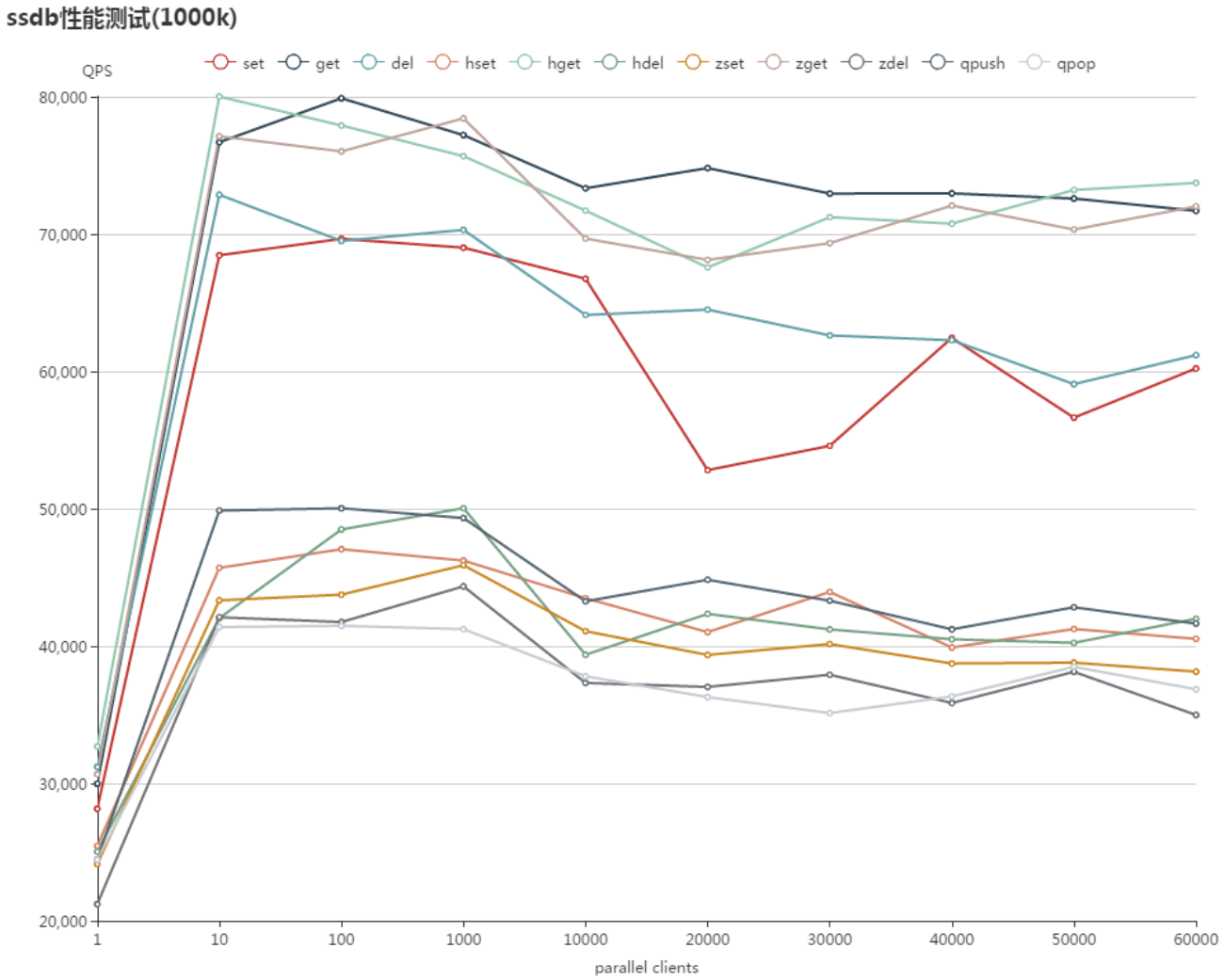
海量数据处理

在关系型数据库中，数据基于表模型和表关系处理，常规做法是对数据进行分片来分库、分表用了实现横向扩展。然而切分的粒度越细，给传统事务处理带来的复杂度也越高。采用NoSQL技术的数据库产品譬如Hbase，基于HDFS天然举报分布式存储的优势，在HBase中数据以列的方式存储，通过指定业务相关的行键值使相关数据聚集在一起提高访问效率。HBase支持多种filter查询方式，但是基于性能要求一般还是以行或者列查询为主。

NoSQL技术中的性能指标

系统吞吐量

如图是ssdb数据库主从模式在1000k量级下的并发与QPS关系图，测试过程中128G内存占用率不到5%，32核CPU占用率最高400%



事务

Redis和SSDB都支持批量操作，但是批量操作只是一个pipeline，对于一批数据中的事务仍然不能保证。而且在一般分布式系统中基于CAP理论，事务强一致性往往是被放弃的一环，一般采用补偿机制来实现最终一致性。

容量

Redis基于内存存储容量有限，SSDB容量理论上可以达到硬盘TB级别，HBase的容量可以随HDFS横向扩展达到PB级别。综合比较相对于关系数据库的亿级、千万级也是遥遥领先。

数据容错/安全灾备

在Redis的两种模式中主从模式可以实现数据同步复制和容错，集群模式中数据会被sharding到不同的节点导致数据丢失。HBase由于基于HDFS具备很高的容错性。SSDB在容错和灾备方便表现不佳，其数据不可二次使用，只可以使用官方提供的方式热备份数据，导致节点失效后即使数据文件仍然存在但是无法读取使用。另外热备份和一些性能消耗较大的命令在高负荷运转的生产环境中很容易导致宕机。

问题和展望

NoSQL技术仍然存在较大的发展空间

基于本文谈到的NoSQL技术产品可以看到，NoSQL技术发展方向较多，一般功能和优势比较单一，需要组合使用。各项NoSQL技术在事务控制方面仍然不如传统数据库，在传统OLTP这类对事务要求严格的系统中难以用作核心数据存储。

随着互联网的发展图数据库迎来了新的发展机会，但是其自身按照整张大图存储的特性导致在分布式领域难以推广。

结语

报告编写期间适逢多个需求上线，难以有精力做详尽调查研究。NoSQL技术发展多年，最新随着互联网的兴起走进大众视野，NoSQL技术涉及面较广，每个团队在NoSQL的探索上都会走出一条自己的路，总之在产业界，业务需求决定了技术研究的方向，业务发展决定了技术研究的深度。最后感谢小组同学的帮助，感谢学校和老师的指导，感谢各位同行朋友的分享和启发。