# 缓存技术选型

## 缓存常见使用场景

缓存技术作为系统性能提升的重要技术，在银行系统有着普遍的使用场景，极大的提高了交易性能和吞吐量，有效的降低了系统核心数据组件、系统整体的压力和交易链路延时，对各类应用场景起到显著加速的效果，改善客户的使用体验。

1）缓存平台作为分布式技术平台的重要技术组件，也是企业级IT架构中的一块重要拼图，在PaaS层的基础设施中承担着重要的使命。

2）建设分布式缓存平台，提供完善统计、监控、运维等功能，减少运维成本和误操作，提高机器的利用率，提供灵活的伸缩性，将复杂的各类缓存技术难题（如一致性、多活等）集中在平台内部解决，为各系统提供简单的使用方式和通用的解决方案，降低应用使用缓存的技术成本。

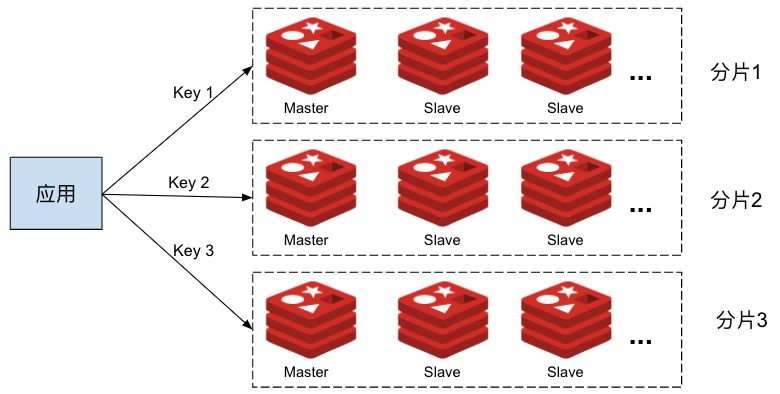
3）通过缓存平台将应用分布式能力下沉至基础设施，通过标准API向应用提供能力服务，支持开发人员以业务逻辑为核心进行组装开发，显著降低开发可扩展、可维护的分布式应用的技术门槛。

4）实现技术架构的标准化，避免重复造轮子，降低系统的复杂度，提升开发测试、部署管控、运维管理效率。

## 缓存技术主流产品

### Redis分布式缓存

1）架构图



1. Redis介绍

Redis是一个开源（BSD许可），内存存储的数据结构服务器，可用作数据库，高速缓存和消息队列代理。它支持字符串、哈希表、列表、集合、有序集合，位图，hyperloglogs等数据类型。内置复制、Lua脚本、LRU收回、事务以及不同级别磁盘持久化功能，同时通过Redis Sentinel提供高可用，通过Redis Cluster提供自动分区。

Redis是目前使用最为广泛的开源缓存中间件，基于C语言开发，性能非常高（单节点10万），占用资源少，运行稳定，功能丰富，文档和生态工具齐全，学习使用成本非常低。

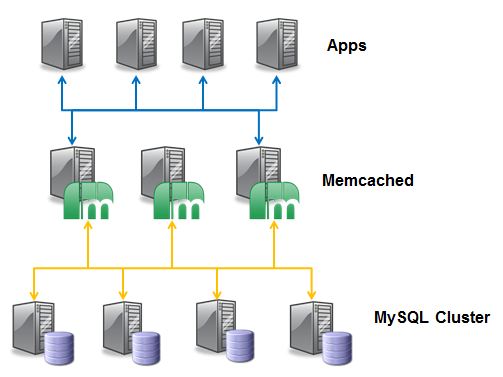
支持3种部署模式（单机，主从+sentinel，分片集群），5种常用缓存数据结构和3种扩展数据结构，8种缓存淘汰策略，2种持久化策略。近300个操作命令。

支持批量管道操作，也支持事务机制，同时对于复杂处理可以使用Lua脚本保障原子性操作。对于海量数据可以通过集群分片实现支持。

与常见的各类框架和软件一版都能很好的集成适配工作。

### Memcached缓存

1. 架构图



1. Memcached介绍

Memcached是一个开源的，高性能，分布式内存对象缓存系统。

Memcached是以LiveJournal旗下Danga Interactive公司的Brad Fitzpatric为首开发的一款软件。现在已成为mixi、hatena、Facebook、Vox、LiveJournal等众多服务中提高Web应用扩展性的重要因素。

Memcached是一种基于内存的key-value存储，用来存储小块的任意数据（字符串、对象）。这些数据可以是数据库调用、API调用或者是页面渲染的结果。

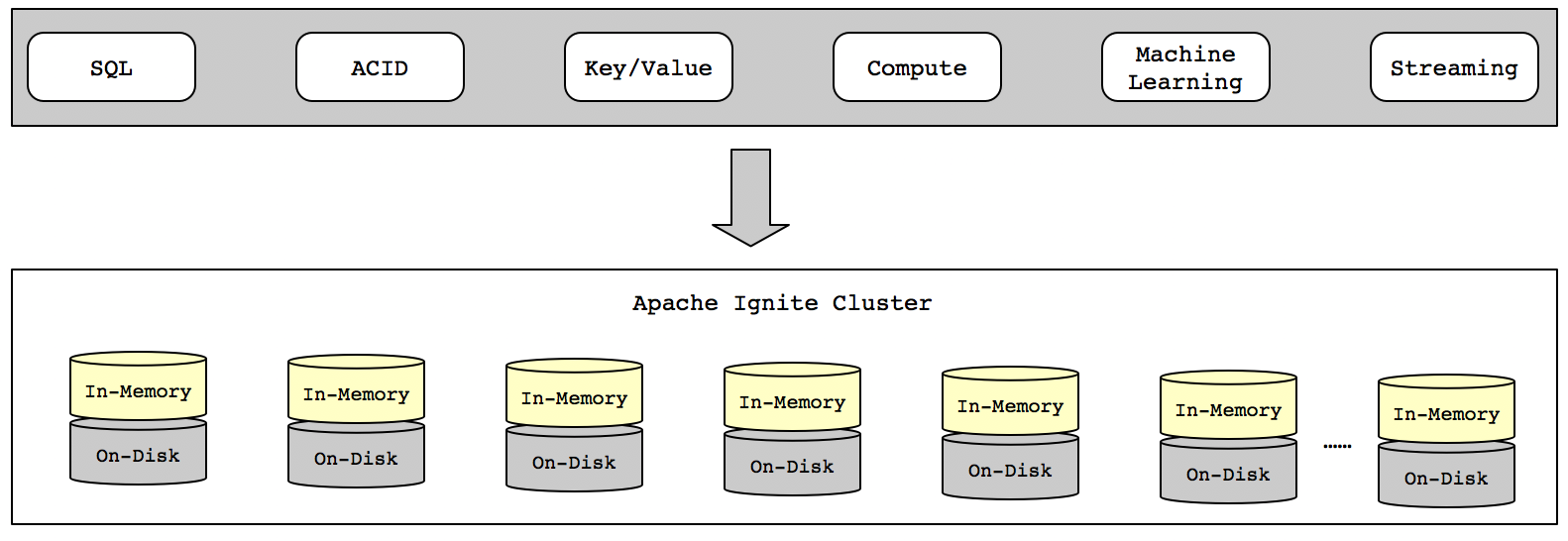
Memcached简洁而强大。它的简洁设计便于快速开发，减轻开发难度，解决了大数据量缓存的很多问题。它的API兼容大部分流行的开发语言。本质上，它是一个简洁的key-value存储系统。

与Redis不同点在于：Memcached仅支持String数据类型，且没有持久化机制。但是同时它是多线程操作缓存的。

### 内存网格技术-Apache Ignite

内存网格技术以Apache Ignite和Hazelcast这两个非常相似的技术为代表。

1. Ignite架构



1. Ignite介绍

Apache Ignite内存数据组织平台是一个高性能、集成化、混合式的企业级分布式架构解决方案，核心价值在于可以帮助我们实现分布式架构透明化，开发人员根本不知道分布式技术的存在，可以使分布式缓存、计算、存储等一系列功能嵌入应用内部，和应用的生命周期一致，大幅降低了分布式应用开发、调试、测试、部署的难度和复杂度。

Ignite是一个以内存为中心的数据平台，具有强一致性、高可用、强大的SQL、K/V以及其所对应的应用接口（API）。

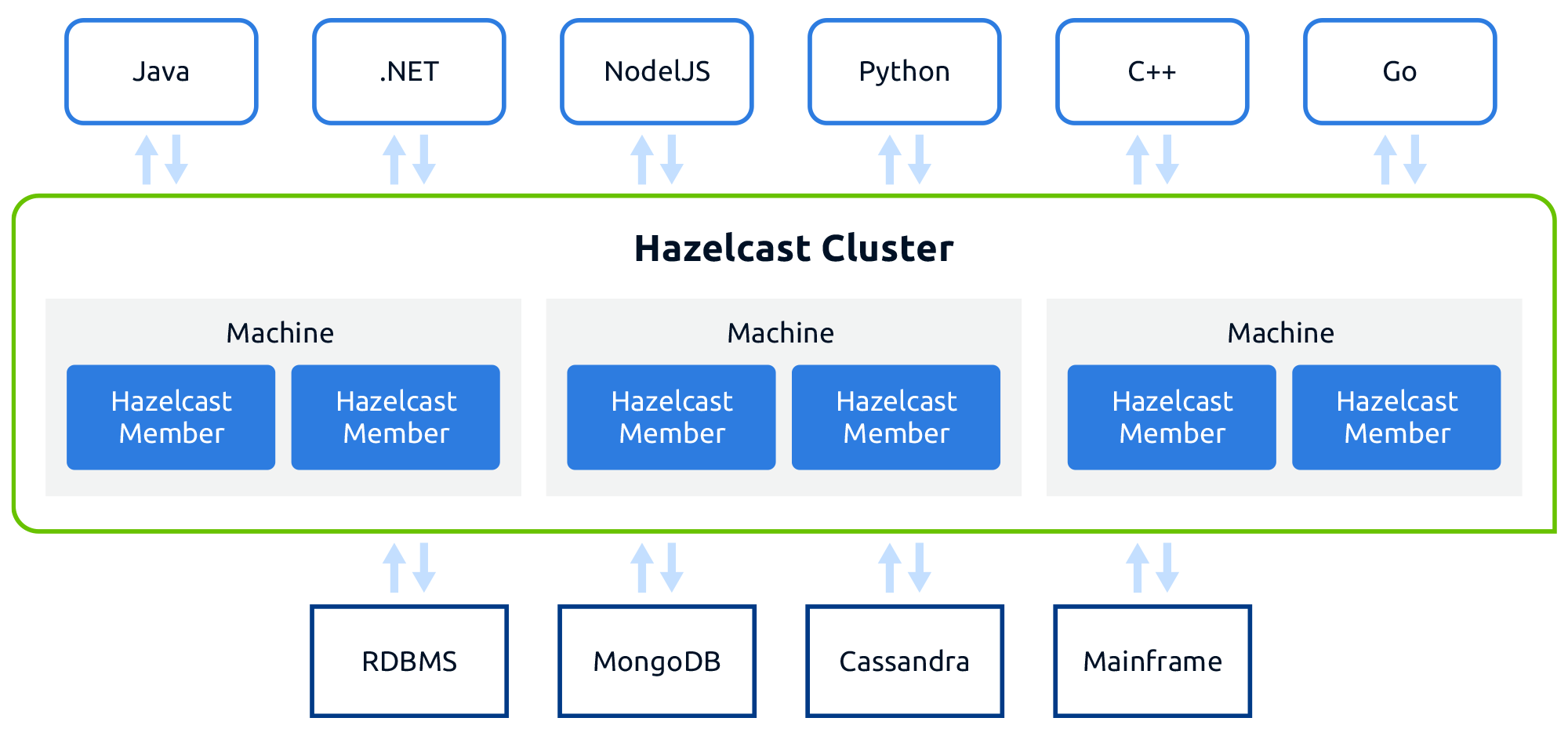
在整个Ignite集群中的多个节点中，Ignite内存中的数据模式有三种，分别是LOCAL、REPLICATED和PARTITIONED。这样增加了Ignite的扩展性，Ignite可以自动化的控制数据如何分区，使用者也可以插入自定义的方法，或是为了提供效率将部分数据并存在一起。

　　Ignite和其他关系型数据库具有相似的行为，但是在处理约束和索引方面略有不同。Ignite支持一级和二级索引，但是只有一级索引支持唯一性。在持久化方面，Ignite固化内存在内存和磁盘中都能良好的工作，但是持久化到磁盘是可以禁用的，一般将Ignite作为一个内存数据库来使用。

由于Ignite是一个全功能的数据网格，它既可以用于纯内存模式，也可以带有Ignite的原生持久化。同时，它还可以与任何第三方的数据库集成，包含RDBMS和NoSQL。比如，在和Hadoop的HDFS、Kafka等，开发基于大数据平台下的SQL引擎，来操作HDFS、Kafka这类的大数据存储介质。

### 内存网格技术-Hazelcast

1. Hazelcast架构图



1. Hazelcast介绍

Hazelcast是基于内存的开源数据网格开源项目，提供弹性可扩展的分布式内存计算，通过实现分布式的数据结构Map、Queue、ExecutorService、Lock和JCache接口使分布式计算变得更加简单。。

Hazelcast基于Java语言编写，没有任何其他依赖。Hazelcast基于熟悉的Java util包对外暴露相同的API和接口。只要将Hazelcast的jar包添加到classpath中，便可以快速使用JVM集群，并开始构建可扩展的应用程序。

Hazelcast集群中的节点是对等的，集群中没有主备角色之分，因此Hazelcast无单点故障问题。集群内所有节点存储和计算同量数据。可以把Hazelcast内嵌到已有的应用程序中或使用客户端服务器模式。

Hazelcast被设计为可以扩展到成百上千个节点，简单的增加节点，新加入的节点可以自动发现集群，集群的内存存储能力和计算能力可以维持线性增加。集群内每两个节点之间都有一条TCP连接，所有的交互都通过该TCP连接。

所有数据都存储在内存中，Hazelcast支持快速写和更新操作。

## 缓存技术选型对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Redis** | Memcached | Apache Ignite | Hazelcast |
| 开发语言/协议 | C/BSD | C/BSD | Java/Apache | Java/Apache |
| 高可用 | 主从/分片架构 | 主从架构 | 分布式架构 | 分布式架构 |
| 集群扩容 | 增加从节点  增加分片 | 增加从节点 | 增加一致性节点 | 增加一致性节点 |
| 单机吞吐量 | 高（十万级） | 高（万级） | 高（十万级） | 高（十万级） |
| 集群容量 | 1000节点/10T | 几十节点/100G | 理论不限制 | 理论不限制 |
| 持久化 | 支持 | 不支持 | 支持 | 支持 |
| 原子性操作 | 支持 | 不支持 | 支持 | 支持 |
| 事务操作 | 支持 | 不支持 | 支持 | 支持 |
| 缓存脚本支持 | 支持 | 不支持 | 支持 | 支持 |
| JVMGC影响 | 不影响 | 不影响 | 影响 | 影响 |
| 事件机制支持 | 支持 | 不支持 | 支持 | 支持 |
| 常用数据结构 | 支持 | 不支持 | 支持 | 支持 |
| 分布式数据结构 | 支持(通过Redisson) | 不支持 | 支持 | 支持 |
| 技术成熟度 | 成熟 | 成熟 | 接近成熟 | 接近成熟 |
| 社区活跃度 | 活跃 | 不活跃 | 一般 | 一般 |
| 银行案例 | 各家银行都有不同层度的使用 | 未知 | 未知，国内案例较少 | 摩根大通，国内案例不多 |

## 缓存技术选型建议

缓存技术是当前最为通用的中间件技术之一，性能，容量，稳定性，扩展性，高可用性都是衡量缓存技术的重要指标。综合以上信息考虑：

1）Redis是功能和非功能都较为完备的缓存中间件，性能高，稳定性好，使用方便，且在银行系统有非常广泛的应用案例，积累了大量的开发、测试、集成、部署、运维经验。

2）Memcached作为早期的缓存技术，功能较为单薄，目前使用得很少。不建议作为当前的缓存中间件引入。

3）Apache Ignite/Hazelcast作为新一代缓存-内存网格技术的2个典型对标实现，都是基于Java开发，功能也很强大，但是目前在国内金融行业的使用案例较少，缺乏进一步的金融级稳定性和一致性的验证。暂不建议考虑。

**综合上述分析，建议采用Redis作为本平台的缓存中间件技术。**