京东商城架构设计分享

明确设计目标

高扩展性

系统架构简单清晰,应用系统间耦合低,容易水平扩展, 业务功能增改方便快捷

高可用性 0

自动化运维。整体系统可用性99.99%,单个系统可用性99.99%。全年故障时间整个系统不超过50分钟,单个系统太超过5分钟。系统故障不超过5分钟

02

低成本

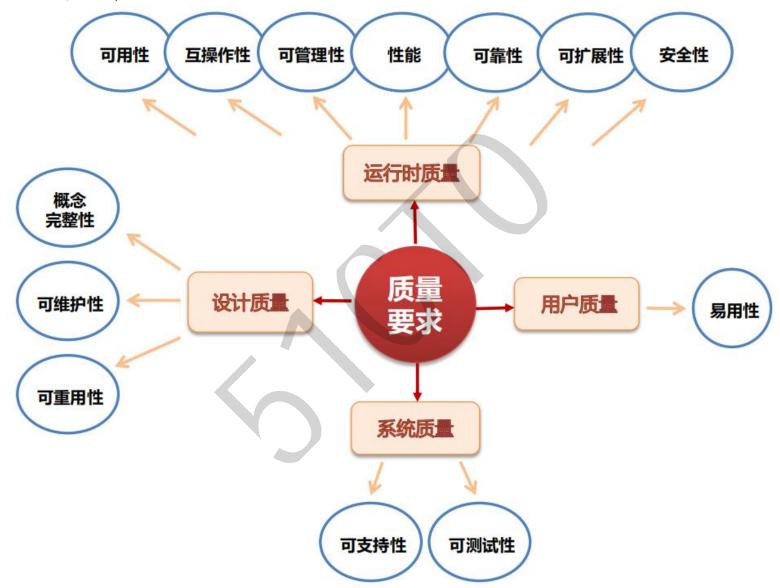
93 增加服务的重用性,提高开发效率降低人力成本;利用成熟开源技术,降低软硬件成本;利用虚拟化技术,减少服务器成本

04

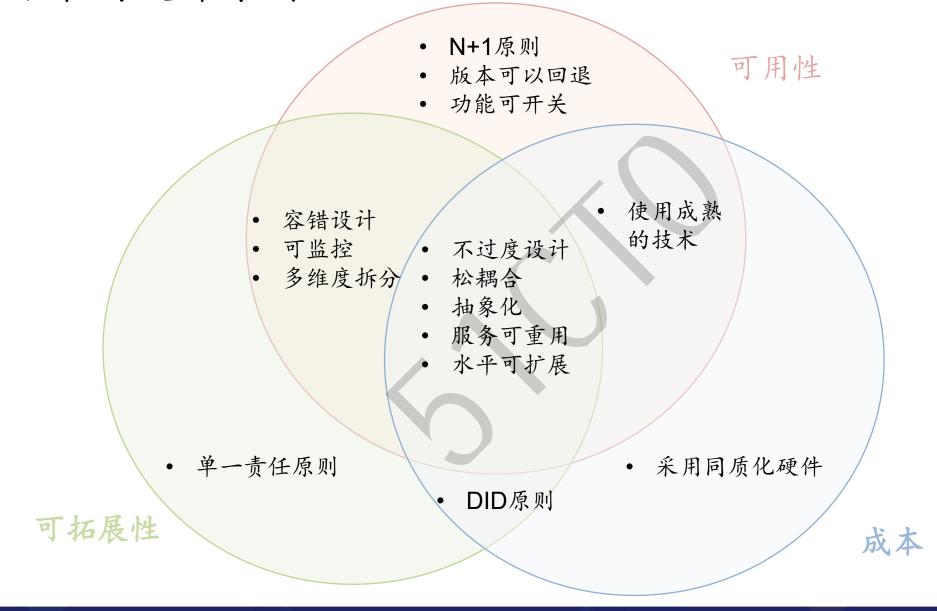
多快好省

构建超大型电商交易平台, 兼顾效率和性能, 达到高人效、高时效和低成本的目标

架构设计的13个评测维度



总体架构设计原则



四种架构观测维度

齐老师,这个部分的内容感觉有点像企业架构的内容啦,对于一个技术方向的架构师的话是否也需 要掌握这么多内容呀

如果是这样的话,感觉后面普适性的原则也有点多,因为很多原则性的东西在网络上可能也可以看 到,那这样的话我们就需要把落脚点放在京东的架构设计上,可以讲一下京东的架构设计是如何满



业务架构总览



业务架构设计原则

1. 业务平台化

- 业务平台化,相互独立。如交易平台、仓储平台、物流平台、支付平台、广告平台等
- 基础业务下沉,可复用。如用户、 商品、类目、促销、时效等



2. 核心业务、非核心业务分离

 电商核心业务与非核心业务分离, 核心业务精简(利于稳定),非核 心业务多样化。如,主交易服务、 通用交易服务

4. 区分主流程、辅流程

 分清哪些是电商的主流程。运行时, 优先保证主流程的顺利完成,辅流 程可以采用后台异步的方式。避免 辅流程的失败导致主流程的回滚。 如,下单时,同步调用快照,异步 通知台账、发票

3. 隔离不同类型的业务

- 交易业务是签订买家和卖家之间的 交易合同,需要优先保证高可用性, 让用户能快速下单
- 履约业务对可用性没有太高要求, 可以优先保证一致性
- 闪购业务对高并发要求很高,应该 跟普通业务隔离

应用架构设计原则



稳定性原则

- 一切以稳定为中心
- 架构尽可能简单、清晰
- 不过度设计



解耦/拆分

- 稳定部分与易变部分分离
- 核心业务与非核心业务分离
- 电商主流程与辅流程分离
- 应用与数据分离
- 服务与实现细节分离



抽象化

- 应用抽象化:应用只依赖服务抽象, 不依赖服务实现细节、位置
- 数据库抽象化:应用只依赖逻辑数据库,不需要关心物理库的位置和分片
- 服务器抽象化:应用虚拟化部署,不需要关心实体机配置,动态调配资源



容错设计

- 服务自治:服务能彼此独立修改、 部署、发布和管理。避免引发连 锁反应
- 集群容错:应用系统集群,避免 单点
- 多机房容灾:多机房部署,多活

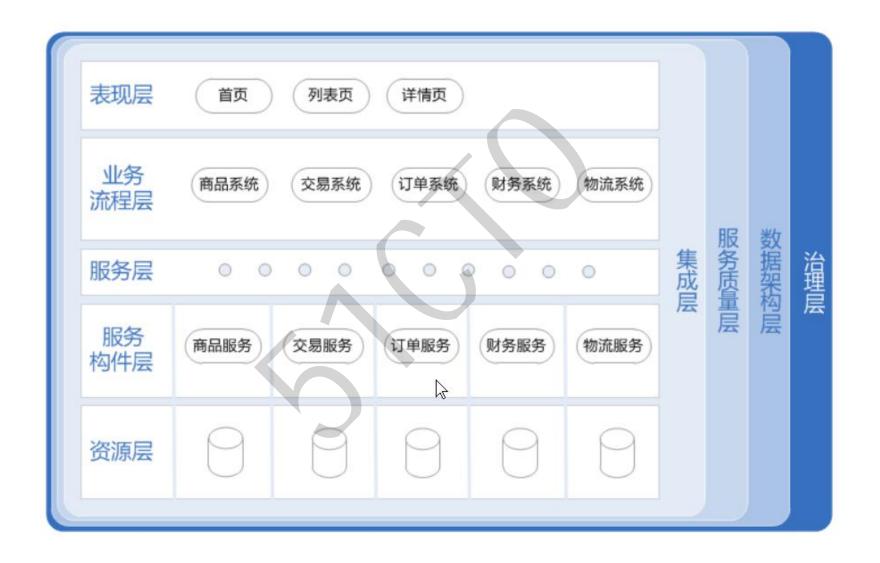




松耦合

- 跨域调用异步化:不同业务域之间 尽量异步解耦。
- 非核心业务尽量异步化:核心、非 核心业务之间,尽量异步解耦
- 必须同步调用时,需要设置超时时间和任务队列长度

应用架构的分层设计



架构在不同维度下的要求

应用系统

数据库

1. 水平扩展 (复制)

多机集群,提高并发能力

读写分离 如,商品读库,商品写库 高井发

2. 垂直拆分 (不同业务拆分)

按业务域划分系统 如 , 商品系统 交易系统

按业务分库 如,商品库,订单库

3. **业务分片** (同业务分片) 按功能特点分开部署 如,秒杀系统

分库分表,提高数据容量如,订单库按ID分库分表



4. 水平拆分 (稳定与易变分离)

服务分层 功能与非功能分开 冷热数据分离,历史数据 分离

分层间的依赖原则

2. 跨域弱依赖

跨业务域调用时,尽可能异步 弱依赖

1. 依赖稳定部分

- 稳定部分不依赖易变部分
- 易变部分可以依赖稳定部分
- 要求:避免循环依赖

6. 核心服务依赖

- 核心服务不依赖非核心服务
- 非核心服务可依赖核心服务
- 条件:核心服务稳定



5. 平台服务依赖

- 平台服务不依赖上层应用
- 上层应用可依赖平台服务
- 条件:平台服务稳定

3. 基本服务依赖

- 基本服务不能向上依赖流程服务
- 组合服务、流程服务可以向下依赖 基本服务
- 条件:基本服务稳定

4. 非功能性服务依赖

- 非功能性服务不依赖功能性服务
- 功能性服务可依赖非功能性服务
- 条件:非功能性服务稳定

数据架构设计原则

2 数据、应用分离

- 应用系统只依赖逻辑数据库
- 应用系统不直接访问其它宿主的数据库,只能通过服务访问

1 统一数据视图

• 保证数据的及时性、一致性、准确性、完整性

B

6 合理使用缓存

- 数据库有能力支撑时,尽量不要引入缓存
- 合理利用缓存做容灾

3 数据异构

- 源数据和目标数据内容相同时, 做索引异构。如商品库不同维度
- 内容不同时,做数据库异构。如 订单买家库和卖家库。

数据架构

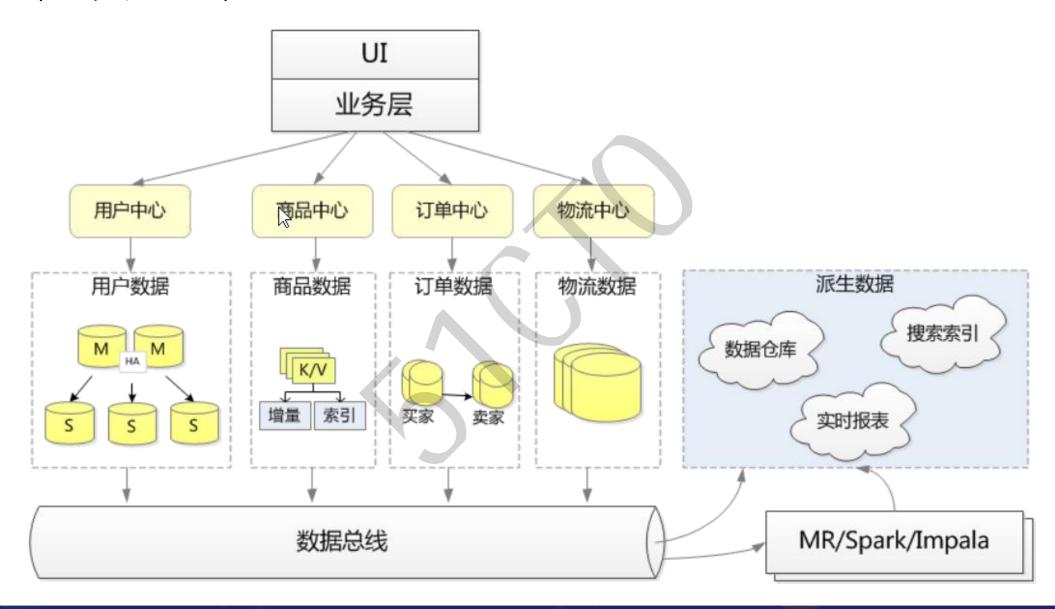
5 用Mysql数据库

 除成本因素外, Mysql的数据 库扩展性和支持高并发的能力 较强, 公司研发和运维在这方 面积累了大量经验

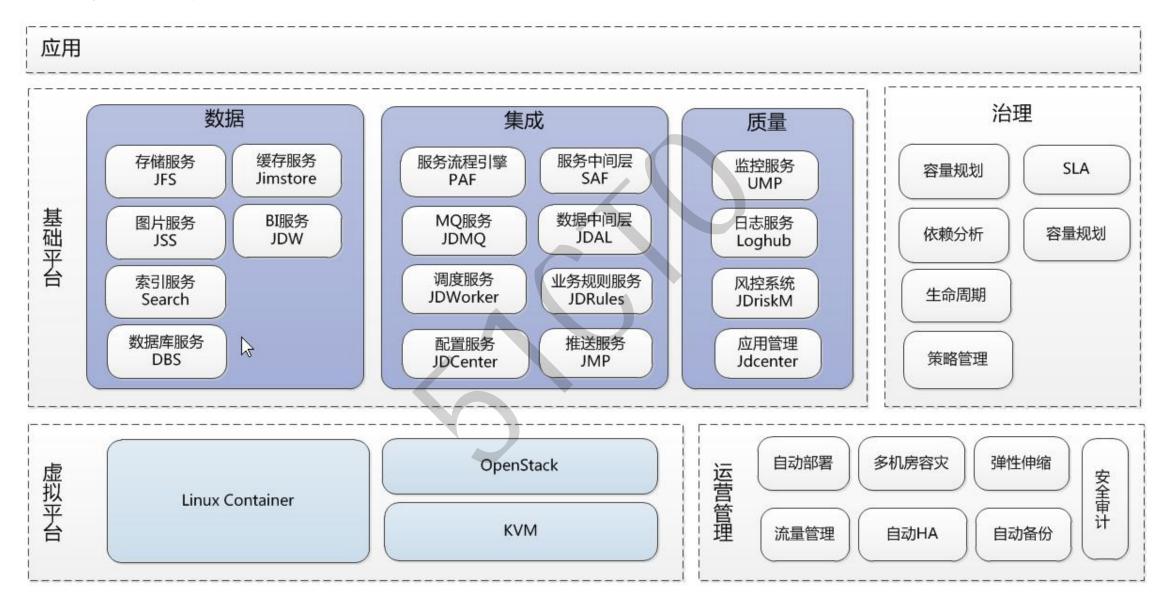
4 数据读写分离

- 访问量大的数据库做读写分离
- 数据量大的数据库做分库分表
- 不同业务域数据库做分区隔离
- 重要数据配置备库;

数据架构示意图



技术架构图



技术架构运行时原则

2、应用可回滚,功能可降级

应用出现问题时,要求能回滚到 上一版本,或做功能开关或降级

1、可监控

- 服务的TPS和RT是否符合SLA
- 是否出现超预期流量

、在线扩容

超预期流量时,应用系统可选择在线水平扩展

6、可故障转移

多机房部署,发生故障时能即时切换

4、安全保证

- 确保系统的保密性和完整性
- 具有足够的防攻击能力

5、可容错

核心应用要求多活,避免单点 设计,并且自身有容错和修复 能力。故障时间TTR小

运行时

51CTO 学堂

系统部署原则

2 D-I-D原则

- 设计20倍的容量 (Design)
- 实现 3 倍的容量 (Implement)
- 部署1.5倍的容量 (Deploy)

1 N+1原则

- 确保为故障多搭建一套系统,避免 单点问题。例如,多机房部署、应 用系统集群、数据库主备等
- 功能开发与运维分开。系统开发完成后,交给专业的运维团队管理和运营。

业务子网

 机房部署以业务域划分:基本服务 和数据库,相同业务域的服务器部 署在一起;不同业务域的服务器物 理隔离

3 支持灰度发布

• 系统新上线, 要求支持"灰度" 发布, 分步切流量, 故障回滚

系统部署

4 虚拟化部署

- 虚机部署:二级系统、三级系统 采用虚拟机部署,节省资源和管 理成本
- 虚拟化部署:一级系统应用服务器,采用虚拟化部署

谢谢!



为梦想增值!

版权声明,本文档全部内容及版权归北京无忧创想信息技术有限公司所有。

