Atitit 高性能架构法艾提拉著作

目录

[1. 前期可以立即使用的技术 2](#_Toc18808)

[2. 分离法 3](#_Toc8792)

[2.1. Web db分离 3](#_Toc1823)

[2.2. 读写分离 4](#_Toc18243)

[2.3. CDN加速技术 4](#_Toc17756)

[2.4. 动静分离文件服务器 4](#_Toc12729)

[2.5. 负载均衡 4](#_Toc4635)

[2.6. 冷热数据分离 一般当年数据访问多 4](#_Toc13368)

[2.7. NFS网络共享存储文件系统 4](#_Toc10501)

[2.8. App db file 分离 4](#_Toc2661)

[3. 更快的语言与策略 4](#_Toc29531)

[3.1. Java vs js vs sql 4](#_Toc27623)

[3.2. 更高的io、 nodejs 4](#_Toc6129)

[3.3. 最小化事务 4](#_Toc760)

[3.4. 开放缓存服务OCS， 4](#_Toc9187)

[3.5. Gzip 压缩，减少 IO  5](#_Toc19664)

[3.6. 业务前端化 尽可能前端迁移 后端只数据 5](#_Toc23021)

[3.7. Msa架构 5](#_Toc23061)

[3.8. 安全机制 抢购、促销和恶意攻击 断路器 5](#_Toc26004)

[3.9. 银行模式 金字塔模式 vs医院水平分层扩展模式 5](#_Toc22500)

[4. 更快的服务器云服务器法 云mongodb 5](#_Toc22654)

[4.1. 更快的数据库 云mongodb 5](#_Toc20034)

[4.2. Ssd 对sql提升巨大 5](#_Toc19374)

[4.3. 高性能：高端存储的I/O能力很强 5](#_Toc9970)

[4.4. 2004年 oracle emc 到IOE架构 5](#_Toc2944)

[4.5. OSS取代EMC存储实现海量数据文件的存储 5](#_Toc26617)

[4.6. 高性能服务器 大内存 高iops tb存储ssd 5](#_Toc11412)

[5. 备用高性能存储 6](#_Toc10082)

[5.1. Mq es 云mysql 6](#_Toc6413)

[5.2. MongoDB 取代 MySQL 6](#_Toc10747)

[5.3. 缓存技术 redis 6](#_Toc24783)

[5.4. 云es 6](#_Toc27941)

[5.5. 内嵌嵌套存储 反模式 6](#_Toc11424)

[6. 分布式存储 水平切分 7](#_Toc7144)

[6.1. 按照地域更加简单些 7](#_Toc29526)

[6.2. 子服务系统若按业务划分太细太多，部署工作量很大且难管理 7](#_Toc12780)

[6.3. 按业务重要性级别划分 7](#_Toc18810)

[7. 著名大网站架构演变 7](#_Toc25357)

[8. 淘宝当初架构 7](#_Toc9996)

[8.1. 2003起步LAMP架构，当时整个网站应用服务器大概10台左右， 7](#_Toc21656)

[8.2. MySQL数据库采用了读写分离、一主两备的部署方式。 7](#_Toc7237)

[8.3. 2004 Oracle+IBM小型机的数据库架构和EMC存储方式（图2） 8](#_Toc23933)

[8.4. 对于商品搜索功能，采用自己开发的ISearch搜索引擎来取代在Oracle数据库中进行搜索， 8](#_Toc30091)

[8.5. 后来 2005？？ php》java 8](#_Toc27890)

[8.6. 从2006年开始，，开始建立自己的CDN站点， 8](#_Toc4088)

[8.7. 2007年，淘宝 ，每天有100多万笔交易被创建 分布式缓存取代商业的NAS fs系统 8](#_Toc29539)

[8.8.   2008年初 业务拆分 mq 9](#_Toc27424)

[8.9. 从2010年开始 云架构 oss SLB、ECS、RDS、OSS、ONS、CDN 9](#_Toc18333)

[8.10. 合适的技术策略和最佳实践，包括：应用无状态，有效使用缓存 异步 9](#_Toc10746)

[9. 京东架构 10](#_Toc4758)

[10. 团队架构与规模 10](#_Toc1046)

[11. ref 10](#_Toc24137)

# 前期可以立即使用的技术

Cdn Nginx负载均衡 微服务 Gzip

Java Redis 缓存 ocs oss

云架构 Nfs 文件 Redis mongodb集群 es

ssd

业务 存储前段话

# 分离法

## Web db分离

按照地域 vs 按照业务来拆分成不同的服务

## 读写分离

## CDN加速技术

## 动静分离文件服务器

## 负载均衡

## 冷热数据分离 一般当年数据访问多

## NFS网络共享存储文件系统

## App db file 分离

# 更快的语言与策略

## Java vs js vs sql

## 更高的io、 nodejs

## 最小化事务

## 开放缓存服务OCS，

将部分查询数据加载至分布式缓存中，减少RDS的数据查询次数，提升系统的数据查询并发效率和降低响应时间，如图7所示

增加本地的 Cache

## Gzip 压缩，减少 IO

## 业务前端化 尽可能前端迁移 后端只数据

## Msa架构

## 安全机制 抢购、促销和恶意攻击 断路器

## 银行模式 金字塔模式 vs医院水平分层扩展模式

# 更快的服务器云服务器法 云mongodb

## 更快的数据库 云mongodb

## Ssd 对sql提升巨大

## 高性能：高端存储的I/O能力很强

## 2004年 oracle emc 到IOE架构

## OSS取代EMC存储实现海量数据文件的存储

，OSS存储最大容量可以达40PB，同时由于OSS是分布式存储方式，可以通过多个节点的并行读写显著提高数据访问性能。对于大文件，还可以通过Multipart Upload的方式，将大文件分块并行传输与存储，实现高性能。

## 高性能服务器 大内存 高iops tb存储ssd

目前阿里云的RDS实例最高是48GB内存，14000IOPS，1TB的存储容量（SSD存储），支持MySQL和SQL Server。这个配置作为单数据库服务器来使用可以满足很多场景的数据库应用需求，可直接取代大部分场景下的IBM小型机+Oracle数据库+EMC存储

# 备用高性能存储

## Mq es 云mysql

## MongoDB 取代 MySQL

## 缓存技术 redis

## 云es

## 内嵌嵌套存储 反模式

内嵌模型可以给应用程序提供很好的数据查询性能，因为基于内嵌模型，可以通过一次数据库操作得到所有相关的数据。同时，内嵌模型可以使数据更新操作变成一个原子写操作。然而，内嵌模型也可能引入一些问题，比如说文档会越来越大，这样就可能会影响数据库写操作的性能，还可能会产生数据碎片（data fragmentation）。

# 分布式存储 水平切分

## 按照地域更加简单些

## 子服务系统若按业务划分太细太多，部署工作量很大且难管理

## 按业务重要性级别划分

不是按业务功能分区服务的，而是按业务重要性级别划分了 0、1、2 三个级别不同的子业务服务系统

# 著名大网站架构演变

# 淘宝当初架构

## 2003起步LAMP架构，当时整个网站应用服务器大概10台左右，

## MySQL数据库采用了读写分离、一主两备的部署方式。

**在之后的20天里就完成了10000名注册用户**

**2003年全年完成3400万元交易额，商品数达到了80万件，注册会员超23万人。**

## 2004 Oracle+IBM小型机的数据库架构和EMC存储方式（图2）

。虽然方案成本昂贵，但性能非常好

## 对于商品搜索功能，采用自己开发的ISearch搜索引擎来取代在Oracle数据库中进行搜索，

降低数据库服务器的压力。做法比较简单，每天晚上全量将Oracle小型机的数据dump出来，Build成ISearch的索引，当时商品量也不大，一台普通配置的服务器，基本上可以将所有的索引都放进去，没做切分，直接做了一个对等集群。

## 后来 2005？？ php》java

**2005件，商品数达到了700万件，成为亚洲最大的网络购物平台。注册会员数超过1000万。**

## 从2006年开始，，开始建立自己的CDN站点，

由于淘宝的主要流量来源于各种商品图片、商品描述等静态数据，

## 2007年，淘宝 ，每天有100多万笔交易被创建 分布式缓存取代商业的NAS fs系统

取代商业的NAS存储设备来存储淘宝的各种文件信息，如商品图片、商品描述信息、交易快照信息，来达到降低成本和提高整体系统的容量和性能的目的，同时可以实现更灵活的扩展性。第一期上线大概200台TFS服务器。另外，将ISearch搜索引擎改为分布式架构，支持水平扩展，部署了48个节点。图3展示了这一架构思路。

## 2008年初 业务拆分 mq

   2008年初，为了解决Oracle数据库集中式架构的瓶颈问题（连接数限制、I/O性能），将系统进行了拆分，按照用户域、商品域、交易域、店铺域等业务领域进行拆分，建立了20多个业务中心，如商品中心、用户中心、交易中心等。所有有用户访问需求的系统，必须使用业务中心提供的远程接口来访问，不能够直接访问底层的MySQL数据库，通过HSF这种远程通信方式来调用业务中心的服务接口，业务系统之间则通过Notify消息中间件异步方式完成调用。图4是淘宝的分布式架构图。

## 从2010年开始 云架构 oss SLB、ECS、RDS、OSS、ONS、CDN

## 合适的技术策略和最佳实践，包括：应用无状态，有效使用缓存 异步

（浏览器缓存、反向代理缓存、页面缓存、局部页面缓存、对象缓存和读写分离），服务原子化，数据库分割，异步解决性能问题，最小化事物单元，适当放弃一致性。以及自动化监控/运维手段包括监控预警、配置统一管理，基础服务器监控，URL监控，网络监控，模块间调用监控，智能分析监控，综合故障管理平台，容量管理。可以很好地解决以上问题，从而达到整体系统的高可扩展性、更低的成本、更高的性能和可用性的实现效果

# 京东架构

# 团队架构与规模

梯队化建设，，架构师（包括部署运维，保证非功能） ，测试压力，开发功能为主，

实际上代码规模膨胀的很快。 与代码一块膨胀的还有团队，从最初的 4 个人到近 30 人

# ref

大型网站架构演变过程\_网站迁移\_网站构架\_it构架-阿里云.html

每秒1500万并发计算背后高性能、高可用实时搜索系统的架构演变 - 51CTO.COM.html

(9+条消息)高性能、高可用平台架构演变史 - weixin\_34417814的博客 - CSDN博客.html

(9+条消息)京东架构专家分享京东架构之路 - lsxf\_xin的专栏 - CSDN博客.html