针对大型网站的架构

架构体系——多系统间的通信问题，分布式协作

系统基础设施——保证系统基本运行

系统安全问题——防止非法操作

系统稳定性——保证系统正常运行

数据分析——提高业务质量

总结：希望业务能够安全、正常地发展，同时根据反馈改善自身业务，使其更好地发展，建立这样一个良性循环。

分类：

系统基础设施

架构体系

系统稳定性

系统安全问题

数据分析

架构体系

RPC

实现

TCP(Socket+ObjectStream字节流传输)

HTTP(web容器端口+string字符串传输，实质也是字节流传输)

具体实现

HTTPClient，servlet，springMVC

优势：容易理解，通用性强，线程问题开箱即用

劣势：协议上层——效率低

格式标准：

JSON jackson-all

XML：Extensible Markup Language xtream-1.4.4.jar

URL风格：

RESTFul /xx/provider/sayHelloService/xxaaaaaa.json

RPC xx.do?service=xx.yy.zz&method=fff

包含概念

HTTP服务网关 gateway

路由

负载均衡

分类

硬件：F5

软件：LVS（传输层），Nginx（应用层）

算法

轮询法，随机法，原地址hash法，加权轮询，加权随机，最小连接

动态配置

脚本语言——Groovy

Zookeeper

集群：leader election

结点：persistent，ephemeral

watcher：触发一次

facade：

zkClient

问题：

单点

解决

服务单机->双机 一台stand by，Zk注册服务地址

序列化

传输

时延性

A同B异

A同B同

A异B同

A异B异

系统基础设施

缓存 解决传统数据库磁盘低吞吐

memcache

实现：libevent

算法：

LRU

api：

java：Memcahed-Java-Client

特点：

非分布式->自行hash实现->导致雪崩效应->改善：consistent hash

分布式session

coockie 大小受限，安全性差

持久化到DB

持久化到缓存集群

实现：

tomcat实现：memcached-session-manager，Sticky和Non-Sticky模式

数据库

需求理论

CAP

ACID

类型：

传统：

MySQL(复杂查询、事务)

扩展：

业务拆分：DB->MultiDB

复制策略：Master-Slave，Dual-Master

分表:Order->Order\_1..Order\_n 解决量大查询，但不解决写

分表分库：db.Order->db\_0.Order\_1,db\_n.Order\_y

海量DB:

HBase（难支持复杂查询，但写性能高，易扩展）

概念:

原型Google BigTable,运行于HDFS

集群角色：HMaster和HRegionServer

行、列族、列、版本（时间戳）

迭代行

rowkey->

rowkey's range []

全表扫描 开销大

优化：

rowkey按商品id\_时间生成

二级索引表，存在性能、一致性问题

Key—Value

Redis:(多种存储结构，集群，吞吐量，并发数，支持MQ)

api：

java：Jedis

消息

JMS规范

消息类型：

发送类型：

实现

ActiveMQ:

搜索引擎

其它

1. a
2. fsdfds
3. sdfsfs
   1. dsfsdf
   2. fsdf