

淘宝网技术架构介绍

目标

了解淘宝架构的需求

了解淘宝的技术演变

了解架构的一些基本准则

淘宝@海尔

淘宝相关的一些数据

每天

- 40亿次的页面访问，其中搜索宝贝过亿次，浏览宝贝过亿次
- 超过7亿次的用户访问，超过6亿此的交易访问，超过6亿此的宝贝访问
- 超过400万笔有效交易

高峰期

- 每秒超过25G的流量，核心业务每秒超过4.5G的流量
- 每秒生成几百笔交易，8万次的用户访问，1.5万次的商品访问

淘宝总共

- 超过22TB的宝贝图片存储
- 超过4亿条在线交易记录，超过2亿的在线宝贝

非功能性需求

- 高容量
- 高性能
- 高稳定性
- 高可维护性

淘宝架构的版本演变

V1.0

2003.05-2004-01

- LAMP
- MySQL读写分离

V1.1

2004.01-2004.05

- MySQL->Oracle

V2.0

2004.02-2005.08

- PHP->JAVA
- 3层架构
- 通用组件

V2.1

2004.10-2007.01

- Weblogic->Jboss
- 支持数据库分库
- 抛弃EJB, 引入spring
- Session框架重构
- 淘宝自己的CDN

V2.2

2006.6-2007.12

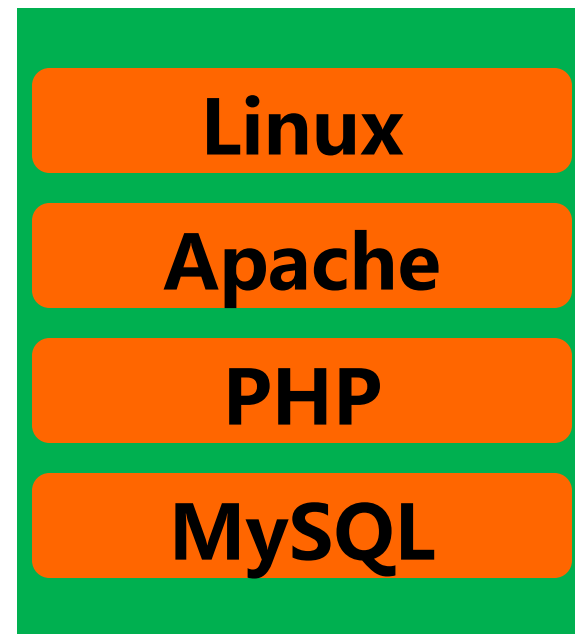
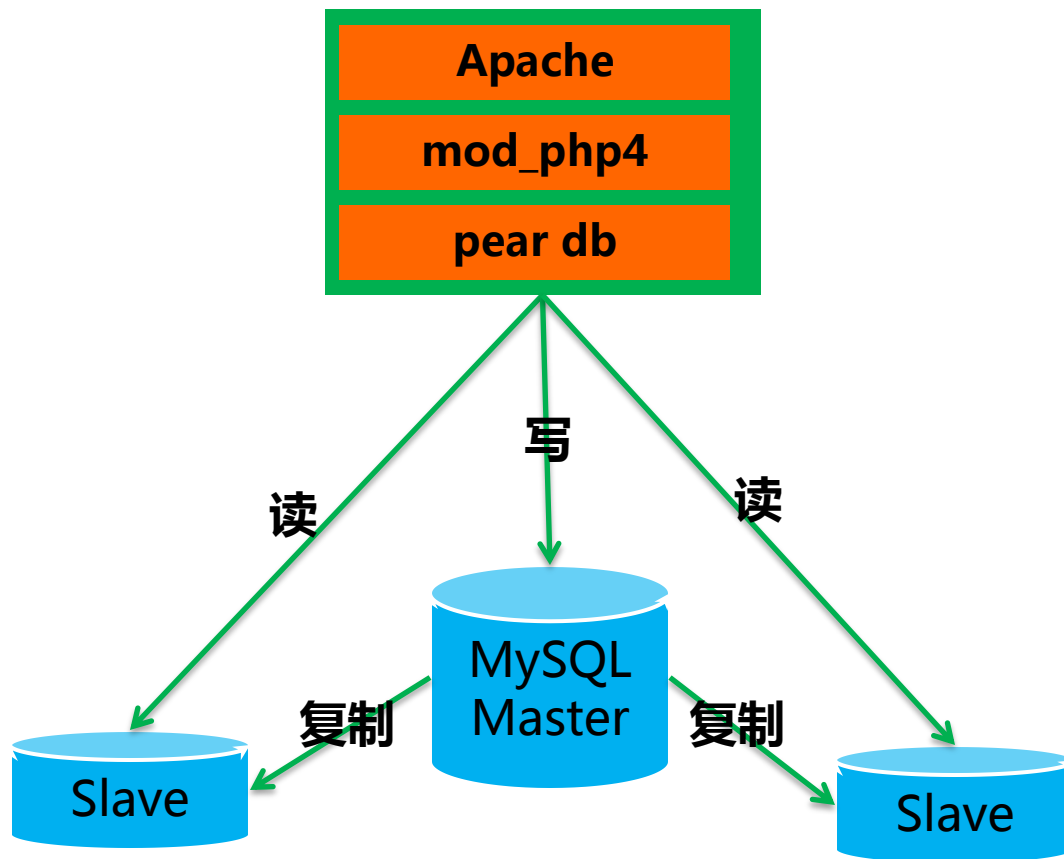
- 分布式存储TFS
- 分布式缓存Tair
- 页面缓存ESI
- 搜索引擎升级

V3.0

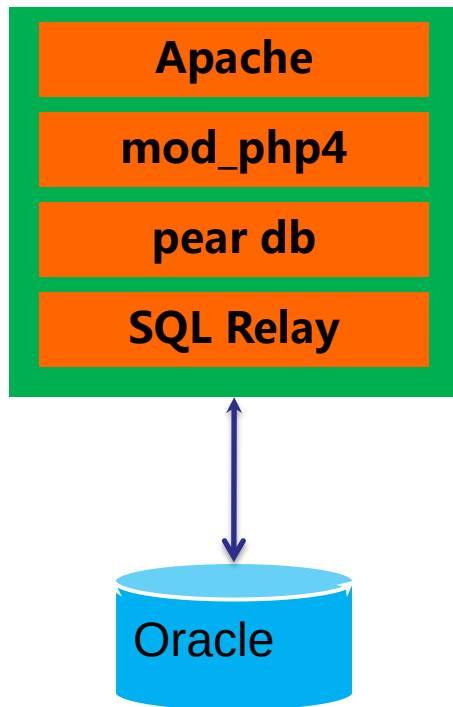
2007.12- ?

- 数据访问层TDDL
- 面向服务框架HSF
- 消息系统Notify
- Oracle->mysql

V1.0架构2003.05-2004.01



- 数据库容量限制
- 数据稳定性
- 数据库性能问题

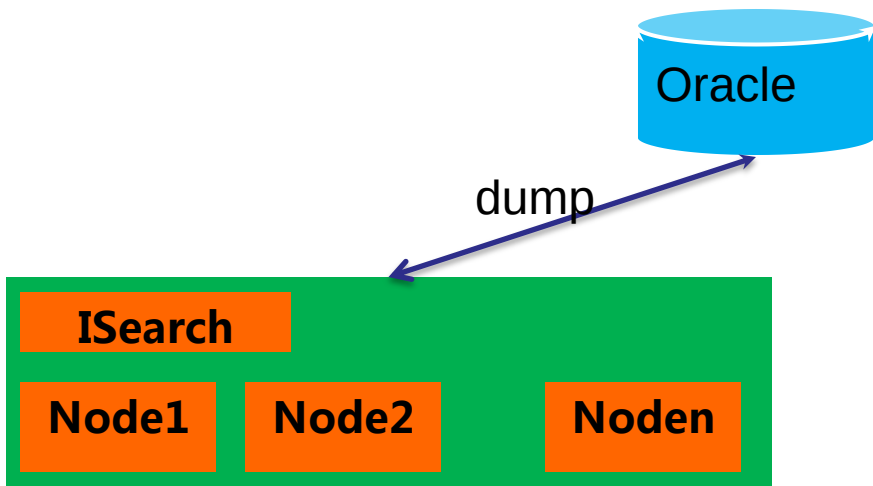


•开发效率

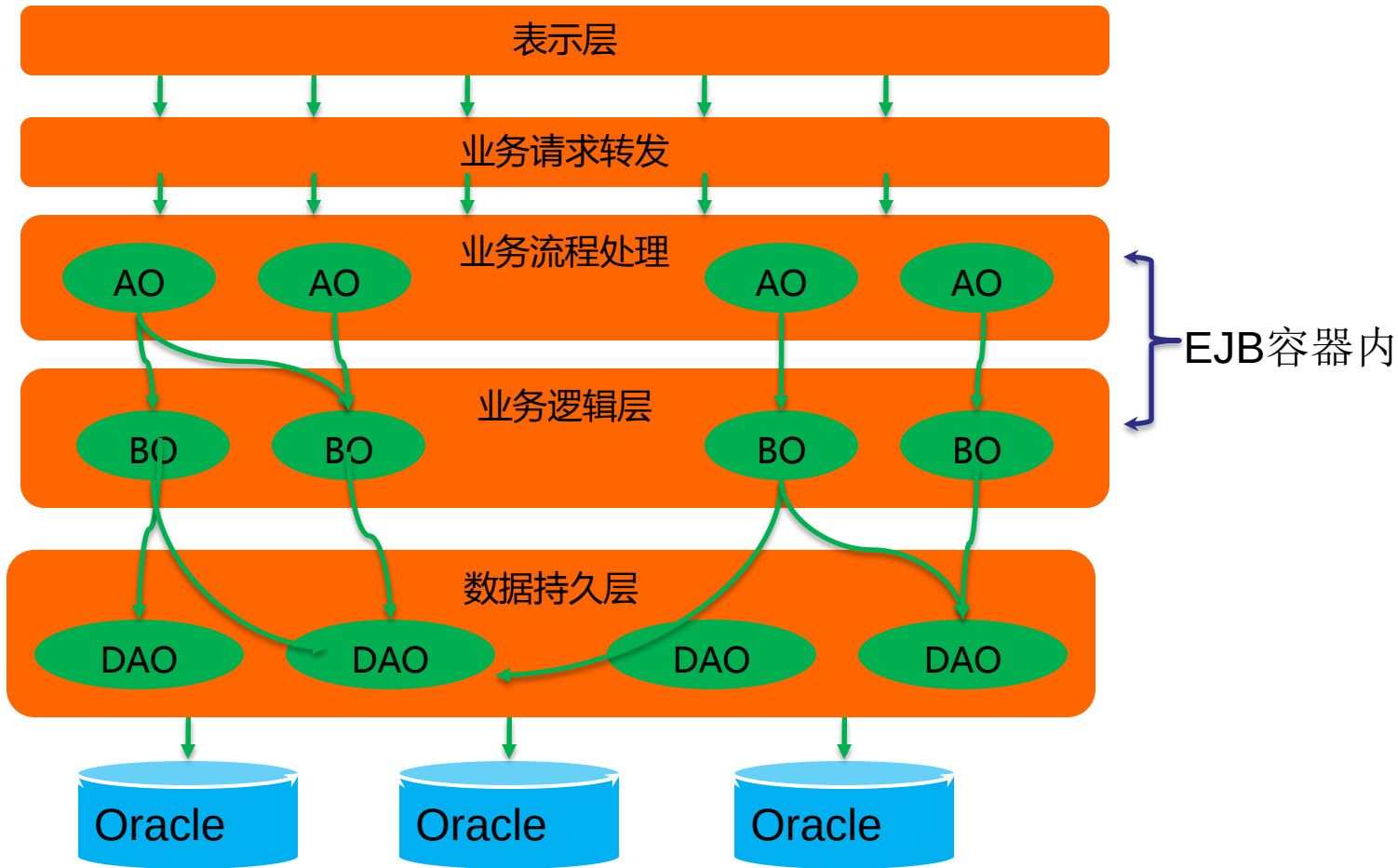
- 无技术积累沉淀
- 无法满足团队并行开发
- 不能满足长期持续维护
- 难以支撑高速业务发展
- 难以支撑系统可伸缩性
- 数据库连接达到上限（每个Oracle数据库大约提供5000个链接）

V2.0架构 2004.02-2005.03

- PHP->Java
- 三层结构
- WebX
- ServiceFramework
- AntX
- ISearch



V2.0架构-以EJB为核心



- 基于Apache Turbine框架改造
- 模块化(car)
- 基于管道流(pipeline)的架构
- 页面布局(Screen Layout Control)
- 基于模板的页面渲染(JSP,Velocity,Freemarker)

- 类似Maven
- Ant ++
- AutoConfig
- 页面布局(Screen Layout Control)
- 基于模板的页面渲染(JSP,Velocity,Freemarker)

- 性能
- 提高开发效率
- AutoConfig
- 降低成本

V2.1架构-2004.10-2007.01

- Weblogic -> Jboss
- JDK4 -> JDK5
- 支持数据库分库
- 抛弃EJB
- 引入Spring
- Session框架重构
- 缓存框架TBCache(Tair前身)
- 淘宝自己的CDN

Linux

Apache

JBoss

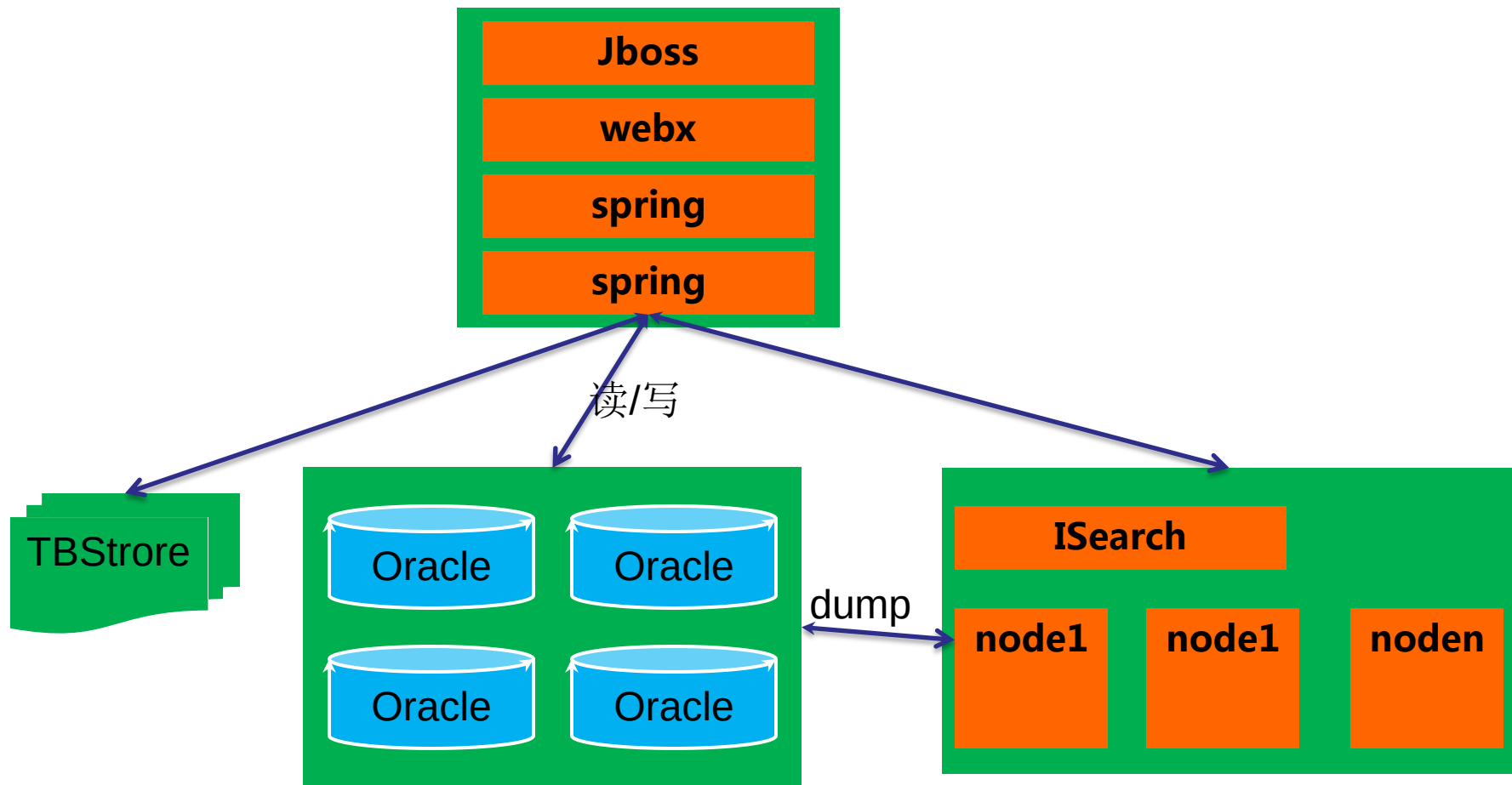
WebX

Spring

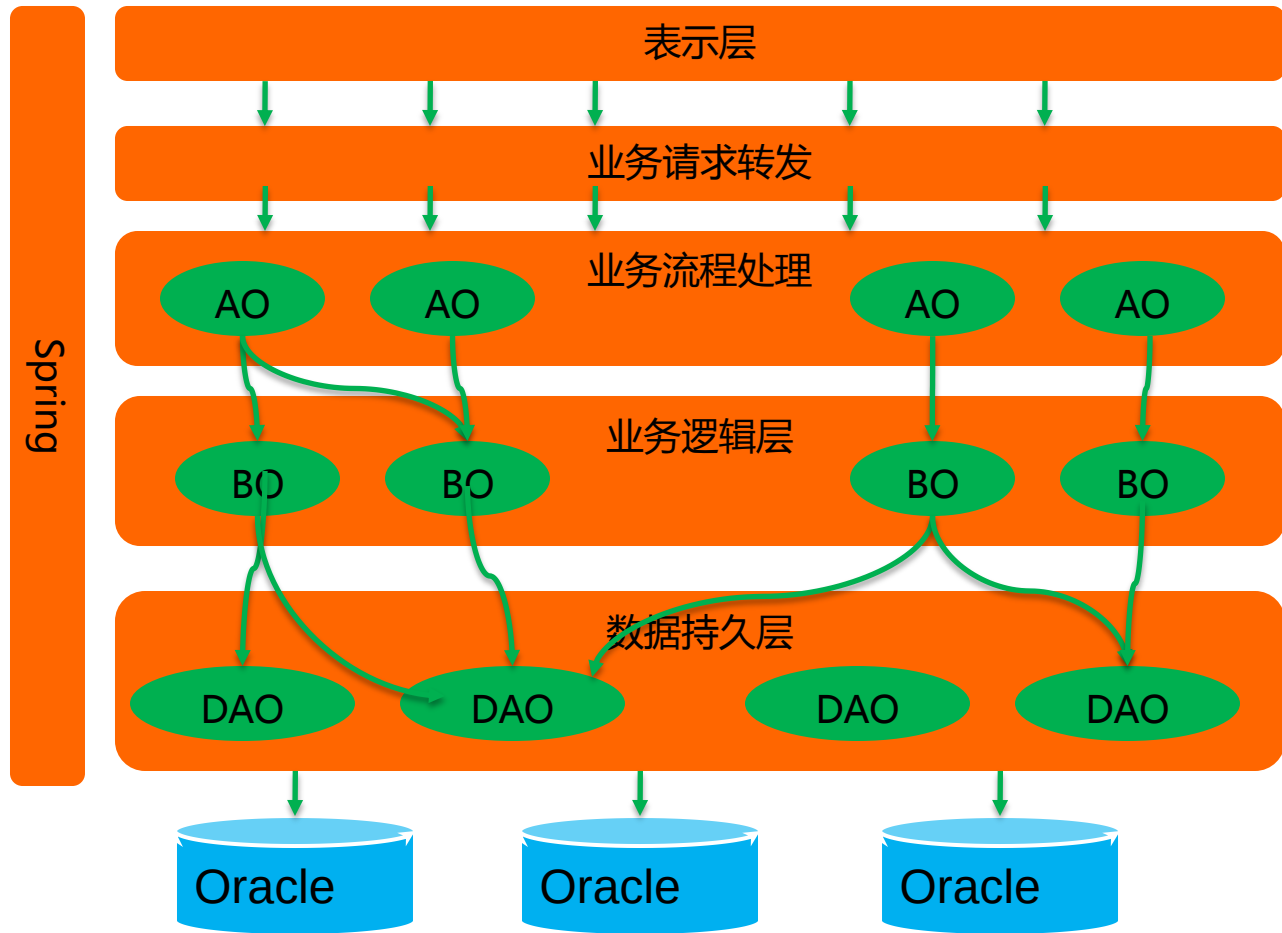
TaobaoIbatis

Oracle

V2.1架构-2004.10-2007.01



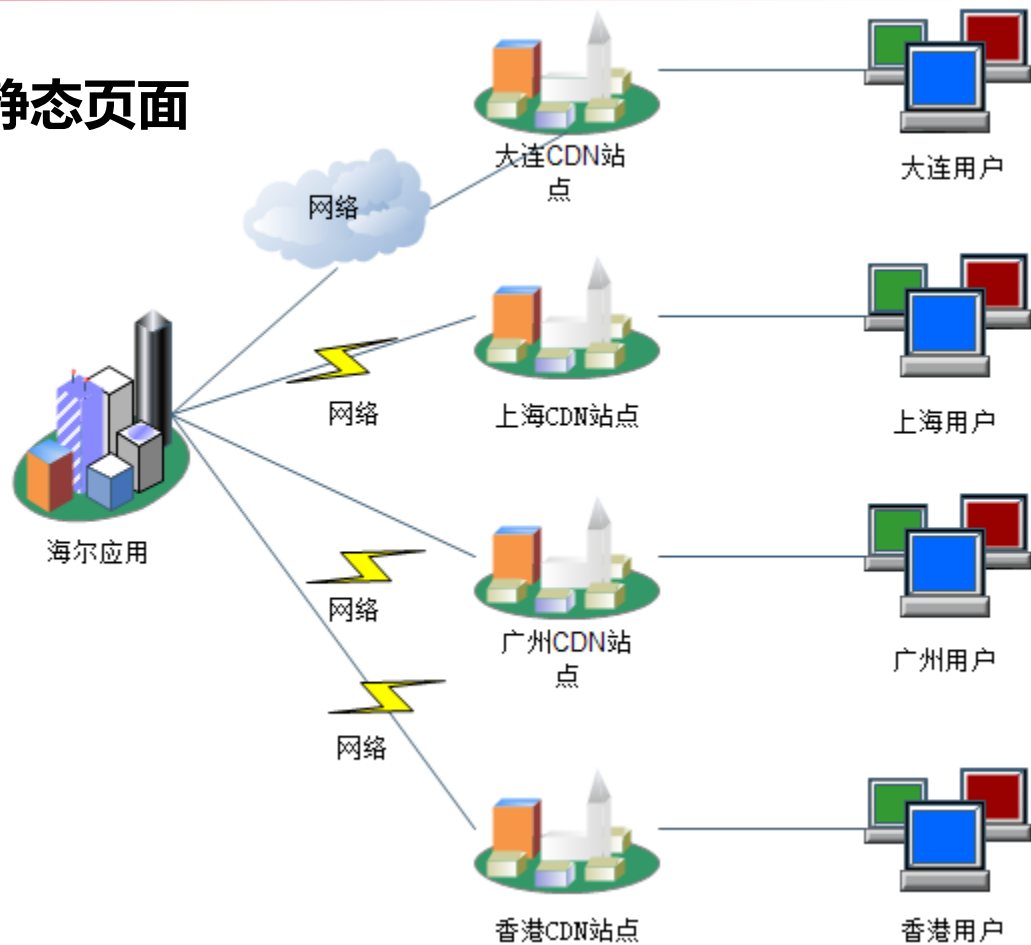
V2.1逻辑架构-以Spring为核心



- 基于Key-Value的高速缓存
- 中心化缓存存储
- 多种缓存策略

V2.1架构-TaobaoCDN

- 静态内容：图片，描述，js,css,静态页面
- 全国20余个站点
- 高峰30G/秒



- 集中存储
- 对代码透明

V2.2架构-需求2006.6-2007.12

- 提高系统性能
- 降低存储成本
- 支持海量数据搜索

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/456021013133010153>