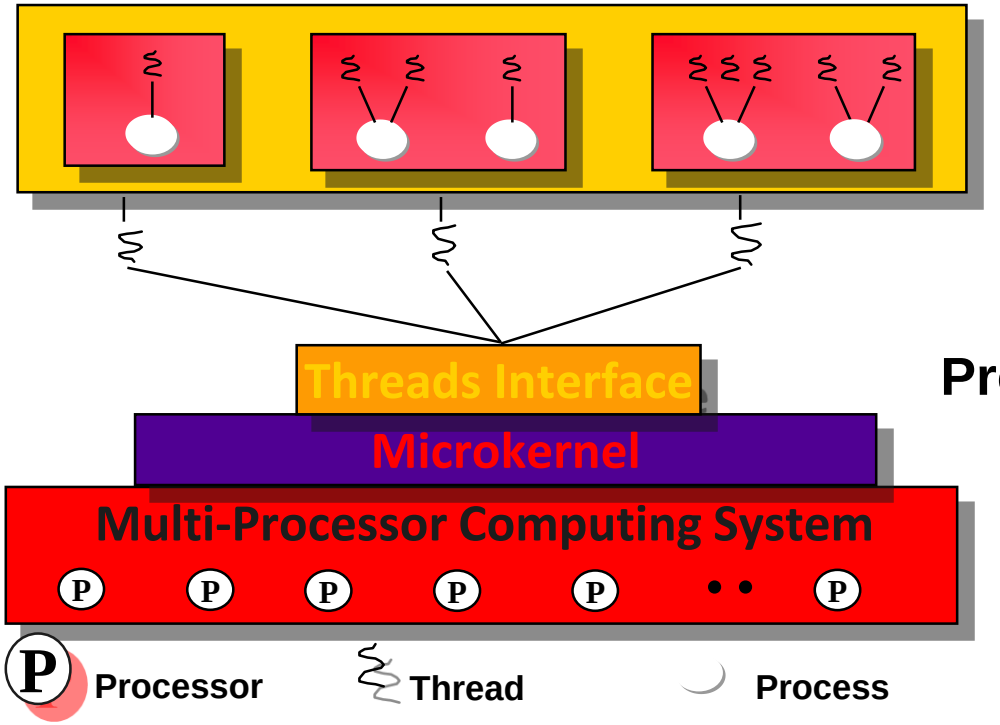
04

并发计算

04 并发计算

- 并发 vs. 并行
- 线程模型
- 案例：高并发网站架构解决方案

4.1 并发 vs. 并行



Applications

Programming paradigms

Operating System

Hardware

4.1 并发 vs. 并行

- **并行计算:** 并行使用多个处理器，比单个处理器更快地解决问题
- 并行计算的例子:
  - **计算机集群**，包含多台PC，多台PC之间使用高速网络相互连接
  - **对称多处理器**(SMP*)，一台计算机上汇集了一组处理器(多CPU)，各CPU之间共享同一个内存系统
  - **芯片多处理器**(CMP)，单个芯片上汇集了多个处理器（多核）
- 并发执行源于对性能的渴望, 与多用户分布式系统中固有的并发性不同

4.1 并发 vs. 并行

- 为什么要并发?
  - 充分利用硬件/软件的先进技术，例如多处理器和操作系统对多线程的支持
  - 提高性能，例如重叠计算和重叠通信
  - 加快响应速度，例如GUI和网络服务器
  - 简化程序结构，例如同步与异步网络IPC

4.1 并发 vs. 并行

- 定义
  - 并发（Concurrency）
    - 逻辑上地同时执行
    - 并不意味着有多个处理器模块
  - 并行（Parallelism）
    - 物理上地同时执行
    - 存在多个处理器模块或多个独立的机器设备
  - 无论是并发还是并行，都需要控制对共享资源的访问进行，例如： I / O设备，文件，数据库，内核数据结构，控制台等。

4.1 并发 vs. 并行

- 费林分类(**Flynn's Classical Taxonomy**)
  - 单指令流单数据流(SISD)，常见的单处理器计算机
  - 单指令流多数据流(SIMD)，特殊用途的低粒度多处理器，一个控制单元将相同的指令广播到所有持有不同数据流的处理器。例如，nVIDIA图形协处理器
  - 多指令流单数据流(MISD)，例如，流水线技术。
  - 多指令流多数据流(MIMD)，这是最流行的模型。单程序多数据（SPMD）是它非常有用的一个自己。 请注意，这与SIMD不同。

4.1 并发 vs. 并行

- 数据并行：SIMD和SPMD
- 功能并行：MISD
- 特殊地，MIMD既然在数据上并行，也在功能上并行。后者可以在任何指令级别并行，不同的指令可以在任何时间跨处理器执行，或者也可以在高级别的功能空间上并行

4.2 线程模型

- 多进程 vs. 多线程

- 多进程（Multiprocess）

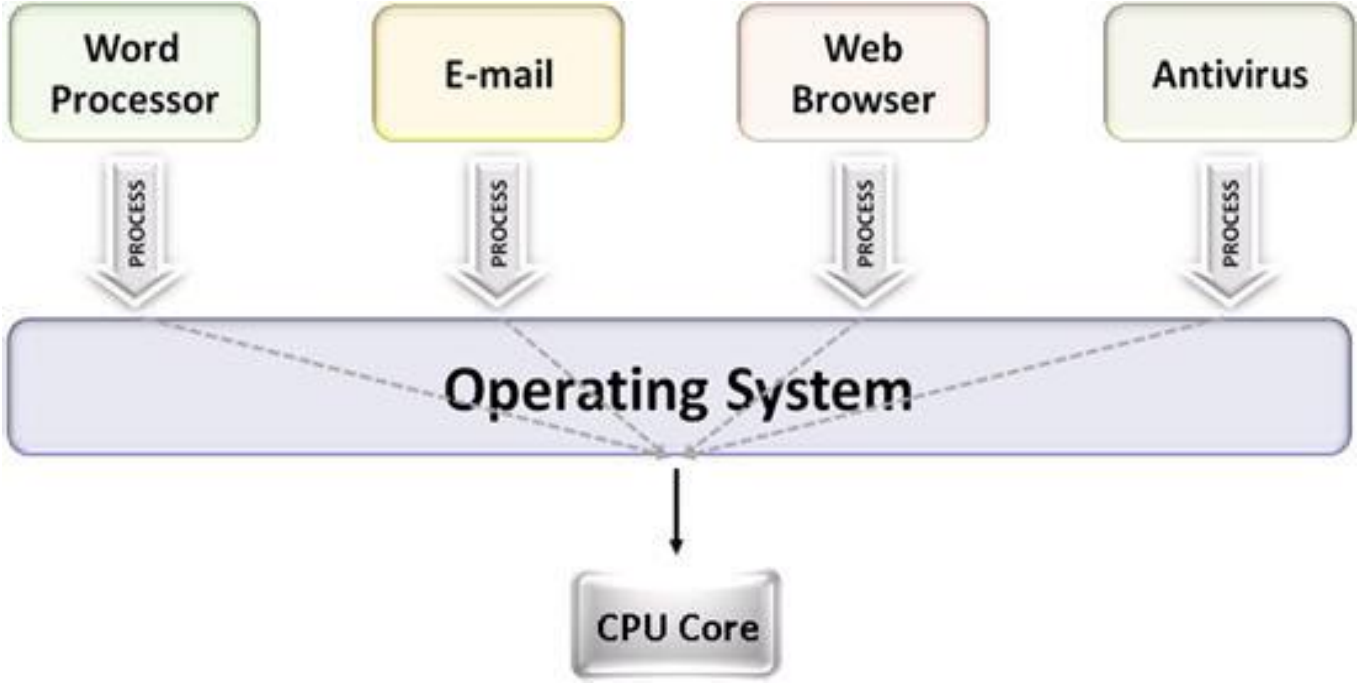
  多个任务或进程共享公共系统资源，如CPU，I / O等。

- 多线程（Multithread）

  进程内的多个执行单元共享系统资源。如CPU，I / O等。

4.2 线程模型

- 多进程 （单核）



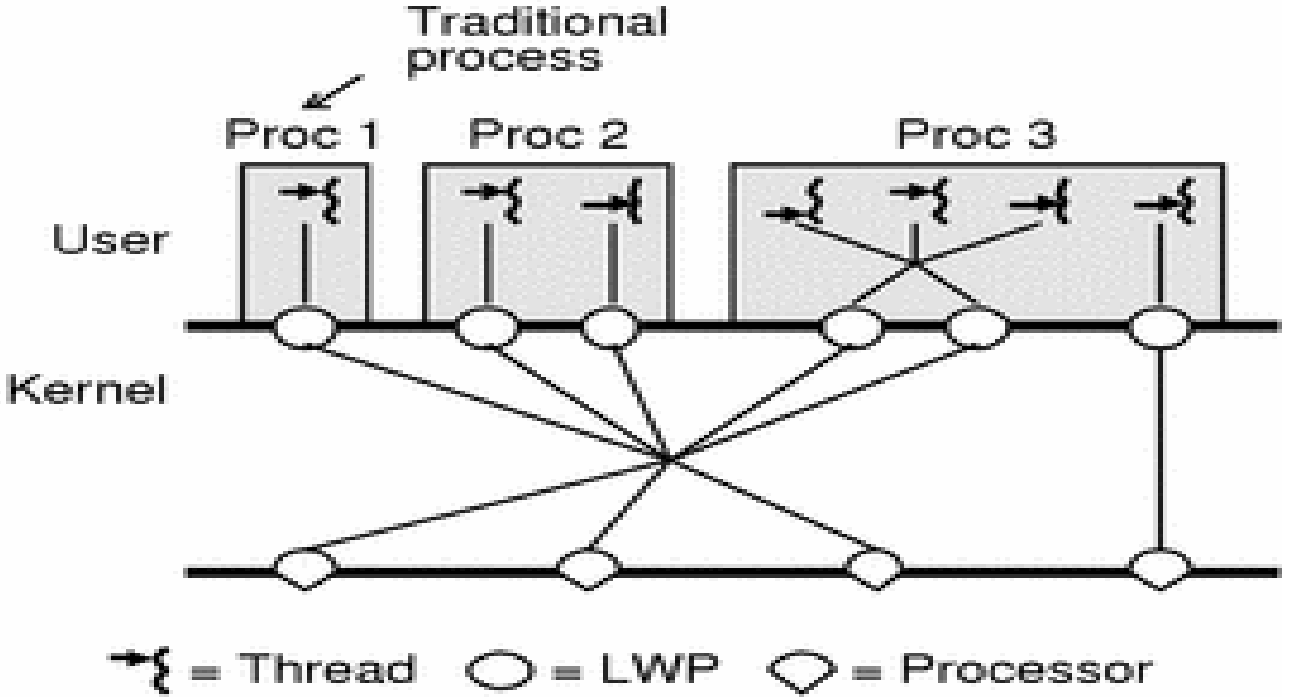
4.2 线程模型

- Java线程模型

- 两种级别的进程
  - 用户线程（Java线程类）
  - 内核线程（操作系统内核支持）
- 三种线程模型
  - 一对一
  - 多对一
  - 多对多

4.2 线程模型

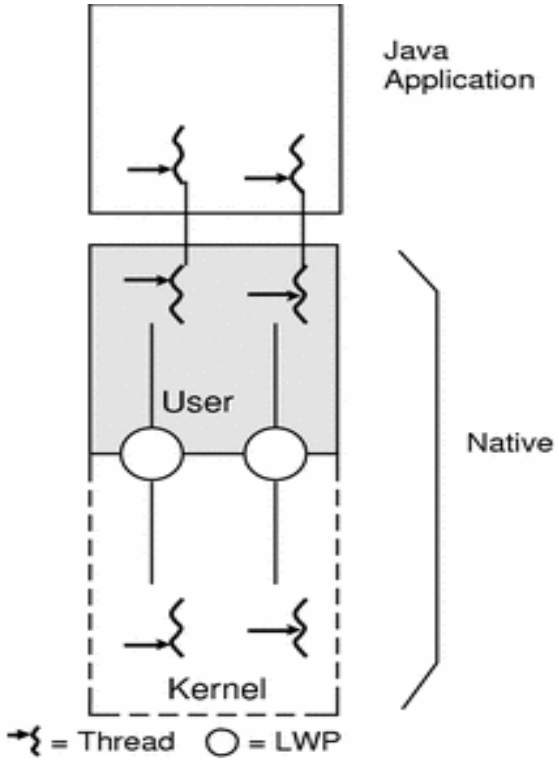
- Solaris两级线程结构



4.2 线程模型

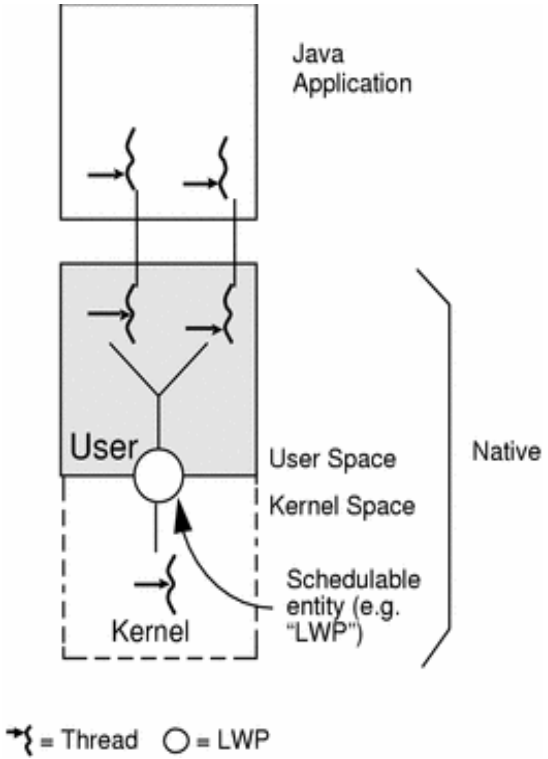
- 一对一模型

  - 一个用户线程映射为一个内核线程
  - 高并发
  - 可能会导致内核过载



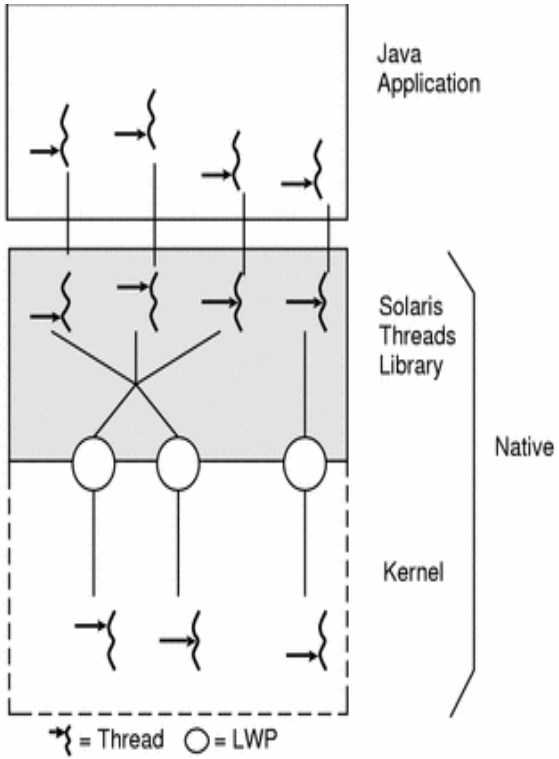
4.2 线程模型

- 多对一模型
  - 多个用户线程映射为一个内核线程
  - 吞吐量低
  - 现在几乎没有系统使用这个模型



4.2 线程模型

- 多对多模型
  - M个用户线程映射为N个内核线程
  - 有效地利用资源
  - 需解决调度问题



4.3 案例：高并发网站架构解决方案

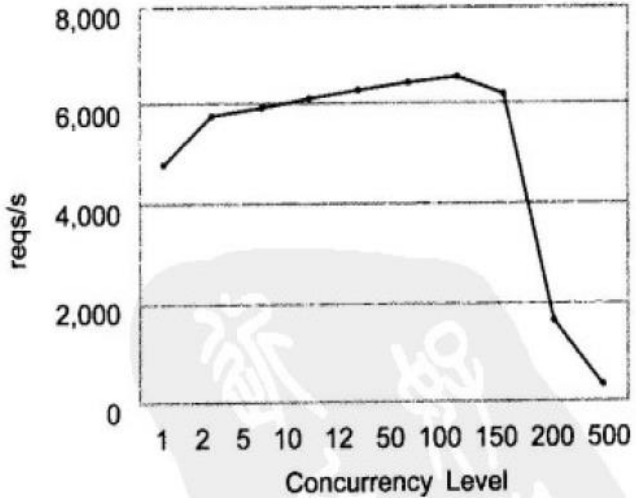
- 并发计算架构一个主要的应用领域
  - 高并发高负载网站
    - 用户数庞大
    - 峰值高负载
    - 同时读写/数据库
    - 性能瓶颈
    - 采用支持高并发技术

4.3 案例：高并发网站架构解决方案

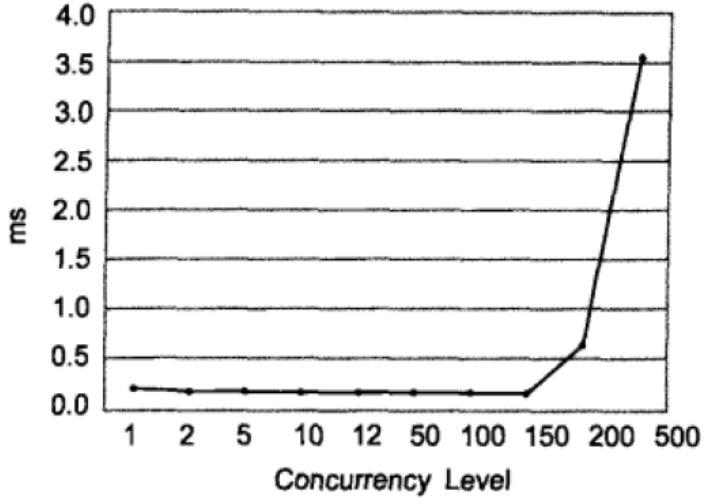
- 高并发网站架构设计案例 – 阿里巴巴中国站
  - 中国站网站的正常流量情况
    - 并发（单台），高峰期 < 10
    - 吞吐量（TPS，单台）　高峰期 <60
    - CPU负载 Load　高峰期，<2，大部分服务器<1
    - CPU使用率，一般只占1颗核，平均60%左右
    - 服务器平均响应时间 高峰期 <150ms
  - 高并发下的风险
    - 网络带宽耗尽
    - 服务器Load飙高，停止响应
    - 数据库瘫痪
  - 高并发下的事故
    - 事故：网站运营旺旺推广页面弹出，1兆大图片导致带宽耗尽
      增加审核机制：运营推广增加的图片流量不能超过现有流量的30%
    - 合作媒体推广：迅雷，暴风影音浮出广告，导致旺铺集群Crash

4.3 案例：高并发网站架构解决方案

- 高并发度对网站性能的影响



并发数对吞吐量的影响



并发数对服务器平均响应时间的影响

4.3 案例：高并发网站架构解决方案

- 高并发高负载实例 – 1688.com秒杀活动

  - 商业需求
    - 并发（单台），高峰期 < 10
    - 为庆祝1688.com开业退出88小时不间断秒杀活动
    - 每小时整点推出8款商品，拖拉机，牛，马桶，沙发……
    - 每款商品供168件，每人限批3件，成交人数56人
    - CCTV黄金广告时间，各种网络,平面媒体轰炸，总广告费：1.5亿

4.3 案例：高并发网站架构解决方案

- 高并发高负载实例 – 1688.com秒杀活动
  - 技术挑战
    - 瞬间高并发
      - 8000并发：预估秒杀在线人数可达8000人
      - 风险：带宽耗尽
      - 服务器：崩溃，可以理解成自己给自己准备的DDoS攻击
    - 秒杀器
      - 第一种：秒杀前不断刷新秒杀页面，直到秒杀开始，抢着下单
      - 第二种：跳过秒杀页面，直接进入下单页面，下单

4.3 案例：高并发网站架构解决方案

- 网站并发架构
  - 服务器集群
    - style服务器（Lighttpd集群）：5台
    - 图片服务器（Nginx集群）：5台
    - 静态服务器（Apache集群）：10台
    - 交易服务器（JBoss动态集群）：10台
  - 网络带宽
    - 图片出口带宽上限：2.5G （出口带宽支持10G，但图片服务器集群的处理能力：图片服务集群最大并发处理能力 X 网站平均图片大小 = 2.5G)
    - CDN准备：Chinacache沟通；借用Taobao CDN

4.3 案例：高并发网站架构解决方案

- 1688.com秒杀系统：架构目标

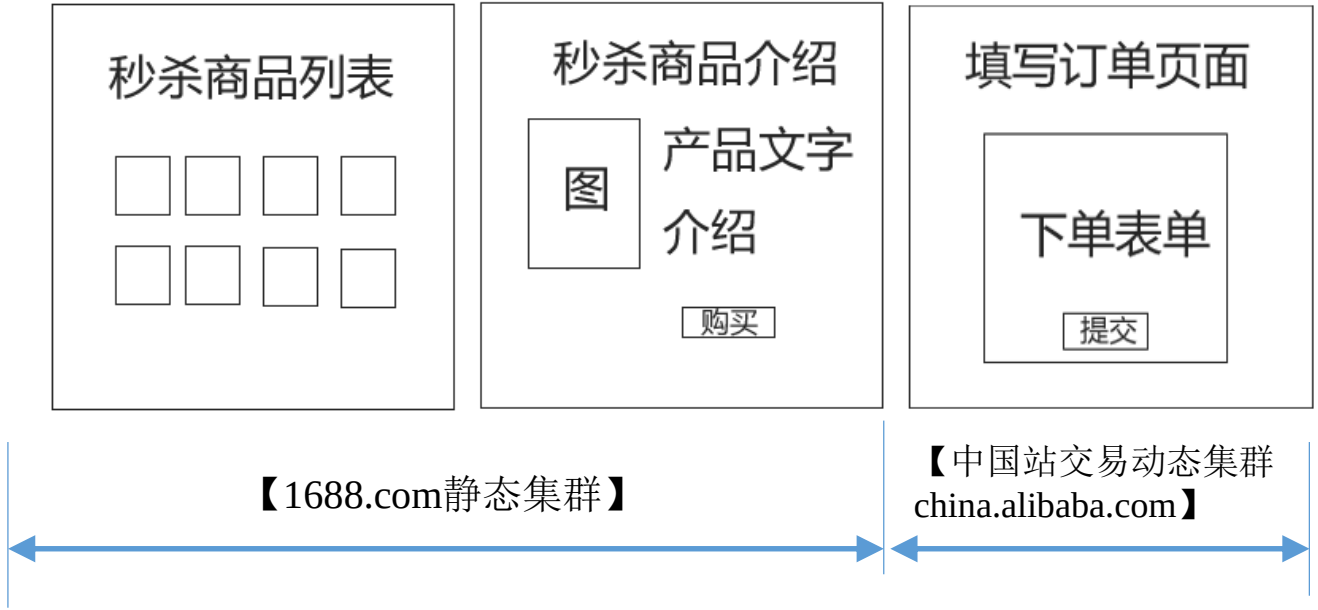
1. 图片网络带宽：1.0G
   - 新增图片带宽：必须控制在 1.0G 左右
   - 每件商品秒杀页面的图片总大小不得超过：1000000/(1000*8) = 125K/每商品　静态服务器(Apache集群)：10台

2. 网站并发
   - 单件商品并发：1000 【来自运营的预估】
   - 总并发：8（件商品）X 1000（人/商品）= 8000

4.3 案例：高并发网站架构解决方案

- 秒杀系统组成

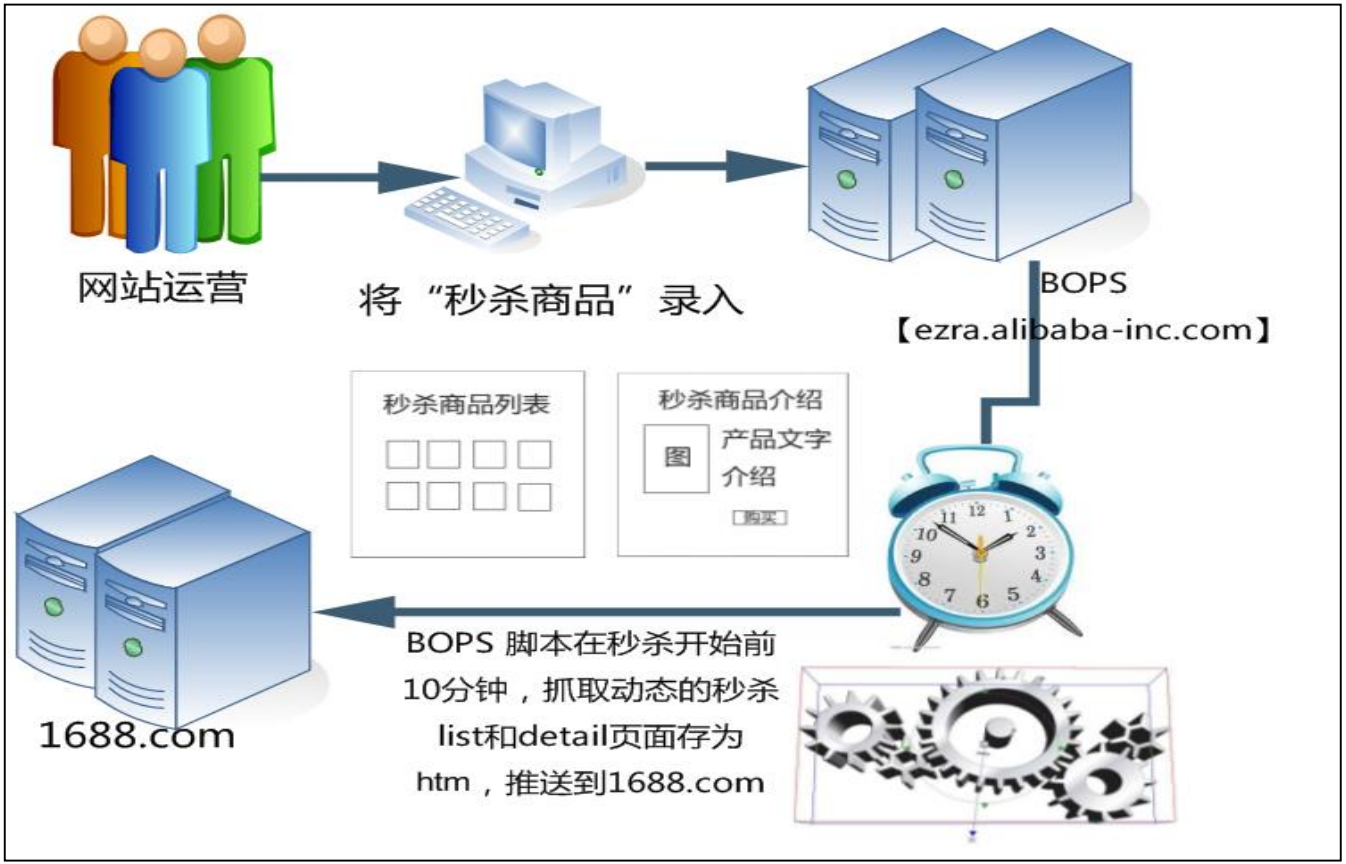
三个页面组成：**秒杀商品列表，秒杀商品介绍，下单**



4.3 案例：高并发网站架构解决方案

- 1688.com秒杀系统：设计原则
  - 静态化
    - 采用JS自动更新技术将动态页面转化为静态页面
  - 并发控制，防秒杀器
    - 设置阀门，只放最前面的一部分人进入秒杀系统
  - 简化流程
    - 砍掉不重要的分支流程，如下单页面的所有数据库查询
    - 以下单成功作为秒杀成功标志。支付流程只要在1天内完成即可
  - 前端优化
    - 采用YSLOW原则提升页面响应速度

4.3 案例：高并发网站架构解决方案

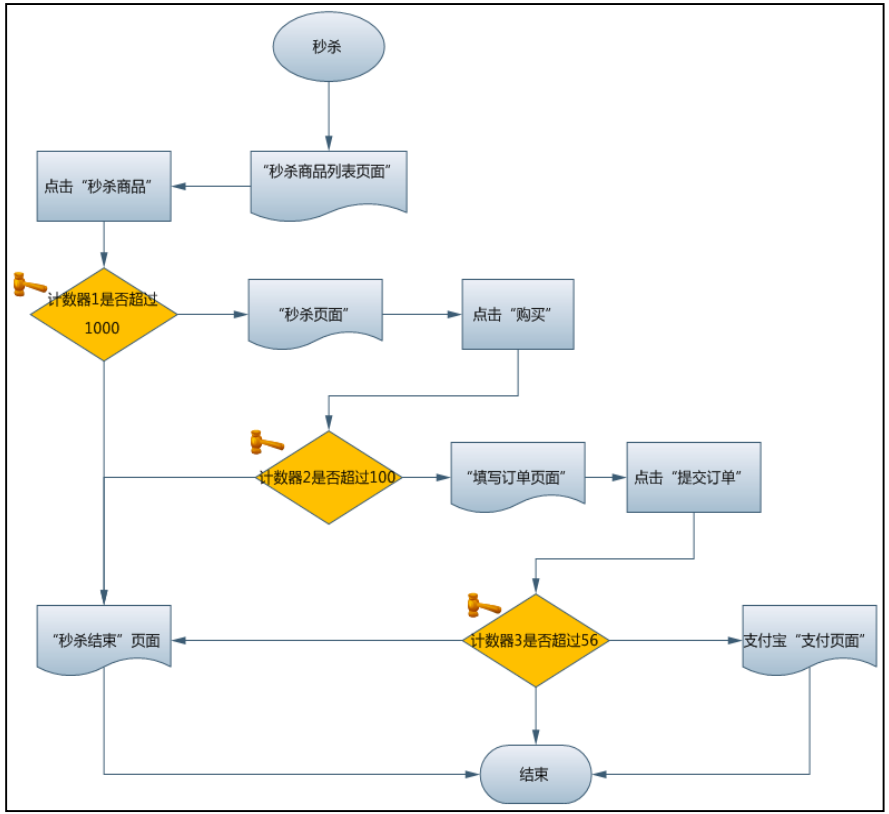


4.2 线程模型

- 三道阀门的设计

阀门：基于TT的计数器

序号	阀门上限
1	限制进入秒杀页面，1000
2	限制进入下单页面，100
3	限制进入支付宝系统，56



4.3 案例：高并发网站架构解决方案

- 秒杀器的预防
  - 秒杀Detail页面
    - URL：随机
    - 秒杀前2秒放出,脚本生成，秒杀前
    - 1000次访问上限控制【每件商品只能放入1000人浏览】
  - 下单页面
    - 订单ID，随机
    - 不能直接跳过秒杀Detail页面进入
    - 每个秒杀商品，带预先生成的随机Token作URL 参数
    - 如果秒杀过，直接跳到秒杀结束页面
    - 100次访问上限控制【每件商品只能放入1000人下单】

4.3 案例：高并发网站架构解决方案

- 秒杀静态页面优化
  - 图片合并
    - 减少HTTP请求数，减少请求等待数
    - 减少发送cookies的量
  - HTML内容压缩
  - 图片压缩：图片Bytes < 长X宽/2250
  - HTML Header Cache-Control设置
  - CSS，JS 精简
    - CSS,JS 精简到极致，部分直接写在页面中，减少Http请求次数

4.3 案例：高并发网站架构解决方案

- 下单页面优化
  - 数据库操作：全部砍掉
    - 原下单页面要访问8次数据库，全部砍掉
  - 秒杀流程精简
    - ✱ 砍掉填写或选择收货地址，放在秒杀成功后填写
    - ✱ 砍掉调用是否开通支付宝接口，秒杀首页文案提示必须开通
  - 采用内存缓存
    - 秒杀Offer数据，支付宝项目信息，缓存

4.3 案例：高并发网站架构解决方案

- 交易系统性能优化
  - 交易系统调优目标：

	并发	TPS
下单页面 （优化前）	20	100
下单页面 （优化后）	40	400

  - 关闭 Keep Alive(分析交易系统访问日志，用户在短时间内连续点击概率很低)
  - JVM优化，优化CMS 垃圾回收器的参数
  - 消灭 Top10 Bottlenecks
    - Velocity参数调优
    - 采用DBCP1.4 替换C3P0
    - Offer 产品参数的XML解析

4.3 案例：高并发网站架构解决方案

- 应急预案
  - 域名分离，独立域名，不影响中国站原有业务
    - Style集群： style.1688.china.alibaba.com
    - 图片服务器集群：img.1688.china.alibaba.com
    - 静态页面集群：page.1688.china.alibaba.com
  - 机动服务器10台，备用
  - 拆东墙补西墙战略
    - ✱ 5天时间来不及采购服务器，因此SA待命随时准备将非核心应用集群的冗余服务器下线，加入到秒杀集群

4.3 案例：高并发网站架构解决方案

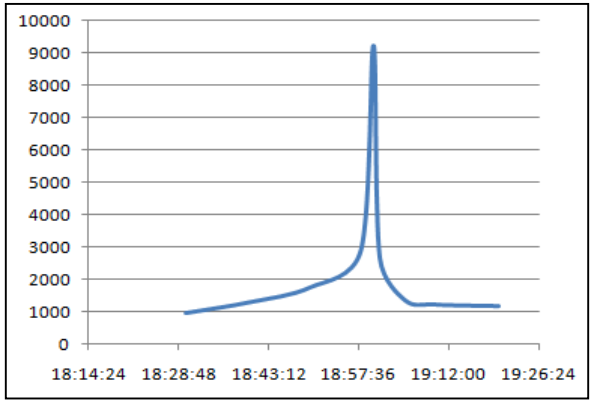
- 应急预案（续）
  - 壁虎断尾策略

    当所有办法均失效的情况下，例如流量耗尽

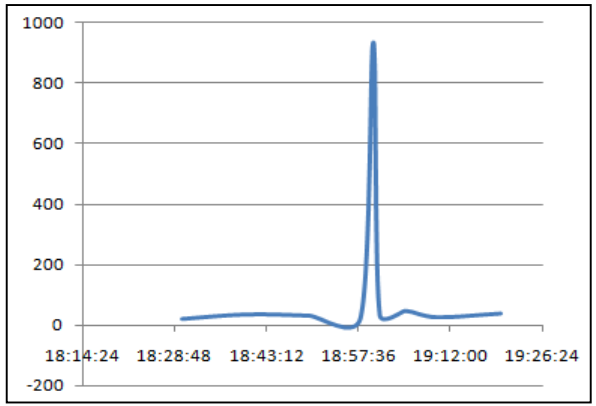
    - 非核心应用集群统统停止服务，如资讯，论坛，博客等社区系统
    - 保住首页，Offer Detail,旺铺页面等核心应用的可用性
  - 万能出错页面：秒杀活动已经结束
    - 任何出错都302跳转到此页面
    - 位于另外集群

4.3 案例：高并发网站架构解决方案

- 秒杀活动结果
  - 88小时秒杀，坚守阵地，大获成功
  - 秒杀还是被秒杀？终于有了答案
  - 三道阀门设计非常有效，拦住了秒杀器



1688.com 静态集群总并发情况
（首页，秒杀列表，秒杀商品页面）



交易系统集群总并发情况
（下单页面）

4.3 案例：高并发网站架构解决方案

- 超大型门户网站高并发架构改进解决方案

  改进一：采用更轻量/快速的服务器

  - 采用Lighttpd 替代 Apache 杀手锏（AIO）

    Lighttpd 1.5　VS　Apache2.2.4

小页面性能（100K）

lighttpd			
backend	MByte/s	req/s	user + sys　0KB/S
writev	82.20	802.71	90%
linux-sendfile	70.27	686.32	56%
gthread-aio	75.39	736.23	98%
posix-aio	73.10	713.88	98%
linux-aio-sendfile	31.32	305.90	35%
others			
Apache 2.2.4 (event)	70.28	686.38	60%
LiteSpeed 3.0rc2	70.20	685.65	50%

4.3 案例：高并发网站架构解决方案

- 超大型门户网站高并发架构改进解决方案

Lighttpd 1.5　　VS　　Apache2.2.4

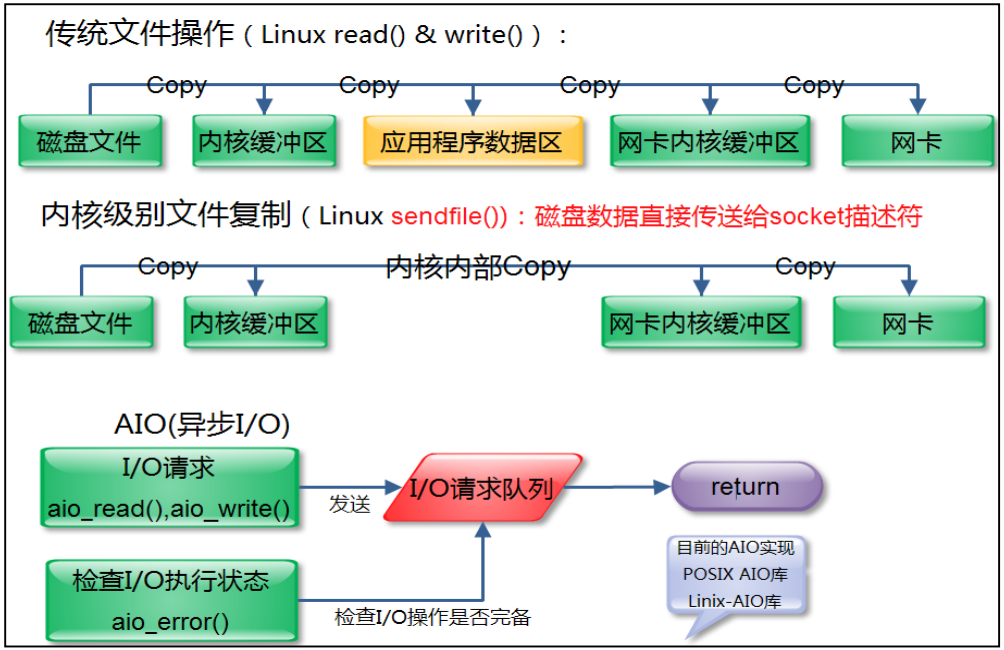
大页面性能（10M）

lighttpd			
backend	MByte/s	req/s	user + sys
writev	82.20	8.76	80%
linux-sendfile	53.95	5.65	40%
gthread-aio	83.02	8.66	90%
posix-aio	82.31	8.60	93%
linux-aio-sendfile	70.17	7.35	60%
others			
Apache 2.2.4 (event)	50.92	5.33	40%
LiteSpeed 3.0rc2	55.58	5.80	40%

4.3 案例：高并发网站架构解决方案

改进一：采用更轻量/快速的服务器

- 性能关键：Web Server 的高性能I/O



Apache1.3

Apache2.2支持

Lighttpd1.5

4.3 案例：高并发网站架构解决方案

改进二：前端优化自动化

- 中国站服务器响应时间<150ms, 但Offer Detail页面用户等待时间5s,大部分时间耗在路上（资源请求和网络传输）
- 图片自动压缩 （CMS自动压缩）
- Cookies服务化（控制cookies的大小）
- 中国站前端延迟加载框架SmartLoad（只加载首屏数据）
- Google mod_pagespeed module
  - 自动压缩图片，静态资源，智能浏览器缓存技术
- Google Diffable （增量下载静态资源技术）

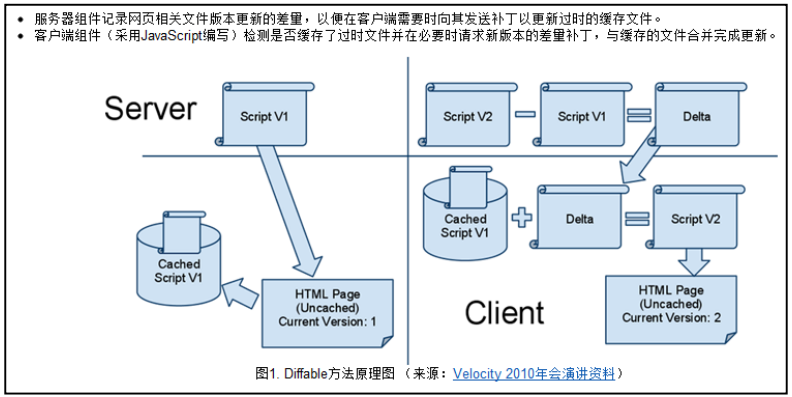
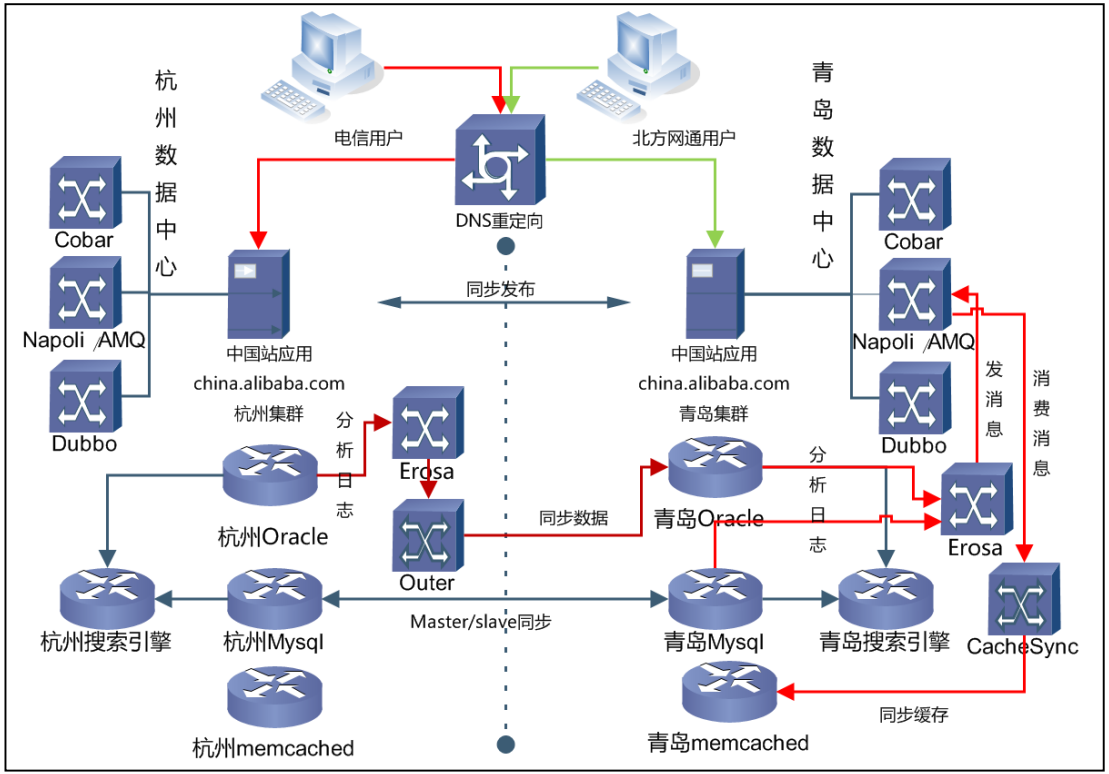


图1. Diffable方法原理图（来源：Velocity 2010年会演讲资料）

4.3 案例：高并发网站架构解决方案

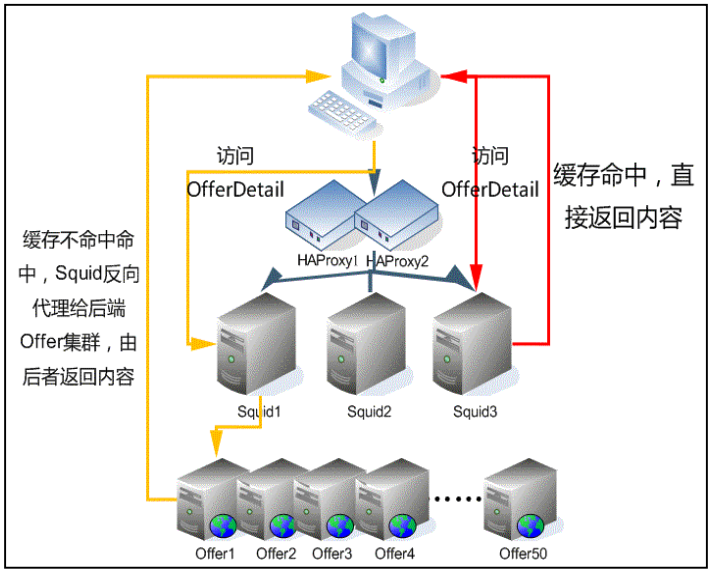
改进三：架设镜像站组建CDN



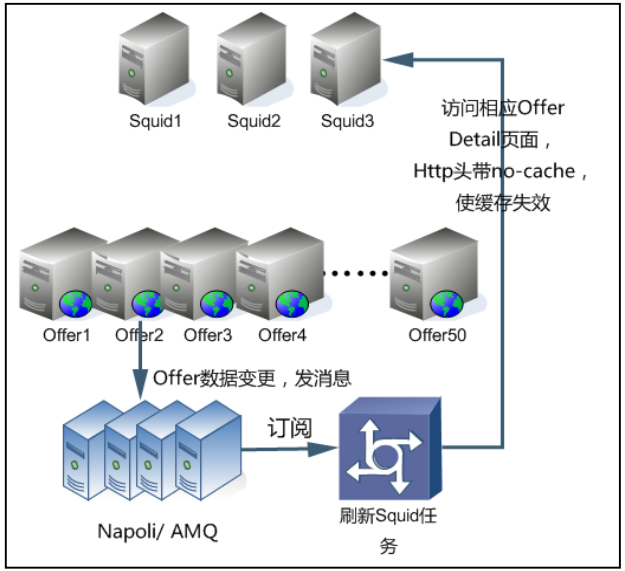
4.3 案例：高并发网站架构解决方案

改进四：采用反向代理 加速核心页面

- 在Offer集群前部署Squid反向代理集群



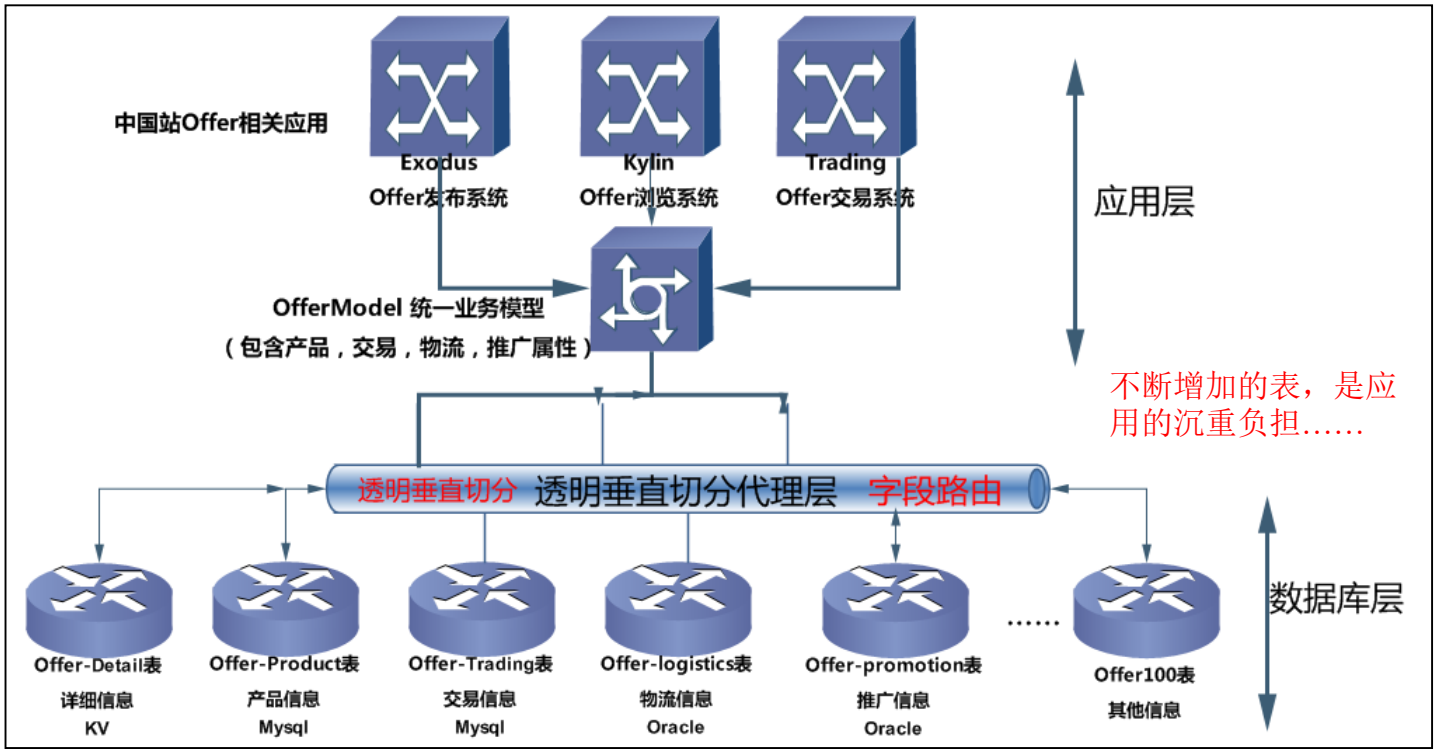
Offer Detail的Squid反向代理改造



基于消息的Squid缓存更新机制

4.3 案例：高并发网站架构解决方案

改进五：海量数据的透明垂直切分



End of Session
Thank you!