

# 淘宝前台系统性能分析与优化

叔同



追風堂



- **产品技术部-中间件&稳定性平台-叔同**
- 2005年毕业于吉林大学，计算机科学与技术专业；2010年加入淘宝. 4年行业软件开发经验，3年互联网开发经验；对构建高性能、可扩展的Web应用十分感兴趣；目前在淘宝Java中间件团队从事性能优化相关的工作



:[淘宝叔同](#)



:[jlusdy](#)



- **提升容量**

- 提升QPS
- 降低Server RT
- 降低服务器成本

- **提升用户体验**

- 降低响应时间
- 提升用户体验
- 提升转化率和交易量





- 优化是持续性的工作

- 衡量现状

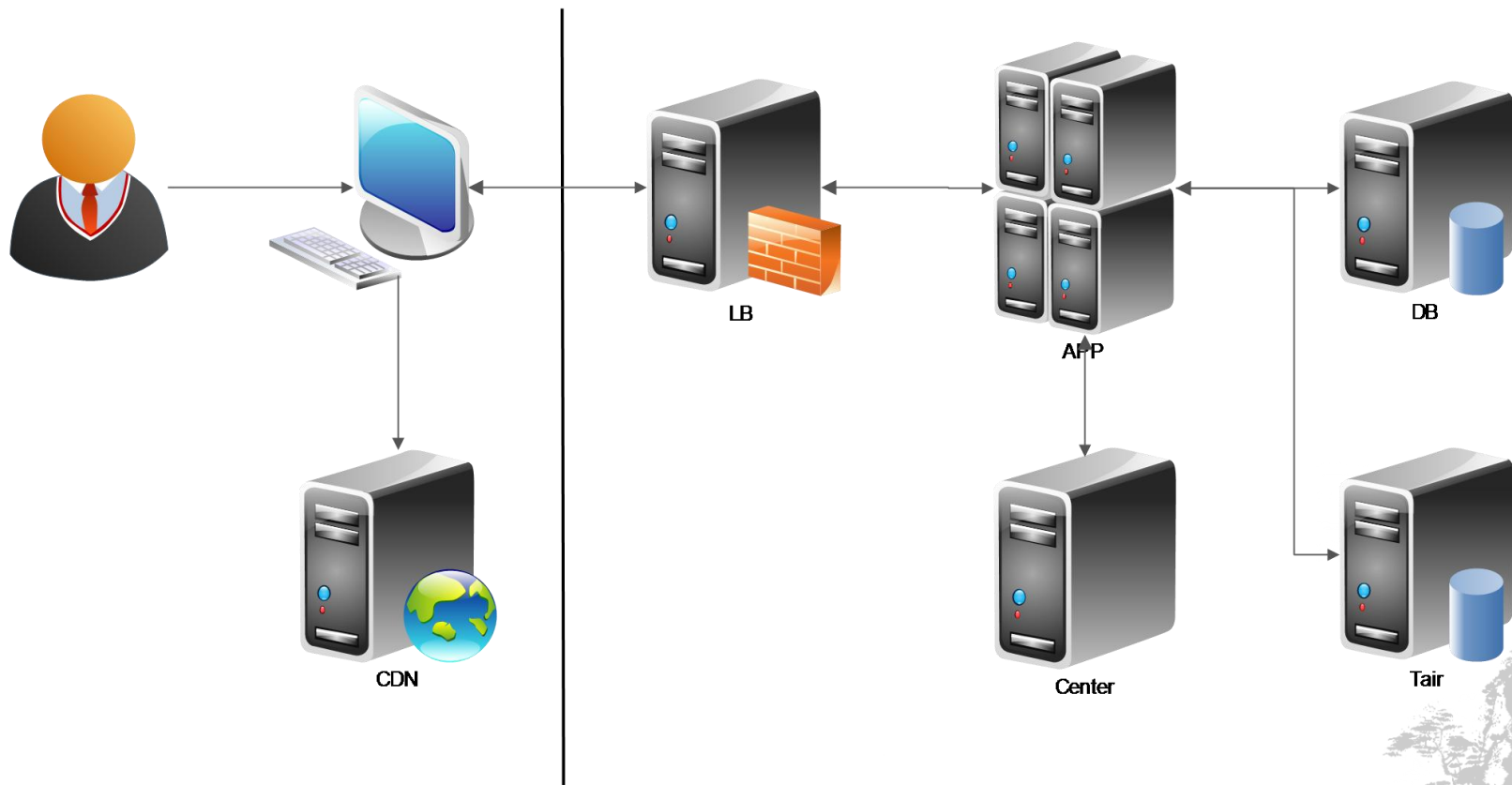
- Qps
    - RT
    - 页面大小
    - 每请求内存

- 设定目标

- 性能优化



# 前台系统结构





- 服务器端分析
- 服务器端优化
- 浏览器端分析
- 浏览器端优化



# 服务器端分析



追風堂



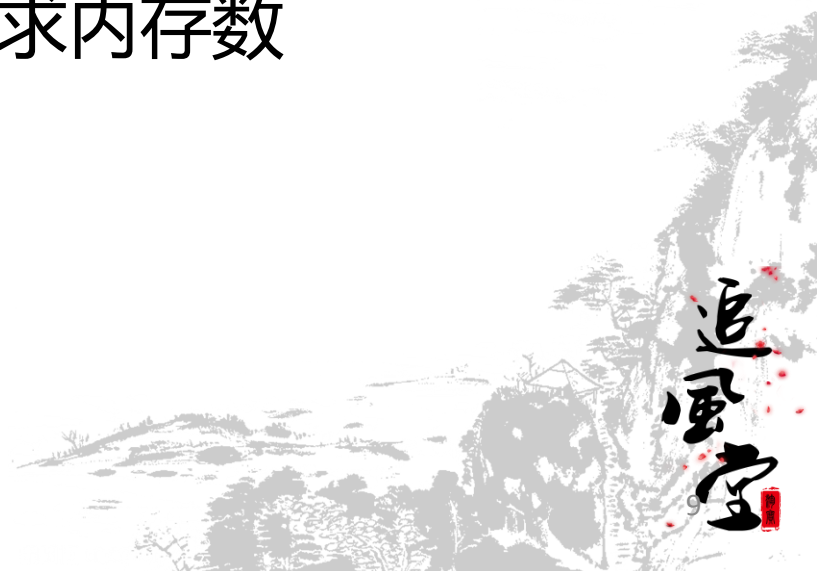
1. 线上压测
2. 分析工具
3. 时间消耗分析
4. 瓶颈查找







- Ab
  - `ab -c 12 -n 20000 localhost/market.htm`
- Httpload
  - `http_load -rate 300 -seconds 120 item-urls`
  - `http_load -parallel 40 -seconds 120 item-urls`
- Apache Nginx分流
  - 非幂等性请求
- 得到QPS/RT/页面大小/请求内存数
- 注意事项
  - Offline机器
  - 避免本机压测
  - 关闭Keep alive
  - 关注系统表现



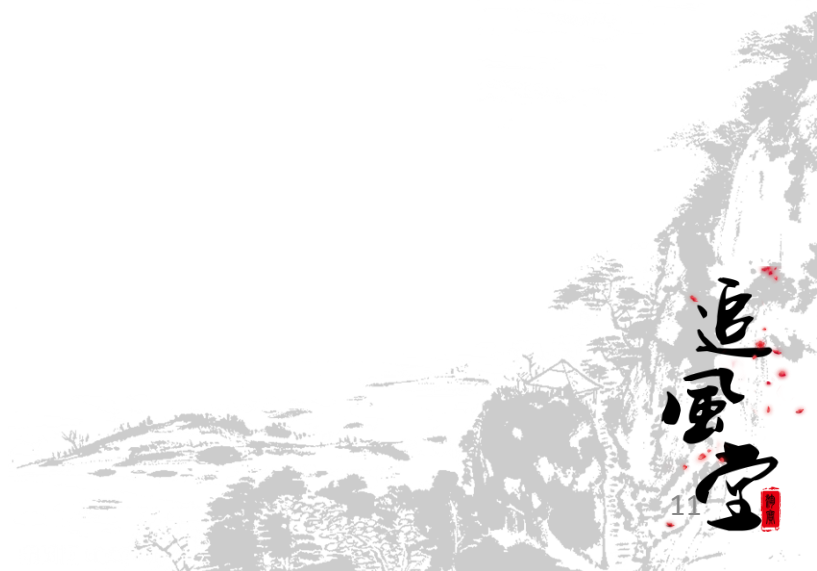


- httpload
- 负载均衡配置
- VIP offline机器
- 注意事项
  - 凌晨压测
  - 关注系统表现
  - 关注依赖集群表现
  - 相关人员在场





- **OS**
  - top, sar, vmstat, mpstat, iostat, netstat
- **JVM**
  - jps, jstack, jmap, jstat
- **GC**
  - Gchisto, PrintGCStats





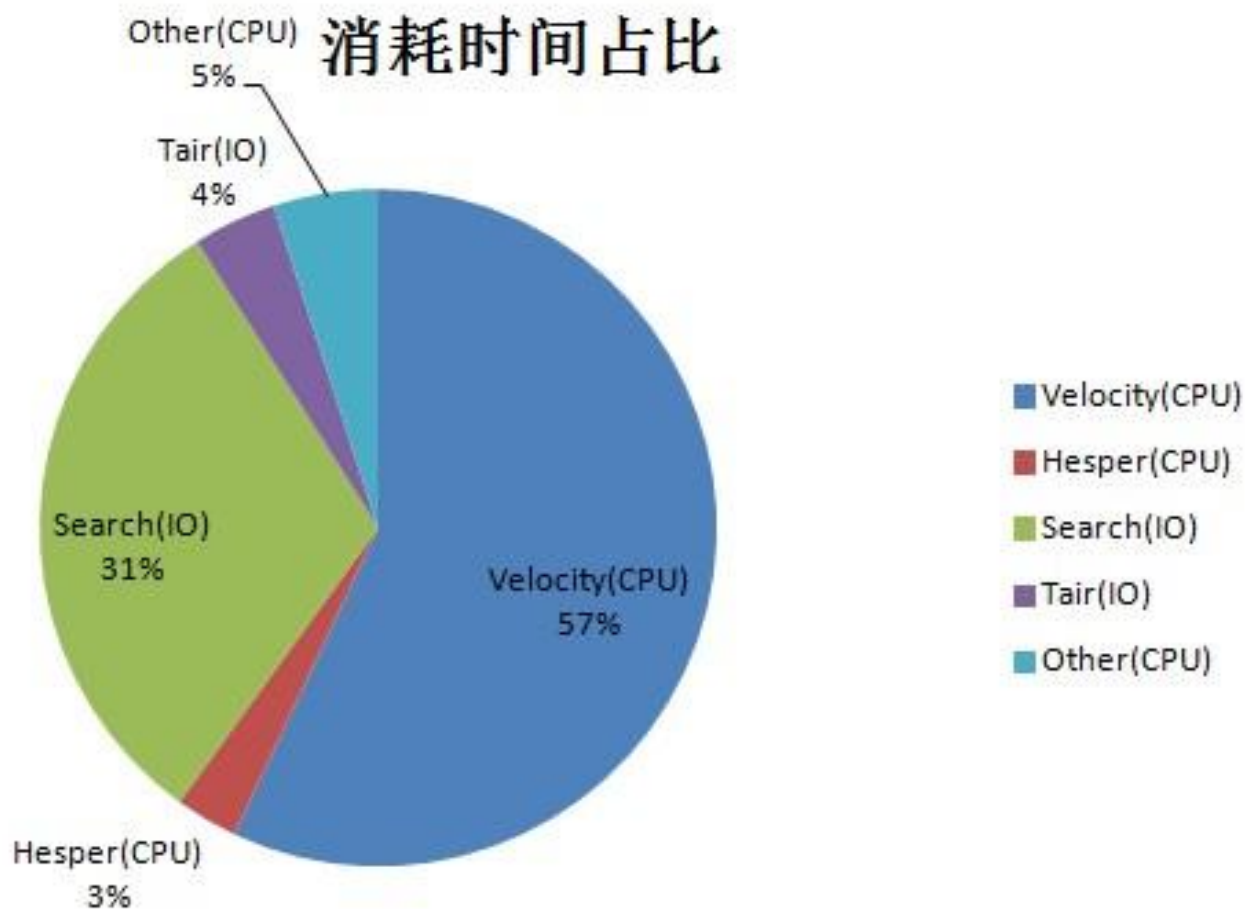
- WebX Profiler
- TProfiler
- BTrace
- VisualVM(Remote Edition), Xming
- 性能分析平台
- OProfile/Perf Google-Perftools



- **页面模板渲染**
  - Velocity
  - 页面较大
- **外部调用多**
  - DB, HSF, Tair, Search, TFS
- **磁盘读写少**
  - Log
  - Vmcommon
- **QPS相对较低**



# 时间消耗分析





- **什么是瓶颈**

- 瓶颈是系统中比较慢的部分,在瓶颈完成前,其他部分需要等待

- **使用TProfiler**





- **CPU计算**

- 字符串查找 替换 拼接
- 编码 解码
- Gzip压缩

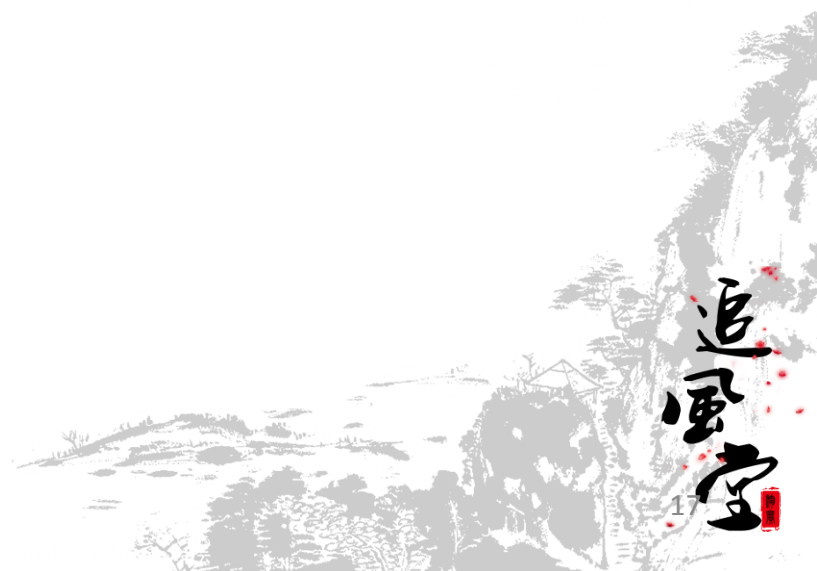
- **外部调用**

- 网络IO开销
- 序列化反序列化





# 服务器端优化

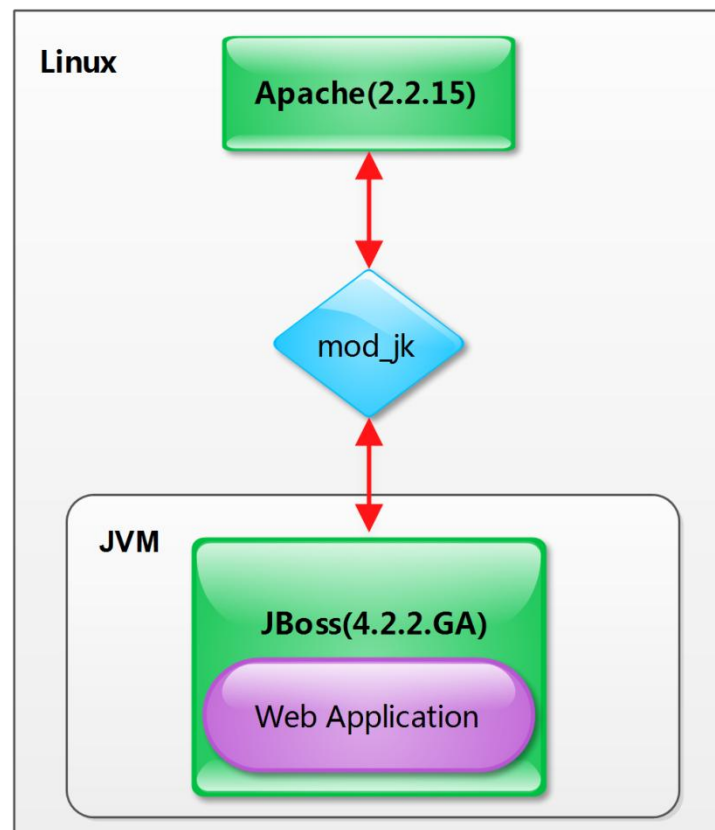




1. 软件升级
2. JVM调优
3. 应用自身优化



- **Apache -- > Nginx**
  - 提升10%+
- **Jboss 4 -- > Tomcat 7**
  - 线程池/ NIO
  - 提升10%-
- **JVM 1.6.0\_23 -- > 1.6.0\_32**
- **OS**
  - 32bit -- > 64bit
  - 内核升级 提升40%
- **成本最低 效果明显**





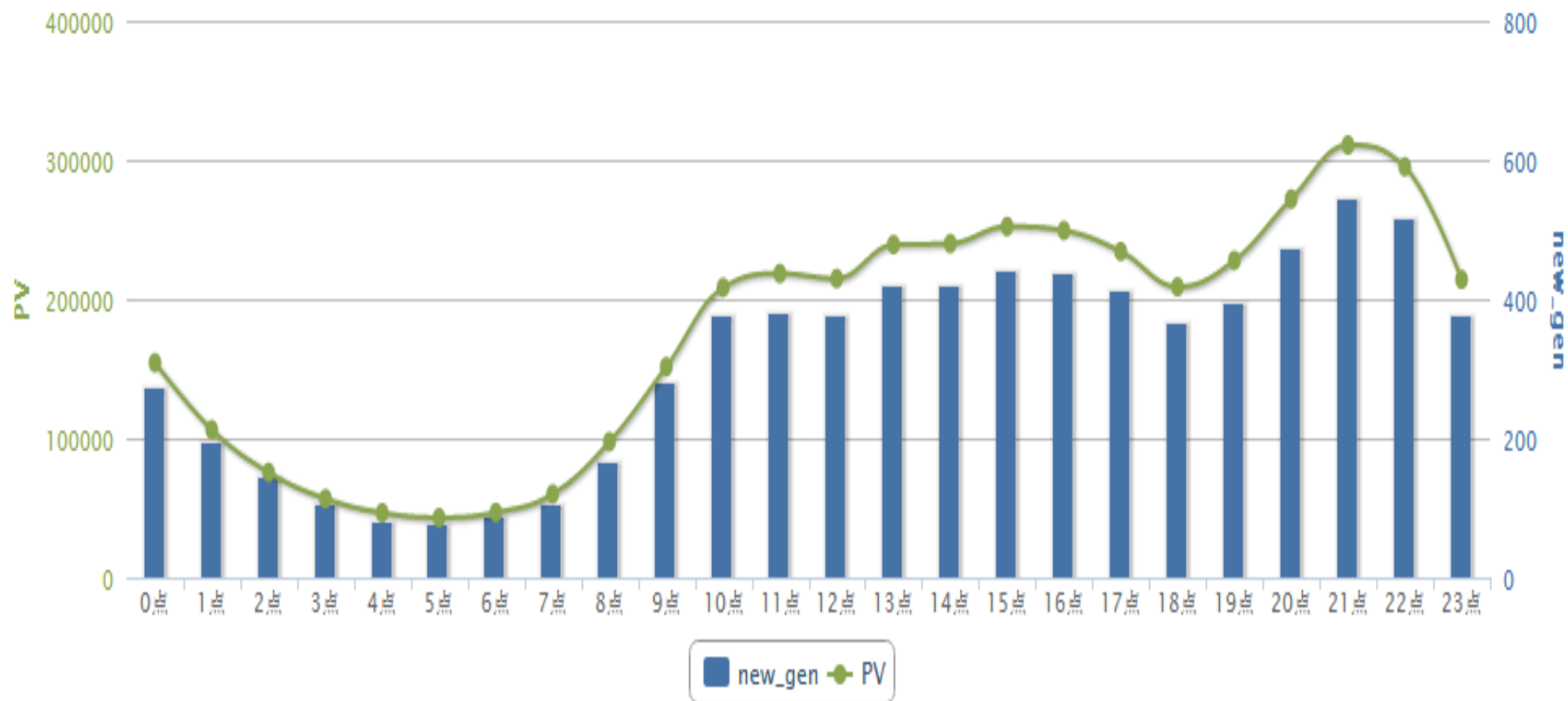
- 减少major GC
- 合理分配堆大小
- 优化相关参数



# PV与GC的关系



新生代垃圾回收与PV



# 吞吐量与GC的关系



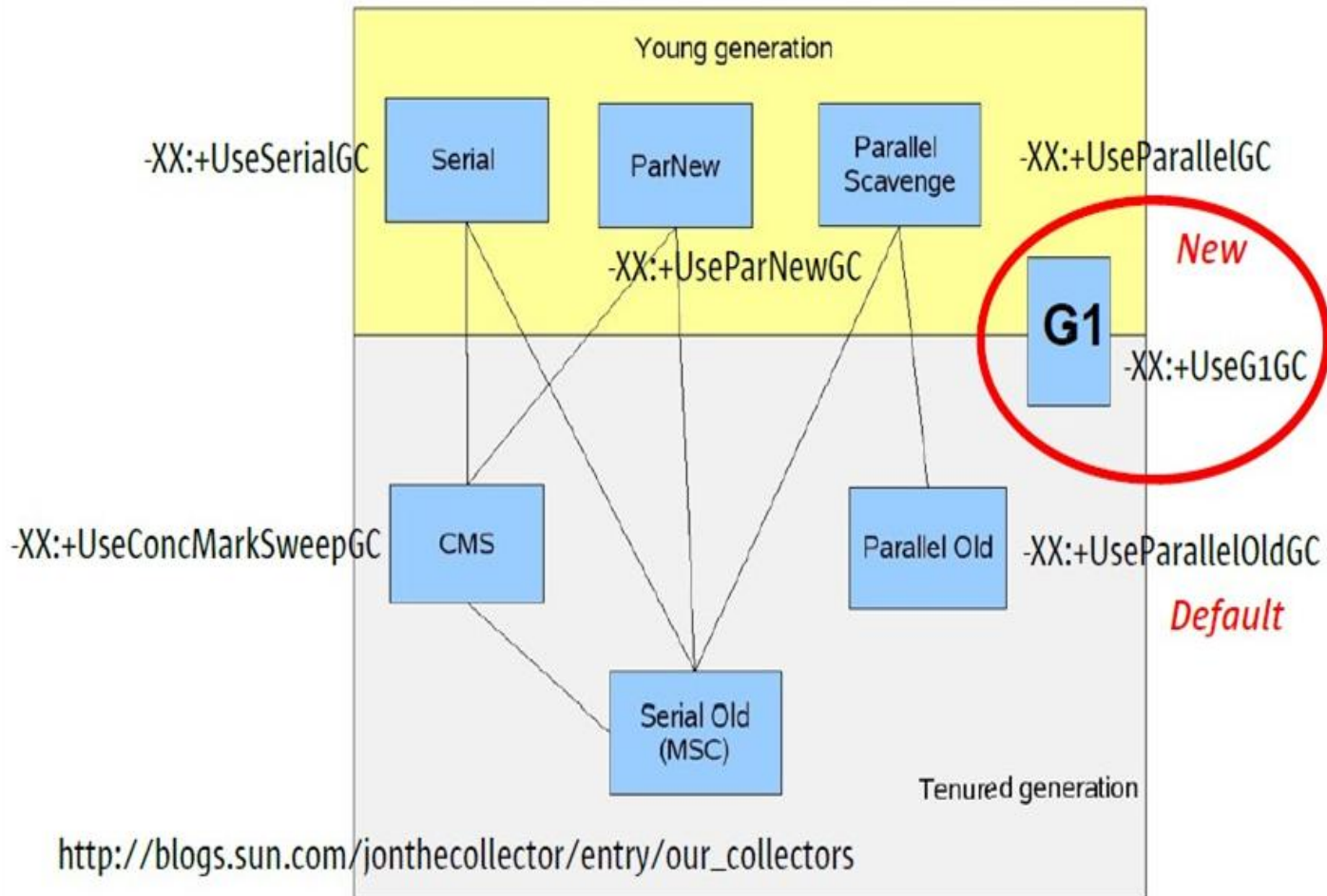
NewSize(M)	QPS(3M)	QPS(4M)	QPS(5M)	QPS(6M)	QPS(7M)	GC(count)	GC(real)
78	767	509	381	292	240	13590	137.19
128	881	634	456	382	314	8098	94.2
178	943	704	548	427	350	5533	60.77
228	951	739	576	469	385	4219	37.72
278	983	757	601	490	409	3316	28.62
328	980	767	609	507	427	2734	23.24
378	953	769	610	513	436	2306	19.28
428	996	766	618	521	449	2002	16.8
478	973	757	629	528	450	1756	15.78
528	992	765	633	531	453	1576	13.23
578	1006	786	624	533	466	1431	11.66
628	977	776	634	535	461	1304	11.03
678	1003	775	633	544	462	1202	10.2
728	1001	783	638	534	465	1115	9.19
778	999	792	643	531	474	1037	8.68
828	1014	787	642	538	472	971	8.18
878	1014	782	640	545	471	913	7.59
928	1006	790	645	542	473	859	7.12
978	998	793	643	535	466	812	6.76



# 选择垃圾收集器



## Sun HotSpot Garbage Collectors in Java SE 6

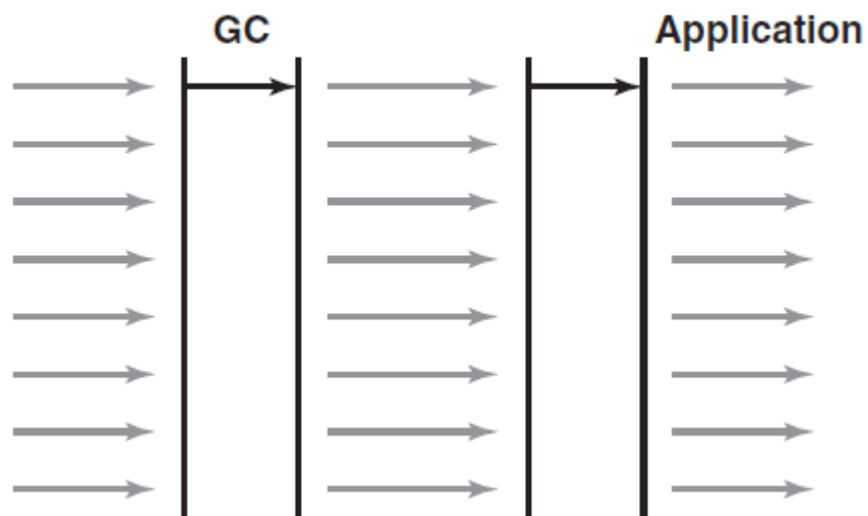




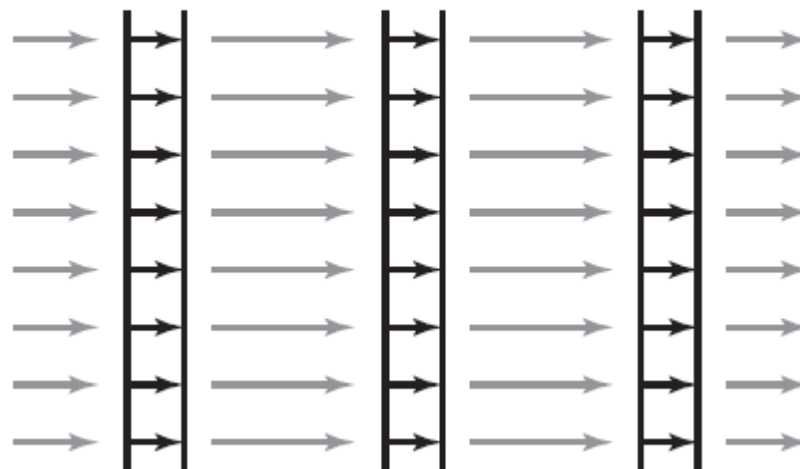
# GC Pause



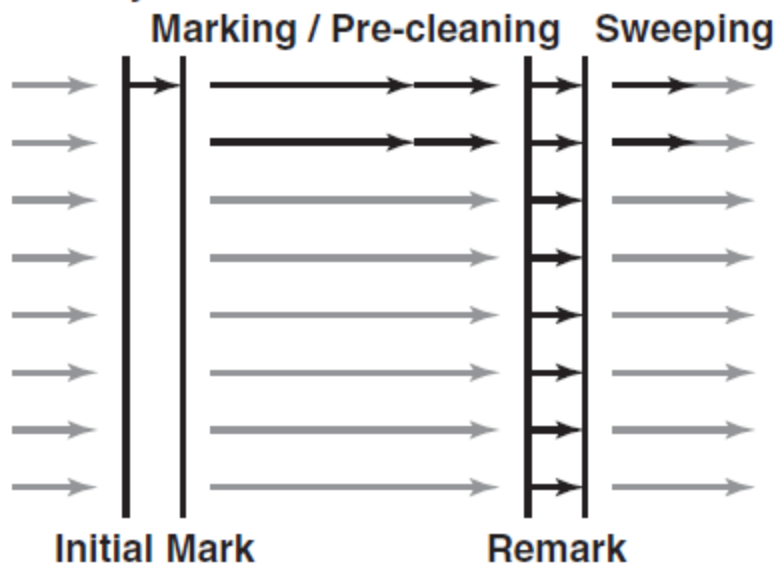
Serial GC



Parallel GC

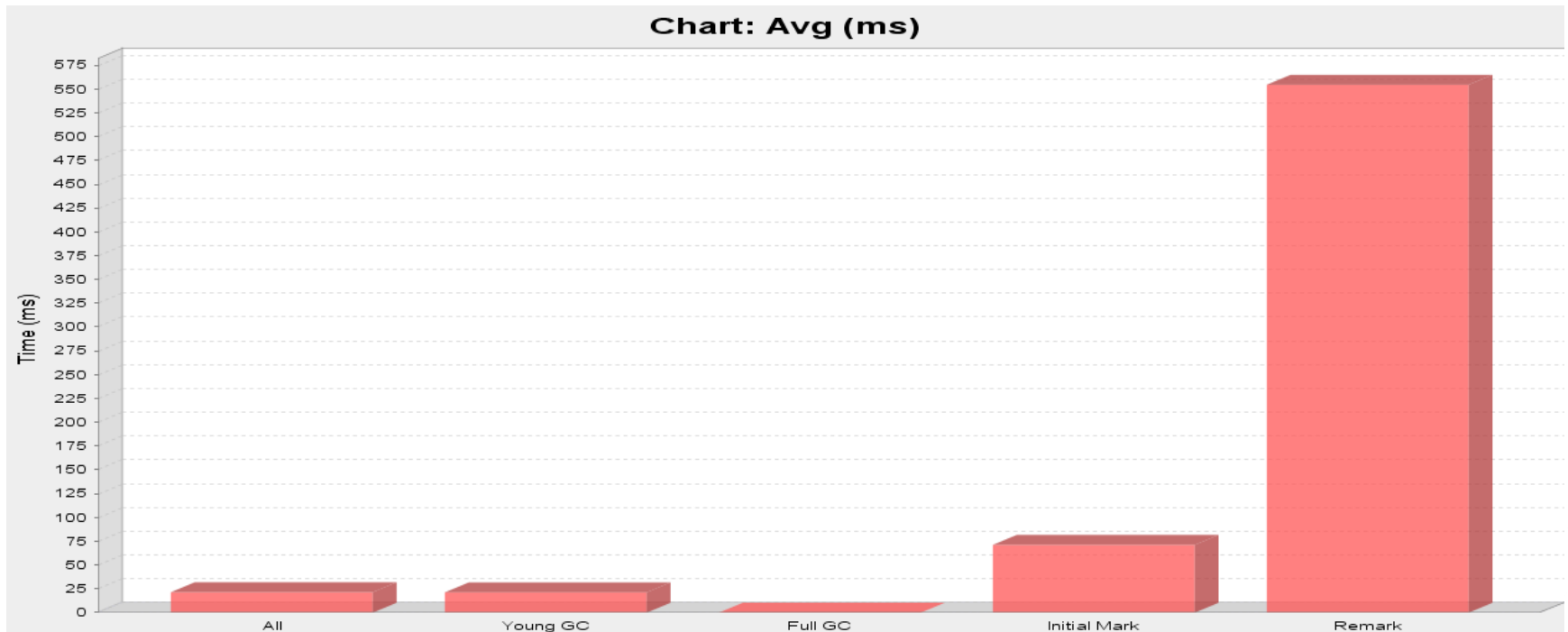


Mostly-Concurrent GC





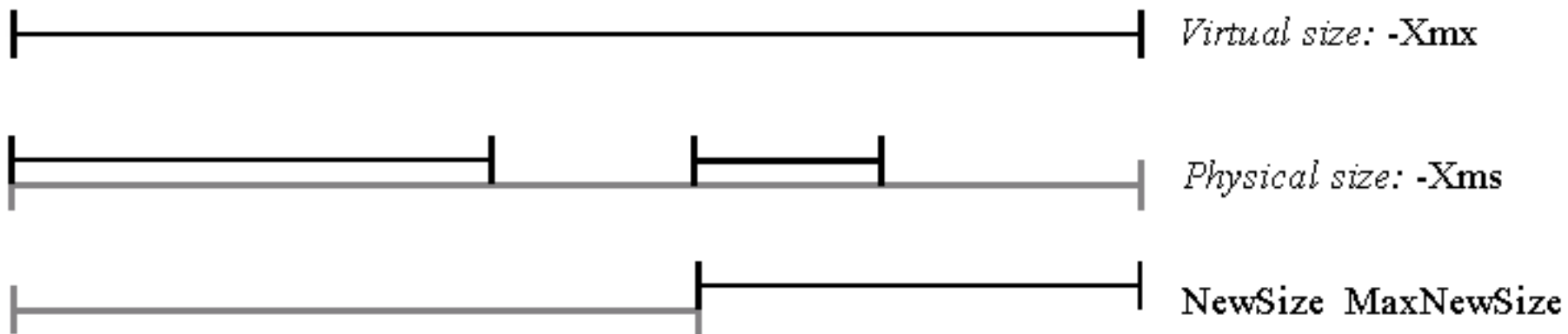
# GC Pause Time



You should try to maximize the number of objects reclaimed in the young generation

----Tony

# JVM堆结构



MaxPermSize

SurvivorRatio





- 减少young晋升到old的对象

- 合理设置各个分区大小
- Survivor Space Size
- Tenuring Threshold
- 优化系统代码
  - 及时回收对象
  - 减少内存使用
  - 减小页面大小

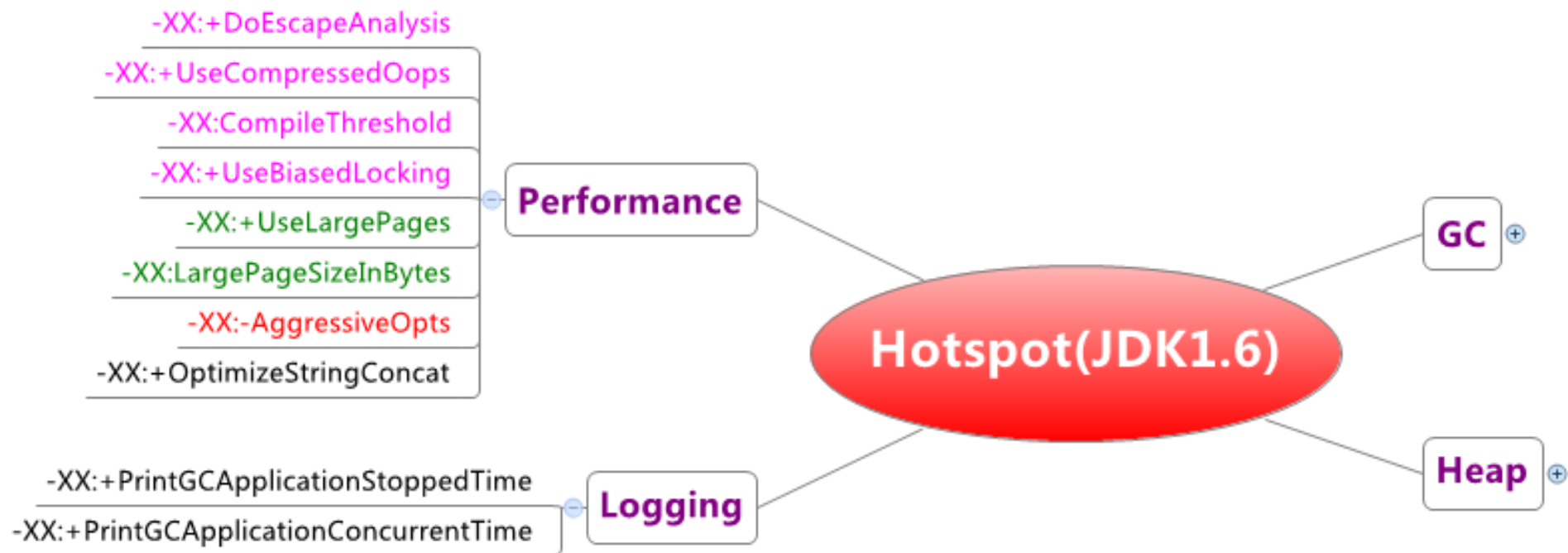




- **线上现状**
  - 线上参数扫描
- **新生代占堆的30%-50%**
- **新生代500M-->2560M QPS提升70%**



# 优化相关参数





- 优化Velocity
- 压缩模版大小
- 设置最佳并发数
- 代码瓶颈优化
- 二方包优化
- 外部调用
- 其他优化





- Sketch Char to byte 提升100%
- 解析执行改编译执行 提升10%

页面大小	优化前 QPS	优化前 RT(ms)	优化后的 QPS	优化后 RT(ms)	提升%
47355	319.05	109.7	455.87	76.776	43%
48581	306.85	114.061	445.39	78.582	45%
55735	296.65	117.983	437.46	80.007	47%
63484	193.69	180.698	302.55	115.684	56%
83152	180.88	193.498	236	148.305	30%
92890	170.68	205.064	214.27	163.342	26%
99732	103.64	337.707	161.46	216.77	56%
144292	108.76	321.81	148.18	236.199	36%
67144	148.49	235.714	268.07	130.565	81%
79703	124.51	281.1	243.64	143.657	96%
92537	123.85	282.595	190.8	183.44	54%
127047	117.52	297.829	164.1	213.284	40%
129479	105.36	332.197	155	225.8	47%



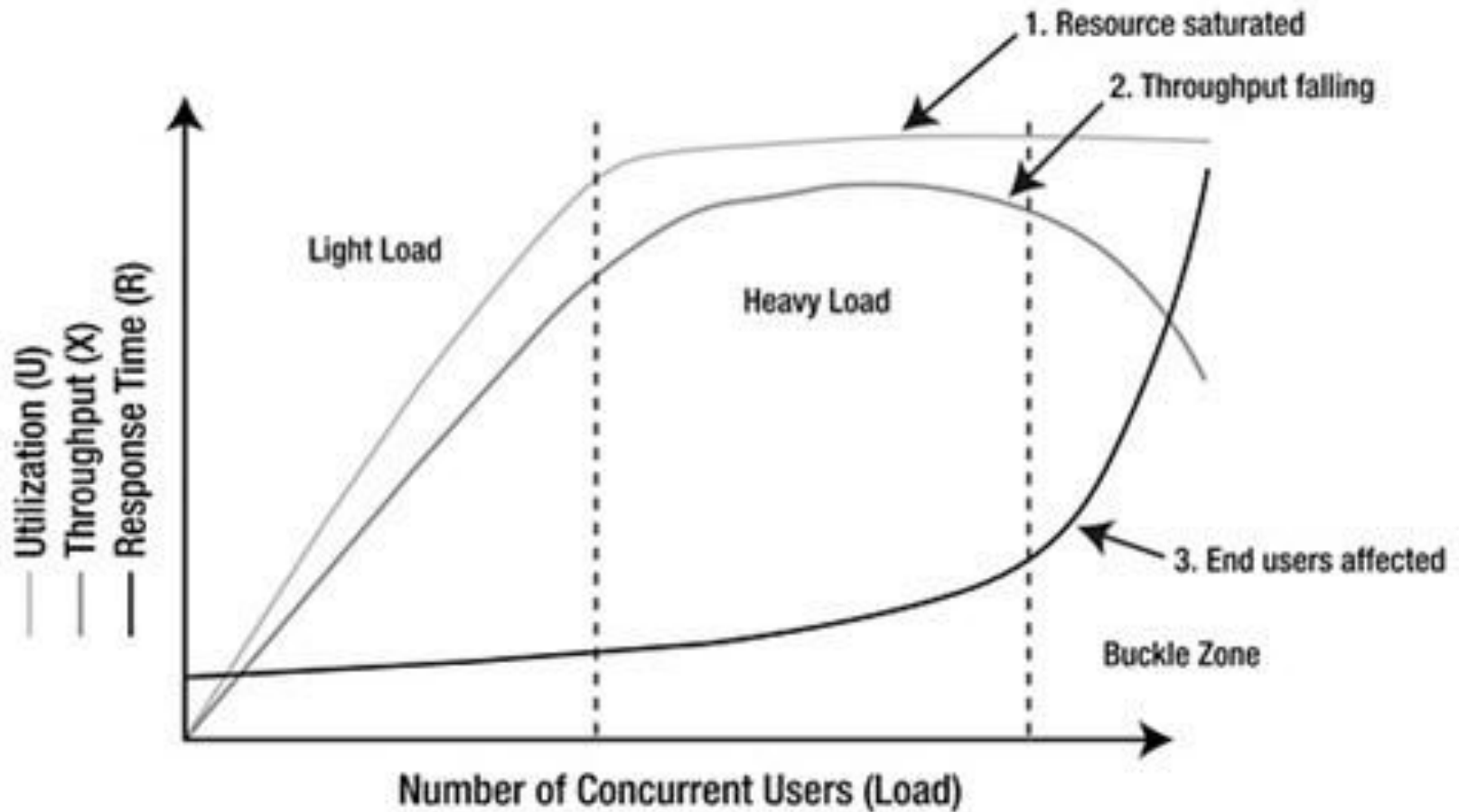


- **删除空行**
- **删除多余空格**
- **长URL压缩**
  - List 减小5%以上
- **用URL别名**
  - List 去掉http:头,减小1%
- **业务上去重**
- **模板大小和QPS成反比**
  - 大小减少10% QPS提升10%





# 并发数与资源消耗





- 从CPU的角度计算：
  - 并发数 =  $((\text{CPU时间} + \text{CPU等待时间}) / \text{CPU时间}) * \text{CPU数量}$
- 从堆内存的角度计算：
  - 并发数 =  $\text{young gc} / ((\text{小GC时间间隔} / \text{rt}) * (\text{request memory}))$



# 代码瓶颈优化



方法信息	执行时间(ms)	执行次数	总时间(ms)
<a href="#">com/taobao/common/tair/comm/TairClient:invoke:143</a>	4	12996	53478
<a href="#">com/taobao/detail/web/module/screen/ItemDetail:processMarket:1176</a>	9	3796	35687
<a href="#">com/taobao/clientSide/web/ThreadLocalCacheFilter:doFilter:49</a>	3	9753	31563
<a href="#">com/taobao/item/domain/protobuf/ItemDOHolder\$PbltemDO\$Builder:mergeFrom:1693</a>	7	3104	21090
<a href="#">com/taobao/session/TaobaoSessionFilter:doFilter:228</a>	1	9852	14406
<a href="#">com/taobao/biz/item/detail/ao/impl/DefaultItemDetailAO:getItemDetailQuery:550</a>	3	4183	10763
<a href="#">com/taobao/remoting/impl/DefaultRespFuture:get:89</a>	11	840	9064
<a href="#">com/taobao/item/serialize/type/ProtoBufSerializer:deserialize:973</a>	2	3487	7298

## • 案例

- 去掉吞异常,CPU使用率提高30%



- **案例**
  - Bean copy 提升20倍以上
    - 可以做一次不做多次
  - IP库 提升1倍
    - 可以提前做的提前做
- **不受业务影响,永久受益**



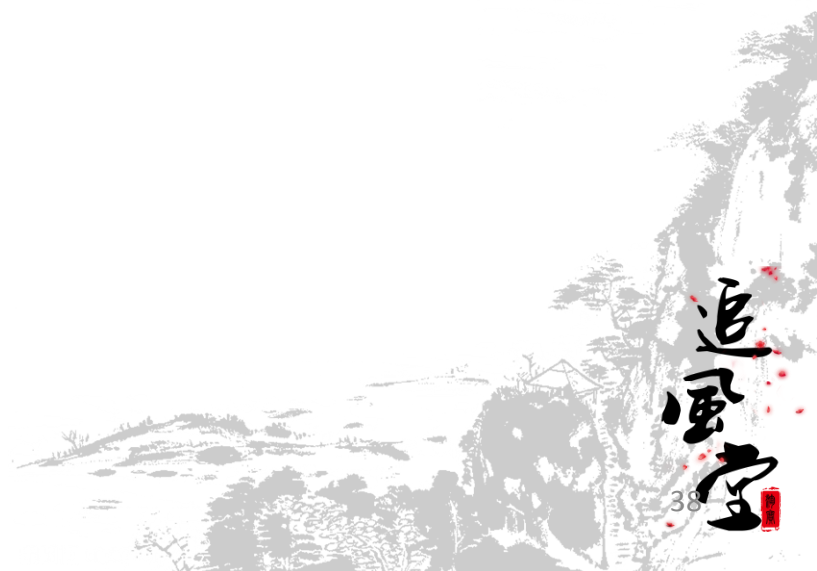


- 并行HSF
- 并行搜索
- 合并外部调用
- 会降低响应时间,不会提升吞吐量
- 减少压缩解压
  - Tair client
- 使用更优的序列化协议
  - Protocol Buffers
  - Kryo 比PB快20%





- **类中Field要排序**
  - 频繁使用的放一起
  - 基本类型引用类型分开放
- **批量处理数组**
  - 按行处理不要按列 可以快20倍
- **使用乐观策略**
- **正确处理异常**





- **动态资源静态化**
- **后台依赖前台化**
  - 降低复杂度
  - 提升性能
- **浏览器渲染页面**
  - 数据远小于页面的场景
- **善用缓存**
  - 整页缓存



# 性能黄金法则



- *80 % -90% of the end-user response time is spent on the frontend*

-- Steve Souders







# 浏览器端分析



1. 基础知识
2. 请求分析
3. 分析工具
4. 性能指标





- **光速**

- 光纤中的光速 200000km/s
- 北京到杭州1120km 来回需要10ms

- **网速**

- 2Mbps(256KBps) 下载50KB页面 需要195ms

- **网卡**

- 服务器1000M网卡
- 虚拟机理论最大QPS =  $1000 / 8 / 3(\text{台}) / 50\text{KB} * 1024 = 850$



- 浏览器并发数限制

<u>Browser</u>	<u>HTTP /1.1</u>	<u>HTTP /1.0</u>
IE 6,7	2	4
IE 8	6	6
Firefox 2	2	8
Firefox 3	6	6
Safari 3,4	4	4
Chrome 1,2	6	?
Chrome 3	4	4
Opera 9.63,10.00alpha	4	4

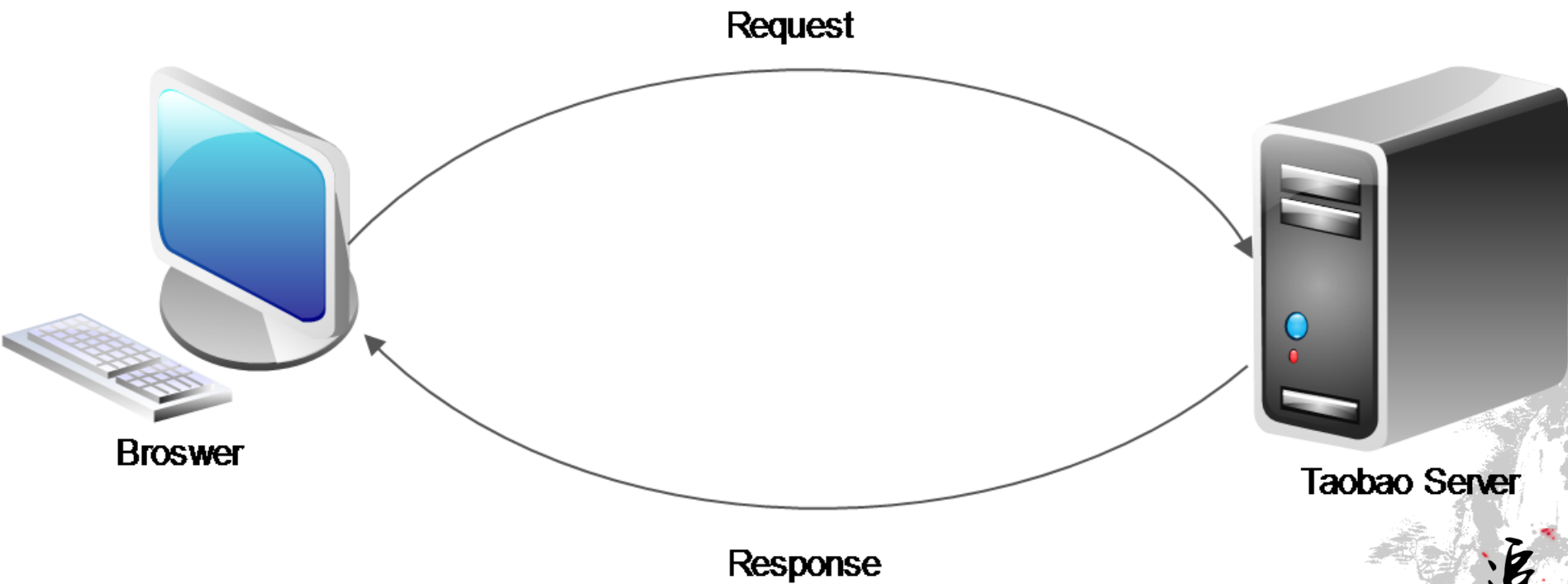


- **请求头**
  - Cache-Control
  - keep-alive
  - User-Agent
  - Cookie
  - If-Modified-Since
- **响应头**
  - Expires
  - Cache-Control
  - Content-Encoding
- **状态码**





## 用户角度



# 页面请求



0 Request start time since the beginning

Request phases start and elapsed time relative to the request start:

0	3ms	DNS Lookup
+3ms	4ms	Connecting
+7ms	0	Sending
+7ms	36ms	Waiting
+43ms	17ms	Receiving

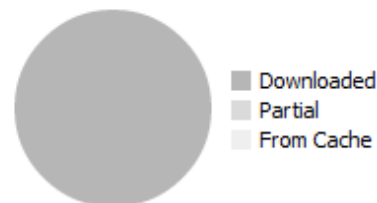
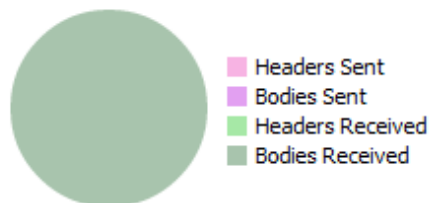
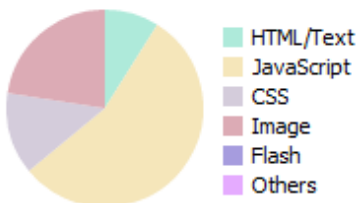
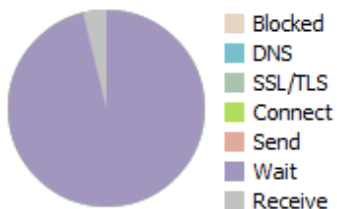
Event timing relative to the request start:

| +858ms DOMContentLoaded

# 页面请求



## 实际情况



GET 1.0??mods/c/head/head-n	200 OK	58.8 KB	30ms
GET item.htm?id=4358168297	200 OK	45.2 KB	54ms
GET 1.gif?cache=6079843&pre-	200 OK	49 B	19ms
GET a.tbcdn.cn	200 OK	99.7 KB	20ms
GET sea.js	200 OK	9.3 KB	5ms
GET deal_quantity.htm?isStart-	200 OK	73 B	9ms
GET T1UJ1VXiRaXcWeqbX.desc	200 OK	20.9 KB	116ms
GET red.css?1.1.css	200 OK	1.5 KB	16ms
GET trace.js?t=20120329.js	200 OK	463 B	15ms
GET postage_fee.htm?opt=&ca	200 OK	455 B	9ms
GET T1CCRNxmRFXXXXXXXXX.png	200 OK	1 KB	69ms
GET a.tbcdn.cn	200 OK	70.2 KB	114ms
GET ac.js	200 OK	944 B	69ms
GET atp.js	200 OK	11.5 KB	68ms
GET T17miCXo0dXXXXXXXXX-14-1	200 OK	155 B	9ms
GET T1AW4cXdpnqABXXXXX.jpg	200 OK	765 B	35ms
GET T1JLtrXhIAXXXXXXXXX-128-1	200 OK	2.6 KB	52ms
GET T1iLBpXe4cXXXXXXXXX-5-151	200 OK	413 B	45ms
GET T1pvBpXdXcXXXXXXXXX-200-	200 OK	1.2 KB	45ms
GET T1C0BpXclfXXb1upjX.jpg_41	200 OK	678 B	44ms
GET T1a_aeXoNXXXXXXXXX-16-1	200 OK	855 B	33ms
GET T1lcyQXnJaXXXXXXXXX-16-1	200 OK	417 B	64ms
GET T16WJqXaXeXXXXXXXXX-32-	200 OK	3.1 KB	66ms
GET T1IC5aXbXcXXXXXXXXX-152-	200 OK	3.9 KB	68ms
GET T1Fz14XcBXXXXXXXXX-20-4	200 OK	901 B	67ms

<http://hotspot.taobao.net:9999/hotspot/waterfall>







- **Firefox**
  - Firebug
  - YSlow
  - NetExport
- **Chrome**
  - Speed Tracer
  - PageSpeed
- **Fiddler 2**
- **HttpWatch**
- **dynaTrace AJAX Edition**
- **WebPagetest/Aol PageTest**

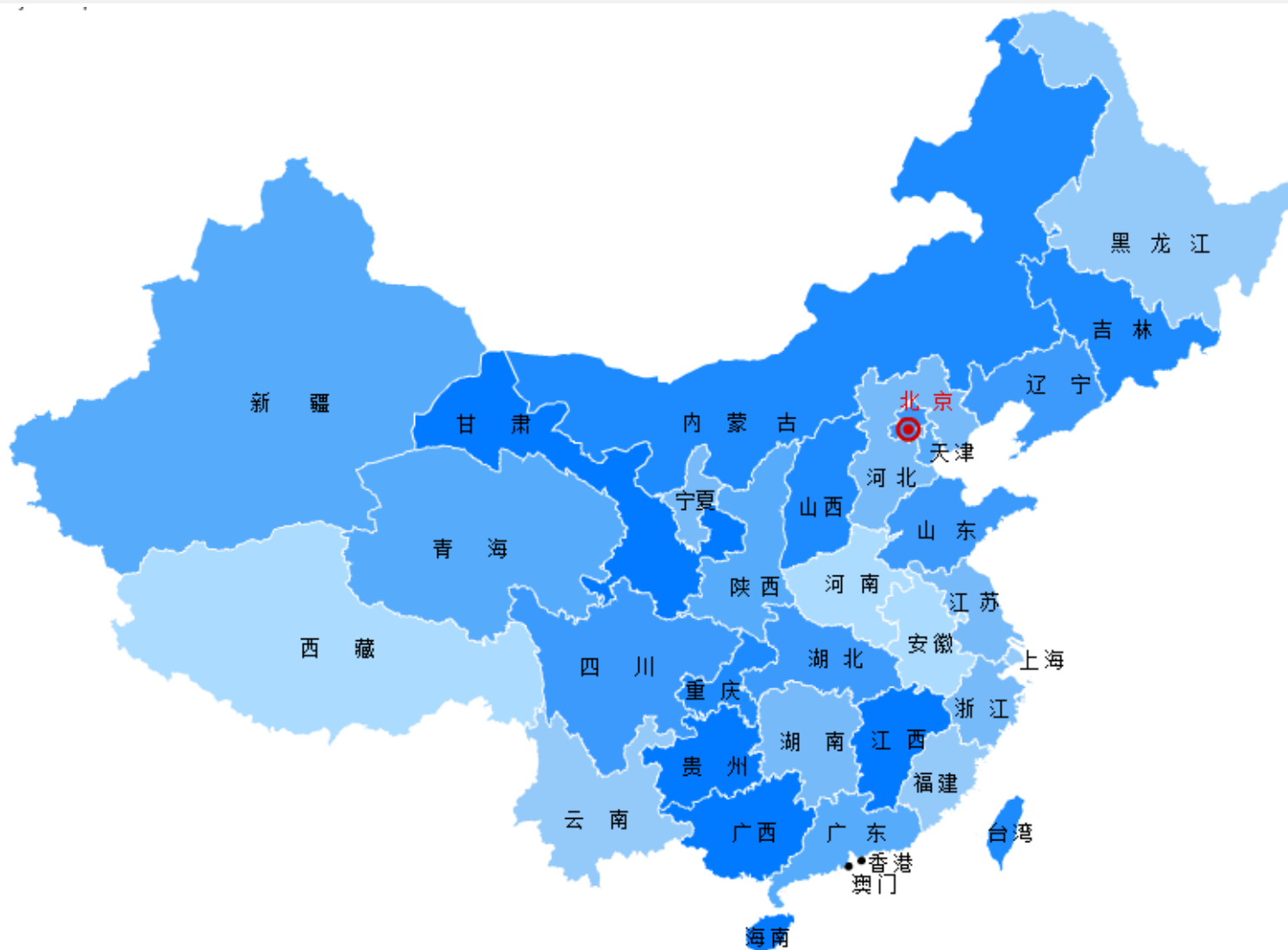




- 开始渲染时间
- Domready
- 可交互时间
- 首屏时间
- onLoad



# 线上监测



Alibench / CommonJS/ Navigation Timing

# 浏览器端优化





1. 减少请求数
2. 减小响应大小
3. 减少DNS查询
4. 浏览器渲染优化





- 缓存资源
- 组合CSS/JS
- CSS Sprites
- 必要时再加载
  - 根据用户行为
- 杜绝404
- 减少重定向





- **减小HTML**
  - 页面越小延迟越小
  - Gzip
  - HTML瘦身
  - 减小Cookie大小
- **减小静态资源**
  - 最小化CSS/JS
  - 优化图片





- 移除空白
- 删除属性引号
- 避免行内样式
- 为JS变量设置别名
  - `document.getElementById -- > $`
- 使用相对的URL





# 使用相对URL



http://www.example.com/path/page.html		
完整URL	相对URL	减小
http://subdomain.example.com	//subdomain.example.com	17.86%
http://www.example.com/path/page2.html	page2.html	73.68%
http://www.example.com/index.html	/index.html	66.67%
http://www.example.com/path2/page.html	../path2/page.html	52.63%
http://www.example.com/path/page.html#f=bar	#f=bar	86.05%
http://www.example.com/path/page.html?q=foo	?q=foo	86.05%

# 减少域名查询



- 数量上平衡
  - 多, DNS查询多
  - 少, 并发受限制
- 有些域名不推荐使用
  - img05~8.taobaocdn.com

域名

出现次数

http://www.taobao.com/

1

http://a.tbcdn.cn/

15

http://img02.taobaocdn.com/

11

http://img03.taobaocdn.com/

8

http://img04.taobaocdn.com/

4

http://img01.taobaocdn.com/

7



- **首屏优先**
  - 减少Dom数量
- **按需加载**
- **次要信息异步化**
  - 多,请求数多
  - 少,页面大
- **避免页面缩放图片**





## 基本信息

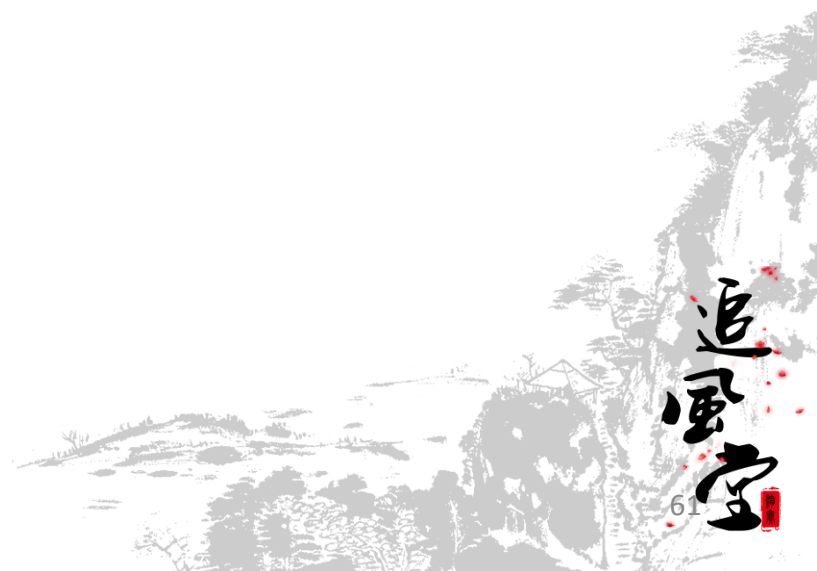
综合评价:C 分数:75

<b>F 控制JS脚本数量</b>	<b>F 控制JS脚本数量</b> [ 45/100 ] <a href="#">帮助</a>
<b>B 控制css样式文件数量</b>	js 21 个, 其中 global x 4 , apps x 4 , other x 3 , kissy x 6 , jsdata x 2 , ad x 2 ,
<b>A 控制css背景数量</b>	4 x <b>global</b> <ul style="list-style-type: none"> <li>▪ <a href="http://a.tbcdn.cn/??s/kissy/1.1.6/kissy-min.js,p/global/1.0/global-min.js,apps/hesper/list/2gest.js,apps/hesper/list/20110922/base.js?...">http://a.tbcdn.cn/??s/kissy/1.1.6/kissy-min.js,p/global/1.0/global-min.js,apps/hesper/list/2gest.js,apps/hesper/list/20110922/base.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/ac.js">http://a.tbcdn.cn/s/ac.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/atp.js">http://a.tbcdn.cn/s/atp.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/p/global/1.0/minicart-min.js?...">http://a.tbcdn.cn/p/global/1.0/minicart-min.js?...</a></li> </ul>
<b>F 控制图片数量</b>	4 x <b>apps</b> <ul style="list-style-type: none"> <li>▪ <a href="http://a.tbcdn.cn/p/market/2011/common_v2.js?...">http://a.tbcdn.cn/p/market/2011/common_v2.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/apps/hesper/list/20110922/p4p.js?...">http://a.tbcdn.cn/apps/hesper/list/20110922/p4p.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/apps/hesper/list/??20110922/list.js,20110922/love-hate.js?...">http://a.tbcdn.cn/apps/hesper/list/??20110922/list.js,20110922/love-hate.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/apps/shop-street/love-jie/mods/public/streettools/love-street-tools.js?...">http://a.tbcdn.cn/apps/shop-street/love-jie/mods/public/streettools/love-street-tools.js?...</a></li> </ul>
<b>A 控制iframe数量</b>	3 x <b>other</b> <ul style="list-style-type: none"> <li>▪ <a href="http://a.tbcdn.cn/tbra/1.0/tbra-aio.js?...">http://a.tbcdn.cn/tbra/1.0/tbra-aio.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/sys/promo/directional_promo.js?...">http://a.tbcdn.cn/sys/promo/directional_promo.js?...</a></li> <li>▪ <a href="http://tmatch.simba.taobao.com/?...">http://tmatch.simba.taobao.com/?...</a></li> </ul>
<b>B 不使用imgextra 的图片</b>	6 x <b>kissy</b> <ul style="list-style-type: none"> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.0.0/build/monitor/monitor-min.js?...">http://a.tbcdn.cn/s/kissy/1.0.0/build/monitor/monitor-min.js?...</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.1.6/suggest/suggest-pkg-min.js">http://a.tbcdn.cn/s/kissy/1.1.6/suggest/suggest-pkg-min.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.1.6/uibase/uibase-pkg-min.js">http://a.tbcdn.cn/s/kissy/1.1.6/uibase/uibase-pkg-min.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.1.6/overlay/overlay-pkg-min.js">http://a.tbcdn.cn/s/kissy/1.1.6/overlay/overlay-pkg-min.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.1.6/datalazyload/datalazyload-pkg-min.js">http://a.tbcdn.cn/s/kissy/1.1.6/datalazyload/datalazyload-pkg-min.js</a></li> <li>▪ <a href="http://a.tbcdn.cn/s/kissy/1.1.6/switchable/switchable-pkg-min.js">http://a.tbcdn.cn/s/kissy/1.1.6/switchable/switchable-pkg-min.js</a></li> </ul>
<b>A 避免404</b>	
<b>C 减少域名使用</b>	





- 1、系统的优化程度，取决于你对它的了解程度
- 2、优化的方法有两种，做得更少或者做得更快





- [淘宝开源](#)
- [Java Performance](#)
- [构建高性能Web站点](#)
- [高性能网站建设指南](#)
- [高性能网站建设进阶指南](#)
- [Troubleshooting Guide for Java SE 6 with HotSpot VM](#)
- [Java SE 6 HotSpot Virtual Machine Garbage Collection Tuning](#)
- [Step by Step GC Tuning in the HotSpot Java Virtual Machine](#)



# Q&A



谢谢！



追風堂