



# 前端性能优化和自动化

2009.12.19

# About me

鄢学鹏

三七 & 秦歌

口碑网 (koubei.com) 前端工程师

<http://ued.koubei.com>

<http://dancewithnet.com>

<http://twitter.com/kavenyan>

kavenyan@gmail.com

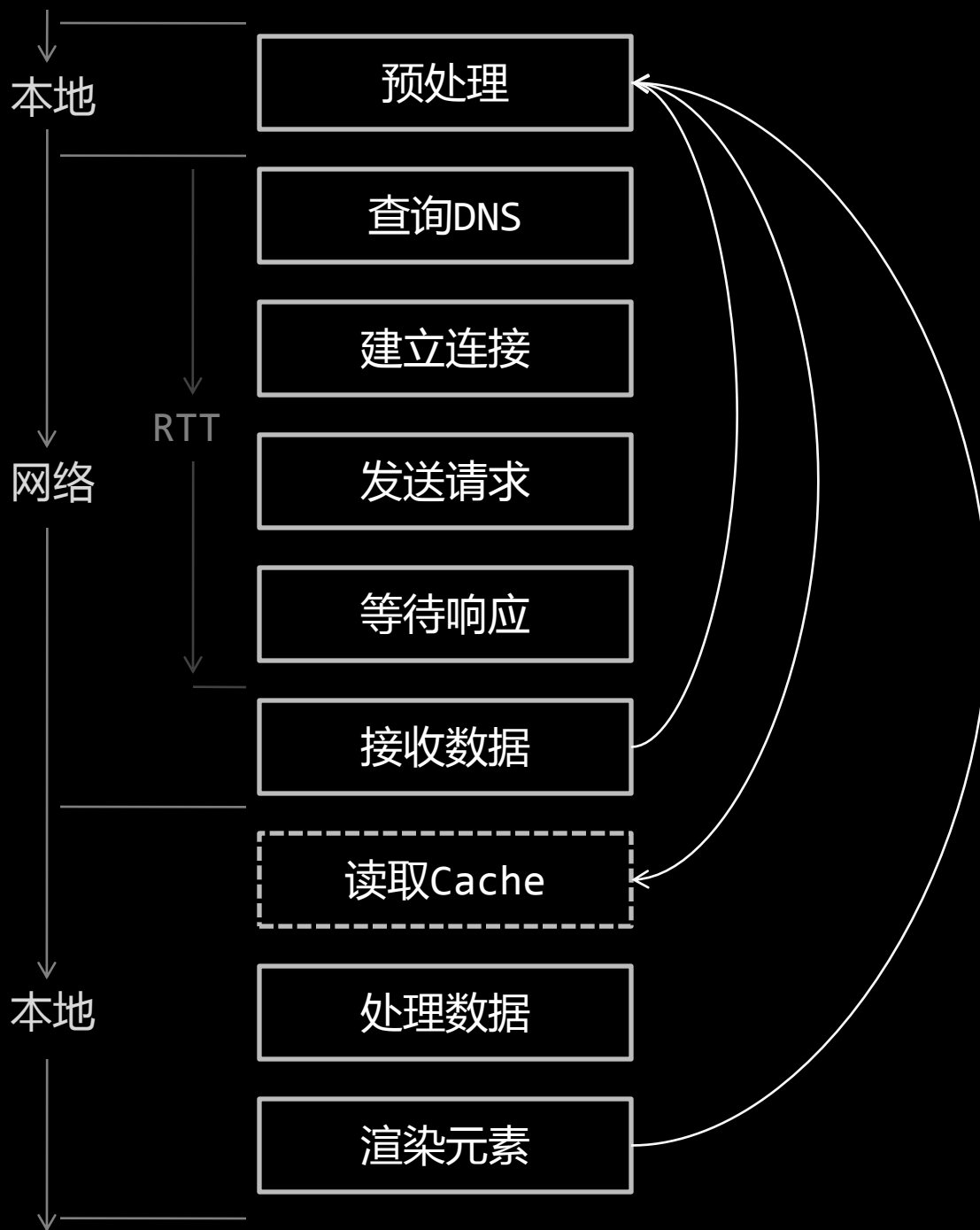
Douglas Crockford:

The page is  
an application with a data connection to a server.

雅虎高性能团队：

> 80%

# 评估 & 最佳实践





预处理

查询DNS

建立连接

发送请求

等待响应

接收数据

读取Cache

处理数据

渲染元素

————→ 浏览器DNS缓存 → 操作系统DNS缓存 → DNS服务器  
2个域名优于1个和4个



预处理

查询DNS

建立连接

————→ 客户端和服务端或代理服务器的TCP连接

发送请求

Connection : keep-alive / 管道

CDN

等待响应

接收数据

读取Cache

处理数据

渲染元素

预处理

查询DNS

建立连接

发送请求

————→ HTTP头 / Cookie / MTU (1500Bytes)  
CDN

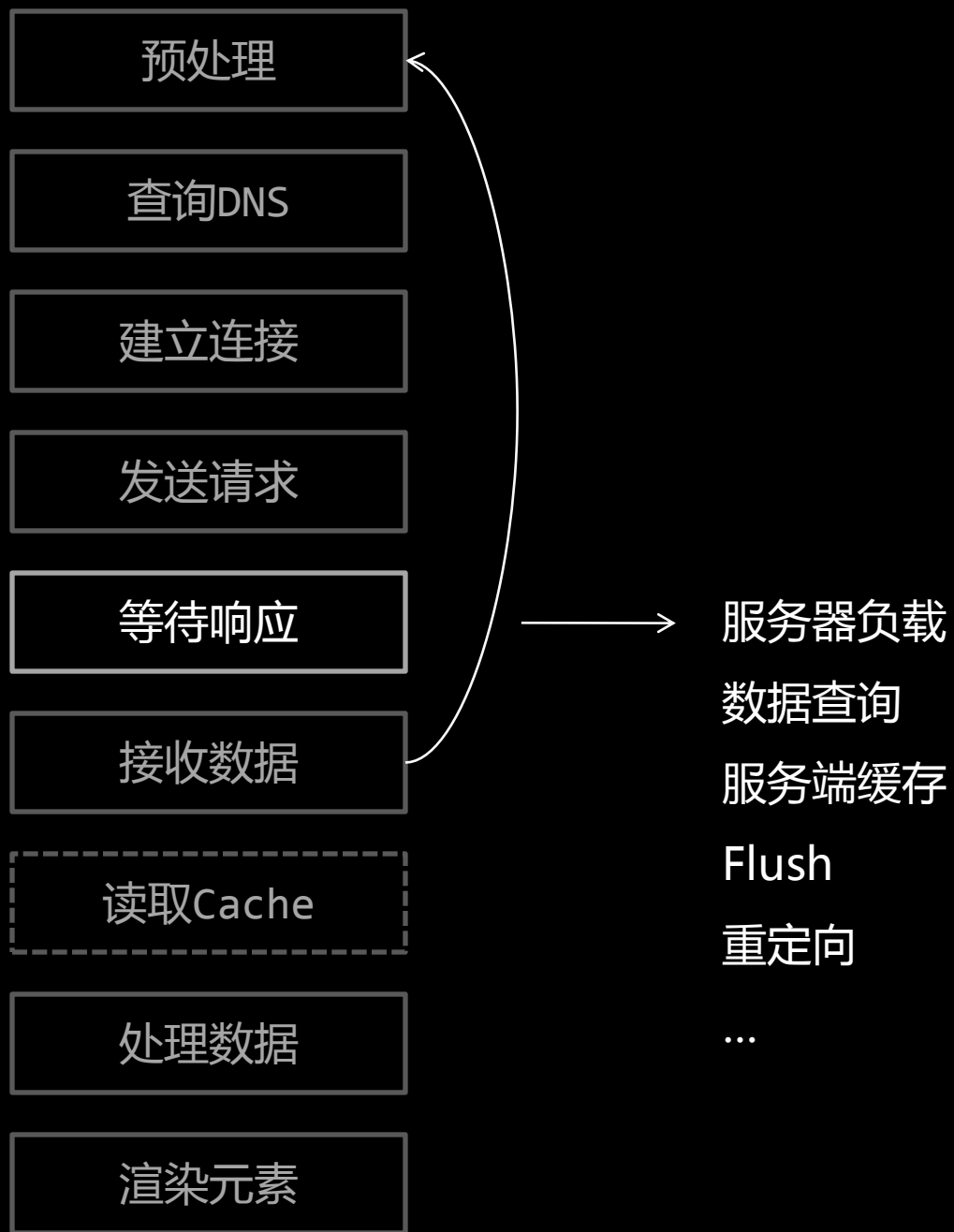
等待响应

接收数据

读取Cache

处理数据

渲染元素



预处理

查询DNS

建立连接

发送请求

等待响应

接收数据

读取Cache

处理数据

渲染元素

带宽

Gzip

HTML、CSS保持一致性（顺序、大小写、引号）

HTML、CSS、JS压缩

移除无效的CSS

IMG优化

favicon.ico小且可缓存

精简HTML

减少DOM元素

HTML5规则

<base target="\_blank">

//koubei.com

去除结束标签

预处理

查询DNS

建立连接

发送请求

等待响应

接收数据

读取Cache

处理数据

渲染元素



Expires & Cache-Control: max-age

Last-Modified & ETag

用外部的CSS和JavaScript

预处理

查询DNS

建立连接

发送请求

等待响应

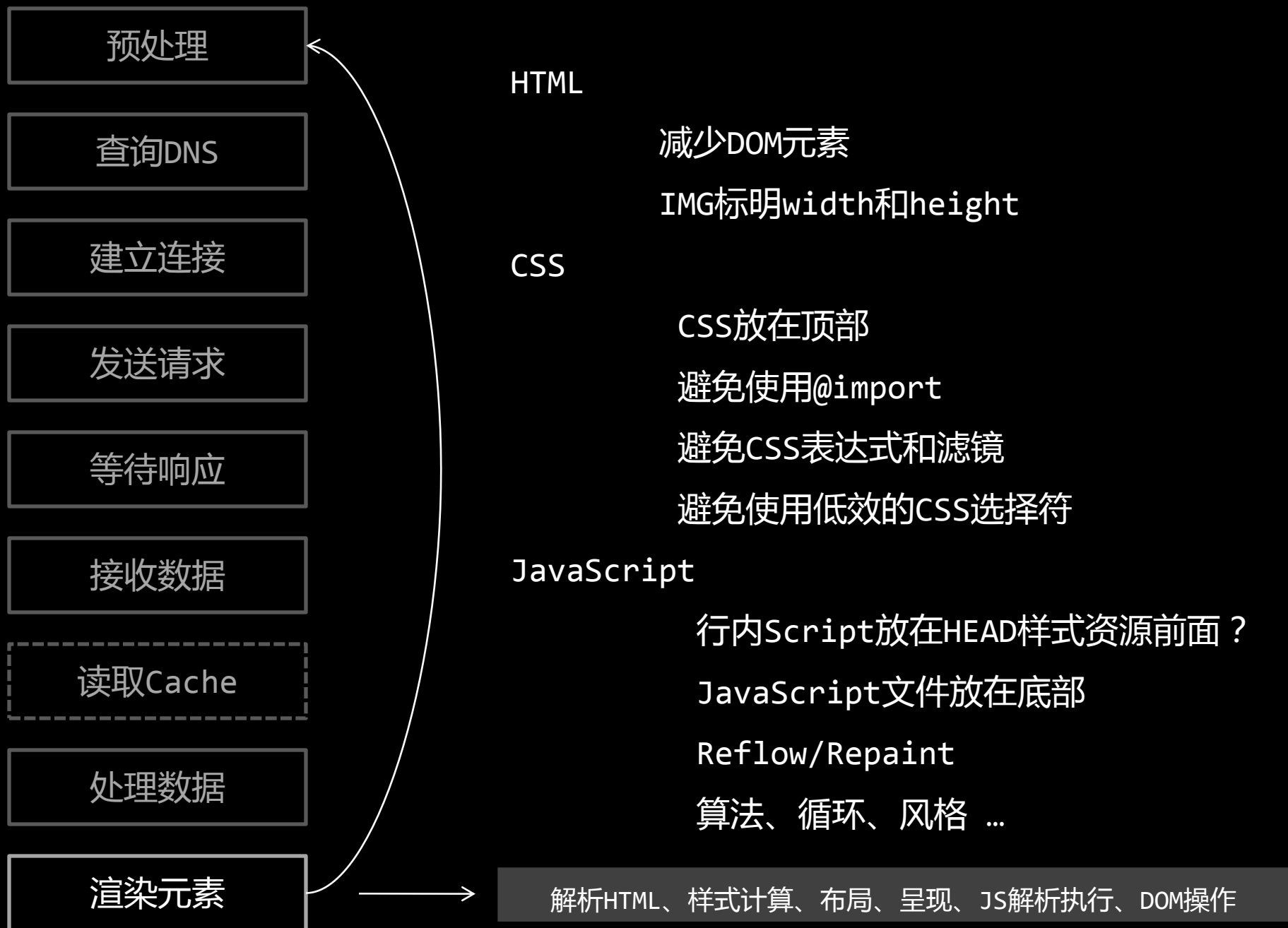
接收数据

读取Cache

处理数据

渲染元素

————→ 不要对IMG / PDF等二进制文件进行 Gzip压缩



# 减少HTTP请求数 !important

CSS、JS文件合并

Ajax GET 取代 POST

CSS Sprites / 图片地图 / DataURI

缓存



## YAHOO Exceptional Performance:

<http://developer.yahoo.com/performance/>

## Google Let's make the web faster:

<http://code.google.com/speed/>

## YAHOO ySlow:

<http://developer.yahoo.com/yslow/>

## Google Page Speed:

<http://code.google.com/speed/page-speed/>

## Google Speed Tracer:

<http://code.google.com/webtoolkit/speedtracer/>

## Microsoft Fiddler 2

<http://www.fiddler2.com/fiddler2/>

## MySpace Performance Tracker

<http://msfast.myspace.com/>

## AOL Web Page Test

<http://www.webpagetest.org/>

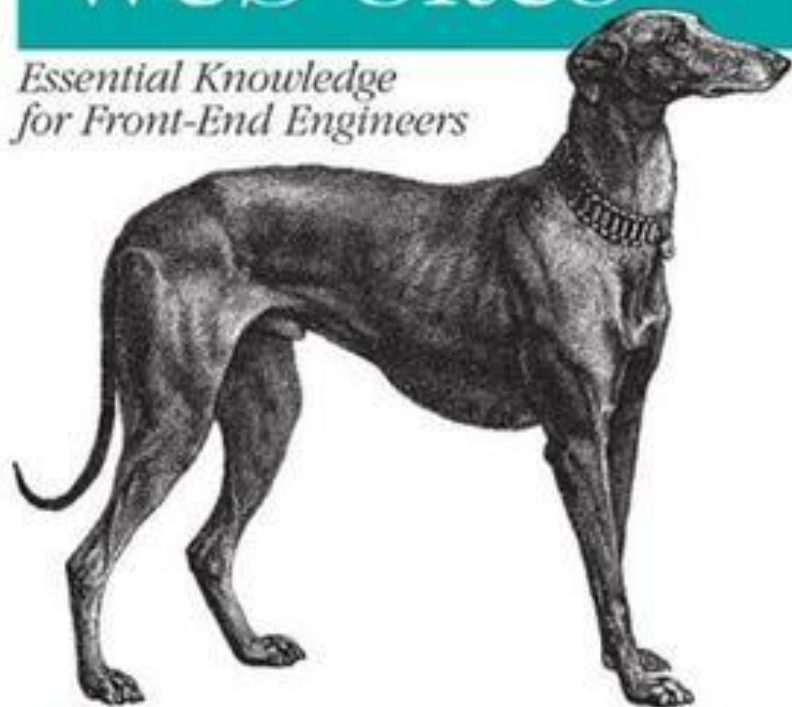
## More Tools:

<http://code.google.com/speed/downloads.html>

*14 Steps to Faster-Loading Web Sites*

# High Performance Web Sites

*Essential Knowledge  
for Front-End Engineers*



O'REILLY®

*Steve Souders  
Foreword by Nate Kowchley*

*Essential Knowledge for Frontend Engineers*



# Even Faster Web Sites

O'REILLY®

*Steve Souders*

口碑的一点实践...

# Mon. 提交测试，工作很忙

## 1. 把Demo上传到SVN

以前：找到SVN目录、右键SVN Commit

## 2. 把Demo发送给PD、ID、VD...

以前：Mail、旺旺、某个临时服务器、自己私人的服务器

## 3. 把CSS、JS、IMG上传到测试服务器

以前：打开Filezilla、找到站点、找到目录、批量上传

现在：

[输入要执行的任务编号]:cp1

请输入过滤关键字[如"jsbase/\*.js"] : base\*.css

...

您要上传

D:/works/workspace/koubei/temp/pkgsglobal/base/2/base.css

到测试服务器吗[y/n]?y

传输 24793 bytes , 上传成功！

# Tue. 开始性能测试

## 1. 压缩CSS和JS文件

以前的以前：少些注释，写成一行

以前：找到CSS、JS文件压缩工具、右键选择压缩

## 2. 压缩图片文件

以前的以前：找到图片PS压缩

以前：上传到Smush.it压缩

## 3. 合并CSS和JavaScript文件

以前的以前：手工合并

以前：利用工具，右键合并

## 4. CSS Sprites

以前的以前：手工合并

以前：SpriteMe.org等

# Tue. 开始性能测试

## 1. 压缩CSS和JS文件

以前的以前：少些注释，写成一行

以前：找到CSS、JS文件压缩工具、右键选择压缩

## 2. 压缩图片文件

以前的以前：找到图片PS压缩

以前：上传到Smush.it压缩

现在：

[输入要执行的任务编号]:cp1

请输入过滤关键字[如"jsbase/\*.js"] : base\*.css

...

您要上传

D:/works/workspace/koubei/temp/pkgs/global/base/2/base.css

到测试服务器吗[y/n]?y

传输 24793 bytes , 上传成功！

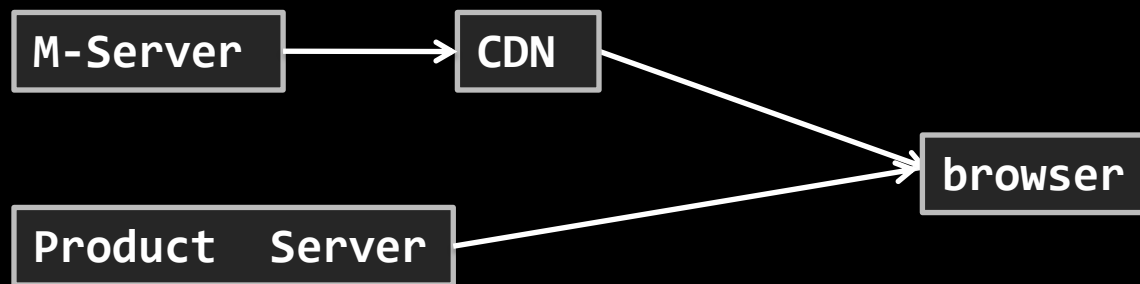
# Tue. 开始性能测试

## 3. 合并CSS和JavaScript文件

以前的以前：手工合并

以前：利用工具，右键合并

[http://k.kbcdn.com/min/f=/global/base/base\\_4\\_2\\_19.css,/product/common/hp/v20091109/asset/hp\\_1\\_1\\_2.css,/product/common/header/v20091109/asset/header\\_1\\_0\\_3.css](http://k.kbcdn.com/min/f=/global/base/base_4_2_19.css,/product/common/hp/v20091109/asset/hp_1_1_2.css,/product/common/header/v20091109/asset/header_1_0_3.css)



# Tue. 开始性能测试

## 4. CSS Sprites

以前的以前：手工合并

以前：SpriteMe.org

未来：自动化合成 & DataURI

CSS Sprites → IMG Sprites

<http://hangzhou.koubei.com/>

[http://k.kbcdn.com/product/common/hp/v20091109/asset/sprite\\_1\\_0\\_8.png](http://k.kbcdn.com/product/common/hp/v20091109/asset/sprite_1_0_8.png)

注意：

1. Logo应该出现在CSS Sprites最前面
2. 标明图片的高和宽



# Wed. 增加新功能了

更新eg.js

1. 新文件： eg\_1.js
2. 时间戳： eg.js?20081217.js

```
/*!  
 * Copyright (c)2009 koubai.com All rights reserved.  
 * @revision:  
 */  
/*  
 * @author: <a href="kavenyan@gmail.com">kaven</a>  
 * @version:4_2_19  
 */
```

# KBCDN

- 独立域名、无Cookie、物理更近
- 10年期限
- 一旦上传不可修改
  - 新文件( hp\_1\_4\_5.js ) vs 时间戳( hp.js?20091209.js )
    - 生效时间与回滚
    - SVN
    - Cache

# Thu. Bug还是来了

回滚eg.js

1. 在SVN中找注释
2. 核对修改时间

[http://k.kbcdn.com/global/base/base\\_4\\_2\\_18.css](http://k.kbcdn.com/global/base/base_4_2_18.css)

```
/*!  
 * Copyright(c)2009 koubei.com All rights reserved.  
 * @revision 4965  
 */
```

# ATT ( Auto Task Tools )

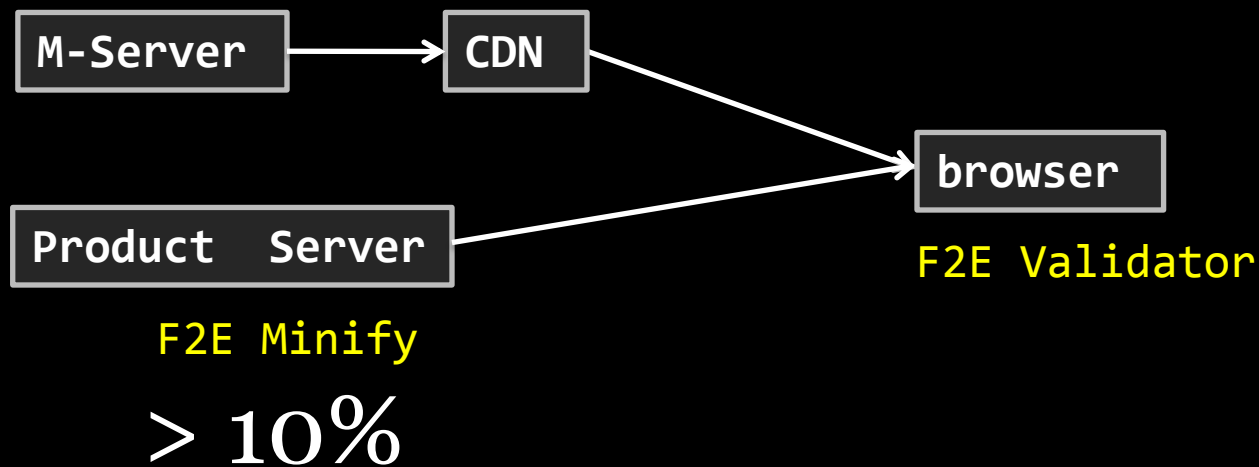
- SVN 操作
- 检测 : Jslint
- 打包 : 文件名、SVN版本号
- 压缩 : YUI Compressor & 图片无损压缩
- 文档 : JS Doc
- 上传 : Test CDN & CDN & 文件同步
- 命令组合

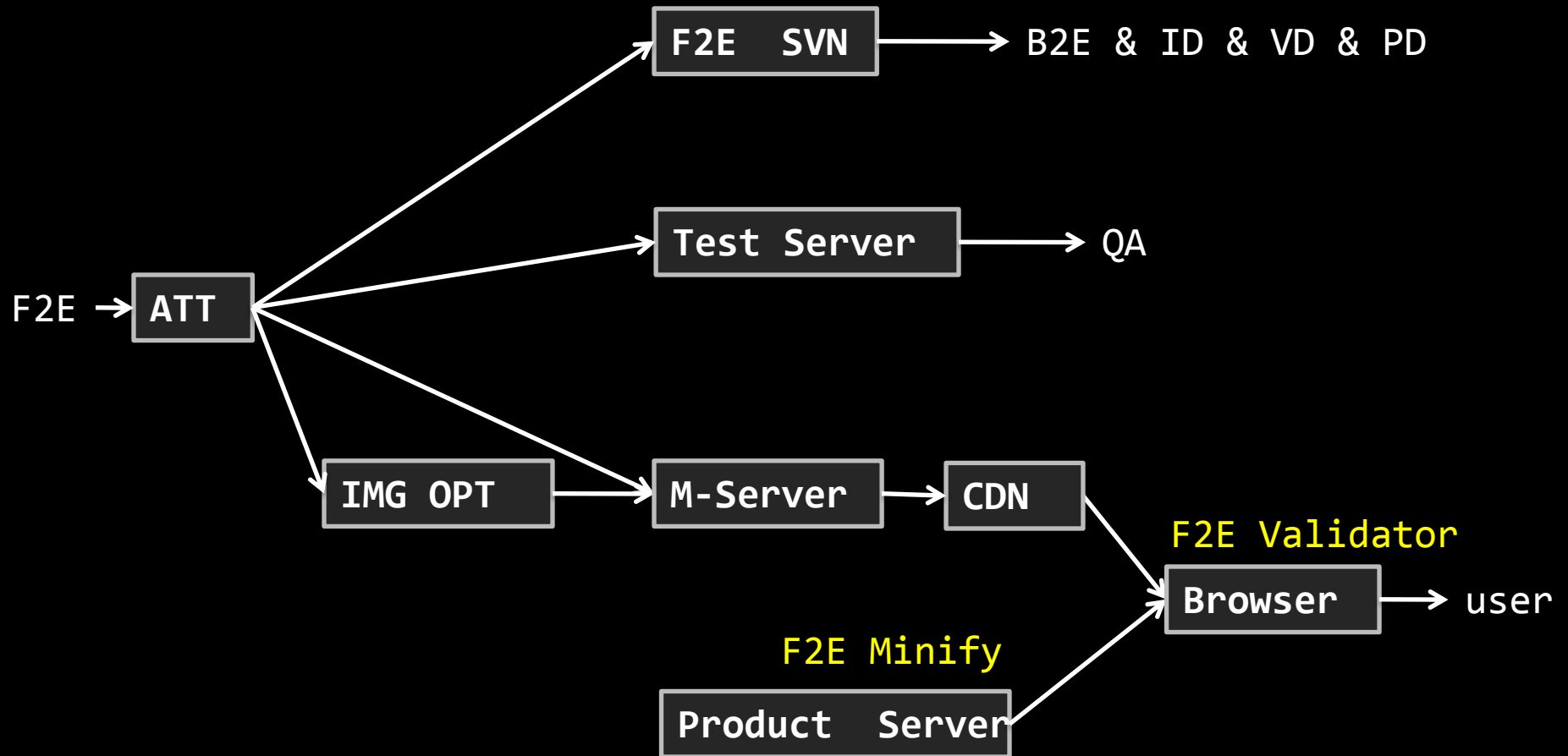
Fri. 杯具啊，又见低级错误



# Fri. 杯具啊，又见低级错误

1. IE6又出现“无法打开该地址，浏览器停止服务”错误提示，查了几次小时发现竟然是某个标签未闭合
2. DTD出现2次， eg.js?20091216.js出现3次
3. CSS中竟然出现了Sprites\_1\_0\_0.png和Sprites\_1\_0\_1.png
4. 面积为80\*80的图片位竟然是一张640\*480的图
5. 我给写给后端看的大段HTML注释都出现在源码中





效率！10%-30%心情！



Sat. 参加D2

一周干五天，天天不加班

# 人 & 基础机制

预加载 vs 延迟加载

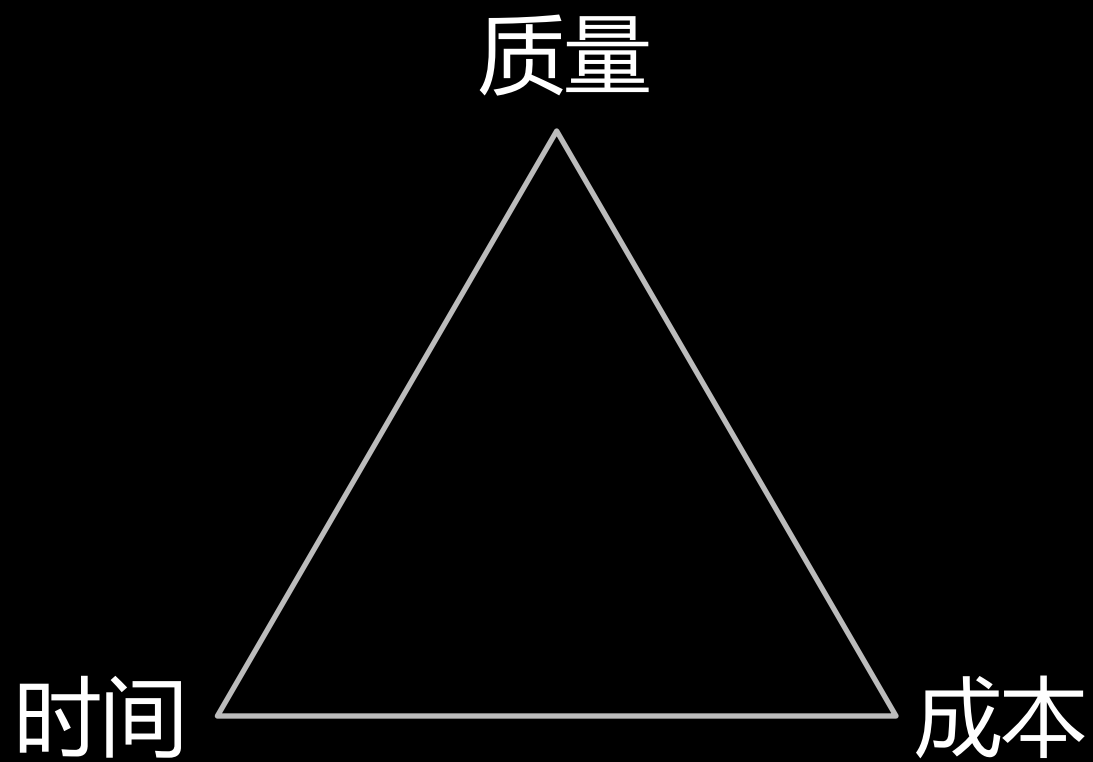
# 时间维度

**0.1s** : 用户操作UI对象的感觉极限

**1s** : 用户自由地操作计算机指令而无需过度等待的感觉极限

**10s** : 用户对其任务保持耐心的极限

# 权衡



- 自动化工具 / 机制
  - 保证基础规则和效率
  - 精确的重复的事情让机器做
- 布道
  - 性能优先成为所有人的共识
- 结果
  - 有可视化(可量化)的被认可的结果

慢500ms = 20% 将放弃访问(Google)

慢400ms = 5-9% 将放弃访问(Yahoo!)

慢100ms = 1% 将放弃交易(Amazon)

我们呢？

高纳德：

97%的情况下，过早的优化是万恶之源！



THANK  
YOU

Q & A