Web 前端性能优化
思路与学习方法

郭碧青

SFDC SegmentFault Developer Conference

个人介绍

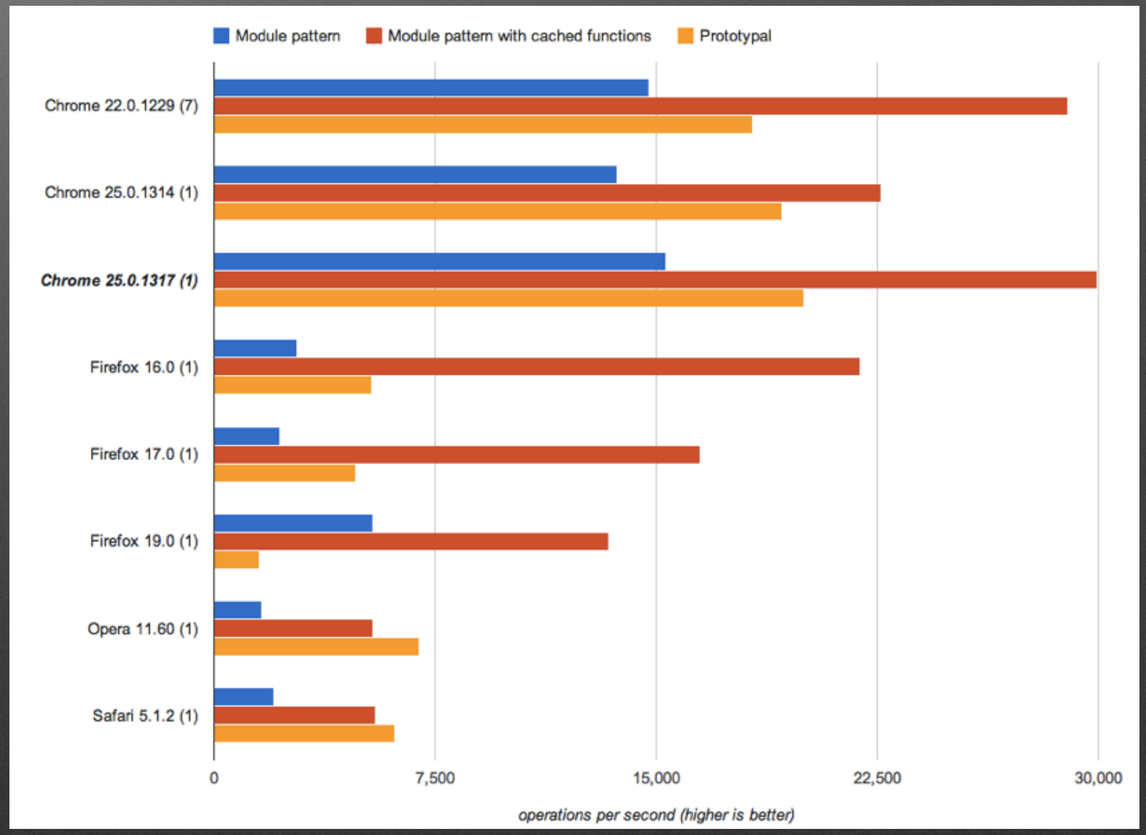
郭碧青 JohnnyGuo

腾讯Web前端高级工程师

参与WebQQ、QQ互联、QQ商家等大型项目研发，对性能优化有大量实践与经验。目前负责手机QQ 兴趣部落的技术架构与研发。

为什么讲性能

- 不像框架成体系
- 代码级到工程级，跨度大
- 特定场景
- 积累 or 飞跃



分享内容

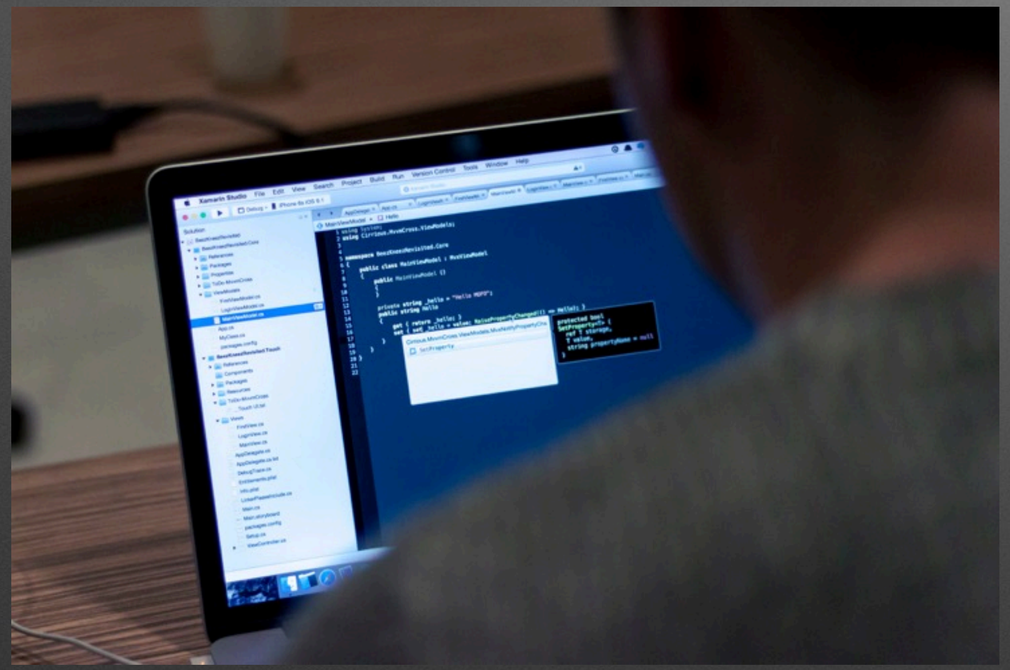
- 前端性能的认识
- 性能分析工具
- 优化思路与学习方法
- 性能优化实践

前端性能的认识

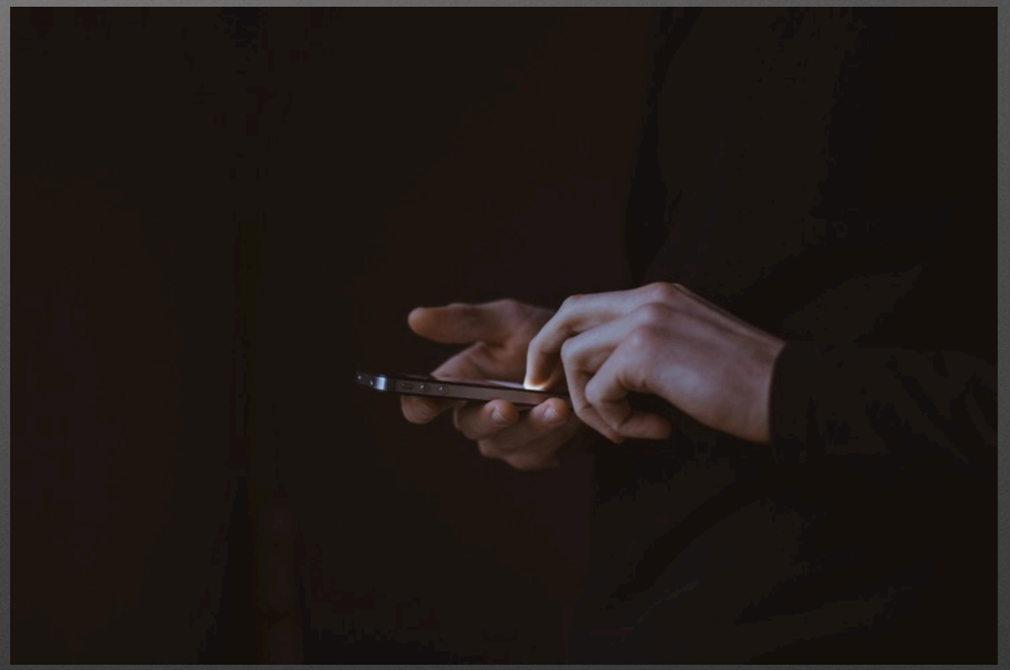
大家眼里的性能

- JS执行效率
- 浏览器渲染性能
- CSS动画性能
- 事件代理
- 雅虎优化建议……

性能视角



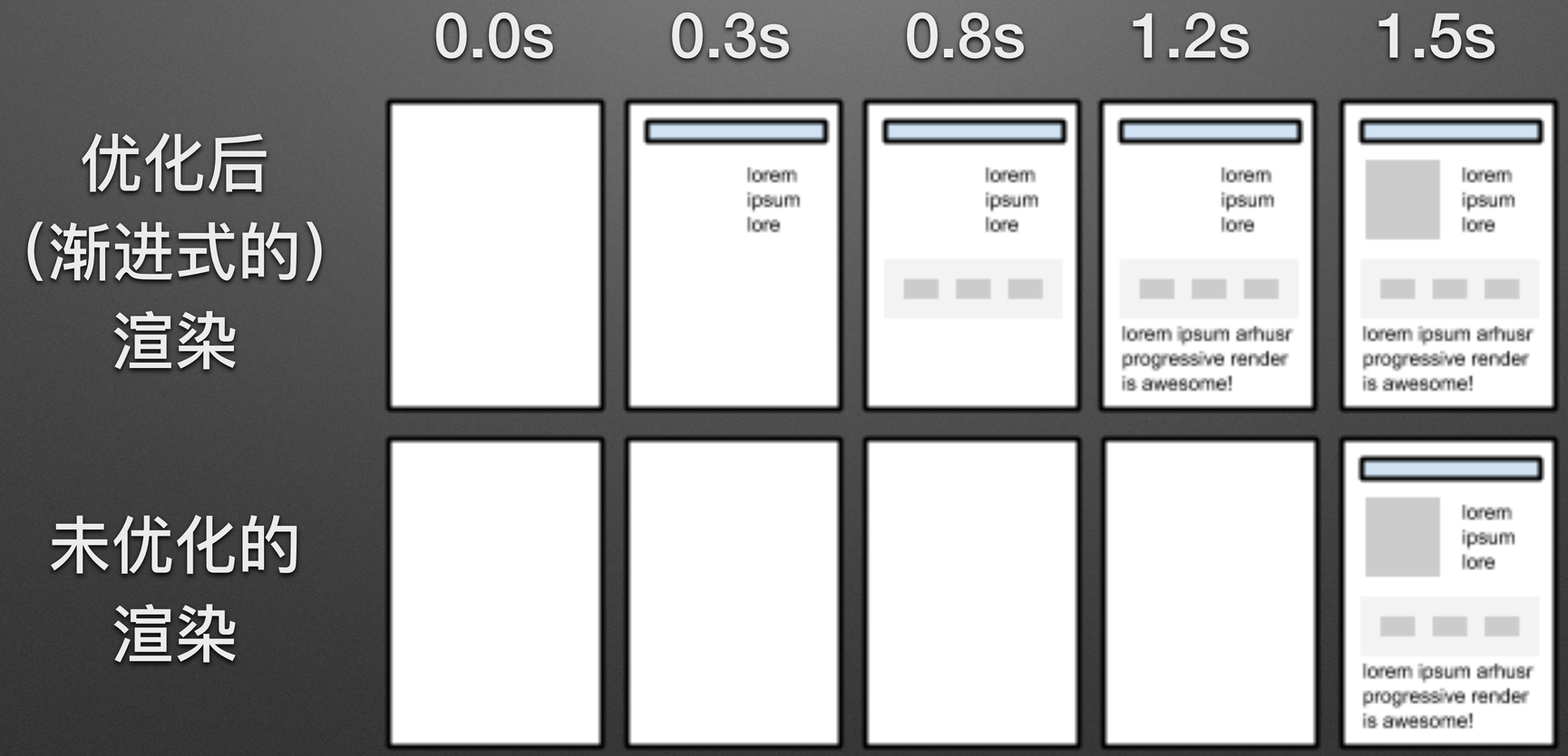
?
=



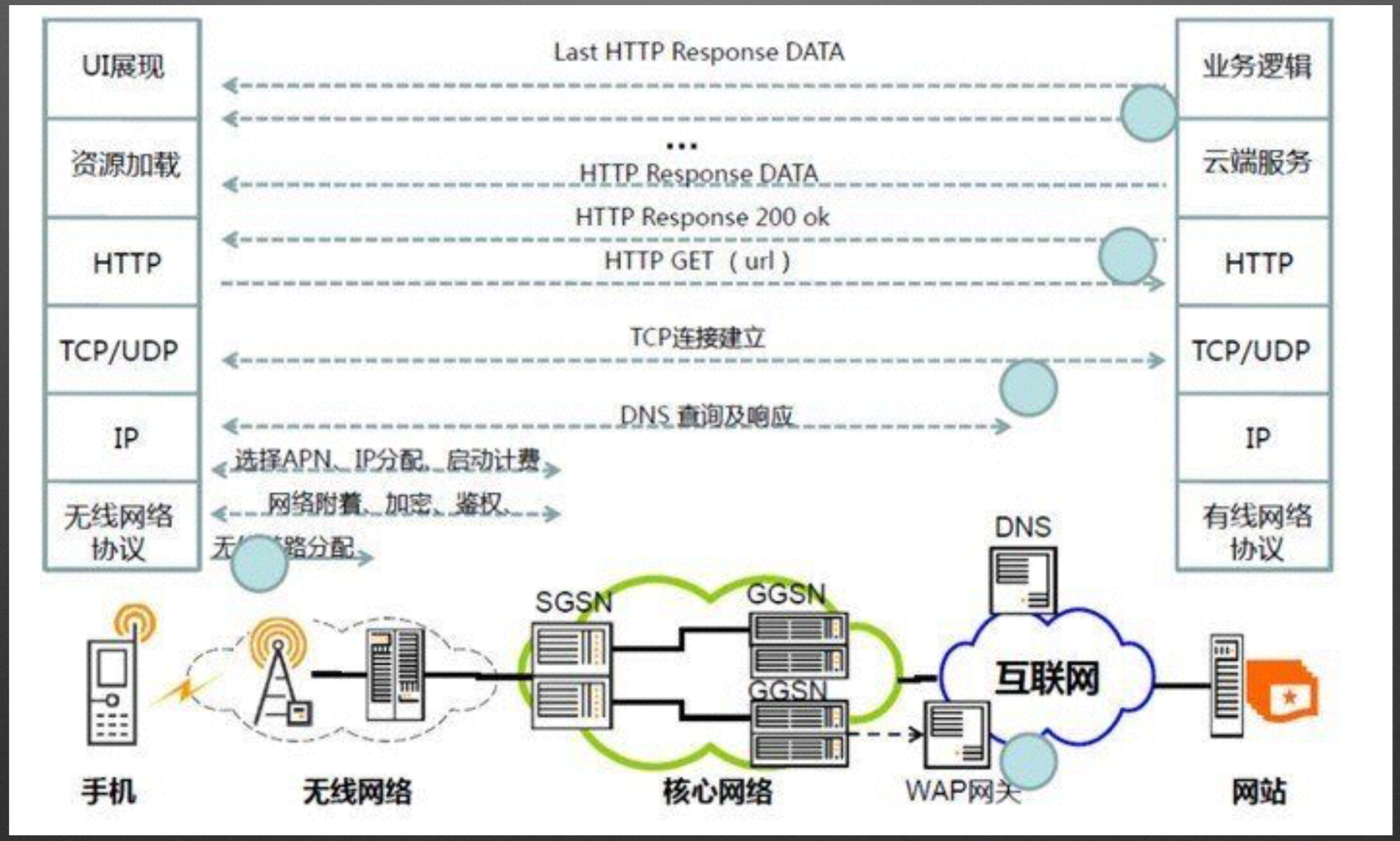
本地运行性能

用户使用性能

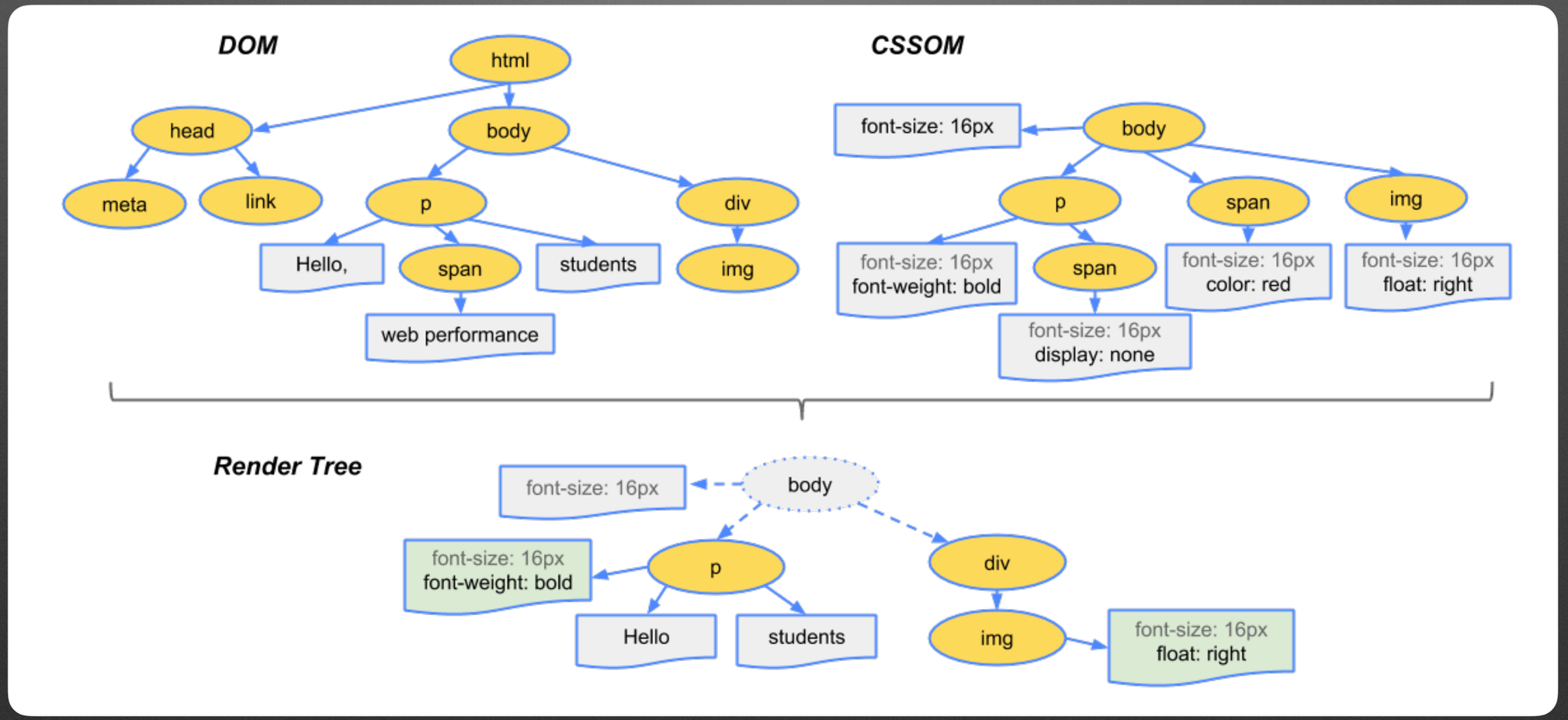
页面加载性能


0.0s
0.3s
0.8s
1.2s
1.5s
优化后（渐进式的）渲染
未优化的渲染
lorem ipsum lore
lorem ipsum arhusr progressive render is awesome!

影响首屏时间因素


UI展现
资源加载
HTTP
TCP/UDP
IP
无线网络协议
Last HTTP Response DATA
...
HTTP Response DATA
HTTP Response 200 ok
HTTP GET（url）
TCP连接建立
DNS 查询及响应
选择APN、IP分配、启动计费
网络附着、加密、鉴权
无线路由分配
业务逻辑
云端服务
HTTP
TCP/UDP
IP
有线网络协议
DNS
SGSN
GGSN
GGSN
互联网
手机
无线网络
核心网络
WAP网关
网站

影响首屏时间因素


DOM
html
head
body
meta
link
p
div
Hello,
span
students
img
web performance
CSSOM
font-size: 16px
body
p
span
img
font-size: 16px
font-weight: bold
span
font-size: 16px
color: red
font-size: 16px
float: right
font-size: 16px
display: none
Render Tree
font-size: 16px
body
font-size: 16px
font-weight: bold
p
div
Hello
students
img
font-size: 16px
float: right

小结

- 前端性能分两部分
  - 网络性能
  - 运行时性能
- 重点关注影响用户体验部分

性能分析工具

PageSpeed

PageSpeed Tools > Insights

指南　参考网页　示例　支持

移动设备　桌面设备

62 / 100 网页加速

! 应当修复：

清除首屏内容中阻止呈现的 JavaScript 和 CSS

▸显示解决问题的方法

! 考虑修复：

按优先级排列可见内容

▸显示解决问题的方法

优化图片

▸显示解决问题的方法

使用浏览器缓存

▸显示解决问题的方法

✓ 已正确实施6条规则

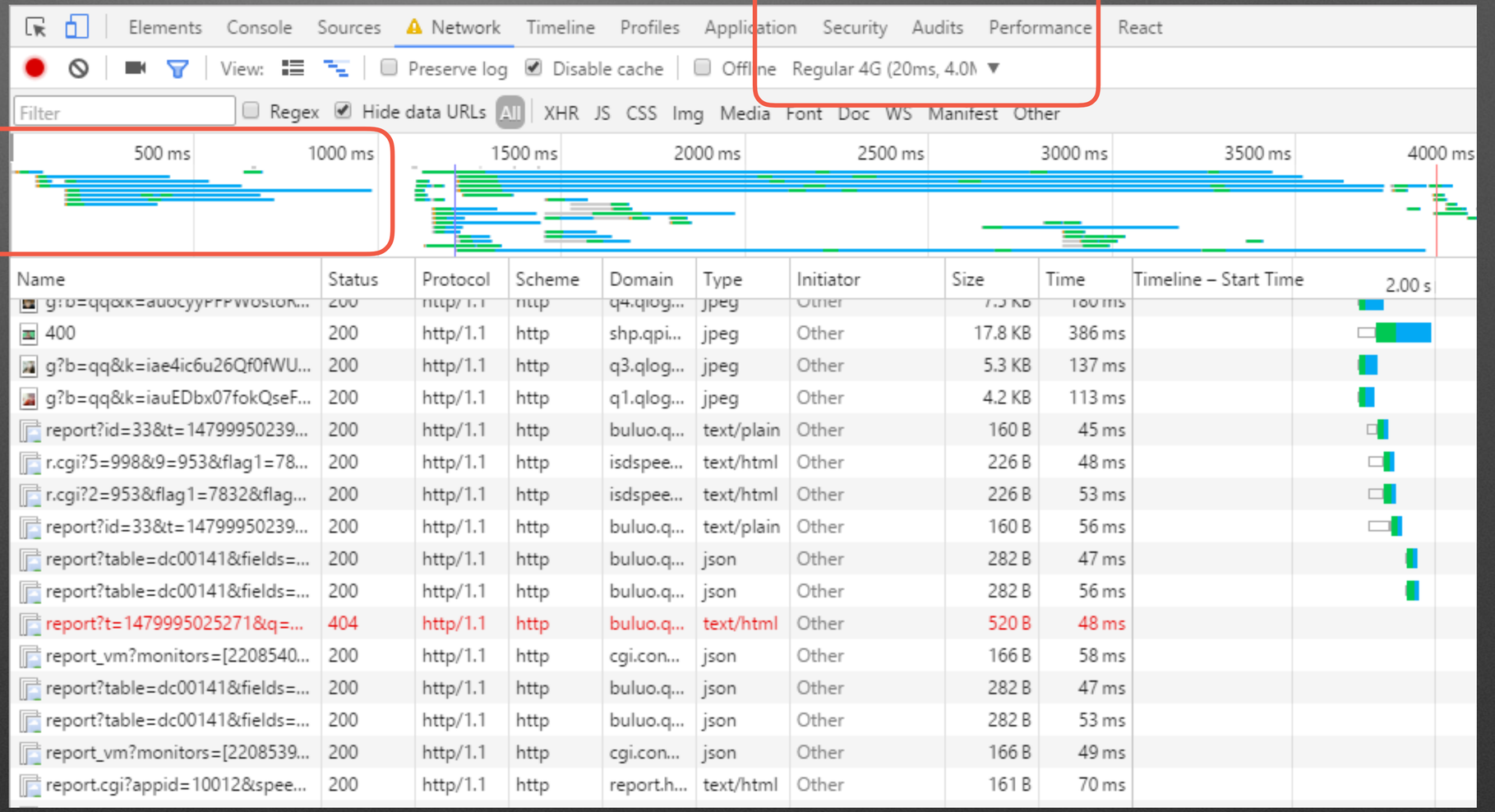
▸显示详细信息

下载已针对此网页优化的图片、JavaScript 和 CSS 资源。

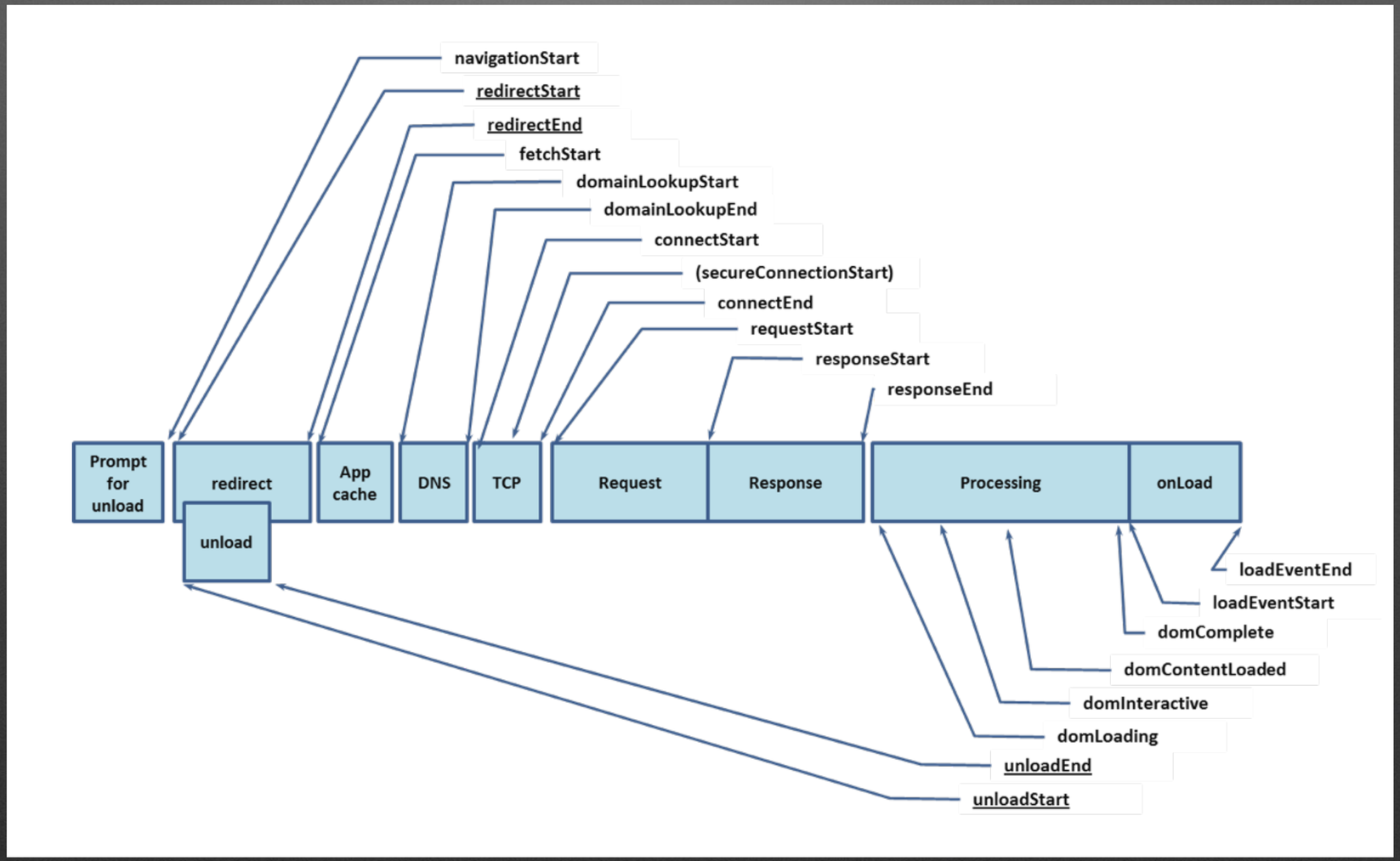
DevTools Timeline

网络模拟

脚本阻塞



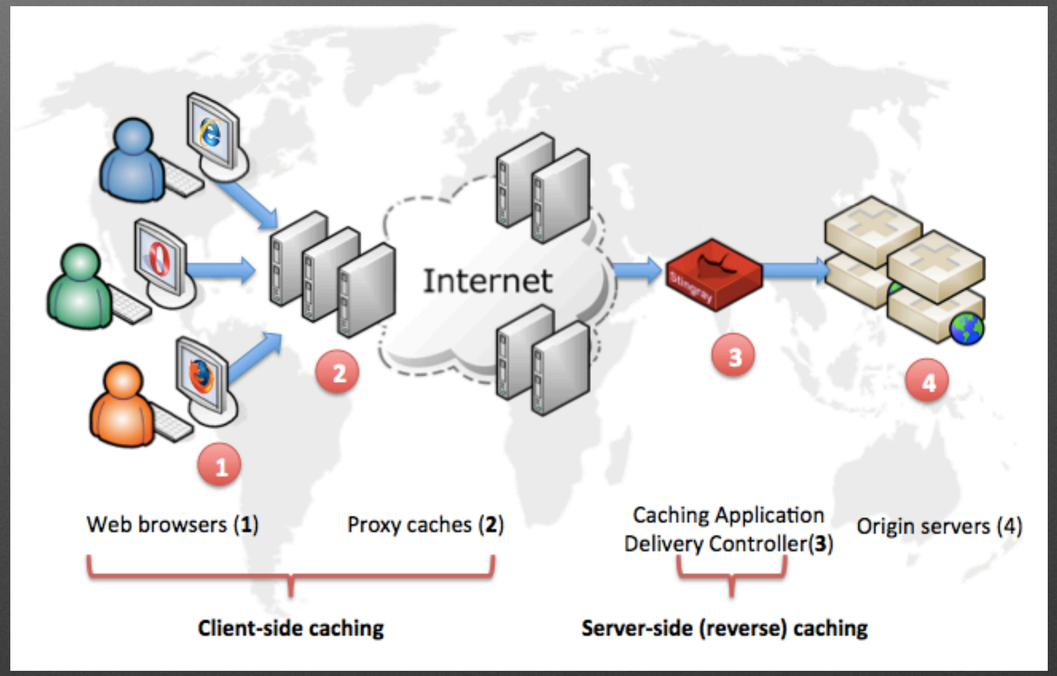
网络耗时测定


navigationStart
redirectStart
redirectEnd
fetchStart
domainLookupStart
domainLookupEnd
connectStart
(secureConnectionStart)
connectEnd
requestStart
responseStart
responseEnd
Prompt for unload
redirect
unload
App cache
DNS
TCP
Request
Response
Processing
onLoad
loadEventEnd
loadEventStart
domComplete
domContentLoaded
domInteractive
domLoading
unloadEnd
unloadStart

优化思路与学习方法

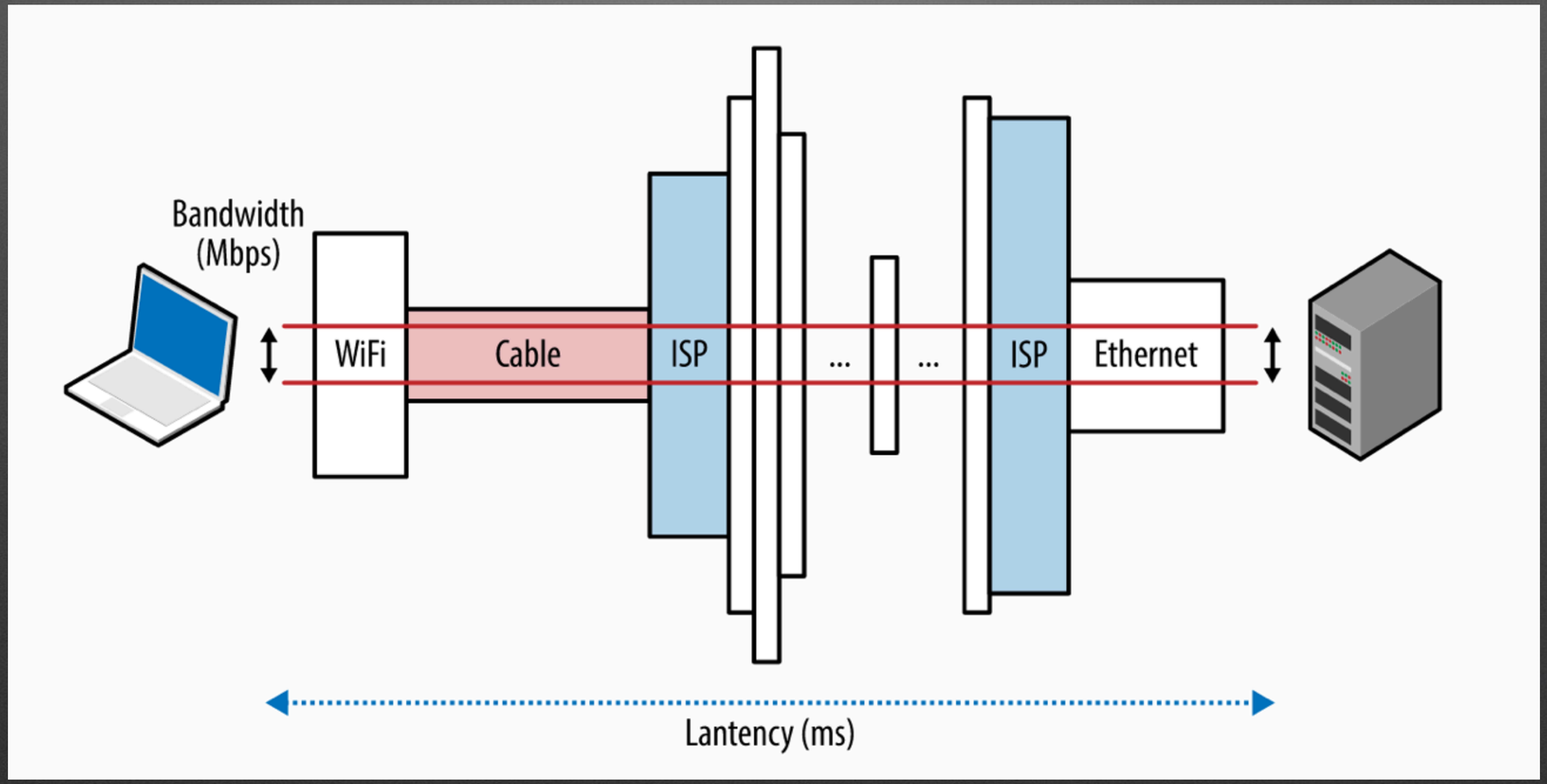
优化思路

减少网络时间

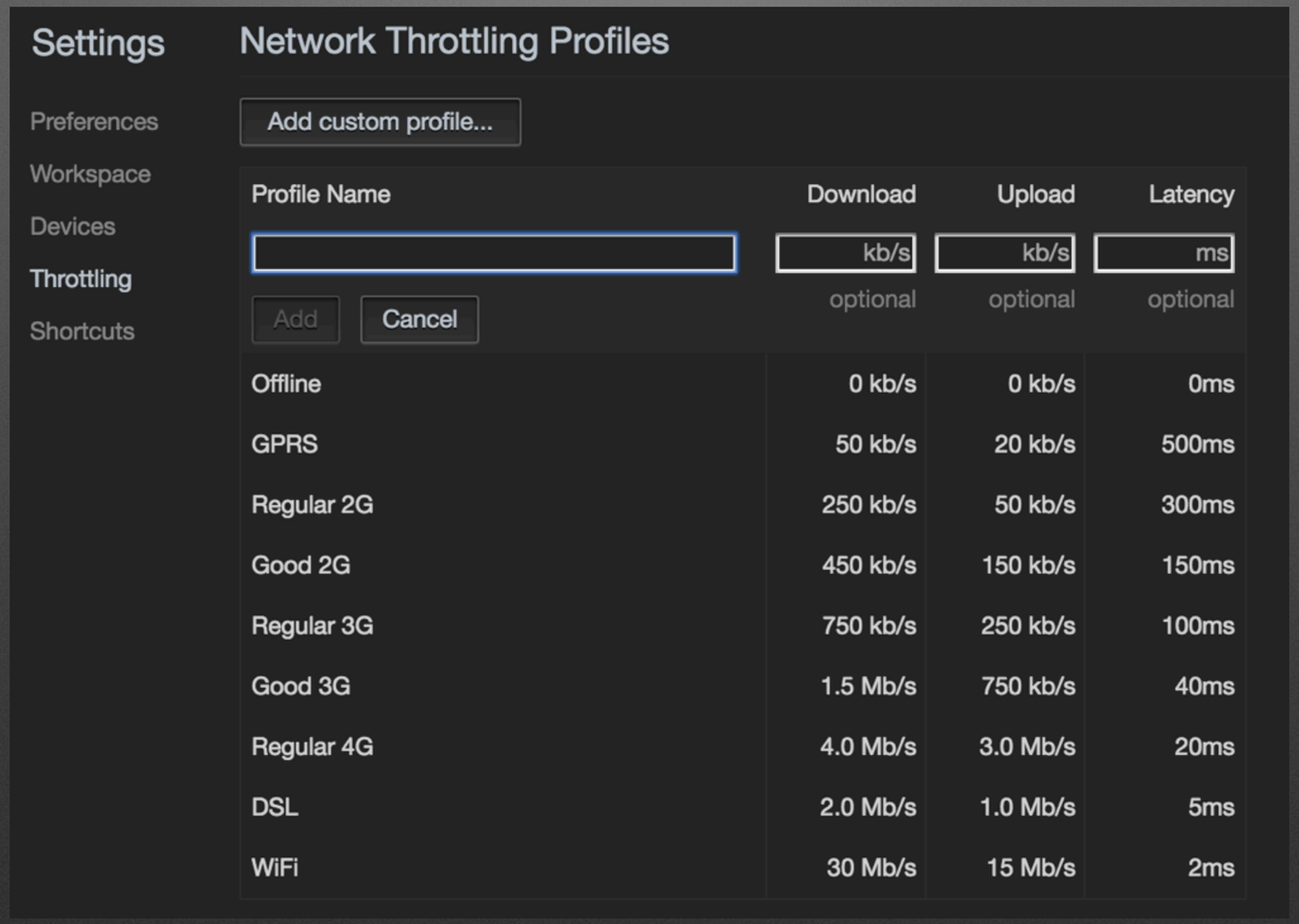


高效利用缓存

网络延迟（RTT）

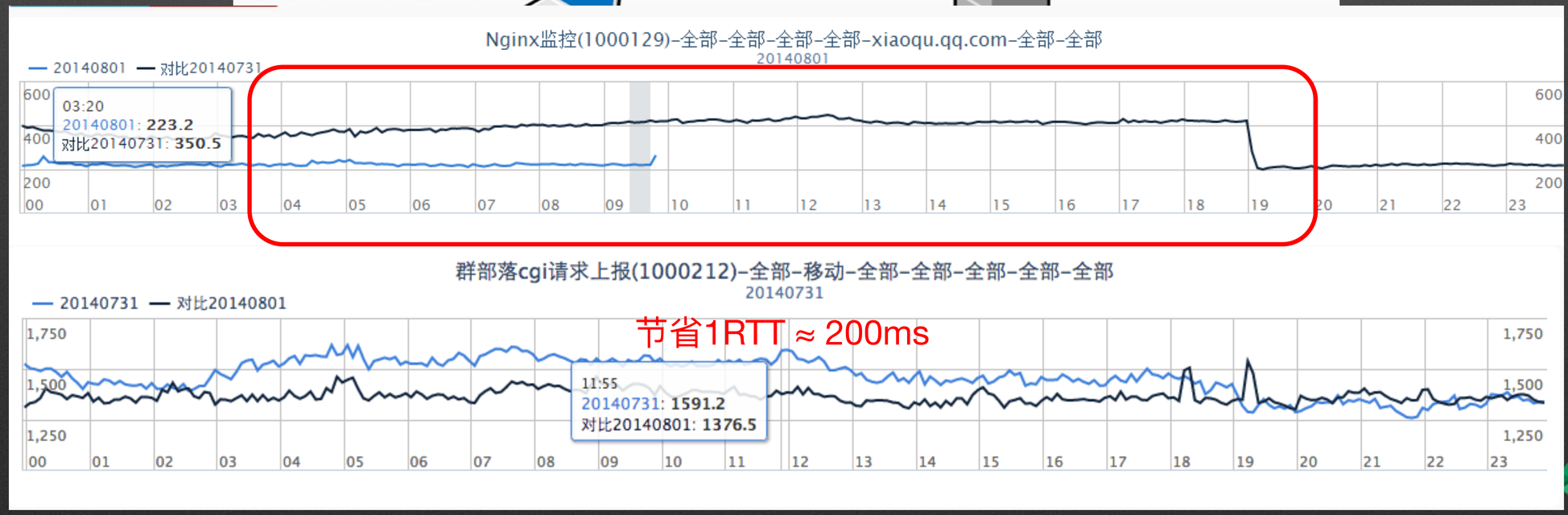


不同网络RTT


Settings
Network Throttling Profiles
Preferences
Workspace
Devices
Throttling
Shortcuts
Add custom profile...
Profile Name
Download
Upload
Latency
kb/s
kb/s
ms
optional
optional
optional
Add
Cancel
Offline 0 kb/s 0 kb/s 0ms
GPRS 50 kb/s 20 kb/s 500ms
Regular 2G 250 kb/s 50 kb/s 300ms
Good 2G 450 kb/s 150 kb/s 150ms
Regular 3G 750 kb/s 250 kb/s 100ms
Good 3G 1.5 Mb/s 750 kb/s 40ms
Regular 4G 4.0 Mb/s 3.0 Mb/s 20ms
DSL 2.0 Mb/s 1.0 Mb/s 5ms
WiFi 30 Mb/s 15 Mb/s 2ms

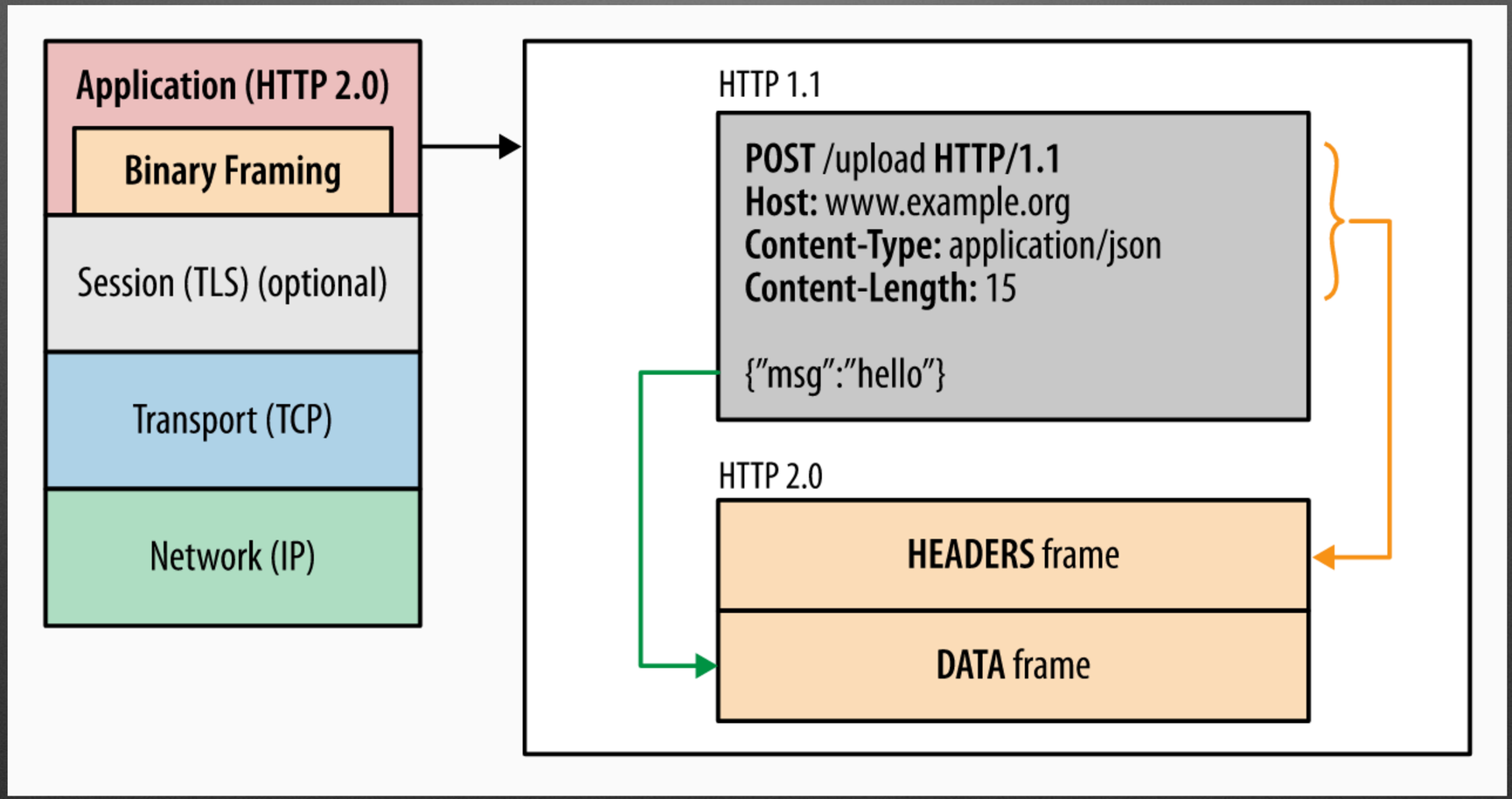
链路复用

- TCP3次握手
- keep-alive提升的部分
- Server内存消耗，预先扩容
- 实际提升效果

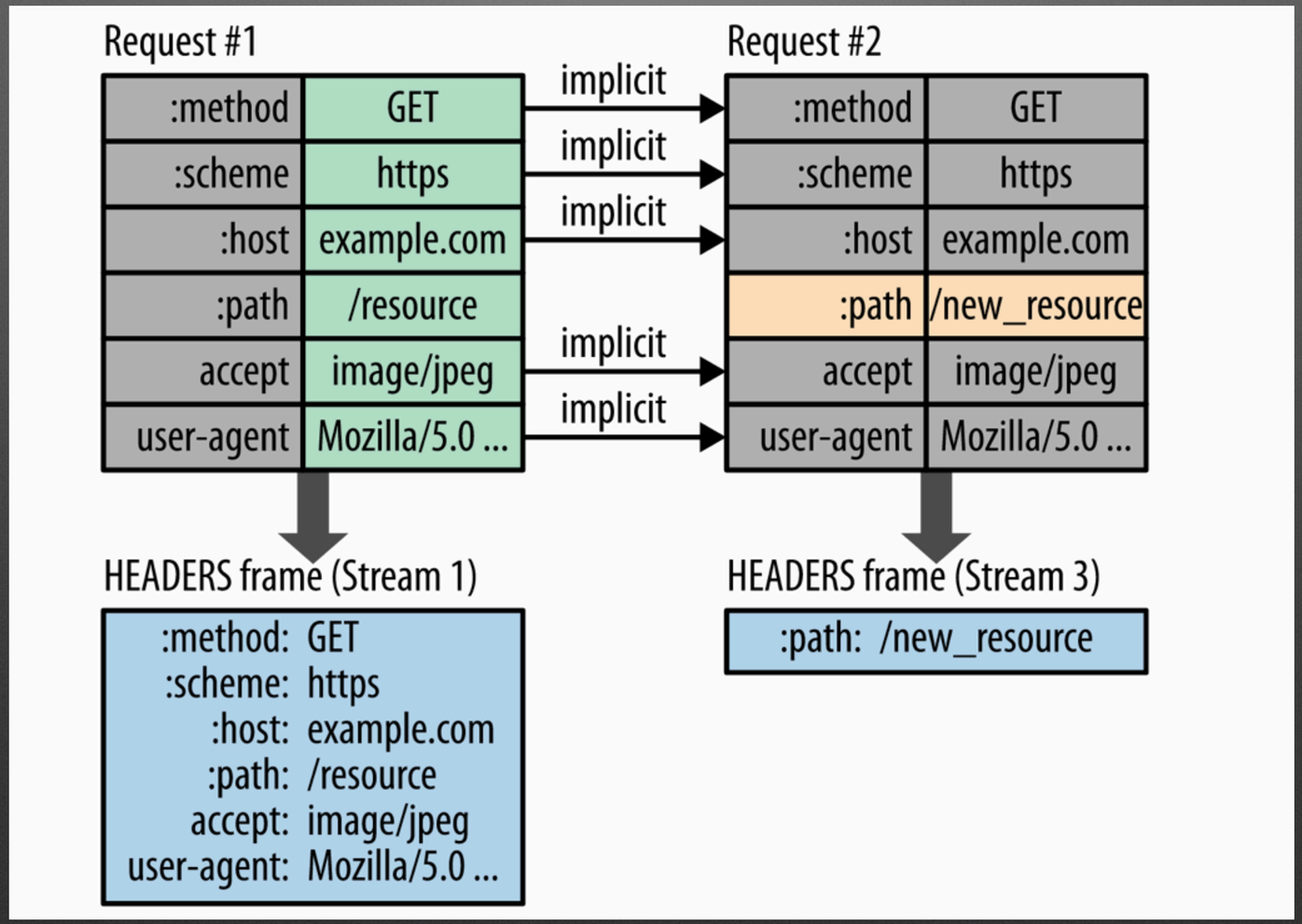


听说 HTTP/2 来了

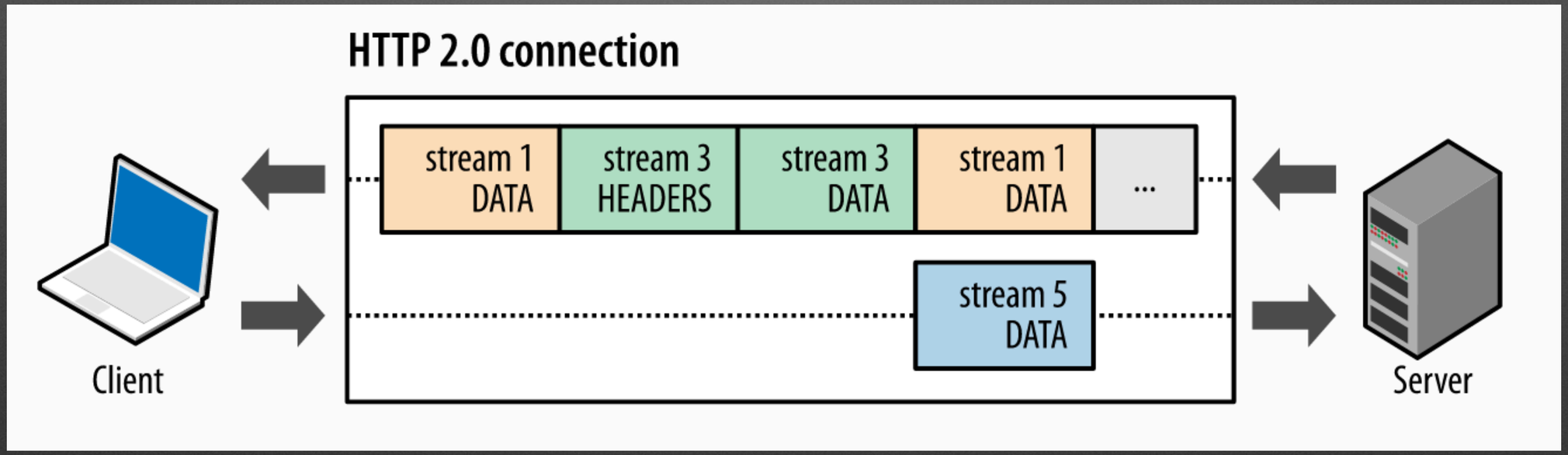
二进制分帧层


Application (HTTP 2.0)
Binary Framing
Session (TLS) (optional)
Transport (TCP)
Network (IP)
HTTP 1.1
POST /upload HTTP/1.1
Host: www.example.org
Content-Type: application/json
Content-Length: 15
{"msg":"hello"}
HTTP 2.0
HEADERS frame
DATA frame

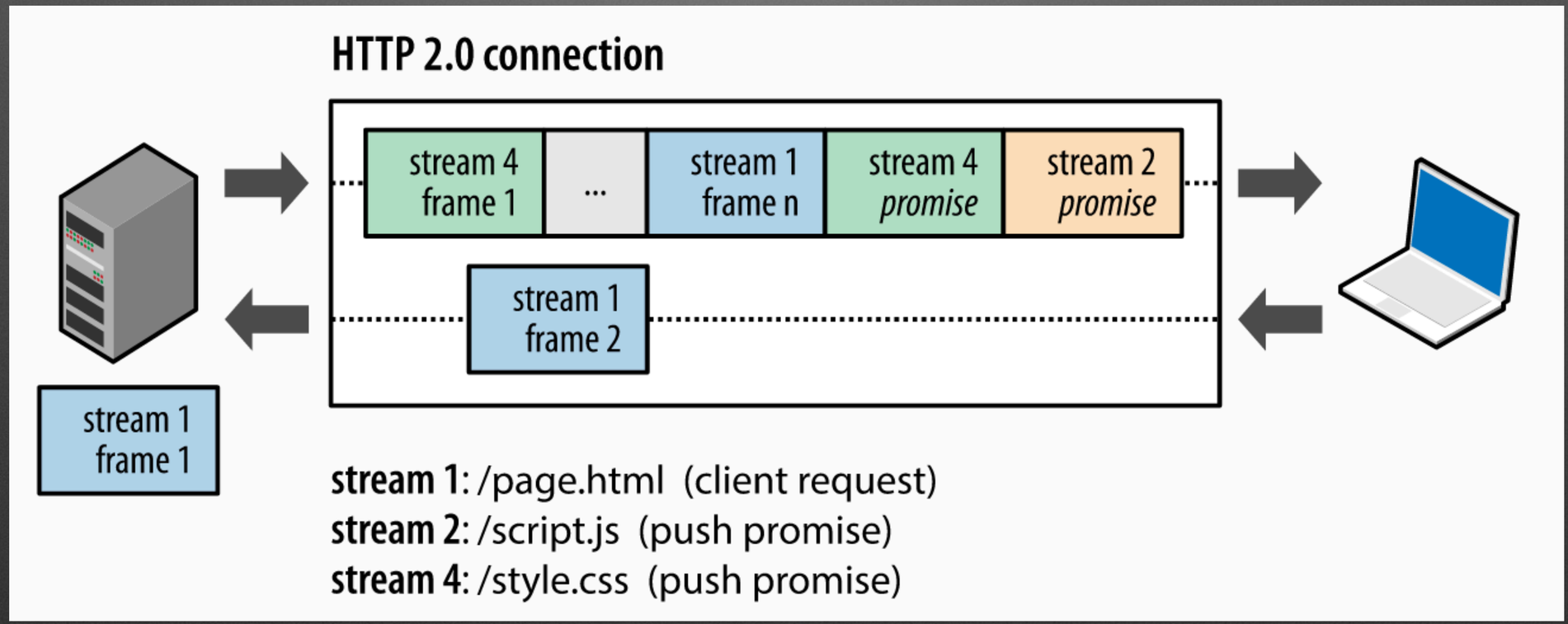
首部压缩


Request #1
:method GET
:scheme https
:host example.com
:path /resource
accept image/jpeg
user-agent Mozilla/5.0 ...
implicit
implicit
implicit
implicit
implicit
Request #2
:method GET
:scheme https
:host example.com
:path /new_resource
accept image/jpeg
user-agent Mozilla/5.0 ...
HEADERS frame (Stream 1)
:method: GET
:scheme: https
:host: example.com
:path: /resource
accept: image/jpeg
user-agent: Mozilla/5.0 ...
HEADERS frame (Stream 3)
:path: /new_resource

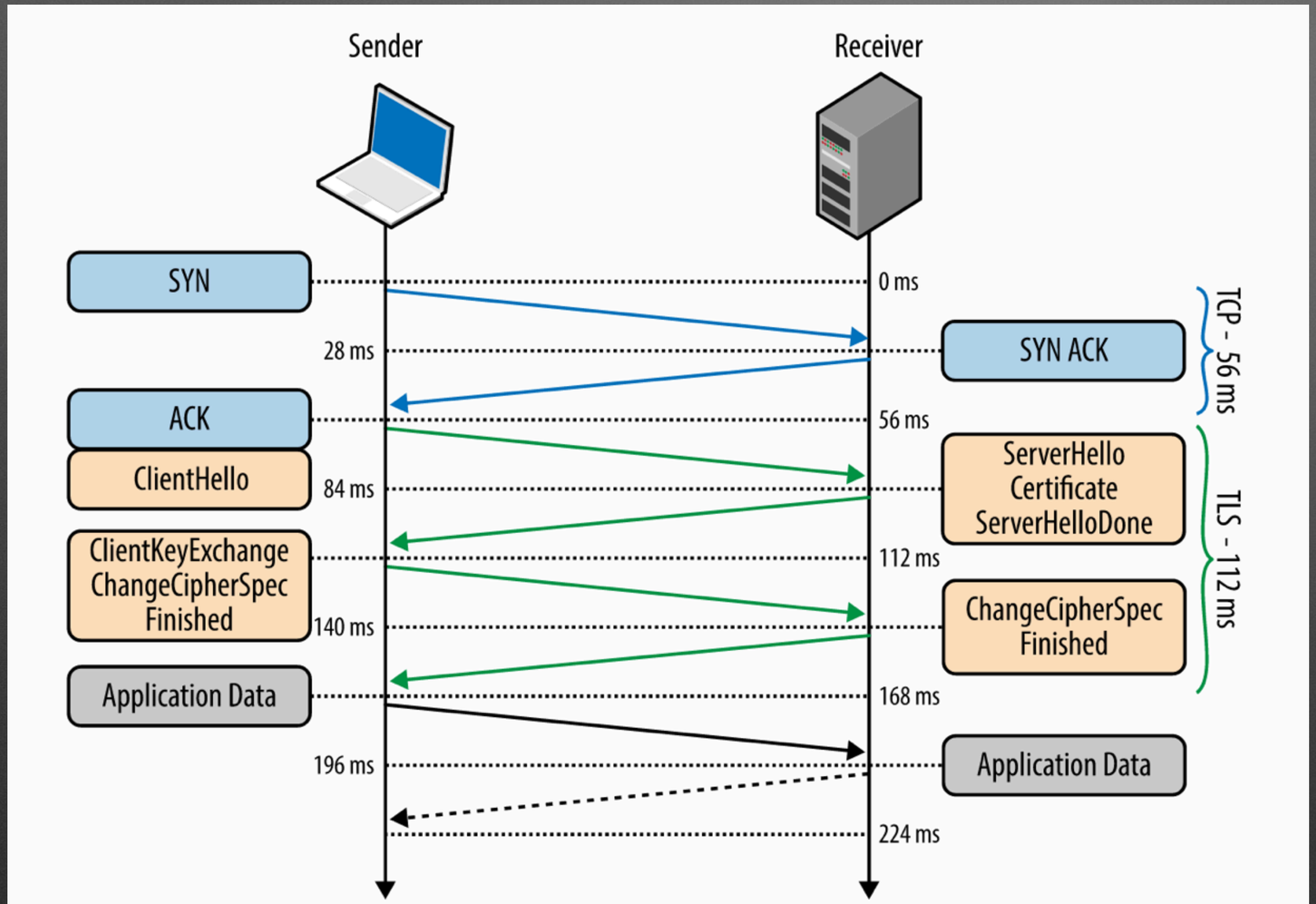
多路复用


HTTP 2.0 connection
stream 1
DATA
stream 3
HEADERS
stream 3
DATA
stream 1
DATA
...
stream 5
DATA
Client
Server

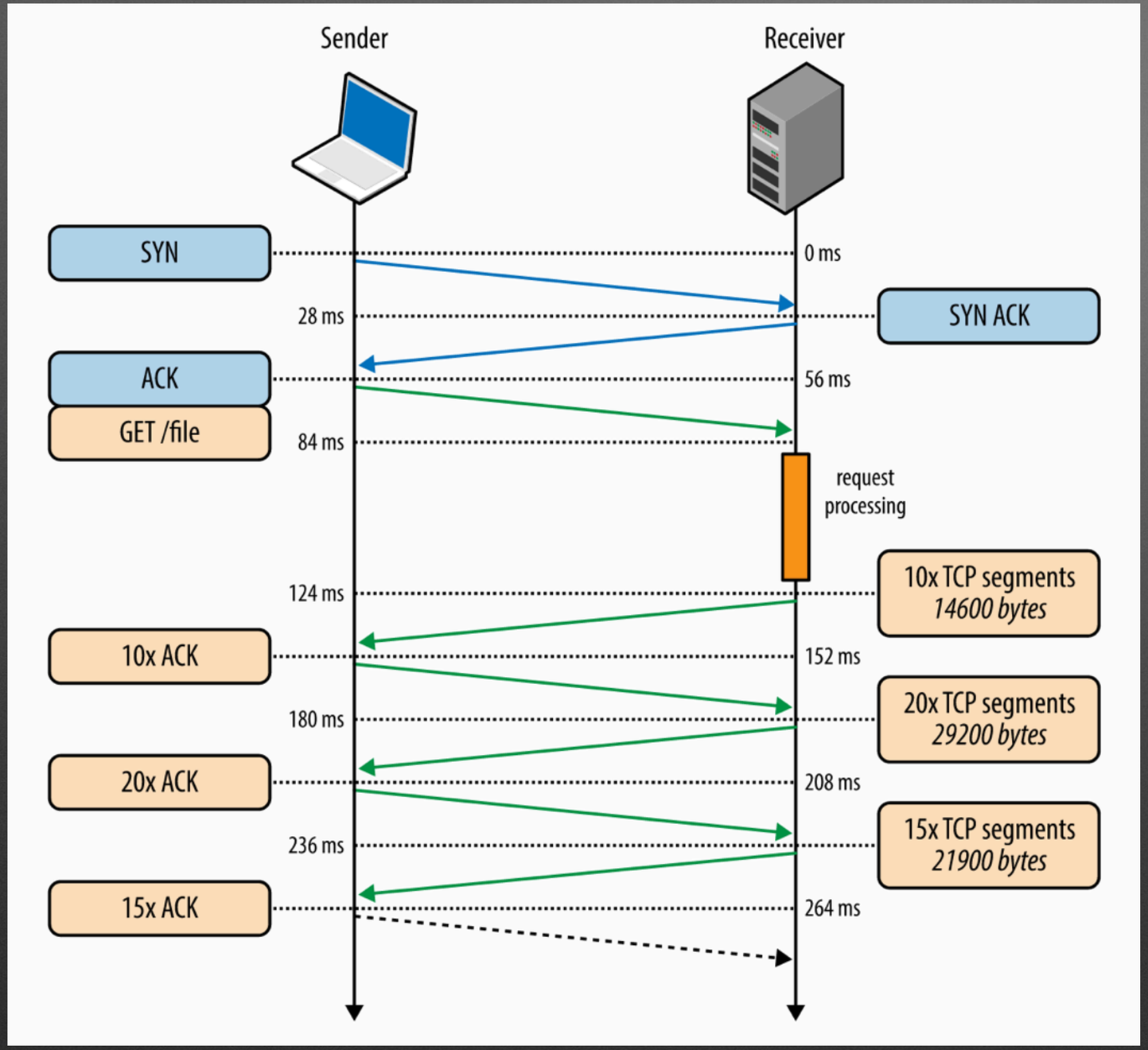
服务端推送


HTTP 2.0 connection
stream 4
frame 1
...
stream 1
frame n
stream 4
promise
stream 2
promise
stream 1
frame 2
stream 1
frame 1
stream 1: /page.html (client request)
stream 2: /script.js (push promise)
stream 4: /style.css (push promise)

Nice! But..


Sender
Receiver
SYN
0 ms
SYN ACK
28 ms
TCP - 56 ms
ACK
56 ms
ClientHello
84 ms
ServerHello
Certificate
ServerHelloDone
ClientKeyExchange
ChangeCipherSpec
Finished
112 ms
TLS - 112 ms
ChangeCipherSpec
Finished
140 ms
Application Data
168 ms
Application Data
196 ms
224 ms

TSL握手



TCP慢启动

- 初始化窗口数（10）
- 单个RTT内传输数据上限14KB
- HTTP/2 依旧受限

HTTP/2 小结

- 性能优势集中体现在“多路复用”和“服务端推送”上
- 请求数越多，RTT越大，优势越明显
- 依旧受HTTPS、TCP协议限制

学习方法

- 深入底层，知晓原理
- 辩证思维，不被新概念洗脑
- 加以实践，论证有效

性能优化实践

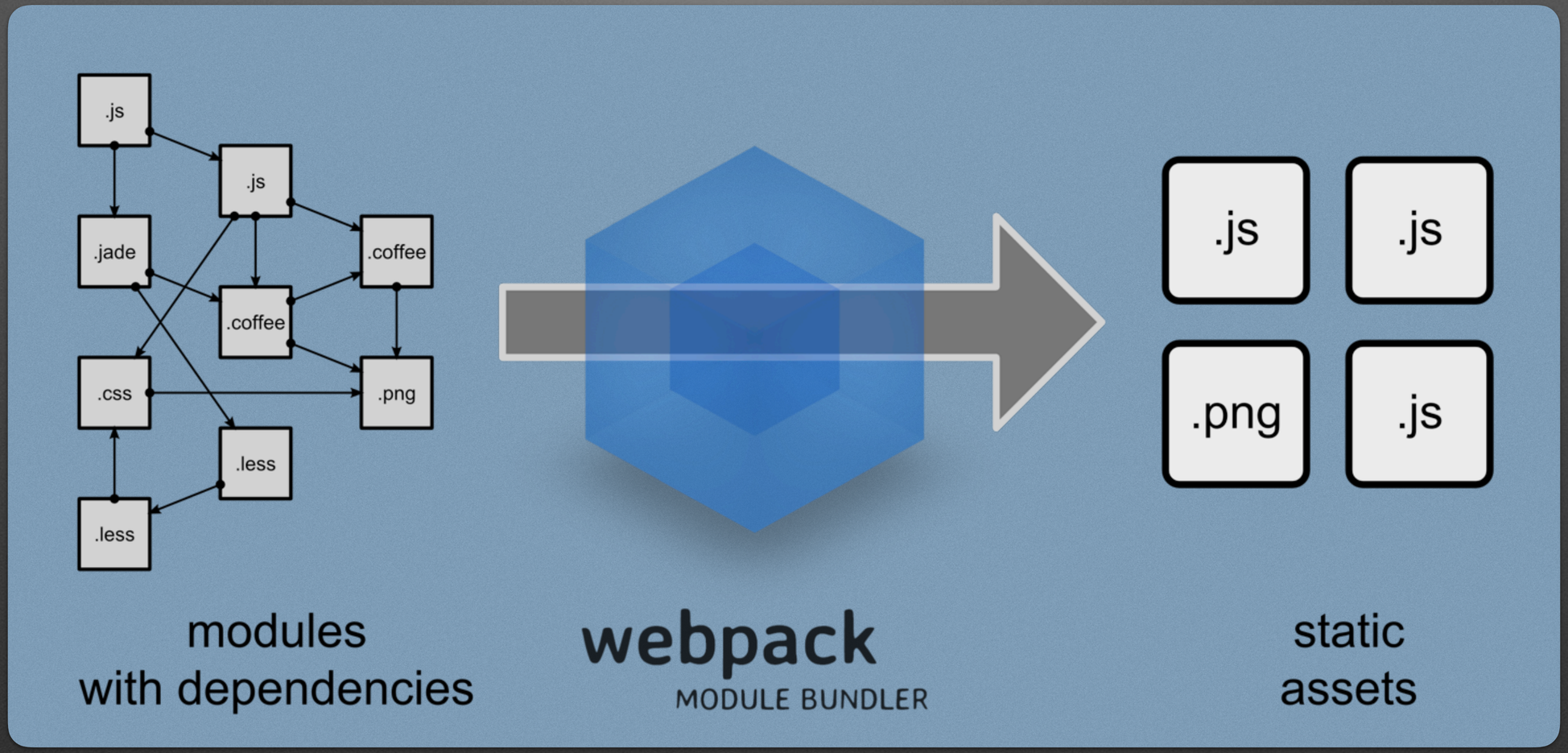
服务端渲染

- Ajax模式下，数据与页面资源串行
- 缩短网络路径，降低网络异常影响
- 使用Node.js进行服务端渲染
- 组件前后端同构

图片懒加载

- 模版渲染时，不赋值src
- 插入DOM树时进行可视区域计算
- 如果图片在可视区内，赋值src
- 如果图片在可视区外，留着scroll处理

更智能的打包


.js
.js
.jade
.coffee
.coffee
.css
.png
.less
.less
modules
with dependencies
webpack
MODULE BUNDLER
.js
.js
.png
.js
static
assets

高效利用缓存

网络缓存

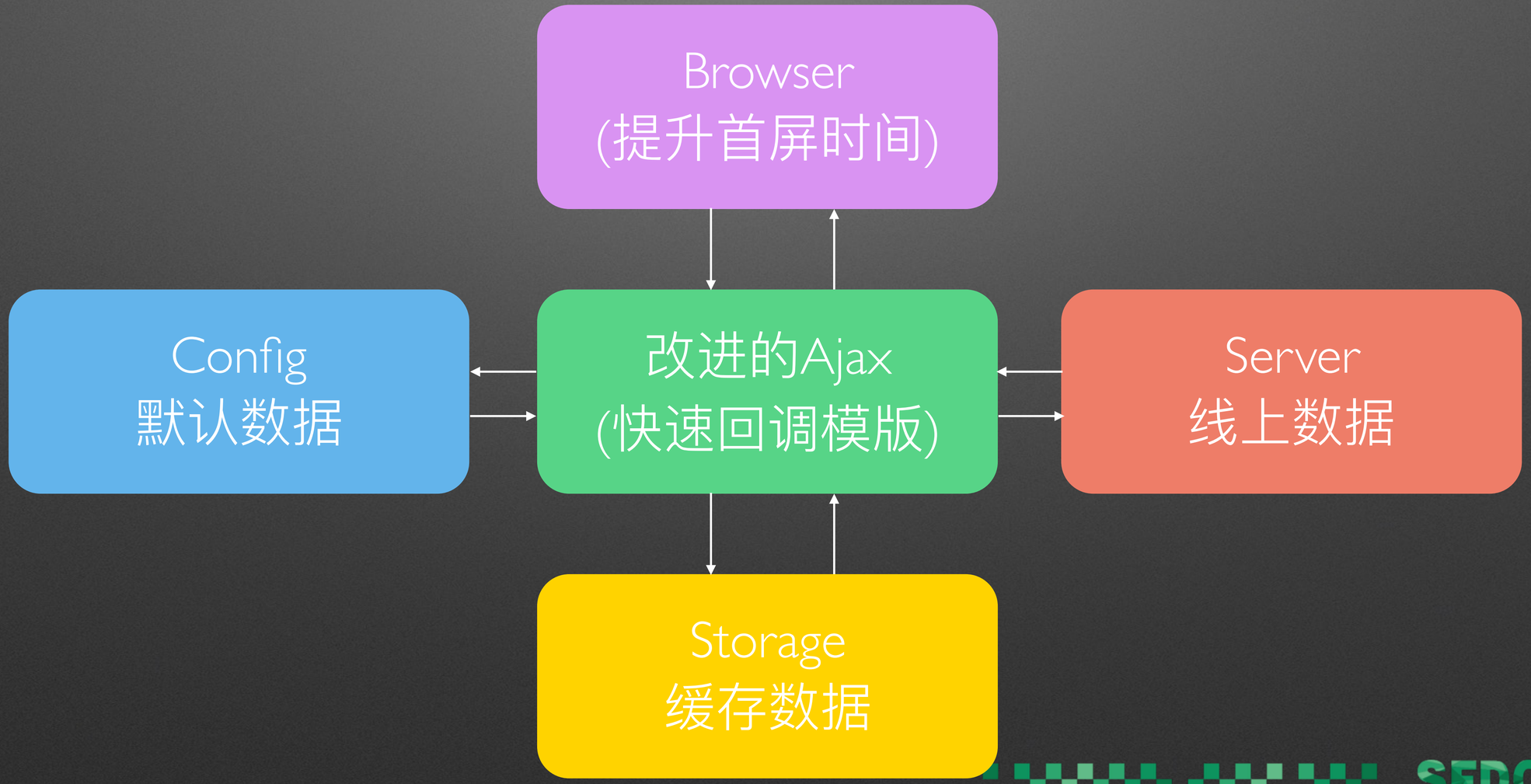
- HTML 304 Cache
- 静态资源Long Cache（max-age）
- 文件MD5命名
- 背后的考虑：RTT影响、CDN刷新问题

原生App差距?

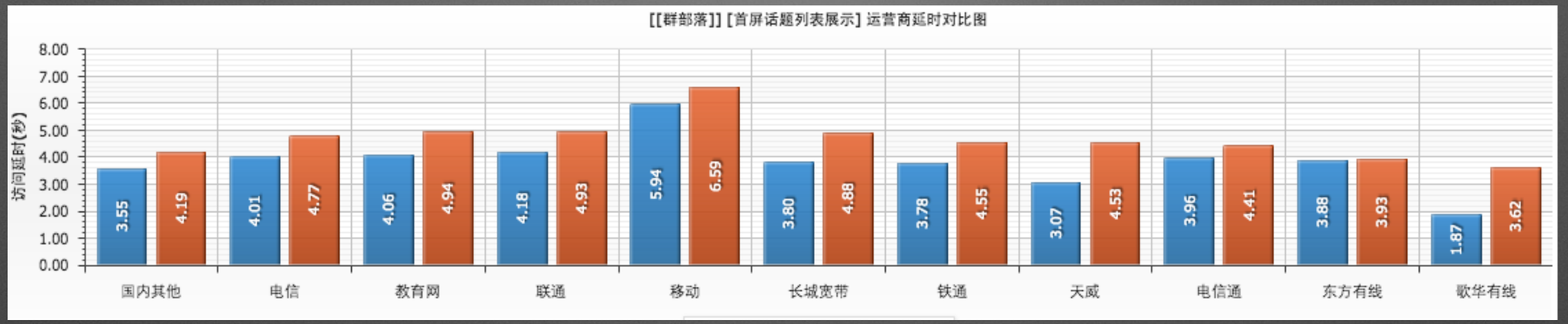
数据缓存设计

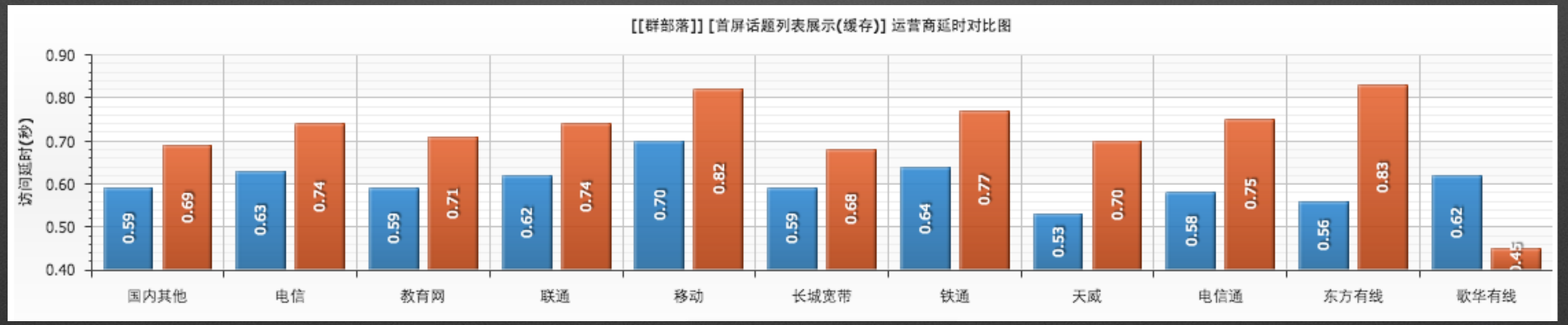
- 高度可配置化（本次是否使用、缓存时间等）
- 更新通知机制
- 默认数据配置
- 重要数据标记
- 过期清除算法

数据缓存架构


Browser
(提升首屏时间)
Config
默认数据
改进的Ajax
(快速回调模版)
Server
线上数据
Storage
缓存数据

优化效果

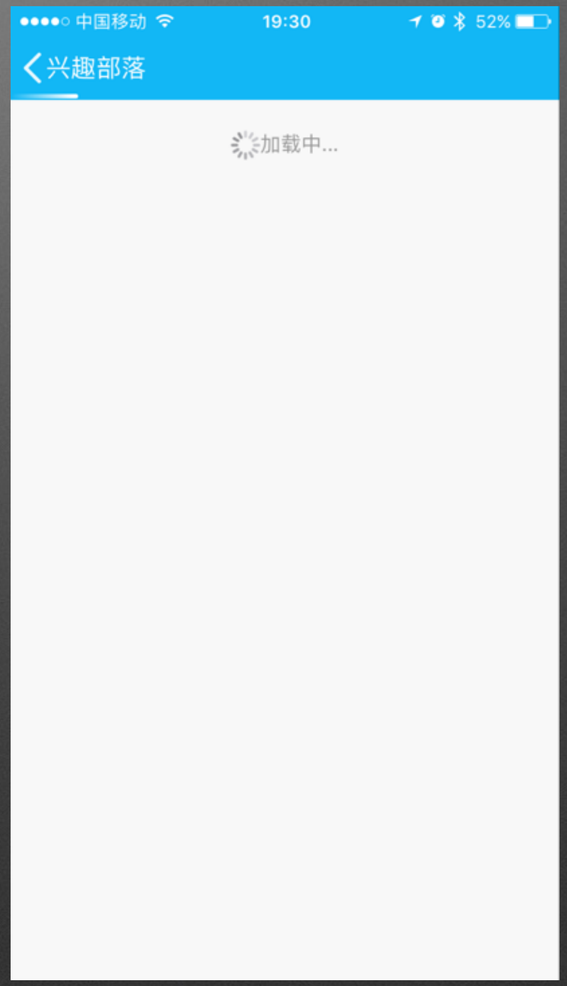




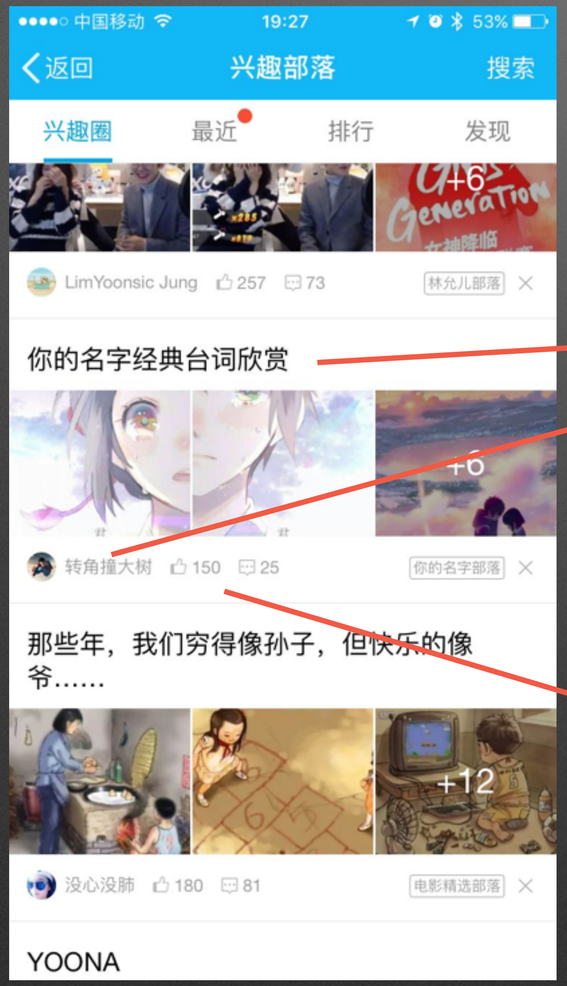
[群部落]	群部落	部落首页	首屏话题列表展示	否	0	7100780	5.34962	12.97%	johnnyguo
[群部落]	群部落	部落首页	首屏话题列表展示(缓存)	否	0	3917357	0.757308	1.13%	johnnyguo

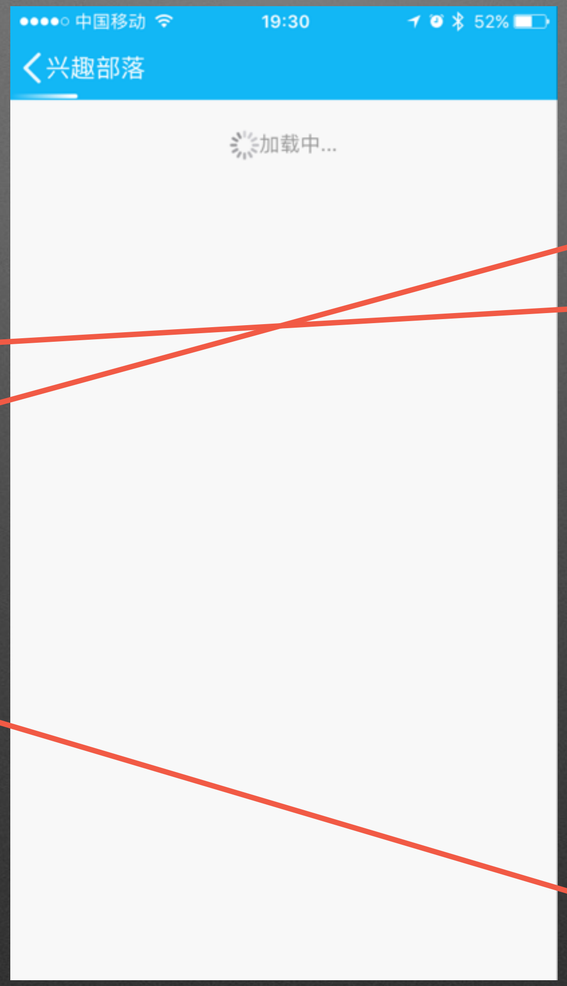
数据复用


中国移动 19:27 53%
返回
兴趣部落
搜索
兴趣圈
最近
排行
发现
+6
LimYoonsic Jung 257 73
林允儿部落
你的名字经典台词欣赏
+6
转角撞大树 150 25
你的名字部落
那些年，我们穷得像孙子，但快乐的像爷……
+12
没心没肺 180 81
电影精选部落
YOONA


中国移动 19:30 52%
兴趣部落
加载中...

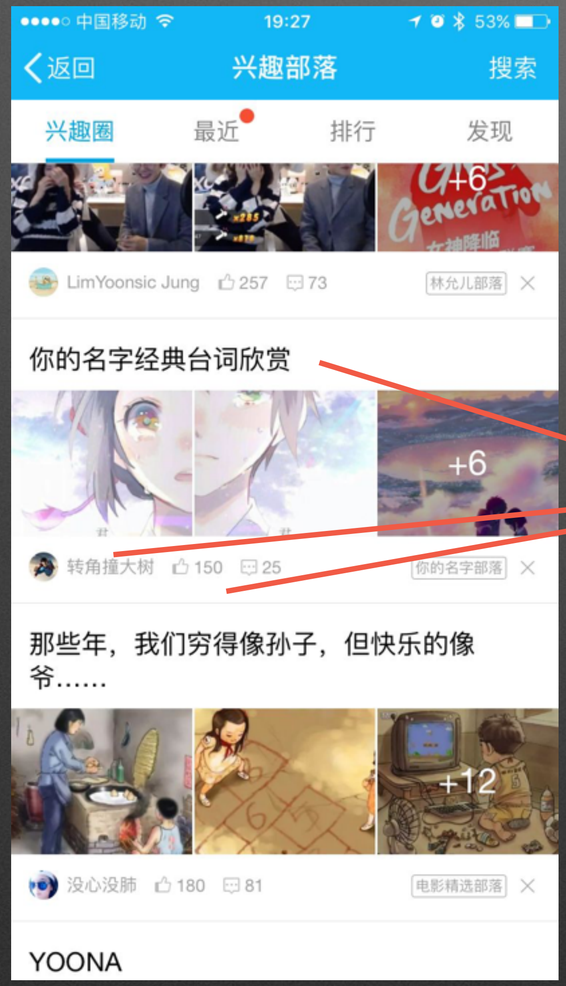
非缓存的数据传递方案


中国移动 19:27 53%
返回
兴趣部落
搜索
兴趣圈
最近
排行
发现
+6
LimYoonsic Jung 257 73
林允儿部落
你的名字经典台词欣赏
+6
转角撞大树 150 25
你的名字部落
那些年，我们穷得像孙子，但快乐的像爷……
+12
没心没肺 180 81
电影精选部落
YOONA


中国移动 19:30 52%
兴趣部落
加载中...

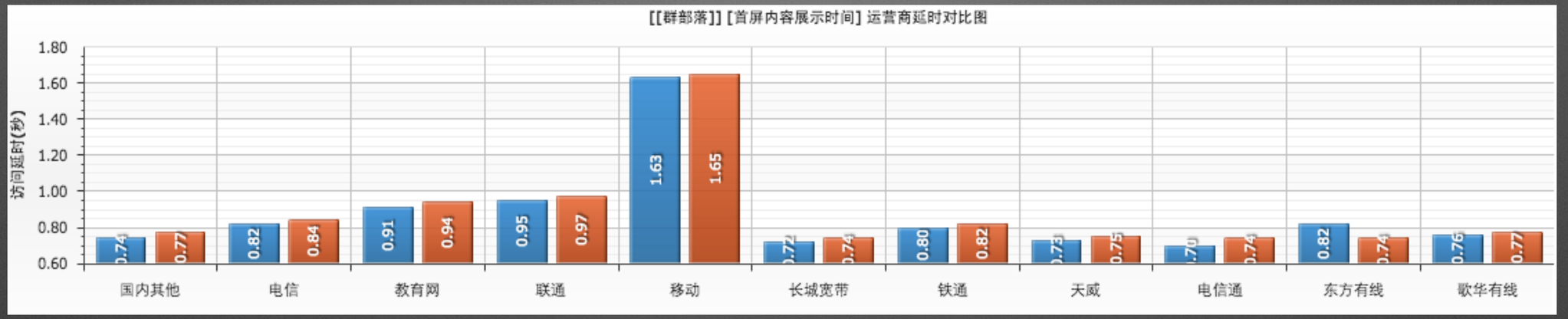

中国移动 19:27 53%
返回
话题
转角撞大树
初级粉丝
我不是随便的人，但随便起来就不是人。
你的名字经典台词欣赏
2016-12-6 19:14
你的名字部落
* 连自己忘却了这件事，醒来的时候， 不知道自己为什么哭， 时常会有的事情 ， 做过的梦总是回想不起， 只是 一种有什么消失的丧失感。
25
150
36

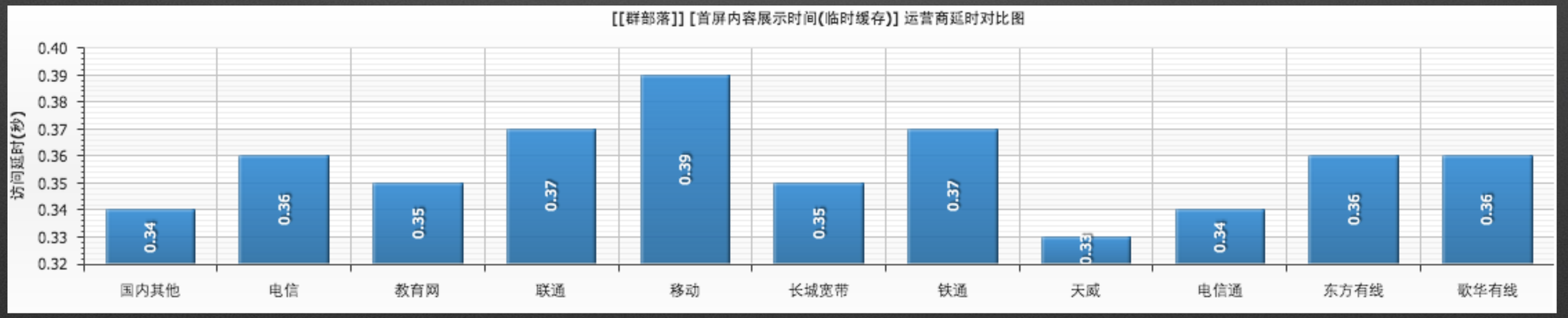
非缓存的数据传递方案



Storage

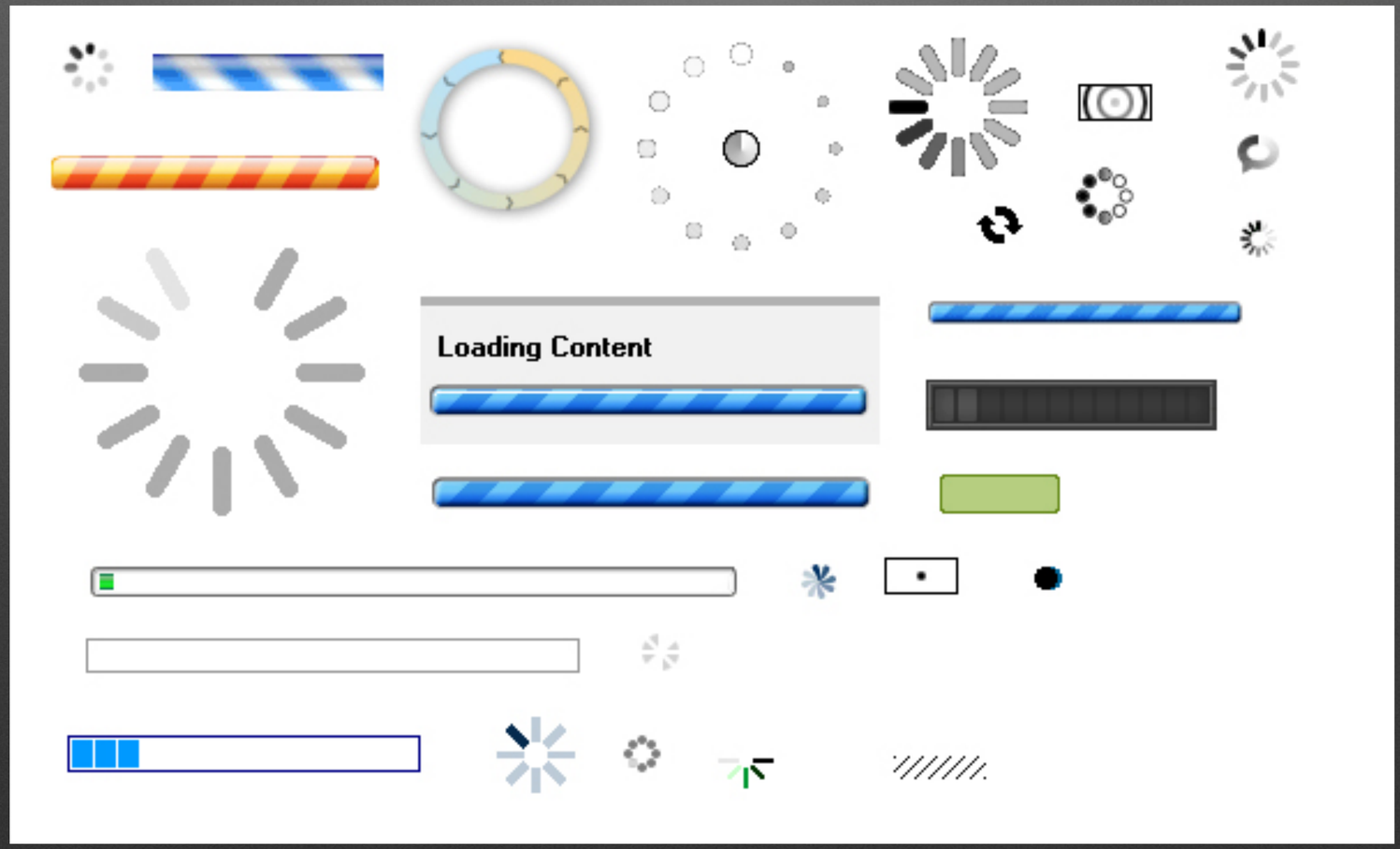
优化效果





实在无法优化了，怎么办?

优化绝学



THANK YOU

Q & A