大数据下的前端性能优化实践

—— 庞锦贵

1 前端性能数据该如何采集

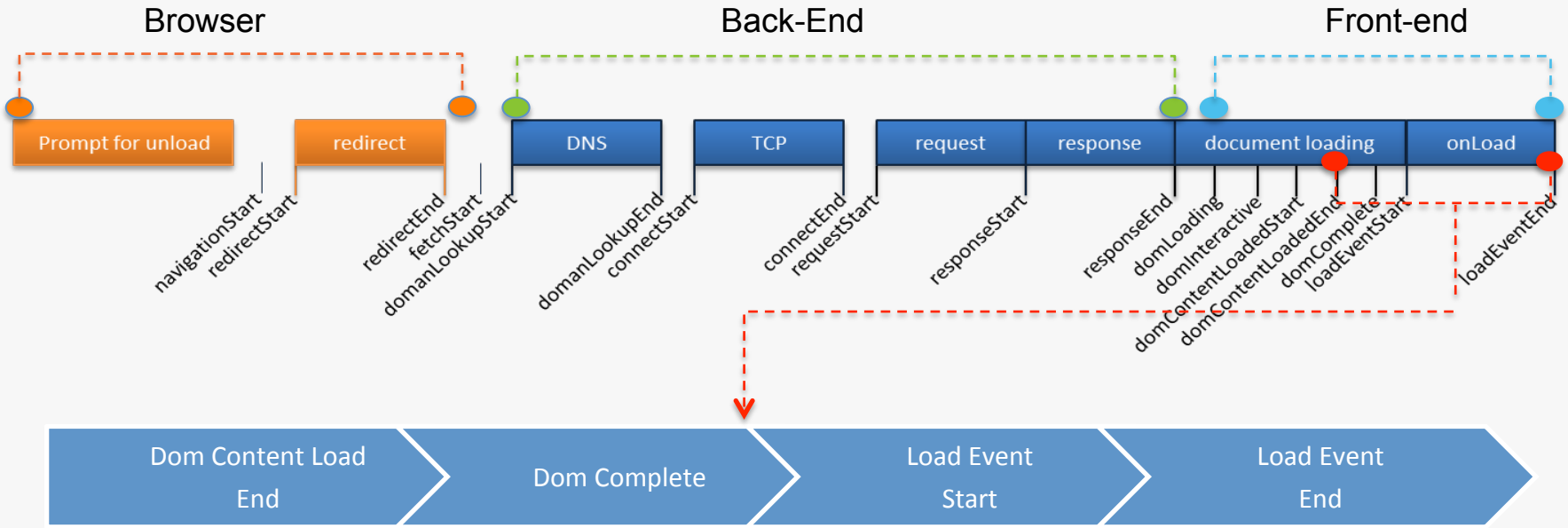
2 性能优化的项目、目标及成果

3 我们采用了哪些优化策略

4 优化过程中的经典案例分享

5 优化总结

前端页面性能指标如何采集?



前端的白屏时间、首屏时间等性能指标如何获取？

常规的页面性能统计模型

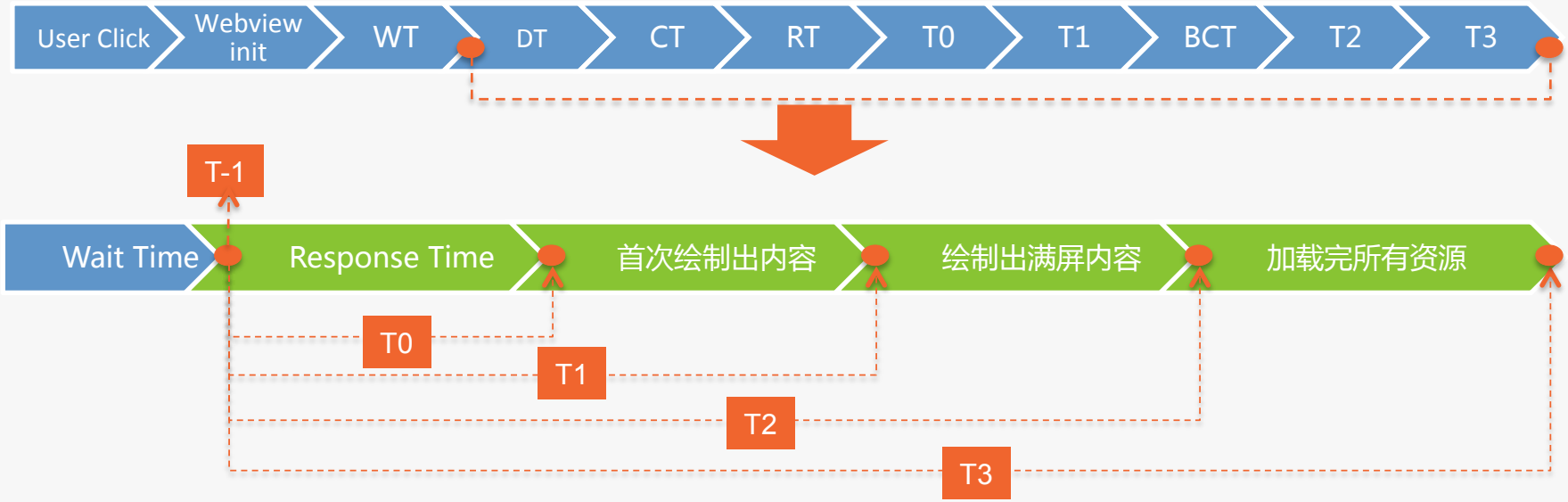
后端渲染

前端渲染

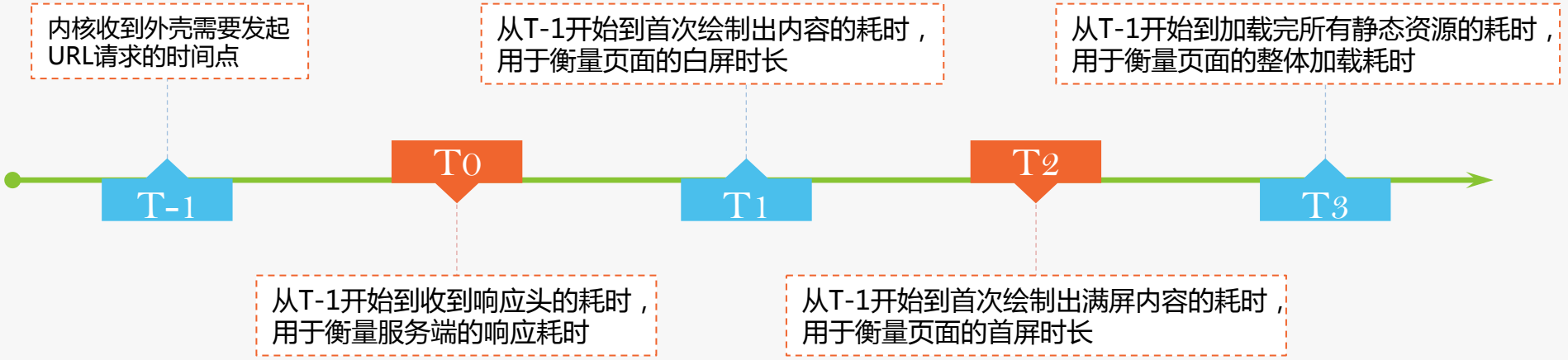
前端打点的性能统计模型存在的问题：

- 不同的业务模式或用户场景，前端性能指标的统计是不一样的
- 无论后端渲染还是前端渲染，基于前端JS的首屏时间统计都是很难精确的

我们的页面性能统计模型



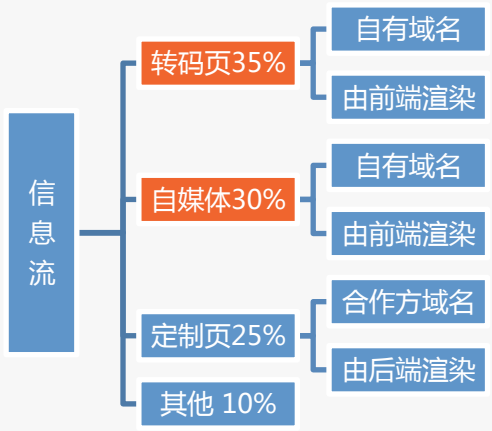
Tn等性能指标概念及其意义



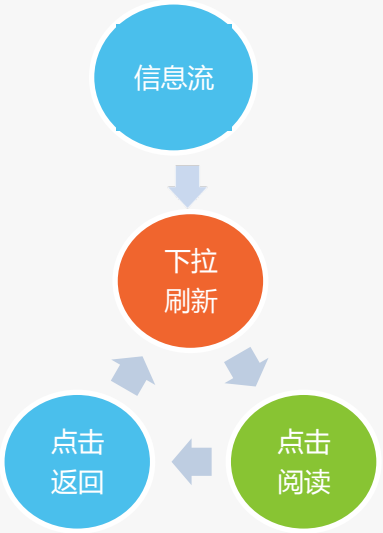
Tn性能指标的意义

- 解决在不同的业务模式或场景下，页面性能指标的一致性问题；
- 解决白屏、首屏等关键性用户体验指标的不精确、不准确或无法统计的问题；
- 统一各端的性能统计模式，避免页面性能数据重复建设的问题。

我们需要优化的项目及其用户场景



主要的用户场景

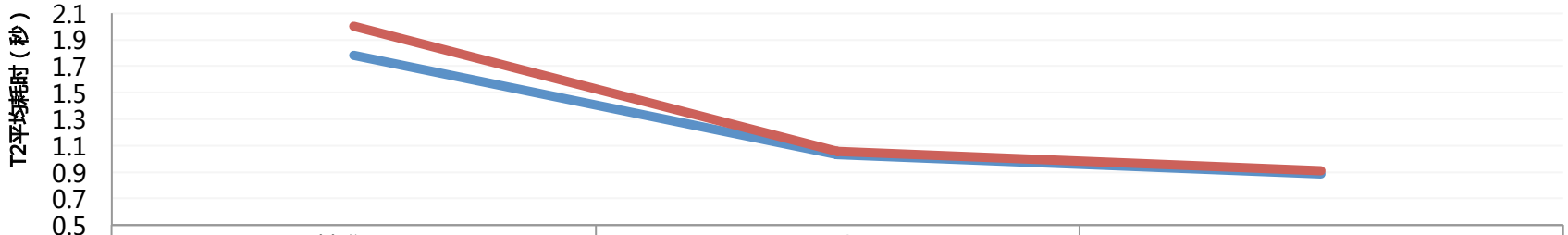


用户访问的特点：

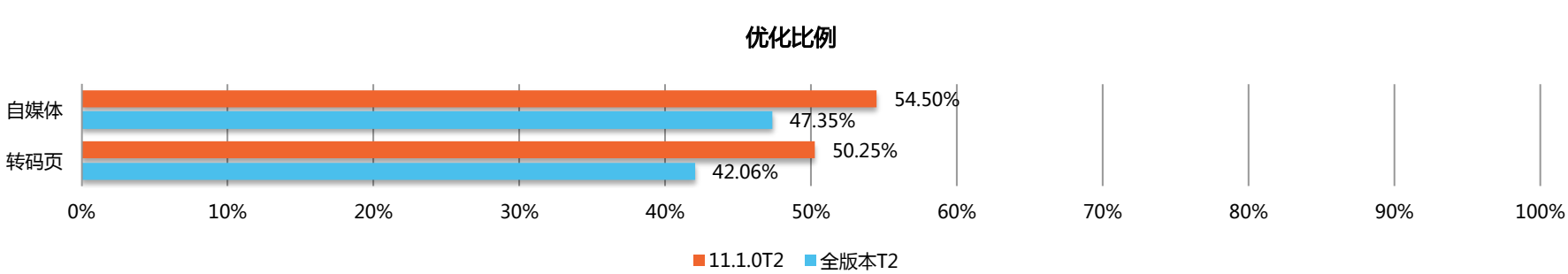
1. 同一用户打开一篇文章的次数一般只有1次
2. 不同的文章之间的差别只是内容不同，但前端资源是相同的；
3. 页面主要是图文为主的内容展示，在首屏幕内的需要交互场景很少

项目优化的目标及成果

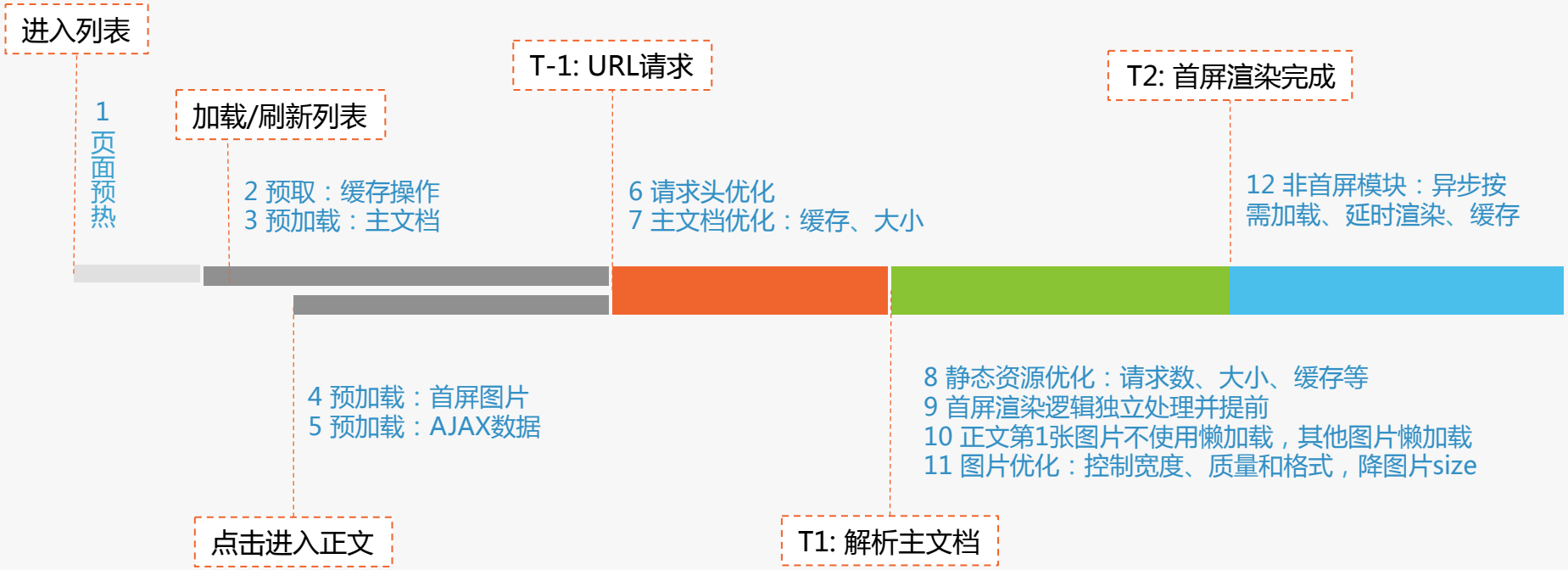
目标: T2(首屏渲染)时间在基准数据的基础上提升30%。



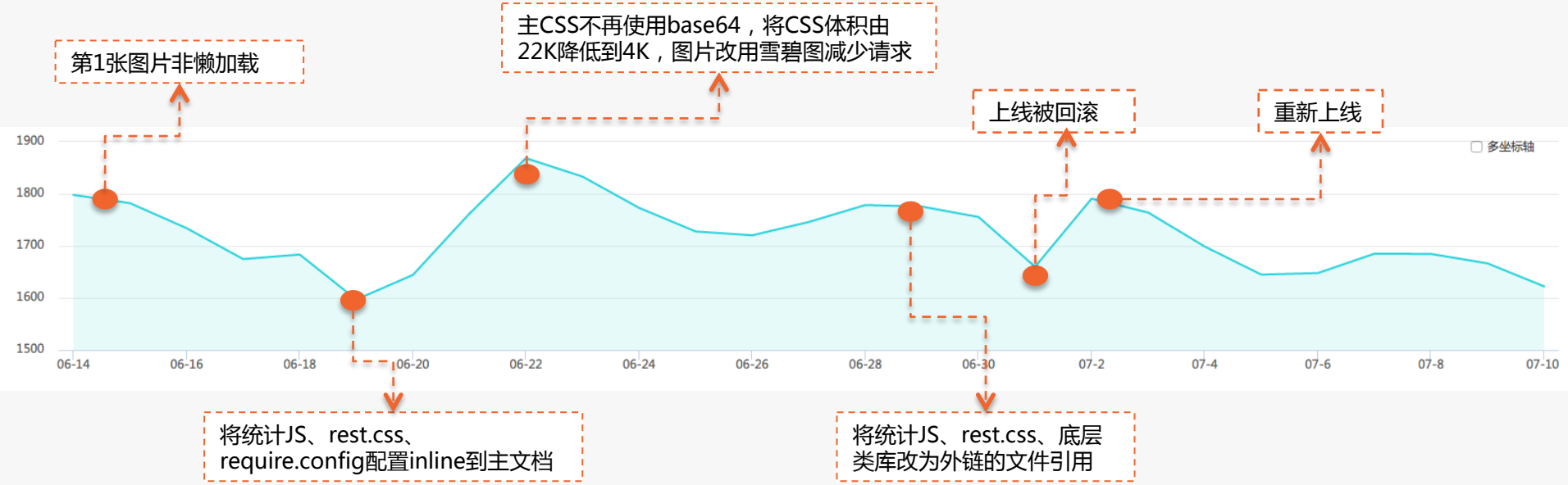
	基准T2	全版本T2	11.1.0T2
转码页	1.781	1.032	0.886
自媒体	2	1.053	0.91



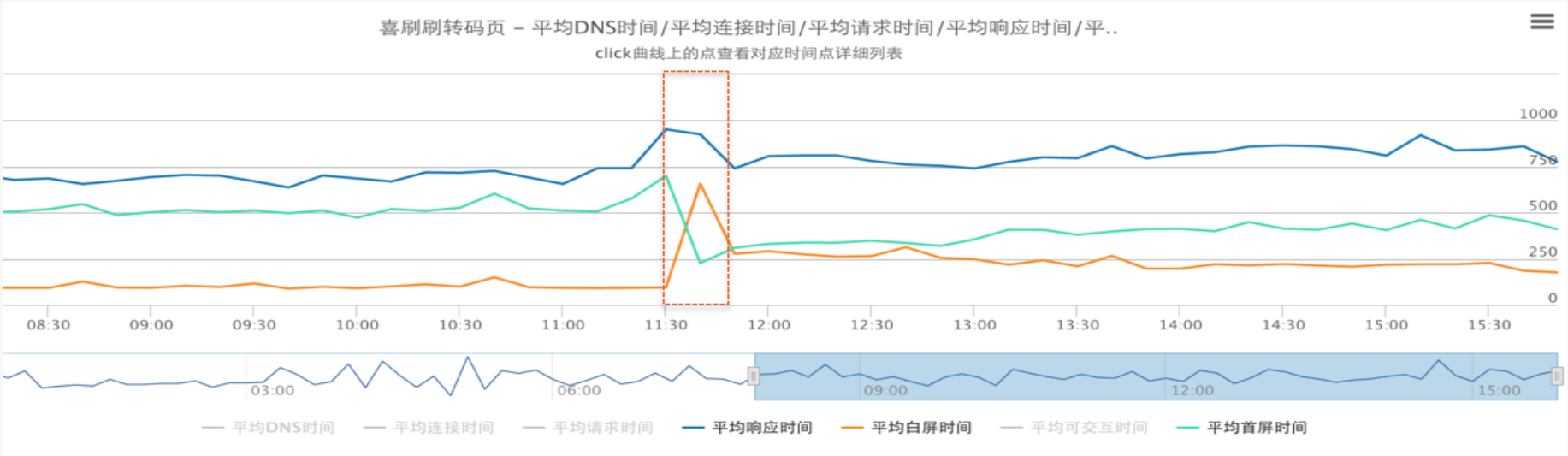
我们采取了哪些关键的优化策略？



大数据的性能优化策略解读(1)：内联CSS/JS不一定能加快首屏渲染



大数据的性能优化策略解读(1)：内联CSS/JS不一定能加快首屏渲染



页端基于 Performance.timing API的性能打点统计：

- 平均响应时间上浮约 100ms；平均白屏时间上浮约 120ms；平均首屏时间下降约 100ms。
- 往后拉通看，首屏时间基本没有变化，但响应时间的增长和白屏时间均有上升了。

大数据的性能优化策略解读(1)：内联CSS/JS不一定能加快首屏渲染

为什么使用内联CSS/JS不能加快首屏渲染？

用户进入场景对比		http链接数	主文档size	综合
	第一次进入			
	内联css/js	1	大	内链有优势
	外链css/js	3	小	
	第2次以上进入			
	内联css/js	1	大	外链有优势
	外链css/js	1	小	

我们的业务特点：

1. 同一用户打开一篇文章的次数一般只有1次；
2. 从信息流到正文的人均阅读文章篇约为7次；

结论：当人均阅读次数 > 2，外链CSS/JS在T2首屏上更有优势。

大数据的性能优化策略解读(2)：页面预热让用户始终是第2次访问

- **页面预热**：用户打开APP进入信息流的时候，用一个隐藏的webview加载一次正文页面。

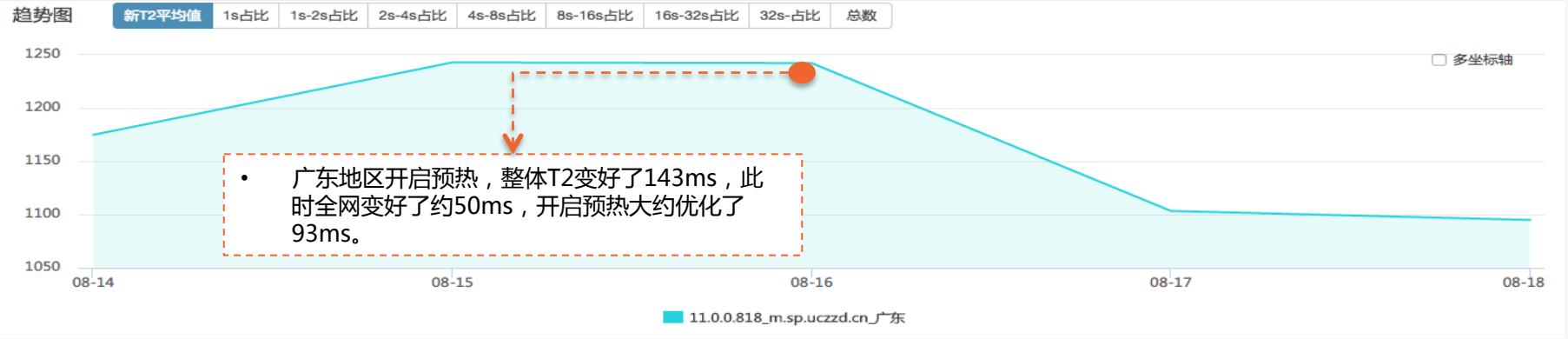
正常的页面网址： http://m.uczzd.cn/webview/news?app=uc-iflow&aid=17502639495837468714&cid=100

预热的页面网址： http://m.uczzd.cn/webview/news

好处	缺点
• 提前缓存静态资源加，快正文的打开速度 • 内核可以把页面需要的静态资源提前读入内存 • 可平衡前端每次发版的缓存更新压力	• 需要多消耗用户的流量 • 增加服务器的访问压力 • 对于前端架构和逻辑控制有一定要求

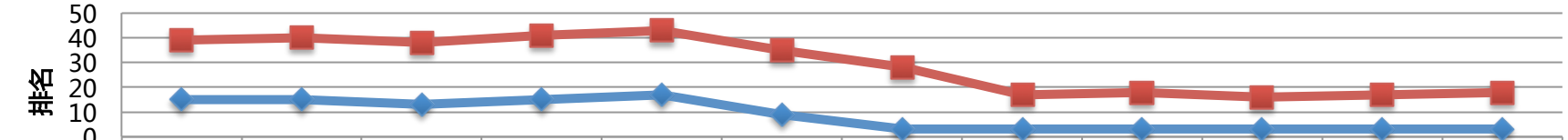
策略应用的注意事项：需控制预热的频率，强网下才启用。

大数据的性能优化策略解读(2)：页面预热效果验证过程



- 广东地区开启预热，整体T2变好了143ms，此时全网变好了约50ms，开启预热大约优化了93ms。

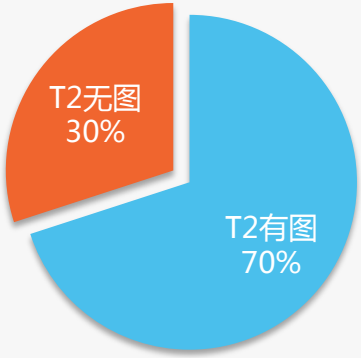
开启预热的省份性能排名变化情况

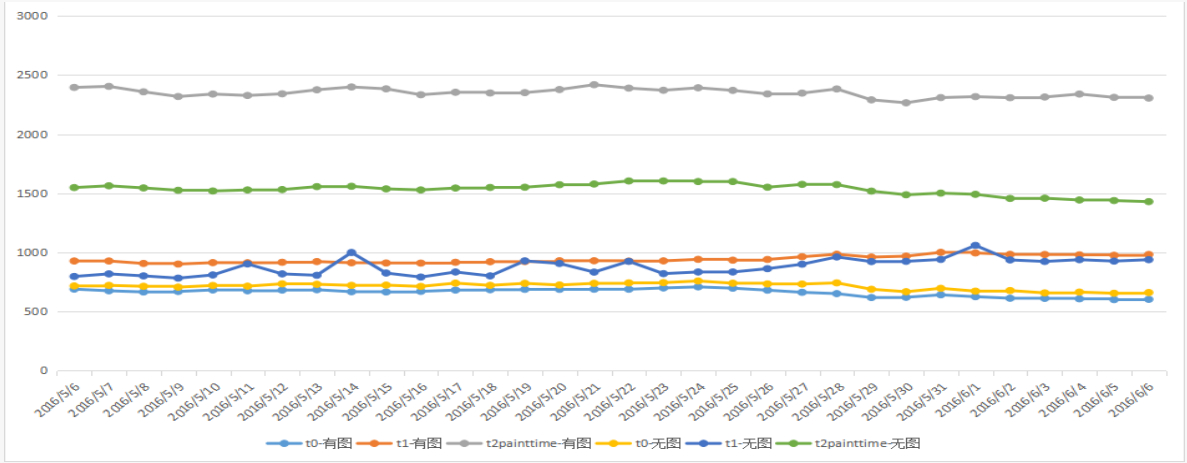


	2016.8.10	2016.8.11	2016.8.12	2016.8.13	2016.8.14	2016.8.15	2016.8.16	2016.8.17	2016.8.18	2016.8.19	2016.8.20	2016.8.21
广东	24	25	25	26	26	26	25	14	15	13	14	15
河南	15	15	13	15	17	9	3	3	3	3	3	3

大数据的性能优化策略解读(3)：正文首图的预加载和非懒加载

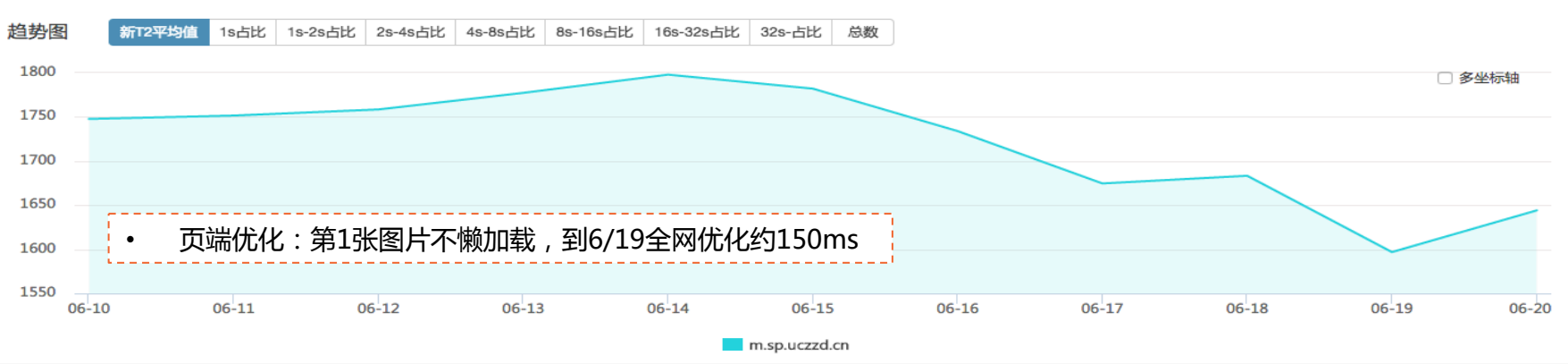
- **预加载**：在发起页面URL请求之前，先加载文章将要用到的图片。
- **非懒加载**：img标签插入到文档中立即发起图片请求，而不是用占位图片替代。



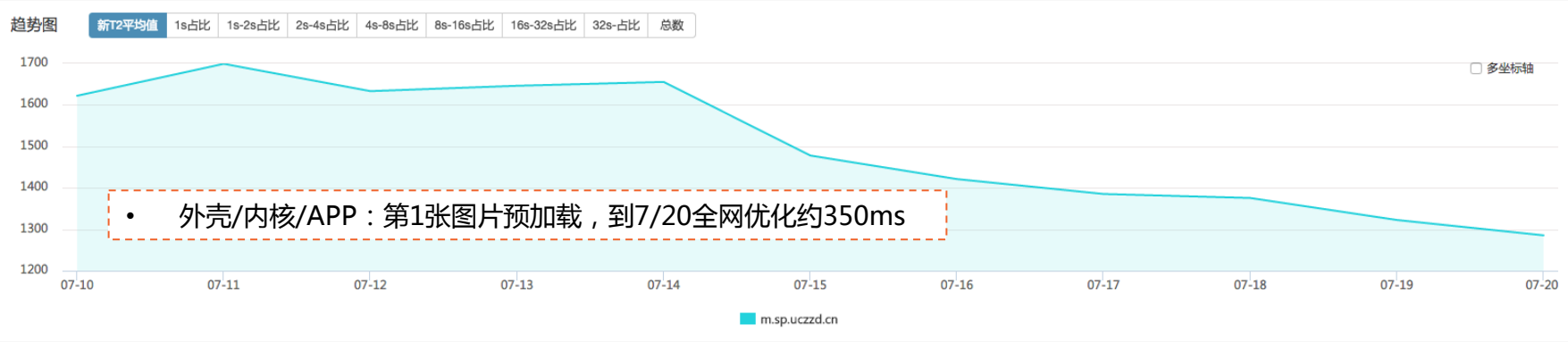


1. 首屏内含有图片的比例占大多数；
2. 在优化前，有图和无图的T2平均耗时差距约1秒。

大数据的性能优化策略解读(3)：预加载和非懒加载效果验证过程



- 页端优化：第1张图片不懒加载，到6/19全网优化约150ms



- 外壳/内核/APP：第1张图片预加载，到7/20全网优化约350ms

我们的优化策略解读

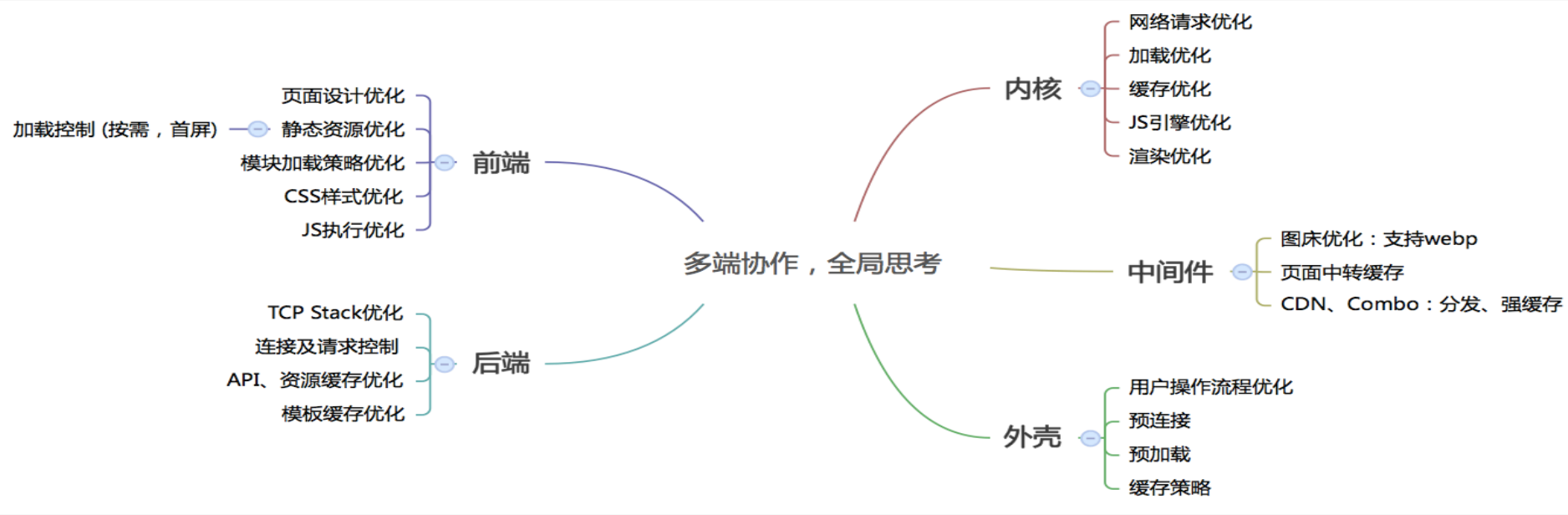
- 页面预热
- 文档预加载
- 图片预加载
- AJAX预请求

- JS、CSS、IMG等强缓存
- 主文档缓存
- 模块LS缓存
- ……

- 渲染拆分首屏和非首屏
- 根据手机屏幕及网络情况按需加载图片

- 非首屏的资源异步加载
- 非首屏的模块延时渲染

性能优化总结：多端协作、全局思考



- 任何的优化成果都应当以数据作为参考，并且优化策略应当准尊“度量 → 评估 → 上线”的流程。
- 前端开发者需要跳出WebView的范畴来审视页面优化，要学会从页面浏览所涉及的全路径上（后端、网络环境、前端、外壳、内核以及中间件等）进行全局性思考。

Q & A

THINKS !