

# QQ移动页面框架优化 实践

lovekidchen(陈志兴)/腾讯Android高级开发工程师

# lovekidchen(陈志兴)



2016-Now

Android开发  
个性化装扮  
QQ动漫  
**WebView提速**



2015-2016

Android开发  
文件收藏  
办公场景(TIM)



2007-2011

软件工程



2011-2015

PC客户端开发  
应用框架  
多终端消息  
文件传输质量优化



爱编码、爱钻研、也爱打游戏

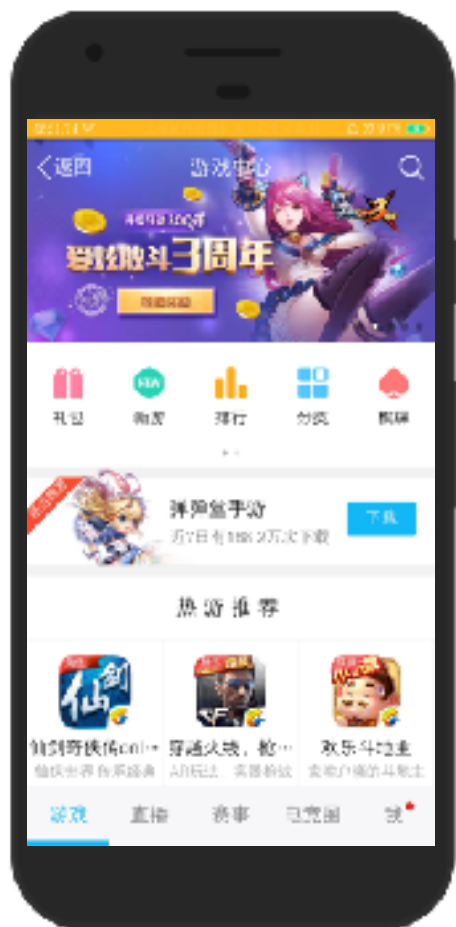
1. 传统页面的优化实践
2. 动态直出页面的优化实践(Sonic)
3. 关于移动页面框架的一点思考

1. 传统页面的优化实践

2. 动态直出页面的优化实践(Sonic)

3. 关于移动页面框架的一点思考

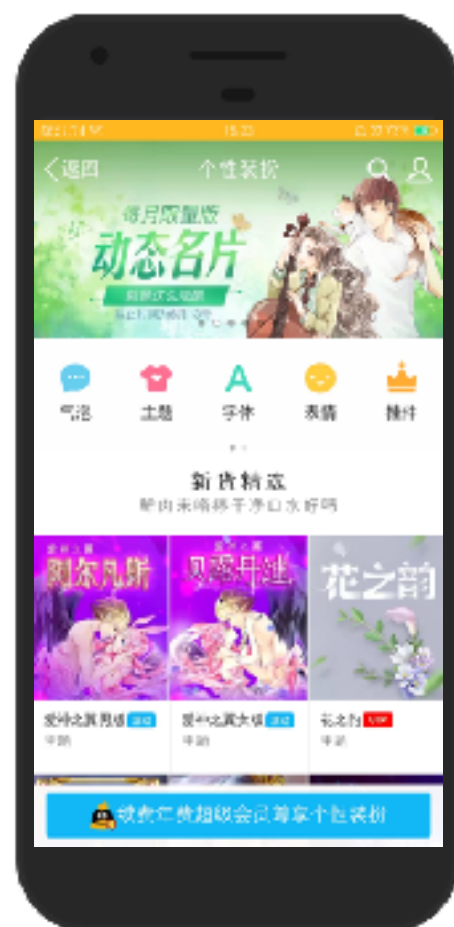
# 业务形态



游戏分发中心



会员特权中心



个性装扮商场



静态资源  
(html/js/css)

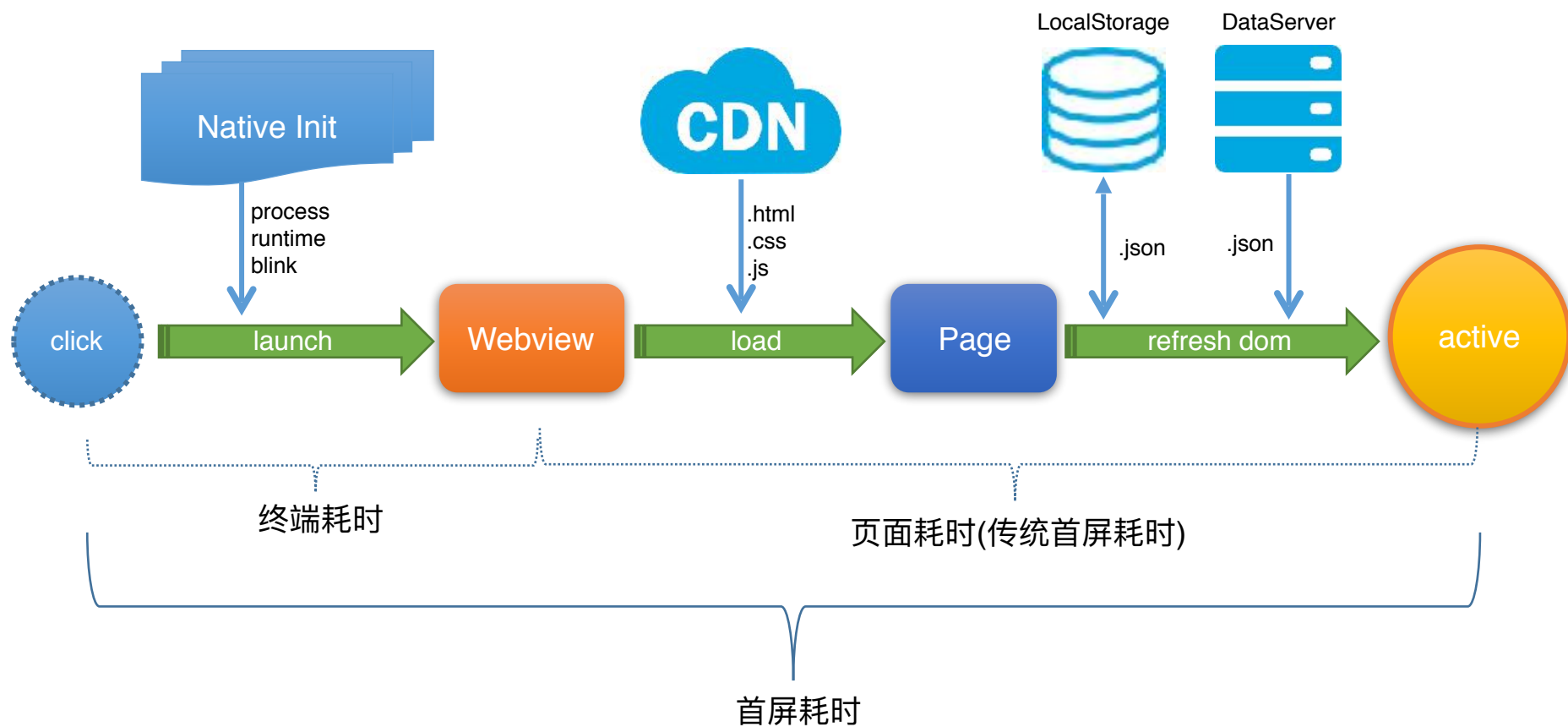


动态数据  
(items/banner)



DataServer

# 加载流程



# 存在的几个问题



终端耗时长

WebView请求资源时机慢



内容依赖网络

弱网络页面白屏时间长



数据动态拉取

首屏元素空白、刷新开销

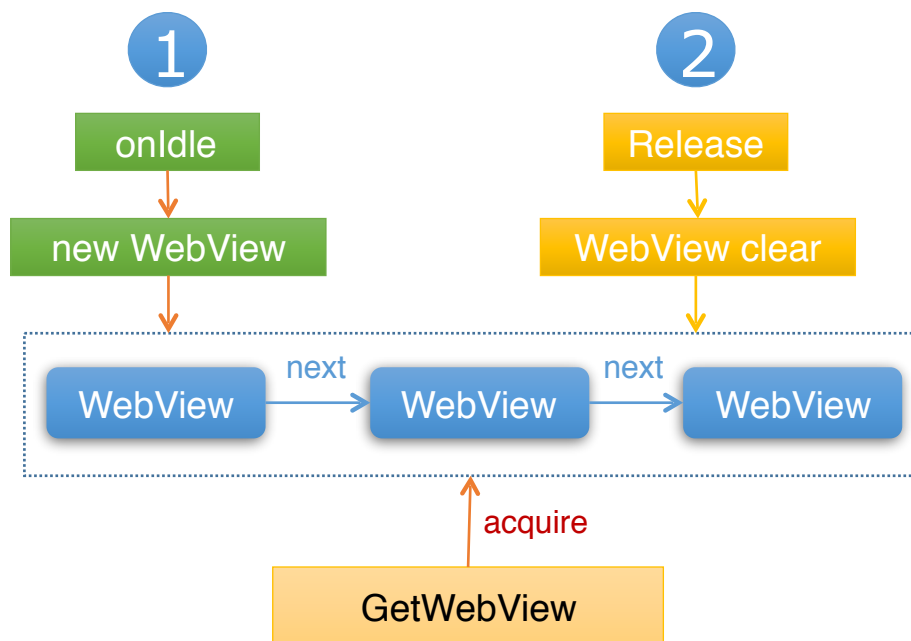




# 对应的优化方案

1. 终端耗时优化：启动流程优化、进程预加载、懒加载、IO异步化、**WebView池**等
2. 页面体验优化：图片自适应、**静态直出**
3. 网络请求优化：DNS预解析、**离线预推**

# WebView池



1 空闲预创建

2 回收复用



webview = new WebView(context)  
Context获取和释放问题?

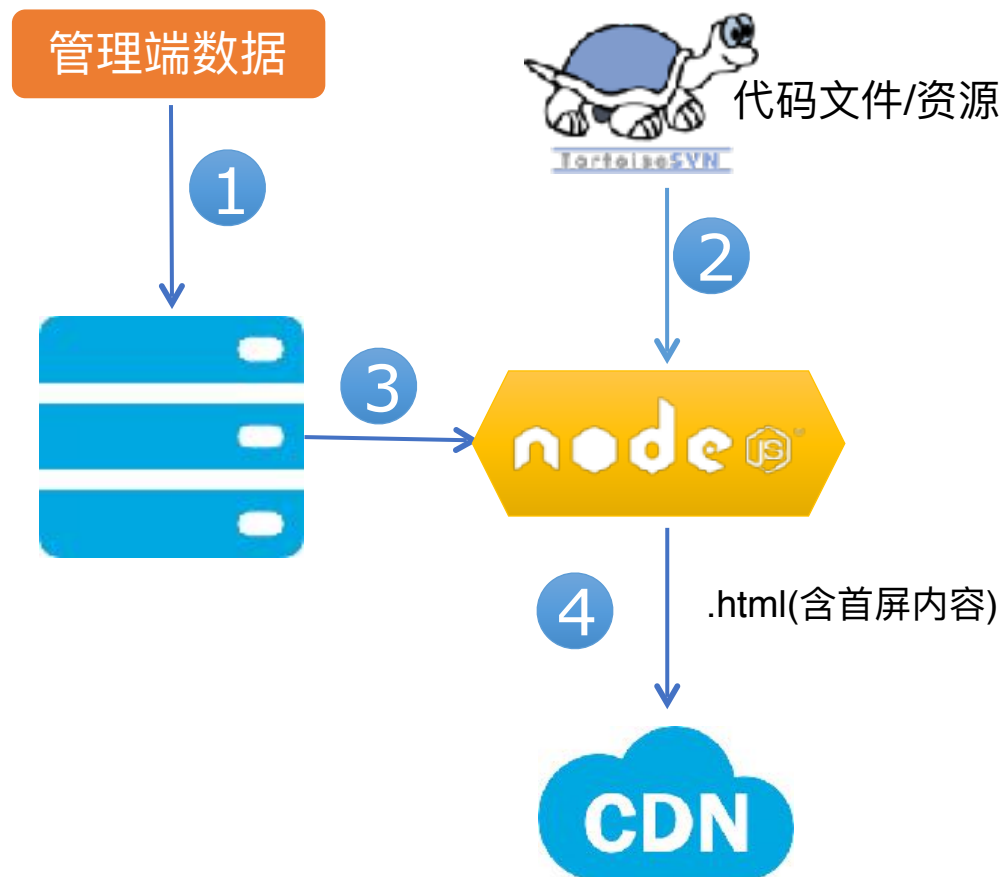
中间层  
ContextWrapper

# ReusableWebView.java

```
public static ReusableWebView acquire(Context context) {  
    ReusableWebView wv = acquireFromPool(); // acquire from pool  
    MutableContextWrapper ct;  
    if (null == wv) {  
        ct = new MutableContextWrapper(context);  
        wv = new ReuseWebView(ct);  
    } else {  
        ct =(MutableContextWrapper)wv.getContext();  
        ct.setBaseContext(context);  
    }  
    return wv;  
}  
  
public void destory() {  
    if (canReuse()) {  
        clearAndAddToPool(); // reset webview and add to pool  
        MutableContextWrapper ct;  
        ct =(MutableContextWrapper)getContext();  
        ct.setBaseContext(getApplication());  
    } else {  
        super.destory();  
    }  
}
```

# 静态直出

- 1 管理端发布数据
- 2 同步代码文件及资源
- 3 获取最新动态数据
- 4 首屏资源结合数据渲染生成Html并发布到CDN

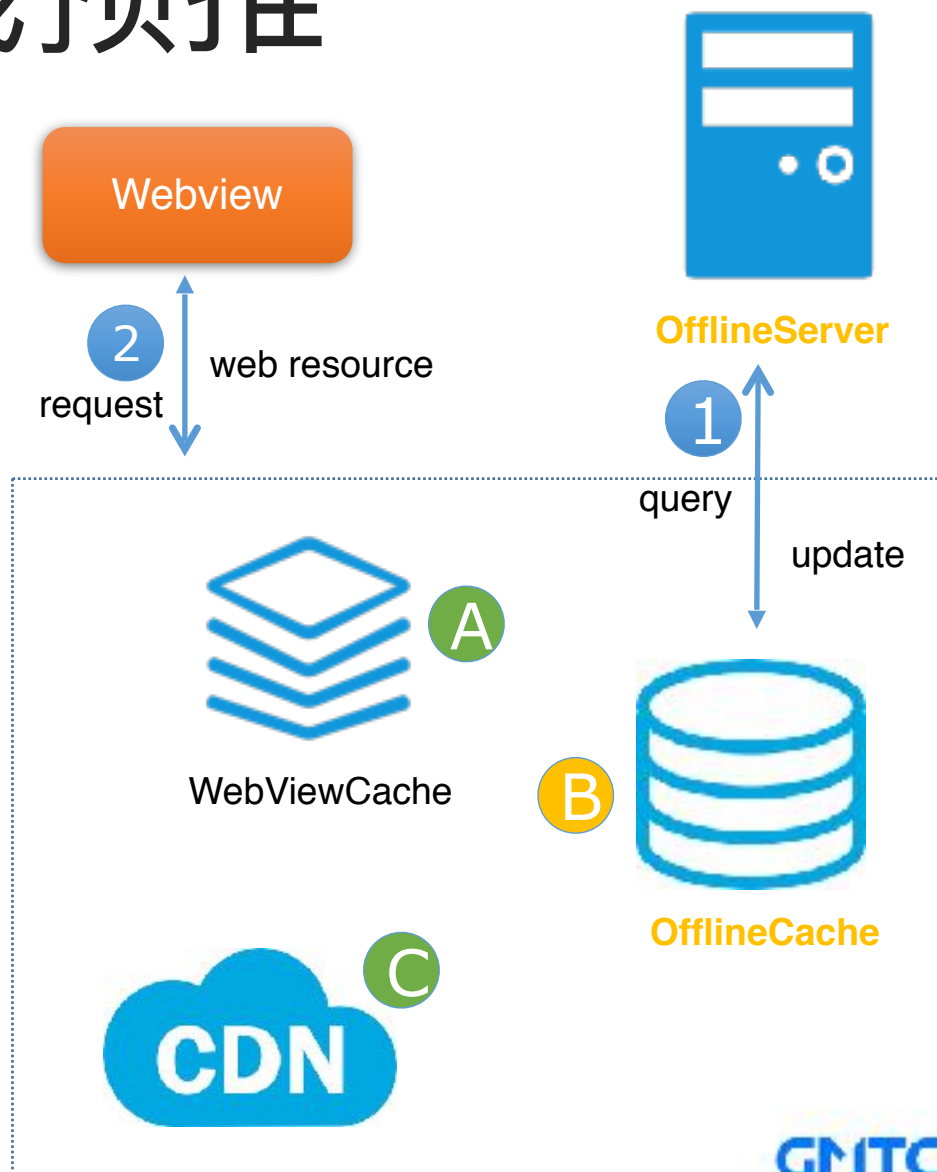


Venus自动构建系统

# 离线预推

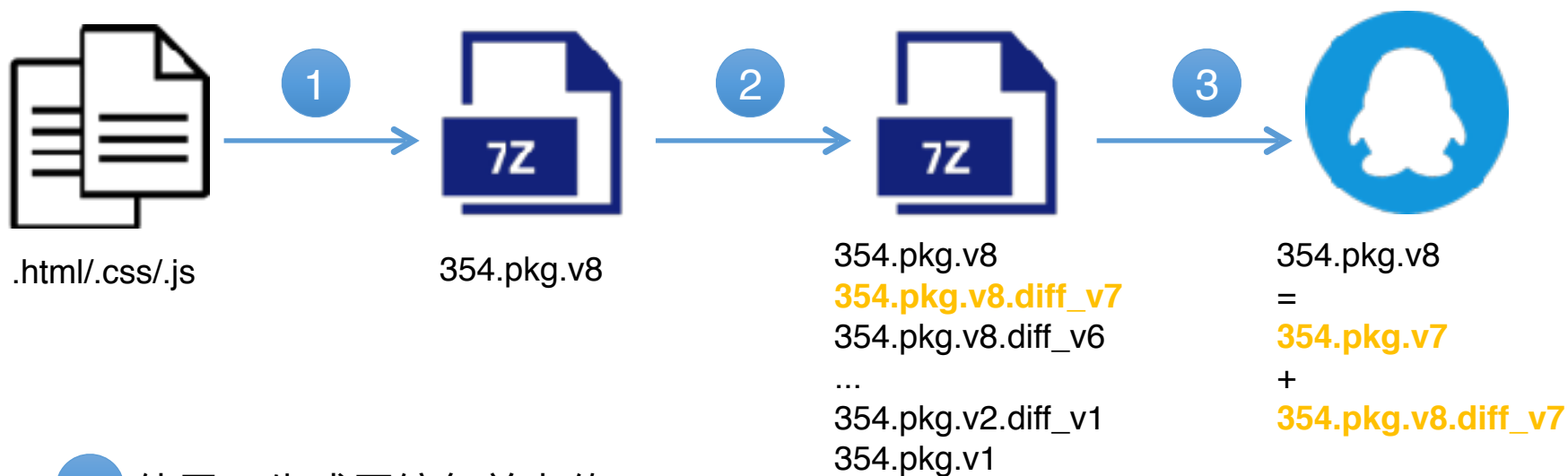
- 1 登陆查询并更新离线包
- 2 WebView加载页面资源
- A 一级缓存: WebView内核缓存
- B 二级缓存: 离线包资源拦截
- C 网络请求: CDN

? 带宽过大?



# 带宽优化

e.g. [https://cdn.vip.qq.com/.../index?\\_bid=354](https://cdn.vip.qq.com/.../index?_bid=354)



1 使用7z生成压缩包并上传

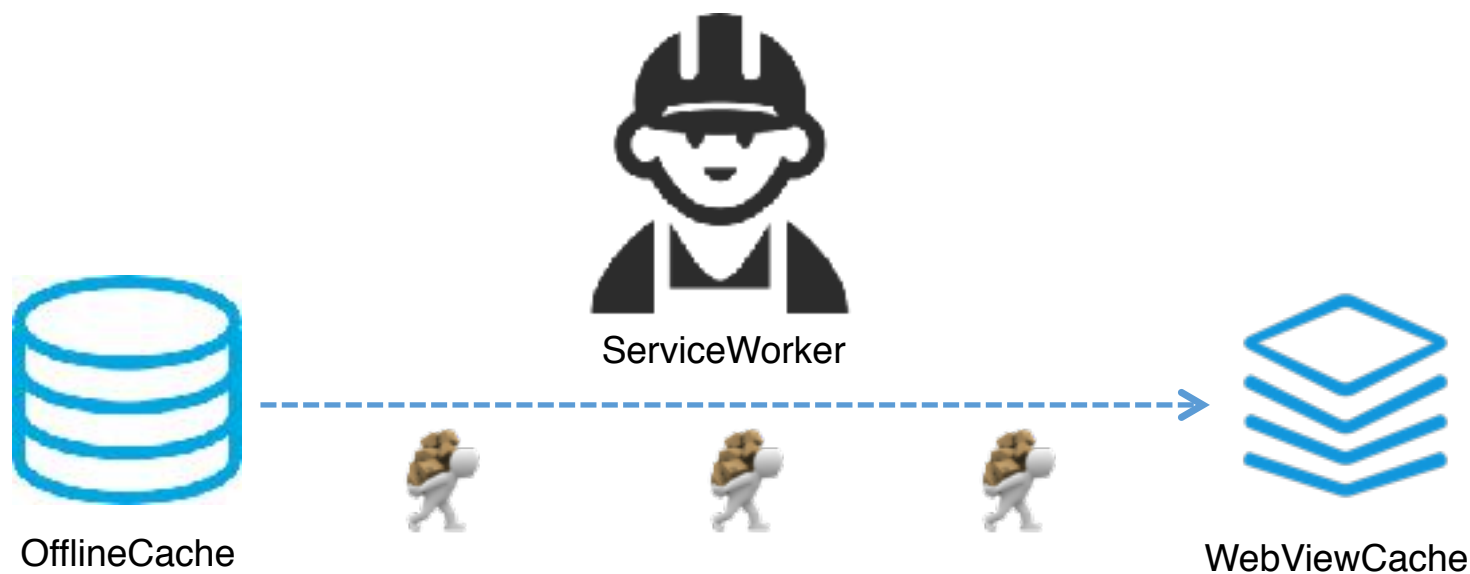
2 使用Bsdiff生成增量包


3 增量更新并合并还原



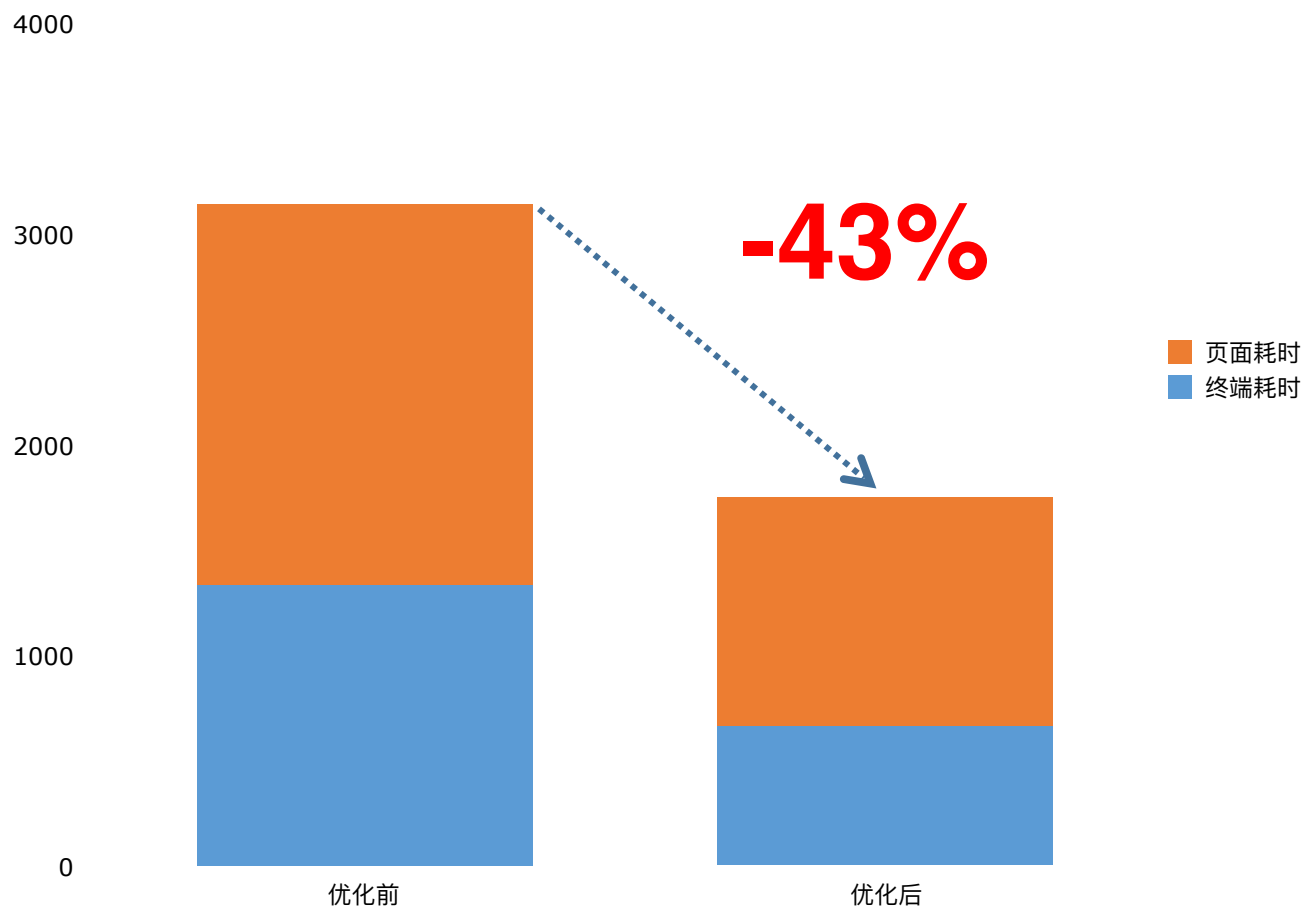
**253kb -> 3kb**

# 让缓存读取更快



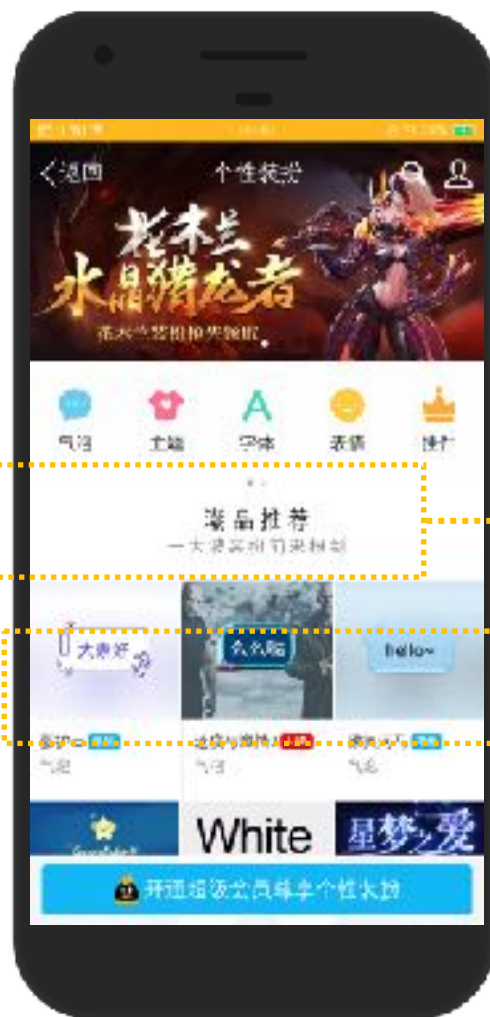
 **+17%**

# 优化数据对比





# 业务变化： 个性化推荐



1 用户画像推荐

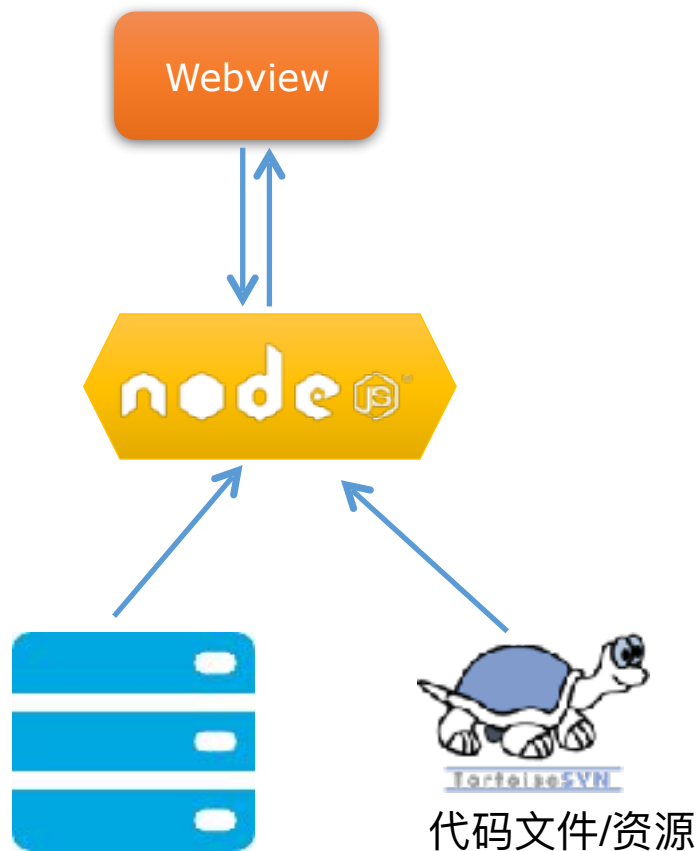
2 随机算法推荐

1. 传统页面的优化实践

2. 动态直出页面的优化实践(Sonic)

3. 关于移动页面框架的一点思考

# 动态直出



## ? 遇到的问题:

- 1 服务端实时拉取数据渲染耗时长
- 2 首屏无法使用离线预推等缓存策略

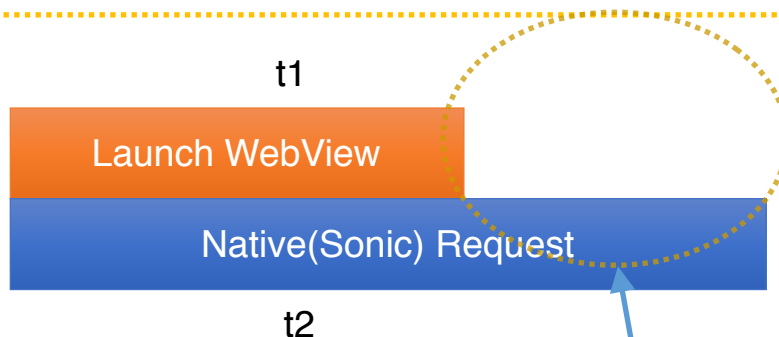
# 并行请求资源

默认模式:



耗时:  $\text{sum}(t1, t2)$ , 网络空等时间: **t1**

并行模式:

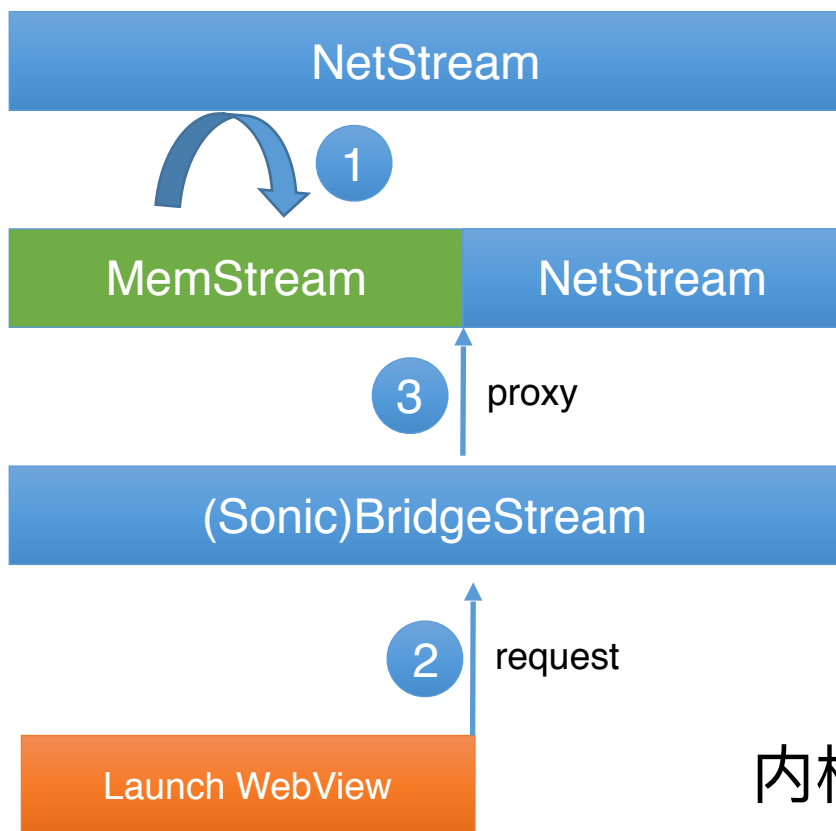


耗时:  $\text{max}(t1, t2)$ , 网络空等时间: 0



内核空等时间:  $t2 - t1$ ?

# 流式拦截

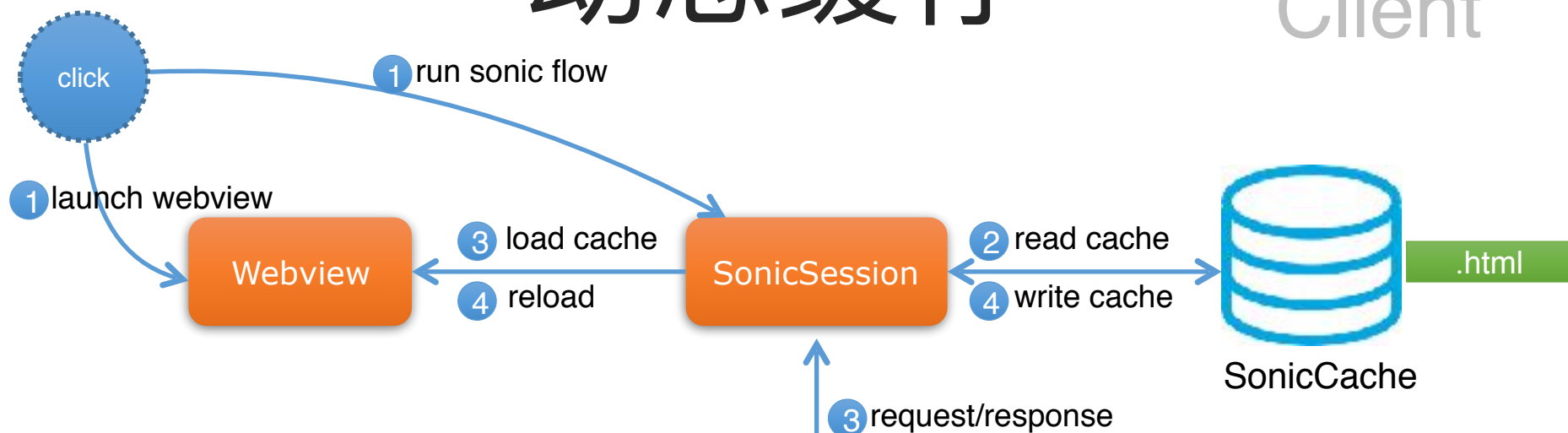


- 1 子线程读取网络流到内存
- 2 WebView开始请求资源
- 3 构造桥接流塞给内核加载

内核无需空等，边加载边解析

# 动态缓存

Client



response -> .html

Server

? 重刷白闪体验差



代码文件/资源

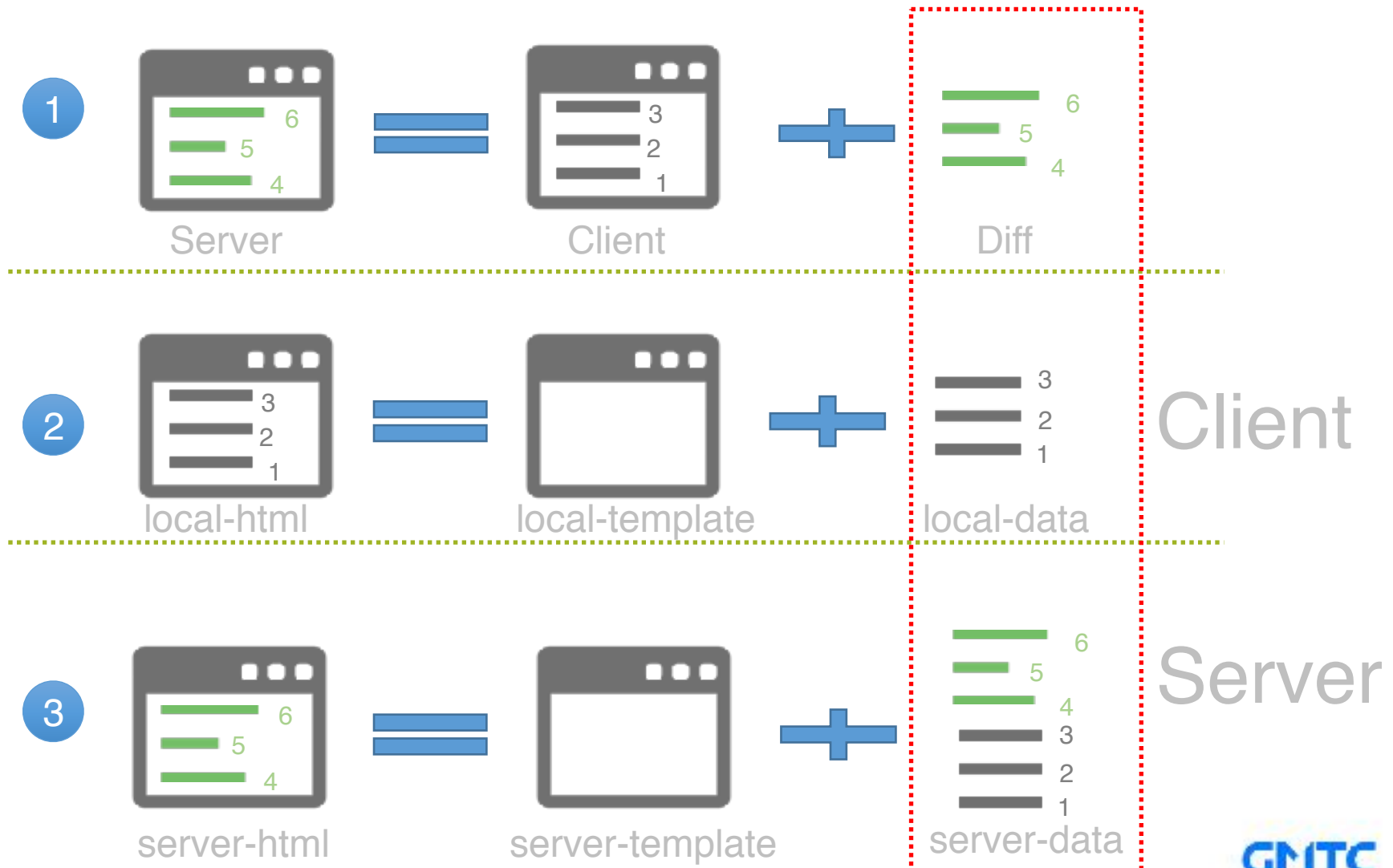
# 增量更新?

很久很久以前...

客户端是这么实现增量更新的



# 如何获取增量数据？





# 格式规范协商

```
<!DOCTYPE html>
<html>
...
  <ul class="banners-wrap" id="albums">
    <!--sonicdiff-albums-begin-->
    <li data-id="0" data-name="qqfamily"
data-image="//
7405227df42509f81fb.png?max_age=31536000"
class="banner-item" data-handler="openUrl"></li>
    <!--sonicdiff-albums-end-->
  </ul>
</section>
...
</html>
```

页面

```
<!DOCTYPE html>
<html>
...
  <ul class="banners-wrap"
id="albums">
    {albums}
  </ul>
</section>
...
</html>
```

模板

```
"albums":
"
  <!--sonicdiff-albums-begin-->
  <li data-id="0" data-name="qqfamily"
data-image="//
7405227df42509f81fb.png?max_age=31536000"
class="banner-item" data-handler="openUrl"></li>

  <!--sonicdiff-albums-end-->"
```

数据

```
method: GET
accept-Charset: utf-8
accept-diff: true
If-none-match:
017764c633893984bfc0a4e22c94b3e947d73027
template-tag:
9ba2a9c1dbb4a51bb5c648553a84ce50d11b7bd1
sonic-sdk-version: Sonic/1.1
```

协议头部扩展

# 首次加载

## 请求头部

```
method: GET
accept-Charset: utf-8
accept-diff: true
sonic-sdk-version: Sonic/1.1
template-tag: ""
...
```

## 响应头部

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: TSW/Node.js
Cache-Control: no-cache
Vary: Accept-Encoding
Content-Encoding: gzip
Etag: d5a74e5230b4860d5a2f242d2b4ded91c49b709
template-tag:
9ba2a9c1dbb4a51bb5c648553a84ce50d11b7bd1
Cache-Offline: true
template-change: true
```

## 响应包体

```
<!DOCTYPE html>
<html>
...
  <ul class="banners-wrap" id="albums">
    <!--sonicdiff-albums-begin-->
    <li data-id="0" data-
name="qqfamily" data-image="//
gxh.vip.qq.com/xydata/vipData/app/
gxhIndex2016/
cfa0c164087407405227df42509f81fb.png?max_a
ge=31536000" class="banner-item" data-
handler="openUrl" data-url="http://
zb.vip.qq.com/
ip?_wv=16778241&id=101229"></li>
    <!--sonicdiff-albums-end-->
  </ul>
</section>
...
</html>
```



.html

.tpl

.data

# 第二次加载

## 请求头部

```
method: GET
accept-Charset: utf-8
accept-diff: true
if-none-match:
017764c633893984bfc0a4e22c94b3e947d73027
template-tag:
9ba2a9c1dbb4a51bb5c648553a84ce50d11b7bd1
sonic-sdk-version: Sonic/1.1
...
```

## 响应头部

```
HTTP/1.1 200 OK
Content-Type: textml; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: TSW/Node.js
Cache-Control: no-cache
Content-Encoding: gzip
Etag: d5a74e5230b4860d5a2f242d2b4ded91c49b7098
template-tag:
9ba2a9c1dbb4a51bb5c648553a84ce50d11b7bd1
Cache-Offline: true
template-change: false
```

## 响应包体

```
{
  "data": {
    "{title}":
    "<!--sonicdiff-title-begin-->个性化装扮<!--
sonicdiff-title-end-->",
    "{albums}":
    "<!--sonicdiff-albums-begin-->
<li data-id='0' data-name='qqfamily' data-
image='//gxh.vip.qq.com/xydata/vipData/app/
gxhIndex2016/
cfa0c164087407405227df42509f81fb.png?
max_age=31536000' class='banner-item' data-
handler='openUrl' data-url='http://
zb.vip.qq.com/ip?_wv=16778241&id=101229'></
li>
<!--sonicdiff-albums-end-->",
    "{hot}":
    "<!--sonicdiff-hot-begin-->...<!--
sonicdiff-hot-end-->",
    ...
  }
}
```

 20Kb -> 5Kb

# 增量数据

## 本地数据

```
{title}:  
"<!--sonicdiff-title-begin>个性装扮</!--sonicdiff-  
title-end-->"
```

```
{albums}:  
"<!--sonicdiff-albums-begin-->XXX<!--sonic-  
diff-albums-end-->"
```

```
{activeList}:  
"<!--sonicdiff-activeList-begin-->..."<!--  
sonicdiff-activeList-end-->
```

```
{redPoint}:  
"<!--sonicdiff-redPoint-begin-->...<!--  
sonicdiff-redPoint-end-->"
```

## 最新数据

```
{title}:  
"<!--sonicdiff-title-begin>个性装扮</!--  
sonicdiff-title-end-->"
```

```
{albums}:  
"<!--sonicdiff-albums-begin-->YYY<!--sonic-  
diff-albums-end-->"
```

```
{activeList}:  
"<!--sonicdiff-activeList-begin-->..."<!--  
sonicdiff-activeList-end-->
```

```
{redPoint}:  
"<!--sonicdiff-redPoint-begin-->...<!--  
sonicdiff-redPoint-end-->"
```

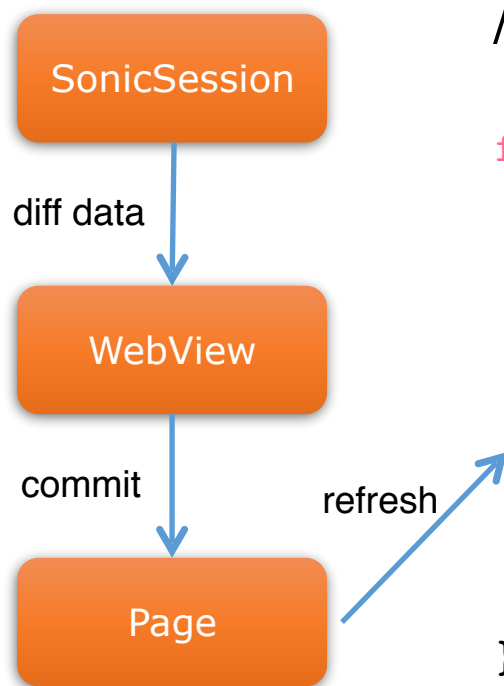
## 差异数据

```
{albums}:  
"<!--sonicdiff-albums-begin-->YYY<!--sonic-diff-albums-end--  
>"
```



< 5Kb

# 提交增量数据



// sonic.js

```
function refresh(data) {
    var result = JSON.parse(data.result);
    for (var key in result) {
        var html = result[key];
        var id = key.match(/\{(\w+)\}/).pop();
        var target = document.getElementById(id)
        if (target) {
            target.innerHTML = html;
        }
    }
}
```

# 动态数据合并

## 模板

```
<!DOCTYPE html>
<html>
...
  <ul class="banners-wrap" id="albums">
    {albums}
  </ul>
</section>
...
</html>
```

## 新数据

```
{title}:
"<title>个性装扮</title>"

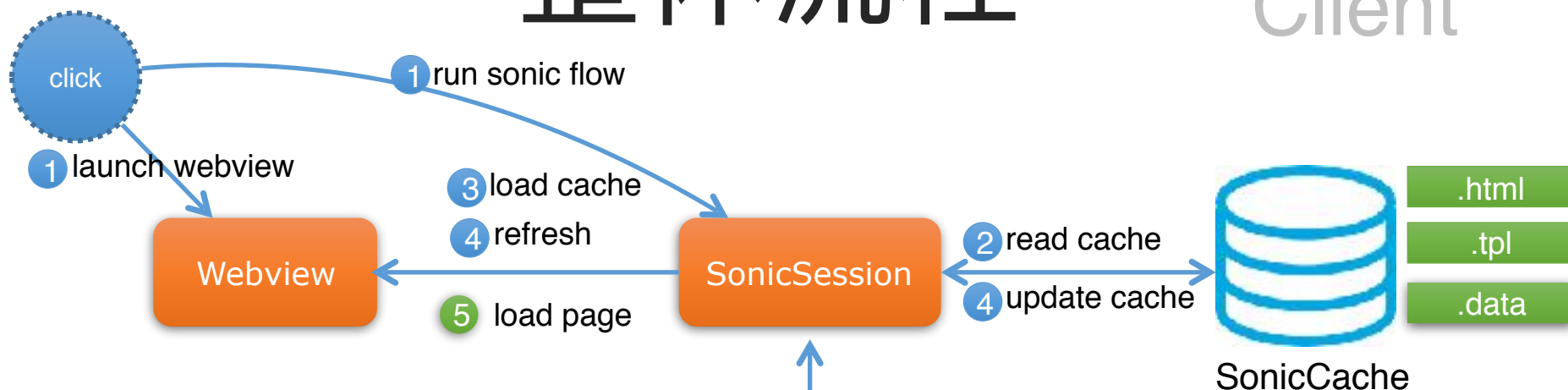
{albums}:
"<!--sonicdiff-albums-begin-->YYY<!--sonicdiff-albums-end-->"
```

## 新页面

```
<!DOCTYPE html>
<html>
...
  <ul class="banners-wrap" id="albums">
    <!--sonicdiff-albums-begin-->YYY<!--sonicdiff-albums-end-->
  </ul>
</section>
...
</html>
```

# 整体流程

Client



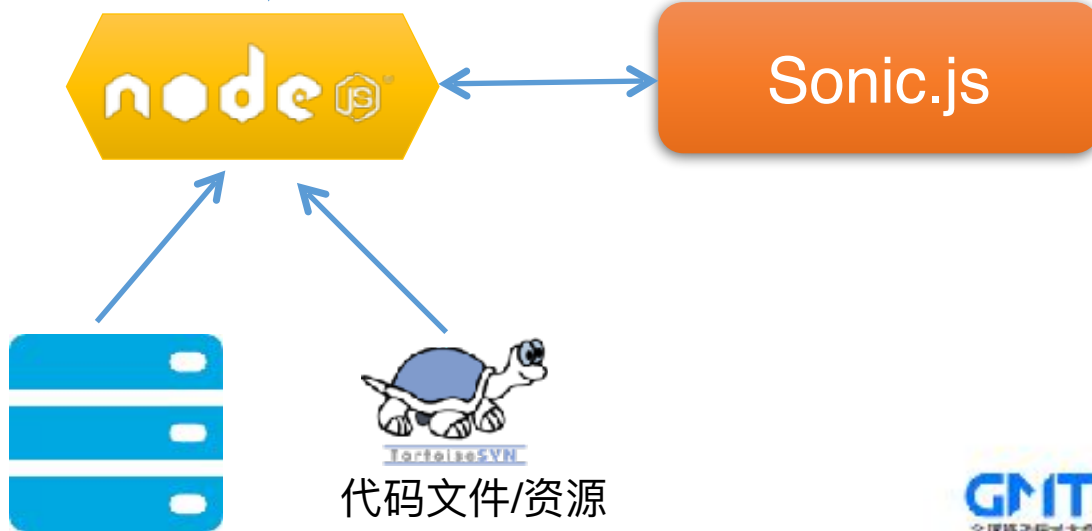
request-> template-tag:xxx

3 request/response

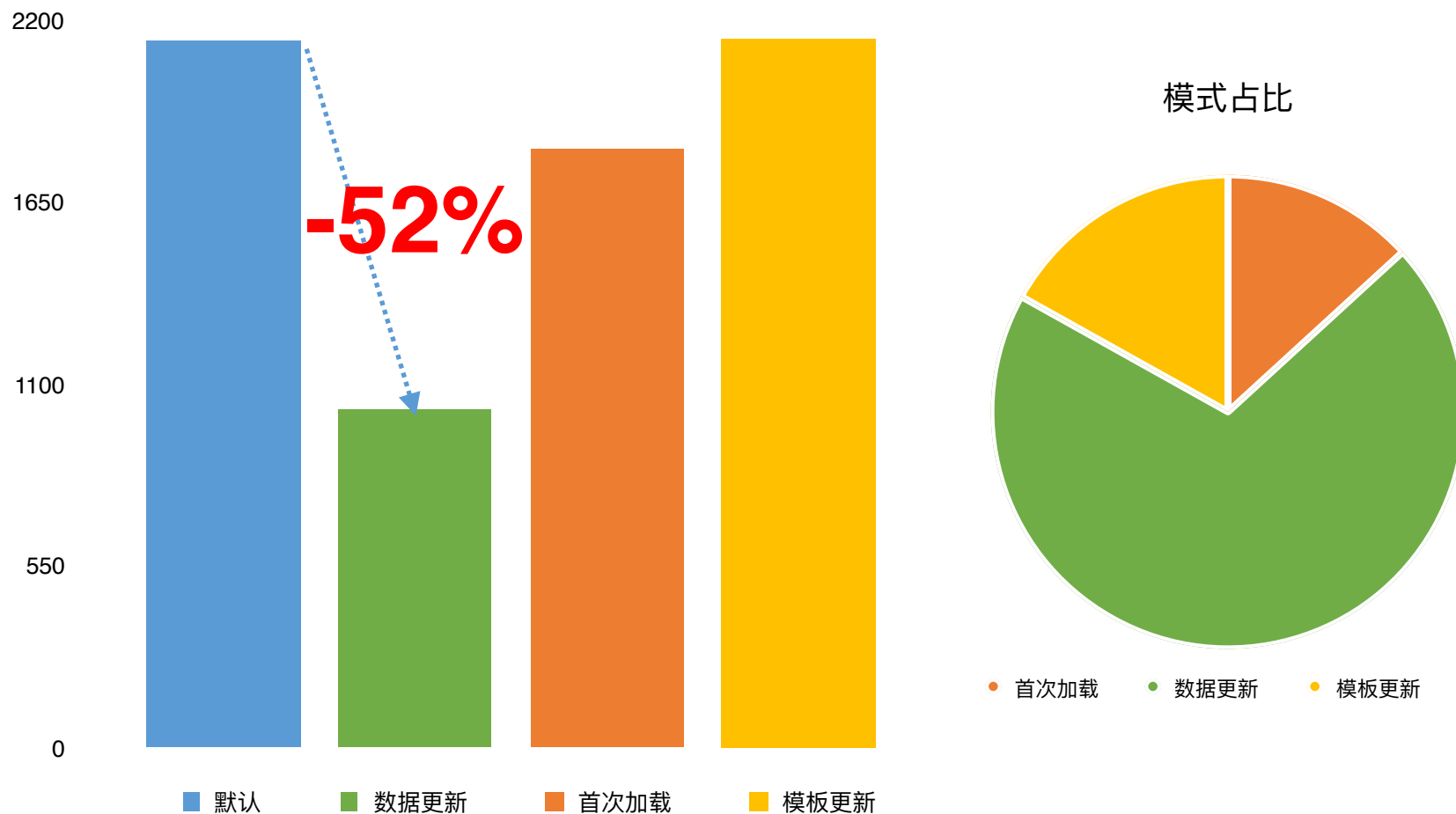
response -> data or html

html -> template+data

Server



# 外网首屏数据

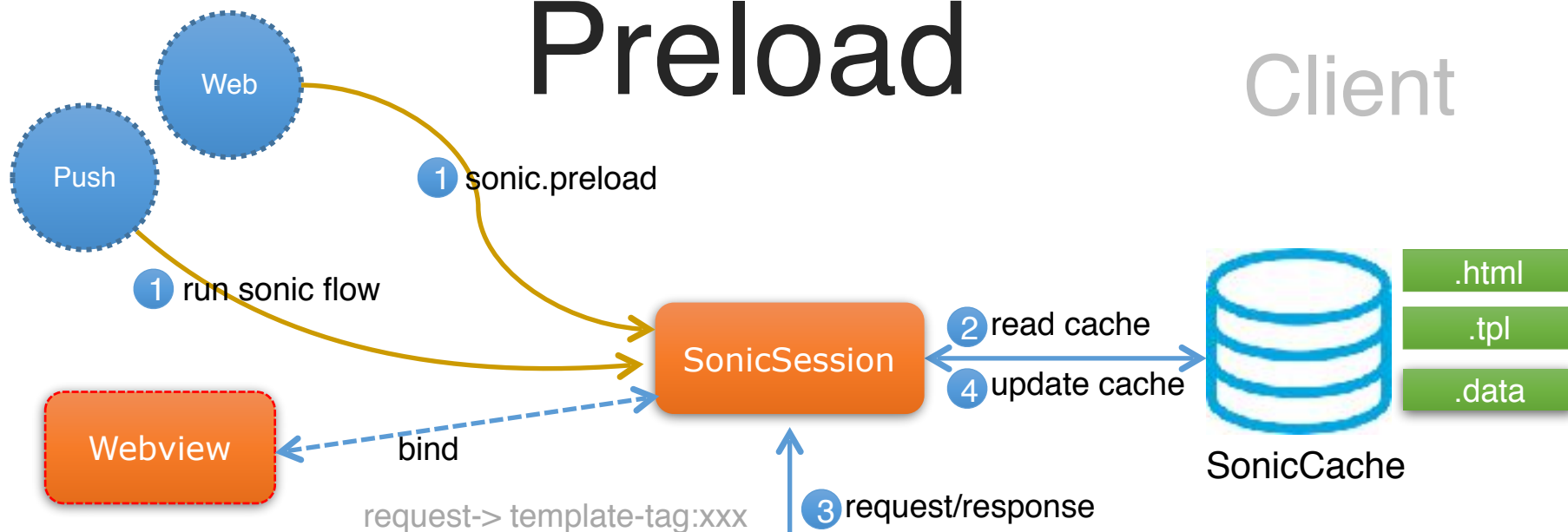


数据来源：外网Android耗时统计



# Preload

Client



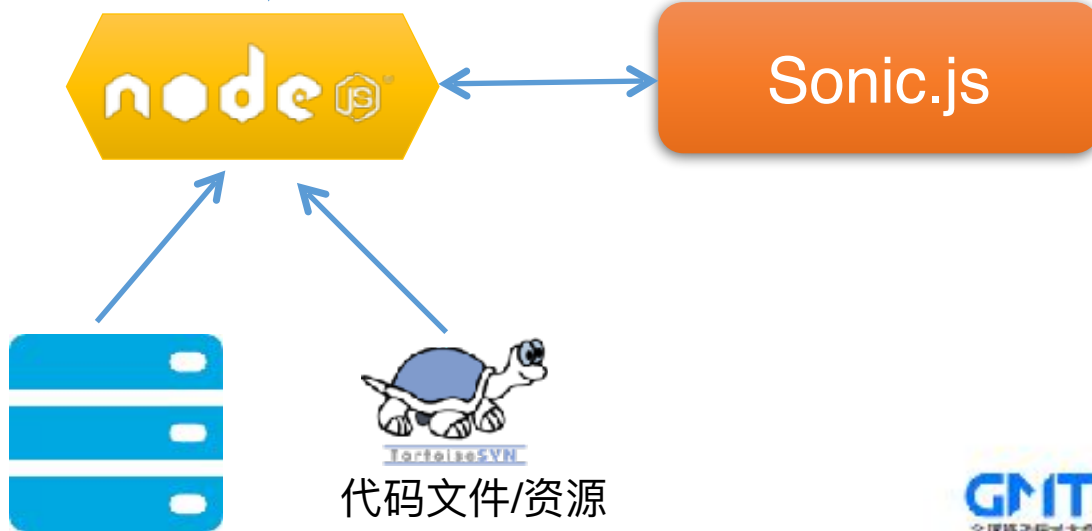
request-> template-tag:xxx

3 request/response

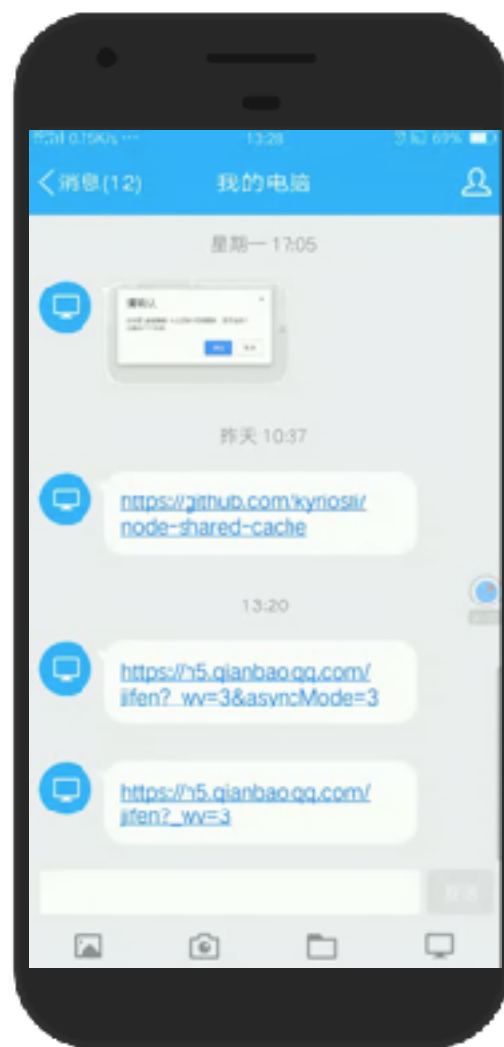
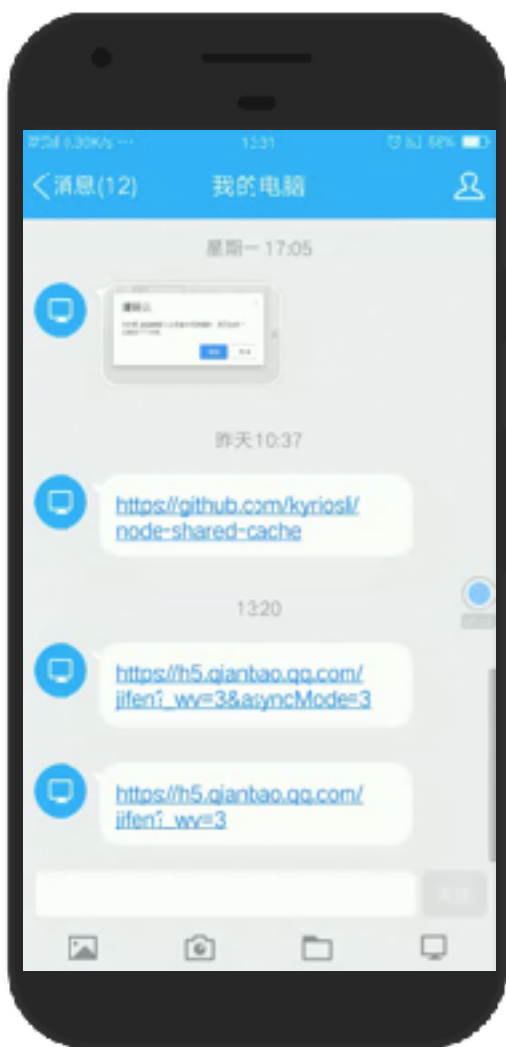
response -> data or html

html -> template+data

Server



# 效果对比



设备：OPPO R9s 网络：3G

# Sonic Open-Source



PV>8000W

并行请求

流式拦截

增量更新

预加载

自定义通道

IP直连

Sonic Sdk (Android/iOS/JS)

“一起打造一个更快、更小的Hybrid框架”

1. 传统页面的优化实践

2. 动态直出页面的优化实践(Sonic)

3. 关于移动页面框架的一点思考

# Web比Native慢在哪?



## 资源加载

资源存放在远端、异步拉取



## 脚本语言

弱类型、Just-In-Time



## 渲染机制

渲染管道长、缓存粒度不够细

PWA  
sonic  
offlinePkg

...

v8  
TypeScript  
webAssembly

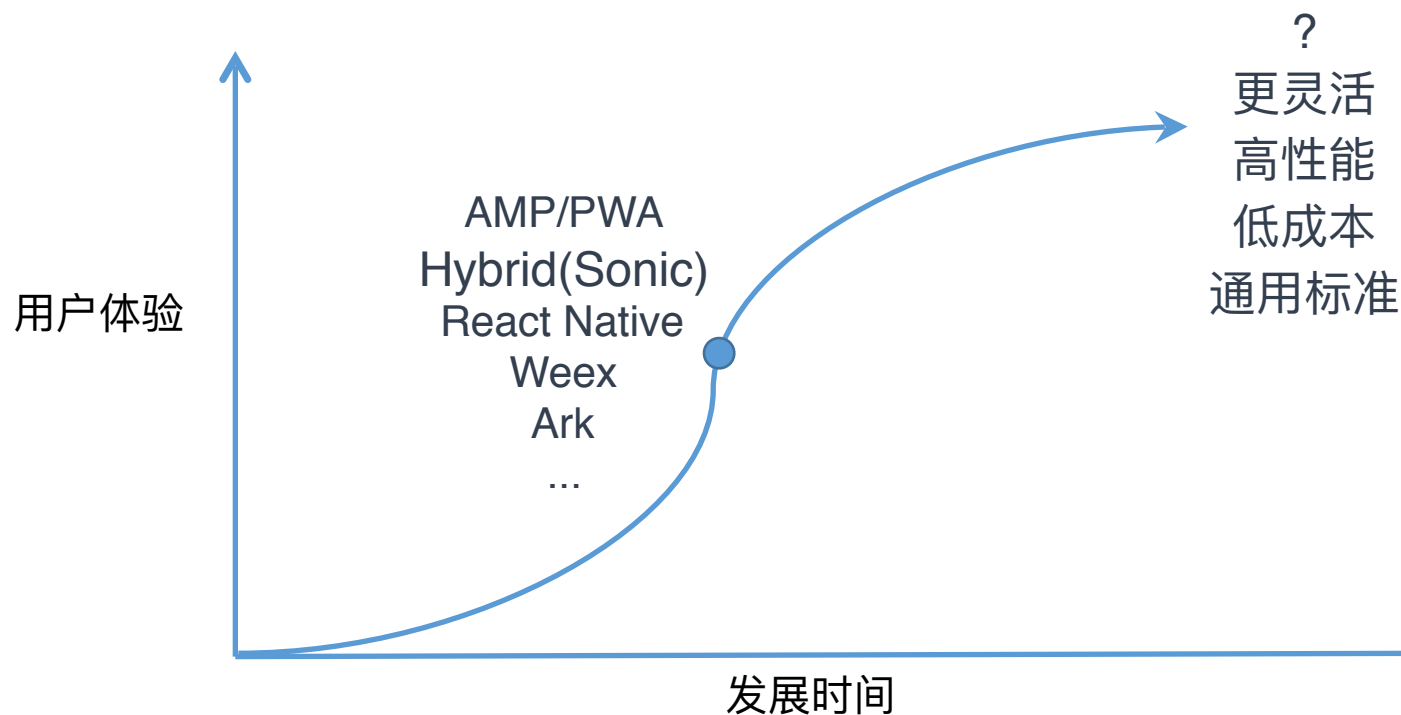
...

blink-sliming-paint  
feedback features  
React Native

...

# 选择最合适你业务的框架

“We are in the midst of a sea change for the mobile web”



# 更多交流



小时光茶社



我

# THANKS!

