



PWA



带来的 极速离线 Web 应用

腾讯 IVWEB 成员



H5 RTMP 播放器作者



个人博客：villainhr.com



Now 直播核心开发



公众号：前端小吉米

PWA 尝鲜

PWA 技术涉及的基础内容

离线缓存

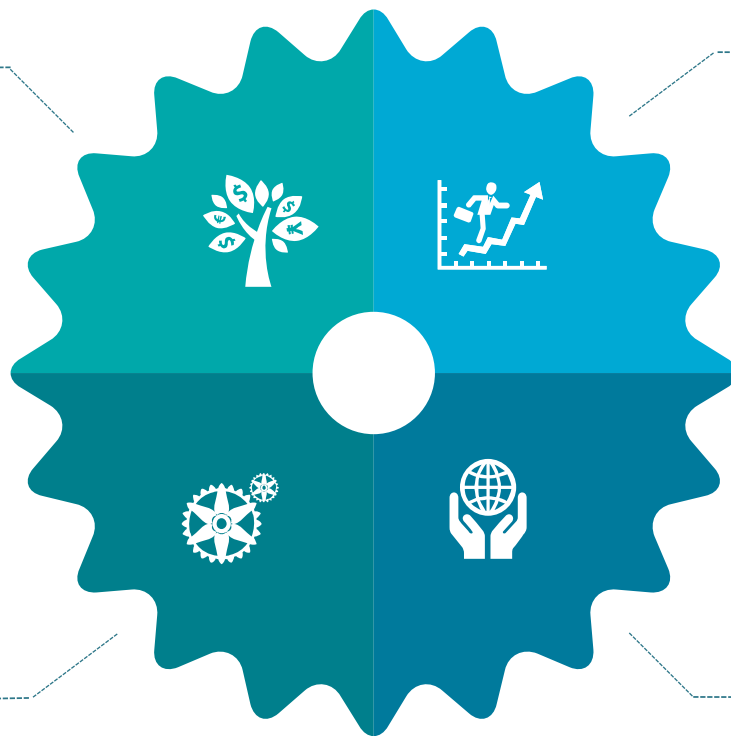
总结实践过程中，SW 加载时延和离线包的对比。

SW 离线包策略

结合多种离线数据库，制定离线策略方法和方案

未来规划

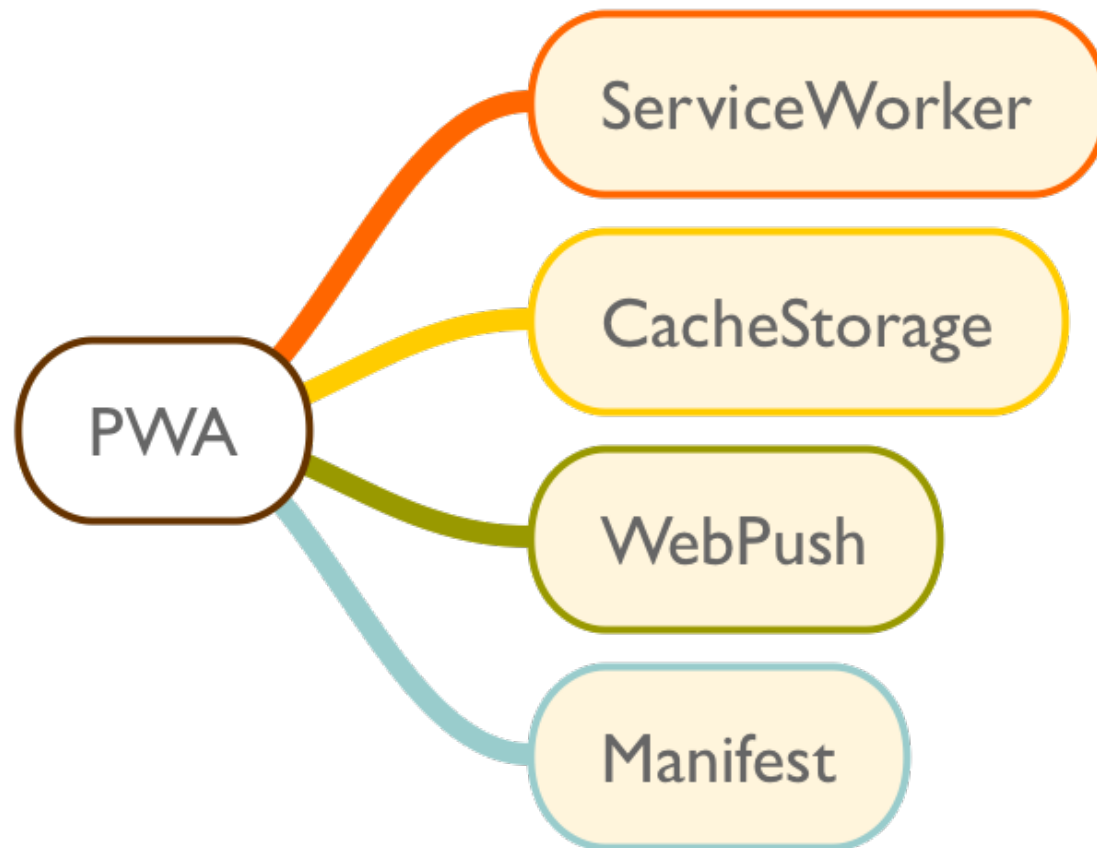
SW 离线库优化和便捷库的提供





1. PWA 尝鲜

- Service Worker：运行 PWA 的骨架，基于 worker
- Push：像 APP 一样实现推送
- Cache：替代以前的 manifest，实现灵活缓存
- Manifest: 用来生成桌面 ICON





1.1 SW feature

- 监听页面请求 类似于 Fiddler *
- 接受后台 Push *
- 管理离线 Cache 资源
- 新开/聚焦页面

Size	Waterfall			
Push Messages	对比分析			
	qb	tbs	chrome	webview
	6.2-6.3	2.0	37	5.0
	6.4-6.6	2.1	38	5.1
	6.7-6.9	2.2	39	6.0
	7.0-7.3	2.3-2.6	40	7.0
	7.4	3.0-3.1	41	
			42	



1.1 SW feature

- 监听页面请求 类似于 Fiddler
- 接受后台 Push
- 管理离线 Cache 资源 *
- 新开/聚焦页面

Storage

- ▶ Local Storage
- ▶ Session Storage
- ▶ IndexedDB
- Web SQL
- ▶ Cookies

Cache

- ▼ Cache Storage
 - mysite-version-1 - https://now.qq.com
- Application Cache



1.1 SW feature

- 监听页面请求 类似于 Fiddler
- 接受后台 Push
- 管理离线 Cache 资源
- 新开/聚焦页面 *

```
function focusOpen(){
var url = location.href;
  clients.matchAll({
    type: 'window',
    includeUncontrolled: true
  }).then(clients=>{
    for(var client of clients){
      if(client.url = url) return client.focus();
    }
    console.log('not focus');
    clients.openWindow(location.origin);
  })
}
```



1.2 SW 难点

- 基于 Worker
- 域名坑点
 - 严格同域 *
 - 只支持 https *
 - 调试域 localhost
 - 支持使用 fiddler or whistle 本地映射 sw.js
- 生命周期
 - 初始注册
 - 脚本更新
 - 终止生效





1.2 SW 难点

- 基于 Worker
- 域名坑点
 - 严格同域
 - 只支持 https
 - 调试域 localhost *
 - 支持使用 fiddler or whistle 本地映射 sw.js *
- 生命周期
 - 初始注册
 - 脚本更新
 - 终止生效

```
/^https?:\\now\\.qq\\.com\\.*sw\\.js$/
```

```
Rule: file:///pathToName/sw.js
```

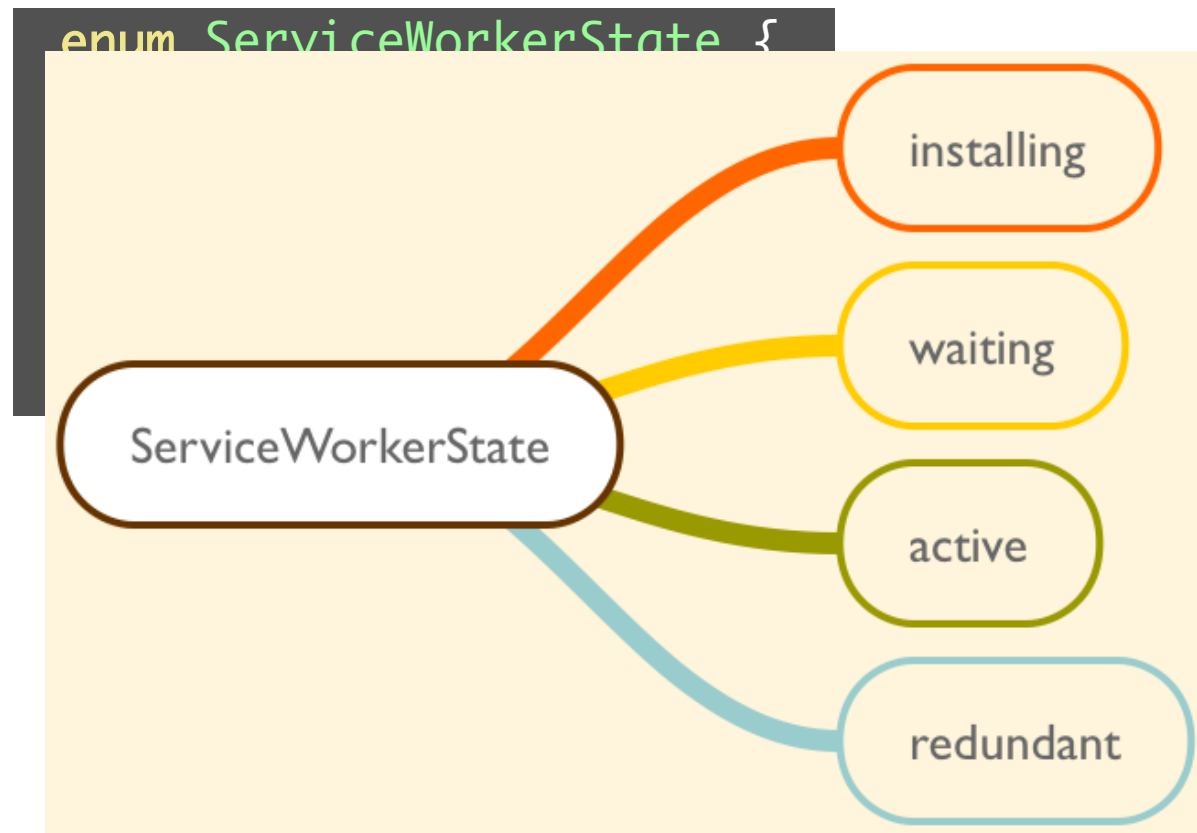
Or

```
Rule: 10.100.64.222
```



1.2 SW 难点

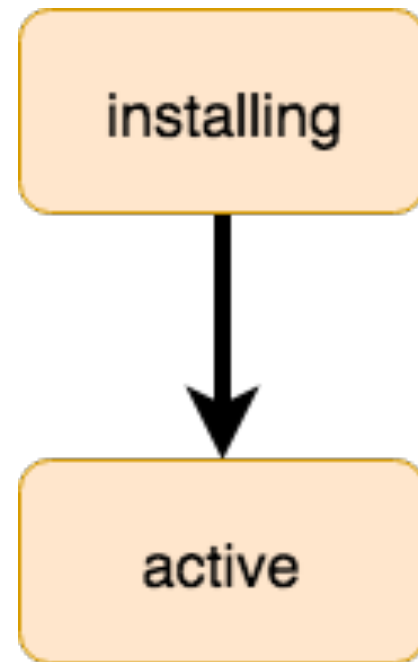
- 基于 Worker
- 域名坑点
 - 严格同域
 - 只支持 https
 - 调试域 localhost
 - 支持使用 fiddler or whistle 本地映射 sw.js
- 生命周期 *
 - 初始注册
 - 脚本更新
 - 终止生效





1.2 SW 难点

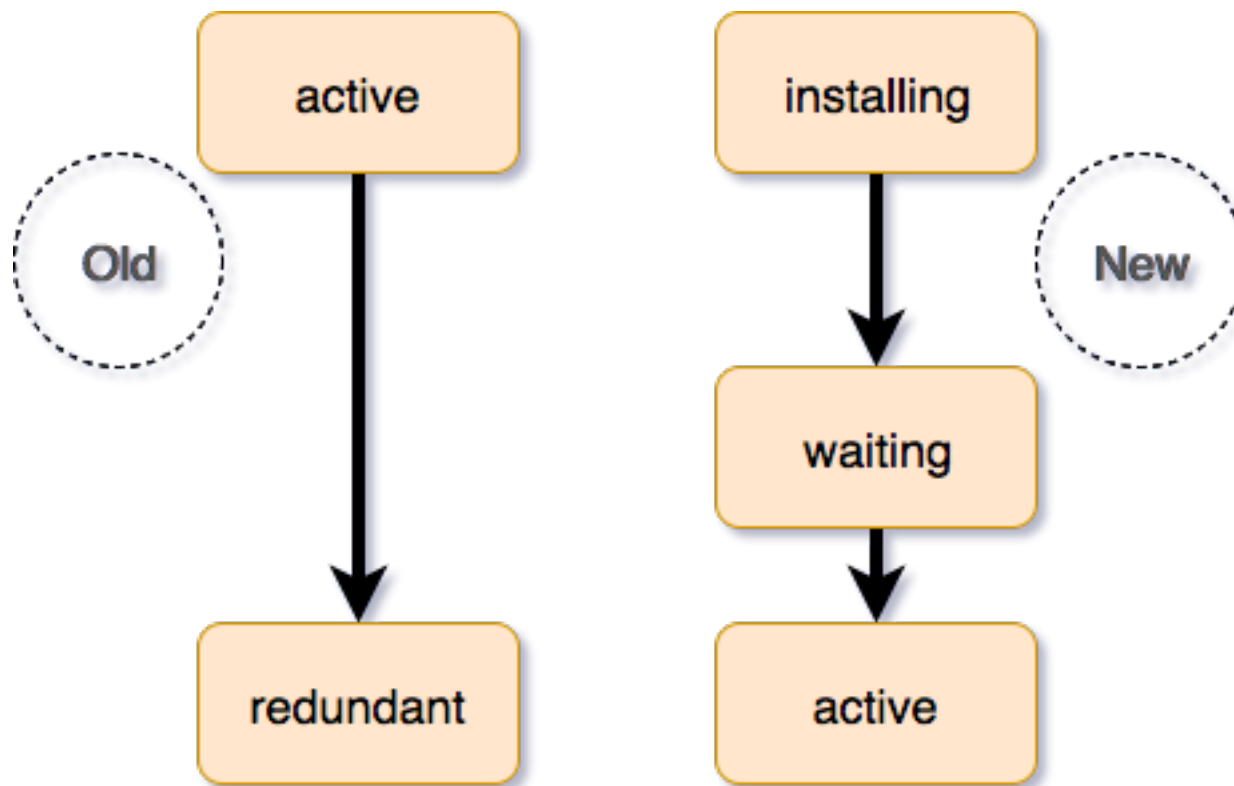
- 基于 Worker
- 域名坑点
 - 严格同域
 - 只支持 https
 - 调试域 localhost
 - 支持使用 fiddler or whistle 本地映射 sw.js
- 生命周期
 - 初始注册 *
 - 脚本更新
 - 终止生效





1.2 SW 难点

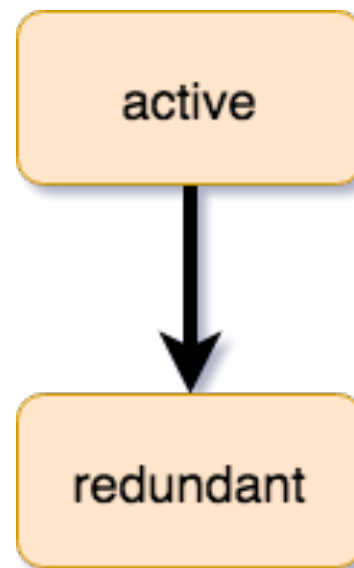
- 基于 Worker
- 域名坑点
 - 严格同域
 - 只支持 https
 - 调试域 localhost
 - 支持使用 fiddler or whistle 本地映射 sw.js
- 生命周期
 - 初始注册
 - 脚本更新 *
 - 终止生效





1.2 SW 难点

- 基于 Worker
- 域名坑点
 - 严格同域
 - 只支持 https
 - 调试域 localhost
 - 支持使用 fiddler or whistle 本地映射 sw.js
- 生命周期
 - 初始注册
 - 脚本更新
 - 终止生效 *





1.3 SW 更新

- SW 缓存设置: 缓存机制设置为 no-cache
- 自动调用更新
- 24h 后强制更新
- 使用 update() 手动更新

```
navigator.serviceWorker  
.register("./sw.js",  
          {updateViaCache:"none"})
```

date: Sat, 09 Dec 2017 13:21:17 GMT

etag: W/"5a0a9cbf-13df"

expires: Sat, 09 Dec 2017 13:21:16 GMT

last-modified: Tue, 14 Nov 2017 07:35:27 GMT

server: tws

status: 304

vary: Accept-Encoding

x-client-proto: https

x-client-proto-ver: HTTP/2.0



1.3 SW 更新

- SW 缓存设置
- 自动调用更新: 会在页面加载完毕后, 发送更新
- 24h 后强制更新
- 使用 update() 手动更新

Name

	80?ts=1511421615654
	80?ts=1510512012431
	360?tp=webp
	360?tp=webp
	360?tp=webp
	360?tp=webp
	360?tp=webp
	360?tp=webp
	360?tp=webp
	80?ts=1511443603290
	360?tp=webp
	360?tp=webp
	80?ts=1511505830793
	inject.js
	report_vm?monitors=[2226255,2...
	tdbank?table_id=personal_live_b...
	report.cgi?appid=20149&platfor...
	code.cgi?uin=&key=domain%2C...
	sw.js



1.3 SW 更新

- SW 缓存设置
- 自动调用更新
- 24h 后强制更新: 内部 last update check time 存储 date 头更新时间
- 使用 update() 手动更新

cache-control: no-cache
content-encoding: gzip
content-type: application/x-javascript
date: Sun, 03 Dec 2017 10:54:07 GMT
etag: W/"5a0a9cbf-13df"
expires: Sun, 03 Dec 2017 10:54:06 GMT
last-modified: Tue, 14 Nov 2017 07:35:27 GMT
server: tws
status: 304
vary: Accept-Encoding
x-client-proto: https
x-client-proto-ver: HTTP/2.0



1.3 SW 更新

- SW 缓存设置
- 自动调用更新
- 24h 后强制更新
- 使用 update() 手动更新: 发送 HTTP 请求
强制进行更新

```
navigator.serviceWorker.register('/sw.js')
  .then(reg => {
    reg.updateViaCache="none";
    reg.update();
  });
```



一. 小节回顾

- 网络代理
- 同域策略
- 状态变换
- 贼复杂的更新方式





2. 离线缓存

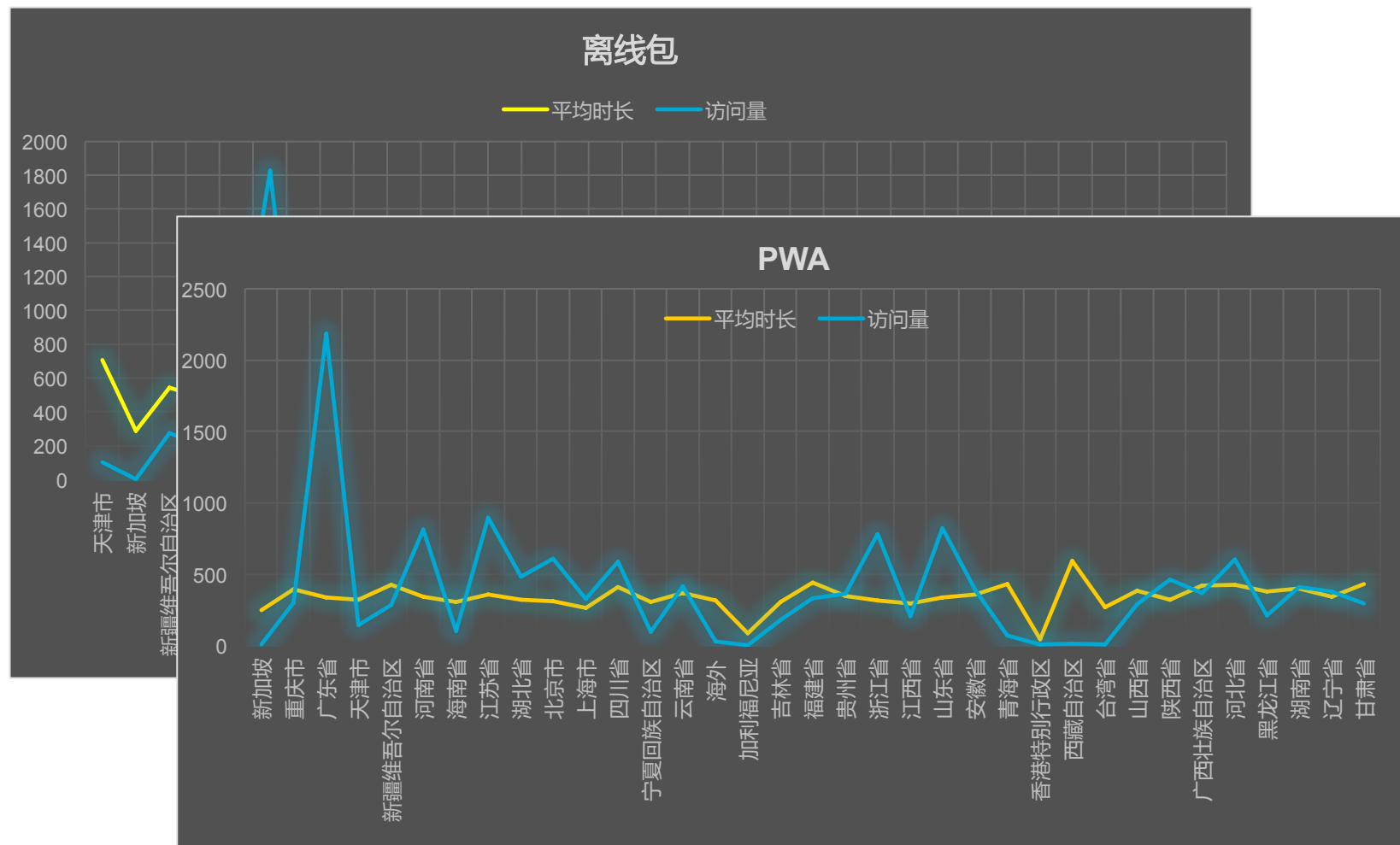
- SW 与 离线包
 - 延时数据对比
 - X5 系中的支持率
- 离线存储





2.1 SW 离线包对比

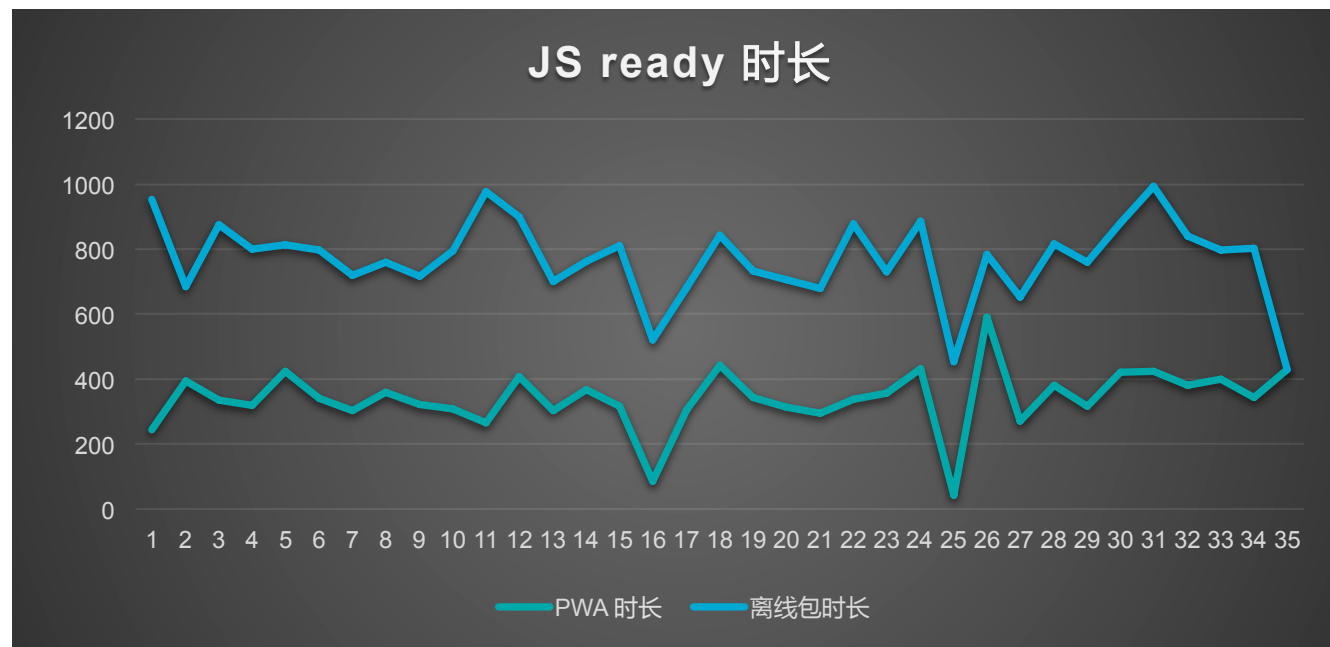
- Now 直播分享页
- JS Ready 时延对比
- 城市平均时延





2.1 SW 离线包对比

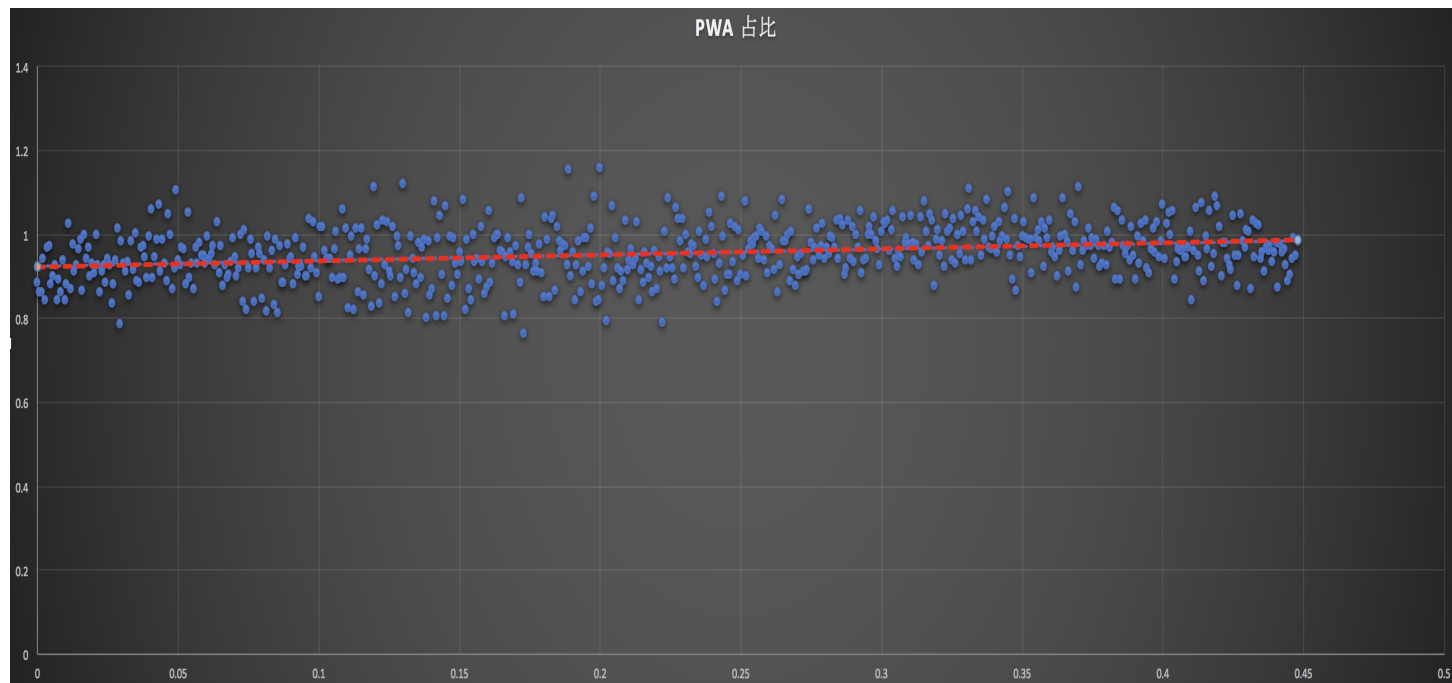
- Now 直播分享页
- JS Ready 时延对比
- 城市平均时延





2.2 X5 支持率

- 平均支持率 96.68%











2.3 离线存储

- 常用存储:
 - CacheStorage IndexedDB , LocalStorage , WebSQL。
- 安全缓存 CacheStorage
- 缓存溢出处理
- 缓存策略

Storage

- ▶  Local Storage
- ▶  Session Storage
-  IndexedDB
-  Web SQL
- ▶  Cookies

Cache

-  Cache Storage
-  Application Cache



2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
 - 同域储存：只针对读取
不针对存储
 - promise API, 完美结合 fetch
- 缓存溢出处理
- 缓存策略

```
interface Cache {  
  [NewObject] Promise<any> match(RequestInfo  
    request, optional CacheQueryOptions  
    options);  
  [NewObject] Promise<void> add(RequestInfo  
    request);  
  [NewObject] Promise<void>  
    addAll(sequence<RequestInfo> requests);  
  [NewObject] Promise<void> put(RequestInfo  
    request, Response);  
}
```




2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
 - 同域储存：只针对读取
不针对存储
 - promise API, 完美结合 fetch
- 缓存溢出处理
- 缓存策略

```
fetch(fetchRequest).then(function (response) {  
    if (!response || response.status !== 200  
        || response.type !== 'basic') {  
        return response;  
    }  
    var responseToCache = response.clone();  
    caches.open(CACHE_NAME).then(function  
    (cache) {  
        cache.put(event.request, responseToCache);  
    });  
    return response;  
});
```



2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
- 缓存溢出处理*
 - 共享空间
 - 大小检测
 - 移除策略
- 缓存策略






Browser	Limit
Chrome	<6% of free space
Firefox	<10% of free space
Safari	<50MB
IE10	<250MB
Edge	Dependent on volume size



2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
- 缓存溢出处理
 - 共享空间*
 - 大小检测
 - 移除策略
- 缓存策略

Storage

- ▶  Local Storage
- ▶  Session Storage
-  IndexedDB
-  Web SQL
- ▶  Cookies

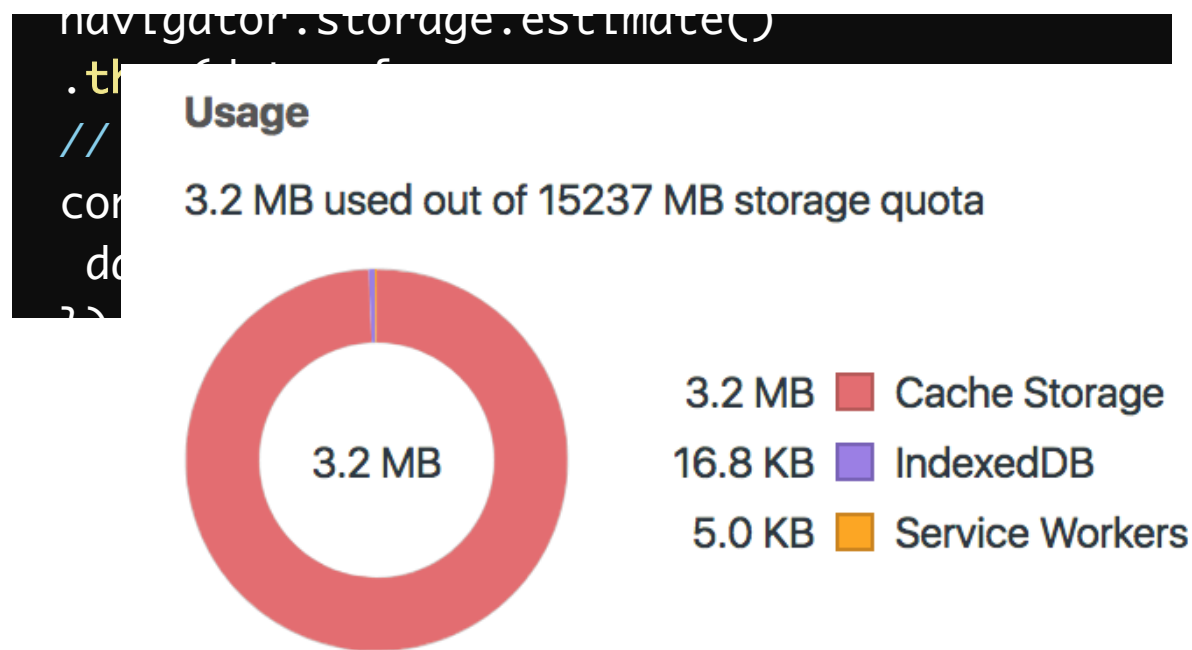
Cache

-  Cache Storage
-  Application Cache



2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
- 缓存溢出处理
 - 共享空间
 - 大小检测：[Quota Management API](#)
 - 移除策略
- 缓存策略

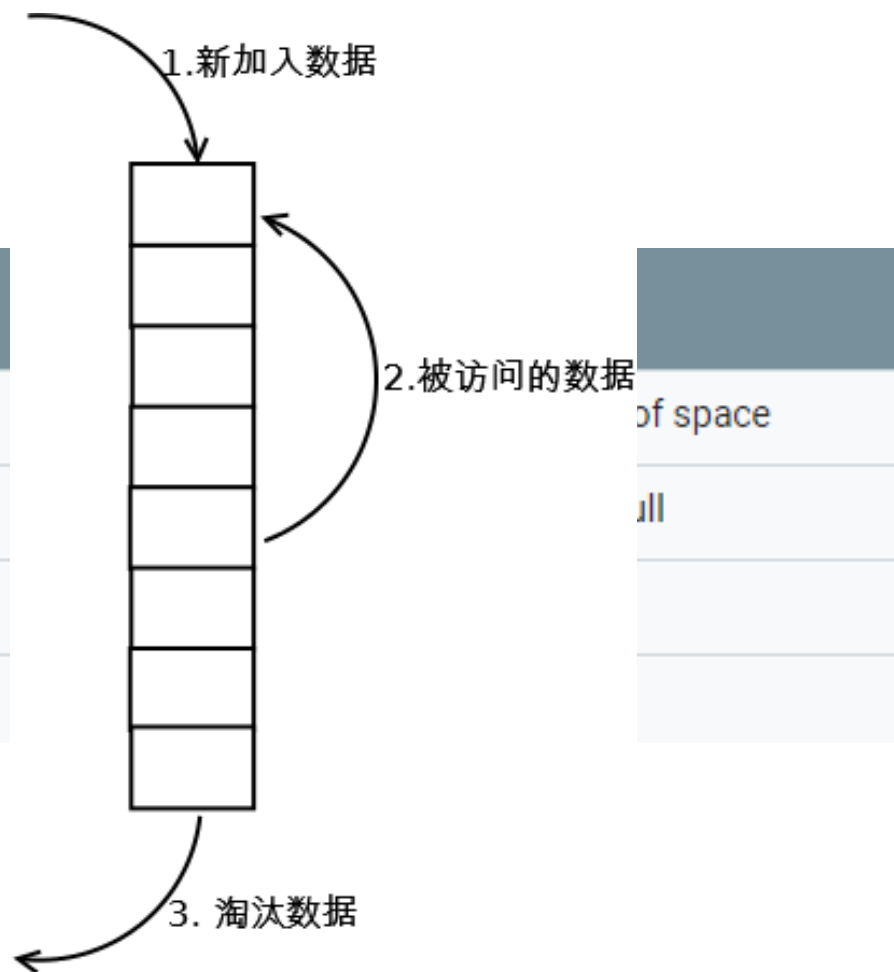




2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
- 缓存溢出处理
 - 共享空间
 - 大小检测
 - 移除策略：LRU（最少使用原则）|| No eviction
- 缓存策略

Browser
Chrome
Firefox
Safari
Edge

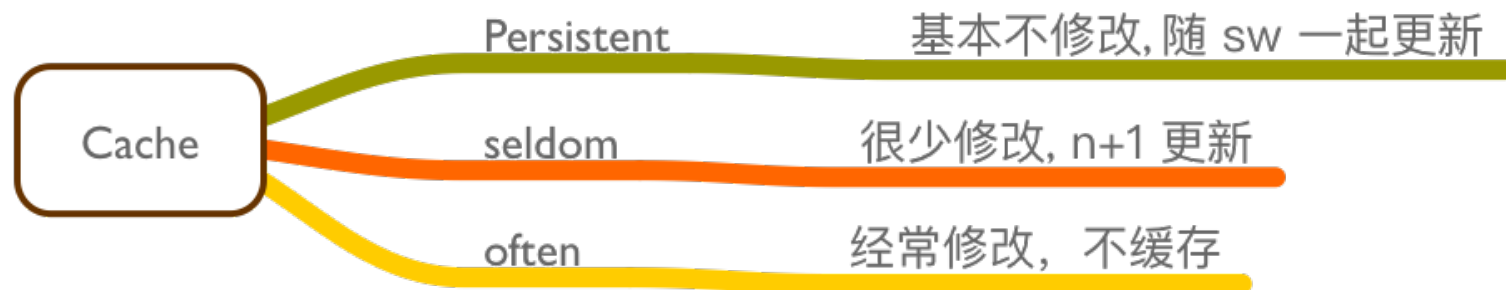




2.3 离线存储

- 常用存储:
- 安全缓存 CacheStorage
- 缓存溢出处理
- 缓存策略

- persistent: eg react-min.js, lib.js, template.css
- seldom: eg index.html, chunk.js, activity.css
- often: eg port_1di21f13.png, cgi





二. 小节回顾

- 性能数据加载对比
- 常用存储数据库
- 数据存满了怎么办？





3.1 模拟离线包

- 离线包

- 灰度
- 增量部署
- Push 下发

- SW 离线

- LocalStorage 存储灰度 List
- HTTP Last-Modified/E-tag 缓存头
- Push Message

发布离线包

修改离线包配置

设置过期

权限管理

仅用于灰度测试的包（不会改为全量发布的）
建议测试结束后都设置为“过期”

bid: 2336

<input type="checkbox"/>	版本号	支持版本	数据包	大小	是否灰度	是否过期	手Q版本	创建时间	备注	功能
<input type="checkbox"/>	54796	 	下载文件	2316KB	否	否		2017-11-24 17:36 by adamhe	pwa和离线包区分 上报	详情
<input type="checkbox"/>	54767	 	下载文件	2316KB	否	是		2017-11-23 21:27 by adamhe		详情
<input type="checkbox"/>	54229	 	下载文件	2316KB	否	是		2017-11-08 17:59 by willliang		详情
<input type="checkbox"/>	53865	 	下载文件	2316KB	否	是		2017-10-31 20:57 by jimmytian		详情
<input type="checkbox"/>	53822	 	下载文件	2316KB	否	是		2017-10-31 11:00 by jimmytian		详情



3.2 优劣对比

	离线包	ServiceWorker
更新方式	Push 下发，客户端拉取	N+1 更新，用户被动更新
更新大小	Diff 增量下发	根据 HTTP 缓存头更新文件
返回效率	已经很快了	还快 ~100ms
离线资源控制	客户端背锅	前端处理
总结对比	文件小，但覆盖很慢	更新速度快



三. 小节回顾

- 手Q 离线包特点
- ServiceWorker 模拟
- 没有离线包技术，试一试 SW





4 未来规划

- iv-pwa
 - 提供现成 ServiceWorker 缓存策略和 library。方便接入，灵活处理。
- iv-pwa-plugin
 - 提供 webpack-plugin 插件，适用于 webpack 构建项目接入 pwa。
- WebPush
 - 实现 Push 动态更新

THANKS