《好移动开发性能实践和搜索》

360 李永剑

李永剑：大家下午好，我叫李永剑，然后我的网名叫耗子，英文名叫 JERRY。2013 年来到 360 一开始导航，2014 年在移动搜索这边，然后这有我的邮箱大家可以把简历发给我。大家有没有看到这是我们的页面，我们最早是叫 360 搜索，后来我们改了一个比较牛的名字叫搜版本，最早的版本就是这样。然后慢慢我们的眼镜，可以看到将来越来越多的功能在上面。我们可以看到功能是越做越多，现在线上大家看到只有一个版本，其实我们在后面做了很多，有一些做了预测试，有一些没有命中。未来是什么样子我也不知道，我们的产品会各种折腾。

然后用户访问需要什么呢？有一些人会有一些想法吗？王老师。

王迎然：你的 PPT 我看过，快、不卡、省流量。

李永剑：大家肯定对所有的网站都很重要，我打开快我的操作一定很流畅这样才能爽，这可能是移动用户关心我的流量。我也不希望有很多时候要去思考，去猜，然后我这功能没有预期，当然更重要搜索快这是最根本，当然我讲的前端还是前面四点。讲优化之前，一服务器资源不足你给我说优化这些都是耍流氓。如果你只给我一个 1EO 服务器，1 兆的带宽去服务这么多的根本没有意义，我们的服务器资源不足的情况我们怎么来进一步的优化网站。

大家可以看一张图这张图是我们的（英文）下面的一个时间表，这在浏览器大家可以看到，它可以委托你的请求的数值，大家可以看到从一开始我们准备去打开这网址，然后可能会出现成像，可能 DMS，然后在 TCP，然后请求然后到回来，然后再浏览器解析，直到页面完成，这么长的过程。从这到这，我们都有可能设为一个白屏，也就是用户在正常的等待状态可能什么都看不到，如果你一个页面就是一个白，或者你写一个（英文）它永远都可能是白屏。

那我们就一步一步来讲，从这前面、后面然后到最后，我们一步一步的优化做了哪些工作。我们先说请求，大家知道 APP 请求，它更底层的像 TIP 对这些封装。我们优化 APP 的话，我们得从最底层优化开始。我们做了哪些事呢？一个是 CNAME，DNS 解析，我们给它转成了 APP。比如说我们绑定了好搜.COM 这个域名，这其实在浏览器它会先去找到这个别名，再把 APP 的域名对应上。我第一次拿别名就可以把 APP 拿到。这个技术 CNAMSWE 有。CFDNS 把一些需要的域名给请求下来，我们的 APP 就不需要通过 DNS 去解析域名，就可以

访问对应的 APP。DNS 五 PREFETCHING 这个大家可能知道标签有一个。它可以指令我下一个页面需要解析哪些域名，可以在这里面声明一下。还有一个 LINKPREFETCHING 是一个标签，可以指明我下一个页面浏览器提前加载一个下一个页面。但是这个标签基本上没有什么表明，除了安卓 4.4 里面有支持。然后我们是有我们自己的浏览器，我们 APP，我们其实在我们的 APP 里面做了这些事情。然后 KEEPALIVE，大家知道就是有一个请求，我希望跟后面的请求去共用，去设 KEEPALIVE 的时间，主动预热就是指我们会去猜测，用户可能下一个要点击的页面。然后等我们确定去定的时候已经加载完了直接展示出来。这个在新版的里面是有这功能的。

刚才说到少跳转，大家知道我们域名从好搜然后一开始搜.COM，然后 360.COM.CN 换了几次域名，其实现状一直在跑，然后特别是我们一些老的 APP，它的版本比较低，是一些旧，没有强制去跳转。第二个是服务器跳转要优于客户端，如果你真要去做一件事，想让它去跳转另一个域名的话，你尽量去在服务端去做，不用在 JS 或者标签去跳。

好，强缓存。我们的静态资源全部要上我们的 CDN，我们 CDN 设了非常长的头，我们的文件就是使用 MD5，作为它的文件名，让它保持不一样的。这个大家可能有一些使用问号，然后等于 1 或者让它等于 2。如果我猜测你版本的话，下一个版本 3，我预先把你这 JS 请求下来，然后你这区域所有用户的 DNS、CDN 拿到就一个旧的文件，这时候你执行 2 的版本文件，这也是一种攻击叫反存投毒。对动态接口，就是比如说大家看我们首页，我们首页有像新闻、天气，这数据其实在每一次请求的时候不一定有变化，它可能几个小时变化一次，对这种接口我们也可以优化，我们不会带来随意的时间戳，我们会有一定的间隔，这样让它在一定时间内数值是不会变的，在这一段时间可以用。

我们还做了更重的一些事情，比如说我们把这请求，第一次请求给它承载缩减，十分钟之内这缩减都会直接去读，然后不会去找网络。还有一些比如像城市切换损失联动，如果你全部存在本地特别的大，做了一些反存毒存了一年的时间。相信一年之内中国也不会凭空多了一个城市。

对我们的同步资源，会把它的第一次打到 THML，然后异步资源像改写 JSONP，还有 AJAS、REQUIRE，然后罗列的这一套逻辑进去。大家看一下这是我们线上开发环境中的 JS 脚本，我们会给它做一个标记，有 DATASCRIPT，这标记伪代码就是这样一个逻辑，我会去请求去找一下它的 Cookie，代码 Md5 作为一个请求，第一次请求肯定没有 Cookie，后走下面的逻辑，会把代码这里执行，执行完了以后会把 script 并且给它设置一个 code、cookie，直接从本地区读，不会再去请求一大串的代码。

好，那我们继续说少请求。第一个是域名收敛，我们之前在移动端使用了很多二级域名，比如说我们的图片、还有我们的压缩。大家知道在 PD 上面一个域名下面它同时只能并发两个资源，我们为了做一些优化，只能去闪电，会使用多高的域名。在移动上面其实我们大部分都是 HTTP2 内核，它的变化很高了，所以我们没必要使用那么多域名。服务端请求多合一。这个待会再说。我们在服务端做了一些代理的工作。

然后 HTTP2 和 SPDY，这个大家都应该了解了，我们很多网站都开始转向 Http2，像百度、阿里全部用 SPDY。可以复用 Tpp2 连接。比如说我设了很长的文字，每一次传输过程中，每一次 http2 是有投压缩，如果第二次传输相同的投体积会非常小。然后图片压缩，我们的图片上线之前，我们会本地使用一些工具去压缩图片，传到我们图片服务器，我们也会去做这一件事，它会把你的图片使用几种扭转的压缩方式，都压一遍看哪个体积最小，使用体积最小保存。

图片还有一个 LP 完全支持的，所以我们的网页然后在安卓下面基本上都会使用 MP，判断 MP 大家知道，我们可能会去一像素 1.gfi 构建一个请求是否成功，看到是否支持。但是我们判断了之后会把这标记给它积到 cookie 里面，这样服务端第二次，我服务端在代码的时候已经知道它能够支持 1Bgf。

我们对 1 倍做一些优化很多效果是可以接受，还有一点 1.GIF。GIF 有时候可能特别大，图片很小，不是所有地方去做一些动画。cookie 大小的控制，我刚才也说了在 APP1.×下面是不可能压缩，如果不是跟服务端交互使用。作用域最小原则。如果我是好搜.COM，然后 Domain 这下面明确设置 path 而，可以减少它的体积。然后 cookie 创始时间，比如说我们要做一个活动比如双十一只有一两天时间，我们服务端需要去判断一个 cookie 进入用户的标志位，我们只需要生命周期设一到两天就可以，让用户双十一永远带着 cookie 上来。

这看看我们做了这些测试之后，我们一些优化特别明显。这条绿色的线是中国移动，大家可以看到它的速度是比这几条是联动和电信。可以看到差别特别大。他们平均的请求就是页面的跳转时间在一秒之内就已经 OK。可以看到电动它二级下面会明显的比它们高。我们作为优化之后在这个时间内可以看到立马非常平直的下来了。

这个你可以看一下我们的首页，我们有非常多的区块，每一个区块是不同的新闻源和不同的部门给我们提供的。比如说像段子像豆瓣的数据。如果我们去前端去请求，比如说我们要请求十个卡片，把十个卡片大家会产生十个并发请求，我们做一些什么优化，我们在后端去做这件事。我们把多 API，我们同步请求下来，之后下来一致性发给页面。因为你用 PHP 去请求这数据，这数据不同板块，如果你有一个数据源特别的慢，可能导致你整个页面加载。

所以我们后端的操作，给它的操作时间设的就是一秒，如果超过一秒的话，我的后端不会去请求这些数据，会直接就已经放弃了。我们前端的代码它就会去自己再去请求这卡片的数据源，再给它显示出来。

这个指的就是我们前端和后端的代码做了一些事情，然后它会第一次从页面的，比如说后端可以从一个 Wionds 页面，所有的先尝试从这里面取，如果取不到的话，会把数据形态下来。

第二次这些数据都会强调，第二次请求也会特别的快。然后域名搜点，我们现在使用的三个域名，我们的 JS 我们的图片还有我们的主站。图片和 JS 我们使用了两个域名，主要就是防止他们去竞争，比如说我们的研究数非常少的情况下，JS 应该是比图片更重要。

这个是大家的一个理念，如果我们数据能从本地的一些制作方取，肯定是内联到你的页面标签里好。如果你要是内联不行的话合并请求，就是把多个请求合并成一个并发，并发也比串行的，挨个的请求要好。然后就是 304，我需要跟服务器做一次交互，然后不反复剥离，到时候你的稳定数据还可以用，比 304 差一点就是 200，比 200 更差一点就是 301。302 其实比 301 还差，因为 301 还可能被浏览器缓存，302 每一次都要经过服务器。

再要说就是渲染的优化，我们大家知道浏览器，大家抱怨最多的就是一个卡，第二是在手机可能很多是白屏。卡的原因主要就是你的整个的 JS 卡顿，白屏主要是你的内存不够了。内存不够了就是你的手机上会释放内存，闪一下屏、白屏。对卡顿白屏做了一些优化。

第一是首屏优化，第一直接从后端渲染，直接后端吐出来比前端要快。Head 有一些而很少量的。然后 Head 它基本上存储的量只跟首屏有关，至于后面的屏我们放在模块里面，用 js 再打到页面上去。然后就是 icofont 内莲花到 css 里面，而不是再发一个请求。大家知道 icofont 其实浏览器做了一些事情，如果网速很慢的话，它就不展示一直要等到你的字体文件请求结束了，再会把这图表给显示出来，这样给用户感觉到会抖一下、闪一下。

然后 css 内联图片尽量减少，如果不是图片特别大的话，尽量给它合并成倍增式。对一些多个地方要重复使用图片的话，让它使用图床，让它地址尽量保持唯一。对于图片的佳奖我们会尽可能放在事件之后，为什么要放在事件之后，我们首先浏览器会有一个建筑条，这建筑条其实他们开发的时候，就是要简称 img，这对用户的感知来说快很多。

首屏重要的 js，比如说某个链接必须要尽量用户去交互，我们直接裸写，然后内联进来，不需要等到库加载完再去做这操作。还有就是链进去同步做这 ajax，而且你在 ajax 请求过程页面没法交互。这个错误大家基本上不会犯的。

好，刚才说到我们卡顿有一个方式就是提高它的针对。对于相对位置不变的元素让它尽

量的再一个 renderlayer，不要放在不同的层次里面，对于一些动画的元素，我们给它一个独立的 renderlayer。如果有我动画慎用圆角、阴影、滤镜因素少用这对你的内存影响比较大的。

这个是一些会触发 renderlayer 的地方。如果你的元素使用 3d，被它创建独立的层去渲染。如果你使用了这些 wil—change 等等，有透明度有动画，可能这 canvas、flash，这本身都很好理解，它不是一个静态它每秒都在变，所以浏览器渲染的时候都再一个新的 renderlayer。gif 并不会创造新的 renderlayer。如果你还让它一直在那转会消耗你的资源。还有像 cssfiters 的绘制成一个新的。还有 z—index 额大于某个相邻节点的 layer 的元素，如果两个元点浮动的话，在你们之间不会出现重叠的话，不且建议给 renderlayer。

另一方面释放内存，控制深度还有一个数量，尽量少使用 DOM 节点。然后图片比较占内存的资源。然后滚动到屏幕外的图片给它删除掉，屏幕到屏幕外的元素删除，事件在里面使用，而不是网站拒绝。对于渲染之外去交互，如果你在网页中很多交互，用户的一些心理，你要让他尽量符合他的预期。比如我说点一个元素这时候的预期我要点它的空白位置，把它收起来，否则我弹出一个层，在他点确定或者点 X 的时候有一些符合预期的动作。比如说点 ioading，这个请求特别快后端的请求，这个时候给他精美的 ioading 的动画，所有的窗口给他一个及时的反对。然后一叶比较大范围的移动元素的话，会给它一个平滑的过度，如果你给它顺移过去的话，就会觉得丢针，给它一个平滑度更好一点。还有局部刷新，大家想到最多的是代页面，其实对用户来说。其实我们在移动页面，我们输入方就是一个 lid 的，这样用户感觉有一个局部刷新的感觉。

好这里面可以看一个交互，这是一个按纽，它在点击的时候，马上给它一些愉悦的反馈。比如说我来点击一个下拉层，它应该是下拉点击空白的时候会暂停掉，掉结束切换的时候会有一个 loding 这效果会看到。对于知数，比如说 suggest，我们为了让他感觉不卡我们减少键盘事件监听间隔。其实大家去做这事情我解决一些请求，比如说五百豪秒里面这其实让反馈会感觉到慢。我们为了这就会去把这间隔设的很短，其实这会产生更多的请求。对于这些请求我们做了另一个优化，就是把我们的 swggest 全部用优化写。然后预热用户可能点击的结果，这刚才也提到了，就是桥页增加目标站的 title，我们很多的点击是要通过页面去中转。如果第三方的网站很烂，可能让用户感觉半天都没有任何的信息。这时候我们做了一件事情，就是我们跳转的时候，我们是把第三方的 title 显示出来，这让感觉第三方很慢。

然后还有 a 链接点击的时候，原窗口的话，我们可以直接直接把 title 而窗口改掉，这样让用户感觉到了第三方。然后还有 throttle，还有 debounce。throttle 相当于一个截流，比如我做一个游戏给它十分钟才能放，如果用户十分钟长按这没有任何，如果第十分钟一到马上

执行。然后 debounce，比如说设十分钟的时间，在十分钟之内不会执行，到最后一次按下的时候，十分钟之后会执行，它们之间有一些细微的差别。如果我们平时你要执行一件事情，给它五百毫秒之后执行，这其实我们做的相当于 debounce 操作。如果你更好的操作，throttle 去截流的玖，你的网站给用户反馈会更好一点。 刚才说到我们的优化。

然后我们在工作中，就是我比较遵守的原则。我们为了节省这些，比如说之前判断各种方法比如说判断它的是安卓版本，是否是 ios，可能在 js 里面有一套 php 里面有一套，我们全部封装在一起加一个 env，它会向三个地方，后端还有 ens，php 是 env，下面的方法都是一样，直接判断可以直接使用。对整个的网站我们的通信，我们封装这几个定了发布机制。它和我们常用的事件绑定还是有区别的。比如说我可以先发布、可订阅或者先订阅后发布都可以执行，这主要是跟我们的客户端通讯使用。或者我们注入一些事件和行为的话，它的实际是跟网页是不知道先后顺序，这页面加载一般地时候，它把 js 注入进去。这是我们封装的时间绑定，这样会造成事件代码跟 lom 结构是解藕的。他们都是使用 checklist.js，我们以前写代码也是这样写的，你在页面上看到 debug，它绑定的事件另外一个去扩张，把事件的绑定和 dom 进行了一些解耦。

然后 checklist.js，我们会针对不同的容器做一些不一样的展示，你针对一个新功能的时候没有问题，然后是上线，过了几天你的 EP 开发人员说页面挂了。这些东西很难去撤，我们就给所有的这些页面上，我们就有一个列表它会检测你一些事情，然后防止你去犯一些错误。这个代码在上线的时候就会被干掉。我们所有的开发骑士队 debug 非常重视，我们开发一个功能可能的四分之一代码都是跟调试有关。那我待会可以说一下，我们是怎么上线把这些代码去掉。

我们先上王迎然老实说的开发环境，我们之前没有本地开发环境，我们全是远程阶段的开发。但是这样对我们电脑没有很好的利用。像大家知道 mp 的 ro 非常快，但是你的机器比较烂的还有网络开销，然后本地区搭建环境，这个我们了解到的。

然后我们 js 纯使用就是写代码，我们用了很多新的特性，以前我们重复文字和寄存都不需要再使用，我们模块使用 Es6 的方式去打包。可以看到我们调试代码就这么干掉的。定义了一个比较特别的一点字符串，这字符串，如果我在开发环境中，它肯定返回来一个这里面就会执行，我们上线就会把字符串条换成 console，下面就不会执行，这在一压的话整个板块都会压掉。这让我们在代码任何地方比较方便去写调试的信息，然后也不会担心这些代码用到线上。像我们的模块，我们使用这样 es6 的原生方式，我们写了一个功能让它变成 cmd 模块，我们又给它编译成了 js。整个过程都是一起去做，也减少了一些重复的工作量。

小，我们之前也说到一个 mock 平台。我们之前开发会有 mock 的模拟和接口。我在开发的时候，就是把请求打到这个接口上，这个接口根据你传播的窗口反馈不一样的数据，然后也可以模拟四百、五百。然后超时，然后直接让后台去点一下，然后就产生这样的效果，测试代码。你要上线的时候，你再把数据的例子改成真实的例子。我们使用另一个域名。我们还做了友上的词条质量的对比工具，我们做了友商的对比结果一些大小体积。就是会在命运行去跑的浏览器，和这样的体积大小。这个过程中也让我们发现了我们一些页面 gif，还有第三方的图片特别小，这是肉眼没法看到的，也是通过分析工具跑出来的，我们对活动页，我们有快速的环境构建工具，还有一个性能分析统计平台，还有做了模拟用户的地理位置还有网络请求，模拟用户的机房。

我们分享一下统计数据，这个图是我 9 月份抓的一个图，右边×360 的分辨数据和 728 的数据是比较大，除了用户群体，对安卓一部分是很屌丝占比比较高，另一部分高端机占比比较高，我们针对做了一些特定的优化。

这张图还是移动，可能这条线的纵坐标是不一样的。我们可以看到移动的速度也在不断的加快，我们可以看到 4G 的用户普及越来越多，未来我们的一些优化也是做一些相应的调整。页面耗时，这是我们统计出来的，我们基本上页面在一秒钟之内，应该有 83.18%的用户在一秒之内打开。然后 2 秒之内有 94%，3 秒钟之后还有 3%的用户可能会打不开，这部分的用户也是我们现在比较关注和优化的一个方面。这是一些链接大家可以看一下。

第一个是协议方面的，第二个和三个关于性能优化方面的。这是雅虎的军规是 24 条还是 36 条，但是现在在移动时代很多东西已经在发生着变化，如果你刻意的完全遵守不一定可以。

好，谢谢大家。

主持人：好，谢谢李老师精彩分享，有同学提问的。

提问：我的问题是两个，第一个就是刚才讲到优化，通过优化系统那一块，我想知道缓存的更新机制是什么？

李永剑：我们缓存就是在我们发布的时候，我们会有一个发布流程。这时候会去增进页面的资源。然后给他们文件算出 MD5 的版本，一个页面上有关的总的 MD5，然后标志。如果这 MD5 变的话就过期了。

提问：其实我的问题是两个方法。我用这两个方式来进行缓存，就想这一块基于什么？

李永剑：就是把你的代码传到你的本地。然后再用 in 的方式给它执行。

提问：好。

李永剑：对宽裕的我们还使用了 Gmp 的方式，把整个代码作为一个字符串，你这样请求一个 gmp，再把这个数据拿过来给执行一下。

提问：这方式应该是建议的是吧？

李永剑：对，这个所有书上都会这么说，但其实我们是实用主义，大家还有很多问题，但是这问题是什么，你能够质疑很清楚就能够知道，你就可以去针对它。比如说你觉得性能差，但是我们这代码只执行一侧，你说不能加地区优化，但是我们这代码只需要执行一次，它毕竟你去请求网络开销，去请求资源去执行，有一些能源好的话我们可以用。

提问：好谢谢。

主持人：好的还有哪位同学？

提问：你好我刚才听说就是你会把所有的文件生成一个 MP5 吗？还是针对一个文件一个文件单独 MP5。

李永剑：我们页面所有文件生成一个。

提问：是在后端做的？

李永剑：是在编译的。

提问：比如说你是跟进这个去做，还是说我只改了一个文件，这样子你把全部的给改，所以现在缓存的文件都得统一跟进？

李永剑：我们的跟进度是页面。

提问：跟进度是页面，一个页面静态文件都会跟进。但是我只改了其中的一个文件，你就要把所有的文件改掉吗？

李永剑：这个其实有很多说法，我们的 SDA 友商针对到每一个文件级别。其实这东西你要权衡的。因为本地的文件其实有一定的风险。比如说我一个漏洞可能是一个反射性的，但是里面有一个代码操作这样我可能在访问别的页面上，这个代码也会执行，所以我们主要的目的是介减少它的请求数，而不是减少它的流量，它更新的频率会比较高，基本上每一次改了之后就要存量更新。

提问：我这边也在做，但是力度非常小，但是对每一个模块去做 MD5，一个模块改一下？

李永剑：但是你怎么去协商，如果你要第一次去跟服务器协商缓存的。

提问：我们纯前端的，所以第四块把数据拿过来，直接存到，第二次的话就直接从（英文）存？

李永剑：那你怎么更新，那你的版本得放。

提问：我每一个文件都放，我现在定义的时候会对每一个文件生成？

李永剑：把这配置比如说页面。

提问：类似于百度首页的做法？但是他们百度首页做了一个更好的，他们会把所有的文件放到一个 MP5 里面，我现在也是在做这方面的优化。我还有一个问题，就是刚才你说到

发展模式，我没明白你说的通过 NA 注入一个程序，我没理解你说的不需要关心 js 注入的实际，因为我在做项目的时候，我是遇到这个问题的，就是我必须页面中，把这观察者设立好，然后在 js 里面之后必须注入发布出去。如果你是在你 js 还没有网端的时候，就可能不可能出去，就可能没办法执行？

李永剑：对，这就是要解决的问题。

提问：那你怎么做？

李永剑：这里顶一个标记，如果它已经 dow 就不会执行。

提问：就是那边会有一个轮寻吗？

李永剑：没有，只用 js。