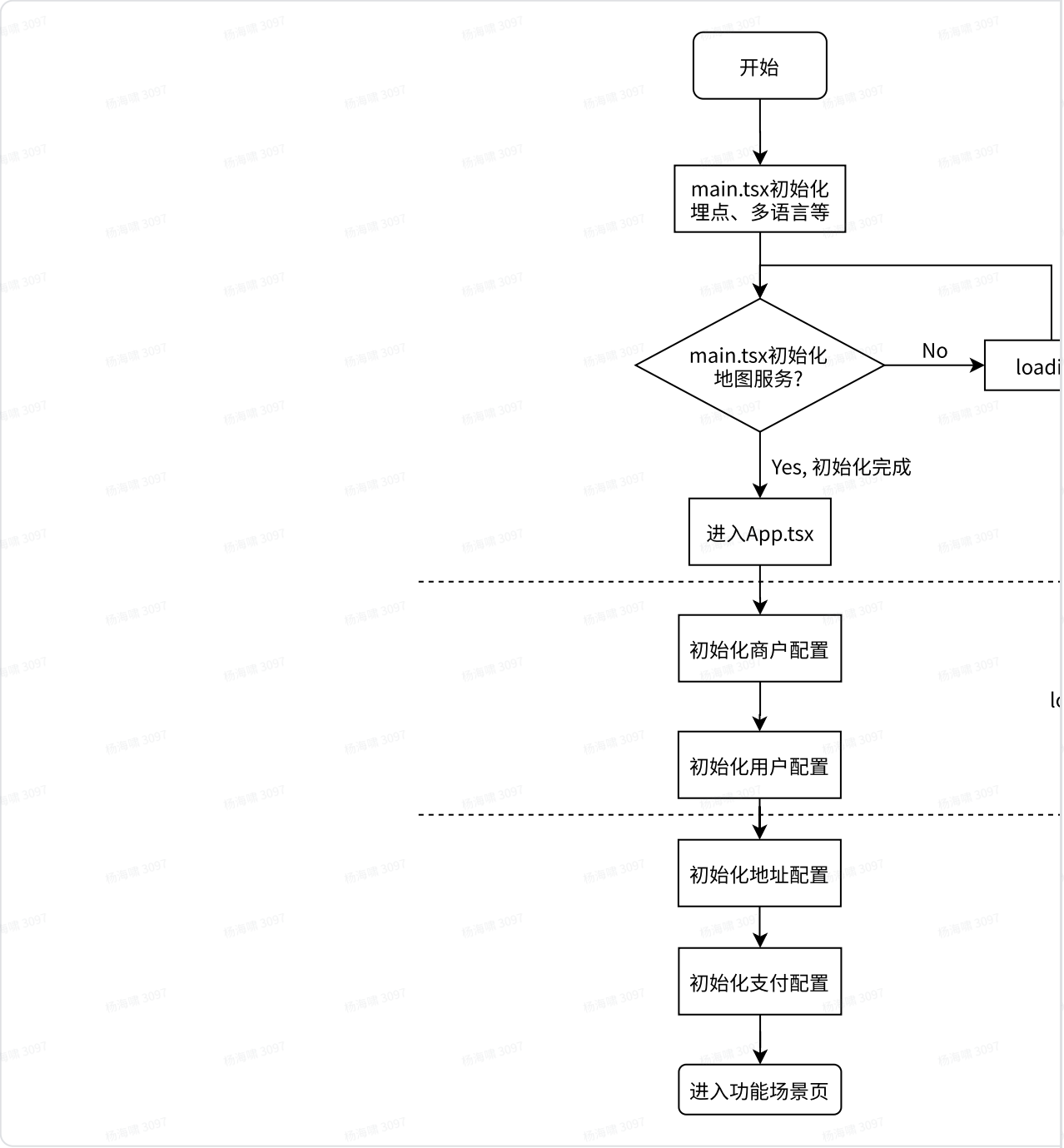


国际化独立站首屏性能分析及优化技术方案

背景

产品反馈独立站，在真实首次访问时十分缓慢，希望能够进行优化，提高用户体验。现对造的35条黄金守则_51CTO博客_雅虎性能优化规则及Google Webvitals原则做出一些分析。

独立站初始化流程




因素


设备第一次访问时首次访问需在 SSL 建连上耗时

以国内挂梯子，小弟的真实网络来说，如果梯子不好，耗时能高达0.5s-3s（耗时极不稳定）

Safari

以国内挂梯子来说，如果梯子不好，耗时…

 杨帅 2月21日 14:55
以海外新加坡节点开全局的模式来测，国内访问慢不是问题

 杨海啸 2月21日 14:57（编辑过）
我就是翻墙的新加坡结点。

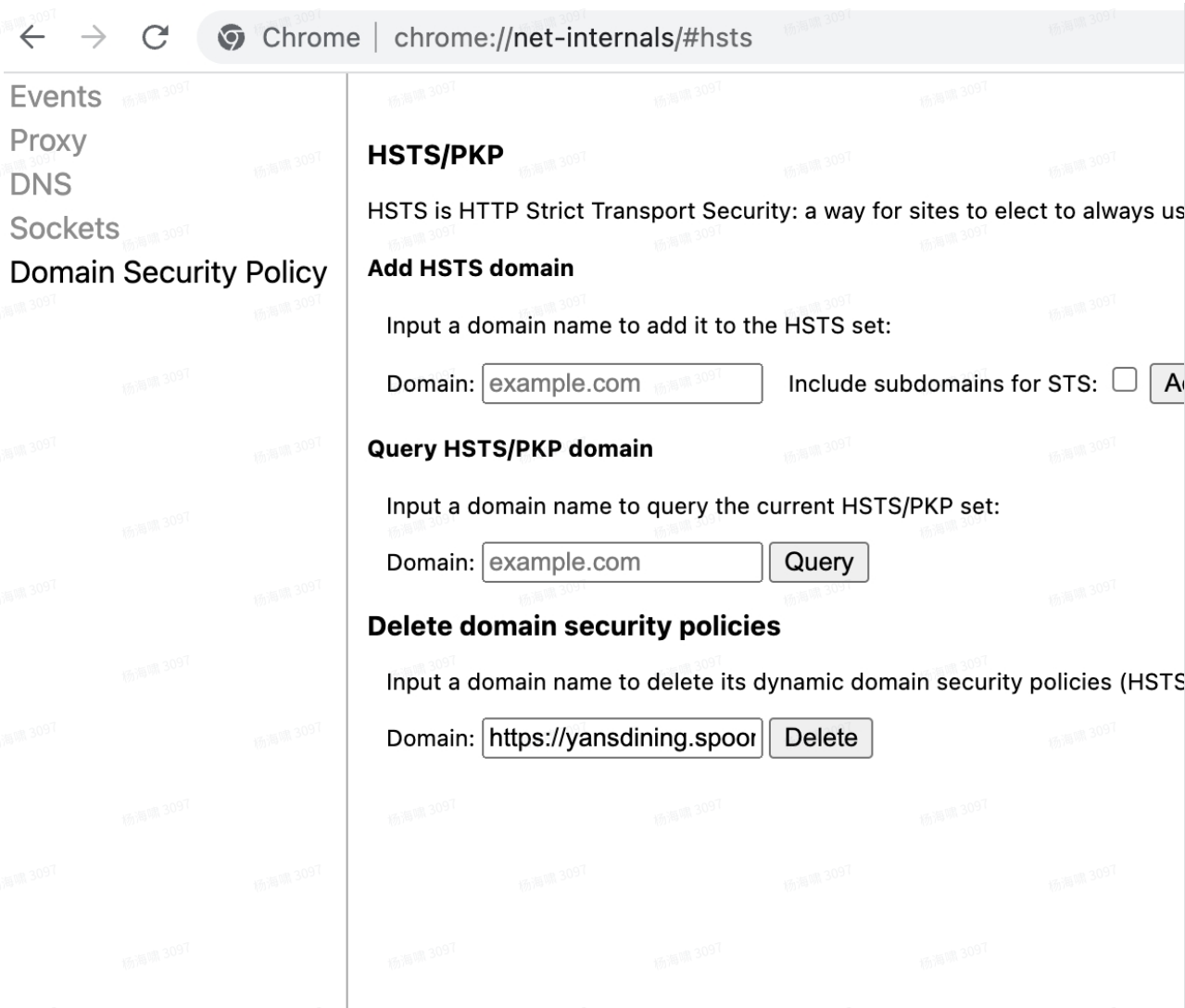
gen_204	maps.googleapis....	xhr	402 B	1
yansdining	shiva.spoonx.com	fetch	681 B	9
nearby	agni.spoonx.com	fetch	2.06 KB	
index.20533990.js	r.kezaihui.com	fetch	11.25 KB	4
formatPrice.cde3d511.js	r.kezaihui.com	fetch	92 B	
index.0da8369d.js	r.kezaihui.com	fetch	1.69 KB	
generateCDNImageUrl.c276f6bc.js	r.kezaihui.com	fetch	1.0 KB	
mobxreact.esm.b83c3503.js	r.kezaihui.com	fetch	4.29 KB	
index.cf3e505e.js	r.kezaihui.com	fetch	2.68 KB	
index.bc048a24.js	r.kezaihui.com	fetch	4.53 KB	
index.f922cd43.js	r.kezaihui.com	fetch	1.43 KB	
validator.09c89fc3.js	r.kezaihui.com	fetch	75 B	
debounce.61c4a282.js	r.kezaihui.com	fetch	1.22 KB	
index.ba4acfc6.js	r.kezaihui.com	fetch	5.82 KB	
index.42454d95.js	r.kezaihui.com	fetch	1.40 KB	
formatPhoneNumber.c671dca7.js	r.kezaihui.com	fetch	75 B	
index.560d3aff.js	r.kezaihui.com	fetch	6 B	
index.7e2661f1.js	r.kezaihui.com	fetch	26.73 KB	
dev-log.086e4fe0.js	r.kezaihui.com	fetch	8 B	
index.6a26c034.js	r.kezaihui.com	fetch	4.36 KB	
range.76ba9f5d.js	r.kezaihui.com	fetch	1.03 KB	5

Chrome

Name	Status	Type	Initiator	Size	Time	Waterfall
?sentry_key=e0cab789c44...	(canceled)	fetch		0 B	2 ms	
?sentry_key=e0cab789c44...	200	fetch		228 B	55	
gen_204?csp_test=true	(canceled)	xhr		0 B	686	Queued at 6.
yansdining/	(canceled)	fetch		0 B	678	Started at 6.5
gen_204?csp_test=true	200	xhr	index.41423c4...	90 B	172	Resource Sc
yansdining/	200	fetch	index.41423c4...	730 B	639	Queueing
nearby/	200	fetch	index.41423c4...	12.7 kB	2.1	
?sentry_key=e0cab789c44...	200	fetch	index.41423c4...	228 B	55	Connection S
6	200	xhr	out-4.5.42.js:1	553 B	1.7	Stalled
?sentry_key=e0cab789c44...	200	fetch	index.41423c4...	228 B	14	Initial conn
4601f7812e42460a89f5221...	200	fetch	index.41423c4...	584 B	650	SSL
6	200	xhr	out-4.5.42.js:1	552 B	537	
?sort=1	200	fetch	index.41423c4...	484 B	643	Request/Res
menus/	200	fetch	index.41423c4...	9.6 kB	625	Request se
addresses/	200	fetch	index.41423c4...	472 B	625	Waiting for
?coupon_scene=pick_up&li...	200	fetch	index.41423c4...	474 B	640	response
?limit=10&offset=0&status=...	200	fetch	index.41423c4...	495 B	639	Content De
?limit=10&offset=0	200	fetch	index.41423c4...	489 B	685	
addresses/	200	fetch	index.41423c4...	472 B	688	Explanation

如何测试 SSL 建连耗时

1. chrome 访问 chrome://net-internals/#hsts
2. 清空当前域名 CA 证书



3. 新建空白标签页，右键单击检查，切换至 network 清空历史请求后，访问对应商户域名

总结

最好新加坡同学测试下，此因素国内测试极大程度受到梯子影响。如果在新加坡 SSL 建连平下

以下因素可从前端角度解决

不考虑 Service SSL建联占用时间

拉取谷歌地图服务耗时

主要是 googlemaps 地图服务。这个会导致整个 app 初始化时被一层 wrapper 包裹，获取量。也就是说在 googlemaps wrapper 没加载好时，会一直 loading，体现就是项目加载变

```
ReactDOM.createRoot(document.getElementById('root') as HTMLElement).render(
  <React.StrictMode>
    <Sentry.ErrorBoundary>
      <Wrapper
        apiKey={config.googleMapsApiKey}
        libraries={['places']}
        render={status => {
          if (status === Status.LOADING) {
            return <HLoading loading />
          }
          return <App />
        }}
      />
    </Sentry.ErrorBoundary>
  </React.StrictMode>,
)
```

此因素及受当前设备访问 google 的速度影响

解决方案

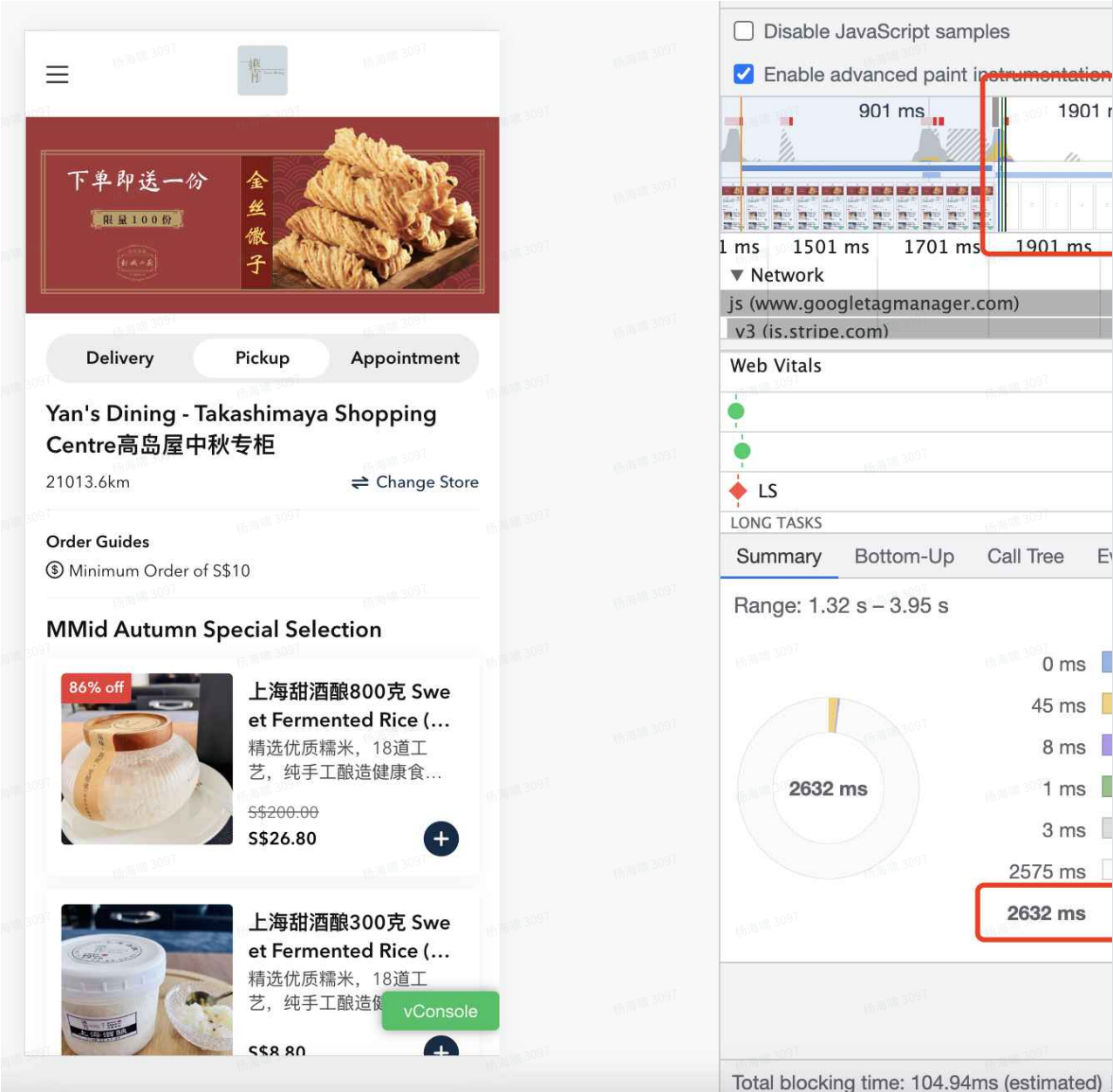
谷歌地图服务是否可以包裹整个app，仅对需要地理位置的场景包裹 wrapper 去加载。可避免白屏时间变慢。后续如果全局有 google.maps 就不用重新挂在 wrapper 了。得实际验证下

优化

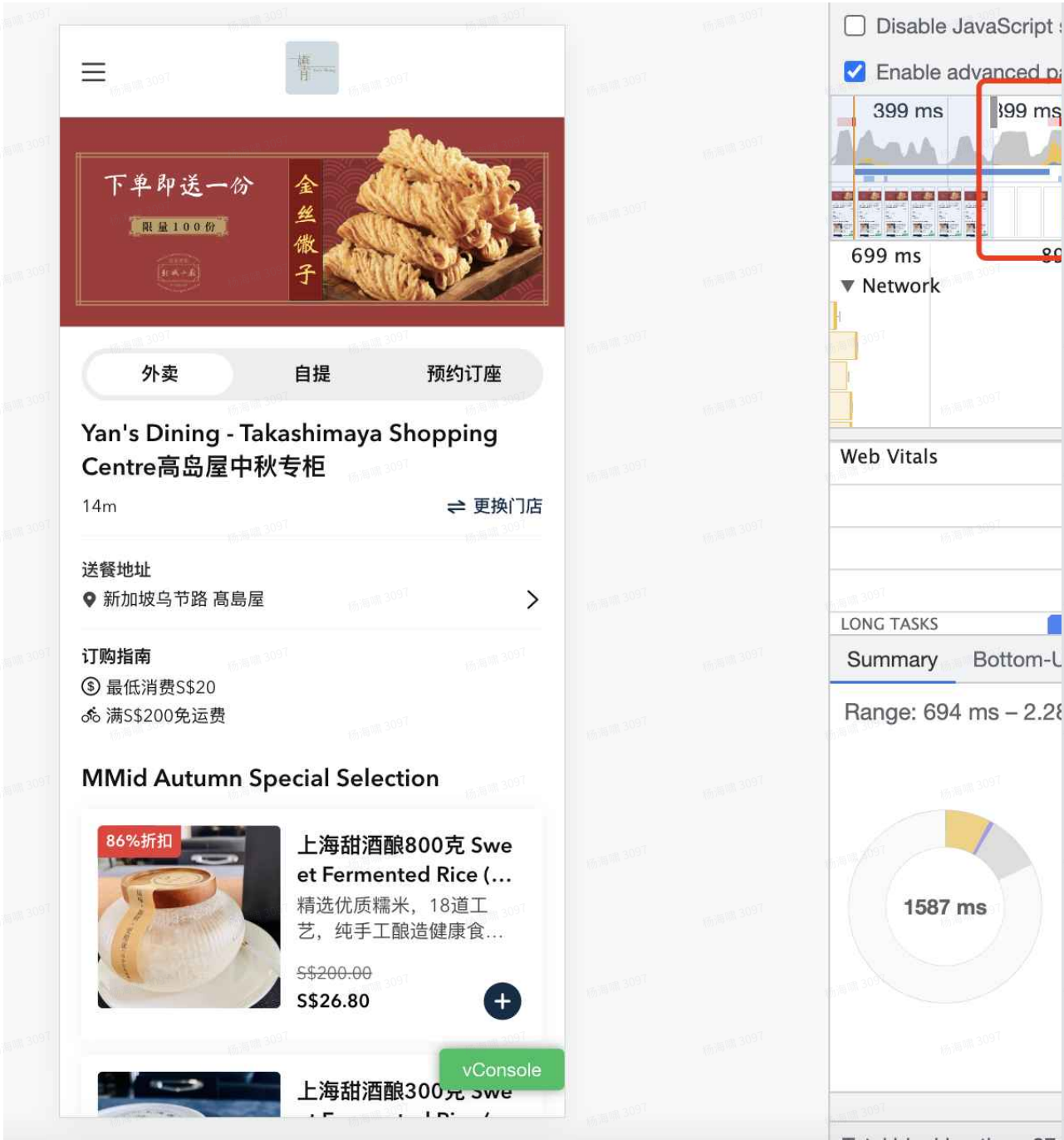
- 剥离 main.tsx 中初始化谷歌地图服务的 wrapper，避免增加页面 loading 态，降低 idle 时间
- 对于需要地理服务的场景。通过封装一个 withGoogleMapsWrapper 的容器，判断 window 是否有 google.maps 确定是否渲染带有 loading 态的 wrapper。（是否开启loading 态设为可配置的）

优化后

优化前，idle（白屏时间）平均在2500ms-3000ms左右。优化后，白屏时间下降到了1400ms左右



After



App.ts 内初始化 App 过程、接口梳理

历史原因，目前 App 初始化时涉及了商户、门店、地址、用户、支付等配置的初始化。基本丢。

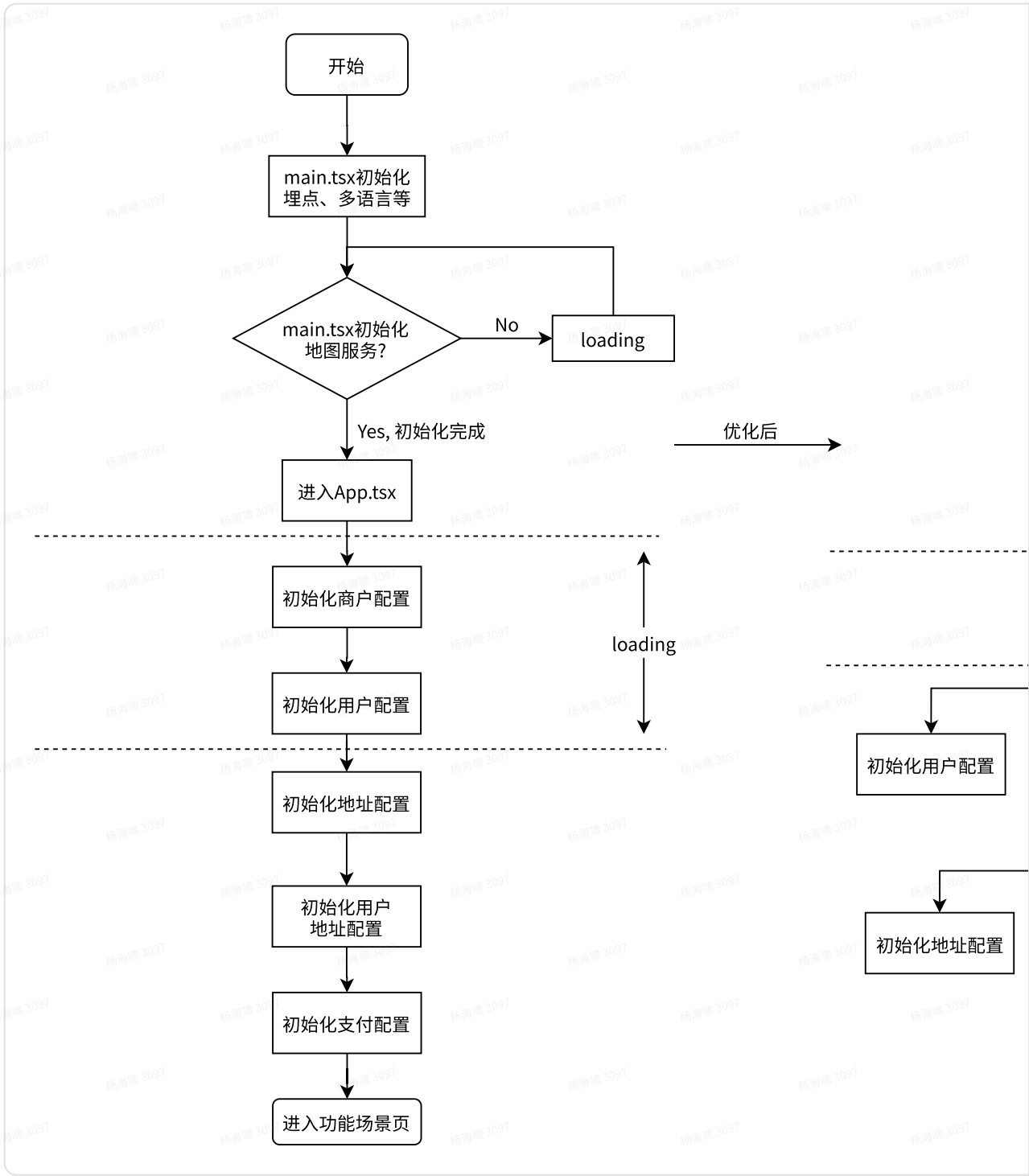
解决方案

- App.ts 加载优化(接口调整)

但是对于 App.ts 内初始化。不应阻塞页面loading态的初始化，无需串行等待，**可以处理成**户、门店配置是必要loading等待外，地址、用户、支付等配置非必要loading。后续这些配

- 减少 HTTP请求次数

区分场景加载。各种模式初始化区分开，比如预订模式不要再立刻初始化菜单、支付、地址



优化

- 剥离Stripe初始化相关逻辑,封装为useStripe
- 剥离App.tsx下单页初始化逻辑,放到下单页处理,避免阻塞加载
- 梳理App.tsx初始化流程,复用部分函数,优化逻辑
- App.tsx加载态仅受商户&门店初始化影响
- 地址&用户信息初始化同步进行
- isLoggedIn 改为computed属性，用于判断登录态
- 菜单页Cart只有totalPrice有价值才渲染dom节点
- 修复侧边栏&券列表副作用请求缺乏依赖重复请求
- 减少 HTTP请求次数

历史原因，品类导航和菜单联动导致菜单…



杨帅 2月21日 14:47

这部分实际请求接口和渲染要花多久



杨海啸 2月22日 11:15（编辑过）
就小弟的网络来说，艳青和鹏程小厨俩的菜单接口。强刷后请求，艳青最快能在 200 多 ms，不会超过 1s，鹏程小厨最快在 300 多 ms，慢的时候有 1s 多



杨海啸 2月21日 14:58

图片懒加载，那个就完全和网络有关系了。

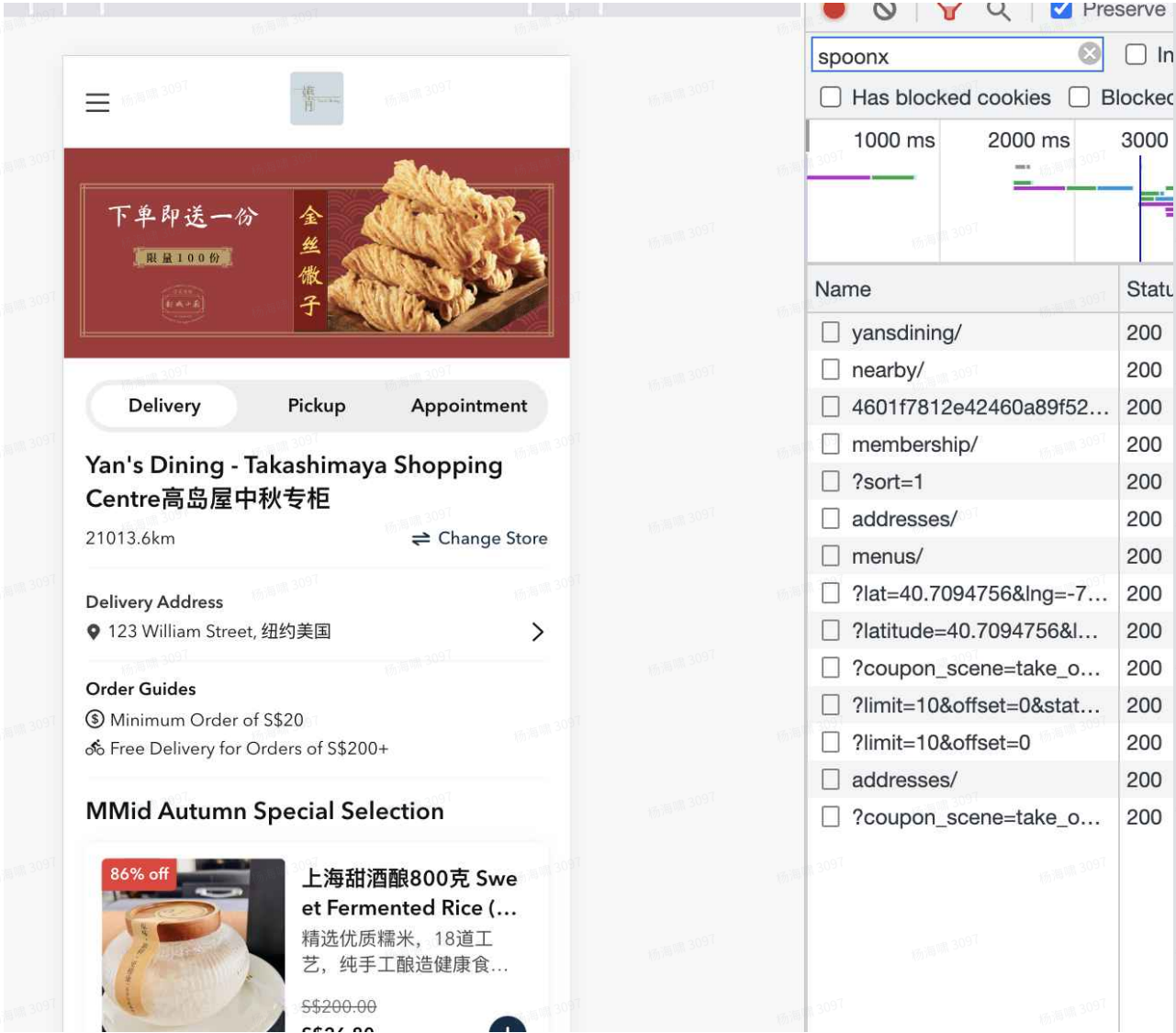


杨海啸 2月21日 15:01（编辑过）
全部菜品请求和前 N 个菜品组请求 loading，网络越差，差距越大



吕心言 2月22日 11:12

@杨海啸 我切了千兆有线网络，全局千兆专线到新加坡，刷新出来倒是快了，不过图片展示

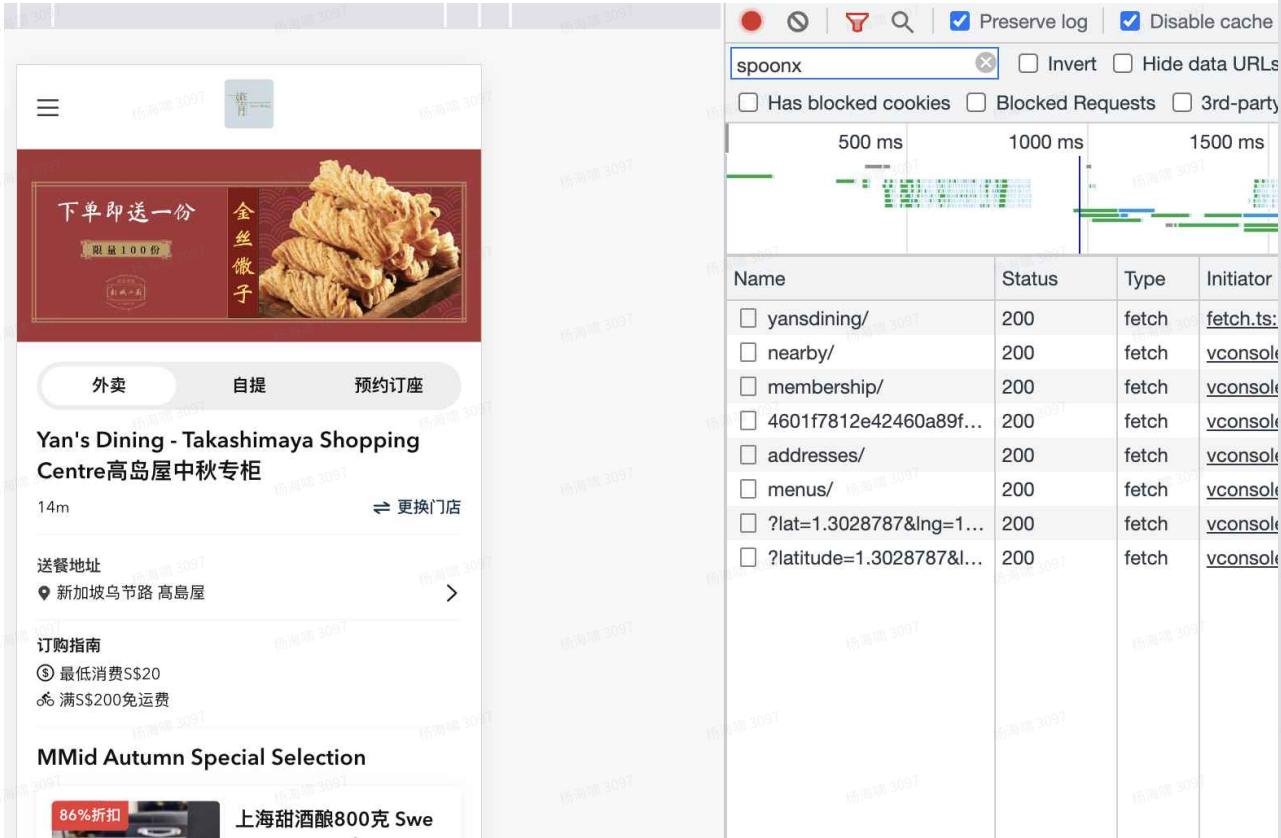


还是一帧帧出来的

杨海啸 2月22日 11:13

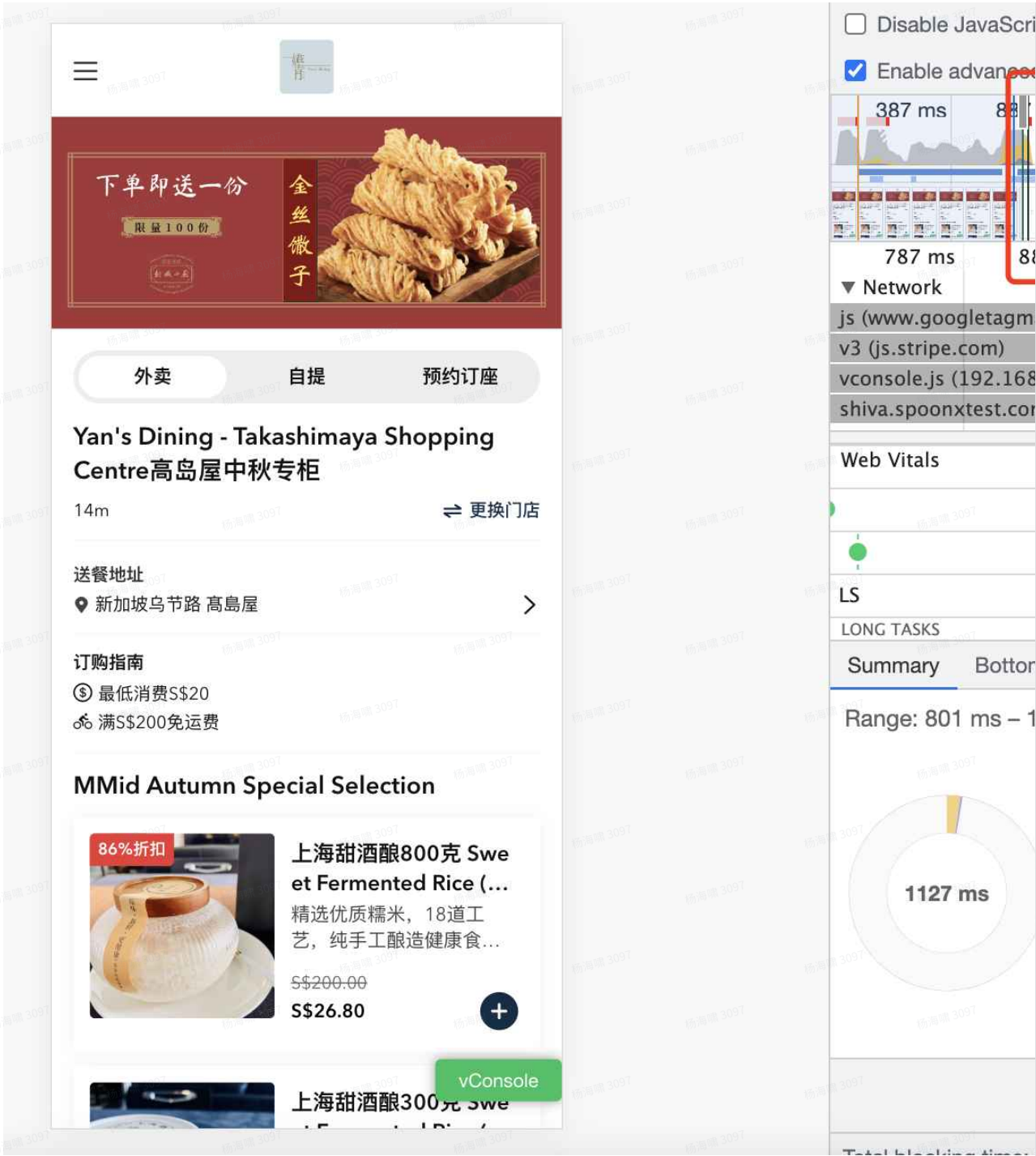
图片都是CDN+懒加载，什么时候下载好什么时候加载出来

优化后，由14个接口优化为8个接口，去掉了重复请求和一些非必要请求



优化后

优化逻辑后，idle（白屏时间）下降了300ms左右，下降到1100ms左右



大接口拆分。独立站首页菜单过多时，影响页面加载过程

历史原因，品类导航和菜单联动导致菜单接口没有做分页。拿鹏程小厨这家店来说，菜品太的情况，前端渲染压力也比较大。图片懒加载（受网络影响）都要很久。

原菜单接口支持分页。菜单页 loading 态…

杨帅 2月21日 14:57
这样可能出现购物车会很晚才出现数据么

杨海啸 2月21日 14:58
应该和原来是差不多的

杨海啸 2月21日 15:05 （编辑过）
相当于是只有第一页菜品请求有 loading 态，但是也会同时请求后续的菜品，但是不影响菜单页 loading 态。理论对购物车没有影响

杨帅 2月22日 10:43
这个接口后端可以优化么，国内的菜比这个复杂多了，也才 100 多 ms

杨海啸 2月22日 10:47
Content Download 的时间会久，鹏程小厨的都 65kb 了，蛮大的

吕心言 2月22日 11:06
我刚去试了下，彭城小厨的 menu，大概是 200ms，这个 1s 多大可能是梯子的问题。加载的最慢的是 strip 和图片的加载

吕心言 2月22日 11:07
图片应该是已经用了 CDN 了吧？有没有做过无损压缩呢？

吕心言 2月22日 11:07
@杨海啸

杨海啸 2月22日 11:07
都是 cdn 的

吕心言 2月22日 11:08
@杨海啸 那感觉好慢，一帧一帧的加载。

杨帅 2月22日 11:09
CDN r.kezaihui.com跟 media.kezaihui.com现在速度上有区别么
这俩一个是全球加速，一个是东南亚地区加速

杨海啸 2月22日 11:10
@吕心言 相同网络环境下，网络越差，差距越大。以上都是小弟的翻墙网络测得，可能梯子不太好

杨海啸 2月22日 11:12
可以假设相同网络环境下同一接口 server response 相同，那么 content 越大，网络越差，content download 时间就越久

杨海啸 2月22日 11:14 （编辑过）
@杨帅 这个怎么加速测试呢

新城市厨

下单就送 (限量100份) 特色地锅系列 Ground Pot [三]

订购指南
⑤ 仅限自取

下单就送 (限量100份)

100%折扣

金丝馓子
【消费满15元即送, 限量100份】 Crispy Dough T...
\$5\$7.80
\$S\$0.00

+

特色地锅系列 Ground Pot Dishes

彭城特色地锅 Ground Pot Dishes (多选)
锅菜起源于苏北和鲁南交界处的微山湖地区, 其...
\$S\$17.80

+

苏北特色菜 Northern Jiangsu Specialities

蟹粉狮子头
Lion's Head Giant Meatball with Crab Roe Paste - ...
\$S\$7.80

+

×

Northern Jiangsu Cuisine

下单就送 (限量100份) (1)

特色地锅系列 Ground Pot Dishes (1)

苏北特色菜 Northern Jiangsu Specialities (22)

精美凉菜 Cold Dishes (9)

铁板系列 Hot Plate Dishes (3)

锅仔系列 Wok Grilled Dishes (5)

美味小龙虾 Crayfish Dishes (3)

家常&小炒 Fine Dishes & Stir Fried Dishes (18)

时令蔬菜 Vegetables (4)

主食面点 Rice & Noodles (2)

精美汤品 Fine Soup (4)

甜食类 Chinese Dessert (2)

饮料 Soft Drinks (7)

啤酒 Beer (5)

白酒 Liquor (8)

起泡酒 Sparkling Wine (3)

澳洲红酒 Australia Wine (8)

解决方案

- 大接口拆分, 推迟加载内容, 减少首屏的 DOM 元素数量

鹏程小厨

Elements Console Performance insights Recorder Sources

● 🔍

☒ Preserve log ☒ Disable cache No throttling

Filter ☐ Invert ☐ Hide data URLs All **Fetch/XHR** JS CSS Img

☐ Has blocked cookies ☐ Blocked Requests ☐ 3rd-party requests

5000 ms 10000 ms 15000 ms 20000 ms

Name	Status	Type	Initiator	Size	Time
<input type="checkbox"/> ?sentry_key=e0cab789c44...	(canceled)	fetch		0 B	13 ms
<input type="checkbox"/> npm-monitoring	200	xhr	index.cd6b904...	437 B	1.5 s
<input type="checkbox"/> gen_204?csp_test=true	200	xhr	index.cd6b904...	90 B	141 ms
<input type="checkbox"/> pengchengxiaochu/	200	fetch	index.cd6b904...	808 B	640 ms
<input type="checkbox"/> nearby/	200	fetch	index.cd6b904...	2.2 kB	574 ms
<input type="checkbox"/> ?sentry_key=e0cab789c44...	200	fetch	index.cd6b904...	229 B	136 ms
<input type="checkbox"/> 6	200	xhr	out-4.5.42.js:1	553 B	1.7 s
<input type="checkbox"/> ad76321cab06453983c366...	200	fetch	index.cd6b904...	584 B	579 ms
<input type="checkbox"/> 6	200	xhr	out-4.5.42.js:1	552 B	538 ms
<input type="checkbox"/> ?sort=1	200	fetch	index.cd6b904...	484 B	576 ms
<input checked="" type="checkbox"/> menus/	200	fetch	index.cd6b904...	65.9 kB	1.31 s
<input type="checkbox"/> addresses/	200	fetch	index.cd6b904...	471 B	632 ms
<input type="checkbox"/> ?limit=10&offset=0&status=...	200	fetch	index.cd6b904...	496 B	577 ms

Queued at 11.15

Started at 12.76

Resource Sched

Queueing

Connection Star

Stalled

Request/Respor

Request sent

Waiting for se

response

Content Down

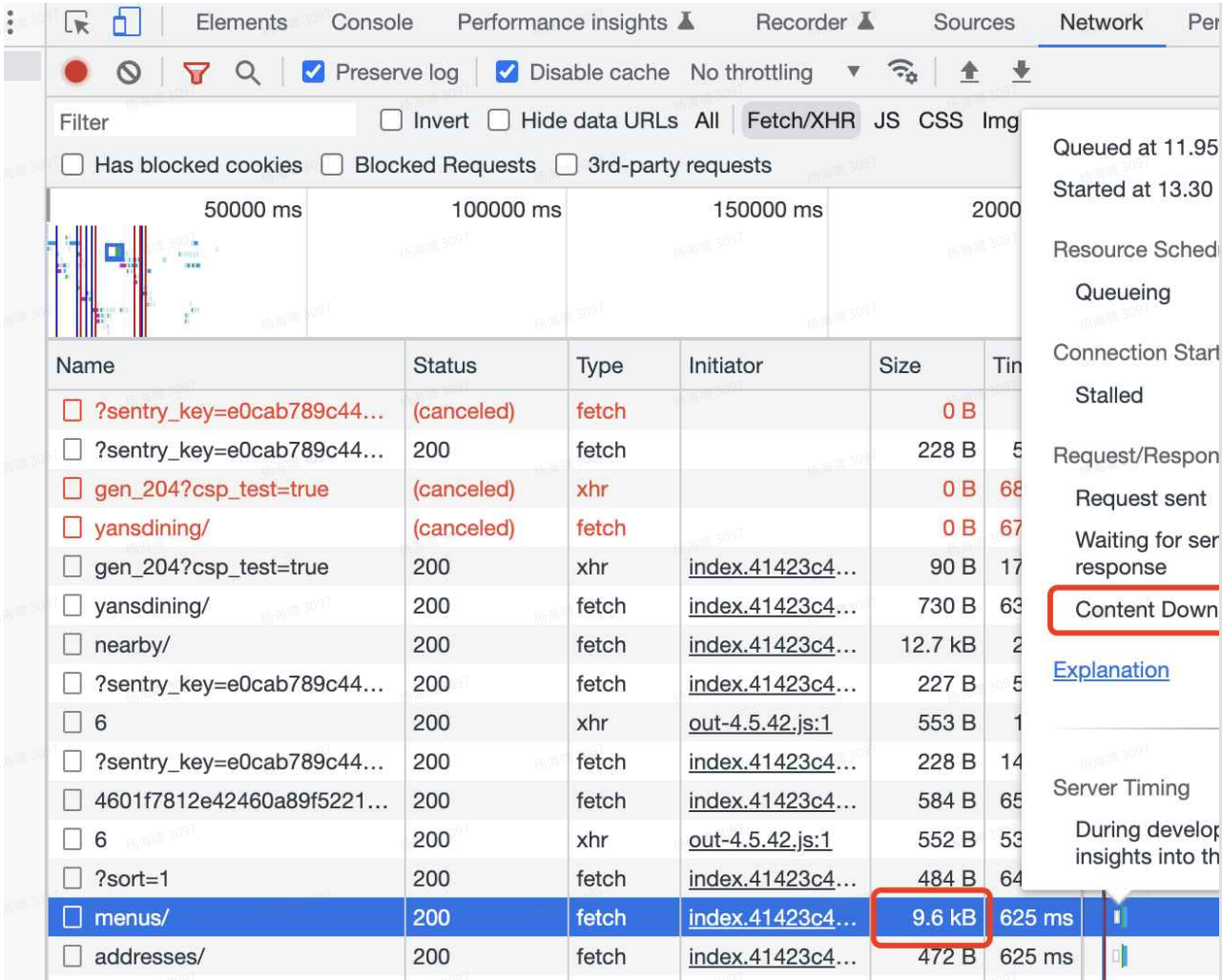
Explanation

Server Timing

During develo

insights into t

艳青

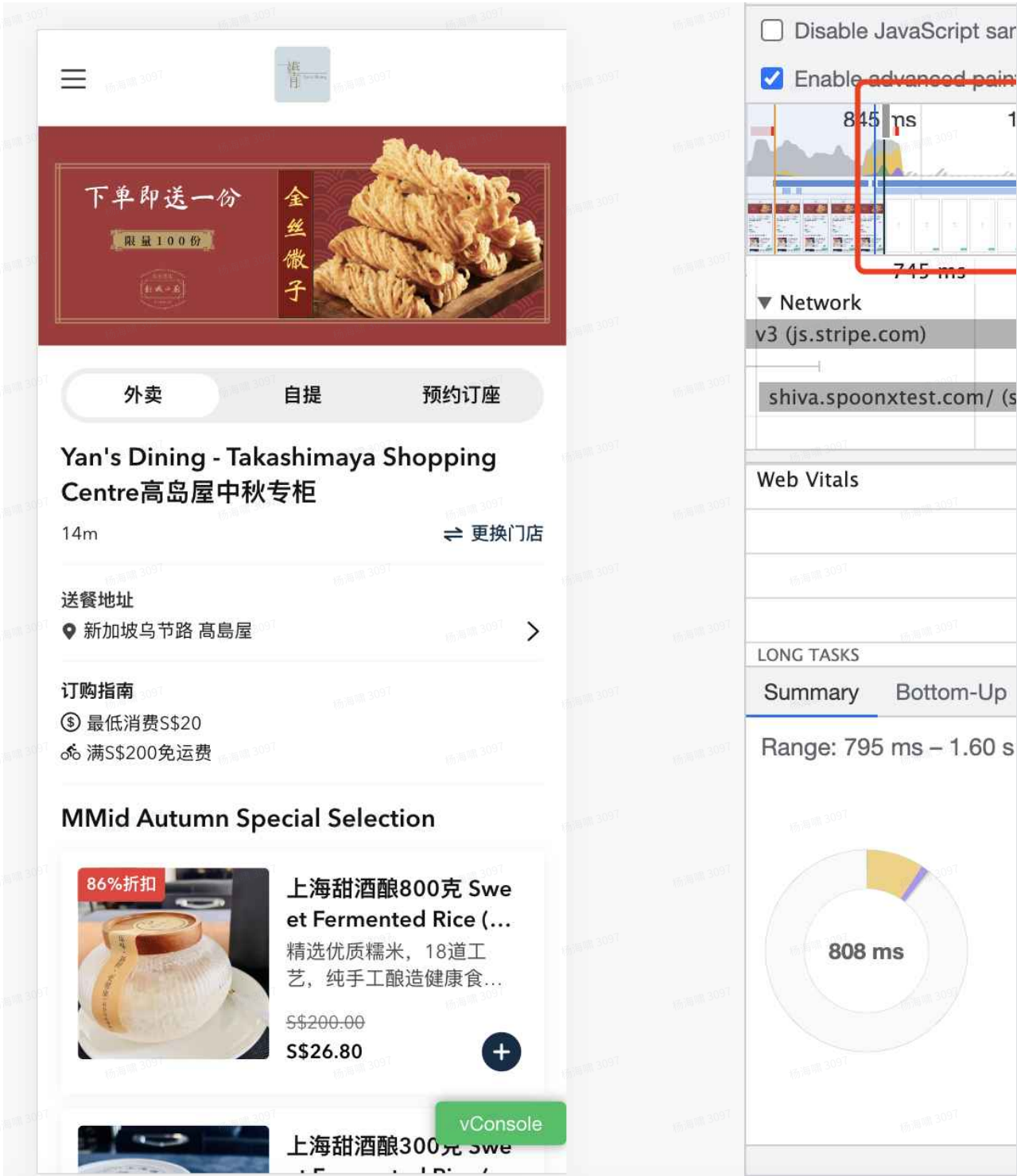


优化

菜单页优化+分页及缓存、去除前端排序逻辑。原菜单接口支持分页。菜单页 loading 态仅显示前几页菜品，通过下拉刷新加载更多菜品。通过加载更多菜品选择器以及超出首屏部分的菜品，首屏加载时，其实只需要请求第一页。后续静默请求剩余菜品，静默请求时间会有效下降至少 100-500ms。购物车也可以等菜品后台静默处理

优化后

优化逻辑后，idle（白屏时间）下降了**400ms**左右，下降到**800ms**左右



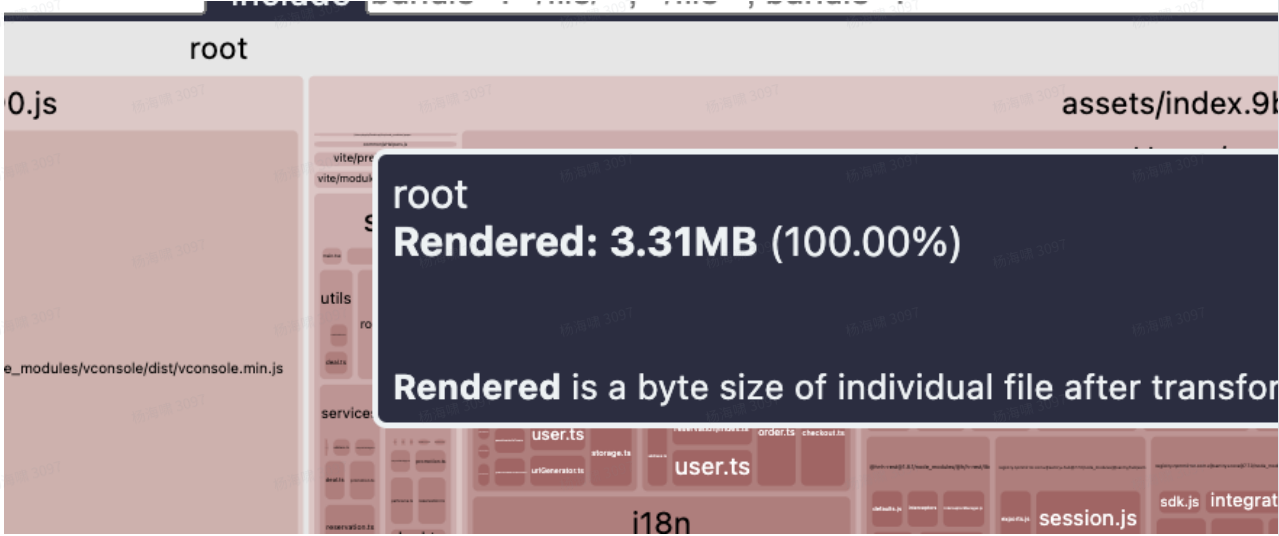
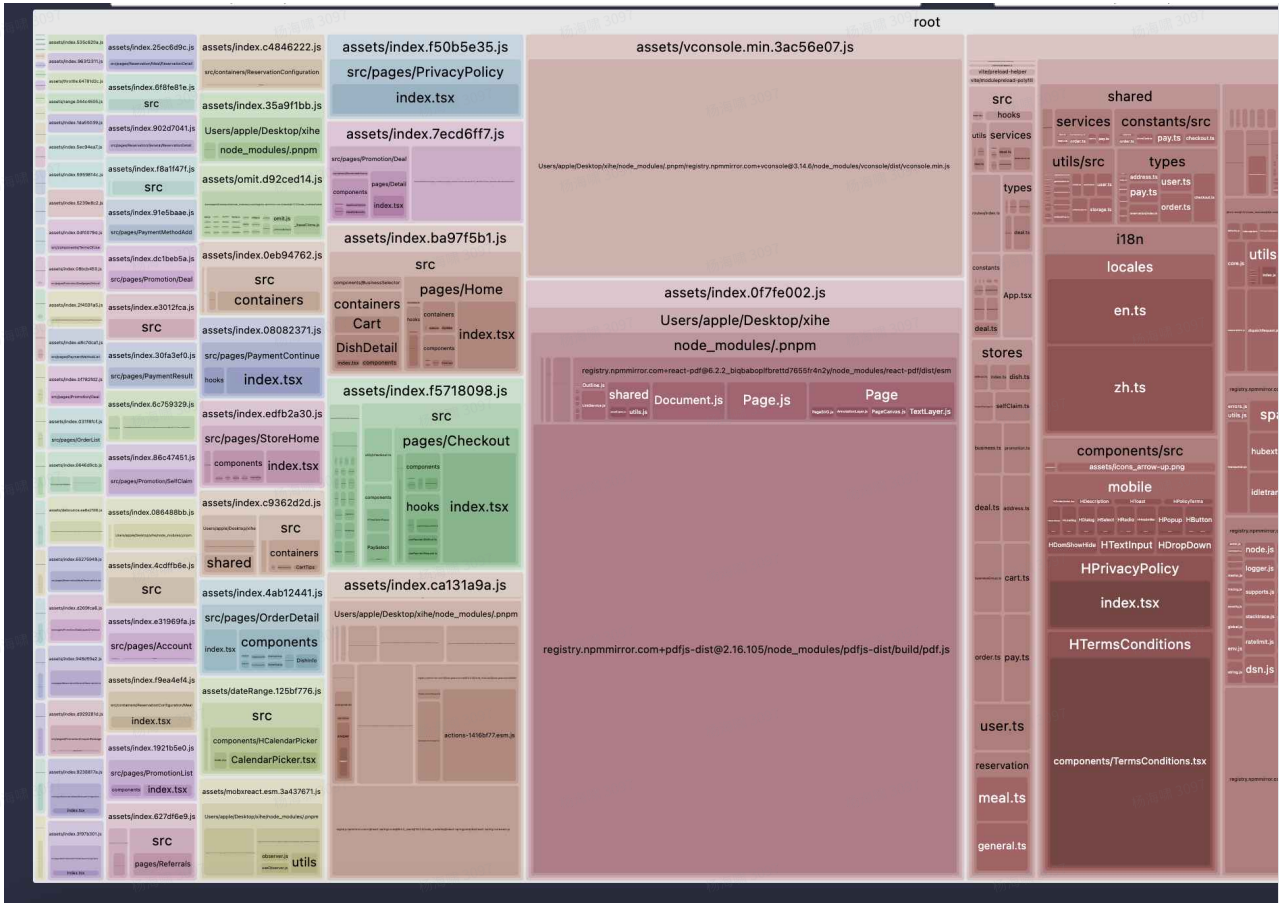
静态资源CDN域名 dns-prefetch 及首屏涉及服务域名 pre-connect

- **DNS-prefetch** (DNS 预获取) 是尝试在请求资源之前解析域名。这可能是后面要加载的目标。

分析依赖

安装依赖分析插件 **Rollup Plugin Visualizer**，发现依赖里第三方包 **lodash** 被全量引入，lodash 包，大约减少包500kb

优化前



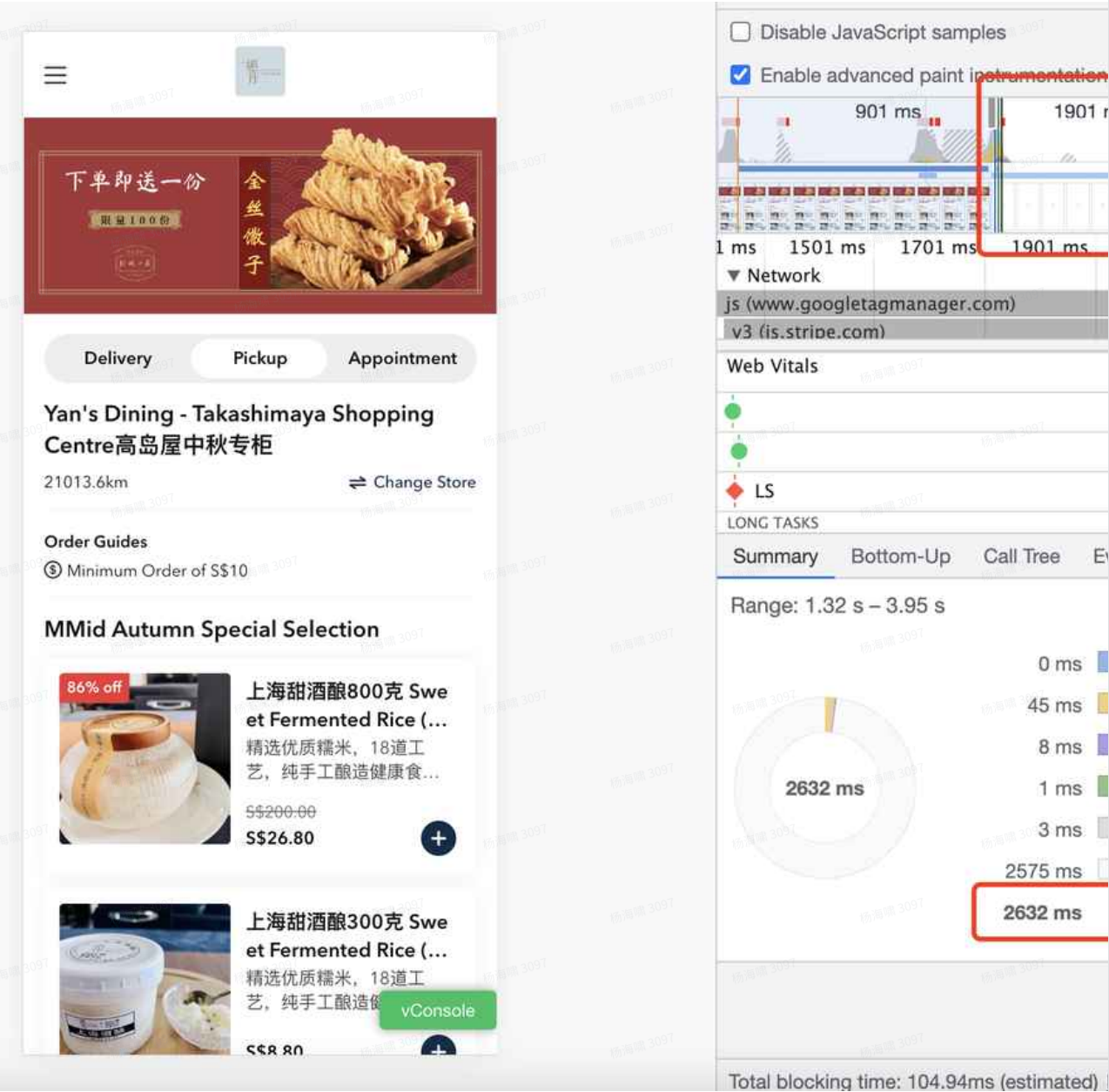
总结

优化角度



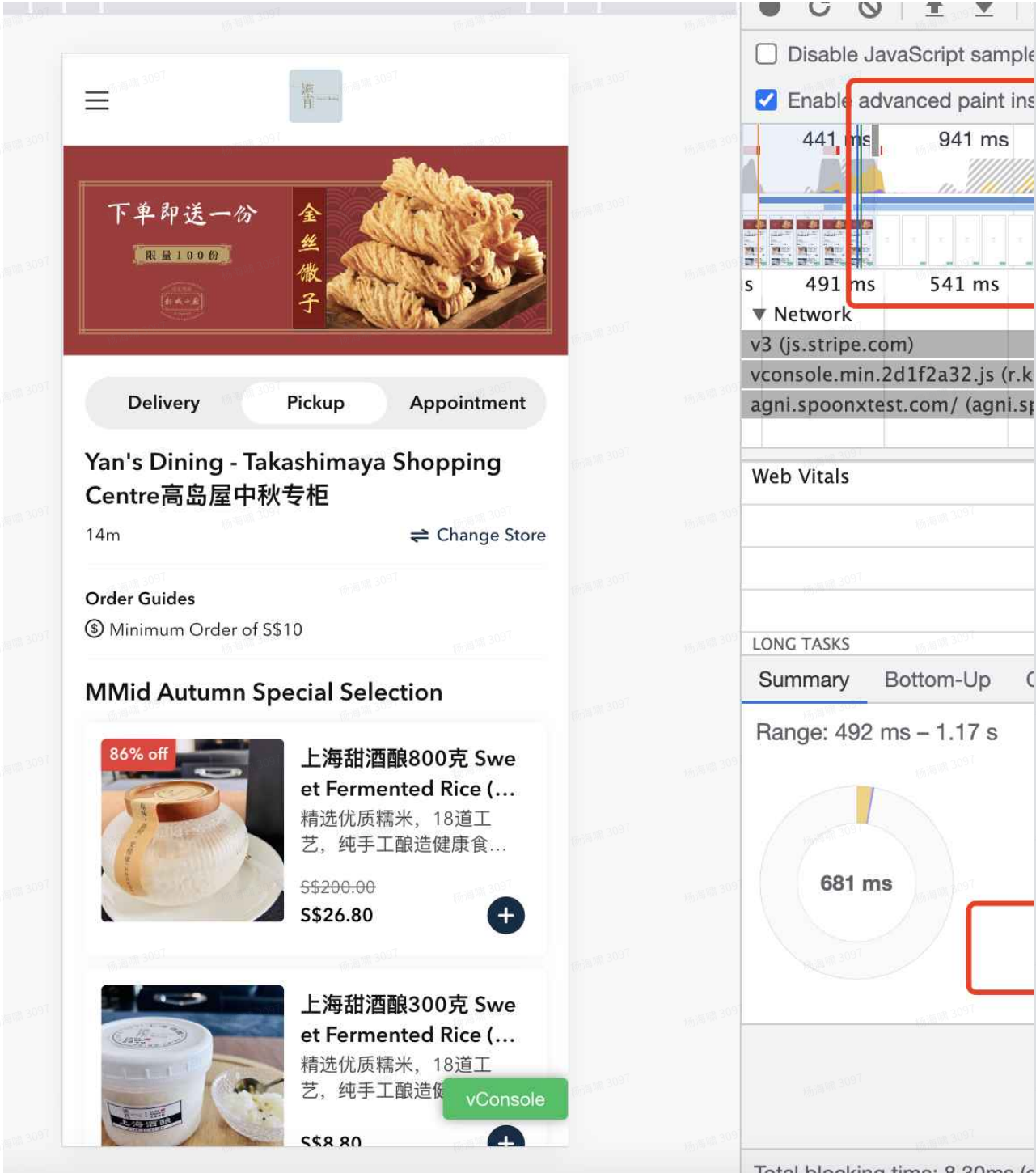
最终效果

优化前



优化后

在新加坡全局代理节点（不考虑SSL建连的情况）下，白屏时间(idle)由2500ms-3000ms左



7月数据。后续接入腾讯RUM监控后，性能表现国际化项目均为健康

