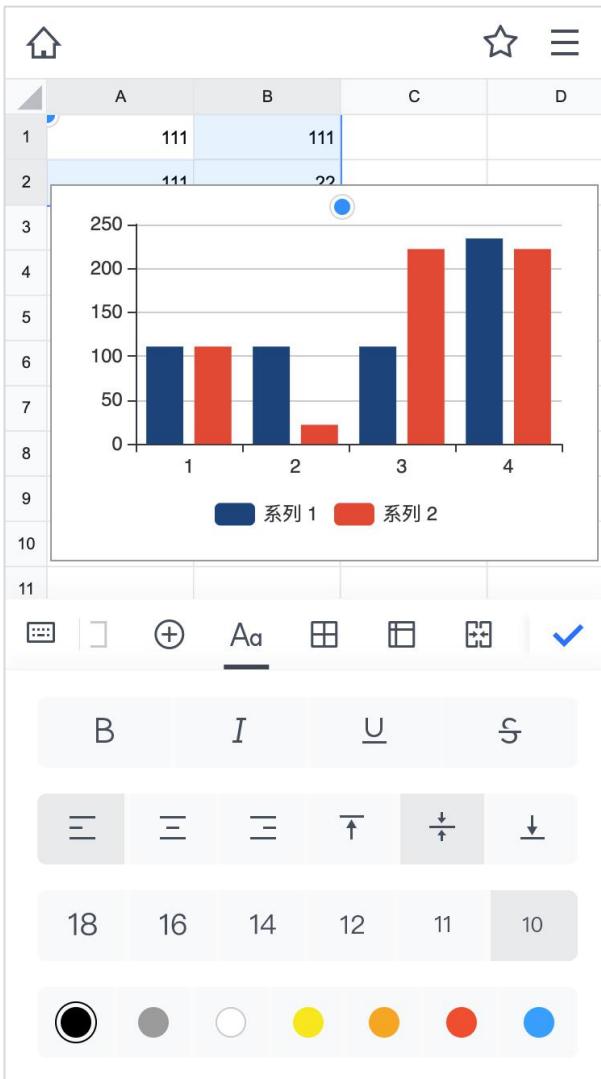


腾讯文档 *Web Worker* 多线程实践

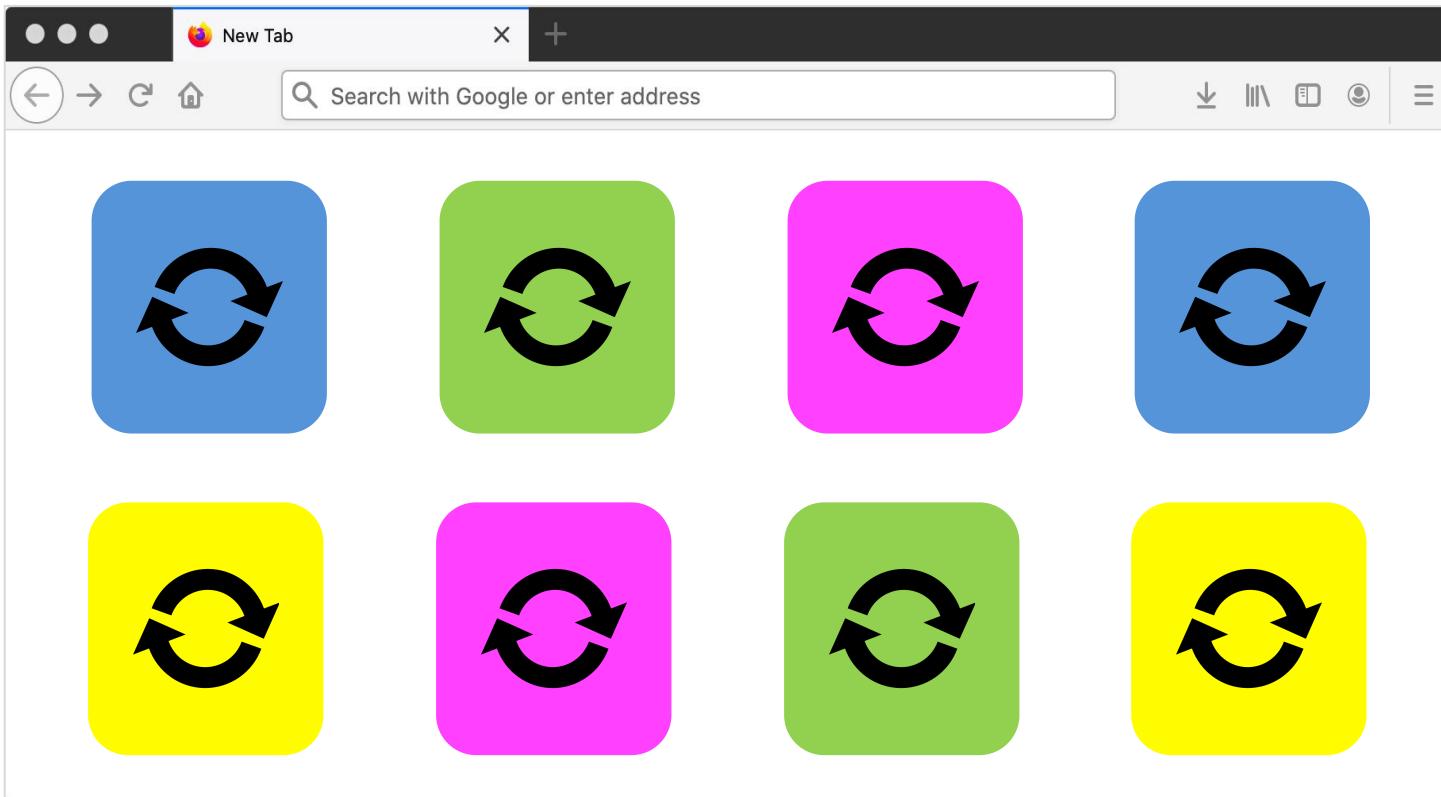
陈韩杰 [@CntChen](#)

2021-11

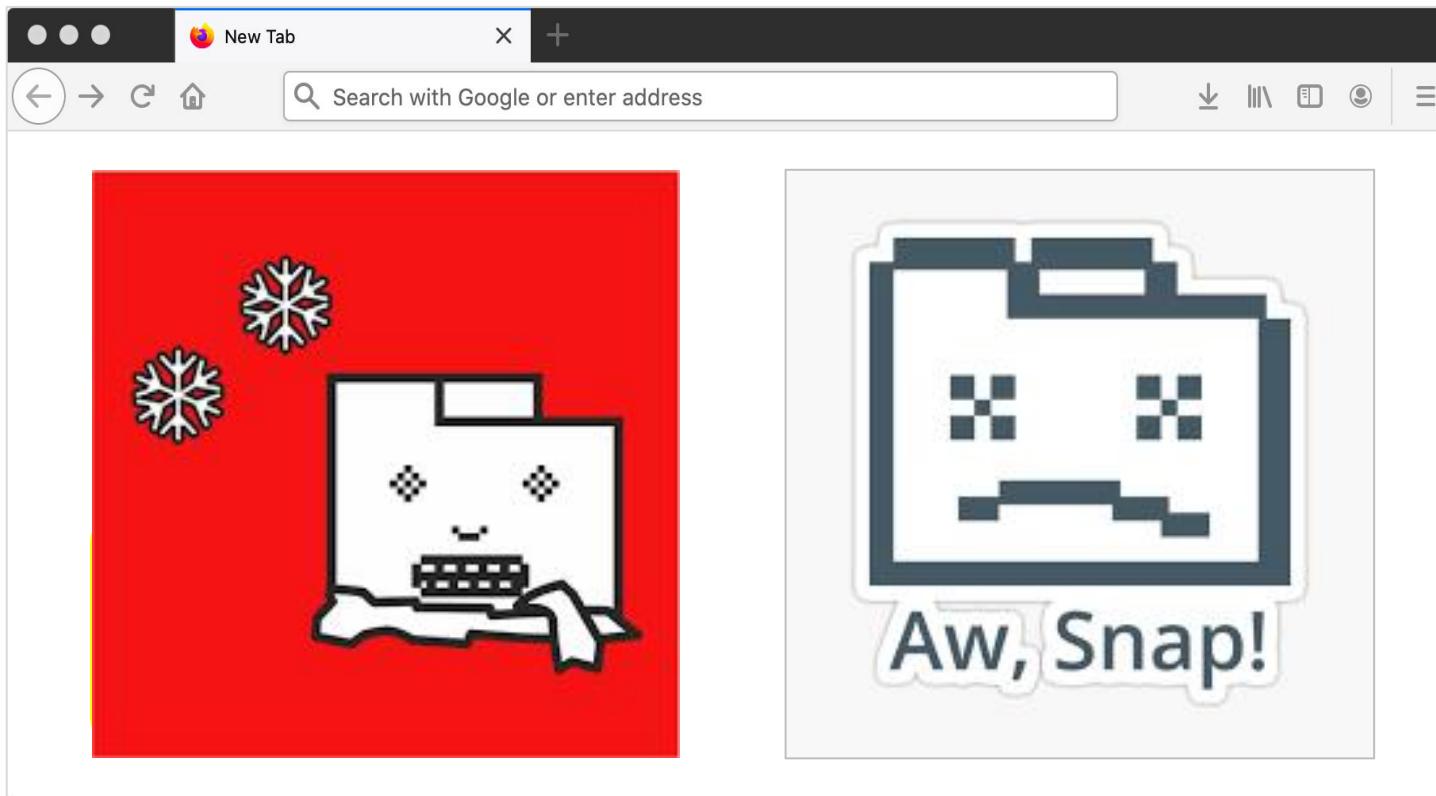
做前端最好的年代



复杂的业务逻辑



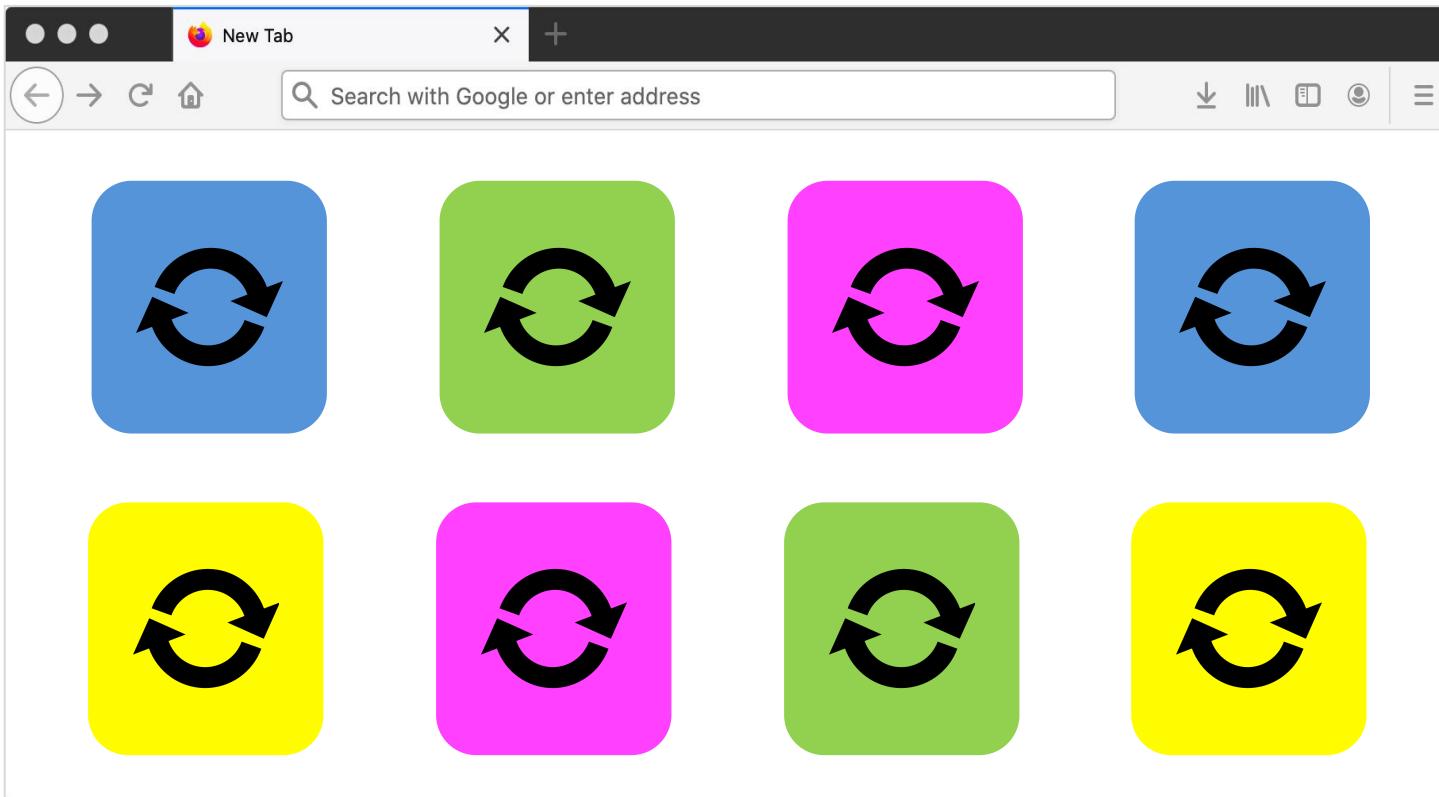
复杂的业务逻辑



取悦用户？



主线程过载



目录

1. *Web Worker* 简介
2. *Worker* 多线程设计
3. 腾讯文档实践

发展历史

- 2009 年 Worker 草案
- 2012 年支持主流浏览器
- 当下无需担心兼容性

Web Workers

W3C Working Draft 24 September 2015

This Version:

<http://www.w3.org/TR/2015/WD-workers-20150924/>

Latest Published Version:

<http://www.w3.org/TR/workers/>

Latest Editor's Draft:

<https://html.spec.whatwg.org/multipage/workers.html>

Previous Versions:

<http://www.w3.org/TR/2012/CR-workers-20120501/>

<http://www.w3.org/TR/2012/WD-workers-20120313/>

<http://www.w3.org/TR/2011/WD-workers-20110901/>

<http://www.w3.org/TR/2011/WD-workers-20110310/>

<http://www.w3.org/TR/2011/WD-workers-20110208/>

<http://www.w3.org/TR/2009/WD-workers-20091029/>

<http://www.w3.org/TR/2009/WD-workers-20090423/>

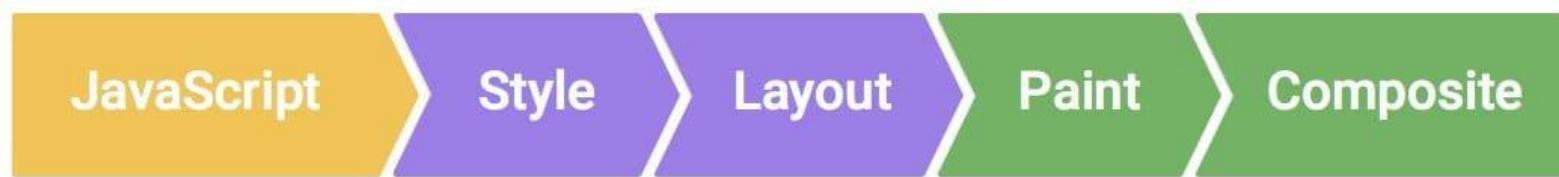
Editor:

[Ian Hickson](#), Google, Inc.

*The Worker interface spawns real
OS-level threads. -- [mdn](#)*

主线程和多线程

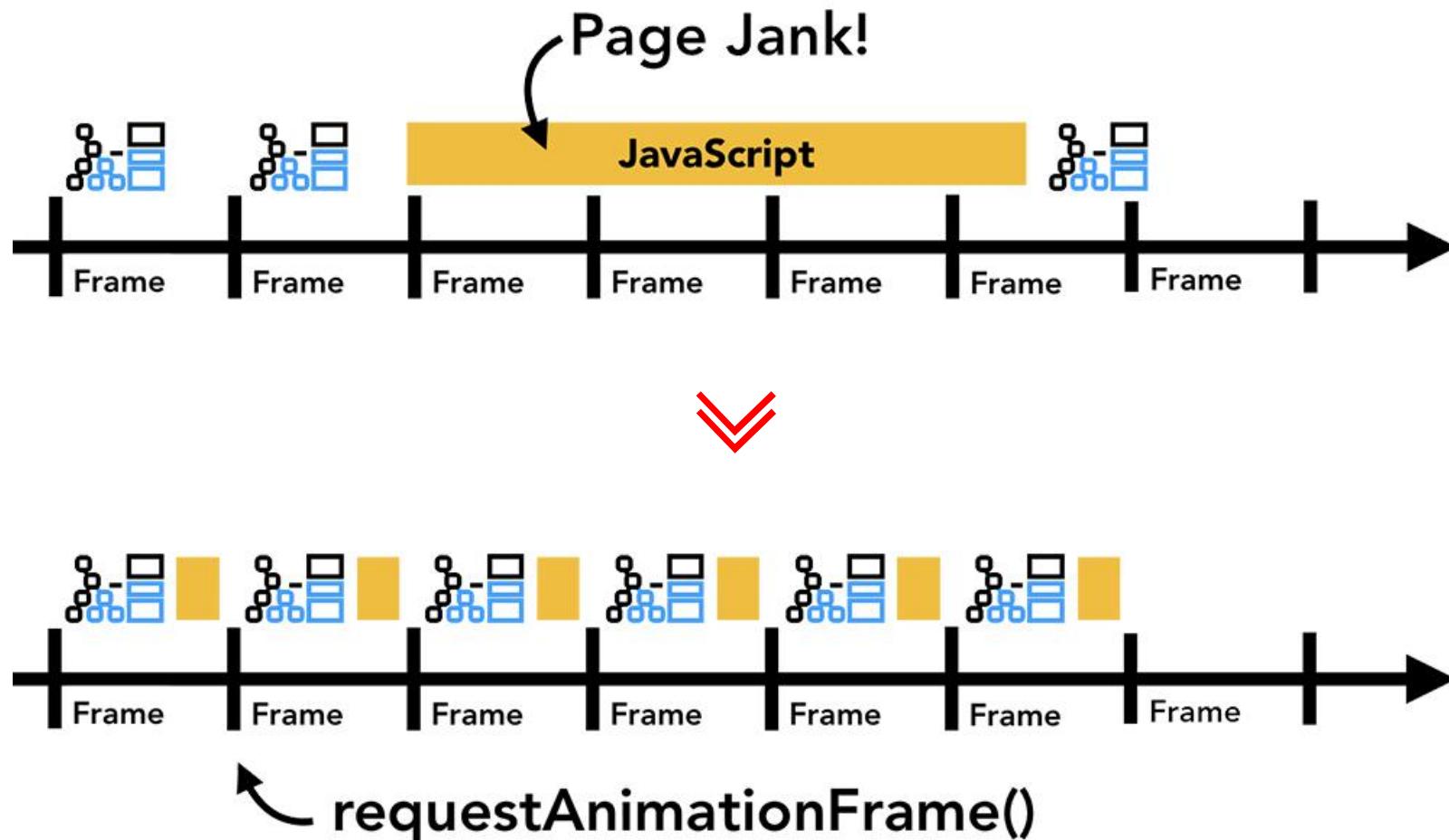
- 页面主线程



- JS 执行阻塞页面渲染



逻辑异步化拆分

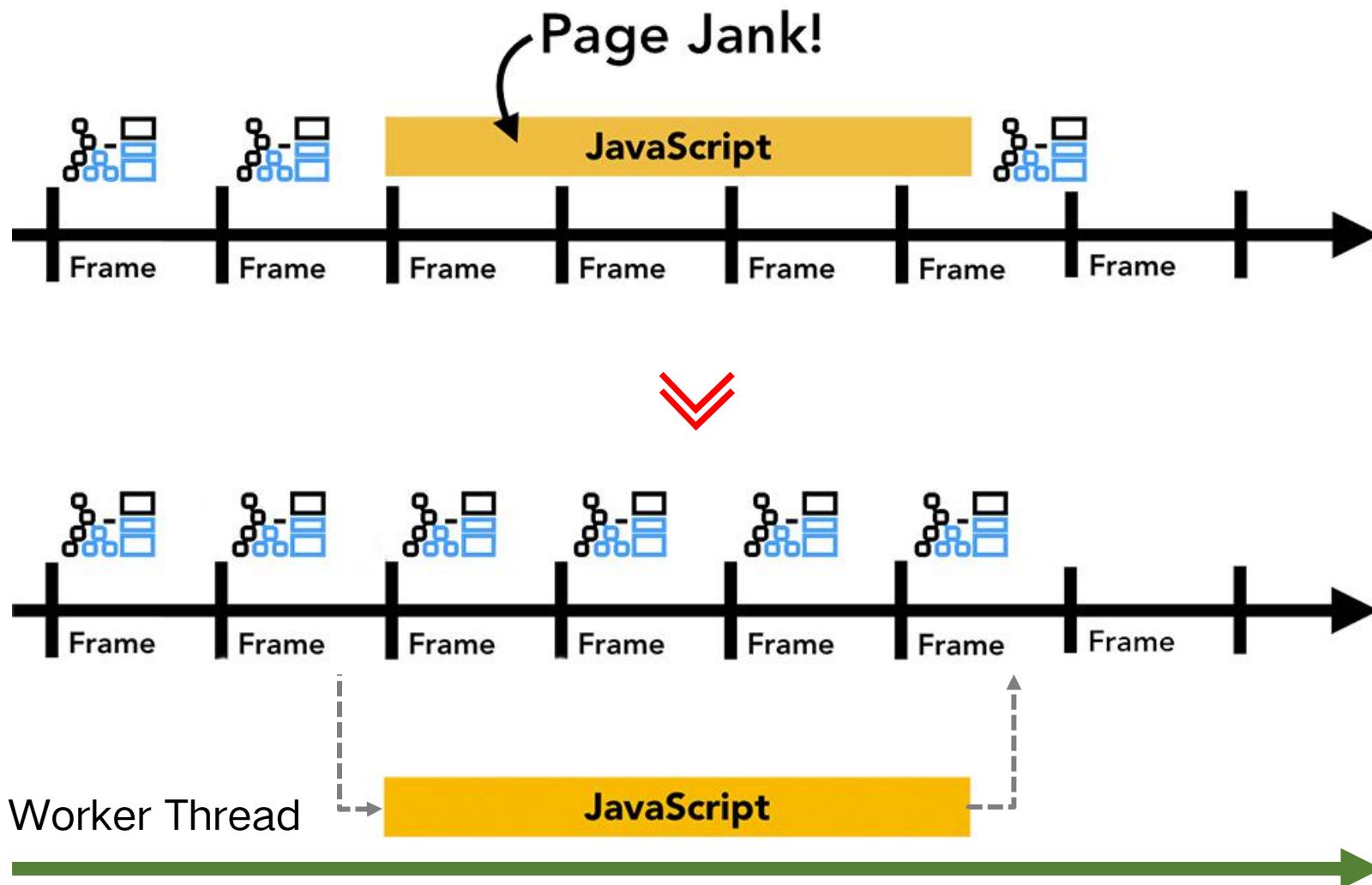


异步拆分难点

- 不是所有 JS 逻辑都可拆分
- 难以把控拆分粒度



Worker 一步到位



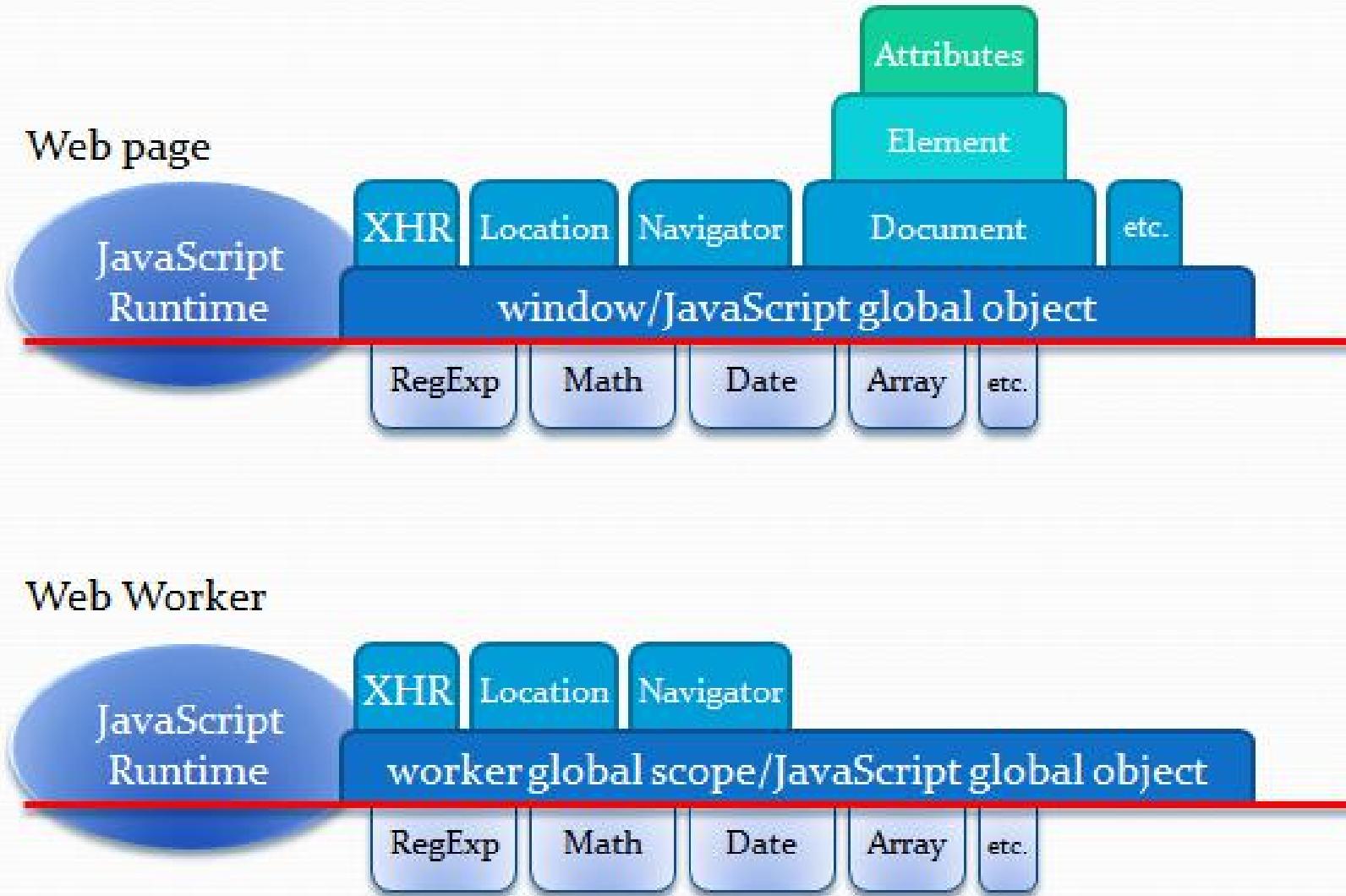
Worker API

```
// main.js
const worker = new Worker('./worker.js');
worker.postMessage('hello');
worker.onmessage = (event) => {
  console.log(event.data); // `world`
}

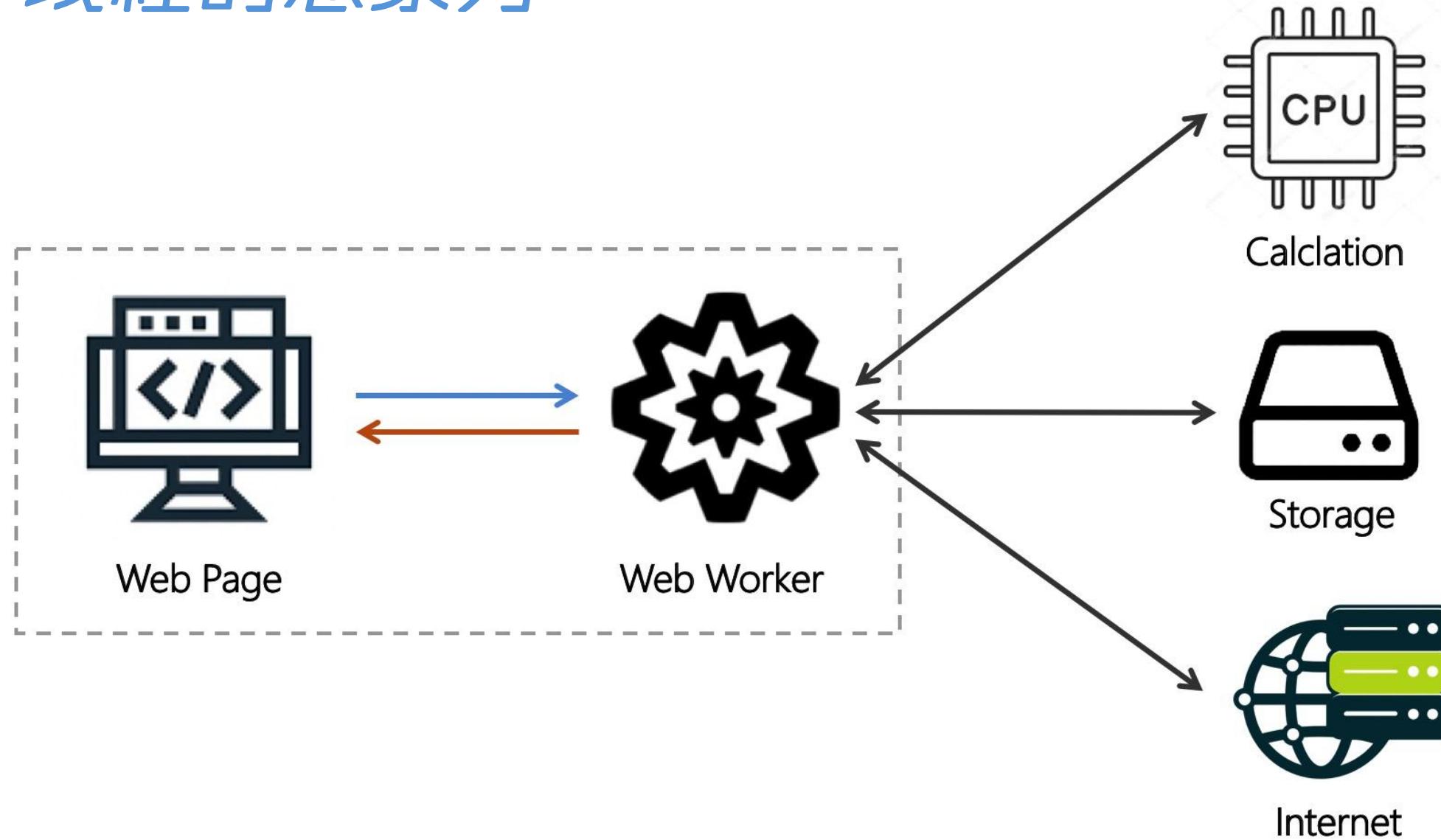
// worker.js
self.onmessage = (event) => {
  console.log(event.data); // `hello`
  postMessage('world');
}
```

与主线程的异同

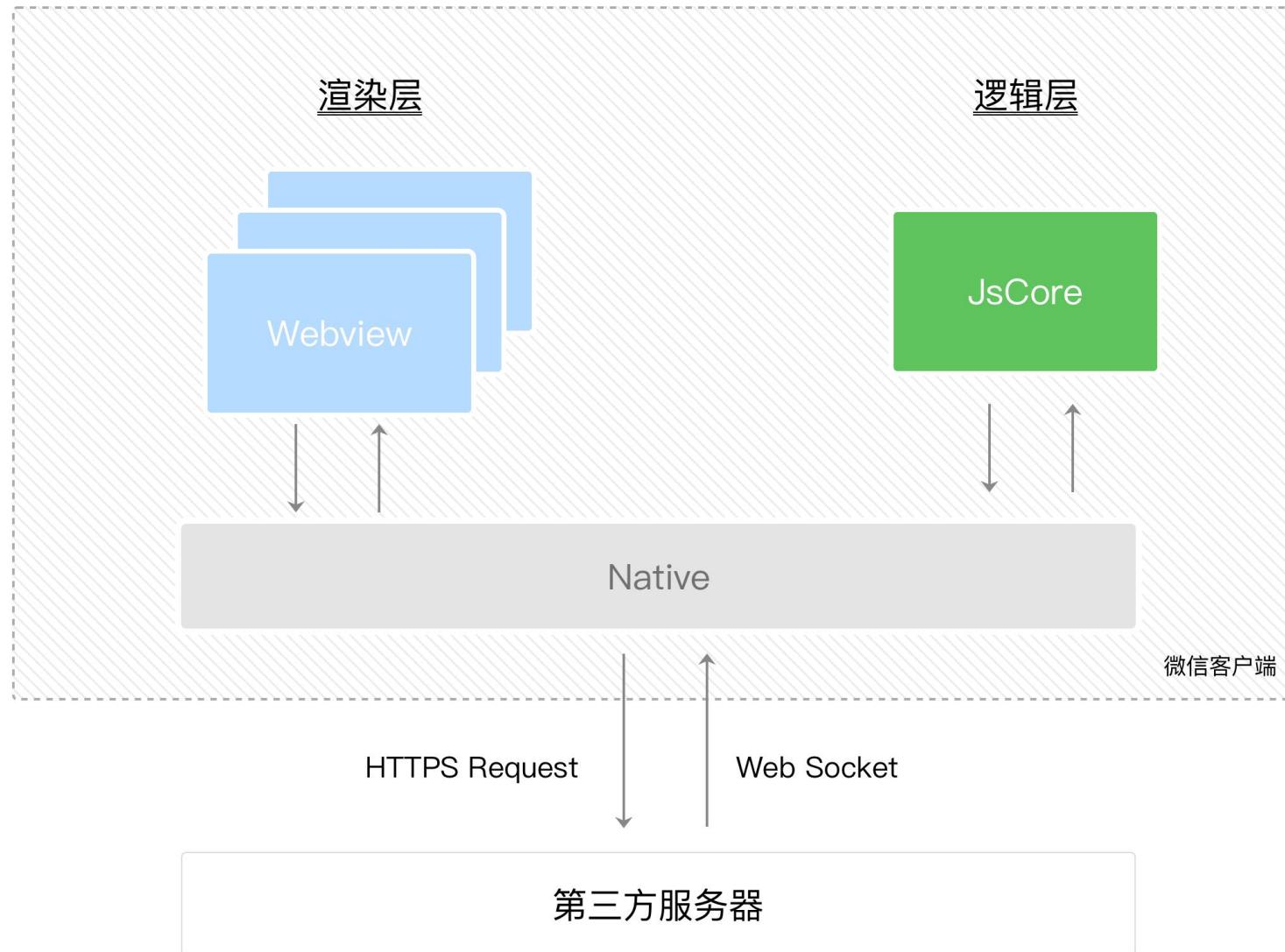
- 完整的 JS 运行环境
- 不能操作 View
- AJAX / indexedDB



多线程的想象力



多线程的想象力



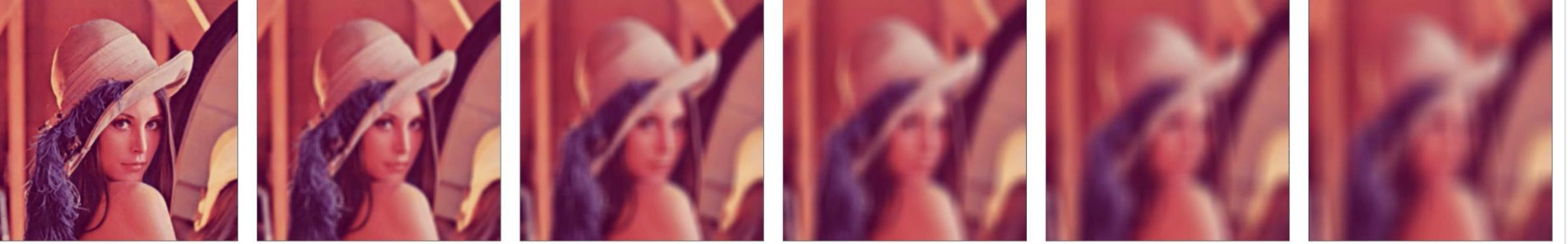
目录

1. *Web Worker* 简介
2. *Worker* 多线程设计
3. 腾讯文档实践

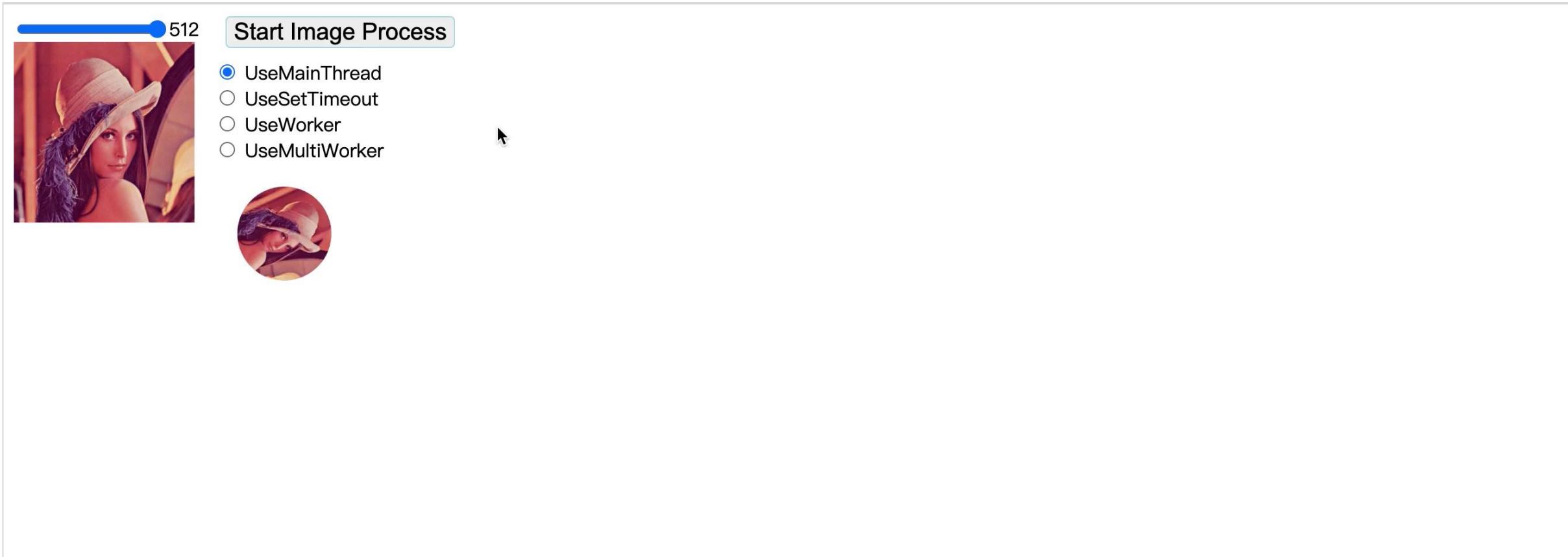
Worker Demo

512 Start Image Process 6233 ms

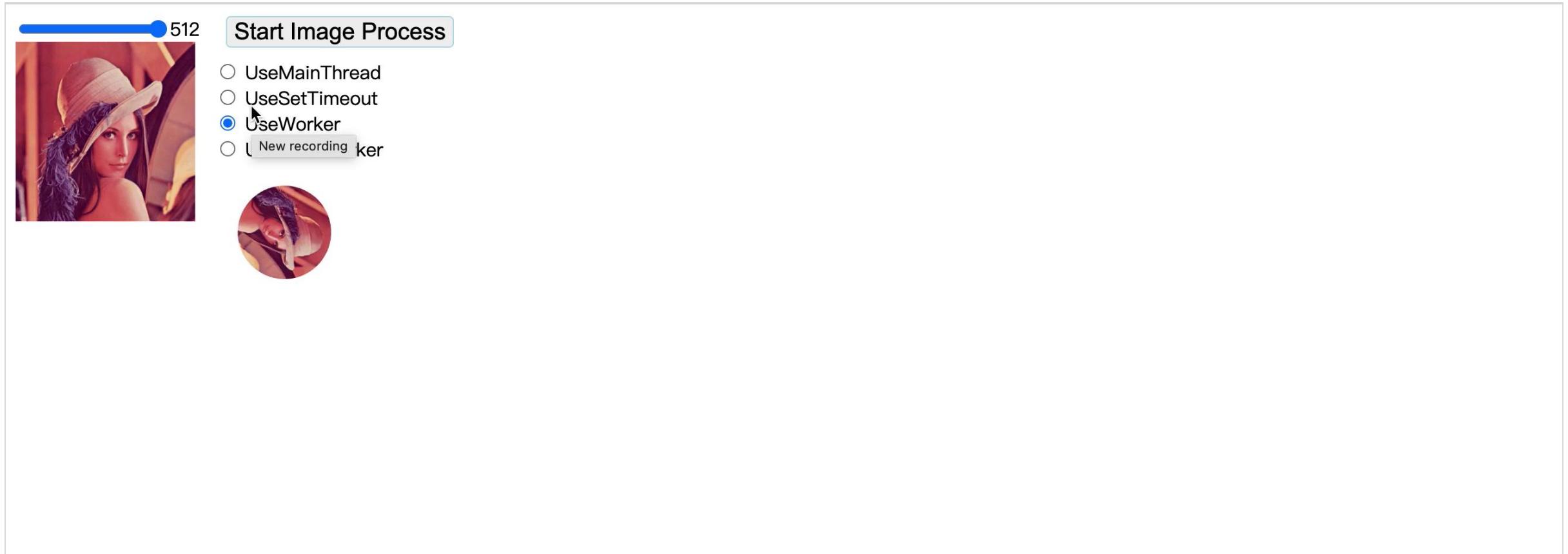
- UseMainThread
- UseSetTimeout
- UseWorker
- UseMultiWorker



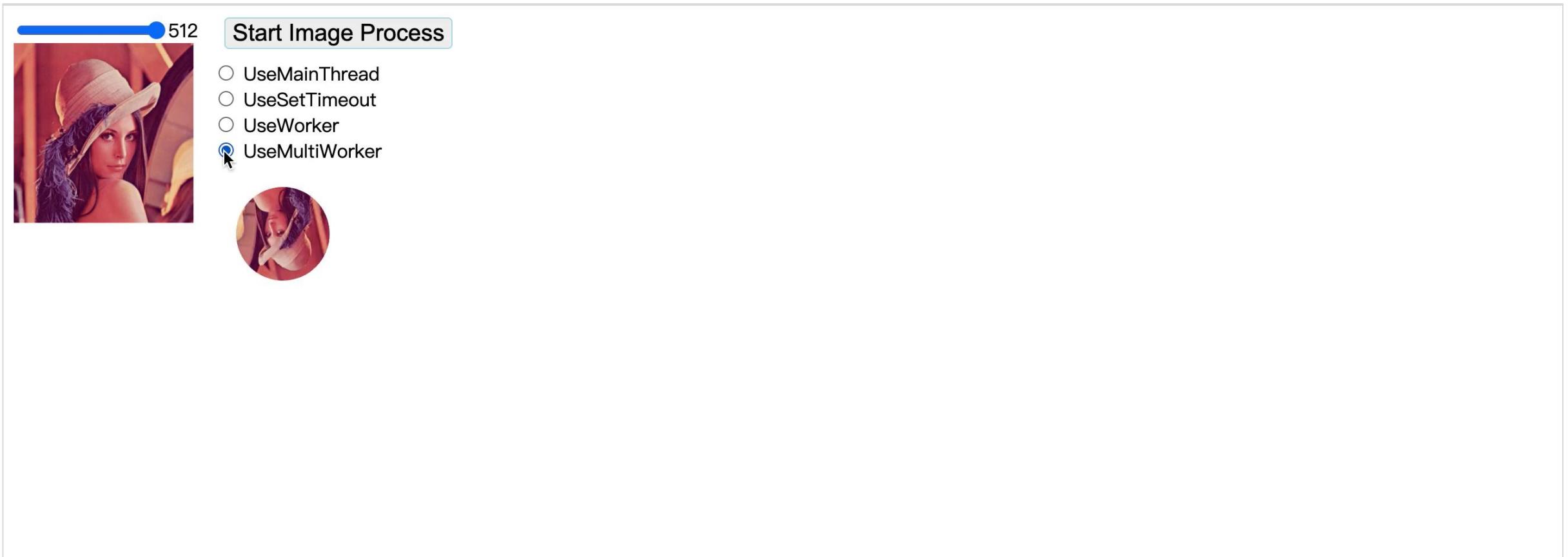
主线程运行



Worker 线程运行



多个 Worker 线程

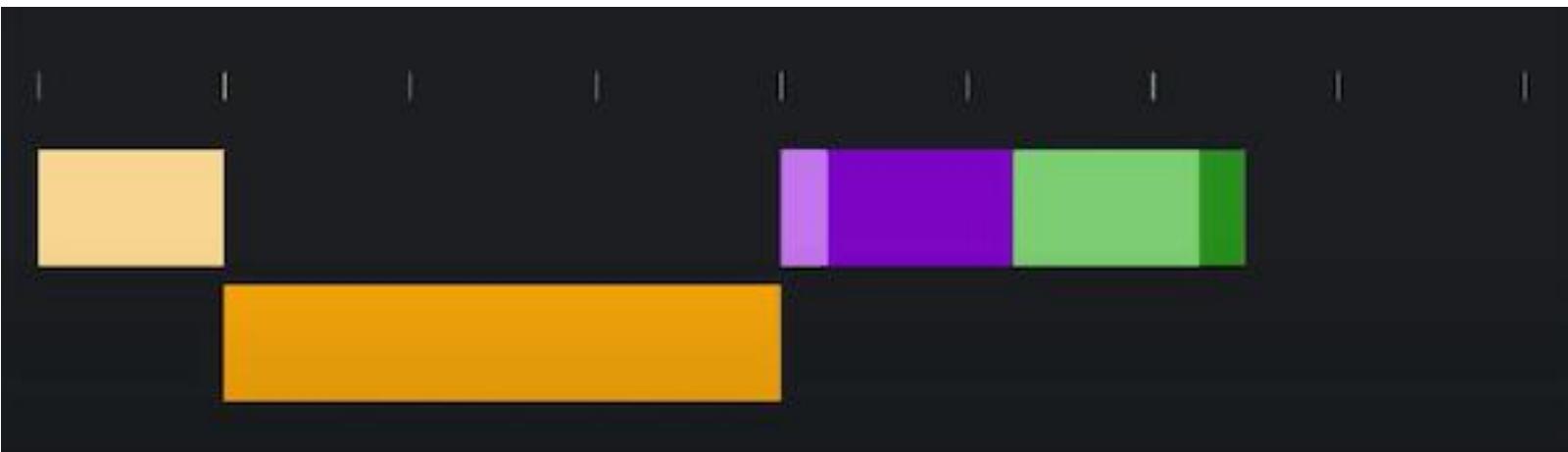
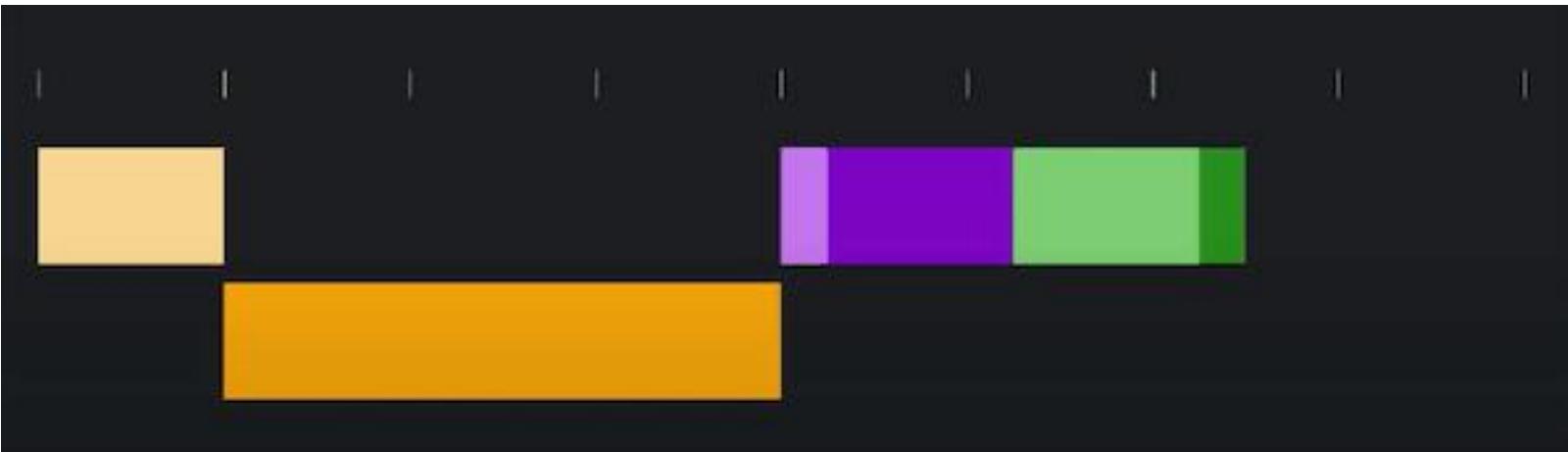


运行对比

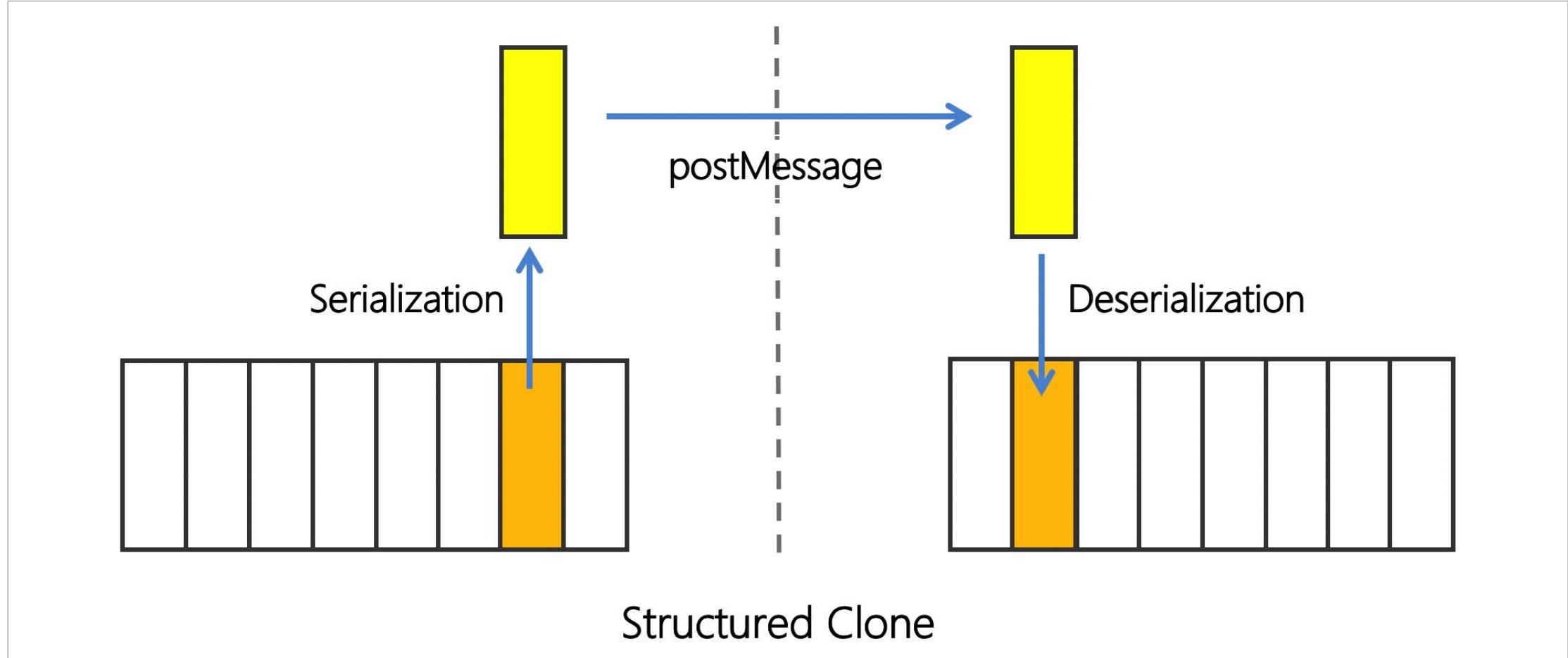
- 提升页面可用性
- 提升页面性能?

运行方式	页面渲染	执行时间 (ms)
主线程	卡顿	6236
主线程 setTimeout	卡顿	6356
单 Worker	不卡	6511
多 Worker	不卡	4151

考虑通信规模

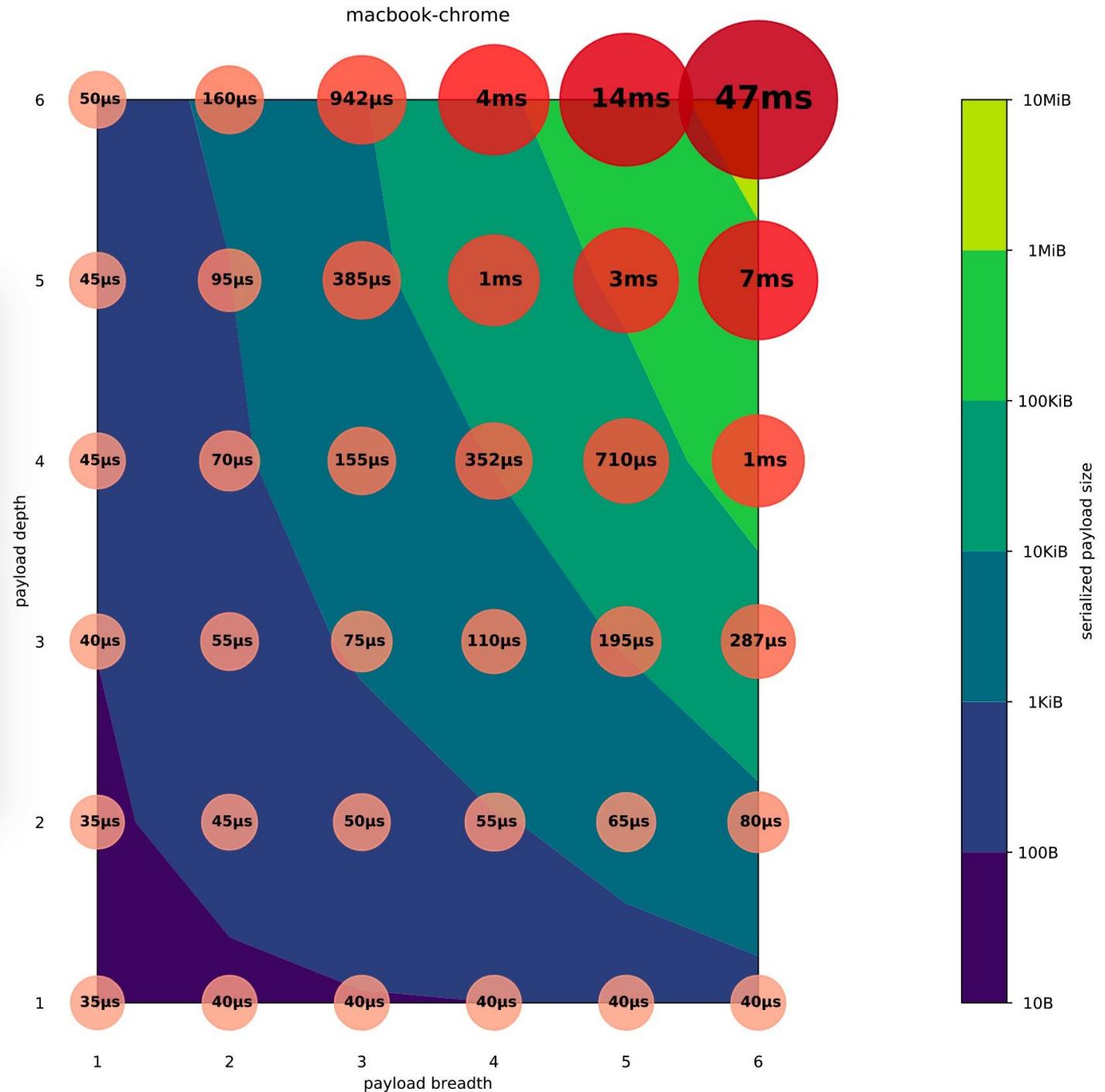


考虑通信规模



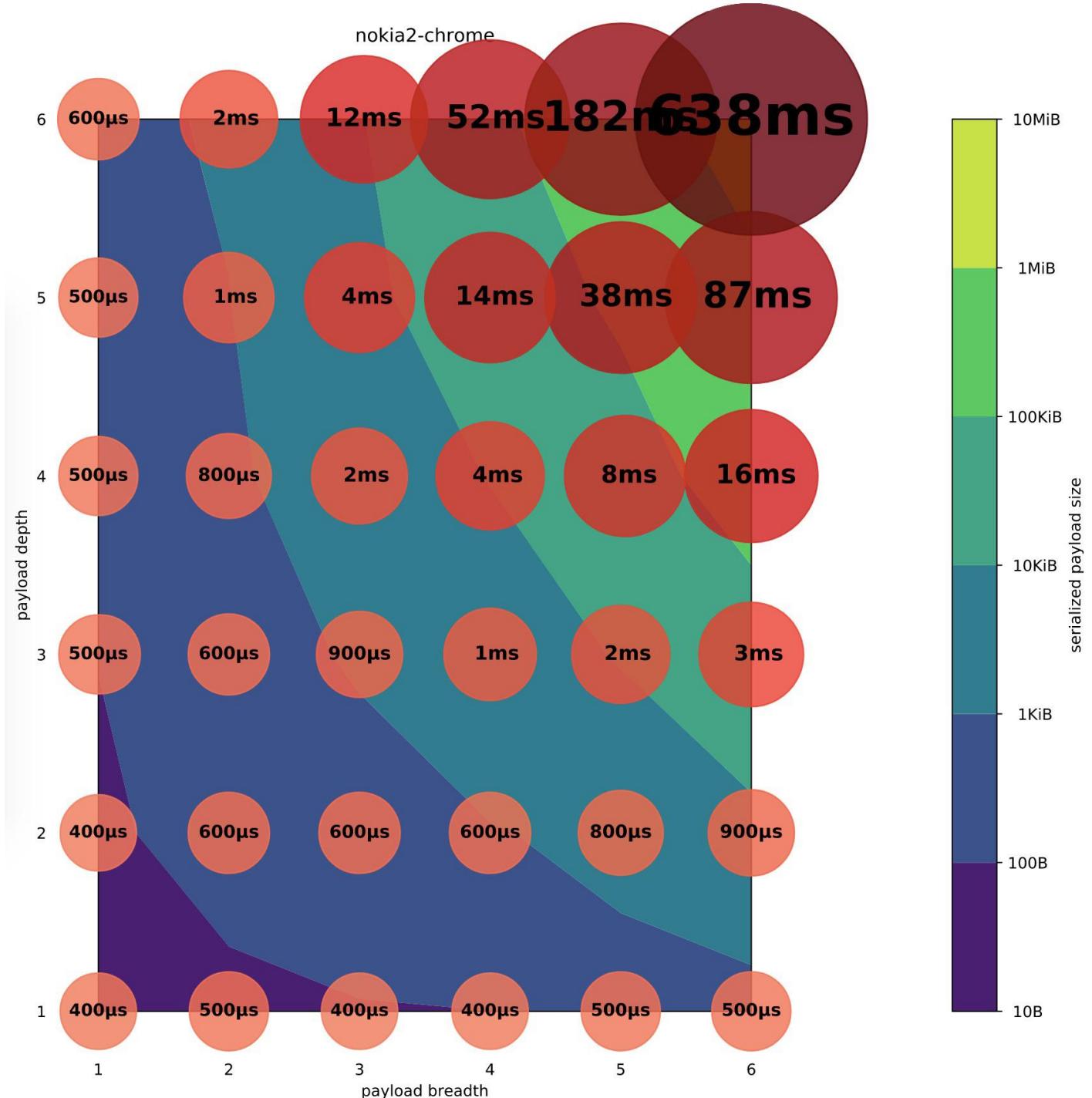
考虑通信规模

- 10MB 耗时 47ms
- 16ms 传输 1000KB



考虑通信规模

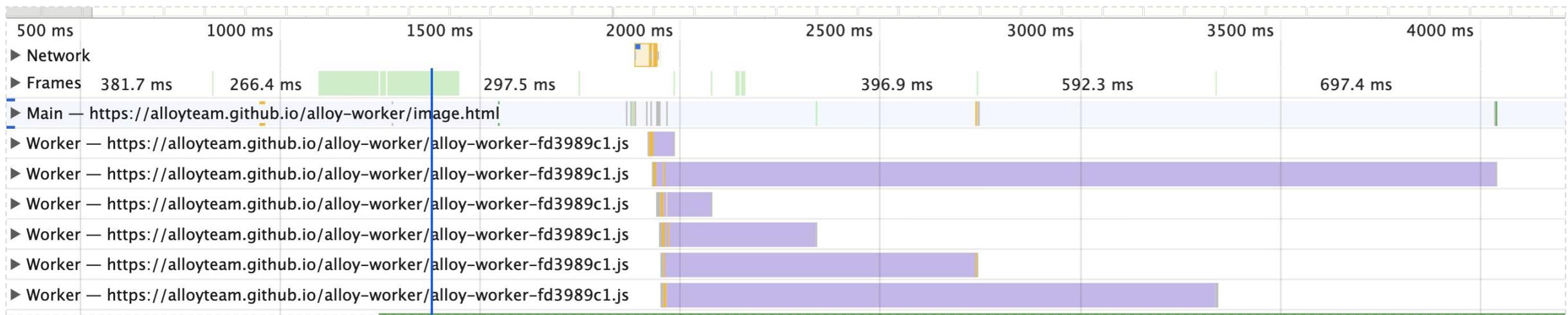
- 10MB 耗时 638ms
- 16ms 传输 10KB



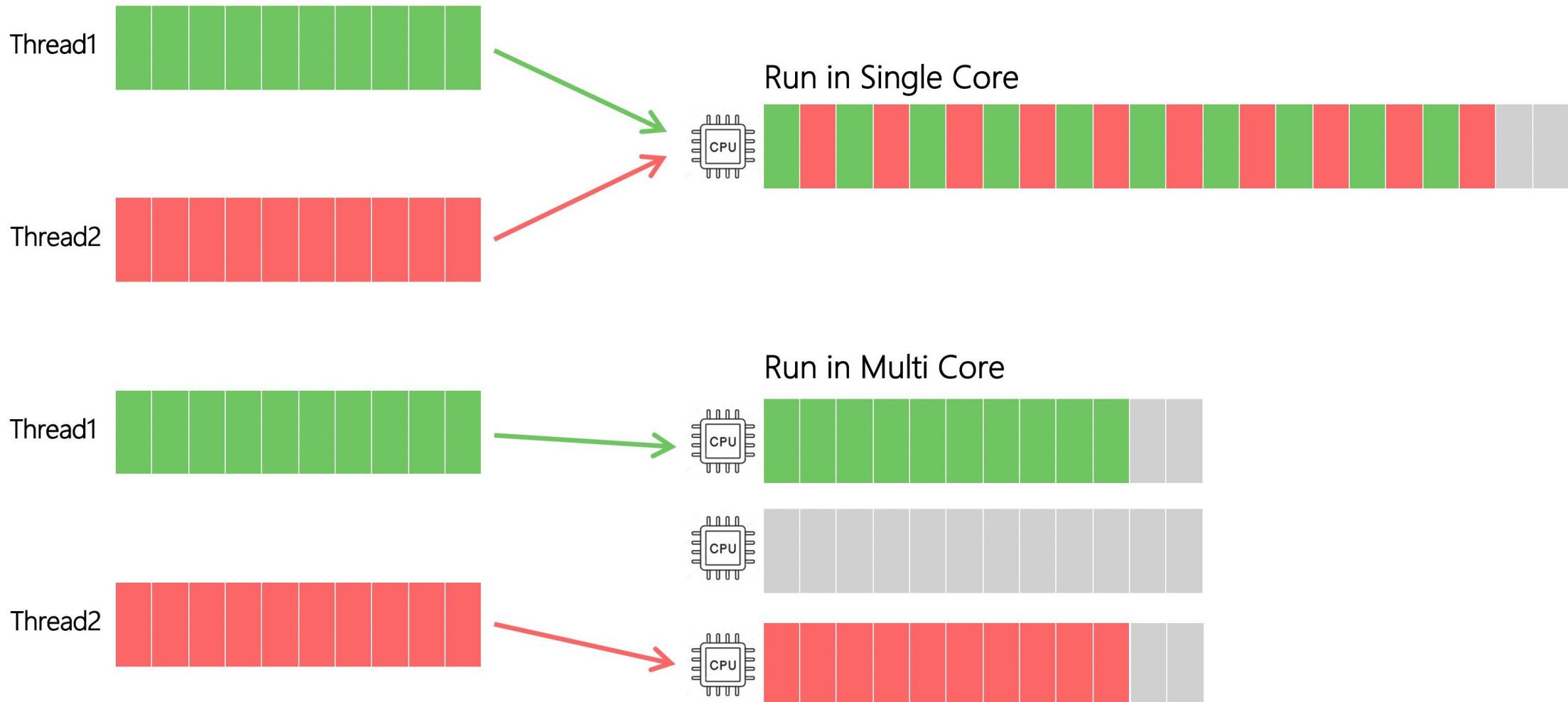
考虑通信规模

- 不同设备有 10 倍差距
- 建议: 页面渲染没有动画, 100KB
- 建议: 页面渲染有动画, 10KB

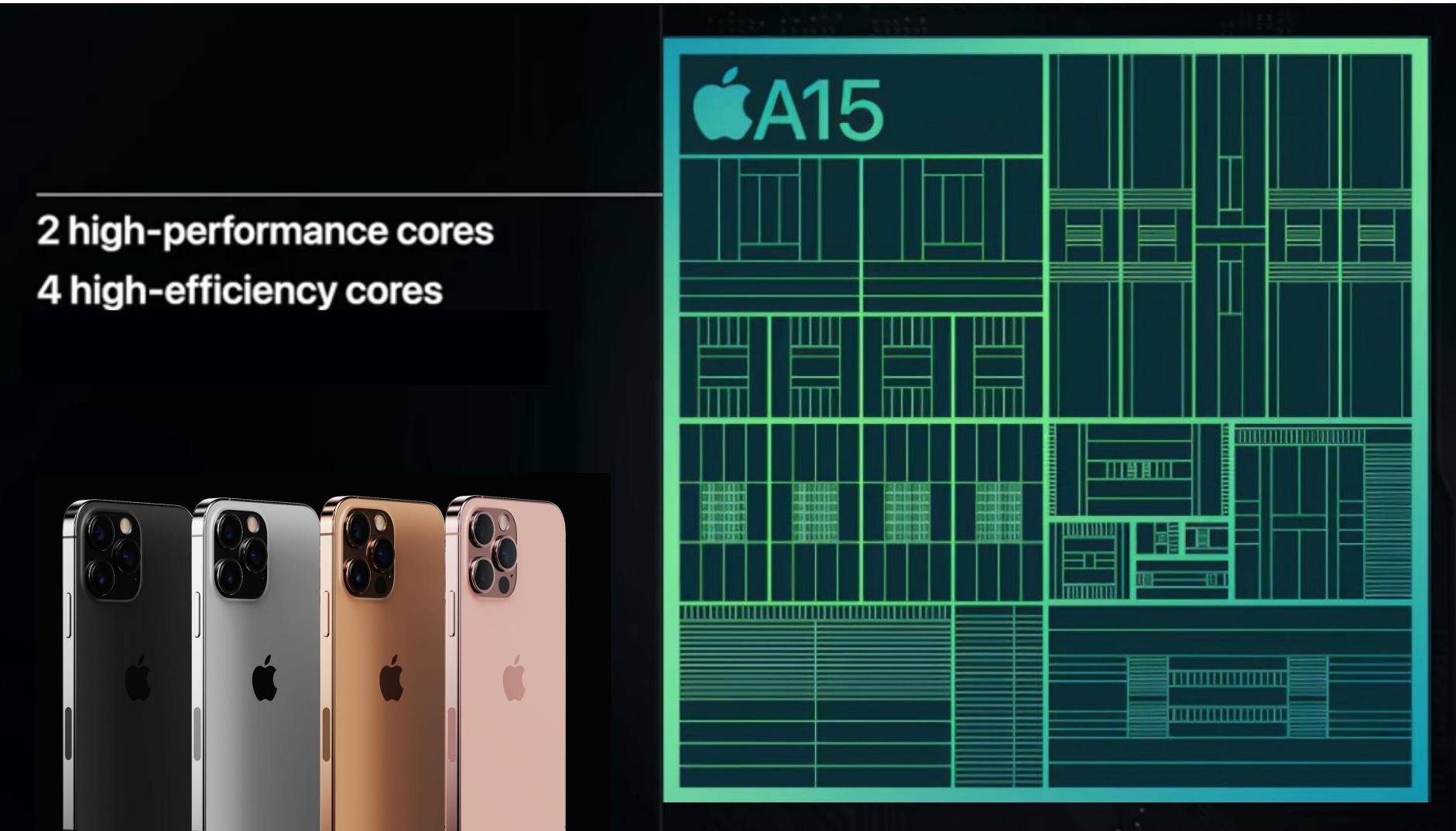
考虑 Worker 线程数



考虑 Worker 线程数



考虑 Worker 线程数

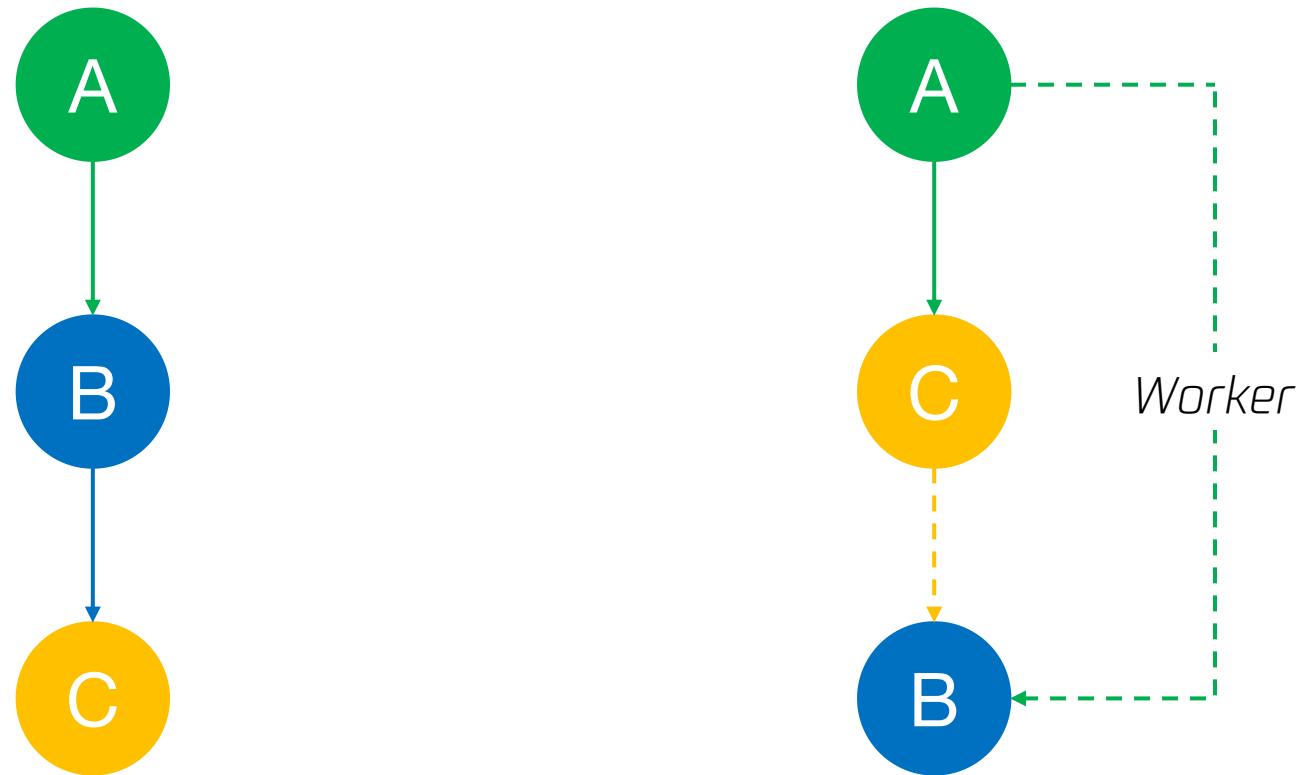


根据场景选择线程数



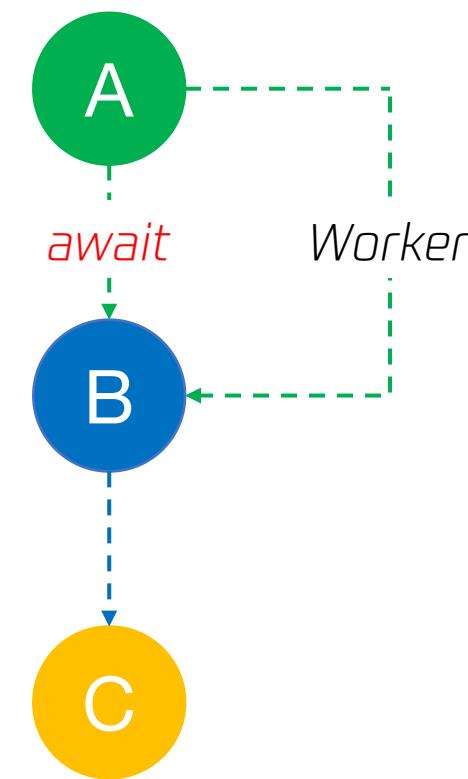
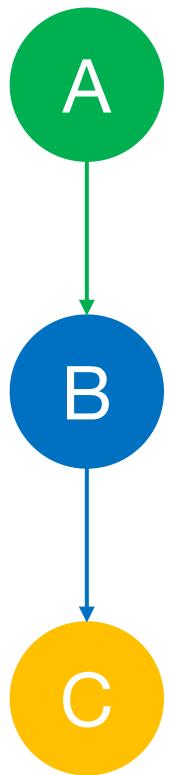
考虑异步逻辑

- *postMessage* 属于 Task



考虑异步逻辑

- Promise 异步化



考虑复杂事务

- 不就是 *postMessage* 吗 ?



- 类比下 AJAX 请求:

- 需支持不同的事务
- 需支持请求和响应的会话
- 需保障通信可靠性



通信 API 封装

考慮 Worker 线程健康

- 线程心跳检测
- 通信超时处理



多线程关注点

- 控制通信规模
- 控制线程数
- 处理异步逻辑
- 支持复杂事务
- 监控线程健康



目录

1. *Web Worker* 简介
2. *Worker* 多线程设计
3. 腾讯文档实践

腾讯文档实践

表格: 千万 PV, 百万代码的单页, 包含多人协同, 函数计算等复杂功能.

A screenshot of a Tencent Document titled "成绩统计表". The document interface includes a toolbar with icons for back, forward, search, and save, and a ribbon with font style, size, and alignment tools. The main content is a table with the following data:

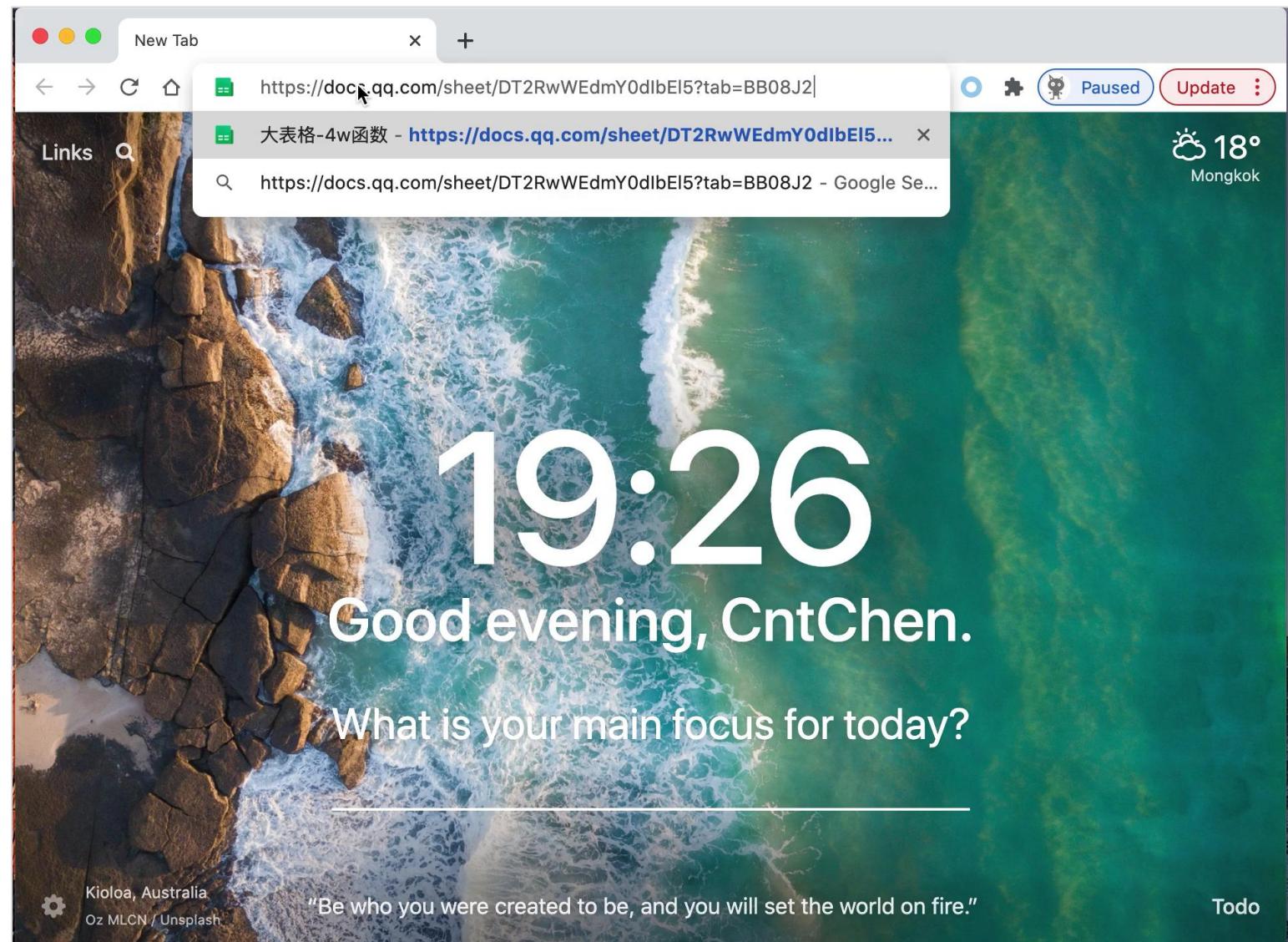
	A	B	C	D	E
1	学号	姓名	语文	数学	总分
2	1	小明	108	96	204
3	2	小红	110	90	=SUM(C3:D3)
4					

=SUM(C3:D3)

A zoomed-in view of the cell E3, which contains the formula "=SUM(C3:D3)". The cell displays the value "204".

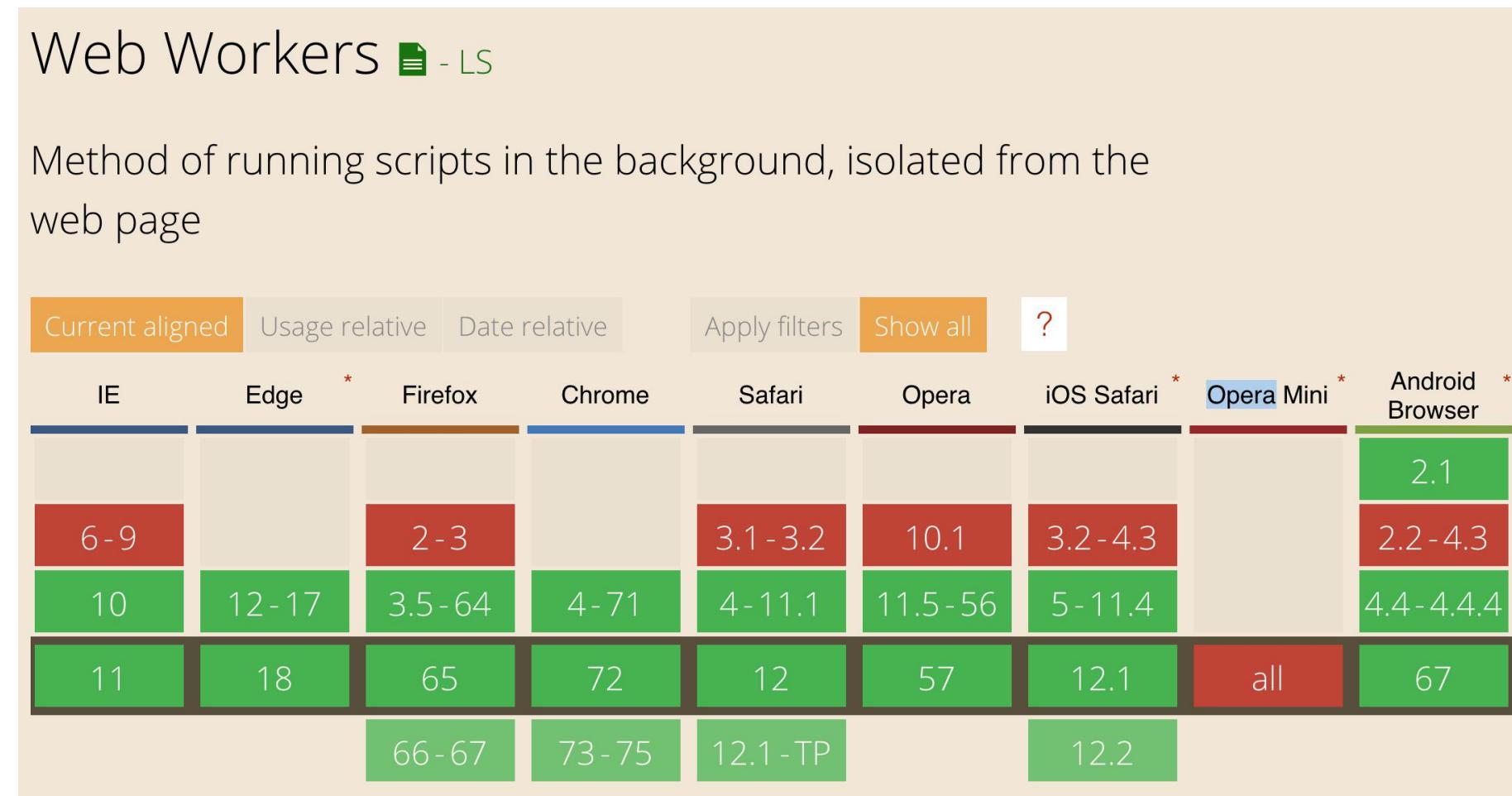
E
总分
204
200

用户反馈



技术选型 – Worker 可用性

- IE10 [2012]
- iOS5 [2012]
- Android [2013]



建立可用性评估性标

Web Worker Ability Test

By [alloy-worker](#)

- HasWorkerClass ✓
> 是否实现了 HTML 规范的 Worker Class
- CanNewWorker ✓
> 能否创建 Worker 线程
- CanPostMessage ✓
> Worker 实例有无通讯能力(脚本加载失败认为无)
- WorkerReadyDuration 183ms
> 第一条信息从发出到收到的时间间隔
- NewWorkerDuration 0ms
> 主线程创建 Worker 的同步耗时



Web Worker Ability Test

By [alloy-worker](#)

- HasWorkerClass ✓
> 是否实现了 HTML 规范的 Worker Class
- CanNewWorker ✓
> 能否创建 Worker 线程
- CanPostMessage ✗
> Worker 实例有无通讯能力(脚本加载失败认为无)
- WorkerReadyDuration -1
> 第一条信息从发出到收到的时间间隔
- NewWorkerDuration 0ms
> 主线程创建 Worker 的同步耗时

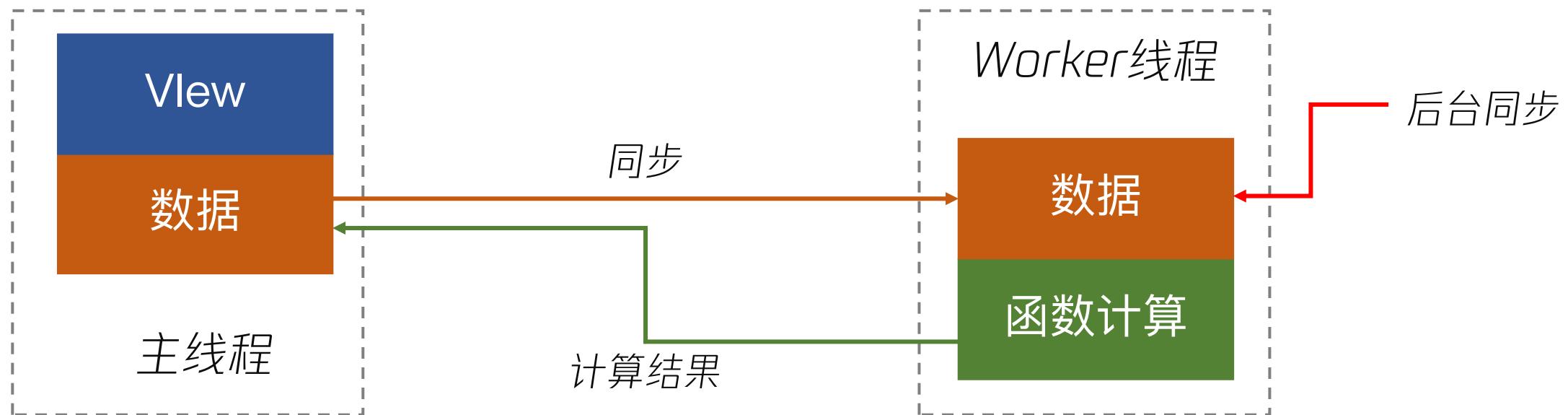
建立可用性评估性标

- 20年4月, 有 worker 能力 99.91%; 完全可用 99.58%
- 21年11月, wokrer 实例化失败, 2例

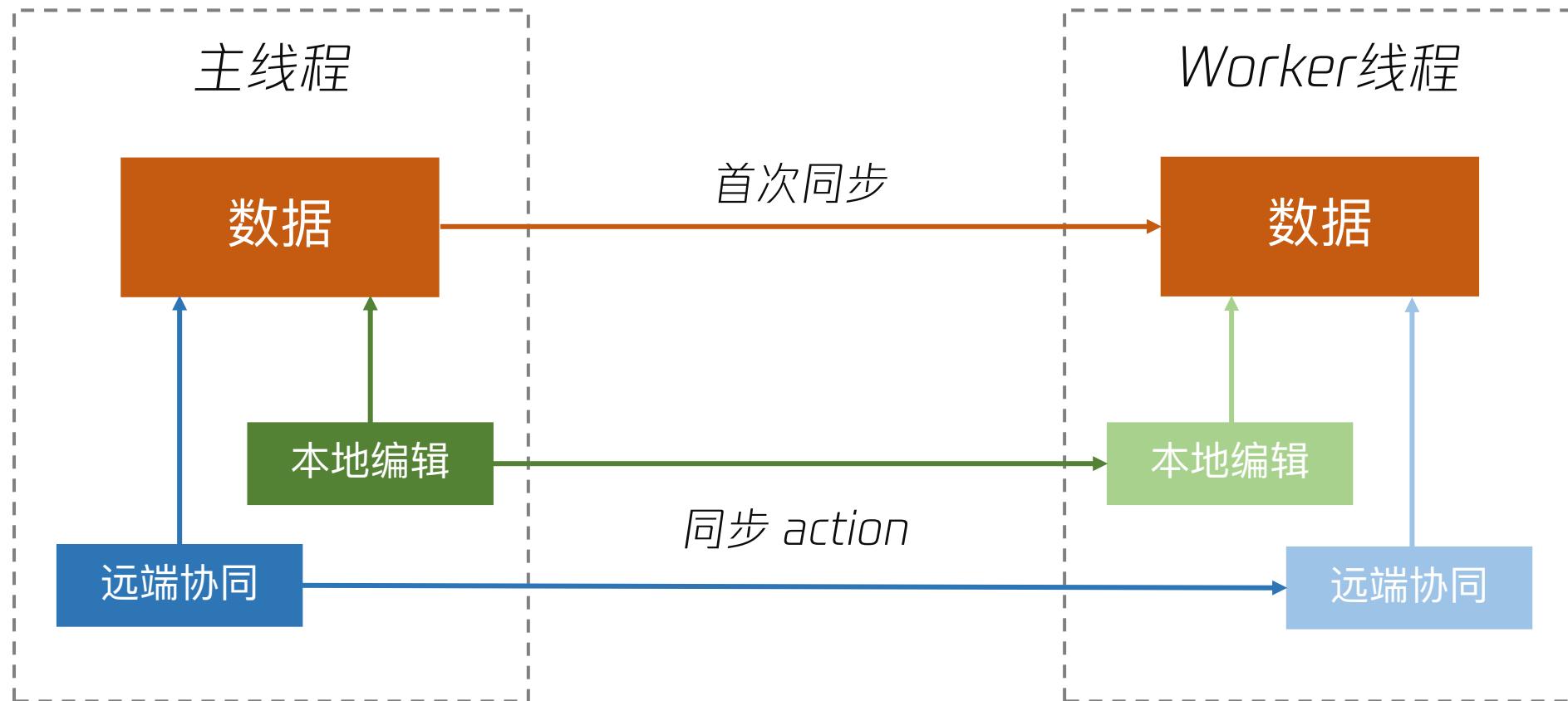
统计日期	2019年4月18号		2020年4月29号	
上报项目	统计	占比	统计	占比
有worker能力	990,120	99.96%	1,508,620	99.91%
无worker能力	355	0.04%	1,364	0.09%
worker实例化成功	988,888	99.84%	1,508,620	99.91%
worker实例化失败	1,587	0.16%	1,364	0.09%
跨线程通讯成功	982,184	99.16%	1,503,585	99.58%
跨线程通讯失败	8,291	0.84%	6,399	0.42%
首次通讯耗时	4340毫秒		5451毫秒	
上报总数	990,475		1,509,984	

技术选型 – 单 Worker 实例

- 函数计算依赖外部数据
- 优先解决卡顿

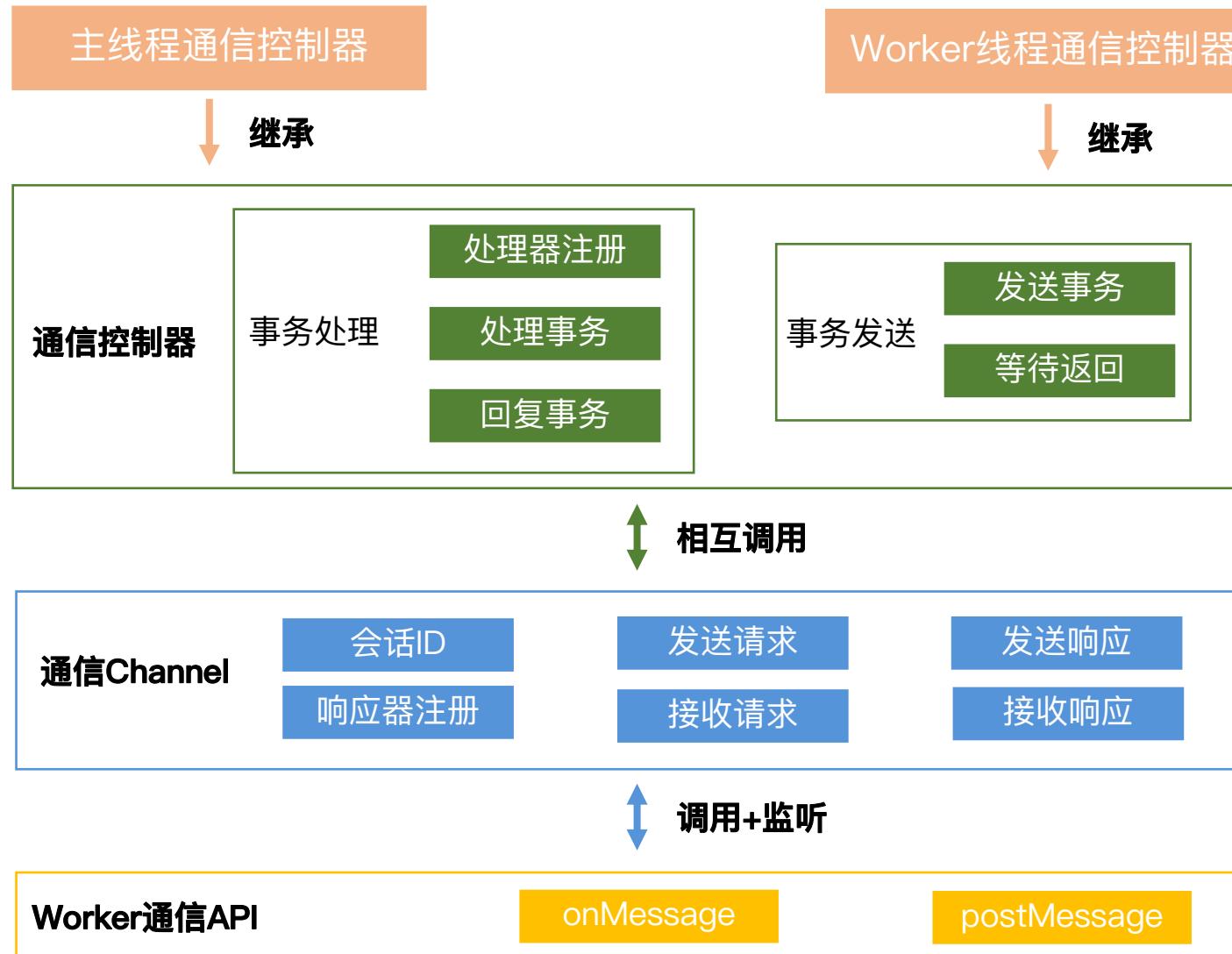


数据同步 – 同步 action



数据同步 - 分片返回

面向事务及命名空间的通信封装



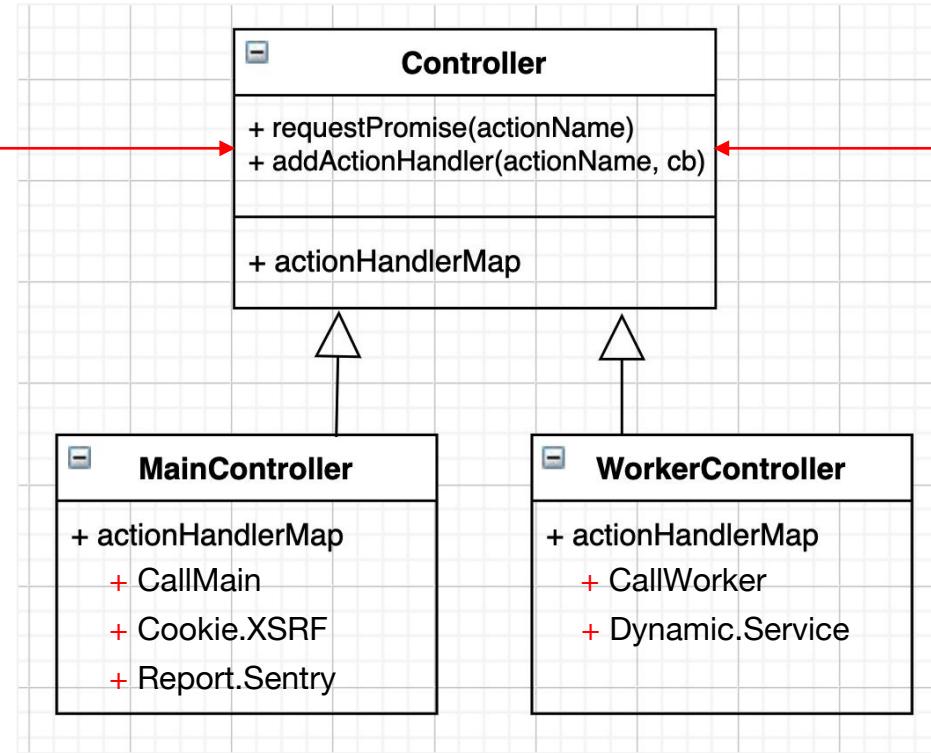
面向事务的通讯通道

事务的插件化设计

```
callWorker() {
    retrun
    requestPromise('CallWorker')
}

addActionHandler('CallMain', cb)
```

事务插件(主线程)



```
addActionHandler('CallWorker', cb)

callMain() {
    retrun requestPromise('CallMain')
}
```

事务插件(Worker)

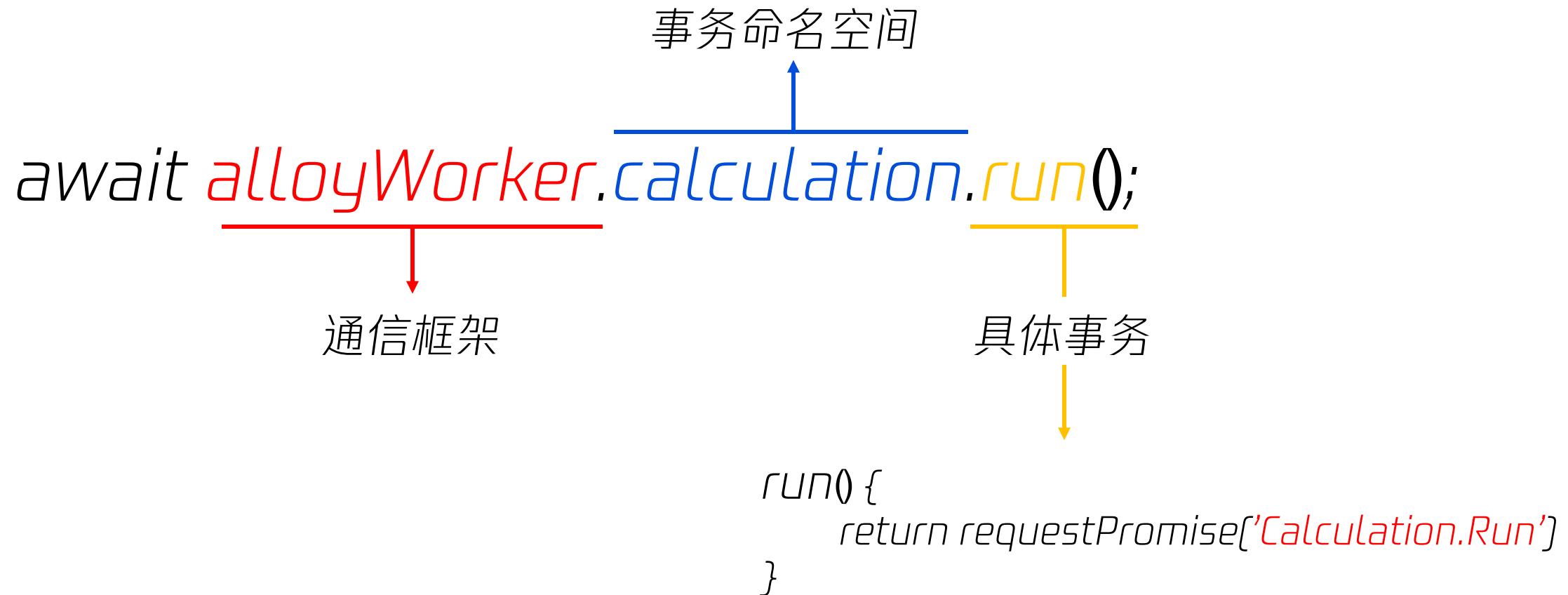
.worker/main-thread

- controller.ts
- cookie.ts**
- dynamic.ts**
- index.ts
- ...
- worker-report.ts**

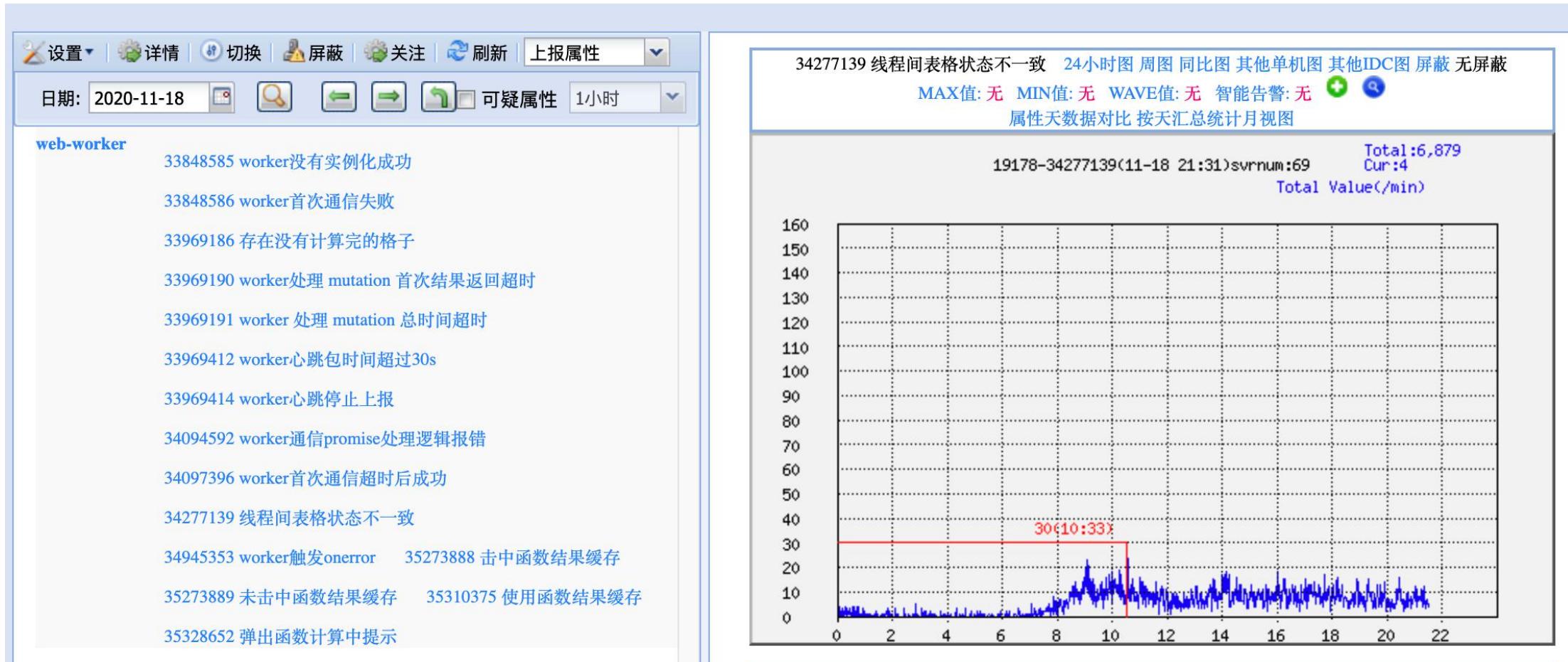
.worker/worker-thread

- controller.ts
- cookie.ts**
- dynamic.ts**
- index.ts
- ...
- worker-report.ts**

面向事务的通讯通道



监控和上报

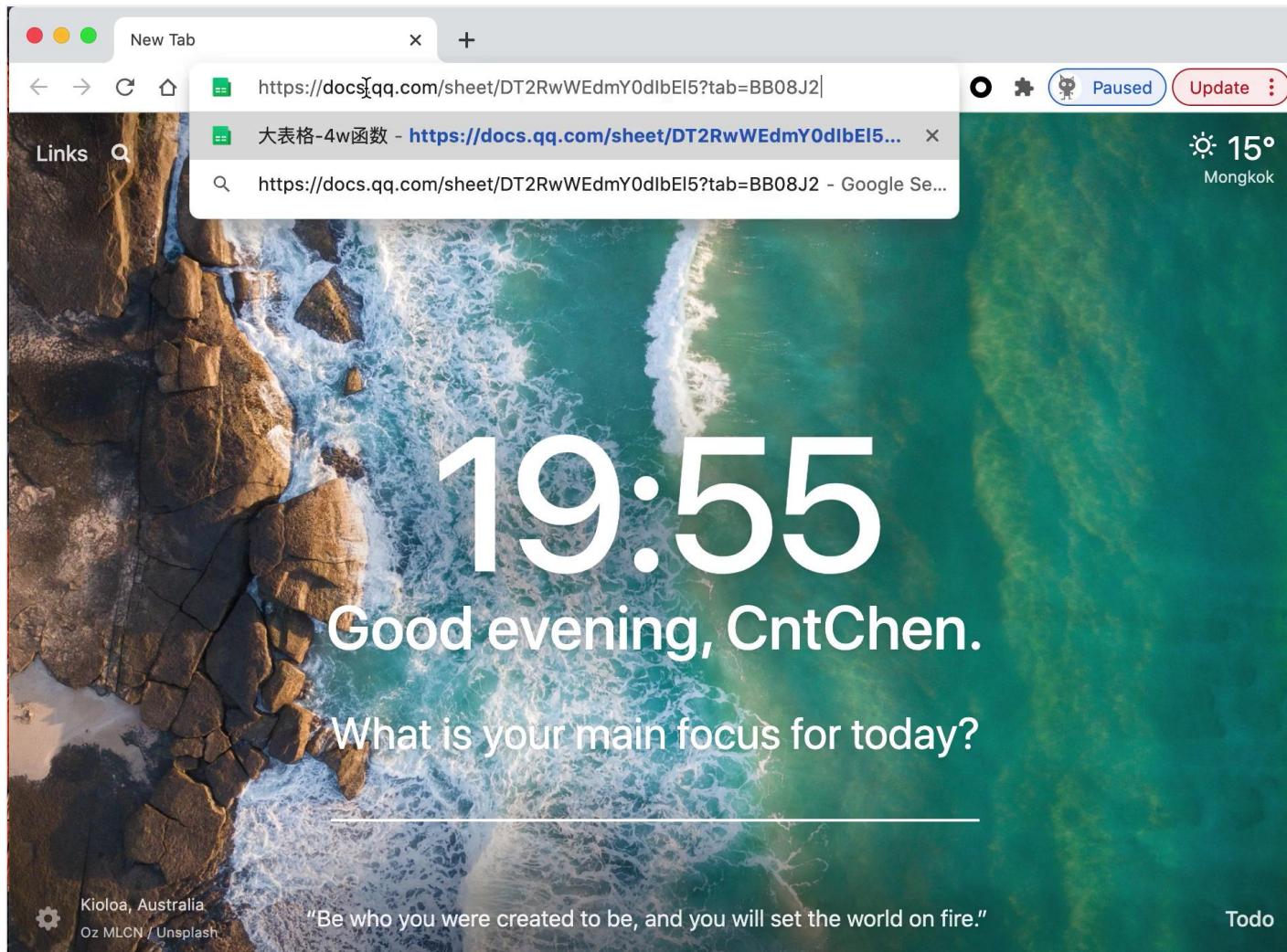


监控和上报

- 源码加载失败
- Worker 实例化失败
- 代码运行报错

<input type="checkbox"/>	SyntaxError	/sheets/static/js/public-worker-968a9c3e16.js i...
	Unexpected end of script	
	TD-SHEET-23	⌚ 38 分钟前 – 4 个月前 javascript
<input type="checkbox"/>	SyntaxError	/sheets/static/js/public-worker-968a9c3e16.js ...
	Unexpected EOF	
	TD-SHEET-A5	⌚ 1 小时前 – 4 个月前 javascript
<input type="checkbox"/>	ReferenceError	/sheets/static/js/public-worker-968a9c3e1...
	console is not defined	
	TD-SHEET-57	⌚ 2 小时前 – 4 个月前 javascript
<input type="checkbox"/>	SyntaxError	/sheets/static/js/public-worker-968a9c3e16.js ...
	\u can only be followed by a Unicode character sequence	
	TD-SHEET-JW	⌚ 4 小时前 – 4 个月前 javascript

卡顿优化



实践小结

- 建立可用性评价指标
- 面向事务及命名空间的通信封装
- 监控和上报

内存占用

- 1680个函数计算, **12.3M**

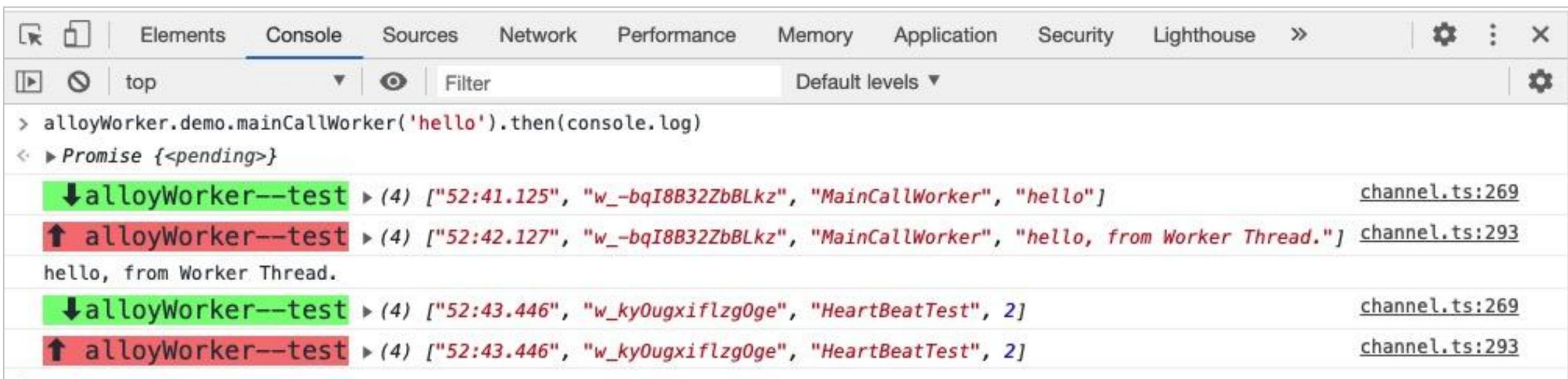
The screenshot shows the Chrome DevTools Performance tab open. On the left, there's a preview of a spreadsheet with data in columns A, B, and C. Column A contains the formula =SUM(B:B,C:C,D:D,E:E,F:F,G:G,H:H,I:I). The first few rows show values like 1440, 0, 0; 1440, 1, 1; etc. On the right, the Performance panel is displayed with the following details:

- Select profiling type:**
 - Heap snapshot**
Heap snapshot profiles show memory distribution among objects.
 - Allocation instrumentation on timeline**
Allocation timelines show instrumented JavaScript memory allocations over time. Choose profile type to isolate memory leaks.
 - Record allocation stacks (extra performance overhead)
 - Allocation sampling**
Record memory allocations using sampling method. This does not record allocation stacks.
- Select JavaScript VM instance:**

56.9 MB	↑283 KB/s	docs.qq.com
2.1 MB	↓1.3 KB/s	wlwk.min.3.4.js
12.3 MB	↓309 KB/s	defaultWorker

Alloy-worker

- 跨线程请求和响应的数据流调试.



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The log output is as follows:

```
> alloyWorker.demo.mainCallWorker('hello').then(console.log)
< ▶ Promise {<pending>}
    ↓alloyWorker--test ▶ (4) ["52:41.125", "w_-bqI8B32ZbBLkz", "MainCallWorker", "hello"] channel.ts:269
    ↑ alloyWorker--test ▶ (4) ["52:42.127", "w_-bqI8B32ZbBLkz", "MainCallWorker", "hello, from Worker Thread."] channel.ts:293
    hello, from Worker Thread.
    ↓alloyWorker--test ▶ (4) ["52:43.446", "w_ky0ugxiflzb0ge", "HeartBeatTest", 2] channel.ts:269
    ↑ alloyWorker--test ▶ (4) ["52:43.446", "w_ky0ugxiflzb0ge", "HeartBeatTest", 2] channel.ts:293
```

Alloy-worker

开源项目	简介	底层API封装	事务声明	可用性监控	易拓展性	网址
Worker-loader	Webpack官方，基本源码打包能力					https://github.com/webpack-contrib/worker-loader
Promise-worker	封装基本API为Promise化通信	✓				https://github.com/nolanlawson/promise-worker
Comlink	Chrome团队开源，worker打包plugin	✓	跨线程同名函数调用			https://github.com/GoogleChromeLabs/comlink
Alloy-worker	面向事务, 高可用的 worker 通信框架	通信控制器	同名函数 TS 声明	完整监控指标 全周期错误监控	命名空间	https://github.com/AlloyTeam/alloy-worker

实践小结

《Web Worker 文献综述》



目录

1. *Web Worker* 简介
2. *Worker* 多线程设计
3. 腾讯文档实践

- 减少卡顿, 提升性能
- 成本 vs 收益
- 腾讯文档 & *Alloy-worker*

朝花夕拾 芳香依旧

陈韩杰 [@CntChen](#)