



100% Performance Improving, 90% Cost Saving

— How Tencent Scales to Millions with Serverless Next.js

Tinafang
Senior Product Manager
@Tencent Cloud

Who Am I?



Serverless Advocate

Product Manager @  Tencent Cloud

github.com/tinafangkunding

[@tinafangkd](https://twitter.com/tinafangkd)

tinafang.com

Agenda

01

**Why Using
Next.js**

02

**Architecture of
Tencent SSR**

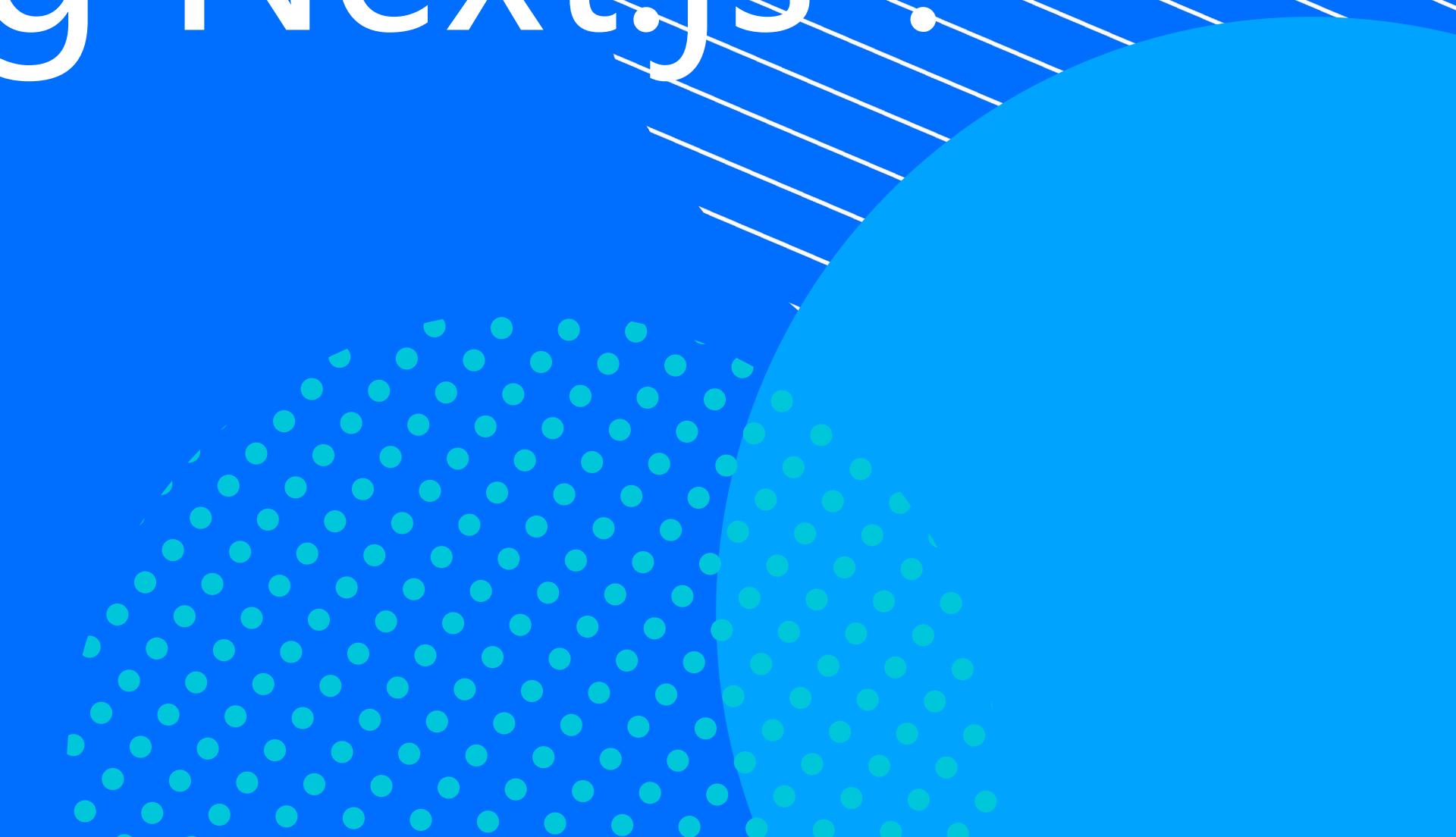
03

**Combine Next.js
with Serverless**

04

Summary

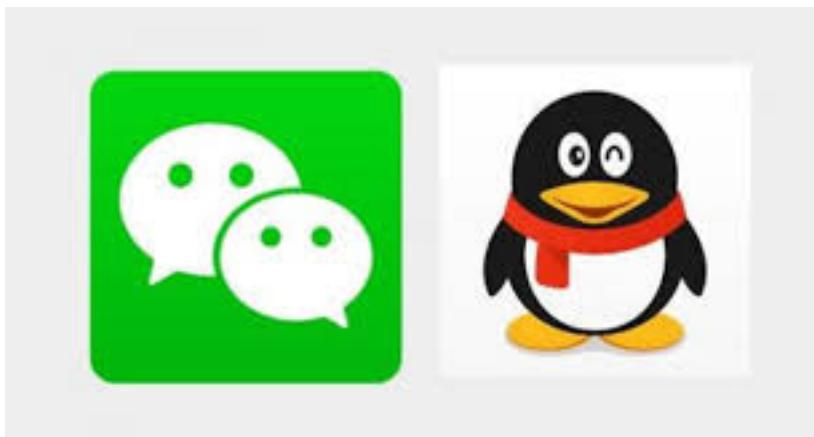
1. Why Using Next.js?



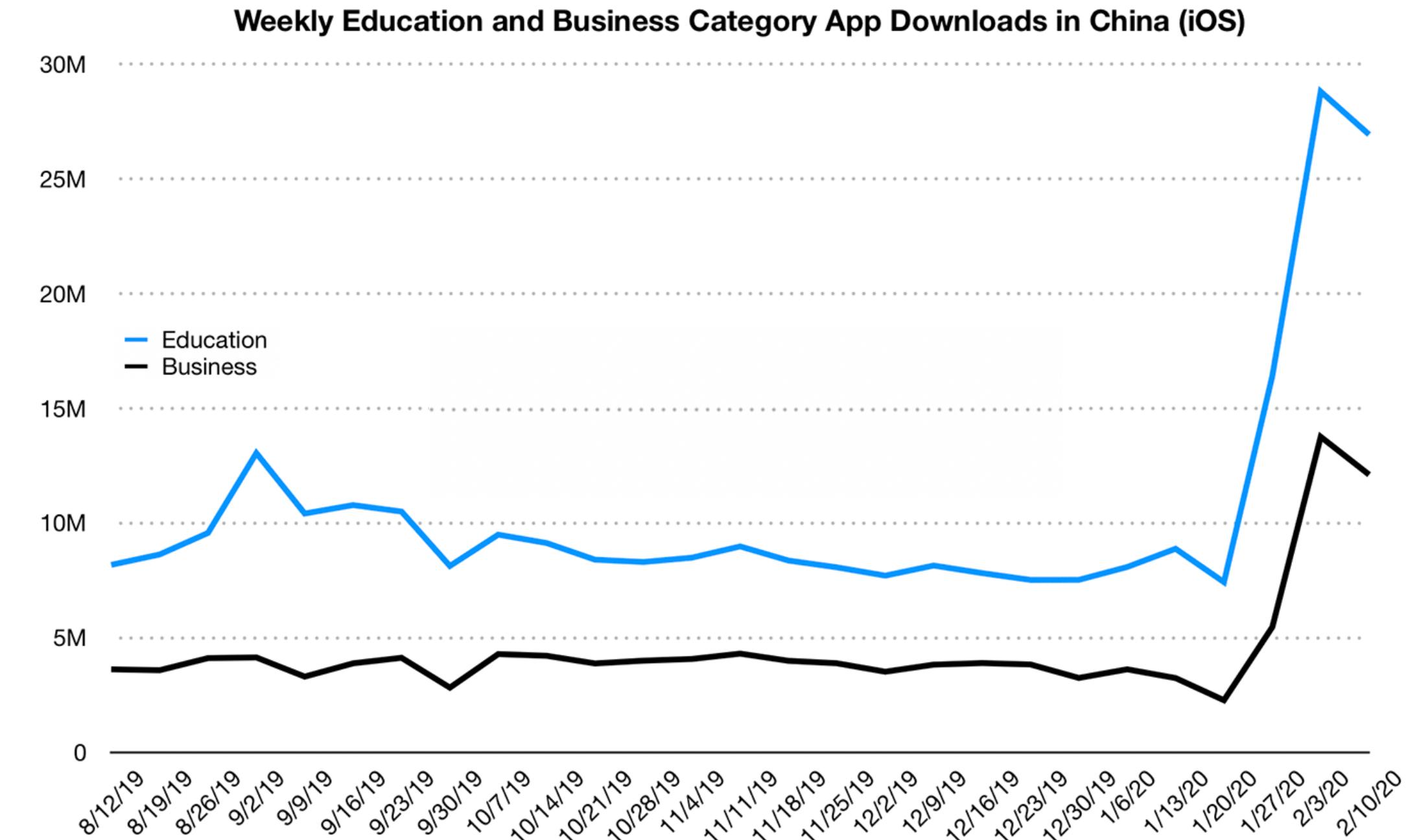
About Tencent

Tencent 腾讯

- One of the world's top technology companies
- Services include social network, music, e-commerce, mobile games, etc.
- Known as instant messengers like Tencent QQ and WeChat



Story Begins From...

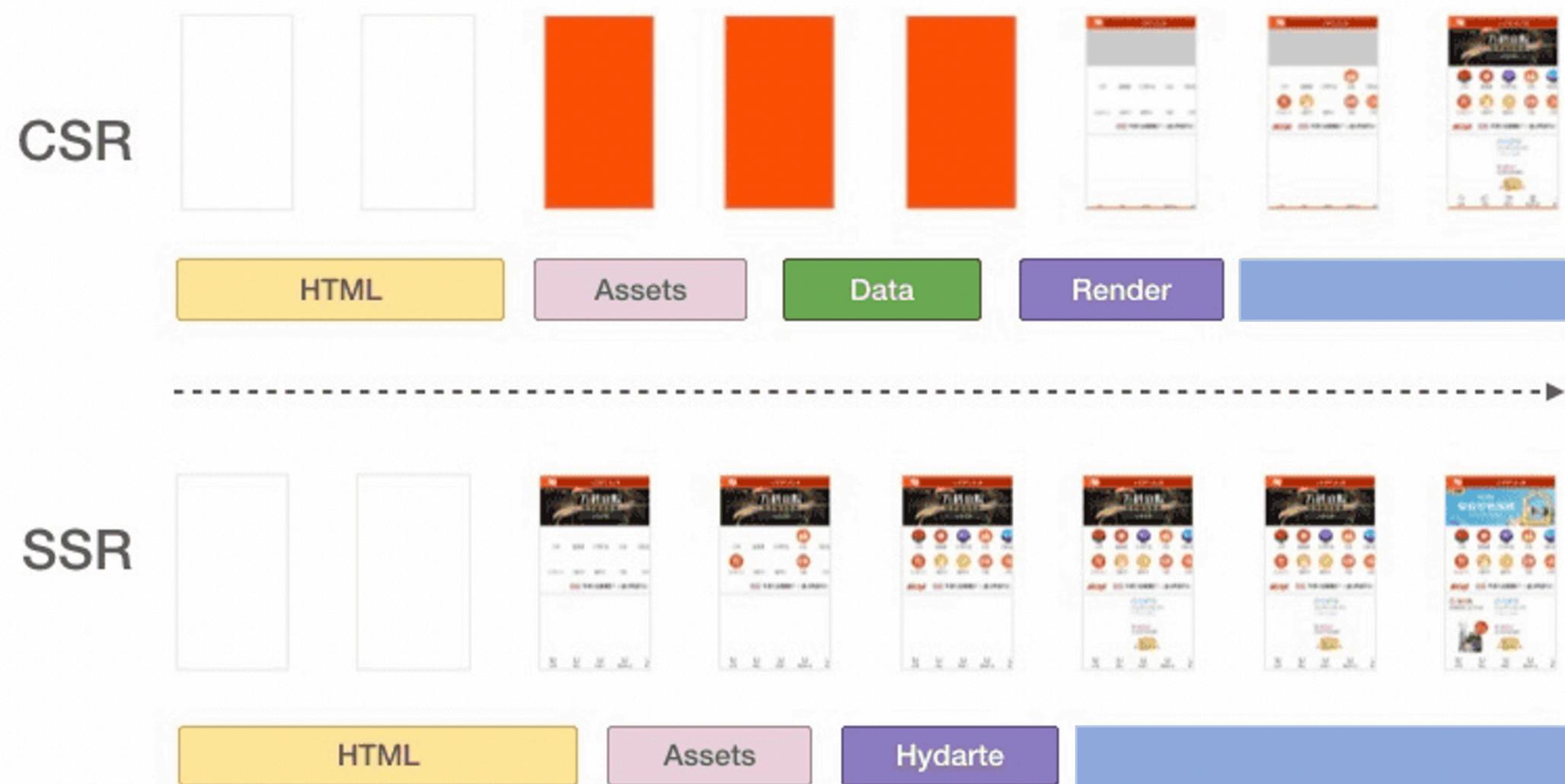


可以增加一下当前团队的业务量，app 使用等

<https://techcrunch.com/2020/02/23/china-roundup-tech-companies-coronavirus/>

Why Using SSR?

1. Performance (FCP or FMP)

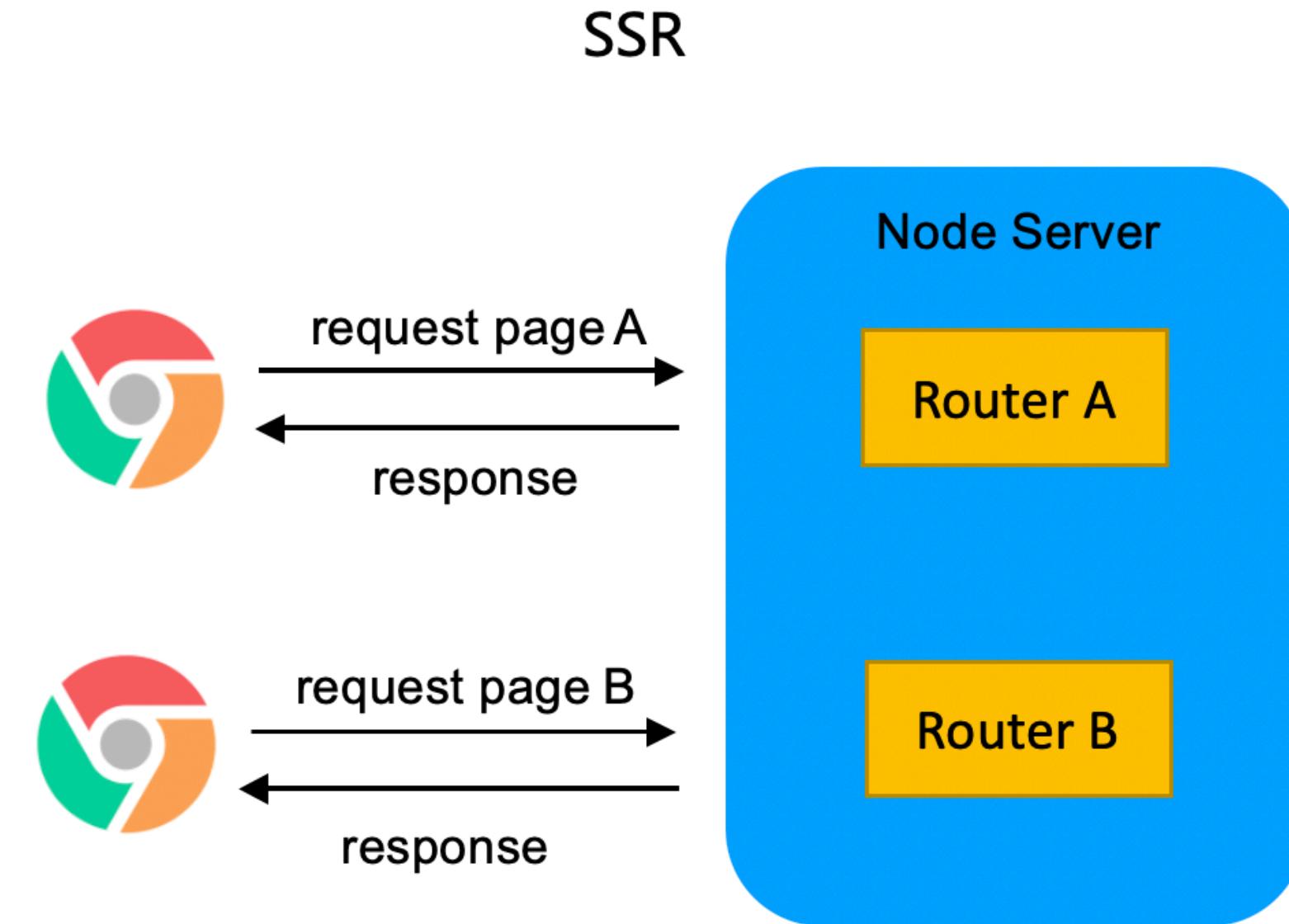


2. SEO



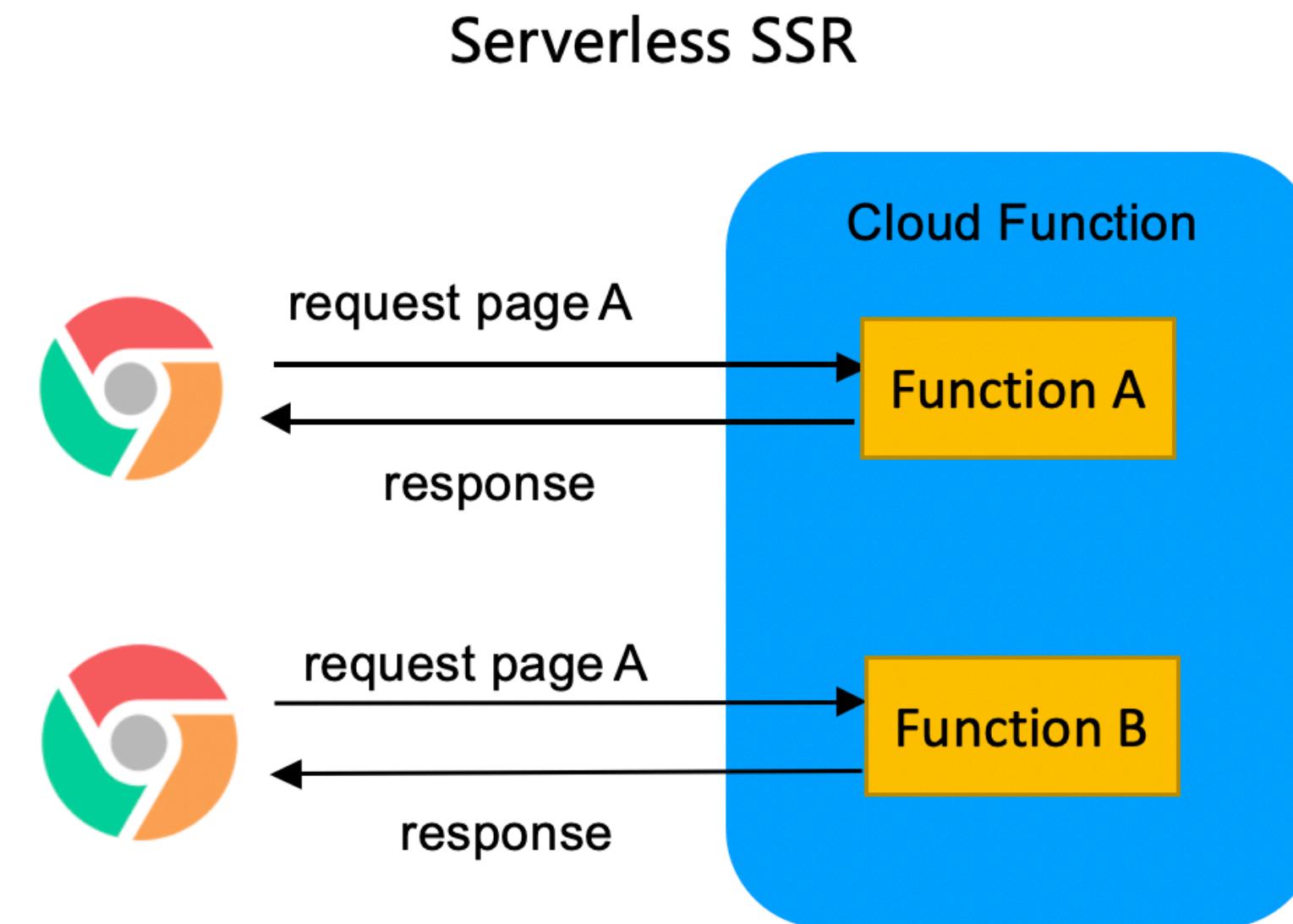
Challenges in SSR

- Maintain Node Server
- Auto Scaling
- Cost of Servers
- Nginx Configurations
- ...



Serverless + SSR

- No need to maintain
- Auto Scaling
- Pay as you Go
- ...



Next.js

 Unstar

54,396

- Static Site Generation (SSG)
- Server Side Rendering (SSR)
- CDN+Static Storage
- Dev Toolkit/Platform
- Open Source
- ...

		Server	Browser			
						
		Server Rendering	“Static SSR”	SSR with (Re)hydration	CSR with Prerendering	Full CSR
Overview:		An application where input is navigation requests and the output is HTML in response to them.	Built as a Single Page App, but all pages prerendered to static HTML as a build step, and the JS is removed .	Built as a Single Page App. The server prerenders pages, but the full app is also booted on the client.	A Single Page App, where the initial shell/skeleton is prerendered to static HTML at build time.	A Single Page App. All logic, rendering and booting is done on the client. HTML is essentially just script & style tags.
Authoring:		Entirely server-side (request-response, HTML)	Built as if client-side (components, DOM*, fetch)	Built as client-side	Client-side	Client-side
Rendering:		Dynamic HTML	Static HTML	Dynamic HTML and JS/DOM	Partial static HTML, then JS/DOM	Entirely JS/DOM
Server role:		Controls all aspects. (thin client)	Delivers static HTML	Renders pages (navigation requests)	Delivers static HTML	Delivers static HTML
Pros:		 TTI = FCP  Fully streaming	 Fast TTFB  TTI = FCP  Fully streaming	 Flexible	 Flexible  Fast TTFB	 Flexible  Fast TTFB
Cons:		 Slow TTFB  Inflexible	 Inflexible  Leads to hydration	 Slow TTFB  TTI >>> FCP  Usually buffered	 TTI > FCP  Limited streaming	 TTI >>> FCP  No streaming
Scales via:		Infra size / cost	build/deploy size	Infra size & JS size	JS size	JS size
Examples:		Gmail HTML, Hacker News	Docusaurus, Netflix*	Next.js, Razzle, etc	Gatsby, Vuepress, etc	Most apps

<https://github.com/vercel/next.js>

2. Architecture of Tencent SSR



Architecture in Tencent Next.js App



Next.js Functionalities

LRU Cache

```
async function cacheRender(req, res) {
  const key = getCacheKey(req);
  // reder /list to /
  // const reqPath = req.path === '/list' ? '/' : req.path;
  const reqPath = req.path;
  if (Cache.has(key)) {
    res.setHeader('X-Cache', 'HIT');
    res.send(Cache.get(key));
    return;
  }

  try {
    const html = await app.renderToHTML(req, res, reqPath, req.query);
    if (res.statusCode !== 200) {
      res.send(html);
      return;
    } else {
      Cache.set(key, html);
      res.setHeader('X-Cache', 'MISS');
      res.send(html);
    }
  } catch (err) {
    res.statusCode = 500;
    app.renderError(err, req, res, reqPath, req.query);
  }
}
```

Next.js Functionalities

Custom Server: Express/Koa

```
const express = require('express')
const next = require('next')

async function createServer() {
  const app = next({ dev: false })
  const handle = app.getRequestHandler()

  await app.prepare()
  const server = express()

  server.all('*', (req, res) => {
    return handle(req, res)
  })

  // define binary type for response
  // if includes, will return base64 encoded, very useful for images
  server.binaryTypes = ['*/*']
  return server
}

// comment out `listen`
// exports app server

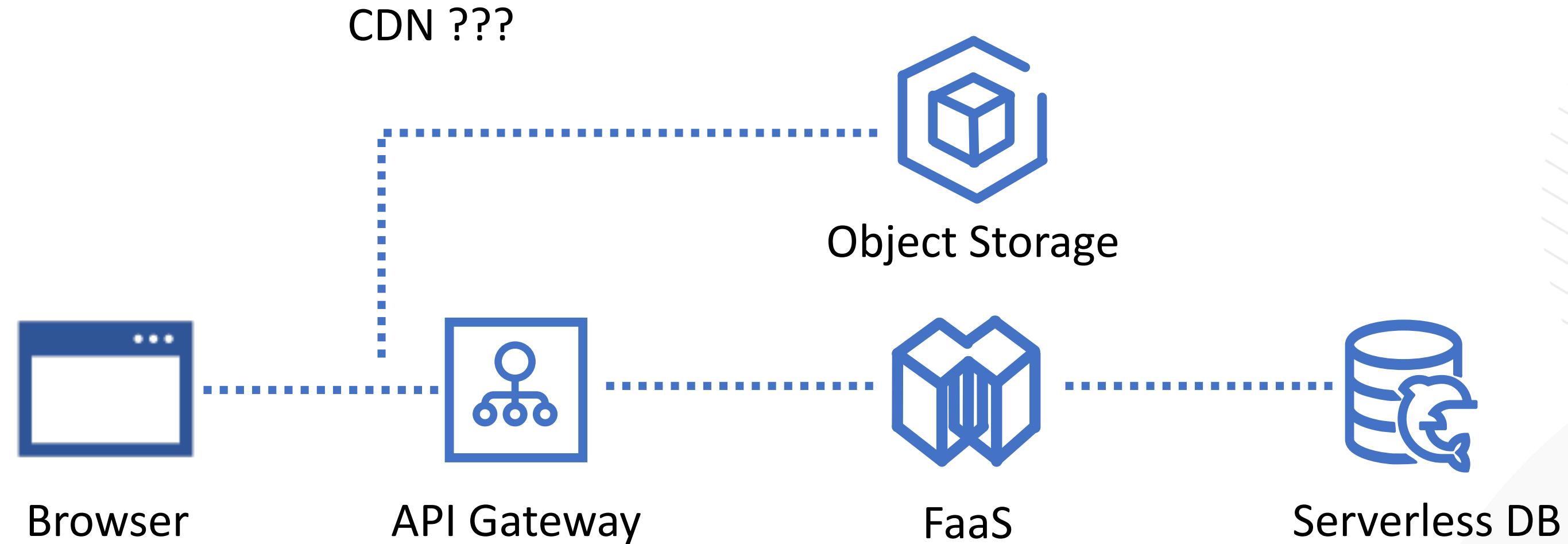
// createServer().listen(3000);

module.exports = createServer
```

3. Next.js + Serverless



Services we use in Next.js Applications



Performance Improving 100%

Using Object Storage to Separate Static Assets, which Includes:

- `next/static` folder
- Files inside `public` folder

Update `next.config.js`

```
const isProd = process.env.NODE_ENV === "production";
const STATIC_URL =
  "https://serverless-nextjs-xxx.cos.ap-guangzhou.myqcloud.com";
module.exports = {
  env: {
    // for local development
    STATIC_URL: isProd ? STATIC_URL : "http://localhost:3000",
  },
  assetPrefix: isProd ? STATIC_URL : "",
};
```

Update Path in `public` folder

```
<!-- before -->
<head>
  <title>Create Next App</title>
  <link rel="icon" href="/favicon.ico" />
</head>

<!-- after -->
<head>
  <title>Create Next App</title>
  <link rel="icon" href={`${process.env.STATIC_URL}/favicon.ico`} />
</head>
```

Performance Improving 100%

CDN Acceleration
with World-Wide Nodes

Update `next.config.js`

```
const isProd = process.env.NODE_ENV === "production";
const STATIC_URL = "https://static.test.yuga.chat";
module.exports = {
  env: {
    STATIC_URL: isProd ? STATIC_URL : "http://localhost:3000",
  },
  assetPrefix: isProd ? STATIC_URL : "",
};
```



可以增加一下当前团队的性能提升数据

Cost Saving 90%

Using “Serverless” Services instead of prevision resources

- SCF – Cloud Function
- API Gateway
- COS
- CDN
- ...

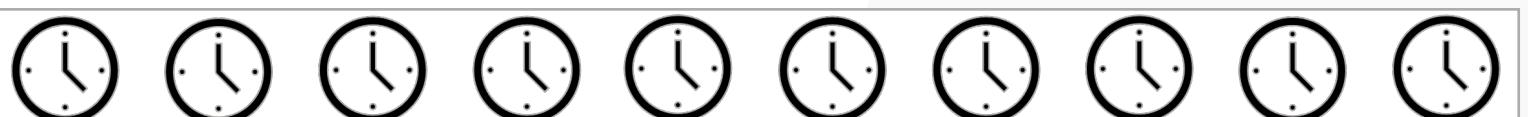
Labor Costs



Resource Cost



Time to Market



可以增加一下当前团队的成本降低数据

CI/CD & Monitoring

CI/CD Integration

- Orange CI (Tencent Internal CI)
- Github Actions
- Coding CI/CD
- ...

The collage includes:

- A GitHub Actions icon featuring a blue play button and a network graph.
- A cartoon character of a man in a tuxedo holding a coffee cup.
- A Coding Cloud Development logo with a monkey head and the word "CODING".
- A screenshot of a web browser showing a merge request for "orange-ci / nodejs-demo · MergeRequests · !20". The page displays build logs, test results, and deployment logs.

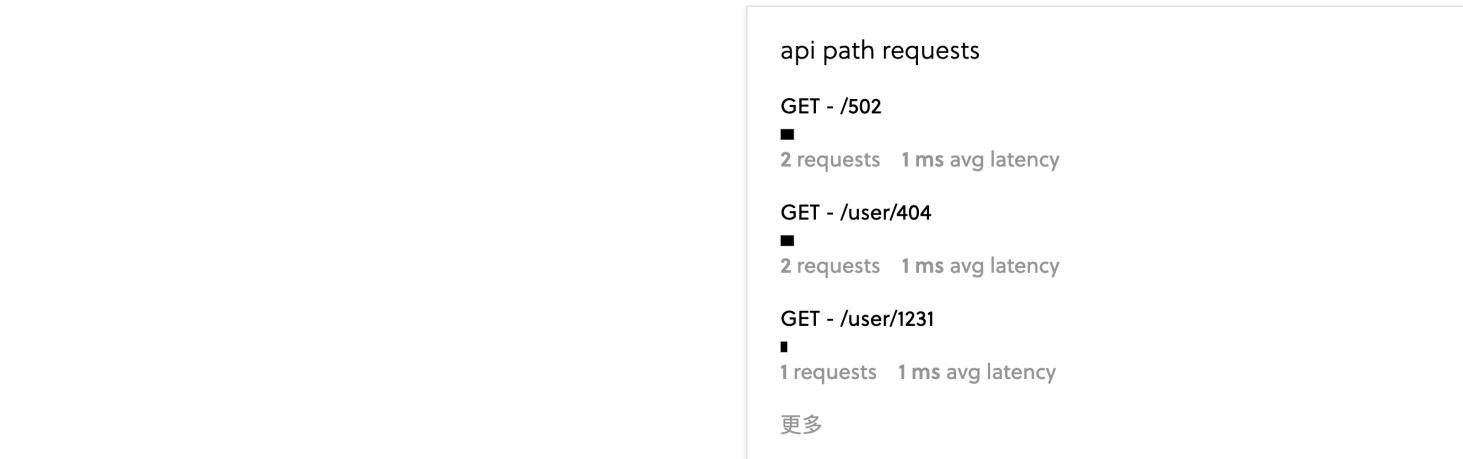
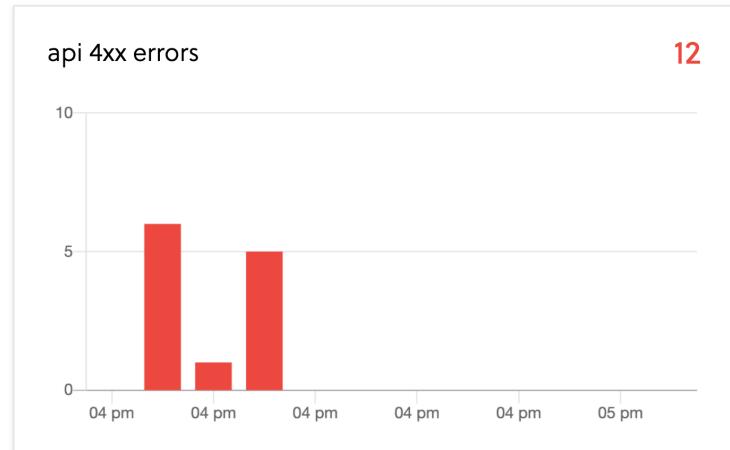
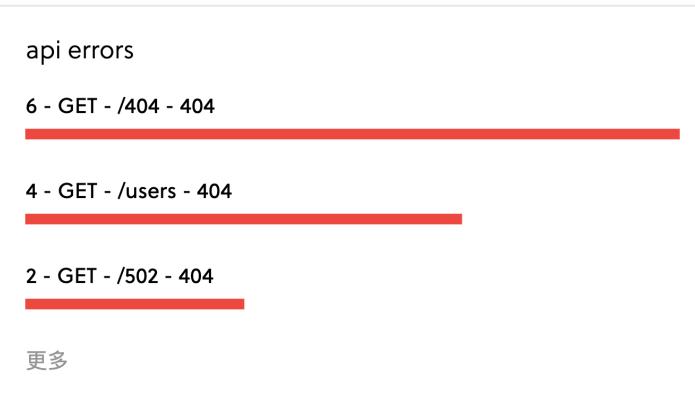
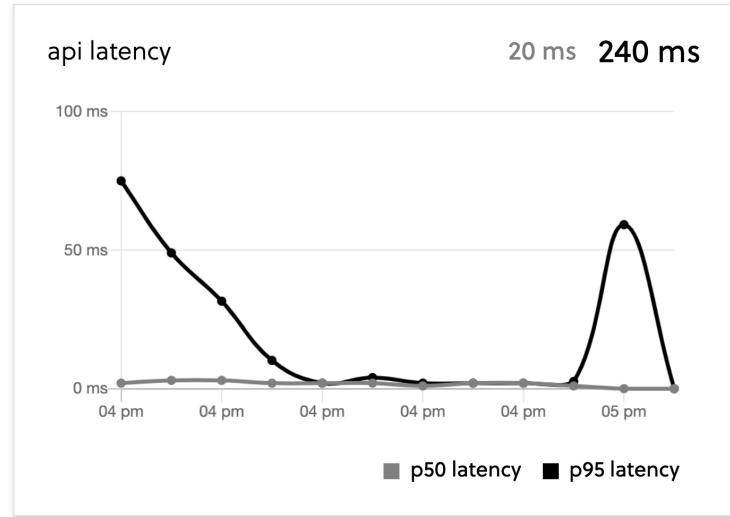
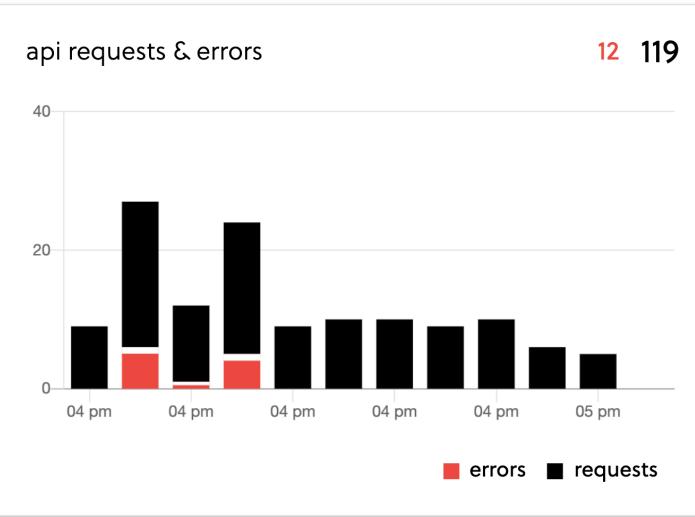
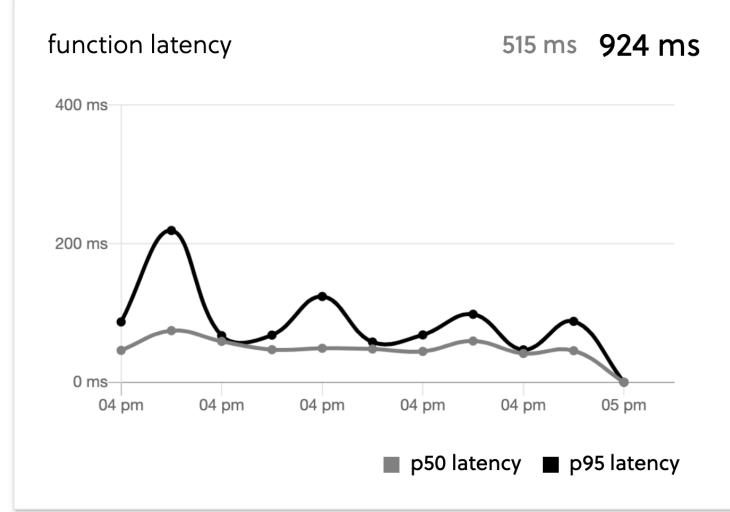
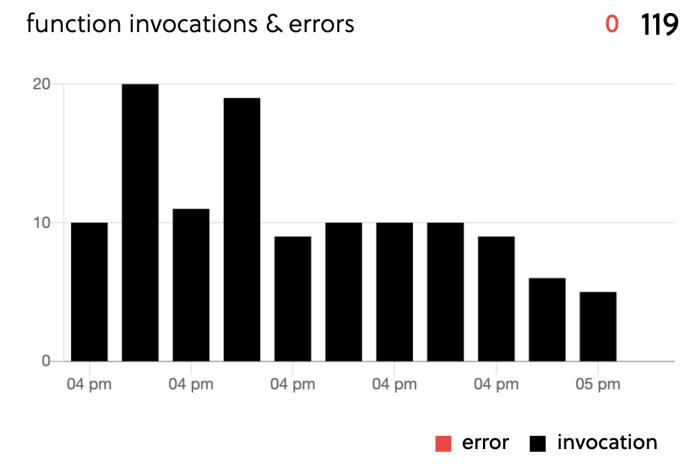
可以增加一下当前团队在 CICD 的最佳实践等

CI/CD & Monitoring

Application Monitoring for Next.js

- API Latency
- 4XX 5XX Status Codes
- API Requests & Errors
- API Path Requests & Errors
- ...

Data reported by an agent : ServerlessSDK



4. Recap & Summary



Recap & Summary

Next.js Functionalities

- ✓ Custom Express Server
- ✓ LRU Render Cache

Serverless Combination

- ✓ Using FaaS and API Gateway to deploy Next.js
- ✓ Using COS to Separate Dynamic/Static Assets
- ✓ CDN Acceleration
- ✓ Using Layer to Deploy node_modules

Operations

- ✓ Application level Monitoring
- ✓ CI/CD Combination

In the Future

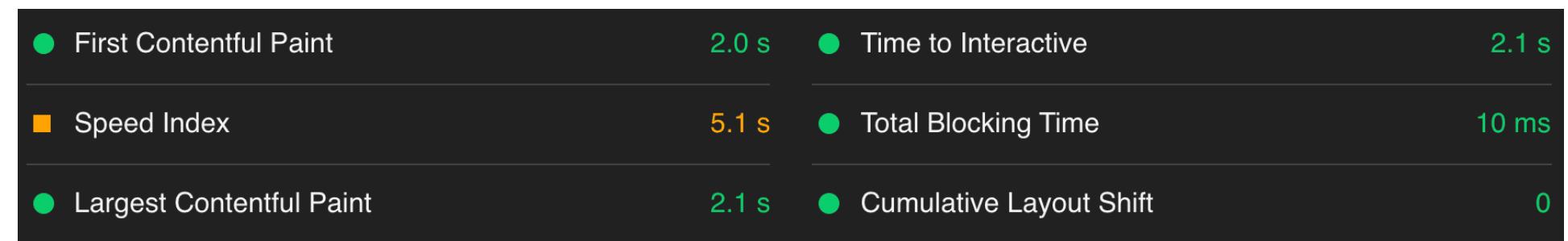
- ✓ Vendor Agnostic(AWS & Azure)
- ✓ 待补充
- ✓ 待补充

Demo of Serverless Next.js

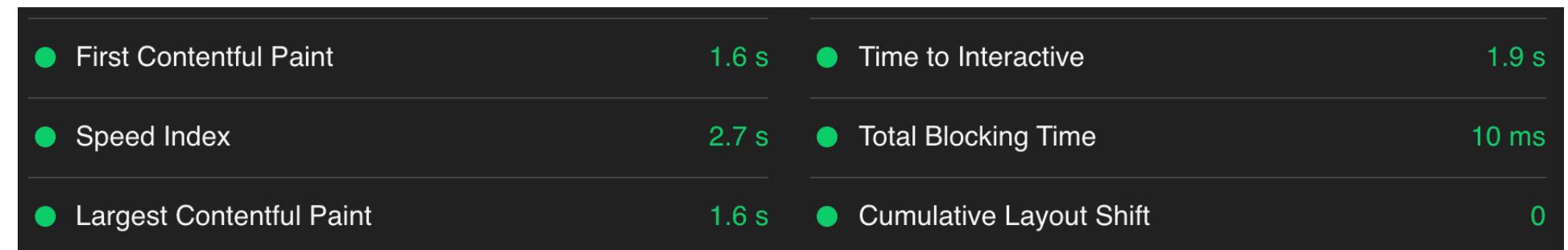
<https://github.com/serverless-plus/serverless-v2ex>

Lighthouse Performance

Before



After



The screenshot shows the homepage of v2ex.yuga.chat. The header includes navigation icons and the URL. Below the header is a blue navigation bar with the text "首页 - Serverless V2EX". The main content area displays a list of posts from various users. Each post includes a small profile picture, the title, the user's name, the posting time, and the number of replies.

- 罗技 MX Master 3 对手小的人来说真是噩梦
计算机 • 166953天前 • 0
- 请问有能实现这种聚合效果的 chrome/edge 插件吗?
Chrome • 166953天前 • 0
- access_log 日志中存在大量 AHC/1.0
程序员 • 166953天前 • 0
- 各大厂商路由器系统更新支持时间?
问与答 • 166953天前 • 0
- 腾讯云 11·11 续费活动低至 2 折操作
云计算 • 166953天前 • 0
- [日经求推荐手机] 原生 (至少流畅使用 google 服务) | 双卡 | 128G 或以上
Android • 166953天前 • 3
- 哪里有 Golang 的 websocket 框架对比
Go • 166953天前 • 1
- iperf 内网测速不理想, 求支招
问与答 • 166953天前 • 0
- 良心云又开始瞎搞活动了, 双 11 提前上线, s4 的 2-4-3 配置 3 年打完折
500 多块, 企业用户特殊活动数据库高防证书等产品白送
云计算 • 166953天前 • 0
- 请教大佬, Win10 有没有办法快速的把一个窗口移动到下一个虚拟桌面
上?
Windows • 166953天前 • 2

THANK YOU!

Q&A

