探索 Serverless 中的前端开发模式

蒋航

阿里云 前端工程师

GTLC
GLOBAL
TECH LEADERSHIP
CONFERENCE

全球技术领导力峰会

Geekbang. 极客邦科技 | TGO 鲲鹏會

500+高端科技领导者与你一起探讨
技术、管理与商业那些事儿

2019年6月14–15日 | 上海圣诺亚皇冠假日酒店

扫码了解更多信息

自我介绍

蒋航 (庄公)

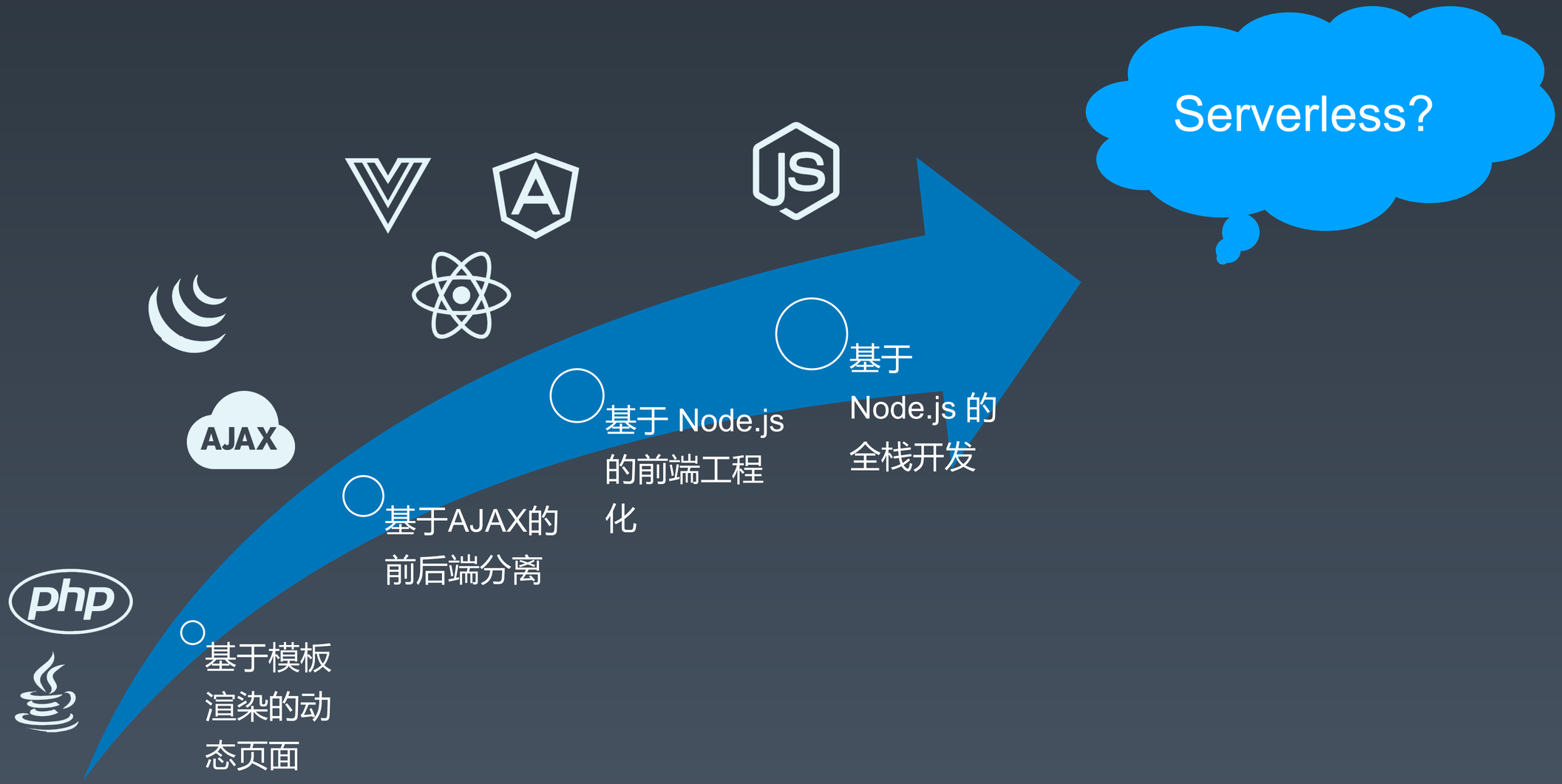
阿里云前端工程师，主要负责阿里云智能产品数字化平台的研发工作，以及平台的质量、监控等基础设施建设。

GitHub: https://github.com/nodejh
Email: jianghangscu@gmail.com

目录

1. 前端开发在 Serverless 时代的演进

前端开发模式的演进


Serverless?
AJAX
php
基于模板渲染的动态页面
基于AJAX的前后端分离
基于 Node.js 的前端工程化
基于 Node.js 的全栈开发

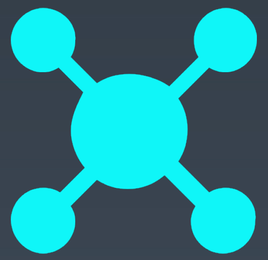
什么是 Serverless

Serverless computing refers to the concept of building and running applications that do not require server management.

无服务器计算是指构建和运行不需要服务器管理的应用程序的概念。

—— CNCF

前端早已与 Serverless 产生了联系

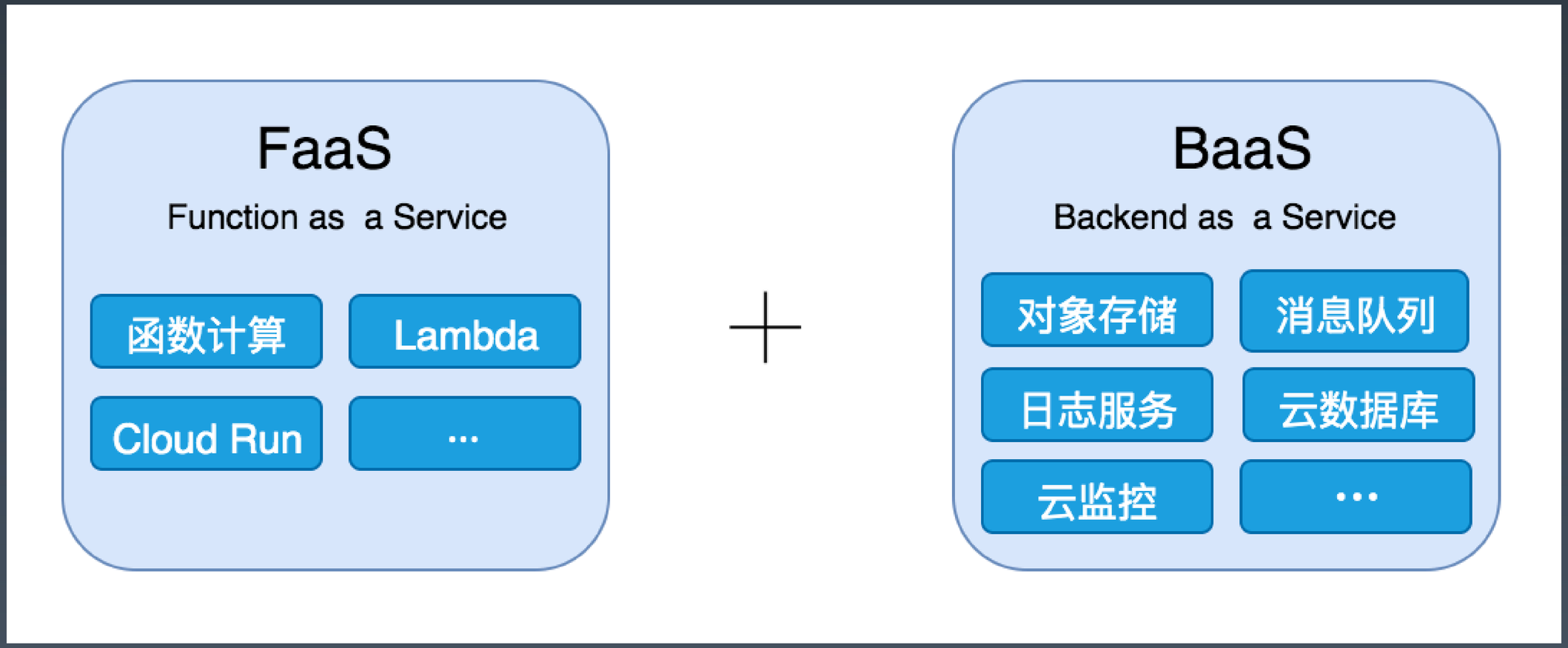


CDN

对象存储

API 服务

Serverless = FaaS + BaaS



Serverless 的特点

事件驱动

Event Driven

无状态

Stateless

无运维

NoOps

低成本

Lowcost

2. Serverless 服务中的前端解决方案

Serverless 服务中的前端解决方案

应用场景

小程序　IoT　• • •

开发工具

CLI　@alicloud/fun　fcli　aws　serverless　now　apex

Web IDE　FC IDE　AWS Cloud9

Framework　serverless　ZEIT　APEX serverless architecture

基础设施

FaaS　函数计算　Lambda　Cloud Run　Azure Functions

Compute Platform　阿里云　aws　Google Cloud　Microsoft Azure

不同 Serverless 服务的对比

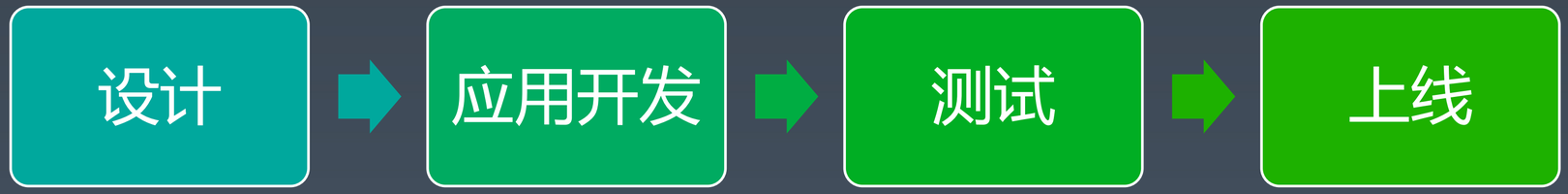
特征	阿里云函数计算	AWS Lambda	Azure Functions
支持语言	Node.js / Python / PHP / Java / C#	Node.js / Java / Go / PowerShell / C# / Python / Ruby	Node.js / C# / F# / Java / Python
触发器	OSS / API 网关 / HTTP / CDN / 日志 / 定时任务 / RDS / 表格存储 / MNS / Datahub / IoT	S3 / DynamoDB / SNS / SES / SQS / Cognito / CloudFormation / CloudWatch / Timer / Alexa / API Gateway / IoT / CloudFront	Blob Storage / Cosmos DB / Event Grid / Event Hubs / HTTP & Webhooks / Timer / Queue storage / Service Bus
边缘计算	IoT Edge	CloudFront	IoT Edge
HTTP(S) 调用	API 网关 / HTTP 触发器	API 网关	HTTP 触发器
函数组	支持	不支持	不支持
开发工具	fun / fcli / Web IDE	AWS SAM CLI / Cloud9 IDE	Azure CLI
价格	100万次免费，之后 ¥1.33/百万次 400,000GB-s免费，之后 ¥0.00011108/GB-s	100万次免费，之后 0.2$/百万次 400,000 GB-s免费，之后 $0.00001667/GB-s	100万次免费，之后 0.2$/百万次 400,000 GB-s免费，之后 $0.000016/GB-s

3. 基于 Serverless 的前端开发模式

Serverless 开发流程

传统开发流程 （前端工程师、后端工程师、运维工程师）

Serverless 开发流程 （前端工程师）



Serverless 实践

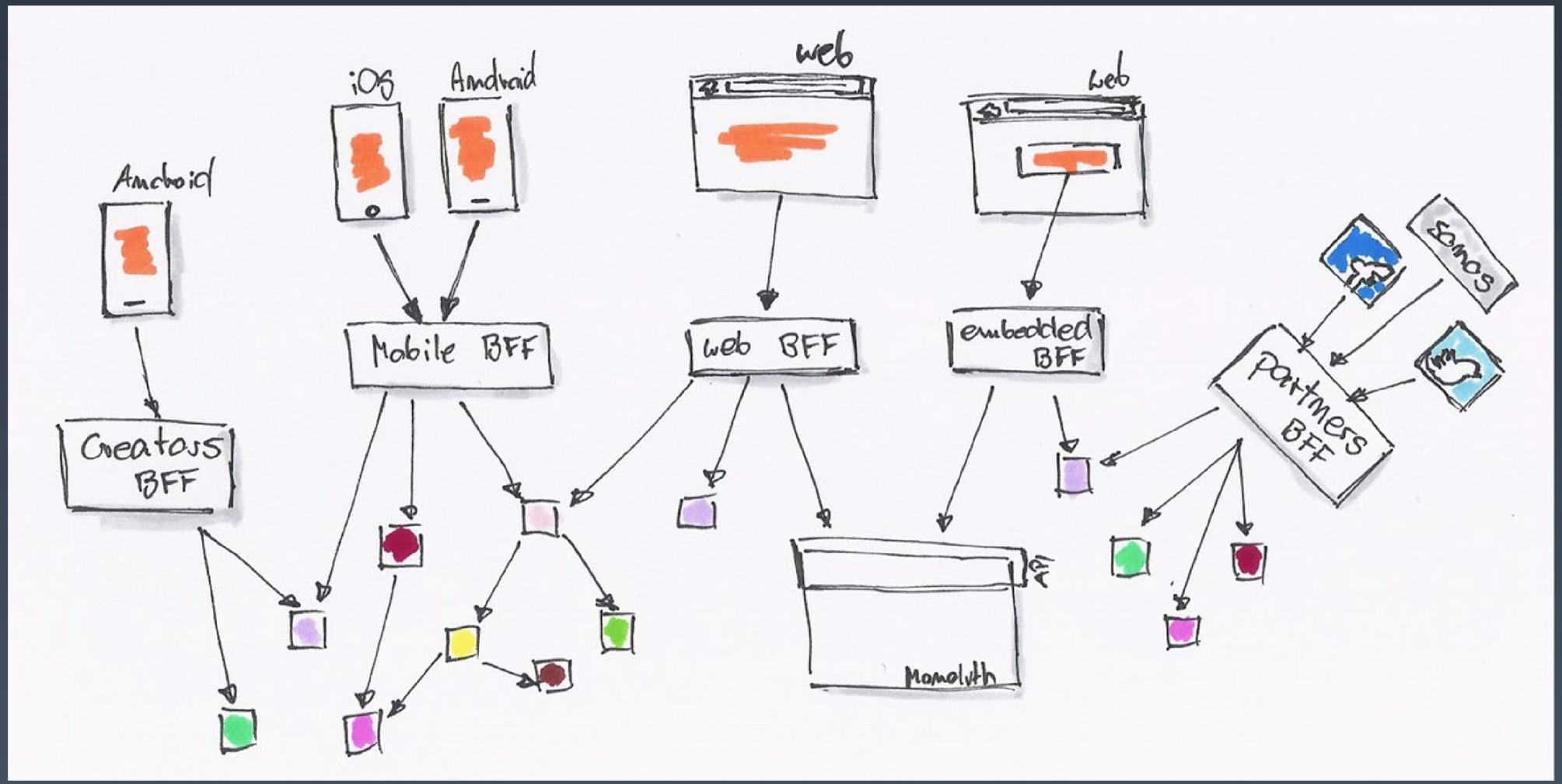
基于 Serverless 的 BFF
(Backend For Frontend)

基于 Serverless 的服务端渲染

基于 Serverless 的小程序开发

基于 Serverless 的 BFF (Backend For Frontend)

传统 BFF (Backend For Frontend) 架构



https://www.thoughtworks.com/insights/blog/bff-soundcloud

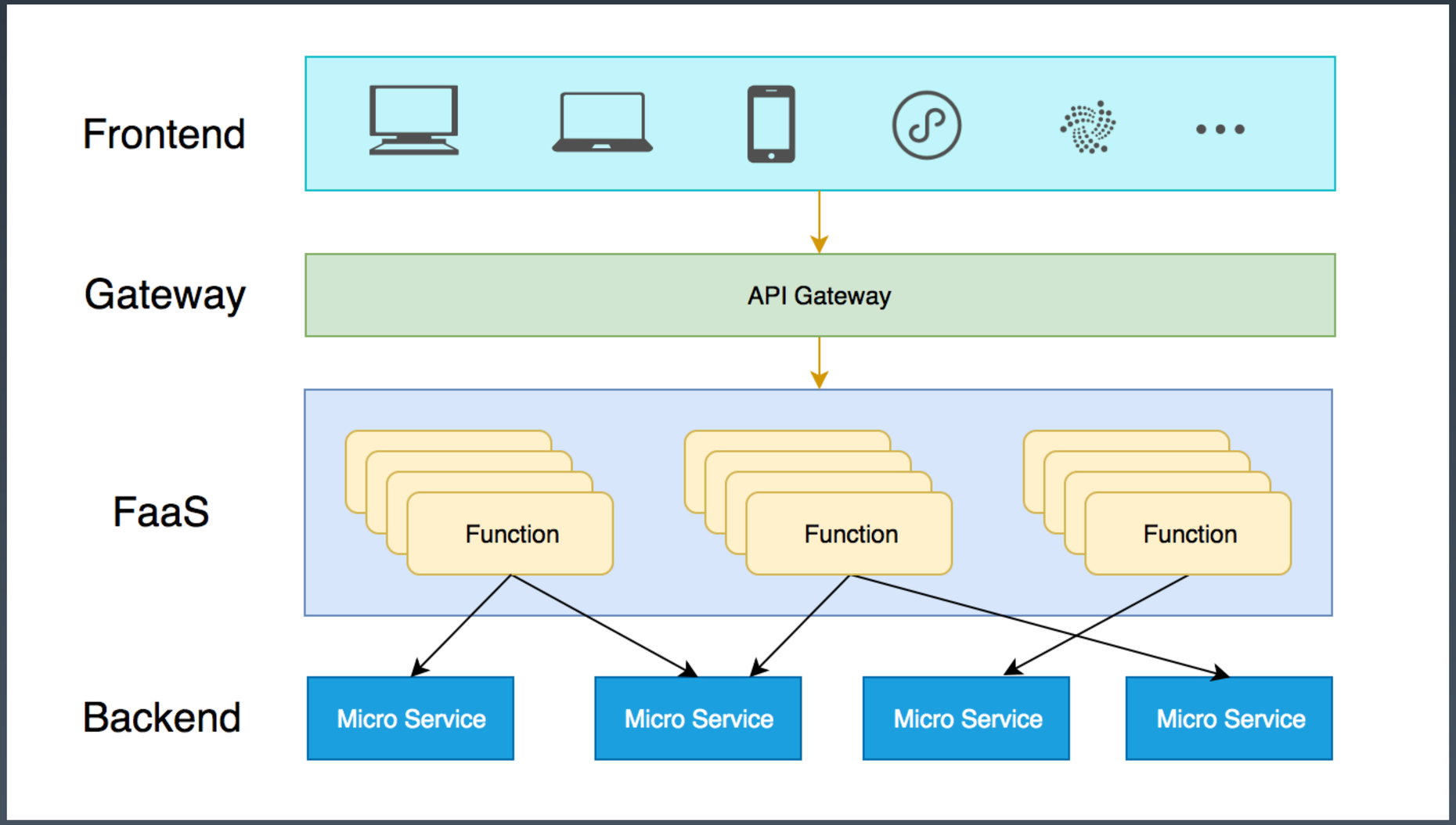
传统 BFF (Backend For Frontend) 的痛点

运维成本难降低

入口分散难管理

重复开发难避免

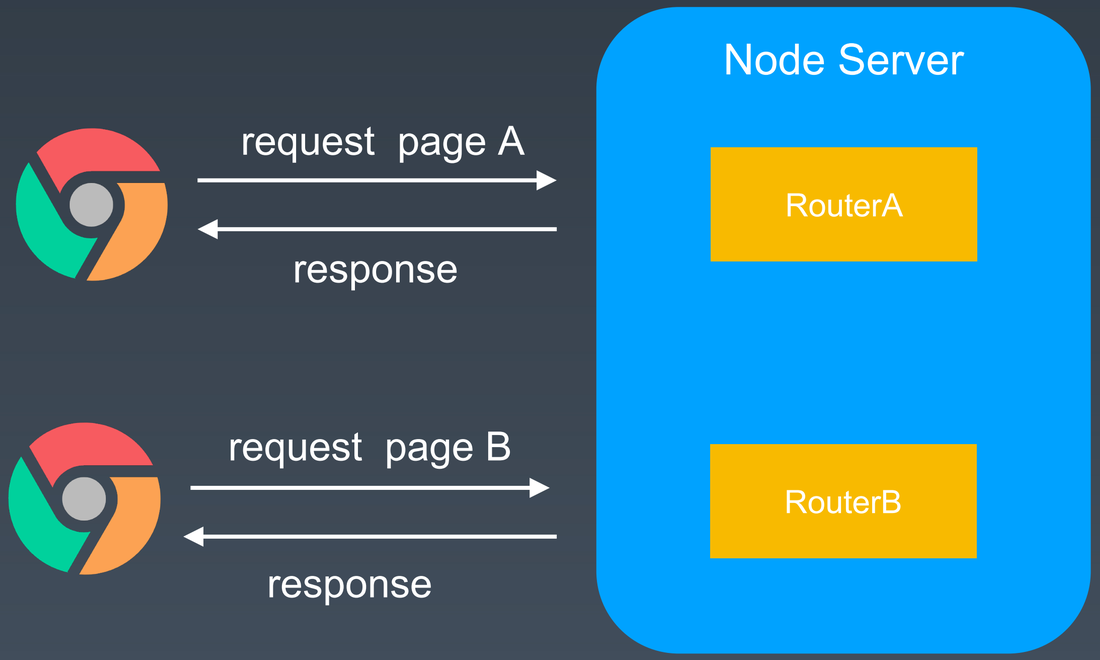
基于 Serverless 的 BFF 架构



基于 Serverless 的服务端渲染

传统服务端渲染

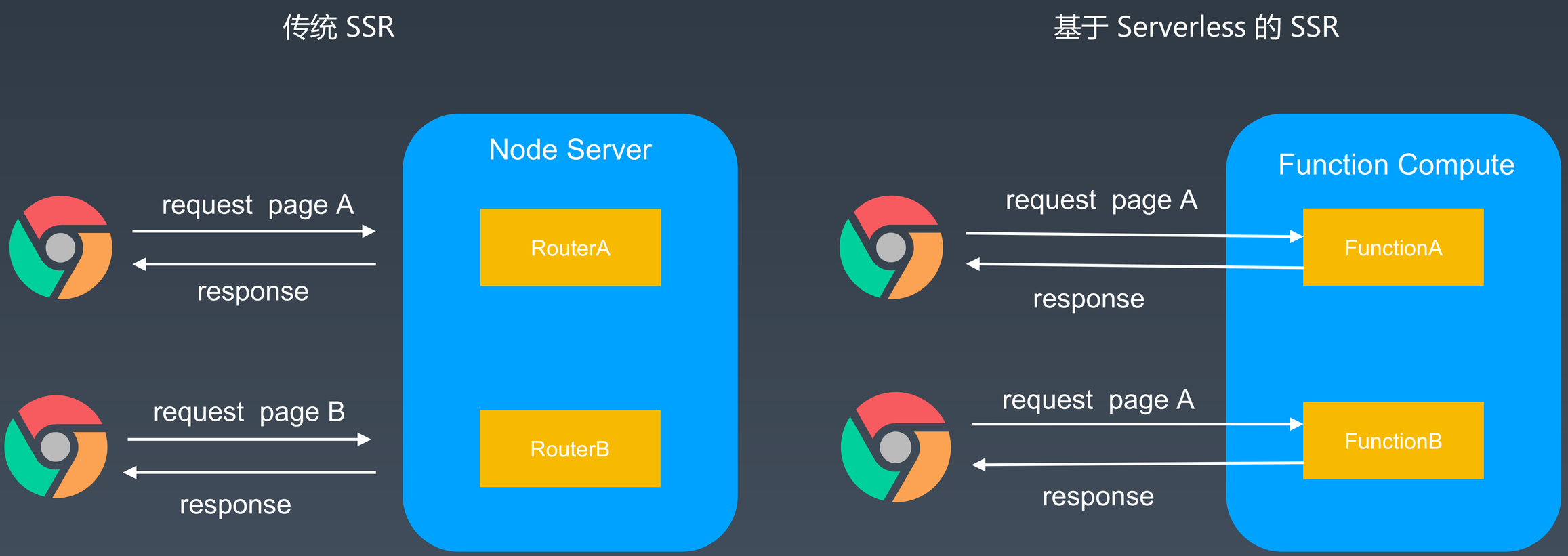
传统 SSR



性能要求高

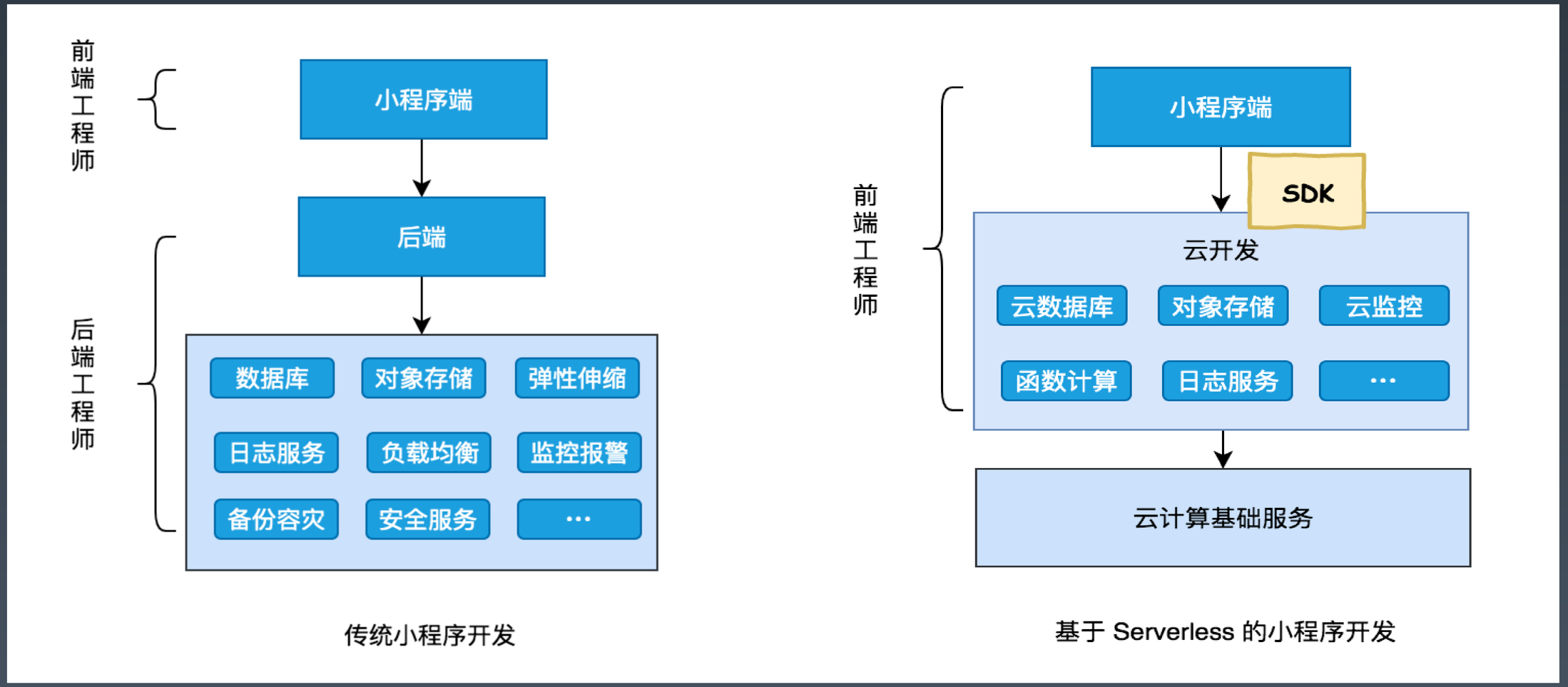
运维成本高

基于 Serverless 的服务端渲染



基于 Serverless 的小程序开发

基于 Serverless 的小程序开发



小程序云开发

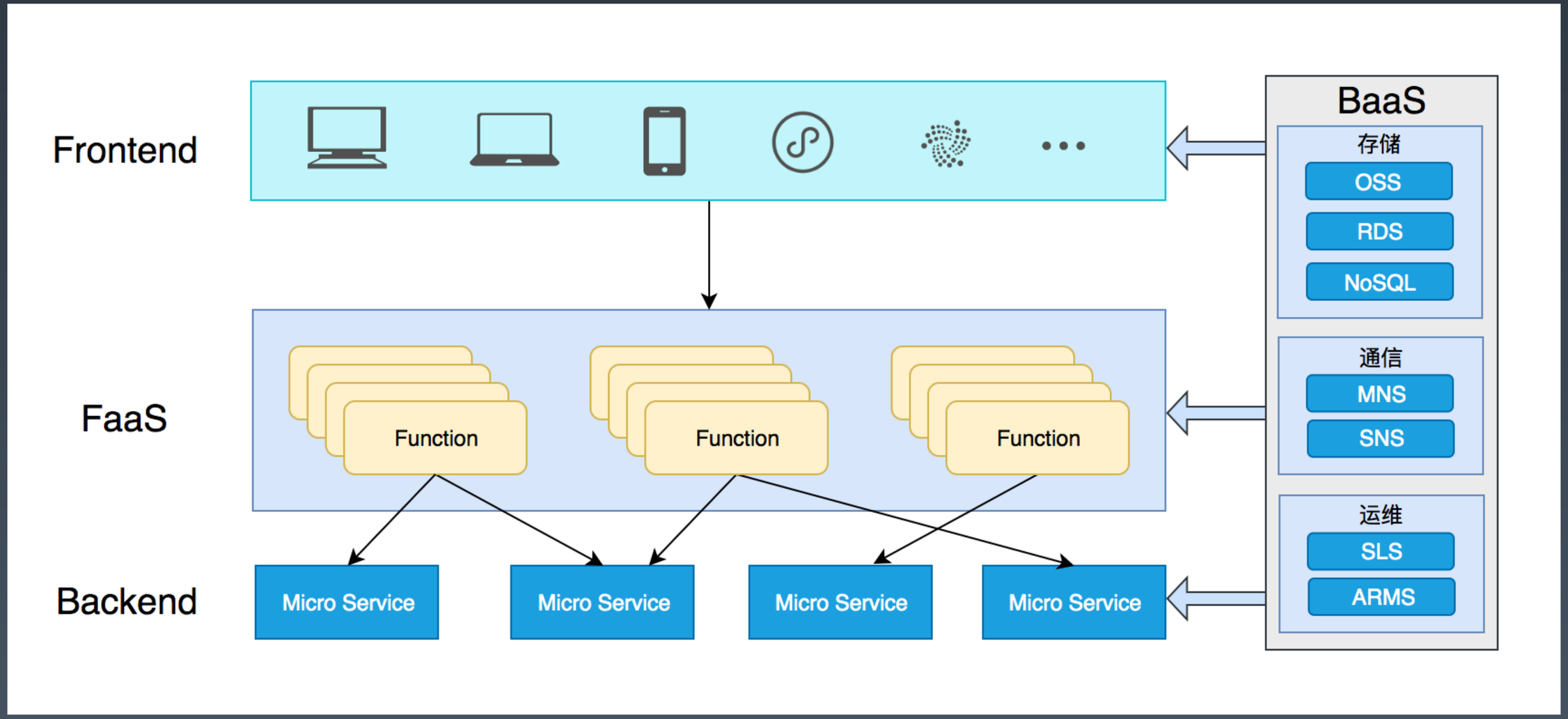
```
// 操作数据库
basement.db.collection('
  .insertOne({
        name: 'node',
        age: 18,
    })
    .then(() => {
        resolve({ succes
    })
    .catch(err => {
        reject({ success
    });
```

```
// 上传图片
basement.file
    .uploadFile(o
    .then((image)
        this.setD
            iconU
        });
    })
    .catch(consol
```

```
// 调用函数
basement.function
    .invoke('getUserInfo')
    .then((res) => {
        this.setData({
            user: res.result
        });
    })
    .catch(console.error}
```

通用 Serverless 架构

通用 Serverless 架构



4. Serverless 开发最佳实践

函数的测试

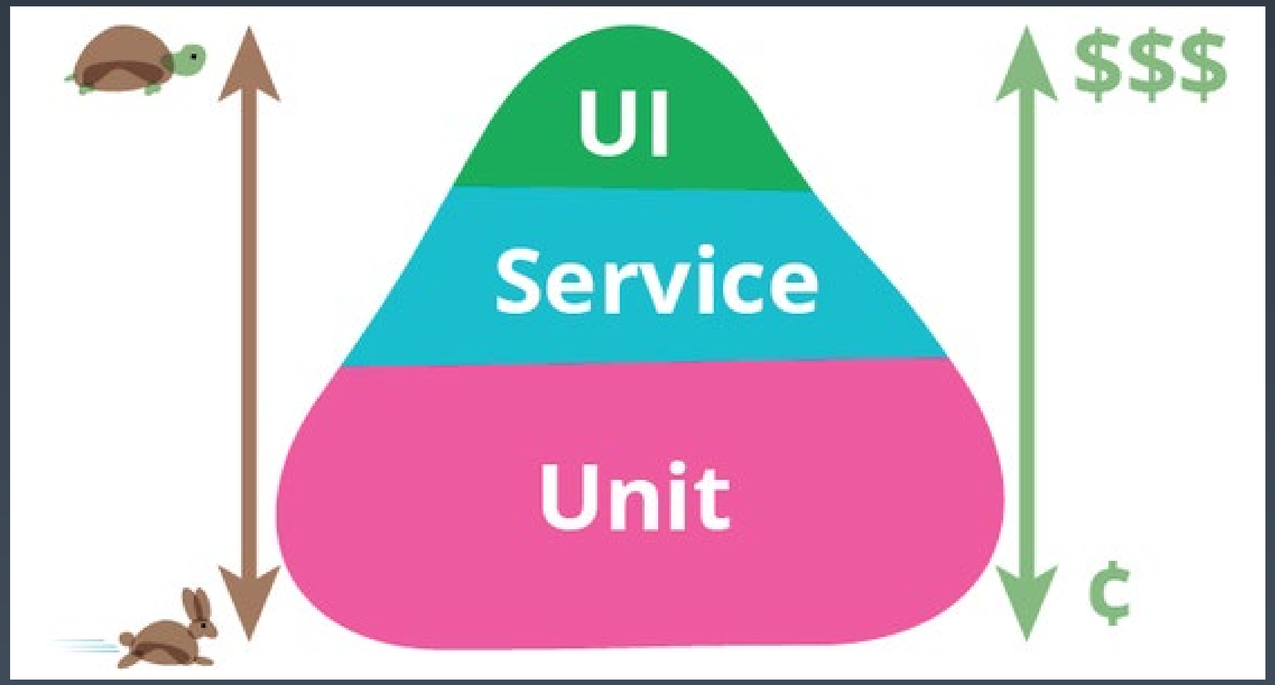
函数测试的难点

函数是分布式的，需要单元测试和集成测试

函数依赖的 FaaS 和 BaaS 服务难以在本地模拟

驱动函数执行的事件难以模拟

测试金字塔



Mike Cohn，敏捷联盟和Scrum联盟的创始成员，著有《应用于敏捷软件开发的用户故事》、《敏捷评估与规划》、《成功与敏捷》。

Serverless 测试原则

1. 业务逻辑与第三方服务分离
2. 对业务逻辑充分进行单元测试
3. 编写集成测试验证与其他服务的集成是否正常工作

一个不好的示例

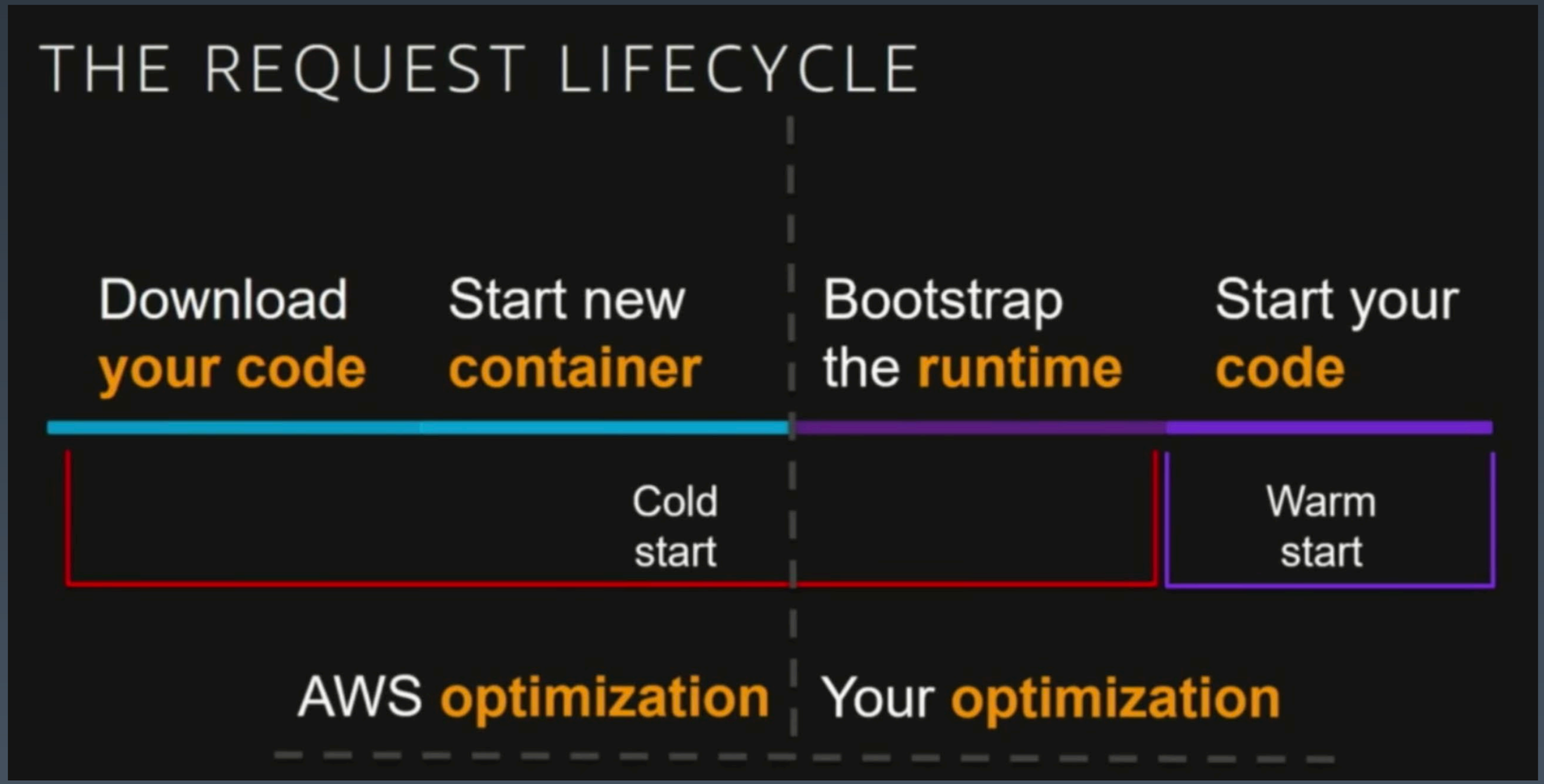
```
const db = require('db').connect();
const mailer = require('mailer');

module.exports.saveUser = (event, context, callback) => {
  const user = {
    email: event.email,
    created_at: Date.now()
  }

  db.saveUser(user, function (err) {
    if (err) {
      callback(err);
    } else {
      mailer.sendWelcomeEmail(event.email);
      callback();
    }
  });
};
```

```
class Users {
  constructor(db, mailer) {
    this.db = db;
    this.mailer = mailer;
  }

  save(email, callback) {
    const user = {
      email: email,
      created_at: Date.now()
    }

    this.db.saveUser(user, function (err) {
      if (err) {
        callback(err);
      } else {
        this.mailer.sendWelcomeEmail(email);
        callback();
      }
  });
  }
}

module.exports = Users;
```

```
const db = require('db').connect();
const mailer = require('mailer');
const Users = require('users');

let users = new Users(db, mailer);

module.exports.saveUser = (
  event,
  context,
  callback
) => {
  users.save(event.email, callback);
};
```

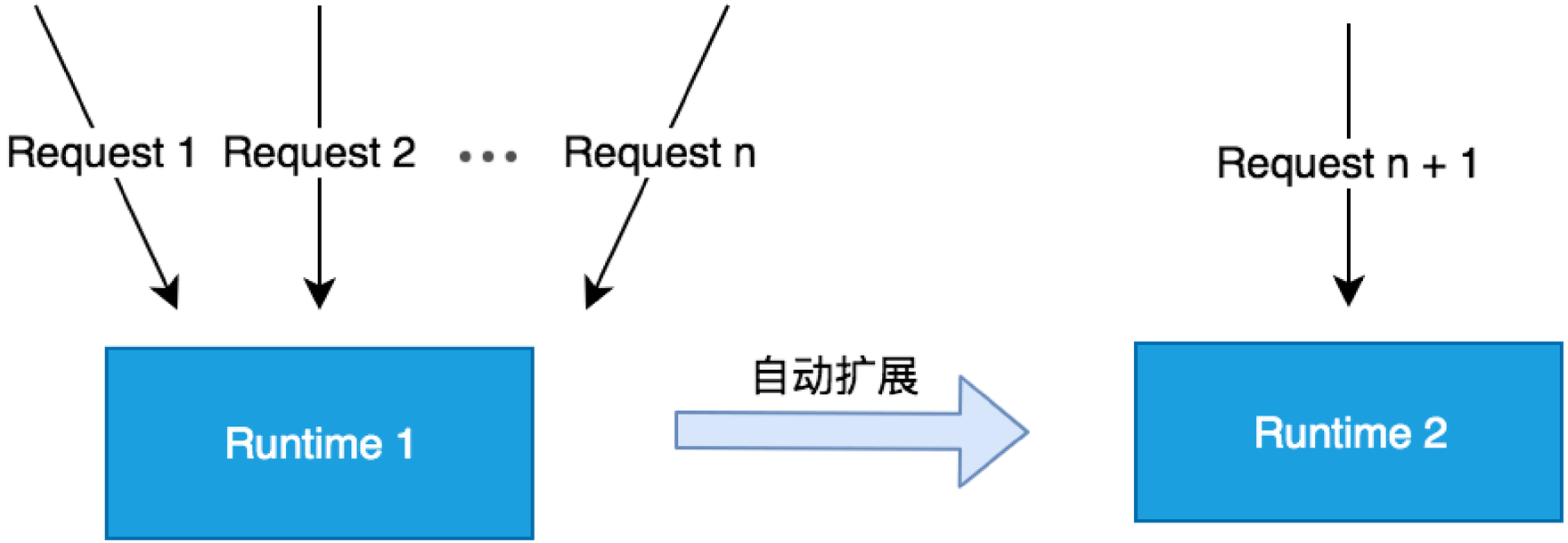
函数的性能

函数生命周期

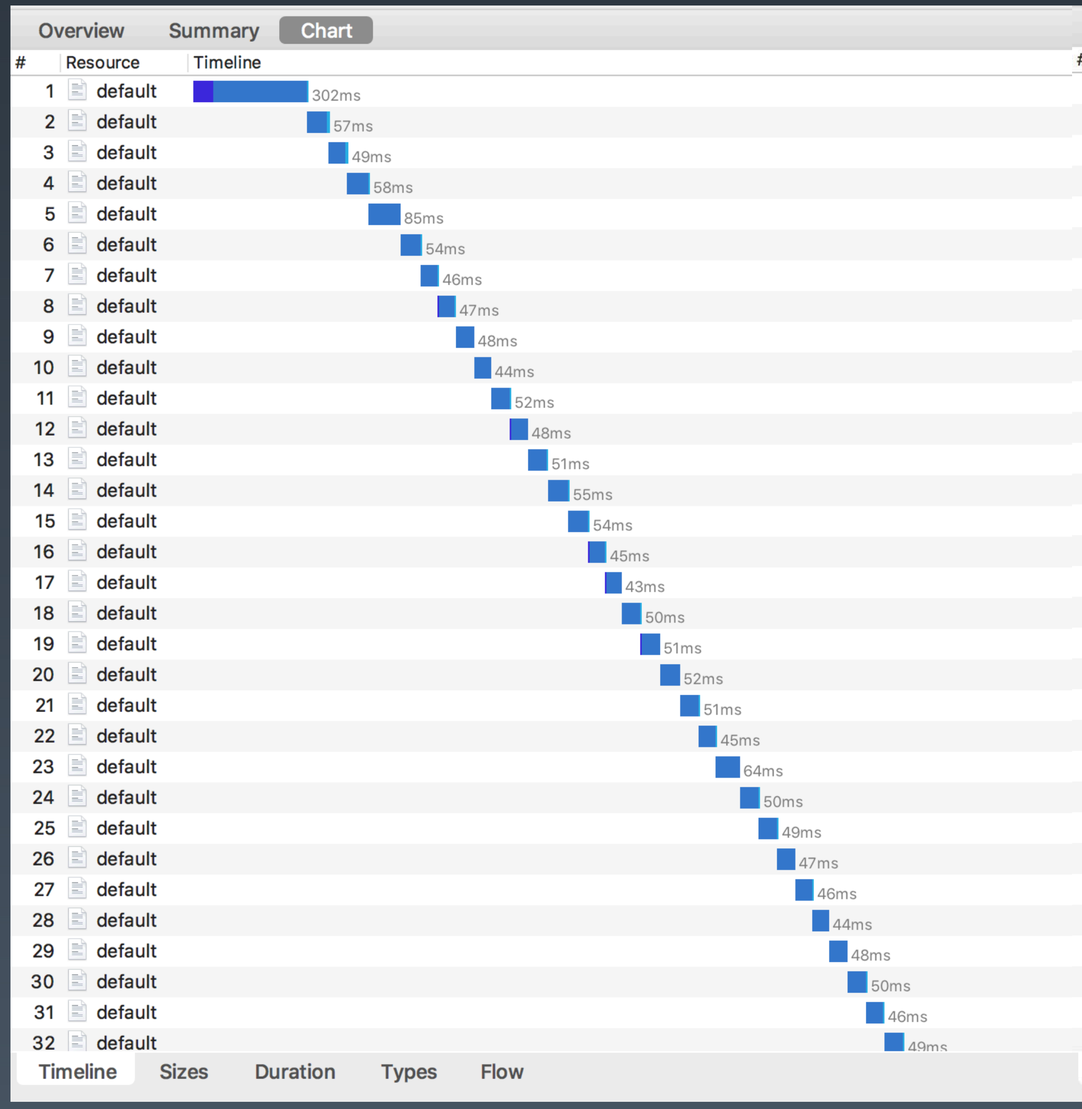


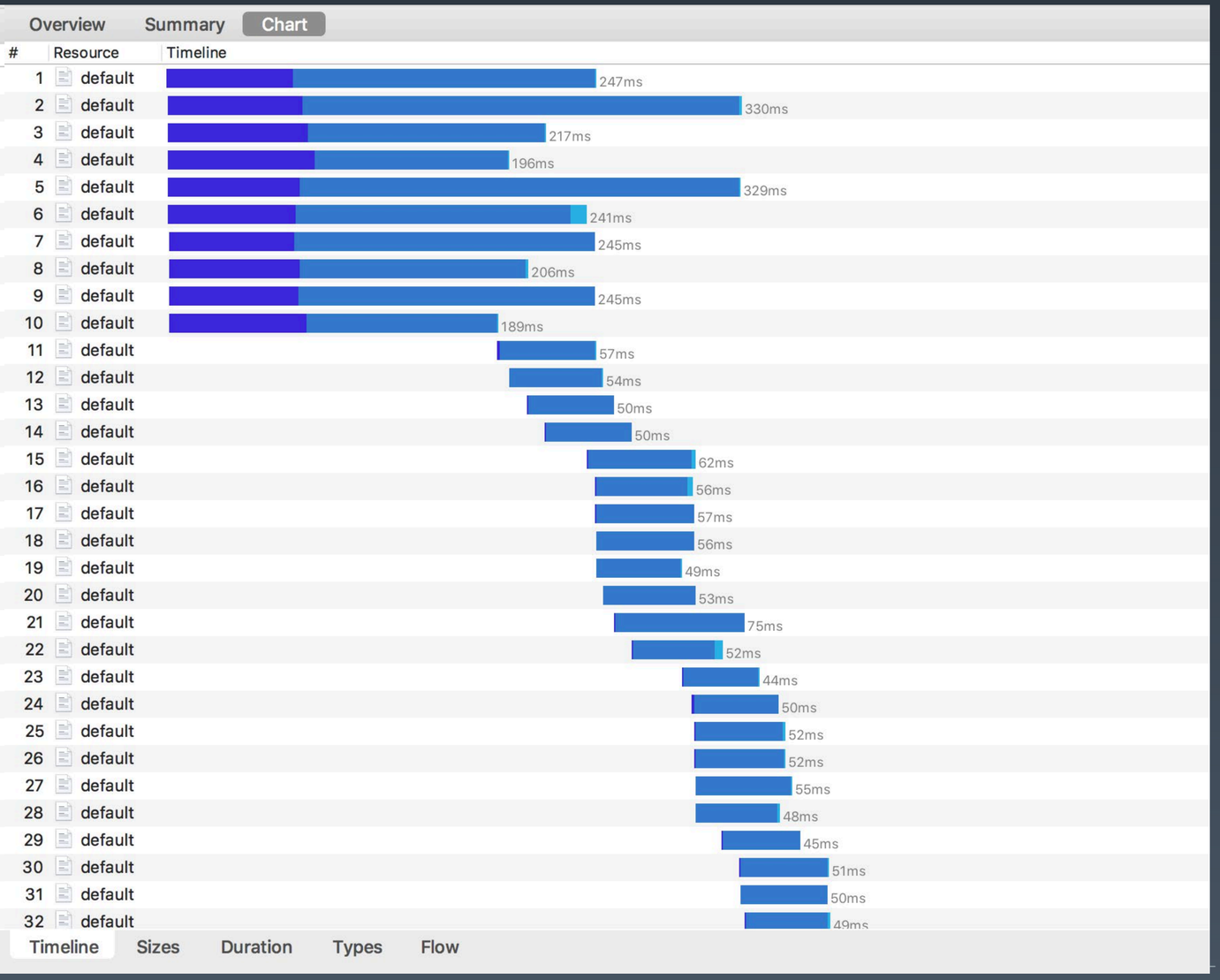
https://www.youtube.com/watch?v=oQFORsso2go&feature=youtu.be&t=8m5s

函数的自动扩展


Request 1
Request 2
Request n
Request n + 1
Runtime 1
自动扩展
Runtime 2

并发对函数运行时间的影响





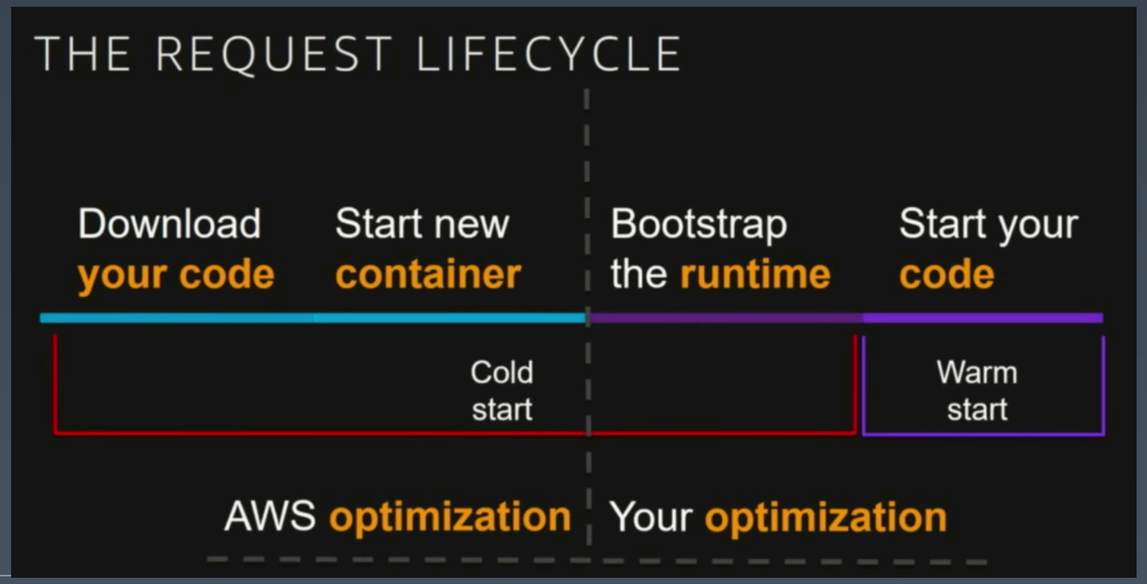
执行上下文重用

```
const mysql = require('

module.exports.saveUser

    // 初始化数据库连接
    const connection =
    connection.connect(

    connection.query('.

};
```

```
const mysql = require('mysql');

// 初始化数据库连接
const connection = mysql.createConnection({ /* ... */ });
connection.connect();

module.exports.saveUser = (event, context, callback) => {

    connection.query('...');

};
```

如何提高函数性能

- 使用 Node.js / Python 等冷启动时间低的编程语言
- 优化依赖关系和包的大小
- 执行上下文重用
- 为函数分配最佳运行内存



4. 总结与展望

总结与展望

- 前端将会重新回归到 Web 应用工程师这一职能
- 实时 SSR 将成为展示端 UI 的主要渲染模式
- 基于场景的云开发将成为主流开发模式

Less is more.

参考

- Cloud Programming Simplified: A Berkeley View on Serverless Computing
- Serverless Architectures
- The Forgotten Layer of the Test Automation Pyramid
- How does language, memory and package size affect cold starts of AWS Lambda?
- How to manage Lambda VPC cold starts and deal with that killer latency
- I’m afraid you’re thinking about AWS Lambda cold starts all wrong
- Serverless 风暴来袭，前端工程师如何应对

Geekbang. 极客邦科技 InfoQ

极客邦科技
会议推荐2019

5月

QCon 北京
全球软件开发大会
大会：5月6–8日
培训：5月9–10日

QCon 广州
全球软件开发大会
培训：5月25–26日
大会：5月27–28日

6月

GTLC GLOBAL TECH LEADERSHIP CONFERENCE 上海
技术领导力峰会
时间：6月14–15日

GMTC 北京
全球大前端技术大会
大会：6月20–21日
培训：6月22–23日

7月

ArchSummit 深圳
全球架构师峰会
大会：7月12–13日
培训：7月14–15日

10月

QCon 上海
全球软件开发大会
大会：10月17–19日
培训：10月20–21日

11月

GMTC 深圳
全球大前端技术大会
大会：11月8–9日
培训：11月10–11日

AiCon 北京
全球人工智能与机器学习大会
大会：11月21–22日
培训：11月23–24日

12月

ArchSummit 北京
全球架构师峰会
大会：12月6–7日
培训：12月8–9日

THANKS!

QCon 10th