Node.js 开发精要

熊伟伦

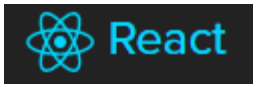
2017/9/13复杂美技术沙龙

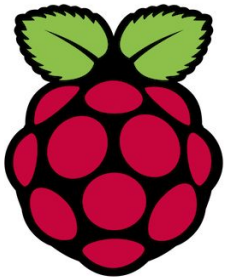
自我介绍

- https://github.com/Azard | http://azard.me
- SJTU SE 12级本科 16级硕士
- 代表性工作
  - Magic 音乐手套，首届全国大学生物联网设计竞赛特等奖
  - Appetizer.io 移动开发 DevOps 平台
- Android
  - 字节码插桩程序分析、自动化测试 Appetizer.io
  - 多篇浏览过万的技术博客，CNCC 2016, DroidCon 2016
- Node
  - 参与 Node, Egg.js 开源贡献，开发多个 Egg.js 插件
  - 开发十多个仍在线上运行的 Node 项目

Node 能做什么

- Web 服务器
  - 视图模板：Pug, Nunjucks
  - API 服务：Express, Koa
  - 长连接：Socket.io
  - 全栈解决方案： Meteor
- 命令行工具
  - 前端构建：webpack, gulp
  - 后端构建：npm, yarn
  - 前端框架：React, Vue, Angular
- IoT
  - 树莓派

Node 能做什么

- 桌面应用
  - NW.js (node-webkit)
  - Electron
- 移动应用
  - React Native
  - Weex, Rax
  - Ionic
- 云计算
  - Serverless
  - 函数计算

发展历程

- Node.js　Node　~~NodeJS~~　~~Nodejs~~
- 开源跨平台 JavaScript 运行时
- 2009: Ryan Lienhart Dahl, JSConf Europe
- 2010: npm
- 2011: Node.js on Windows
- 2014: Version 1.0.0
- 2014: io.js
- 2015: Node.js 基金会
- 2017.05.30, Version 8.0.0
- 2017.8: Ayo.js

创建 HTTP 服务器

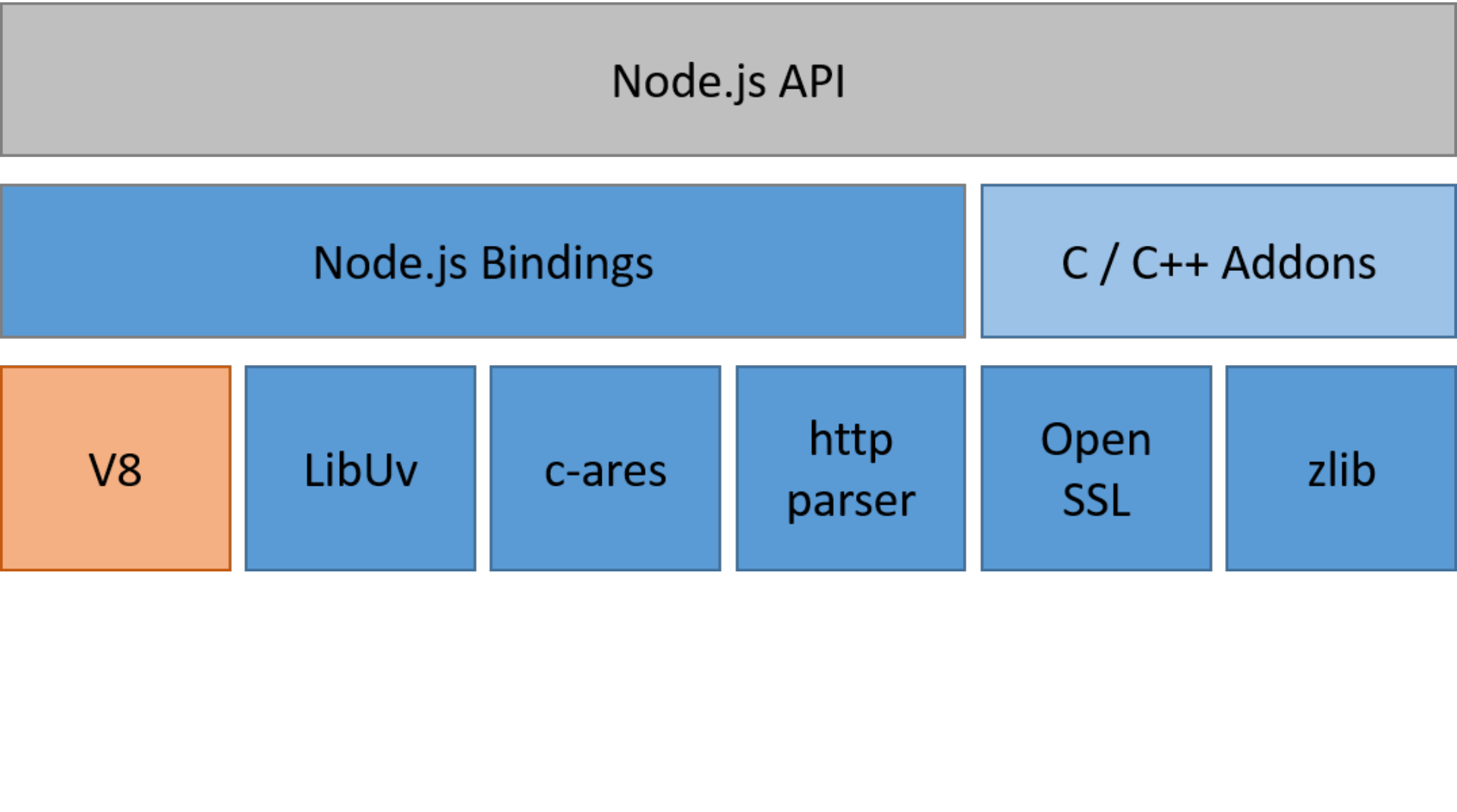
```
const http = require('http');

const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

什么是 Node.js

- 最初目标：高性能 Web 服务器
- 寓意：互联网上的一个节点

集齐龙珠，召唤神龙

- TC39-ECMA, http://node.green/
- V8, Java HotSpot, Google, Chrome
- libuv, npm
- 社区的力量


Node.js API
Node.js Bindings
C / C++ Addons
V8
LibUv
c-ares
http parser
Open SSL
zlib

Node.js 的优势

- 同期的明星们 Scala Clojure Go

- 原因一：用事件回调解决了异步编程问题
- 原因二：JavaScript 上手简单，赋能传统前端
- 原因三：互联网，微服务，天时地利

- 非阻塞异步 IO　　Non-blocking async IO
- 单进程执行代码（libuv 和 V8 部分使用多进程）
- 微核心，第三方扩展

事件驱动模型

- 异步编程的发展

- Stage 0：单进程同步（学生作业）
- Stage 1：Fork 进程（早期的 Apache 服务器）
- Stage 2：线程池 （Tomcat）

- Stage 3：IO 多路复用（轻量级伪线程）
- Stage 3.1：事件驱动（Node.js, Nginx）
- Stage 3.2：协程 （goroutine, python3.5 coroutine）

事件驱动和回调

- 循环消费事件队列

```
const fs = require('fs');
const data = fs.readFileSync('/file.md'); // blocks here until file is read
console.log(data);
// moreWork(); will run after console.log
```

```
const fs = require('fs');
fs.readFile('/file.md', (err, data) => {
  if (err) throw err;
  console.log(data);
});
// moreWork(); will run before console.log
```

uses async/await

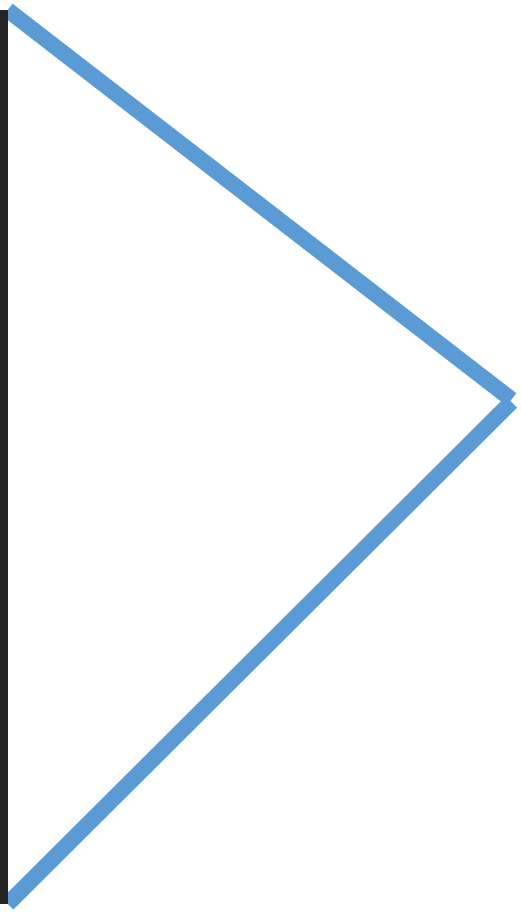
uses generators with co

uses Promise.then chains

uses callbacks

回调地狱，2015年以前

```
1   db.findOne({_id: req.id}, (err, result) => {
2     if (err) {
3       // handle error and return
4     }
5     if (encrypt(req.password, SALT, (err, pwd) => {
6       if (err) {
7         // handle error and return
8       }
9       if (pwd === result.password) {
10        // return success
11      }
12      else {
13          // handle password error
14      }
15    }));
16  });
17
```



Promise，2015~2016

```
db.findOne({_id: req.id})
.then(result => {
  encrypt(req.password, SALT);
})
.then(pwd => {
  if (pwd == result.password) {
    // TODO
  }
  else {
    // TODO
  }
})
.catch(e => {
  // handle error
})
```

async/await，2017年 Node v7.6.0

```
async login() {
  try {
    const result = await db.findOne({_id: req.id});
    const pwd = await encrypt(req.password, SALT);
    if (pwd === result.password) {
      // TODO
    }
    else {
      // TODO
    }
  } catch (e) {
    // TODO
  }
}
```

promisify

- bluebird.promisify()
- util.promisify()
  - Node.js V8.0.0 —— 2017.05.30

```
const util = require('util');
const exec = require('child_process').exec;

/**
 * CallBack
 */
exec('ls', (err, stdout, stderr) => {
  if (err) console.error(err);
  console.log('stdout:', stdout);
  console.log('stderr:', stderr);
});
```

```
const execPromise = util.promisify(exec);

// async/await Node7.6.0 ~
(async () => {
  const {
    stdout,
    stderr
  } = await execPromise('ls');

  console.log('stdout:', stdout);
  console.log('stderr:', stderr);
})();
```

uses async/await

uses generators with co

uses Promise.then chains

uses callbacks

```
function IWantFullCallbacks() {
  setTimeout(function() {
    const localStack = new Error();
    console.log(localStack.stack);
  }, 1000);
}

IWantFullCallbacks();
```

```
Error
  at  Timeout._onTimeout (/Users/pmq20/1.js:3:24)
  at ontimeout (timers.js:488:11)
  at tryOnTimeout (timers.js:323:5)
  at Timer.listOnTimeout (timers.js:283:5)
```

```
function IWantFullCallbacks() {
  setTimeout(function() {
    const localStack = new Error();
    console.log(localStack.stack);
  }, 1000);
}

IWantFullCallbacks();
```

```
Error
  at  Timeout._onTimeout (/Users/pmq20/1.js:3:24)
  at ontimeout (timers.js:488:11)
  at tryOnTimeout (timers.js:323:5)
  at Timer.listOnTimeout (timers.js:283:5)
```

FAQ 1

如何获取全部的调用栈？

FAQ 1

如何获取全部的调用栈？

AsyncHook (Node >= 8.0.0)

什么是 Hook？

```
const stack = new Map();
stack.set(-1, '');
let currentUid = -1;

function init(id, type, triggerId, resource) {
  const localStack = (new Error()).stack.split('\n').slice(1).join('\n');
  const extraStack = stack.get(triggerId || currentUid);
  stack.set(id, localStack + '\n' + extraStack);
}
function before(uid) {
  currentUid = uid;
}
function after(uid) {
  currentUid = -1;
}
function destroy(uid) {
  stack.delete(uid);
}

const async_hooks = require('async_hooks');
const hook = async_hooks.createHook({init, before, after, destroy});
hook.enable();
```

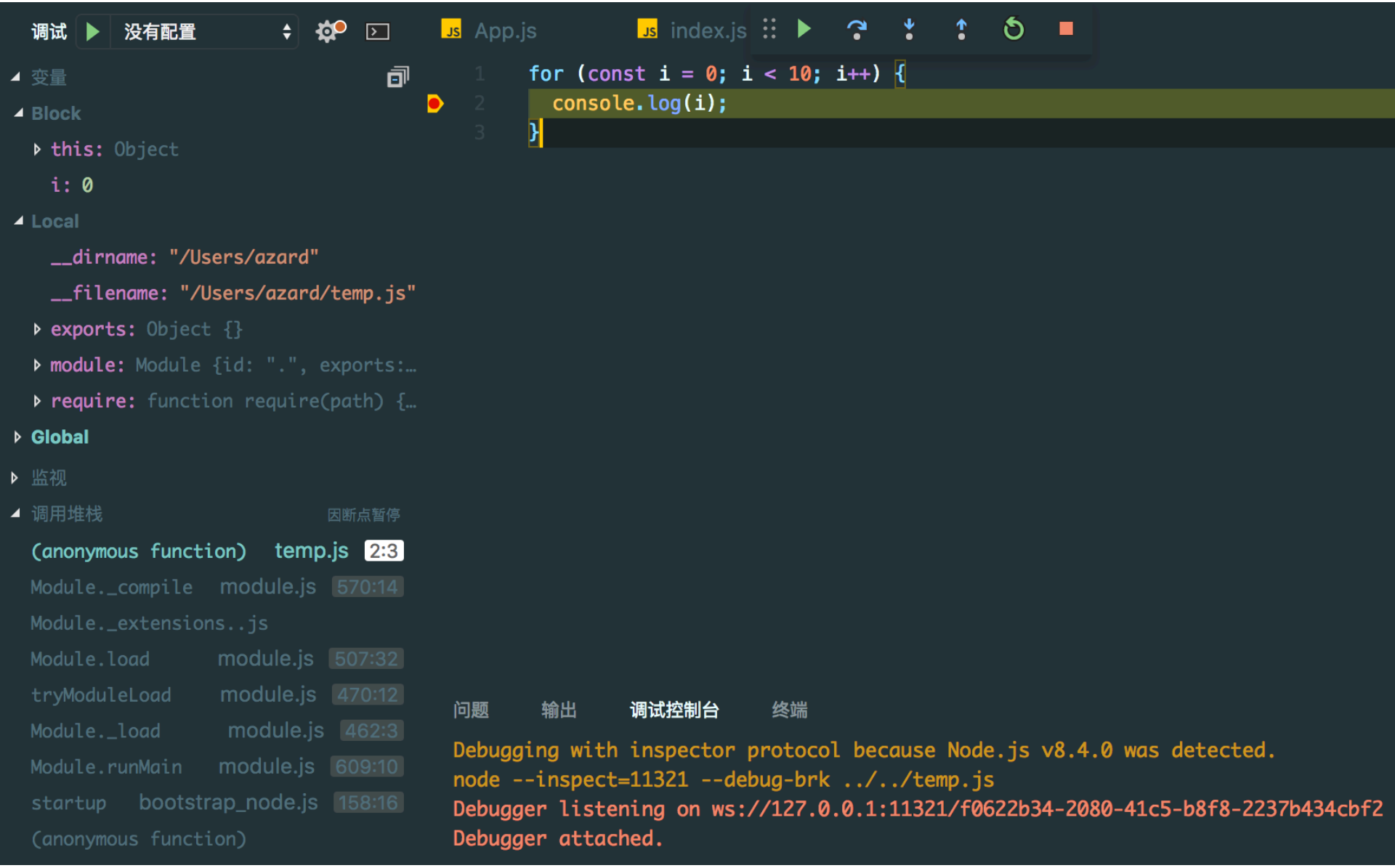
```
function IWantFullCallbacks() {
  setTimeout(function() {
    const localStack = new Error();
    console.log(localStack.stack);
    console.log('---');
    console.log(stack.get(currentUid));
  }, 1000);
}
```

```
Error
    at Timeout._onTimeout (/Users/pmq20/2.js:26:24)
    at ontimeout (timers.js:488:11)
    at tryOnTimeout (timers.js:323:5)
    at Timer.listOnTimeout (timers.js:283:5)
---
    at init (/Users/pmq20/2.js:6:23)
    at runInitCallback (async_hooks.js:459:5)
    at emitInitS (async_hooks.js:327:7)
    at new Timeout (timers.js:592:5)
    at createSingleTimeout (timers.js:472:15)
    at setTimeout (timers.js:456:10)
    at IWantFullCallbacks (/Users/pmq20/2.js:25:3)
    at Object.<anonymous> (/Users/pmq20/2.js:33:1)
    at Module._compile (module.js:569:30)
    at Object.Module._extensions..js (module.js:580:10)
```

FAQ 2

开发环境及调试

- VS Code
- WebStorm


调试
没有配置
App.js
index.js
for (const i = 0; i < 10; i++) {
console.log(i);
}
变量
Block
this: Object
i: 0
Local
__dirname: "/Users/azard"
__filename: "/Users/azard/temp.js"
exports: Object {}
module: Module {id: ".", exports:…
require: function require(path) {…
Global
监视
调用堆栈
因断点暂停
(anonymous function) temp.js 2:3
Module._compile module.js 570:14
Module._extensions..js
Module.load module.js 507:32
tryModuleLoad module.js 470:12
Module._load module.js 462:3
Module.runMain module.js 609:10
startup bootstrap_node.js 158:16
(anonymous function)
问题
输出
调试控制台
终端
Debugging with inspector protocol because Node.js v8.4.0 was detected.
node --inspect=11321 --debug-brk ../../temp.js
Debugger listening on ws://127.0.0.1:11321/f0622b34-2080-41c5-b8f8-2237b434cbf2
Debugger attached.

中场休息

FAQ 3

单进程只能用1个 CPU 核心？

Node.js 是否适合计算密集型的场景？

FAQ 3

单进程只能用1个 CPU 核心？

libuv, V8, C++ Library, Cluster, child process

Node.js 是否适合计算密集型的场景？

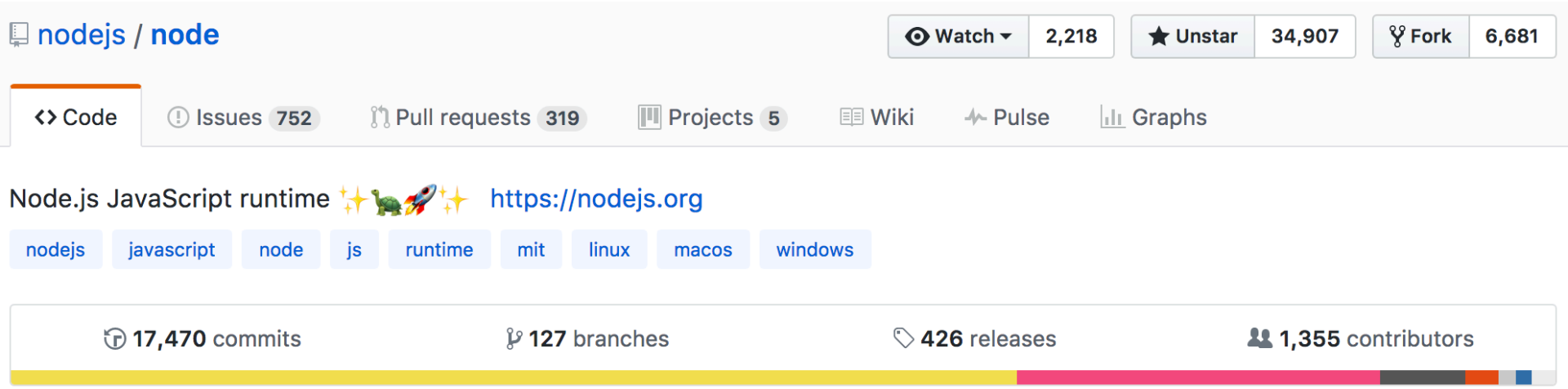
n-body

source	secs	mem	gz	cpu
Node.js	**27.76**	27,704	1297	27.76
PHP	358.21	8,668	1082	358.12
Java	21.54	27,092	1489	21.56
C++ g++	9.31	1,052	1544	9.30
Python 3	787.02	7,744	1181	786.82

https://benchmarksgame.alioth.debian.org/u64q/compare.php?lang=node&lang2=php

社区

完全的 GitHub 运作流程

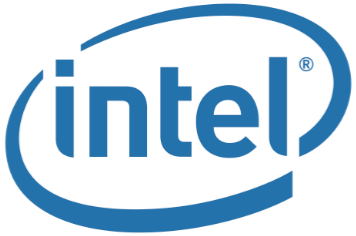


CTC (Core Technical Committee)

Collaborators

白金赞助商

Platinum



npm 与 badge

482,123 total packages

422,374,563 downloads in the last day

2,242,486,910 downloads in the last week

9,267,749,909 downloads in the last month

Born to build better enterprise frameworks and apps

TJ Holowaychuk
tj

Founder of Apex https://apex.sh **@tjholowaychuk** on Twitter & Medium.

Follow

Block or report user

Apex
Victoria, BC, Canada

Overview | Repositories 255 | Stars 1.7k | Followers 28k | Following 46

Pinned repositories

koajs/koa
Expressive middleware for node.js using ES2017 async functions
JavaScript ★ 15.5k 1.4k

apex/apex
Build, deploy, and manage AWS Lambda functions with ease (with Go support!).
Go ★ 5.3k 315

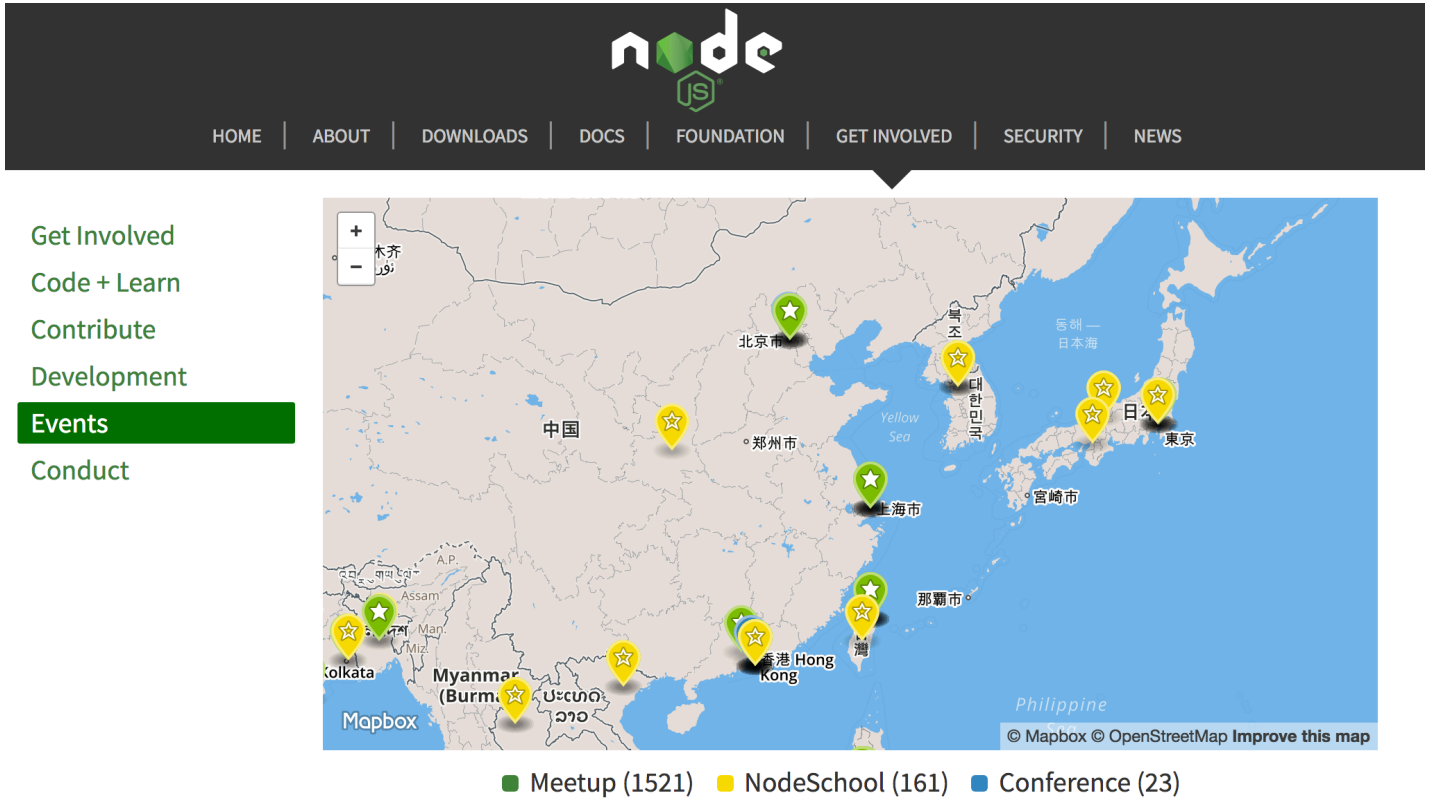
git-extras
GIT utilities -- repo summary, repl, changelog population, author commit percentages and more
Shell ★ 9.9k 727

clibs/clib
C package manager-ish
C ★ 2.5k 131

co
The ultimate generator based flow-control goodness for nodejs (supports thunks, promises, etc)
JavaScript ★ 8.1k 540

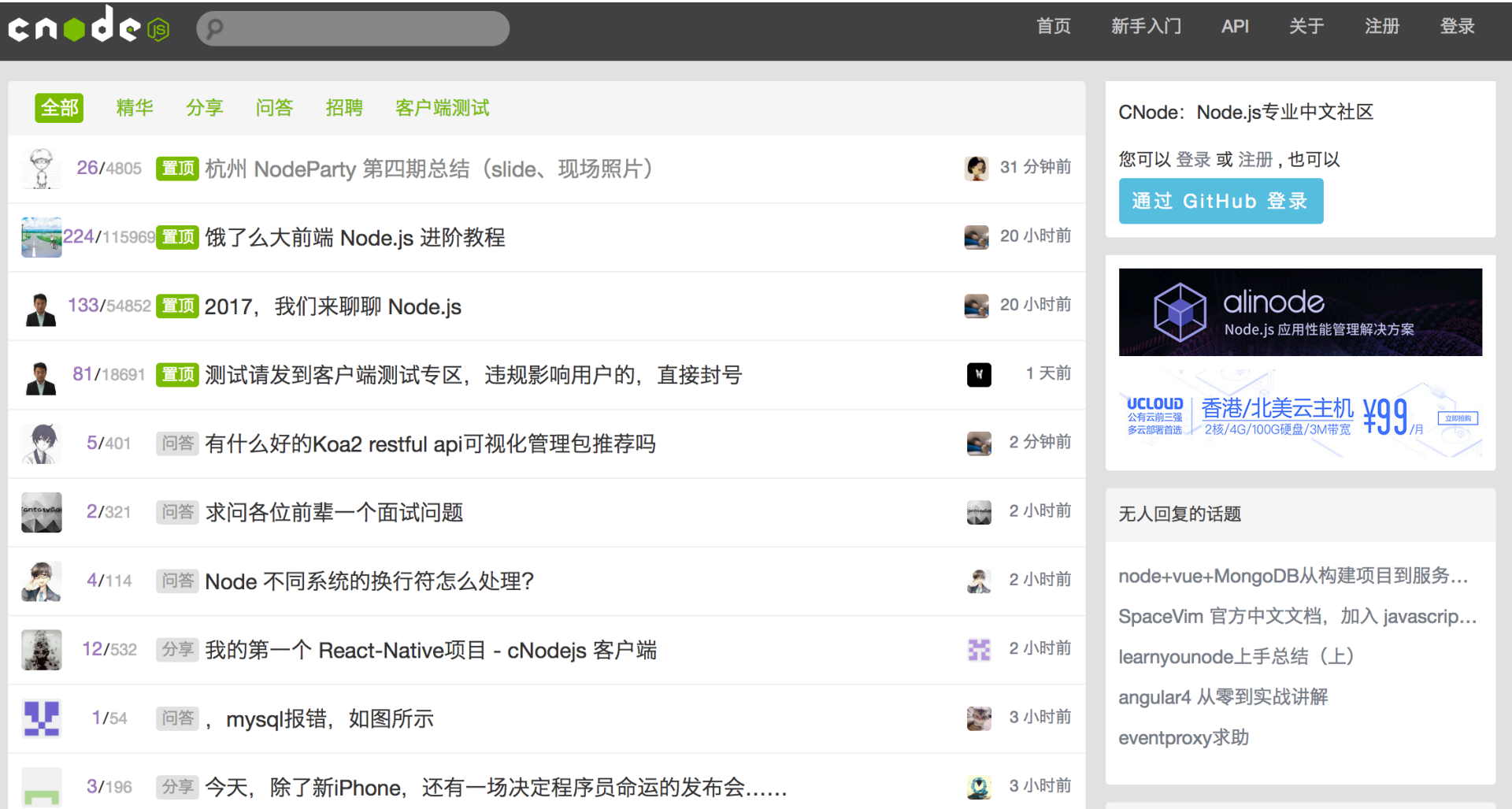
luna
luna programming language - a small, elegant VM implemented in C
C ★ 2.1k 117

社区活动



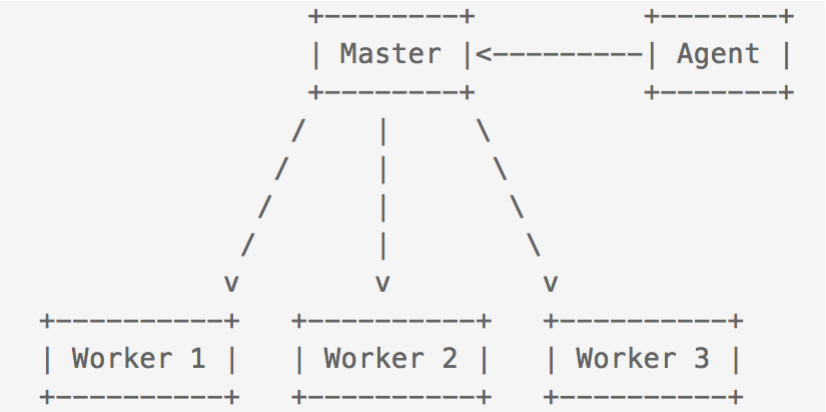
- Node.js Live Beijing 2016
- 每年的 JSConf

国内社区


cnode
首页
新手入门
API
关于
注册
登录
全部
精华
分享
问答
招聘
客户端测试
26/4805 置顶 杭州 NodeParty 第四期总结（slide、现场照片） 31 分钟前
224/115969 置顶 饿了么大前端 Node.js 进阶教程 20 小时前
133/54852 置顶 2017，我们来聊聊 Node.js 20 小时前
81/18691 置顶 测试请发到客户端测试专区，违规影响用户的，直接封号 1 天前
5/401 问答 有什么好的Koa2 restful api可视化管理包推荐吗 2 分钟前
2/321 问答 求问各位前辈一个面试问题 2 小时前
4/114 问答 Node 不同系统的换行符怎么处理? 2 小时前
12/532 分享 我的第一个 React-Native项目 - cNodejs 客户端 2 小时前
1/54 问答 ，mysql报错，如图所示 3 小时前
3/196 分享 今天，除了新iPhone，还有一场决定程序员命运的发布会…… 3 小时前
CNode：Node.js专业中文社区
您可以 登录 或 注册，也可以
通过 GitHub 登录
alinode
Node.js 应用性能管理解决方案
UCLOUD
公有云前三强
多云部署首选
香港/北美云主机
2核/4G/100G硬盘/3M带宽
¥99/月
立即抢购
无人回复的话题
node+vue+MongoDB从构建项目到服务…
SpaceVim 官方中文文档，加入 javascrip…
learnyounode上手总结（上）
angular4 从零到实战讲解
eventproxy求助

集群

- 微服务
- forever
- pm2



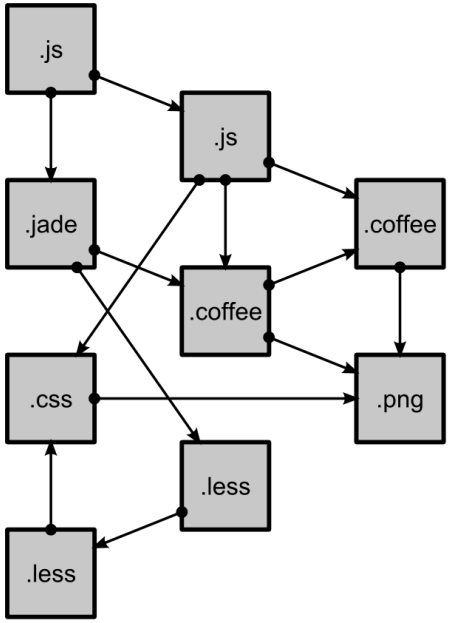

```
[tknew:~/Unitech/pm2] master(+84/-121)+* ± pm2 list
PM2 Process listing
```

App Name	id	mode	PID	status	Restarted	Uptime	memory	err logs
bashscript.sh	6	fork	8278	online	0	10s	1.379 MB	/home/tknew/.pm2/logs/bashscript.sh-err.log
checker	5	cluster	0	stopped	0	2m	0 B	/home/tknew/.pm2/logs/checker-err.log
interface-api	3	cluster	7526	online	0	3m	15.445 MB	/home/tknew/.pm2/logs/interface-api-err.log
interface-api	2	cluster	7517	online	0	3m	15.453 MB	/home/tknew/.pm2/logs/interface-api-err.log
interface-api	1	cluster	7512	online	0	3m	15.449 MB	/home/tknew/.pm2/logs/interface-api-err.log
interface-api	0	cluster	7507	online	0	3m	15.449 MB	/home/tknew/.pm2/logs/interface-api-err.log

知名工具框架

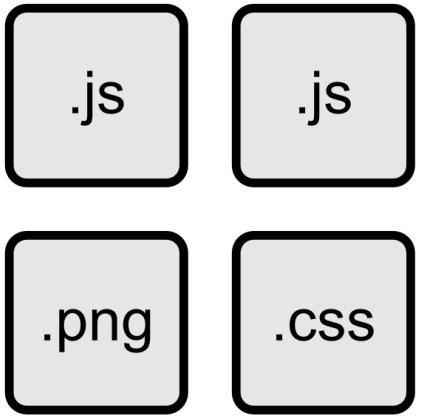
- Web 框架
  - Express
  - Koa
  - Egg.js
- 移动开发
  - React Native
  - Veex Rax
  - Ionic
- 桌面开发 Electron
- 语言扩展
  - TypeScript
  - CoffeeScript
  - Kotlin
- 测试 mocha
- 异步 co bluebird
- 构建工具 cnpm yarn

webpack



modules
with dependencies

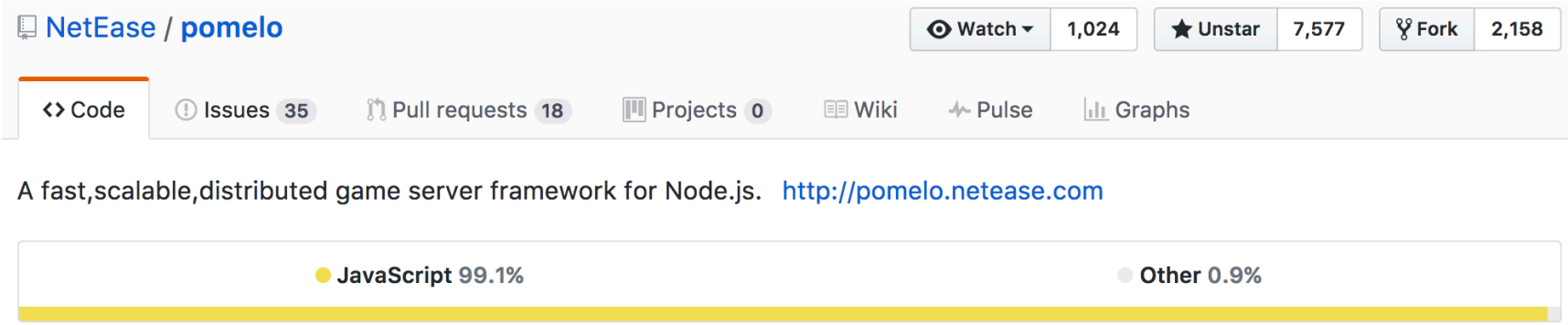
webpack
MODULE BUNDLER



static
assets

世界级公司与产品

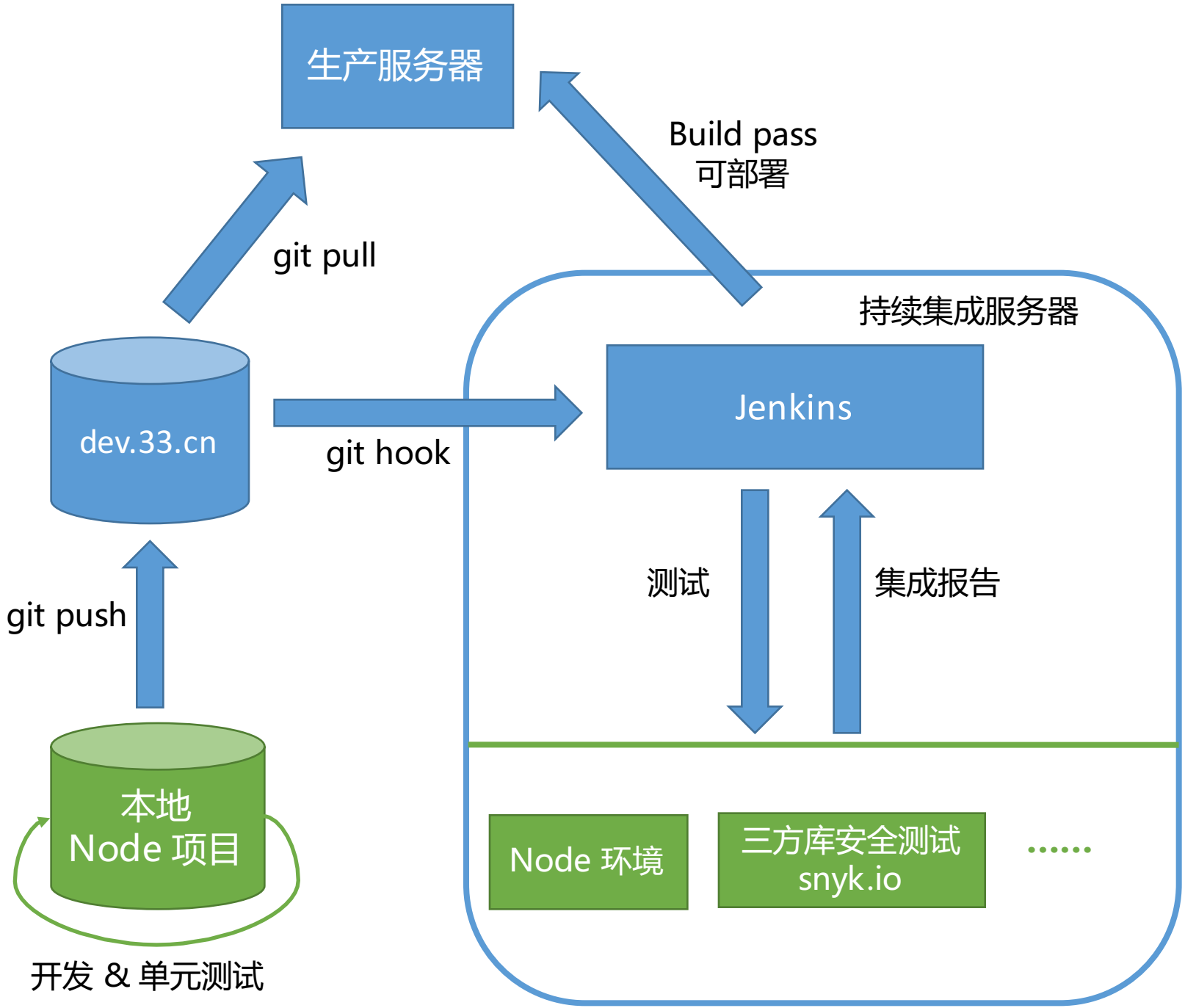
- 微软 TypeScript
- Google
- 英特尔（紫竹开发区）
- Paypal
- Netflix
- 阿里巴巴（阿里云，支付宝）
- 网易游戏



复杂美实践

- Node v8.x
- Egg.js
- Mocha
- PM2
- Keymetrics

- Vue
- ElementUI


生产服务器
Build pass
可部署
git pull
持续集成服务器
dev.33.cn
git hook
Jenkins
测试
集成报告
git push
本地
Node 项目
Node 环境
三方库安全测试
snyk.io
.....
开发 & 单元测试

Node 适用场景与技术选型

没有银弹，技术选型不能脱离业务场景

- 业务简单，计算量不大的后端
- 复杂后端的 Web API 层

- 高并发爬虫
  - 相比于 Python 爬虫优劣

- 命令行工具
  - 兼容性最好的脚本语言

- 美观且对性能要求不高的桌面应用
- 人力不足的移动开发团队

Q & A

Node.js 尝鲜派

Node.js 最新技术资讯

管理专栏

102 人关注

最新文章

Node 新增 ES6 Modules 支持，不用 Babel 实现 import

刚刚不久（9月8日凌晨），Node 项目 master 合并了实验性的 import 关键字支持，全称 ECMAScript Modules。新功能文档见：https://github.com/nodejs/node/blob/master/doc… 查看全文 >

雄霸 · 1 天前　　17 赞 · 5 条评论

Node.js 叒分家了，后续发版将受影响

本周，Node 社区的几个核心成员又从 Node.js master 分支 fork 出了一个项目，名叫 Ayo.js。事情是这样的：一个名叫 rvagg 的老哥是 Node 社区的 TSC（可以理解为常务委员… 查看全文 >

雄霸 · 15 天前　　67 赞 · 32 条评论