

alinode与Node APM

by @朴灵

自我介绍

- 田永强/朴灵
- 《深入浅出Node.js》作者，么么哒
- 开发专家@阿里云



大纲

- Node.js繁荣的现状与问题
- alinode与应用性能管理
- 性能案例分享
- 未来计划



Node.js社区正处于前所未有的繁荣中

Node.js基金会已经成立

- 脱离Joyent, 成立基金会
- Node.js项目从独裁者模式变为基金会模式
- 有正式的技术委员会(TSC)和管理委员会

Node.js基金会成员



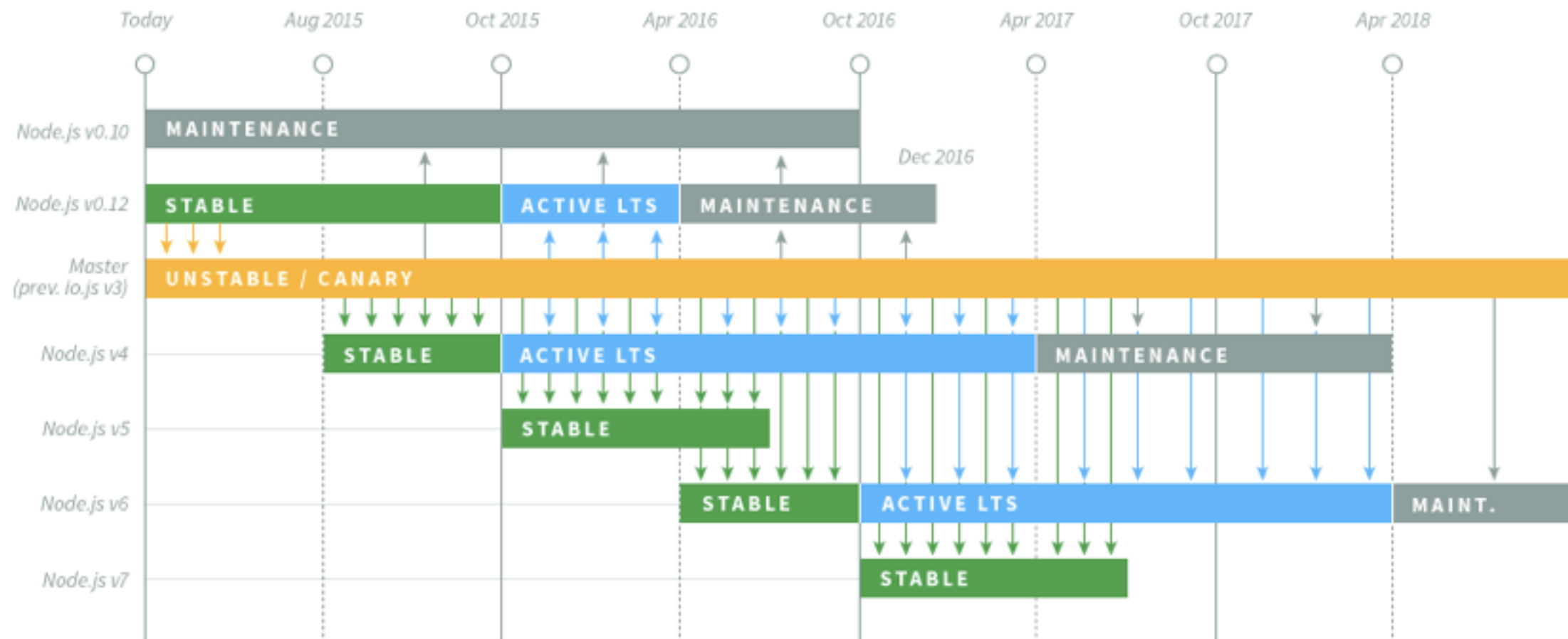
全面拥抱ES6规范

~~ECMAScript 6~~

ECMAScript 2015

提供LTS版本

Node.js Long Term Support Release Schedule



npm已经是世界上最大的模块仓库

- ~21万模块数量
- 每天：~4500万模块下载量
- npm, Inc.成立，提供企业仓库



然并卵



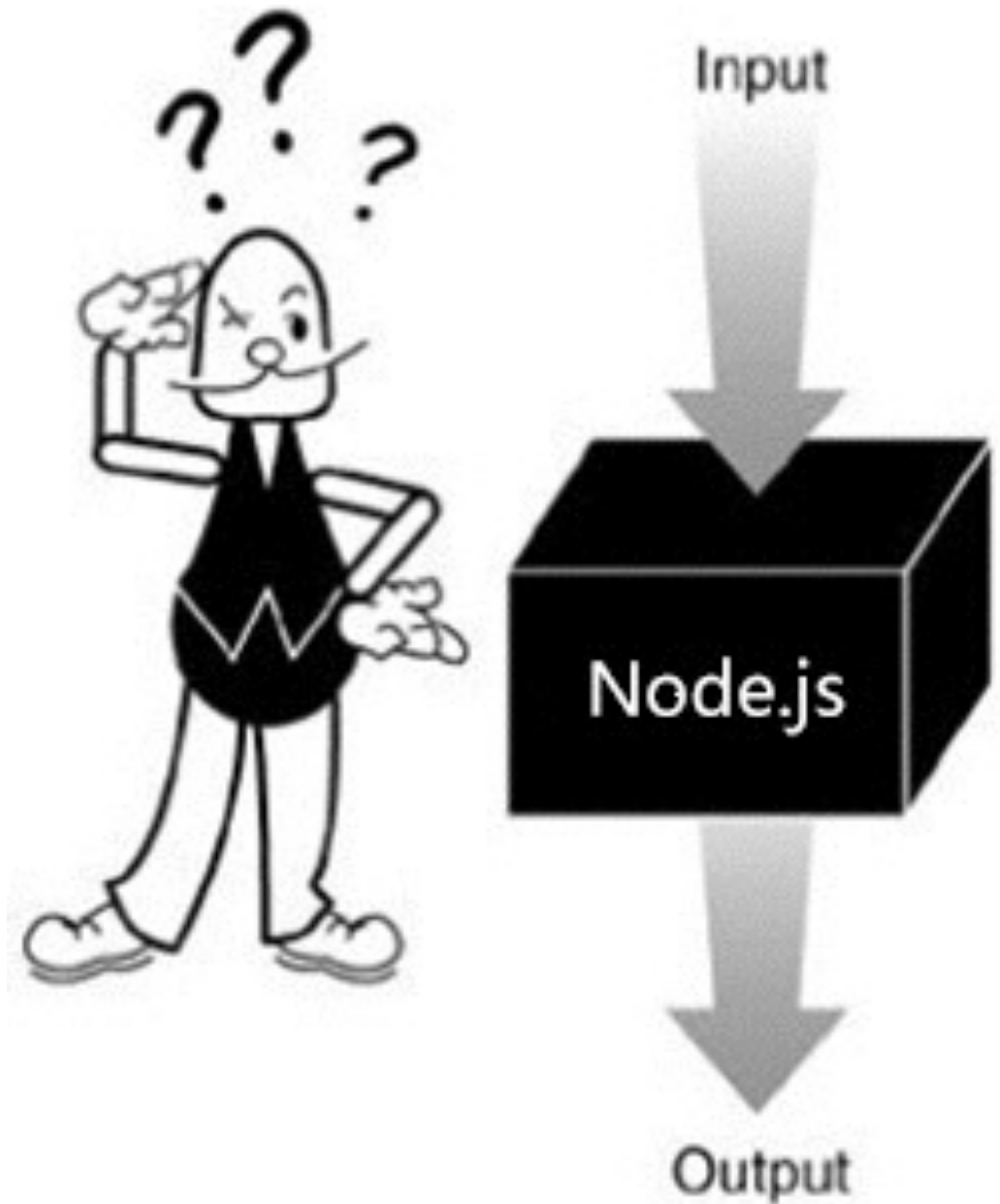
谁能hold住以下问题？

- CPU离奇飙高
- 内存无故泄漏
- GC频繁产生

Node.js现存的一些问题

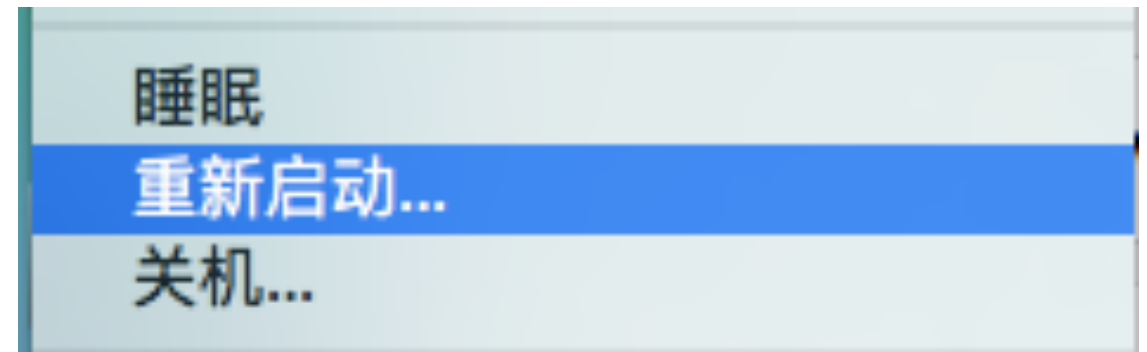
Node.js内核是一个黑盒子

- 缺少成熟的工具，无法协助查看内核状态
- 无法获知Node.js应用运行时的详细状态：堆的使用情况，找出热点函数，GC细节



缺少故障排查工具

- 没有线上问题排查工具
- 重启大法好!
- 线下是好的!
- 一些模块需要额外引入，但日常并不能启用



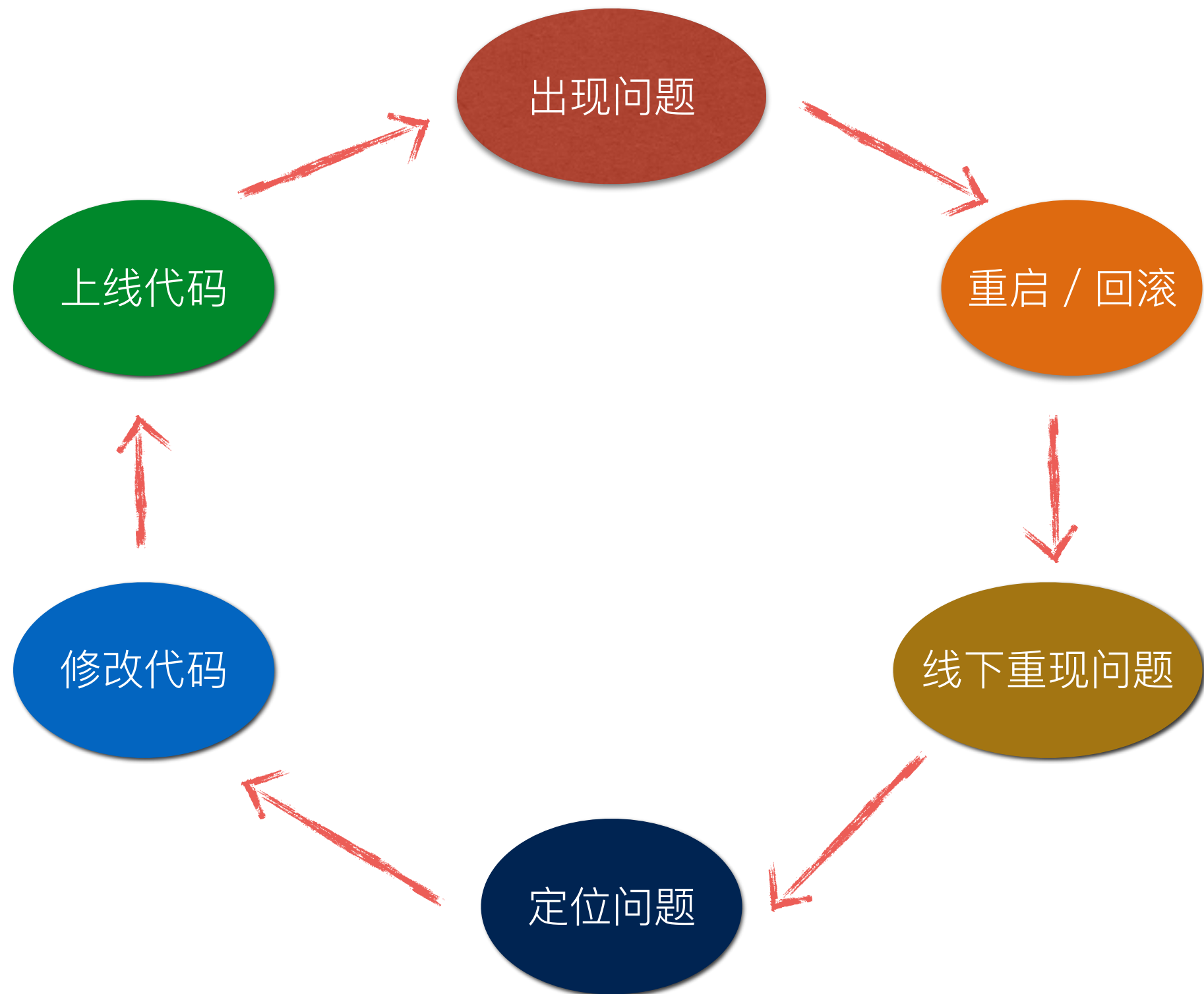
现有APM工具

- 现有的APM工具（new relic）会嵌入探针代码到应用程序
- 仅能针对到业务层问题，对内核无能为力
- 并不能真正、快速地定位和解决问题
- 并不一定好用

“bug，任何时候都要fix！不fix不行，你们想想，你带着老婆出了城，吃着火锅还唱着歌，突然就被bug劫了……所以，没有bug的日子才是好日子！”

—改编自《让子弹飞》

问题修复流程





alinode

~~a + linode~~ ali + node

<http://alinode.aliyun.com/>

alinode的解决方案

- 目标：过上没有Bug的日子
- 如何达成：
 - 过上阳光明媚的日子（让内核可监控）
 - 过上吃着火锅、唱着歌的日子（提供更好的工具）

修改内核

- 让Node.js的内核处于可监控的状态，得到更精细的监控数据
 - 进程CPU占用
 - 内存堆信息
 - GC花销
 - http详情
 - 定时器
 - 活跃句柄, `async_active_handles`
 - and more.



数据可视化

堆整体信息 数据说明 ?



堆空间组成 数据说明 ?

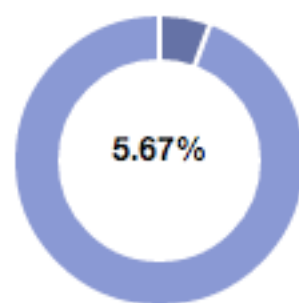
数据建议

健康态



code_space
lo_space

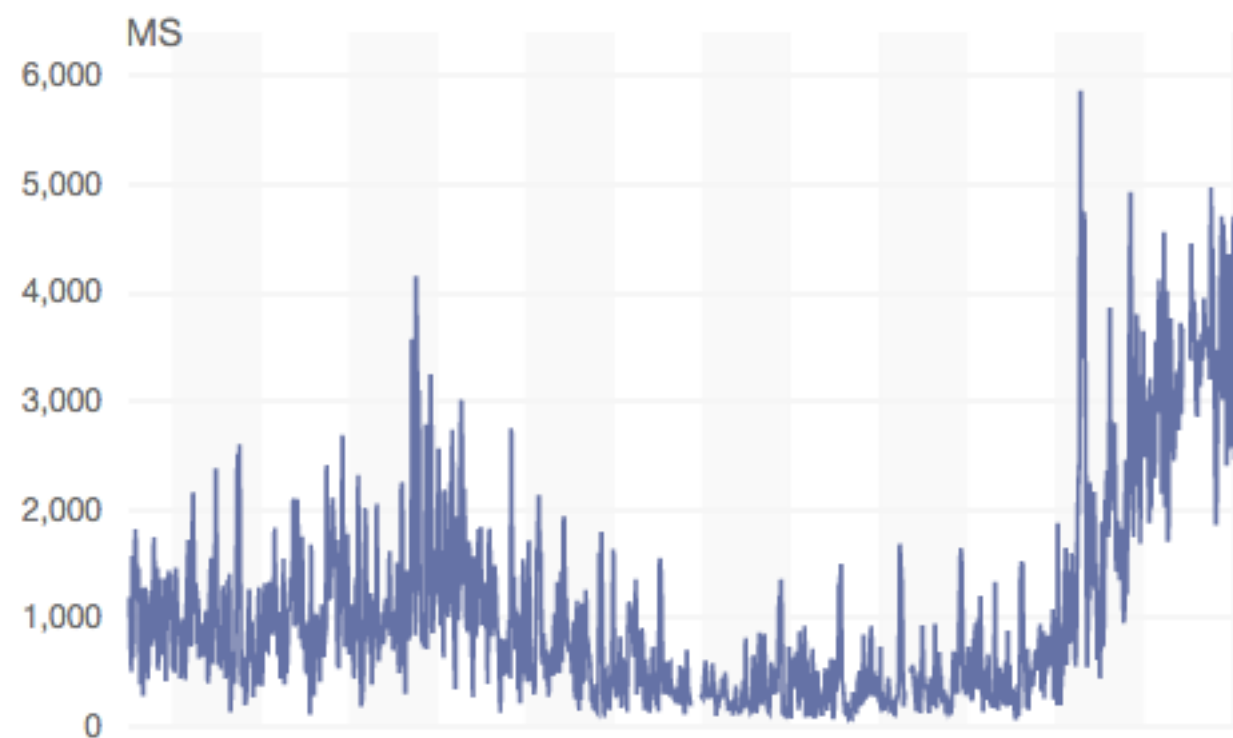
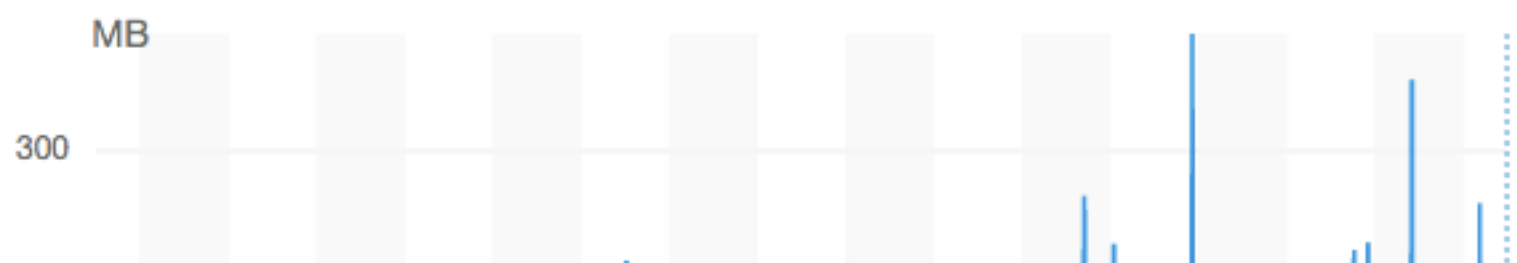
GC信息 数据说明 ?



gc_time_during_last_min
last_min

数据建议 建议说明 ?

不健康态



报警系统

- 集成邮件、短信、电话（实现中）
- 实时推送报警信息到开发者手中（TODO）



在线精准捕获问题

- 抓取堆快照
- CPU Profiling
- 详细GC追踪



案例分享：CPU热点

- 案例介绍：npm.taobao.org是阿里巴巴基于开源的cnpm搭建的npm镜像网站
- 日均峰值QPS为~200。

案例分享： `html minify`

案例分享：模版渲染

- 优化成果：250% -> 150%
- 节省N台服务器 * 1个CPU
- 阿里云CPU价格：63元/月
- 更快的买买买

案例分享：内存泄漏

```
1  var koa = require('koa');
2  var app = koa();
3
4  var session = require('koa-generic-session');
5
6  app.keys = ['im a newer secret', 'i like alinode'];
7
8  app.use(session());
9
10 app.use(function *(){
11   this.session.count = (this.session.count || 0) + 1;
12   this.body = '访问次数: ' + this.session.count + '\n';
13 });
14
15 app.listen(3000);
16
17 console.log("Server started at 3000");
18
```

内存泄漏分析

内存泄漏报表

类视图

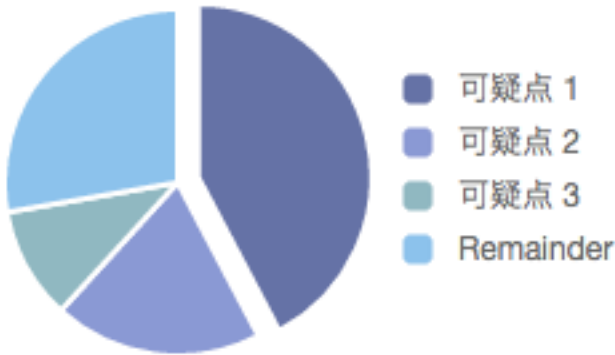
对象簇视图

搜索结果 ✕

可疑点 1

"(context)"实例"@92627"占用了6,039,880 (42.29%)字节.其内存主要积累在"Object"实例"@95229".

关键字
(context)
Object

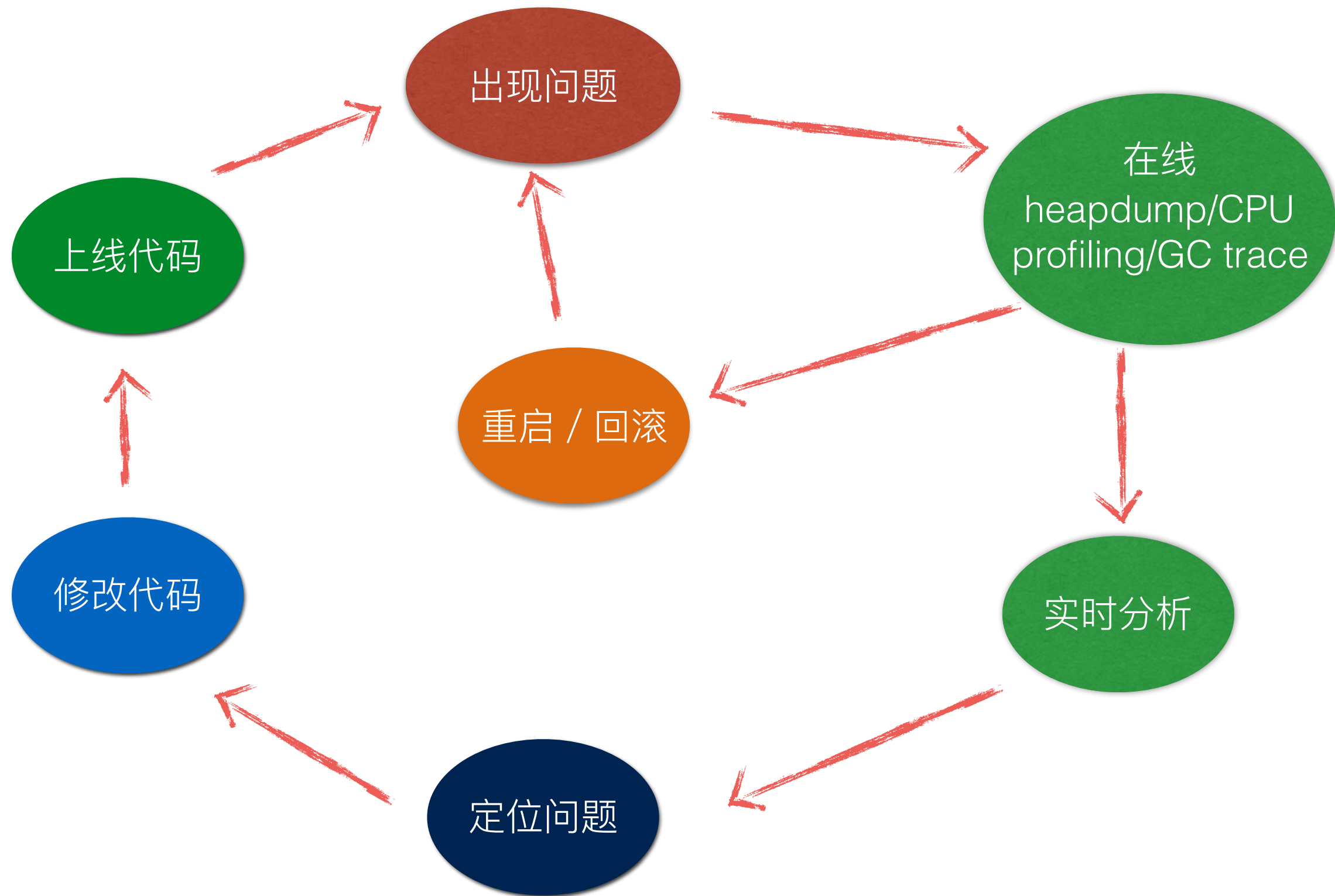


Retained size 比例

依赖路径

Class Name	ShallowHeap	Retained Heap
▼ Object @95229	24	6037976
▼ module.exports @94275	96	6038072
▼ Store @94273	104	6038704
▶ (context) of function module.exports() / session.js @92627	184	6039880

alinode的问题解决流程



超越APM

异常日志

ReferenceError at 2015-12-07 13:38:06

堆栈

```
ReferenceError: res is not defined
    at Promise.<anonymous> (/path/to/routes/admin/pm_talent_develope.js:2387:13)
    at Promise.<anonymous> (/path/to/node_modules/mongoose/node_modules/mpromise/lib/promise.js:177:8)
    at Promise.emit (events.js:107:17)
    at Promise.emit (/path/to/node_modules/mongoose/node_modules/mpromise/lib/promise.js:84:38)
    at Promise.fulfill (/path/to/node_modules/mongoose/node_modules/mpromise/lib/promise.js:97:20)
    at Promise.<anonymous> (/path/to/node_modules/mongoose/lib/model.js:1544:35)
    at Promise.<anonymous> (/path/to/node_modules/mongoose/node_modules/mpromise/lib/promise.js:177:8)
    at Promise.emit (events.js:110:17)
    at Promise.emit (                                e.js:84:38)
    at Promise.fulfill                                mise.js:97:20)
```

修复建议

1. 引用错误，请立即修复
2. 该错误常见于对未定义的变量进行了使用

安全提醒



您的Node版本: v4.2.2处于风险[CVE-2015-8027](#)之中, 请立即更新到安全的版本。

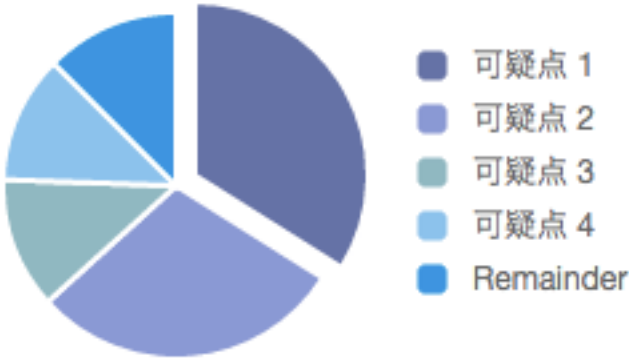
更好的分析工具

内存泄漏报表

可疑点 1

"Timer"实例"@797"占用了2,548,360 (33.93%)字节.其内存主要积累在"Timer"实例"@797".

关键字
Timer



Retained size 比例

依赖路径

Class Name	ShallowHeap	Retained Heap
Timer @797	248	2548360

alinode优势

- 完全兼容Node，无需任何代码修改
- 不侵入任何业务代码
- 不带来性能损失

alinode优势

- 实时捕获异常情况，实时分析
- 精准定位问题，快速解决故障

内部客户

- wormhole
- 收藏夹
- UC Web
- etc.

外部客户

- CNPM
- cnodejs.org
- teambition
- 智思云
- 大搜车
- etc.

未来计划

- 免费提供给开源 / 公益项目
- 加入Node.js基金会
- 成为node core contributor
- 推动Node的稳定性、可运维性建设

process.exit(0);

<http://alinode.aliyun.com/>



女儿怎么不用Node



搞不定线上问题怎敢用



为毛不用alinode





乖女儿你在干嘛



我在重现线上Bug



为毛不用alinode

