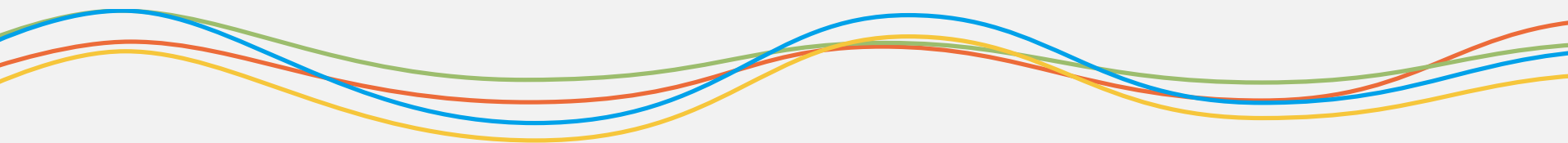


爱奇艺容器云平台性能分析实践

周海维 赵慰



大纲

- PaaS @ 爱奇艺
- 一个 fork 引发的血案
- 一个容器间干扰的性能故障分析
- 容器云平台展望

PaaS @ 爱奇艺

爱奇艺容器云平台简介



- 2014 - 今
- Mesos + Docker
- 节点数量：20 -> 200 -> ~2000
- 业务类型：离线作业 -> 在线服务 -> 任务混布？





PaaS @ 爱奇艺 —— 资源管理



- 节点
 - 离线作业/在线服务
 - 某些超重点业务
 - 托管业务
- 数据中心
 - 外网出口敏感
 - 数据就近访问
- 容器
 - Docker/Mesos



PaaS @ 爱奇艺 —— 应用管理



- 运行时
 - 健康检查
 - 常规/自定义监控指标
 - 平均数/中位数/偏移量
- 升级过程
 - 预先下线
 - 延迟上线
 - 应用预热

一个 fork 引发的血案

山雨欲来

- 用户群
 - 用户甲：我的容器杀不掉，帮看下
 - 用户乙：我的容器一直是启动中状态，什么情况
 - 用户丙：我这儿控制台没响应了
 - 用户丁：我的服务一堆超时报警，怎么回事
 - DevOps A：稳住，我们能赢.....
- 开发群
 - DevOps A：估计 Docker daemon 又出问题了吧
 - DevOps B：今天挂的频率有点高
 - DevOps A：我去瞅一眼.....

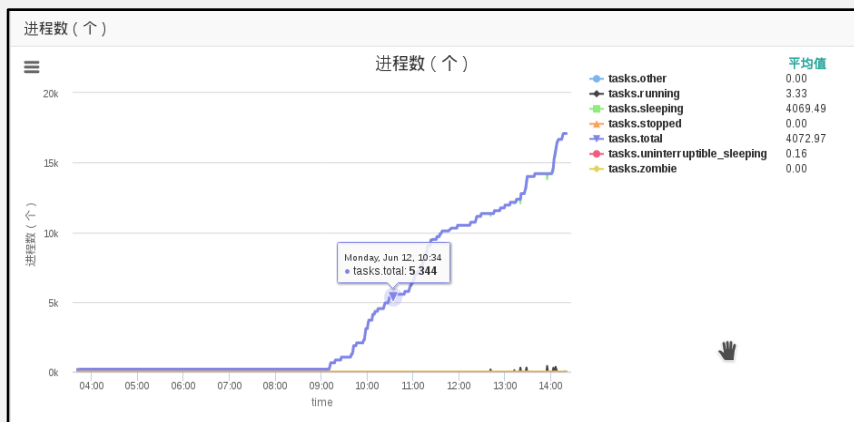
Cannot allocate memory !



```
bash: fork: Cannot allocate memory
-bash# free
bash: fork: Cannot allocate memory
-bash# ps
bash: fork: Cannot allocate memory
-bash# reboot
bash: fork: Cannot allocate memory
-bash# wtf
bash: wtf: command not found...
```

定位问题

- What
 - `Cannot allocate memory` -> fork 炸弹
- When
 - 突发 -> 近期上线/升级业务
- Where
 - 故障节点 -> APP 实例列表 -> 嫌疑 APP
- Who
 - 嫌疑 APP -> 监控历史数据 (进程数) -> 确认嫌疑



定位问题

- How
 - 进程如何暴走？是不是正常行为？
 - tips：特权模式启动新容器，共享问题容器命名空间

```
java 27272 root *717r FIFO 0,8 0t0 379630438 pipe
java 27272 root *718w FIFO 0,8 0t0 379630438 pipe
java 27272 root *719u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *720r FIFO 0,8 0t0 379630439 pipe
java 27272 root *721w FIFO 0,8 0t0 379630439 pipe
java 27272 root *722u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *723r FIFO 0,8 0t0 379630440 pipe
java 27272 root *724w FIFO 0,8 0t0 379630440 pipe
java 27272 root *725u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *726r FIFO 0,8 0t0 379630441 pipe
java 27272 root *727w FIFO 0,8 0t0 379630441 pipe
java 27272 root *728u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *729r FIFO 0,8 0t0 379630442 pipe
java 27272 root *730w FIFO 0,8 0t0 379630442 pipe
java 27272 root *731u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *732r FIFO 0,8 0t0 379630443 pipe
java 27272 root *733w FIFO 0,8 0t0 379630443 pipe
java 27272 root *734u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *735r FIFO 0,8 0t0 379630444 pipe
java 27272 root *736w FIFO 0,8 0t0 379630444 pipe
java 27272 root *737u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *738r FIFO 0,8 0t0 379630445 pipe
java 27272 root *739w FIFO 0,8 0t0 379630445 pipe
java 27272 root *740u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *741r FIFO 0,8 0t0 379630446 pipe
java 27272 root *742w FIFO 0,8 0t0 379630446 pipe
java 27272 root *743u a_inode 0,9 0 7115 [eventpoll]
java 27272 root *744r FIFO 0,8 0t0 379630447 pipe
java 27272 root *745w FIFO 0,8 0t0 379630447 pipe
java 27272 root *746u a_inode 0,9 0 7115 [eventpoll]
```

```
epoll_wait(xx, {}, xx, 1000) = 0
epoll_wait(xx, {}, xx, 1000) = 0
epoll_wait(xx, {}, xx, 1000) = 0
epoll_wait(xx, {}, xx, 1000) = 0
epoll_wait(xx, {}, xx, 1000) = 0
epoll_wait(xx, {}, xx, 1000) = 0
```

追根溯源



- 与应用开发者合作
 - jstack、jmap 等工具

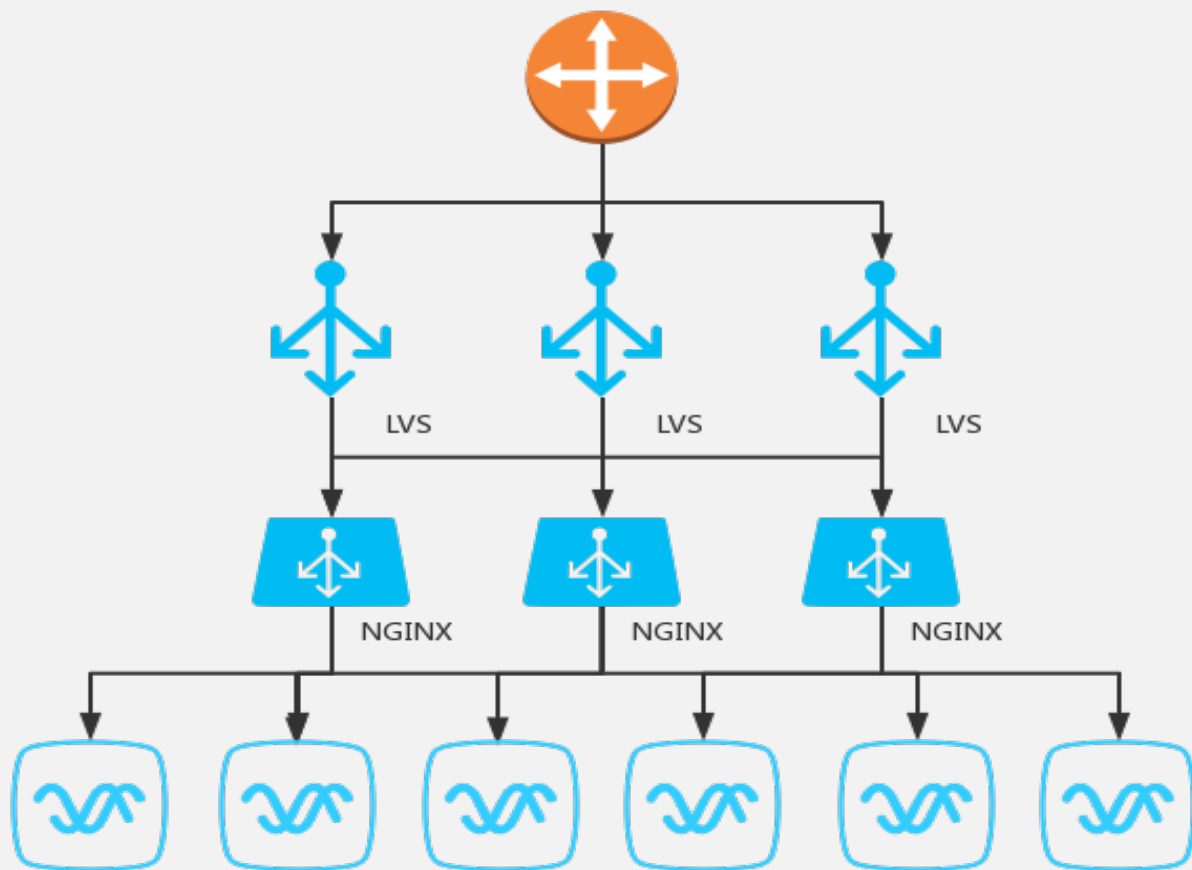
```
"I/O dispatcher 46744" prio=10 tid=0x00007fb27d721000 nid=0xbd56 runnable [0x00007fb275695000]
  java.lang.Thread.State: RUNNABLE
    at sun.nio.ch.EPollArrayWrapper.epollWait(Native Method)
    at sun.nio.ch.EPollArrayWrapper.poll(EPollArrayWrapper.java:269)
    at sun.nio.ch.EPollSelectorImpl.doSelect(EPollSelectorImpl.java:79)
    at sun.nio.ch.SelectorImpl.lockAndDoSelect(SelectorImpl.java:87)
    - locked <0x0000000694790e58> (a sun.nio.ch.Util$2)
    - locked <0x0000000694790e48> (a java.util.Collections$UnmodifiableSet)
    - locked <0x0000000694790e68> (a sun.nio.ch.EPollSelectorImpl)
    at sun.nio.ch.SelectorImpl.select(SelectorImpl.java:98)
    at org.apache.http.impl.nio.reactor.AbstractIOReactor.execute(AbstractIOReactor.java:257)
    at org.apache.http.impl.nio.reactor.BaseIOReactor.execute(BaseIOReactor.java:106)
    at org.apache.http.impl.nio.reactor.AbstractMultiworkerIOReactor$Worker.run(AbstractMultiworkerIOReactor.java:590)
    at java.lang.Thread.run(Thread.java:744)

Locked ownable synchronizers:
- None
```

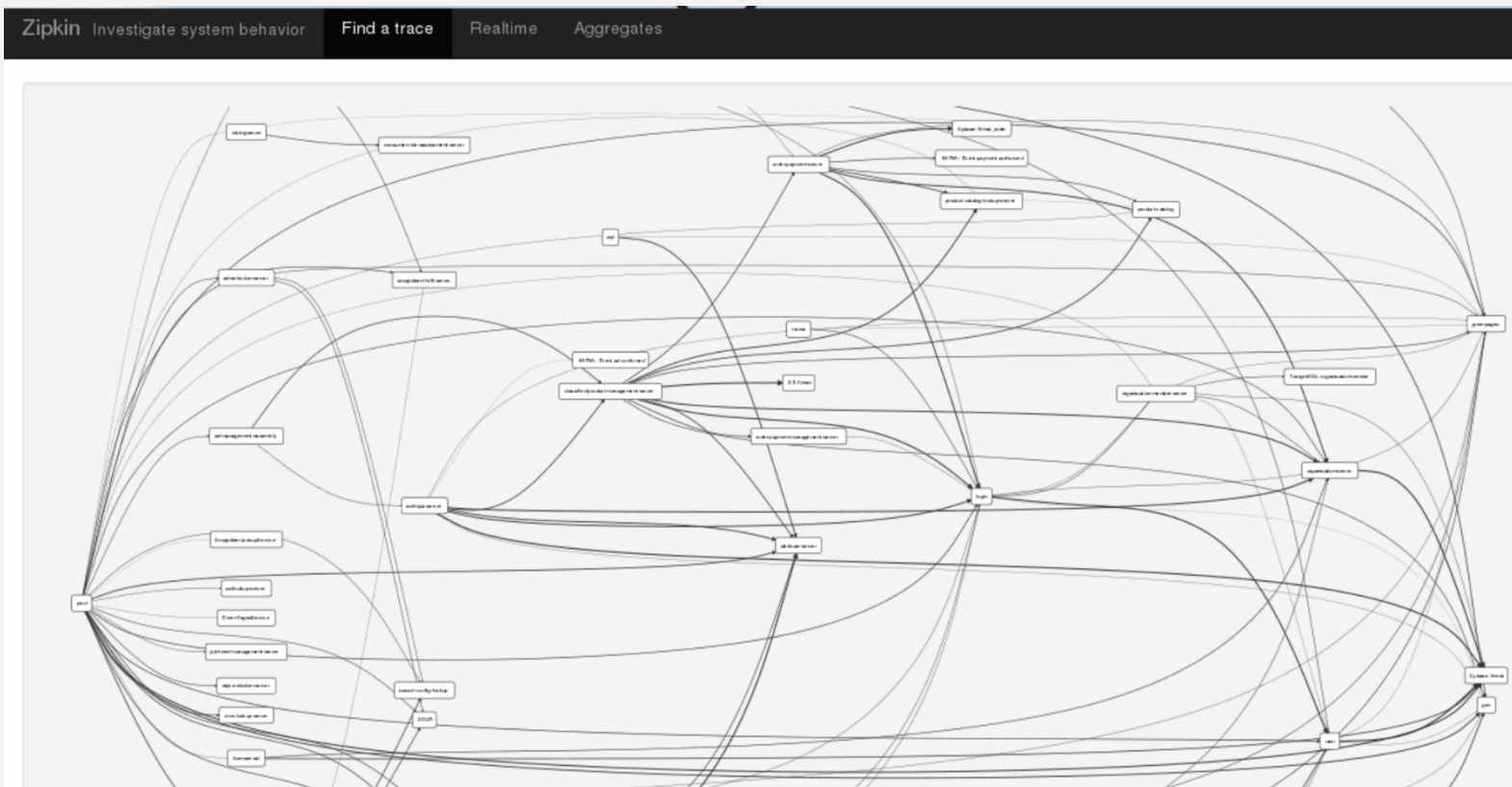
- 故障原因
 - 容器隔离先天不足
 - 健康检查与实际服务接口不一致
 - 上线流程不完善，缺少生产环境模拟
 - 代码质量需要更严格把关，从源头降低意外发生概率
- 排障思考
 - 容器粒度小数量大，难以定位，需要统计数据协助预警
 - 容器内权限不足，影响应用开发者 debug
 - 监控指标尽可能详尽，宁滥毋缺

一个容器间干扰的性能故障分析

在线服务示意图



微服务间复杂关系



微服务容器的性能故障

- 有可能的故障点增多
- 问题扩散，报障点可能不是第一现场
- 可靠性问题与性能问题相互关联
- 容器名字空间使得问题定位更难
- 有时还需要考虑容器之间相互干扰
- “原来”（的性能）是好的
- 在容器内难以找到答案，多数问题推给计算平台调查



用户报障

- 升级过程中，某业务的 nginx 突然出现大量 Code 499



6个问题

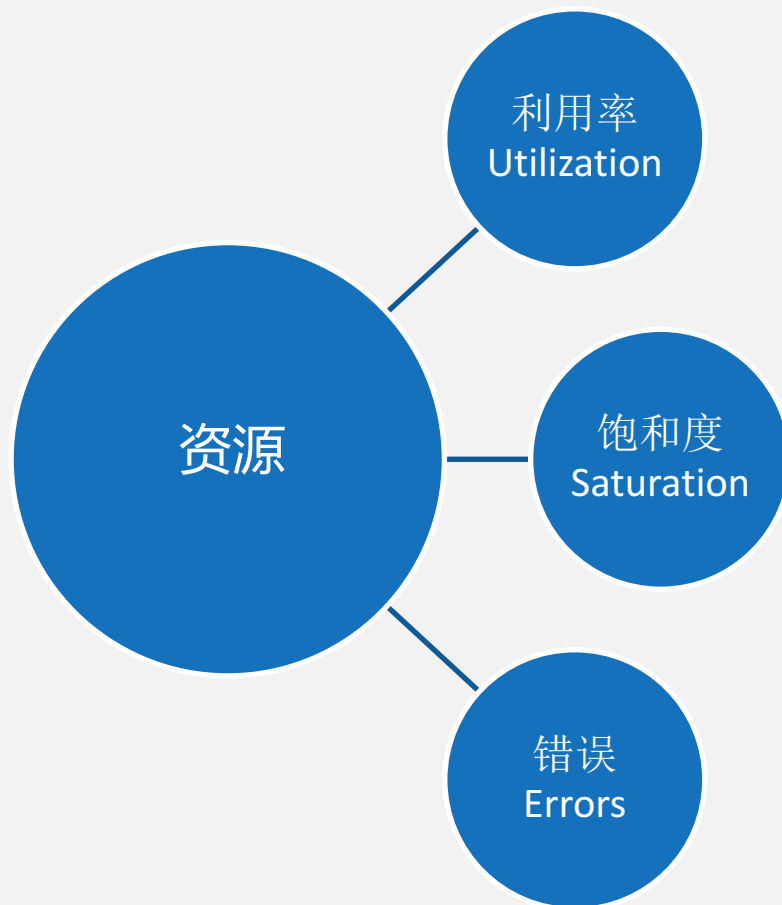
- 什么让你认为这是个性能问题？
- 以前这个系统性能好吗？
- 最近什么改变了？软件？硬件？负载？
- 问题可以用延迟和运行时的措辞表达吗
- 问题影响到其他人或者其他应用吗？只有你遇到了吗？
- 环境是什么？使用了什么软件和硬件？版本？配置？



问题描述

- Q: 什么让你认为这是个（容器）性能问题？
A: 499表示客户端主动关闭连接，意味着大量请求处理超时。
- Q: 以前这个系统性能正常/好吗？
A: 正常运行没有出现 Code 499，异常只出现升级过程中。
- Q: 最近什么改变了？软件？硬件？负载？
A: 运行环境从虚拟机变成容器。
- Q: 问题可以用延迟和运行时的措辞表达吗？
A: 该业务某时刻的大量请求处理超过客户端允许时间。
- Q: 问题影响到其他人或者其他应用吗？只有你遇到了吗？
A: 其他相关业务也受到影响。
- Q: 环境是什么？使用了什么软件和硬件？版本？配置？
A: OpenJDK 1.7 / CentOS 7.2 / Kernel 3.10 / E5 2650 v3。

USE方法论



<http://www.brendangregg.com/usemethod.html>

一分钟调查Linux性能问题的十个命令



- uptime
- dmesg | tail
- vmstat 1
- mpstat -P ALL 1
- pidstat 1
- iostat -xz 1
- free -m
- sar -n DEV 1
- sar -n TCP,ETCP 1
- top

vmstat 1



procs		-----memory-----				---swap--		-----io----		-system--		-----cpu-----					
r	b	swpd	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id	wa	st	
10	0	39480	74939928	450720	95410240	0	0	0	5588	162083	194992	20	7	72	0	0	2017-05-08 13:49:31
10	0	39480	74910624	450720	95422704	0	0	0	17102	165795	198209	20	7	72	1	0	2017-05-08 13:49:32
22	0	39480	74889456	450720	95425072	0	0	0	0	163656	197216	19	8	73	0	0	2017-05-08 13:49:33
12	0	39480	74861792	450720	95426176	0	0	0	0	172461	199134	21	8	71	0	0	2017-05-08 13:49:34
1210	0	39480	75401664	450720	94904992	0	0	0	0	91408	78352	7	73	20	0	0	2017-05-08 13:49:35
1270	0	39480	75266440	450720	94050160	0	0	0	72	47257	19045	1	99	0	0	0	2017-05-08 13:49:36
564	0	39480	78699280	450720	93275224	0	0	0	25460	51805	23883	2	98	0	0	0	2017-05-08 13:49:37
8	1	39480	81399520	450720	92553184	0	0	0	5092	106143	87193	11	76	12	0	0	2017-05-08 13:49:38
10	1	39480	81541232	450720	92396352	0	0	0	6103	163024	203121	20	8	71	2	0	2017-05-08 13:49:39
14	0	39480	81692104	450720	92247176	0	0	0	1259	156011	196875	18	7	74	1	0	2017-05-08 13:49:40
11	0	39480	81684552	450720	92247600	0	0	0	204	163274	199978	20	7	73	0	0	2017-05-08 13:49:41
14	0	39480	81666992	450720	92251632	0	0	0	5156	165260	197309	19	7	73	1	0	2017-05-08 13:49:42
8	0	39480	81647416	450720	92253536	0	0	0	0	163376	198234	20	7	73	0	0	2017-05-08 13:49:43
16	0	39480	81626528	450720	92255728	0	0	0	0	158819	196952	19	7	74	0	0	2017-05-08 13:49:44

Runnable

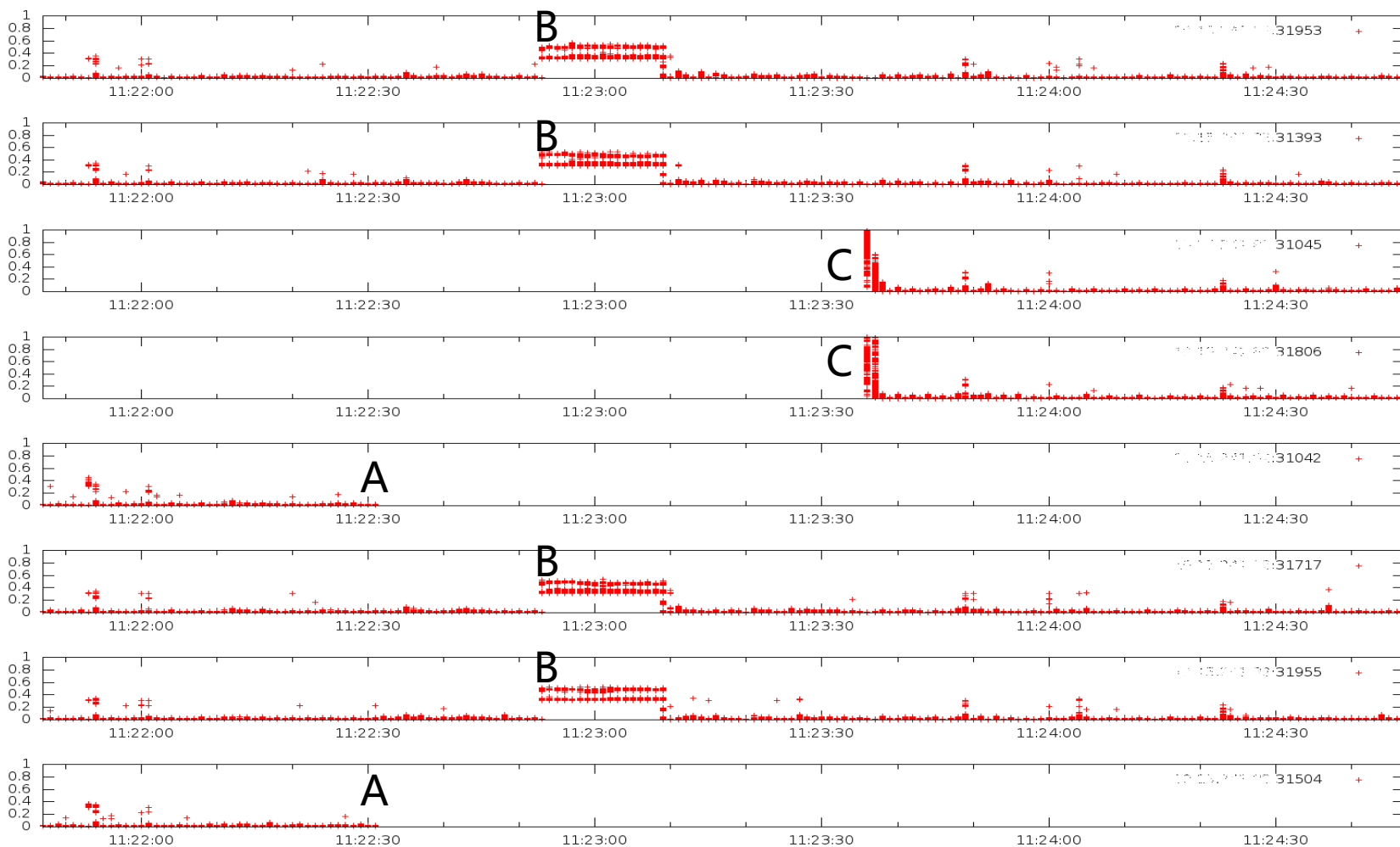
SYS

CPU 资源使用异常

- Runnable 进程异常多
- sys 利用率很高
- 日志中没有发现错误信息

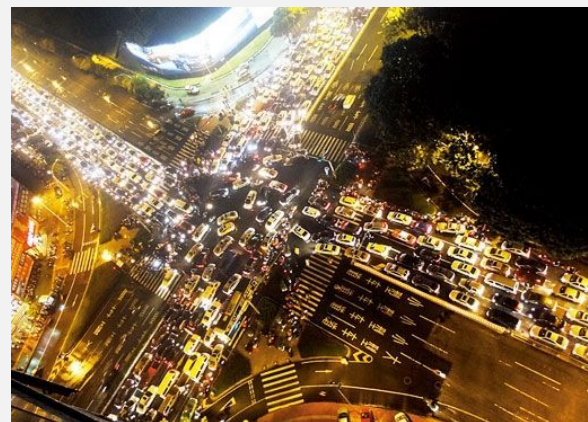


什么场景容易复现问题？

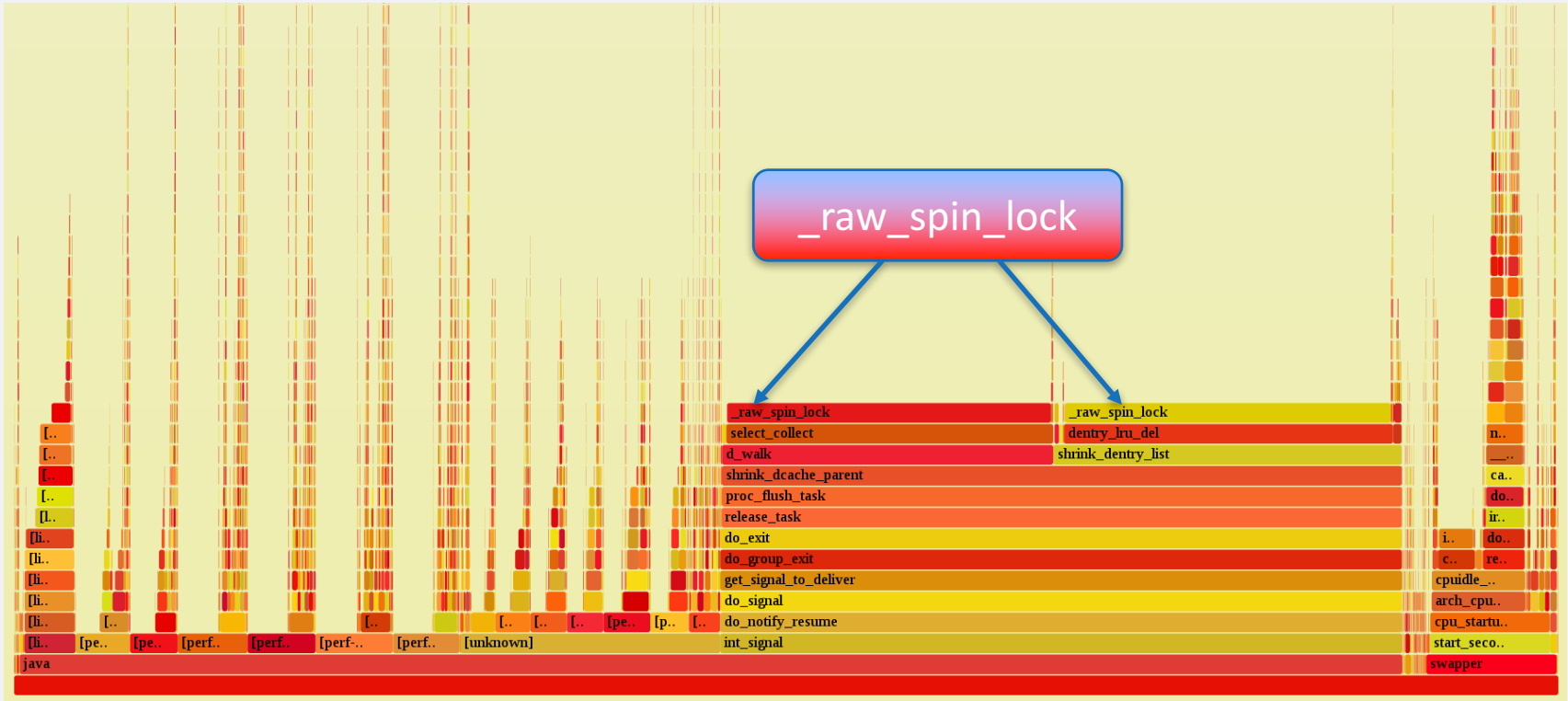


复现场景

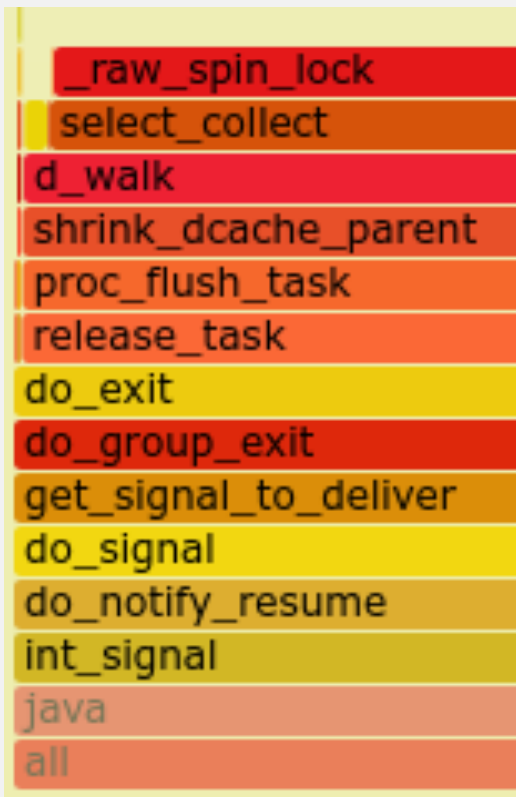
- 正在退出的容器干扰其他容器运行
- 实际重现概率比较低 1 ~ 2%
- 有两个容器同时退出时每次都复现



FlameGraph



FlameGraph (2)



获取 spin lock

清理 dcache

Task 退出

处理 SIG_TERM



科学方法论



- 提问
- 假设
- 推测
- 检验
- 分析

提问

• 为什么 perf 结果表示大量CPU时间处理 spin lock ?

假设

• 大量线程退出造成内核工作异常。

推测

• Spin lock 相关的问题是严重的，最新内核可能修复这个 bug。

检验

• 升级到内核 4.10，问题消失。

分析

• 大量正在退出的线程竞争同一个 spin lock，引发内核 bug。

近期没有内核更新的计划



是对容器线程数
做硬限制？

还是写个文档告诉用
户不要启动太多线程？

CPUSET workaround



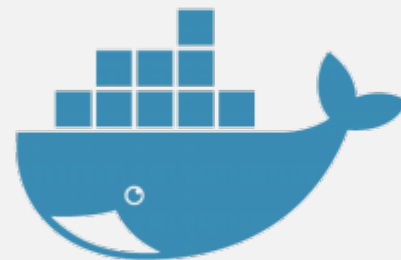
监控工具改进



- 秒级性能数据采集和就地处理，异常事件上报
- 异常时刻 Profiling 自动报告
- 容器内部自助 Profiling 工具集成



- 应用程序根据资源检测结果，进行初始化
拦截容器内检测资源接口，提供容器的实际分配资源数量
- 容器的隔离性比虚拟机弱
对同一宿主机内容器的监控项进行关联和预警
- 线上服务程序，需要平台提供预热方案



容器云平台展望



- Docker 容器迁移到 Mesos 容器
- 从 Device Mapper 迁移到 OverlayFS
- 在线离线任务混合部署
- 集群空余资源 oversubscription
- GPU/FPGA
- 混合云

THANK YOU

