阿里云 | 奥运会全球指定云服务商

systemd稳定性增强

文洋

阿里云-操作系统

2020-10-12

目录

阿里云 | 奥运会全球指定云服务商

问题背景

生产环境中的各种问题

- **集群抖动**

```
64 bytes from 11.80.241.203: icmp_seq=2444 ttl=61 time=0.251 ms
64 bytes from 11.80.241.203: icmp_seq=2445 ttl=61 time=0.253 ms
64 bytes from 11.80.241.203: icmp_seq=2446 ttl=61 time=0.243 ms
64 bytes from 11.80.241.203: icmp_seq=2447 ttl=61 time=31.2 ms
64 bytes from 11.80.241.203: icmp_seq=2448 ttl=61 time=0.239 ms
64 bytes from 11.80.241.203: icmp_seq=2449 ttl=61 time=0.270 ms
64 bytes from 11.80.241.203: icmp_seq=2450 ttl=61 time=0.239 ms
```

- 现象：集群中的若干台机器都有这个现象，在压力测试测下，超时率指标上涨，业务性能受损，影响用户体验。
- 原因：systemd把容器的mount传播到/sys/fs/cgroup/{cpu,memory…}等目录下,创建出成千上万的空cgroup。

- **容器启动失败**

```
$sudo pouch exec -it 0d52ae3d976cd62a1a9cf9c9aac5210822b662236de2eb51078c45354680f025 /bin/bash

[root@0d52ae3d976c /]
#ps -ef
UID         PID   PPID  C STIME TTY          TIME CMD
root          1      0  0 14:47 ?        00:00:00 /sbin/init
root         21      0  0 14:47 pts/0    00:00:00 /bin/bash
root         56     21  0 14:47 pts/0    00:00:00 ps -ef

[root@0d52ae3d976c /]
#systemctl status systemd
Failed to get D-Bus connection: Operation not permitted

[root@0d52ae3d976c /]
#uname -a
Linux 0d52ae3d976c 4.9.168-018.ali3000.alios7.x86_64 #1 SMP Fri Mar 13 17:30:54 CST 2020 x86_64

[root@0d52ae3d976c /]
#busctl
Failed to connect to bus: No such file or directory

[root@0d52ae3d976c /]
#journalctl -b
No journal files were found.
```

- 现象：在某些机型上，容器启动失败，无法部署业务。而同样的容器在其他机型上是可以正常启动的。
- 原因： systemd使用container环境变量来检测容器环境。容器没有设置container环境变量，触发条件是这些机型host内核的smack功能被打开了。

问题背景

生产环境中的各种问题

阿里云 | 奥运会全球指定云服务商

- 内存泄漏

```
top - 11:07:44 up 248 days, 12:55,  1 user,  load average: 11.76, 12.76, 13.65
Tasks: 1884 total,   3 running, 1862 sleeping,   0 stopped,  19 zombie
%Cpu(s):  3.6 us,  4.8 sy,  0.0 ni, 91.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 39514012+total, 11697444 free, 33329574+used, 50146948 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 33052876 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
127240 root      20   0  244.7g 243.9g   1476 R 100.0 64.7   1:21.61 systemd
246566 root      20   0   14.5g 537568  27316 S  87.2  0.1  38527:57 kubelet
231168 root      20   0       4      4      0 R  68.4  0.0   0:02.08 systemd
 36027 root      16  -4 2225288 514068   6728 S  66.1  0.1 211956:30 h2o
221227 root      20   0       0      0      0 Z  30.9  0.0   0:08.28 lvm2-activation
```

- 现象：在某些长期运行的机器上，systemd内存泄漏严重，导致业务进程不断被OOM杀掉。
- 原因：systemd的bug。

- CPU冲高

```
top - 14:07:28 up 222 days,  2:54,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 154 total,   2 running, 152 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.4 us,  0.2 sy,  0.0 ni, 99.4 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 16267912 total,  1343996 free,  2765184 used, 12158732 buff/cache
KiB Swap:        0 total,        0 free,        0 used. 12572036 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
12156 root      20   0  278672 157796   1396 R 100.0  1.0   0:03.14 systemd-logind
```

- 现象：systemd的进程打满CPU，影响业务。
- 原因：systemd的bug。

问题背景

生产环境中的各种问题

阿里云 | 奥运会全球指定云服务商

● 机器无法启动

```
[    3.054638] type=1305 audit(1589698206.765:3): audit_pid=462 old=0 auid=4294967295 ses=4294967295 res=1
[  OK  ] Stopped Crash recovery kernel arming.
         Starting Crash recovery kernel arming...              a     0
[    3.080115] ppdev: user-space parallel port driver
[  OK  ] Started Security Auditing Service.
         Starting Update UTMP about System Boot/Shutdown...    p     p
[  OK  ] Started Update UTMP about System Boot/Shutdown.
         Starting Update UTMP about System Runlevel Changes...
[  OK  ] Started Update UTMP about System Runlevel Changes.
[  OK  ]154638led Crash recovery kernel arming.               2 old=0 auid=4294967295 ses=4294967295 res=1
[    6.243728] AES CTR mode by8 optimization enabled
[    6.247525] alg: No test for __gcm-aes-aesni (__driver-gcm-aes-aesni)
[    6.252479] alg: No test for crc32 (crc32-pclmul)
[    6.261757] intel_rapl: no valid rapl domains found in package 0
[    6.264623] intel_powerclamp: No package C-state availableWelcome to emergency mode! After logging in, type
to view       e
system logs, "systemctl reboot" to reboot, "systemctl default" or ^D to
try again to boot into default mode.
Give root password for maintenance
(or press Control-D to continue):
```

- 现象：系统启动失败，机器进入紧急模式，无法连接网络，机器不可用，需要人工介入，无法进行大规模运维。
- 原因：由于时序原因，当initramfs中的systemd和大系统中的systemd版本不一致时， switch root可能会失败。

● 僵死进程累积

```
root     32278  0.0  0.0      0     0 ?        Z    05:39   0:00 [runuser] <defunct>
root     32280  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32304  0.0  0.0      0     0 ?        Z    Aug21   0:00 [runuser] <defunct>
root     32362  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32448  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32475  0.0  0.0      0     0 ?        Z    Aug20   0:00 [runuser] <defunct>
root     32498  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32509  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
root     32551  0.0  0.0      0     0 ?        Z    09:54   0:00 [runuser] <defunct>
root     32556  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
root     32557  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
root     32611  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32612  0.0  0.0      0     0 ?        Z    Aug22   0:00 [runuser] <defunct>
root     32647  0.0  0.0      0     0 ?        Z    Aug19   0:00 [runuser] <defunct>
root     32699  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
root     32700  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
root     32716  0.0  0.0      0     0 ?        Z    Aug19   0:00 [runuser] <defunct>
root     32753  0.0  0.0      0     0 ?        Z    Aug18   0:00 [runuser] <defunct>
```

- 现象：僵死进程大量累积，影响到了业务。
- 原因：systemd crash，进入freeze状态，无法回收僵死进程。

问题背景

小结

- **前面那些千奇百怪的问题，都和systemd有关。**

- **systemd和kernel一起作为OS的基石：**
  sysytemd是系统和服务的管理器，作为PID 1运行并启动系统的其余部分，它已经是主流发行版的默认INIT、也是使用最广泛的INIT。

- **云计算场景对systemd提出了更高的要求：**
  服务器经常运行好几年都不关机，而且其上跑着成百上千的虚拟机、容器，可能运行着电商等关键业务，要求systemd的稳定性高，能长时间运行而不出故障。

目录

阿里云 | 奥运会全球指定云服务商

systemd介绍

概述

- 由Lennart Poettering于2010推出
  (http://0pointer.net/blog/projects/systemd.html)

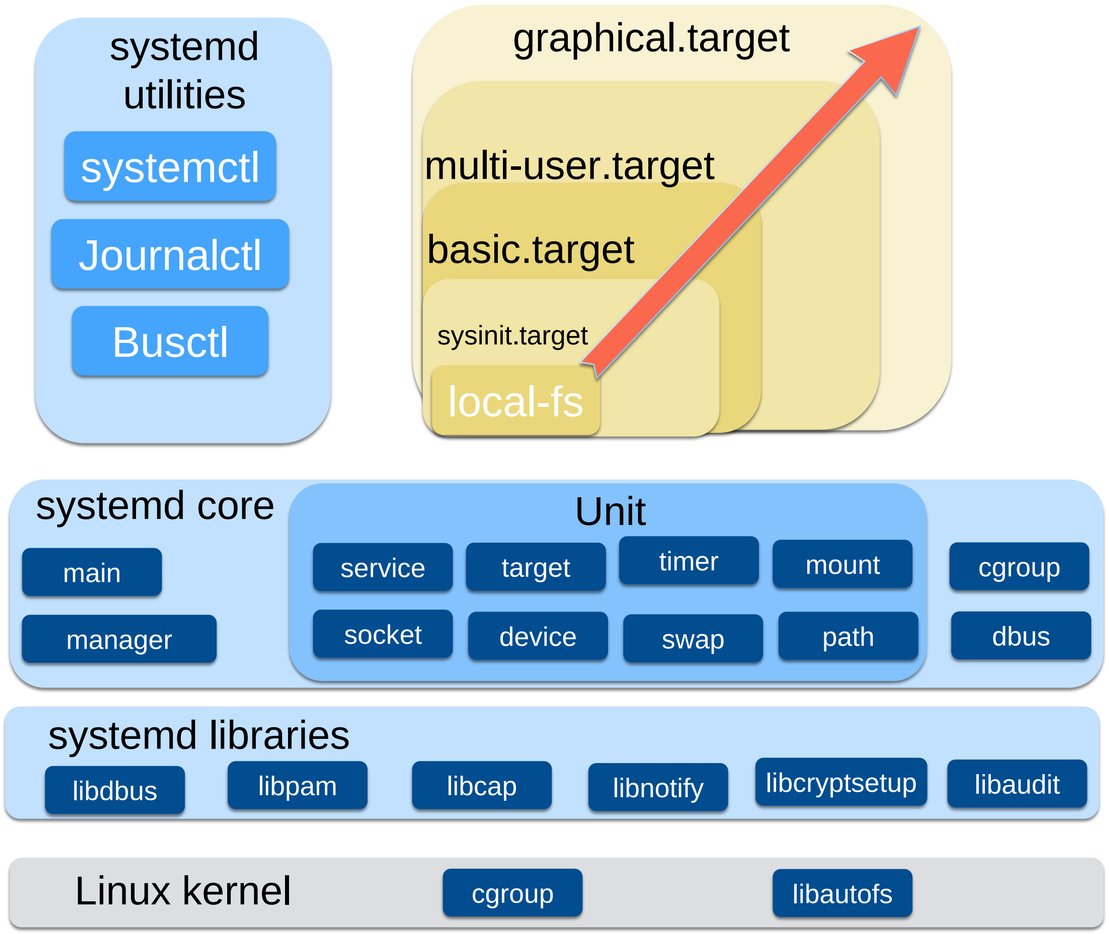
- 主要目标：
  一个先进的系统和服务的管理器（并发启动、细粒度的依赖关系管理）
  一个开放的、可扩展的平台 （OS = kernel + systemd ？）

- 项目主页：https://systemd.io/
- 代码仓库：https://github.com/systemd

systemd介绍

架构图：顶天立地、持续扩展

具有想象空间，有生命力，可以不断扩展：



根据我们在实践中的理解，可以大略划分为3块：

Platform（蓝色）：上接user、下连kernel，横向持续扩展。包括图中systemd libraries、systemd core、systemd utilities等。把kernel的特性赋能给用户态，在服务描述文件中简单配置一下即可使用（比如服务自动拉起、服务的内存限制、日志记录等）；可以持续收编越来越多的其他用户态组件，把它们纳入systemd中。

Init （黄色）：对外提供服务描述文件（可以看作广义的Linux API），实现了并发启动、依赖关系管理、资源管控等功能。

daemons （图中未画）：各种用户态组件，如udevd、networkd、oomd等等，可以被systemd收编。

systemd满足了多个用户群体的需求：

终端用户：希望尽快进入系统、运行业务程序，不希望在init 进程中浪费时间。 systemd的并发快速启动，可以满足用户的需求。

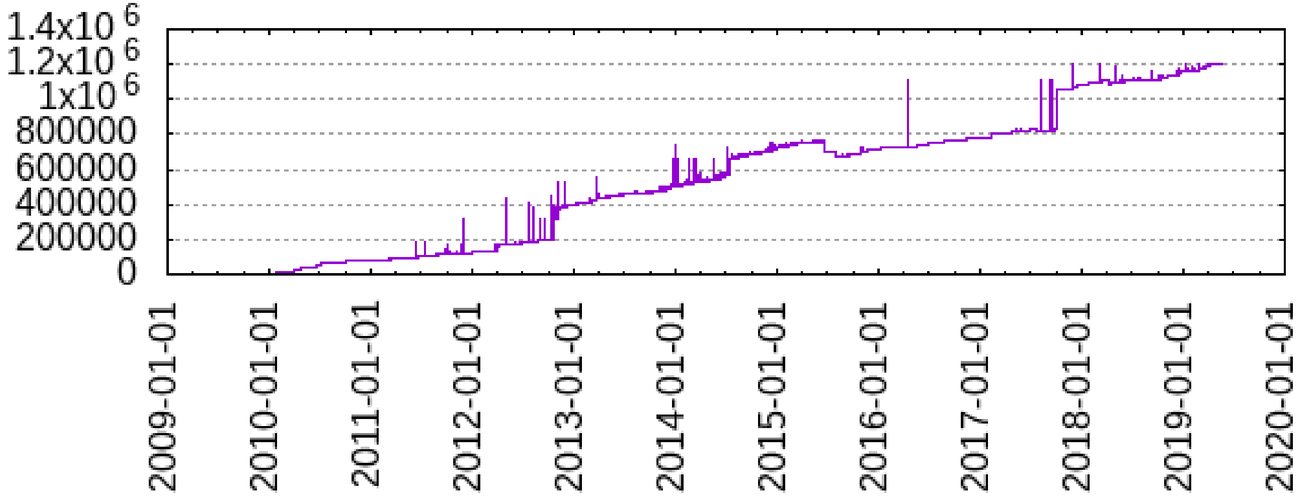
发行版厂商：希望快速、简单地制作出发行版，不希望维护成千上万的不同代码仓库、不同风格、且质量也参差不齐的用户态组件。systemd的平台化战略可以无止境的吞噬其他的用户态组件，恰好可以满足发行版厂商的需求。

软件开发者：希望聚焦在自身的业务逻辑中，不希望把精力浪费在服务的启动停止、自动拉起、日志记录等通用的、业务无关的事情上。而在systemd只需简单的配置服务描述文件，就可以实现这个需求。

systemd介绍

阿里云 | 奥运会全球指定云服务商

现状：主流发型版默认的INIT



What we already cover:

init system, journal logging, login management, device management, temporary and volatile file management, binary format registration, backlight save/restore, rfkill save/restore, bootchart, readahead, encrypted storage setup, EFI/GPT partition discovery, virtual machine/container registration, minimal container management, hostname management, locale management, time management, random seed management, sysctl variable management, console managment, . . .

代码庞大：已经超过一百二十万行，还在迅速增长；
组件众多：左边图中只是一部分，还在迅速吞并其他的组件。

从各种守护进程中提取重复的功能，并将其纳入systemd；
把传统的/etc/下的各种脚本替换为systemd的声明性配置文件。

大部分发行版都使用systemd了：
Fedora, Centos 7, RHEL 7, Mandriva, Suse, Debian, Ubuntu, . . .

systemd介绍

场景：every where，凡有OS处，即有systemd。

场景	对systemd的要求
物理机	上面跑着成百上千的容器，对systemd的要求是稳定性高，能长时间运行而不宕机、不泄漏资源（包括僵死进程、内存、FD等等）。
虚拟机	
容器	迅速拉起服务且资源占用少；对systemd的要求是尺寸小、启动快。
安全容器	

systemd介绍

基本使用：简单的命令、丰富的功能

- **systemctl管理服务**

  enable/disable/start/stop/mask/unmask/status/daemon-reload/daemon-reexec/……

- **journalctl管理日志**

  since/until/boot/dmesg/file/output/disk-usage/vacuum-size/vacuum-time/……

- **systemd-analyze 分析性能**

  critical-chain/blame/time/dot/set-log-level/dump/……

systemd介绍

小结

- OS = kernel + systemd ?
- 主流发型版默认的INIT、INIT的事实标准
- 简单的命令、丰富的功能

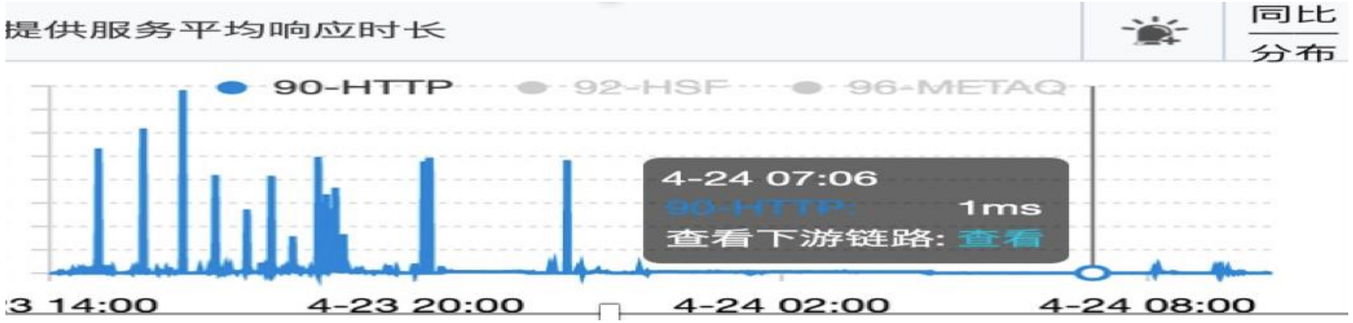
目录

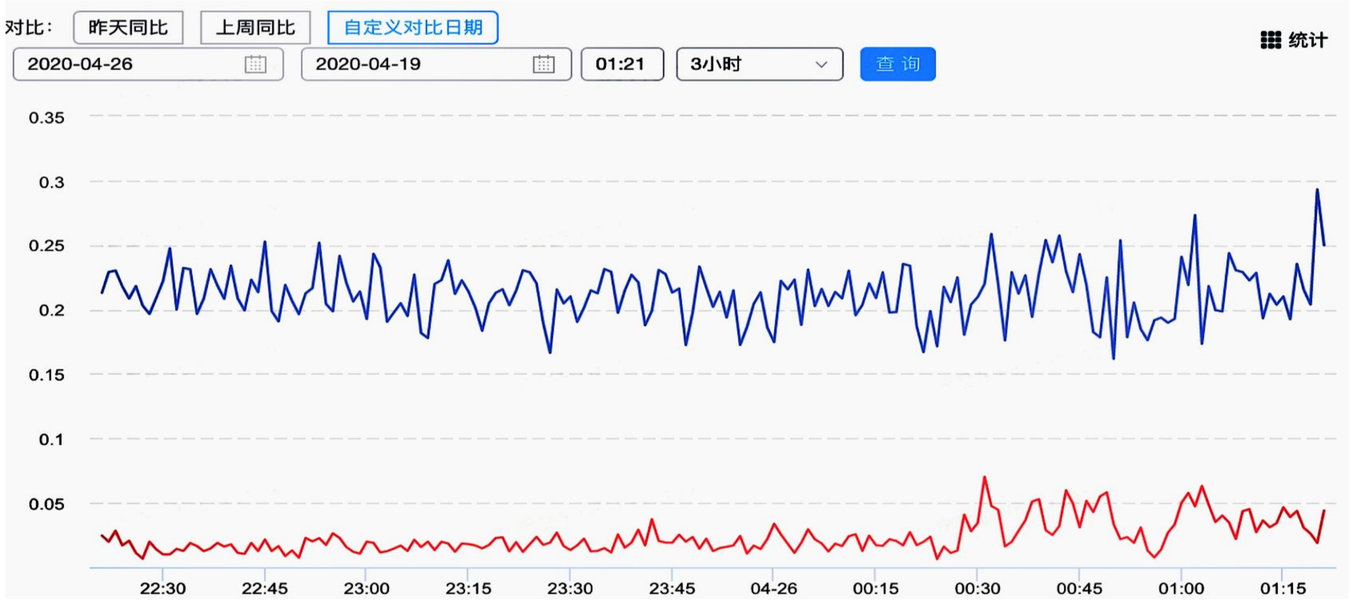
阿里云 | 奥运会全球指定云服务商

稳定性增强

阿里云 | 奥运会全球指定云服务商

集群抖动问题修复后的效果





在某个集群上更新systemd之后，业务的响应速度得到了提高，效果立竿见影。

用同样压力、同样时间周期的数据进行对比，优化之后，超时率指标从千分之二下降到了万分之五，极大提升了业务的性能。

稳定性增强

具体若干故障

阿里云 | 奥运会全球指定云服务商

● 集群抖动

- ◆ 大量创建容器时，会附带创建一系列空cgroup，导致cgroup数量急剧膨胀
- ◆ Kernel进行资源统计、遍历cgroup是一个重量级的操作
- ◆ https://github.com/systemd/systemd/issues/14453
- ◆ https://github.com/systemd/systemd/pull/14554

● 启动失败

- ◆ 由于时序问题，initramfs中的systemd和磁盘上的systemd不一致时，switch root可能会失败，系统进入紧急模式；
- ◆ 虽然重新生成initramfs可以解决；但在大规模的集群中，服务器可能已经运行很多年，initramfs保持着原始的状态，但机器上可能已经安装了很多未知的软件，重新生成initramfs会导致这些驱动、配置文件被打入initramfs，风险不可控。
- ◆ 我们在新的systemd中进行hack，可以解决这个问题：
- ◆ https://github.com/systemd-rhel/rhel-7/pull/117
- ◆ https://bugzilla.redhat.com/show_bug.cgi?id=1825232

稳定性增强

阿里云 | 奥运会全球指定云服务商

具体若干故障

- **mount 选项被篡改**
  - daemon-reexec会从根目录递归更改挂载点为shared，容器中的挂载操作可能会传递到宿主机上，引起安全问题；
  - 我们完善了daemon-reexec，仅在系统引导时才执行这个更改操作，可以避免运行中的安全问题；
  - https://github.com/systemd/systemd/pull/15196
- **daemon-reexec出现crash**
  - 特殊情况下，daemon-reexec可能会出现crash
  - https://github.com/systemd/systemd/pull/17060
- **systemd出现DoS**
  - mount path超过256字符时，systemd的units将只增不减，当达到限额128K时，会导致DoS；
  - https://github.com/systemd-rhel/rhel-7/issues/118
  - https://github.com/systemd/systemd/issues/15221

稳定性增强

具体若干故障

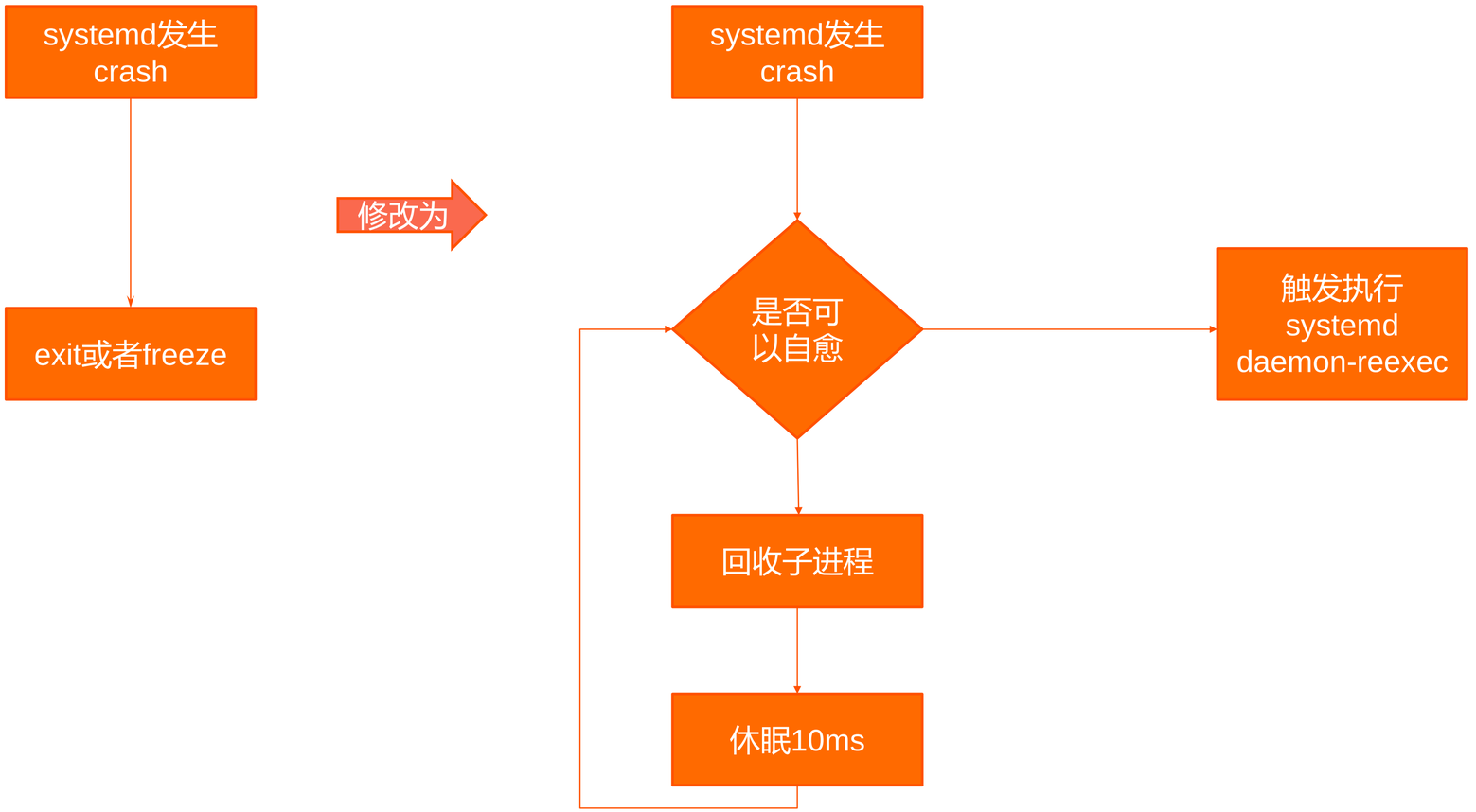
阿里云 | 奥运会全球指定云服务商

- **内存泄漏**
  - sd_bus_message不断累积，持续消耗内存，甚至会引起OOM；
  - systemd的maintainer对此进行了若干修复，可以极大的改善内存消耗；
  - https://github.com/systemd/systemd/issues/8166

- **udev进程hung死**
  - 进程卡住，udev事件无法正常处理，systemd无法reexec；
  - 该bug从Kernel 2.6.17版本开始就存在，已经潜伏14年了；
  - https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git/commit/?id=32830a0534700f86366f371b150b17f0f0d140d7

稳定性增强

稳定性兜底



永不掉线的systemd

systemd出现故障之后，仍然保留了传统init的功能：循环回收子进程，避免产生僵死进程；
同时，用户可以升级到修复了故障的新版本systemd；
最后触发执行 daemon-reexec，加载运行新版本，系统恢复正常。
整个自愈过程中，系统保持运行，不会重启。

稳定性增强

小结

阿里云 | 奥运会全球指定云服务商

- 修复了若干稳定性故障
- 初步实现了一个稳定性兜底方案
- 增强后的systemd可以应用于大规模的生产环境

目录

阿里云 | 奥运会全球指定云服务商

下一步计划

- **增强DBus**

背景：systemd作为1号进程，却强依赖于dbus，dbus本身又是systemd启动的一个服务，而且dbus不支持热升级。DBus出问题的时候，为了稳妥，通常建议重启整个系统，影响大。我们还可以尝试把KDBus再次往kernel推。

- **支持多版本切换的systemd**

背景：要是升级kernel失败了、或者升级后的kernel起不来，可以在grub中选择之前的kernel启动，因为老版本的内核还存在硬盘上。但要是升级systemd失败，系统就基本上变砖了，只有到救援模式去人工修复系统，而且有可能失败。

- **写一本书：《systemd深度解析》**

Linux发行版厂商、系统运维人员、系统开发者甚至终端用户，都离不开systemd，而目前缺乏这方面的资料。我们正在写这样一本书，在书中将系统的分析systemd源代码，同时结合阿里云大规模集群环境上遇到的各种工程实例，总结常用方法和故障诊断技巧，为用户提供帮助。

奥运会全球指定云服务商