

AI在内核故障定位的应用实践

Author: 陈增溪 chenzengxi@huawei.com



HUAWEI

HUAWEI TECHNOLOGIES CO., LTD.



CONTENTS



- 背景介绍
- 故障自动诊断系统
- 关键算法介绍
- 举例演示
- 下一步工作

背景介绍 – 数量多，种类杂

- 虚拟化运维面临的问题
 - 线上/线下百万量级的服务器
 - 内核可用性 < 99.99%
 - 日均问题数量 $1,000,000 \times 0.01\% = 100$
- 常见的问题种类
 - 宕机类 – 硬件失效/MCE/内核/驱动bug/死锁/OOM, etc.
 - 性能类 – sys或软中断冲高/IO时延增大/网络抖动
 - 存储类 – 文件系统无法挂载/只读/设备发现失败
 - 网络类 – 网络不可达/丢包/时延
 - ...

日志分析 -> 根因定位 -> 规避方案 -> 解决方案

背景介绍 - 定位难，耗时长

● Panic内核故障定位过程

- 异常堆栈分析
- 反汇编，参数推导
- 源码分析
- 故障原因推测，故障注入/复现
- 方案测试验证

专业性强，难度大，耗时长

每次宝贵的定位经验不该被浪费

```
#7 [ffff9a62a0883d08] __do_page_fault at ffffffffab52d6a0
#8 [ffff9a62a0883d70] do_page_fault at ffffffffab52d885
#9 [ffff9a62a0883da0] page_fault at ffffffffab529768
[exception RIP: rcu_check_callbacks+467]
RIP: ffffffff9aaf530d3 RSP: ffff9a62a0883e50 RFLAGS: 00010046
RAX: 0000000000000000 RBX: ffff9a624c316220 RCX: ffffffffaba8aba0
RDX: 00000000000000164 RSI: ffff9a624c316220 RDI: ffffffffaba4a678
RBP: ffff9a62a0883ea8 R8: 0000000000000000 R9: 0000000000000001
R10: 00000000000000164 R11: ffff9a1cc602af70 R12: ffffffffaba4a678
R13: 0000000000000012 R14: 0000000000000000 R15: ffff9a1cc602af70
ORIG_RAX: ffffffffffffffff CS: 0010 SS: 0018
#10 [ffff9a62a0883e60] __wake_up_common at ffffffff9aaec967b
#11 [ffff9a62a0883eb0] update_process_times at ffffffff9aaea8ff6
#12 [ffff9a62a0883ed8] tick_sched_handle at ffffffff9aaf07ee0
#13 [ffff9a62a0883ef8] tick_sched_timer at ffffffff9aaf08119
#14 [ffff9a62a0883f20] __hrtimer_run_queues at ffffffff9aaec3bd3
#15 [ffff9a62a0883f78] hrtimer_interrupt at ffffffff9aaec415f
```

```
crash> dis rcu_check_callbacks
0xffffffff9aaf52f00 <rcu_check_callbacks>:      nopl    0x0(%rax,%rax,1)
0xffffffff9aaf52f05 <rcu_check_callbacks+5>:    push    %rbp
0xffffffff9aaf52f06 <rcu_check_callbacks+6>:    mov     %rsp,%rbp
0xffffffff9aaf52f09 <rcu_check_callbacks+9>:    push    %r15
0xffffffff9aaf52f0b <rcu_check_callbacks+11>:   push    %r14
0xffffffff9aaf52f0d <rcu_check_callbacks+13>:   mov     %edi,%r14d
0xffffffff9aaf52f10 <rcu_check_callbacks+16>:   push    %r13
0xffffffff9aaf52f12 <rcu_check_callbacks+18>:   push    %r12
0xffffffff9aaf52f14 <rcu_check_callbacks+20>:   mov     %esi,%r12d
0xffffffff9aaf52f17 <rcu_check_callbacks+23>:   push    %rbx
```

CONTENTS



- 背景介绍
- **故障自动诊断系统**
- 关键算法介绍
- 举例演示
- 下一步工作

青鸟 - 故障自动诊断系统

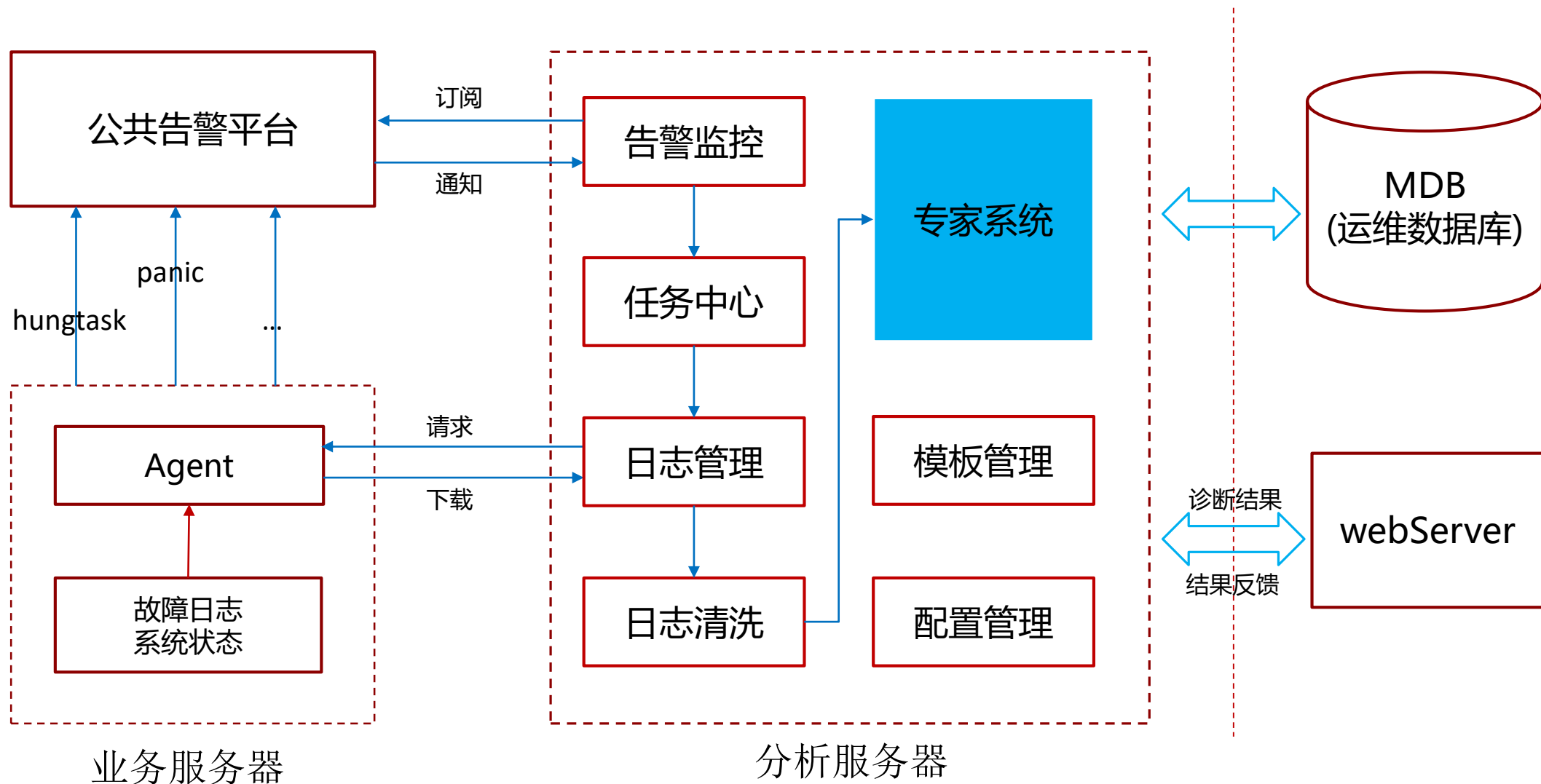
AS-IS

- 1 人工分析
- 2 专业性强，耗时长
- 3 重复分析
- 4 被动跟进

TO-BE

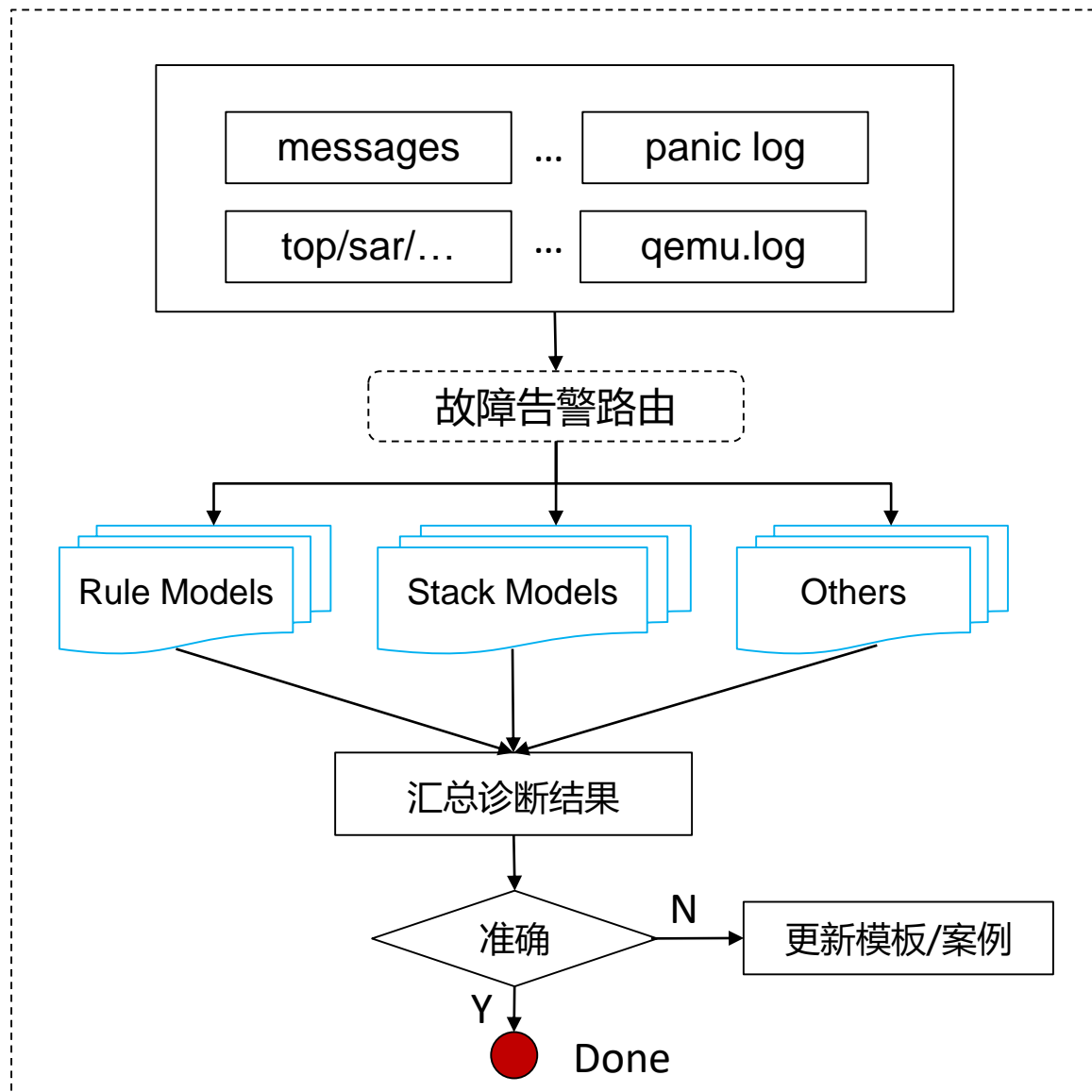
- 1 自动分析
- 2 简单自助，分钟量级
- 3 一次就好
- 4 主动发现

青鸟 - 故障自动诊断系统



青鸟 – 故障自动诊断系统

- **异常检测 – 识别已知的故障模式**
 - 支持按日志模板匹配进行异常识别
 - 支持自定义脚本根据日志上下文进行异常诊断
- **堆栈分析 – 识别已知的宕机类问题**
 - 日志堆栈信息识别、提取
 - 已知相似案例搜索 – **关键：衡量堆栈的相似度**
- **状态分析 – 辅助故障定位**
 - 性能日志分析
 - 系统状态分析
 - 虚拟机生命周期分析
 - ...
- **模板管理 – 学习反馈路径**
 - 日志模板的更新
 - 案例模板的更新



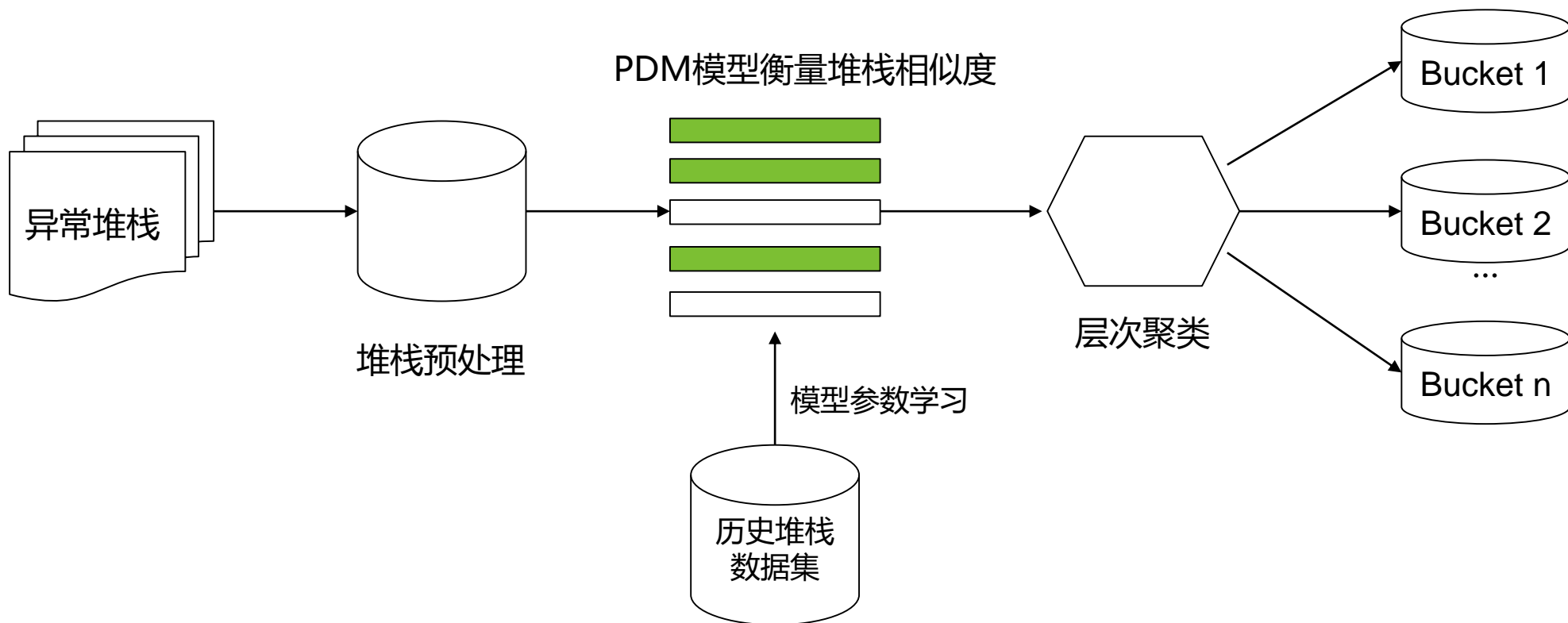
CONTENTS

- 背景介绍
- 故障自动诊断系统
- **关键算法介绍**
- 举例演示
- 下一步工作

关键算法介绍

总体流程

- 堆栈预处理 – 函数白名单
- 使用**PDM模型**衡量堆栈的相似度，其中模型参数通过**历史堆栈数据集**机器学习获取
- 通过层次聚类方法将相似的堆栈聚类到相应的Bucket内



堆栈数据集

数据源

- 总结华为云历史宕机类/软硬死锁/D状态类问题，提取合法堆栈
- 爬取kernel社区中log msg中包含Call Trace的问题，进一步人工筛选

xfs: log head and tail aren't reliable during shutdown

I'm seeing assert failures from xlog_space_left() after a shutdown has begun that look like:

```
XFS (dm-0): log I/O error -5
XFS (dm-0): xfs_do_force_shutdown(0x2) called from line 1338 of file
XFS (dm-0): Log I/O Error Detected.
XFS (dm-0): Shutting down filesystem. Please unmount the filesystem
XFS (dm-0): xlog_space_left: head behind tail
XFS (dm-0):   tail_cycle = 6, tail_bytes = 2706944
XFS (dm-0):   GH   cycle = 6, GH   bytes = 1633867
XFS: Assertion failed: 0, file: fs/xfs/xfs_log.c, line: 1310
-----[ cut here ]-----
```

Call Trace:

```
xlog_space_left+0xc3/0x110
xlog_grant_push_threshold+0x3f/0xf0
xlog_grant_push_ail+0x12/0x40
xfs_log_reserve+0xd2/0x270
? __might_sleep+0x4b/0x80
xfs_trans_reserve+0x18b/0x260
```

数据集格式

```
⊖[
⊖{
  "stack_id":1,
  "duplicate_stack":"","
  "symbols":⊖[
    "__mutex_lock_slowpath",
    "mutex_lock",
    "do_lookup",
    "do_last",
    "path_openat",
    "do_filp_open",
    "do_sys_open",
    "system_call_fastpath"
  ]
},
⊖{
  "stack_id":2,
  "duplicate_stack":"","
  "symbols":⊖[
    "dump_trace",
    "dump_stack",
    "warn_slowpath_common",
    "warn_slowpath_fmt",
    "tcp_recvmsg",
```

PDM模型

Position Dependent Model

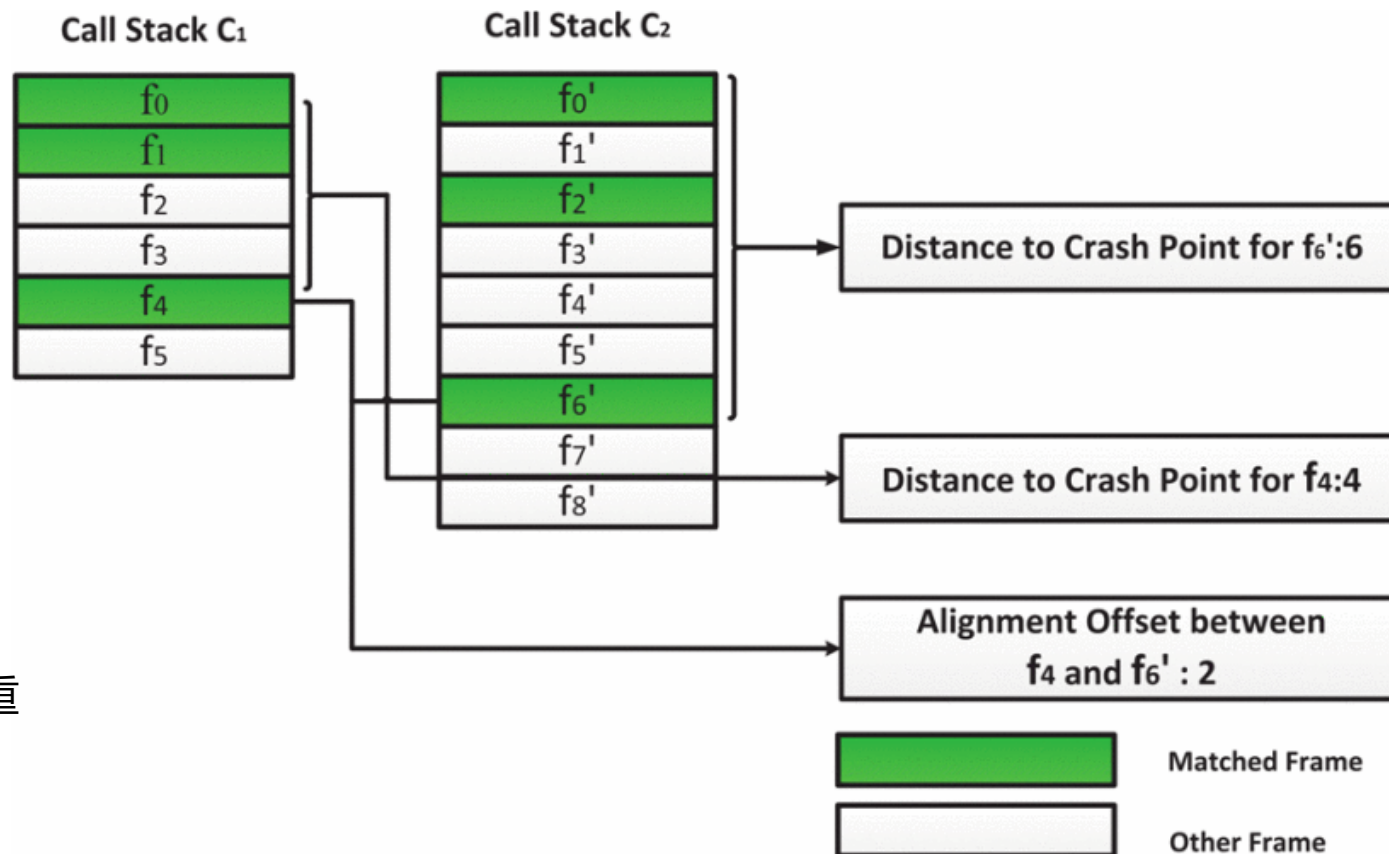
位置相关模型

- 两个关键度量

- Distance to the Top Frame
- Alignment Offset

- 基于两个假设

- 函数离栈顶越近，应分配更多权重
- 相似堆栈的对齐偏移很小



PDM模型

$$L = \{L_1, L_2, L_3 \dots\} \quad L_i = \{S_{i,1}, S_{i,2}, S_{i,3}, \dots S_{i,k} \dots\}$$

$$\begin{cases} sim(C_1, C_2) = \frac{\max_{L_i \in L} [Q(L_i)]}{\sum_{j=0}^l e^{c_j}} \\ Q(L_i) = \sum_{s_{i,k} \in L_i} e^{-c \min(Pos(C_1, s_{i,k}), Pos(C_2, s_{i,k}))} e^{-o |Pos(C_1, s_{i,k}) - Pos(C_2, s_{i,k})|} \end{cases} \quad (1)$$

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + \cos t(i, j) \\ M_{i-1,j} \\ M_{i,j-1} \end{cases} \quad (2)$$

$$\cos t(i, j) = \begin{cases} e^{-c^* \min(i,j)} e^{-o^* |i-j|} & \text{if } i\text{th frame of } C_1 = j\text{th frame of } C_2 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$sim(C_1, C_2) = \frac{M_{m,n}}{\sum_{j=0}^l e^{-c_j}} \quad (4)$$

- L 定义为 C_1, C_2 的公共子序列集合, S 定义为 L_i 内相匹配的函数
- $Pos(C_i, S_{i,k})$ 定义为函数在堆栈内的位置
- l 为堆栈 C_1, C_2 中函数数量的最小值
- 指数函数参数 c, o 分别是到栈顶的距离系数和对齐偏移系数
- $Q(L_i)$ 用来衡量的函数到栈公共帧序列中匹配的函数的相似度值, 其中第一个指数函数考虑了一对匹配顶的最小距离, 第二个指数函数考虑最小对齐偏移
- 由公式可知, 堆栈相似度由最大公共帧序列决定可用二维动态规划方法优化求解降低计算复杂度参考公式(2,3,4)

项目中的算法实现

- 计算两个堆栈距离
 - 设定c, o参数
 - 分别获取两个堆栈长度
 - M二维矩阵保存动态规划过程中各层级的相似度计算结果
 - sum存储较短堆栈的计算结果
 - 按公式计算相似度
 - 返回距离
- 训练态时定义堆栈距离大于0.08的为不同的聚类
- 实际应用时可放宽到0.2, 小于0.2时两个堆栈已非常相似

```
137 def get_distance(stack1, stack2):
138     """
139     计算两个堆栈的距离, 结果范围(0.0 ~ 1.0). 1.0: 完全不同, 0.0: 完全相同
140     :param stack1: 堆栈1列表
141     :param stack2: 堆栈2列表
142     :return: 堆栈距离
143     """
144     # c和o的系数取值由机器学习建模训练获得
145     c = 0.05 # 当前函数到栈顶距离系数(Distance to the Top Frame)
146     o = 0.05 # stack1, stack2中相同函数的对齐偏移系数(Alignment Offset)
147     len1 = len(stack1)
148     len2 = len(stack2)
149
150     if len1 == 1 or len2 == 1:
151         return 1.0
152
153     m = [[0 for col in range(len2 + 2)] for row in range(len1 + 2)]
154     for i in range(1, len1 + 1):
155         for j in range(1, len2 + 1):
156             x = 0
157             if stack1[i - 1] == stack2[j - 1]:
158                 x = math.exp(-c * min(i - 1, j - 1)) * math.exp(-o * math.fabs(i - j))
159             m[i][j] = max(max(m[i - 1][j - 1] + x, m[i - 1][j]), m[i][j - 1])
160
161     _sum = 0
162     for i in range(min(len1, len2)):
163         _sum += math.exp(-c * i)
164
165     similarity = m[len1][len2] / _sum if _sum != 0 else 0
166
167     return 1.0 - similarity
```


CONTENTS




- 背景介绍
- 故障自动诊断系统
- 关键算法介绍
- 举例演示
- 下一步工作

举例演示

案例1

根据重启的异常日志堆栈分析后，找到相似度为94.6%的已知案例。运维人员结合故障场景判定为同一问题。快速给出问题根因及解决方案。

青鸟

总览

智能分析

公有云告警

配置管理

任务中心

异常诊断

OS宕机分析

主机信息

虚拟机信息

性能日志分析

云网络监控

日志列表

OS重启记录1:

OS重启关键堆栈信息

[52056697.434189] Call Trace:
[52056697.434189] [] fc_exch_find+0x44/0x90 [libfc]
[52056697.434189] [] fc_exch_rcv_bls+0x72/0x7b0 [libfc]
[52056697.434189] [] fc_exch_rcv+0x31e/0x640 [libfc]
[52056697.434189] [] fnic_handle_frame+0x67/0xf0 [fnic]
[52056697.434189] [] worker_thread+0x11b/0x400
[52056697.434189] [] kthread+0xcf/0xe0
[52056697.434189] [] ret_from_fork+0x58/0x90

在运维数据库中找到相似案例，请参考

| 相似度 | 案例堆栈 | 案例分析 |
|-------|--|---|
| 94.6% | <div>[] fc_exch_find+0x44/0x90 [libfc] [] fc_exch_rcv_bls+0x72/0x7b0 [libfc] [] ? mempool_free+0x49/0x90 [] fc_exch_rcv+0x31e/0x640 [libfc] [] fnic_handle_frame+0x67/0xf0 [fnic] [] process_one_work+0x17b/0x470 [] worker_thread+0x11b/0x400 [] ? rescuer_thread+0x400/0x400 [] kthread+0xcf/0xe0 [] ? kthread_create_on_node+0x140/0x140 [] ret_from_fork+0x58/0x90</div> | <div>【触发场景】 现网使用的服务器会通过 会概率性触发libfc的缺陷，导致主机panic重启。</div> <div>【影响范围】 之前版本（不包括530），上述涉及版本中，若有通过</div> <div>异常会导致内核访问非法地址，触发主机异常重启；</div> <div>【社区patch】 https://github.com/torvalds/linux/commit/fa06883281afaa158b2b350f16c377c448df6b61。【解决方案】 升级及之后更高版本解决。</div> |

举例演示

案例2

根据自动分析messages中的异常D状态堆栈，可以从运维知识库获取对本次D状态发生的场景、内核流程、可能性原因等。辅助运维人员对问题做进一步分析。



青鸟

- 总览
- 智能分析
- 公有云告警
- 配置管理
- 任务中心

异常诊断OS宕机分析主机信息虚拟机信息性能日志分析云网络监控日志列表

| 异常项 | 严重程度 | 发生时间 | 异常项说明 | 参考案例 | 日志 |
|-------|-------|--------------------|---------------------|------|----------|
| 11000 | INFO | 2021-07-01 23:5... | 发现主机距离上次启动已运行时间... | | messages |
| 11002 | ERROR | 2021-07-02 00:5... | 检测到主机存在进程D状态超过12... | | messages |

```
===== D状态堆栈 =====
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361390] kbox: Hung task httpd:31323 is in D state,more than 120 seconds!
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361391] httpd
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361394] D
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361397] 31323 18135 18061
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361402] ffff8801c9e41b28 0000000000000282 ffff8801c9e41a08 ffff8801c9e41aa8
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361406] ffff8801c9e40010 ffff8801c9e41af0 ffff8801a30d6780 ffff8801a30d6780
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361409] ffff8801a30d6780 ffff8801c9e41fd8 ffff8801c9e41fd8 ffff8801a30d6780
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361412] Call Trace:
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361437] [<ffff80040c70d>] schedule_timeout+0x21d/0x2c0
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361443] [<ffff80040b601>] wait_for_common+0xe1/0x200
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361499] [<ffff80040f6d612>] __ocfs2_cluster_lock+0x4e2/0x9f0 [ocfs2]
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361565] [<ffff80040f736ae>] ocfs2_inode_lock_full_nested+0x1ee/0x480 [ocfs2]
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361599] [<ffff80040f820b2>] ocfs2_inode_revalidate+0xb2/0x150 [ocfs2]
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361652] [<ffff80040f7c5b8>] ocfs2_getattr+0x98/0x150 [ocfs2]
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361686] [<ffff800132211>] vfs_fstatat+0x81/0x90
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361692] [<ffff80013236f>] sys_newstat+0x1f/0x50
Jul 2 00:56:39 HLW-CNA30 kernel: [57475801.361698] [<ffff8004169b3>] system_call_fastpath+0x16/0x1b
通过stat文件获得文件的权限时，需要对文件inode加读锁，当前因为其它节点占用该inode锁，本节点进程加锁等待进入D状态。
===== D状态堆栈 =====
Jul 2 01:28:50 HLW-CNA30 kernel: [57477732.565193] kbox: Hung task httpd:29368 is in D state,more than 120 seconds!
Jul 2 01:28:50 HLW-CNA30 kernel: [57477732.565194] httpd
Jul 2 01:28:50 HLW-CNA30 kernel: [57477732.565197] D
Jul 2 01:28:50 HLW-CNA30 kernel: [57477732.565200] 29368 18135 18061
```

举例演示

案例3

自动分析messages日志，
根据日志模板，识别符合硬件故障的日志特征
(Hardware Error)。

根据时序判断这是由硬件故障导致的宕机事件，
提单硬件解决。



青鸟

总览

智能分析

公有云告警

配置管理

任务中心

异常诊断

OS宕机分析

主机信息

虚拟机信息

性能日志分析

云网络监控

日志列表

| 异常项 | 严重程度 | 发生时间 | 异常项说明 | 参考案例 | 日志 |
|---|---------|--------------------|--------------------|---|--------------------------|
| 11000 | ● INFO | 2021-10-08 21:1... | 发现主机距离上次启动已运行时间... | http://3ms.huawei.com/hi/group/1501 | messages |
| 2021-10-08T21:14:51.219299+08:00 err kernel[-] [28527609.892288] kvm [37977]: vcpu0 ignored rdmsr: 0x611 主机距离上次启动已运行330.0天 | | | | | |
| 10001 | ● ERROR | 2021-10-08 21:3... | 检测到主机发生硬件错误异常 | http://3ms.huawei.com/hi/group/1501 | messages |
| 2021-10-08T21:37:57.802872+08:00 warning kernel[-] [28528996.406964] {1}[Hardware Error]: Hardware error from APEI Generic Hardware Error Source: 4 2021-10-08T21:37:57.802940+08:00 warning kernel[-] [28528996.406968] {1}[Hardware Error]: It has been corrected by h/w and requires no further action 2021-10-08T21:37:57.802965+08:00 warning kernel[-] [28528996.406970] {1}[Hardware Error]: event severity: corrected 2021-10-08T21:37:57.803135+08:00 warning kernel[-] [28528996.406972] {1}[Hardware Error]: Error 0, type: corrected 2021-10-08T21:37:57.803165+08:00 warning kernel[-] [28528996.406973] {1}[Hardware Error]: fru_text: B1 | | | | | |
| 11001 | ● INFO | 2021-10-08 22:1... | 发现主机重启事件现象，请参考 | http://3ms.huawei.com/hi/group/1501 | messages |
| 2021-10-08T21:50:02.007956+08:00 info journal[-] [ramdisk-sync] not in ramdisk mode 2021-10-08T21:50:02.077168+08:00 info journal[-] message repeated 2 times: [[ramdisk-sync] not in ramdisk mode] 2021-10-08T21:50:02.085647+08:00 info systemd[-] Stopping User Slice of root. 2021-10-08T21:50:02.612115+08:00 info sh[-] Checking service vna-api ... 2021-10-08T21:50:02.612164+08:00 info sh[-] Service vna-api is normal 2021-10-08T21:50:02.935776+08:00 info sh[-] Checking service vna-beat ... 2021-10-08T21:50:02.935842+08:00 info sh[-] Service vna-beat is normal 2021-10-08T21:50:03.205444+08:00 err jeth_mem_reserve[25110] Read file failed: /opt/uvp/uvp-conf/uvp_custom.conf, error:[Errno 2] No such file or directory: /opt/uvp. 2021-10-08T21:50:03.882194+08:00 info rcdromsvr[7363] info : Daemon : finish to get all rcdrom ##### 主机重启事件 ##### 2021-10-08T22:18:14.385361+08:00 info kernel[-] [0.000000] Initializing cgroup subsys cpuset | | | | | |

小结

- 已知问题基于日志自动化分析
- 未知问题根据日志做初步辅助分析
- 未知问题完成分析后反馈系统，未知问题 -> 已知问题

下一步工作

- 持续扩充运维知识库，全面覆盖现有已知问题 → 已知问题全自动化
- 未知问题，提供更全面的辅助分析信息 → 未知问题半自动化
- 更高的异常检测效率
- 基于可靠的根因诊断尝试故障自动修复

Thank you

www.huawei.com

Copyright©2015 Huawei Technologies Co., Ltd. All Rights Reserved.

The information in this document may contain predictive statements including, without limitation, statements regarding the future financial and operating results, future product portfolio, new technology, etc. There are a number of factors that could cause actual results and developments to differ materially from those expressed or implied in the predictive statements. Therefore, such information is provided for reference purpose only and constitutes neither an offer nor an acceptance. Huawei may change the information at any time without notice.

HUAWEI TECHNOLOGIES CO., LTD.

