Memory RAS 提升云服务器高可靠性

Linux Memory RAS 实现及增强 **宋有泉 @ Intel**

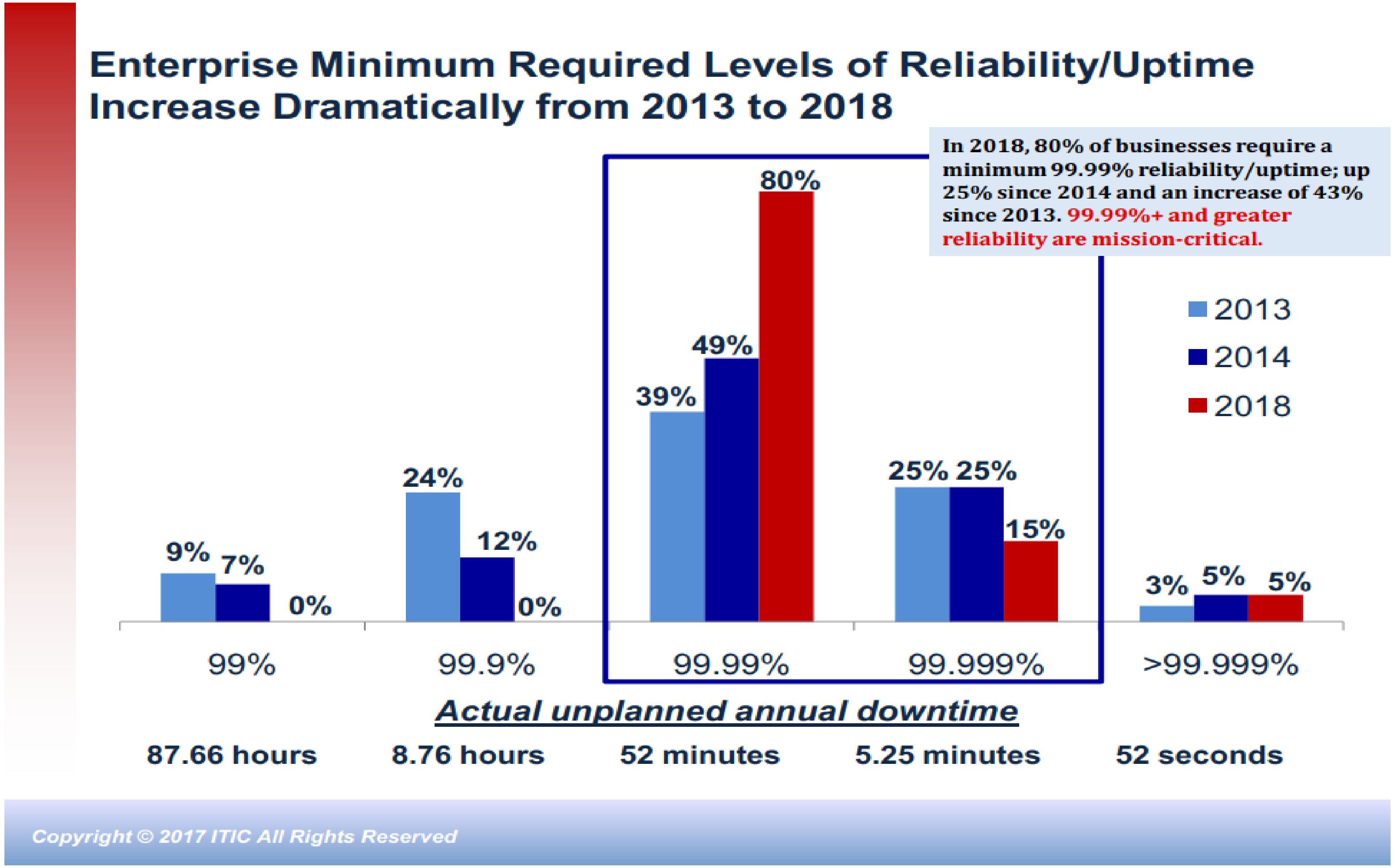
Memory RAS 在腾讯云的应用实践 **吴永楷 @ Tencent**

Agenda

- **Linux Memory RAS 实现及增强**
  - MCA
  - Linux Memory MCA Recovery
  - Linux Memory RAS status

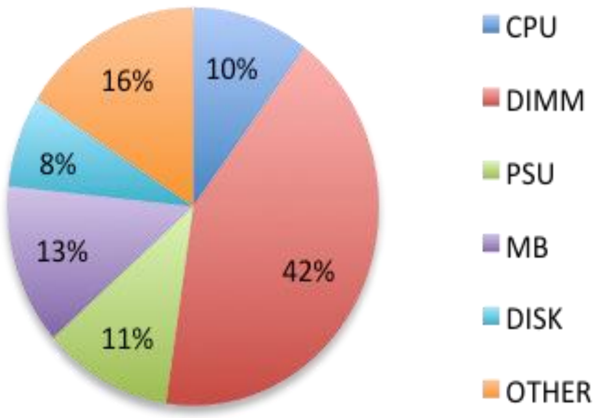
- **Memory RAS 在腾讯云的应用实践**

Server Reliability Required Level Increase Dramatically



Enterprise Minimum Required Levels of Reliability/Uptime Increase Dramatically from 2013 to 2018

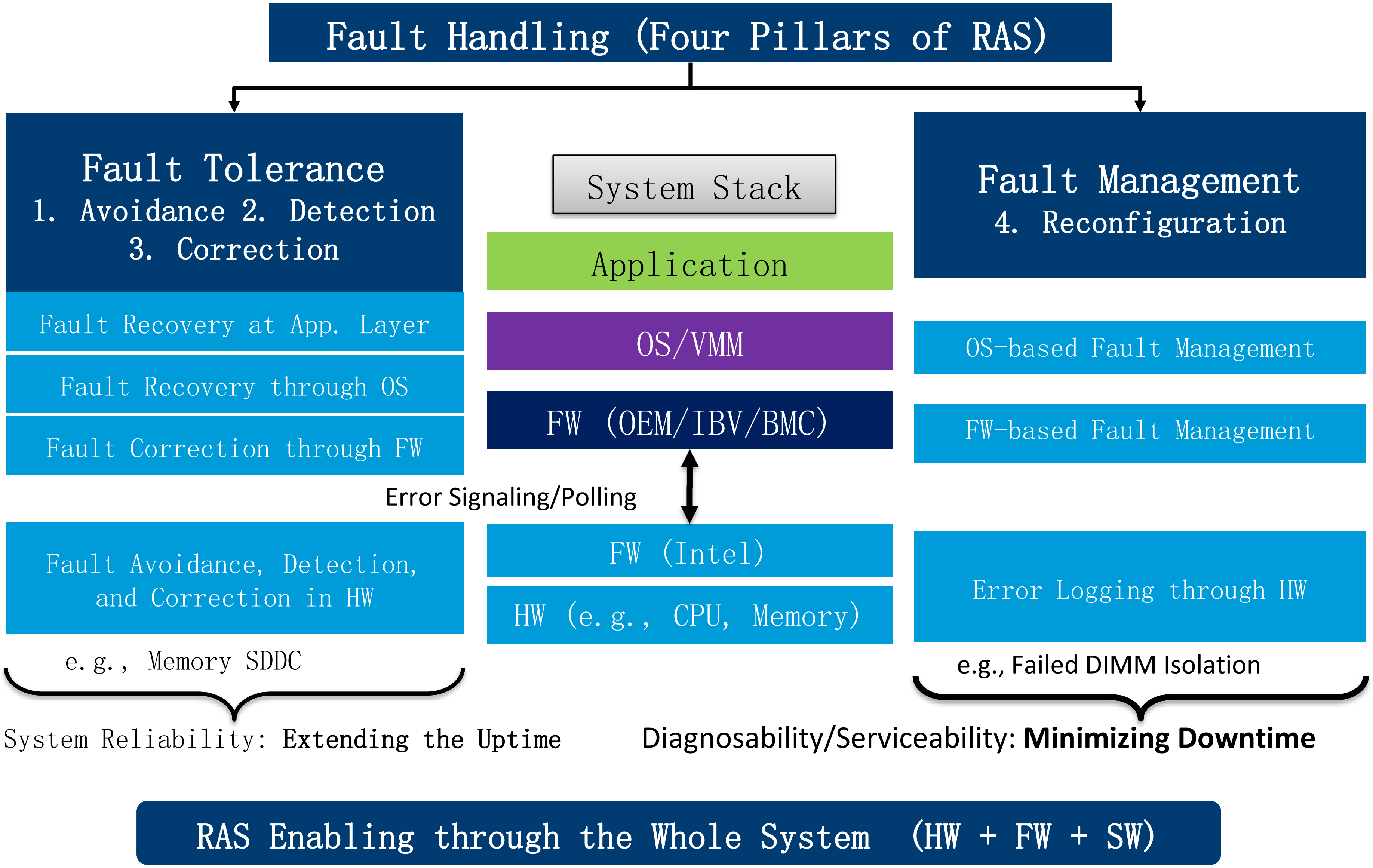
In 2018, 80% of businesses require a minimum 99.99% reliability/uptime; up 25% since 2014 and an increase of 43% since 2013. 99.99%+ and greater reliability are mission-critical.



Top 5 hardware components failure ranking in one datacenter; Memory failure rate is the top one.

Source: ITIC 2017-2018, Global Server Hardware & Server OS Reliability Survey

RAS Enabling Framework



Fault Handling (Four Pillars of RAS)

Fault Tolerance
1. Avoidance 2. Detection
3. Correction

Fault Recovery at App. Layer

Fault Recovery through OS

Fault Correction through FW

Fault Avoidance, Detection, and Correction in HW

e.g., Memory SDDC

System Reliability: **Extending the Uptime**

System Stack

Application

OS/VMM

FW (OEM/IBV/BMC)

Error Signaling/Polling

FW (Intel)

HW (e.g., CPU, Memory)

Fault Management
4. Reconfiguration

OS-based Fault Management

FW-based Fault Management

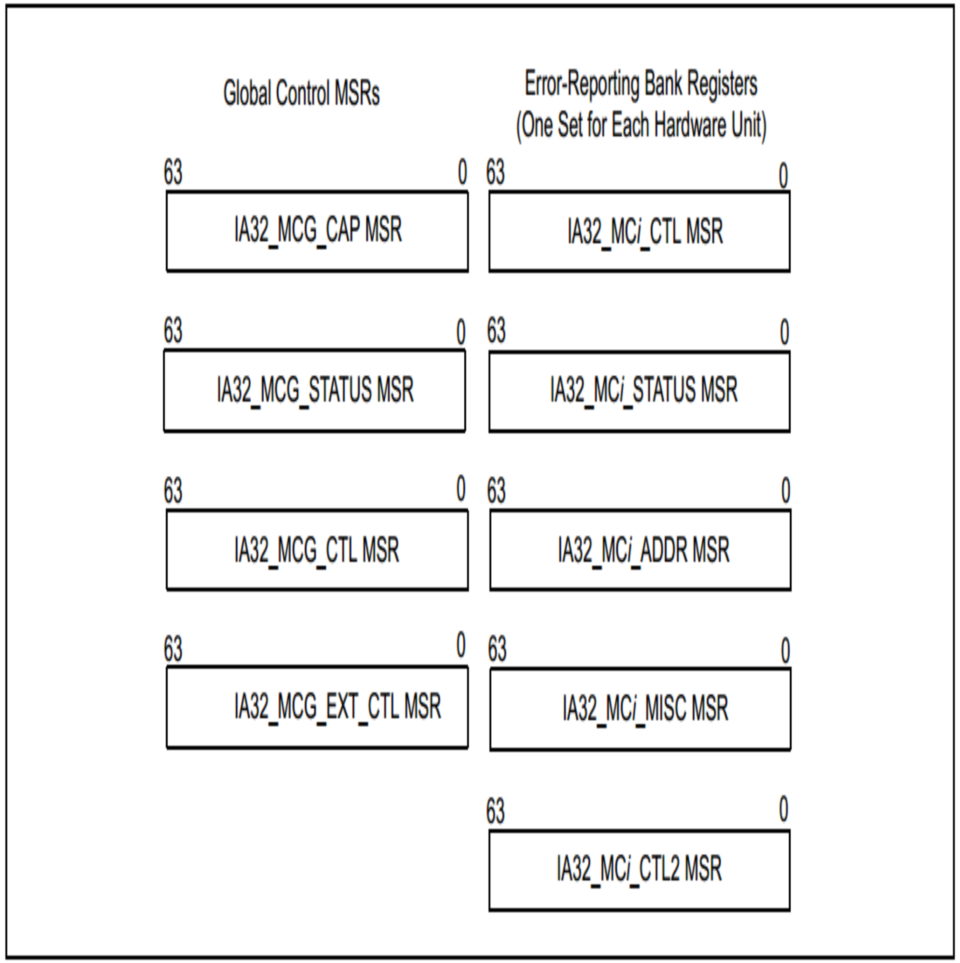
Error Logging through HW

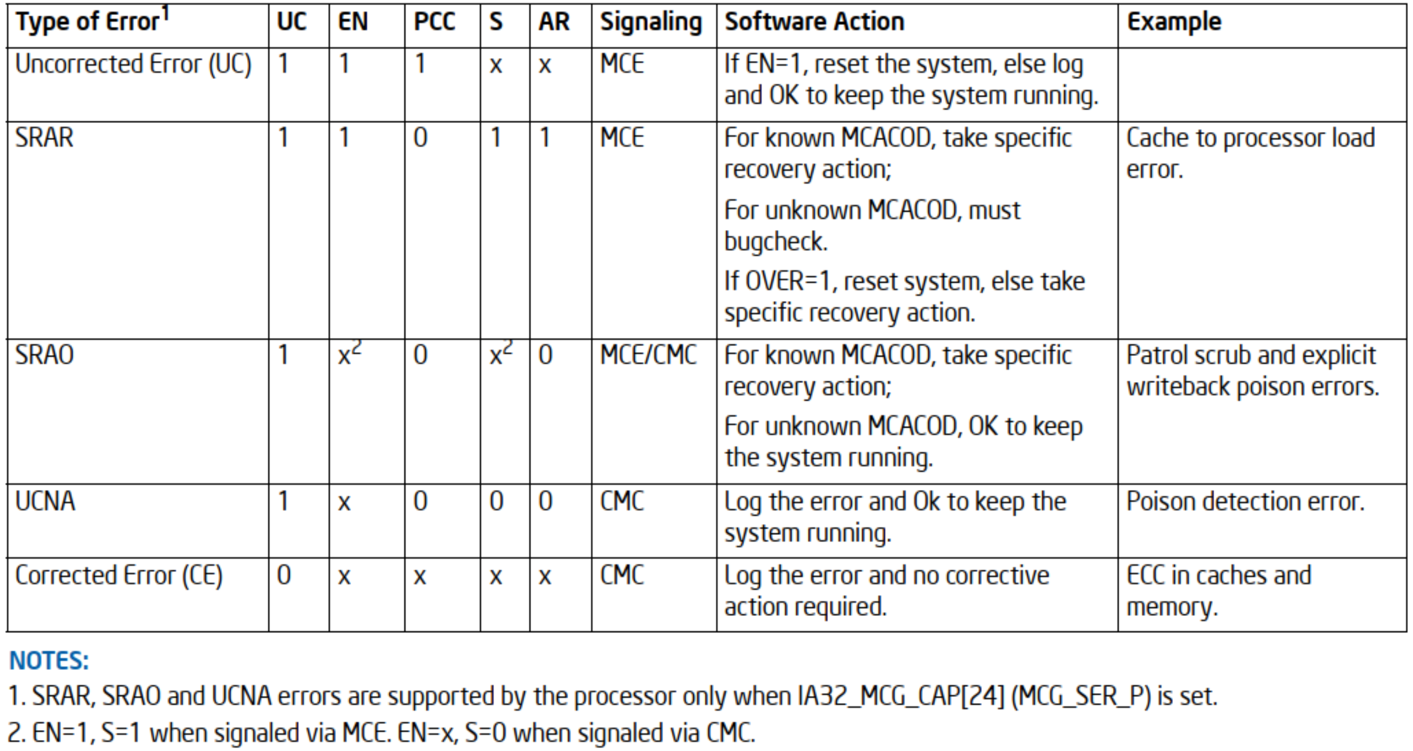
e.g., Failed DIMM Isolation

Diagnosability/Serviceability: **Minimizing Downtime**

RAS Enabling through the Whole System (HW + FW + SW)

MCA(Machine Check Architecture) on Intel® Xeon®

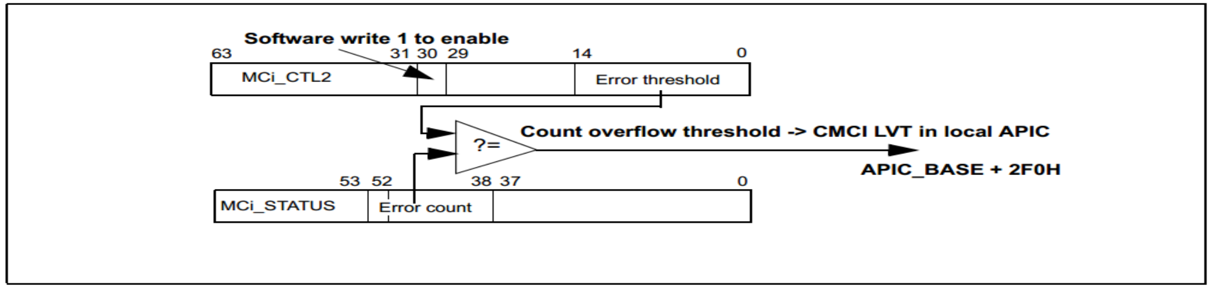




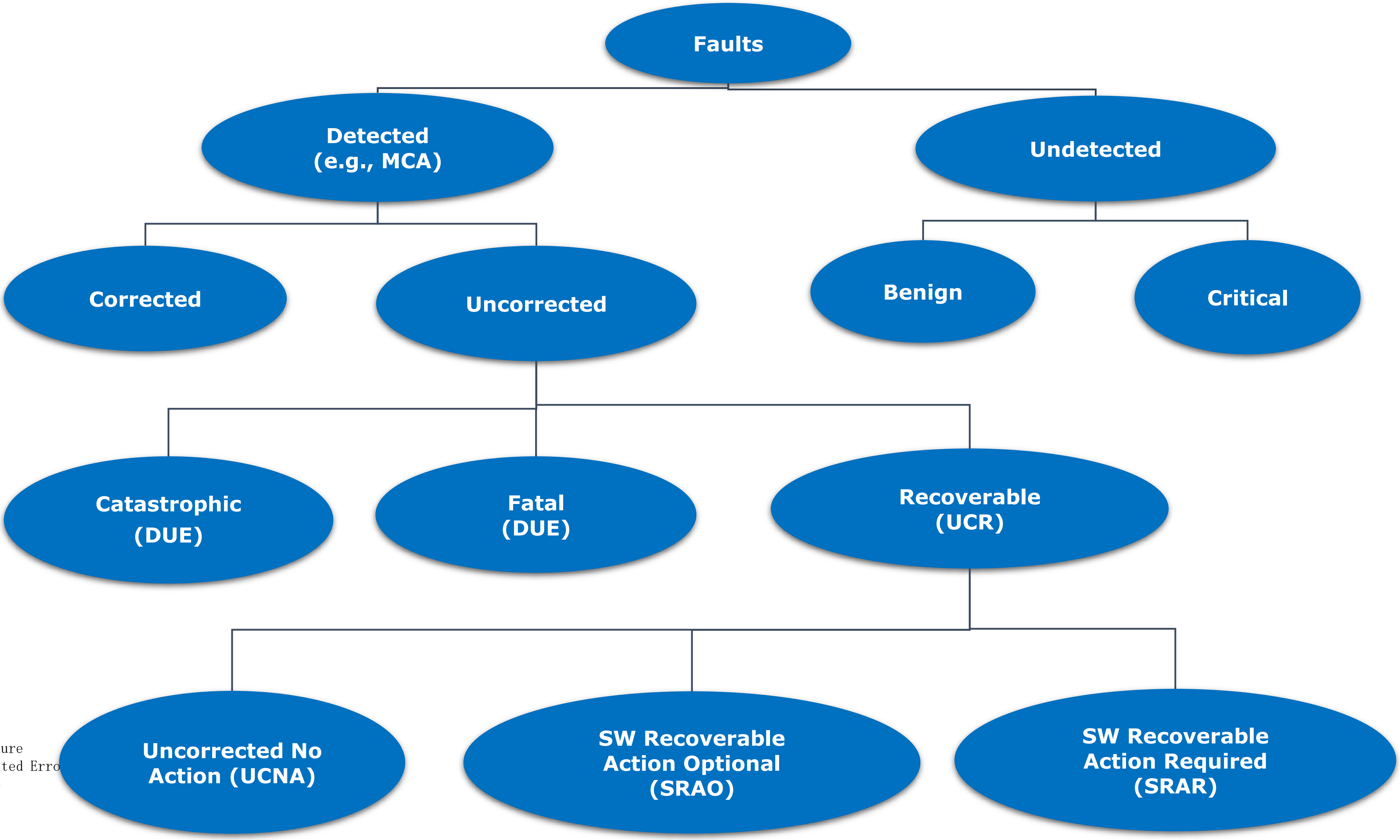
Type of Error[1]	UC	EN	PCC	S	AR	Signaling	Software Action	Example
Uncorrected Error (UC)	1	1	1	x	x	MCE	If EN=1, reset the system, else log and OK to keep the system running.	
SRAR	1	1	0	1	1	MCE	For known MCACOD, take specific recovery action; For unknown MCACOD, must bugcheck. If OVER=1, reset system, else take specific recovery action.	Cache to processor load error.
SRAO	1	x[2]	0	x[2]	0	MCE/CMC	For known MCACOD, take specific recovery action; For unknown MCACOD, OK to keep the system running.	Patrol scrub and explicit writeback poison errors.
UCNA	1	x	0	0	0	CMC	Log the error and Ok to keep the system running.	Poison detection error.
Corrected Error (CE)	0	x	x	x	x	CMC	Log the error and no corrective action required.	ECC in caches and memory.

NOTES:

1. SRAR, SRAO and UCNA errors are supported by the processor only when IA32_MCG_CAP[24] (MCG_SER_P) is set.
2. EN=1, S=1 when signaled via MCE. EN=x, S=0 when signaled via CMC.



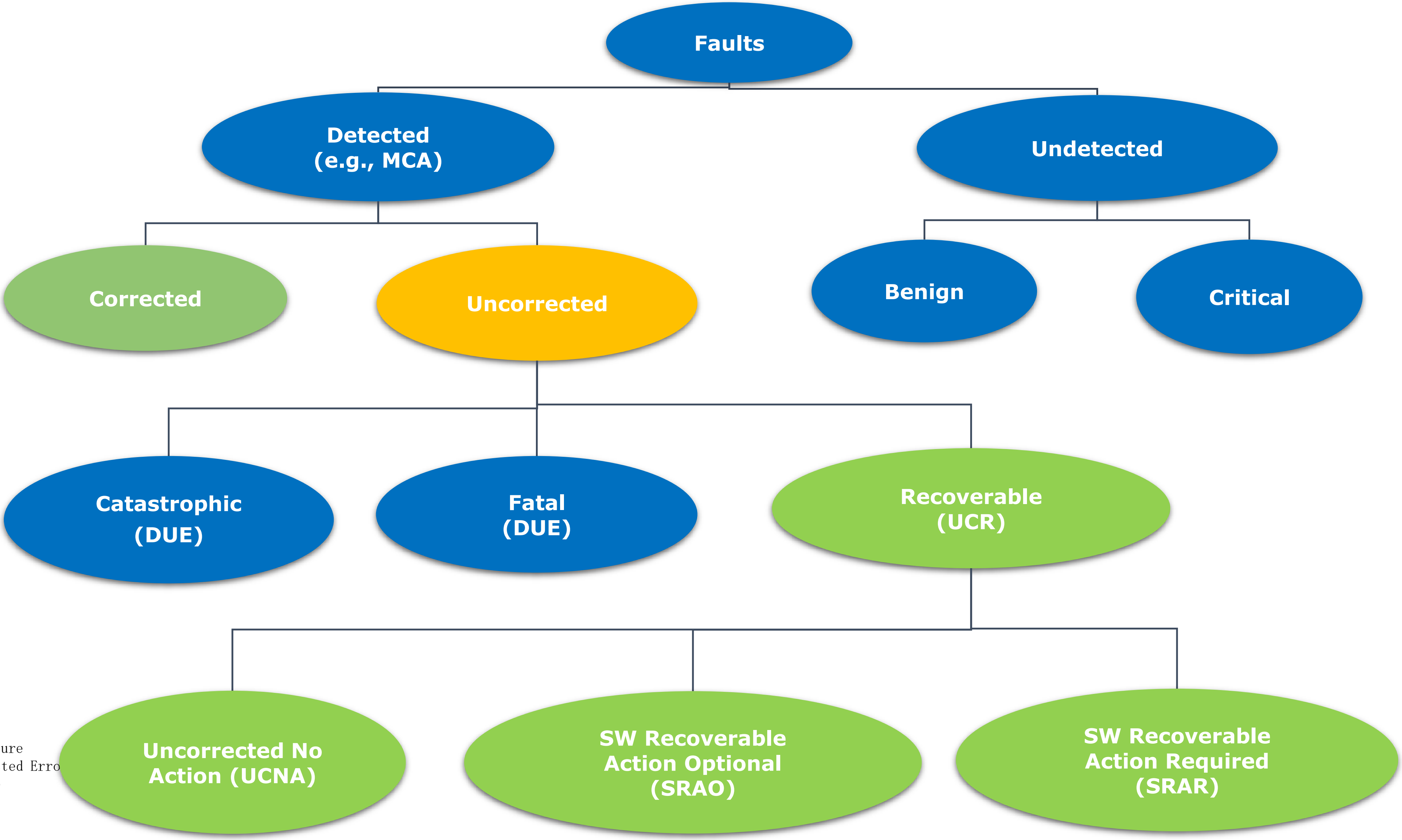
Intel® Xeon® Processor Fault Classification



- Faults
  - Detected (e.g., MCA)
    - Corrected
    - Uncorrected
      - Catastrophic (DUE)
      - Fatal (DUE)
      - Recoverable (UCR)
        - Uncorrected No Action (UCNA)
        - SW Recoverable Action Optional (SRAO)
        - SW Recoverable Action Required (SRAR)
  - Undetected
    - Benign
    - Critical

MCA: Machine Check Architecture
DUE: Detectable but Uncorrected Erro
UCR: Uncorrected Recoverable

Intel® Xeon® Processor Fault Classification



Faults

Detected (e.g., MCA)

Undetected

Corrected

Uncorrected

Benign

Critical

Catastrophic (DUE)

Fatal (DUE)

Recoverable (UCR)

Uncorrected No Action (UCNA)

SW Recoverable Action Optional (SRAO)

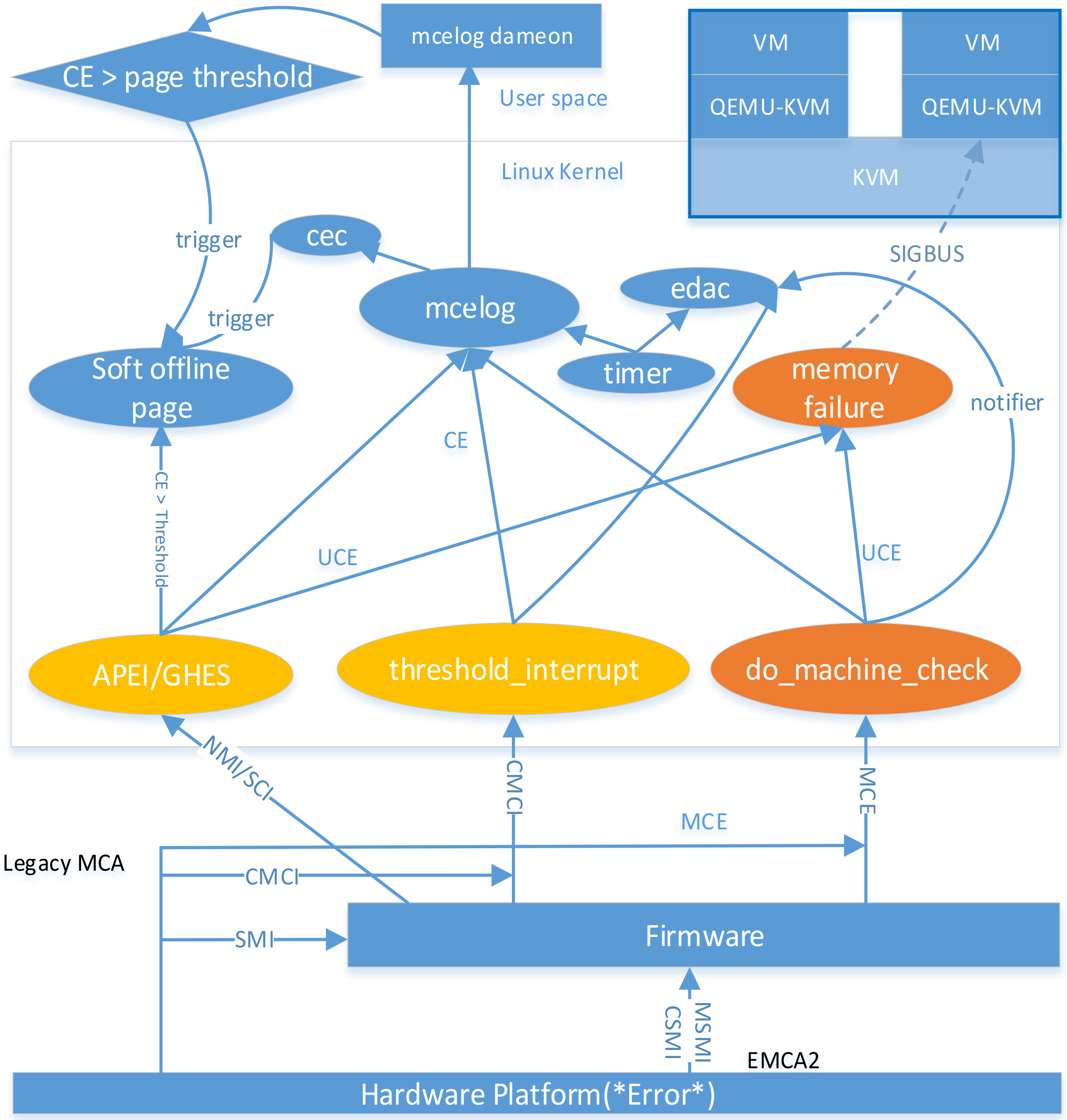
SW Recoverable Action Required (SRAR)

MCA: Machine Check Architecture
DUE: Detectable but Uncorrected Erro
UCR: Uncorrected Recoverable

Linux MCA Recovery

- Legacy MCA/EMCA/EMCA2
- CE/UCE handling
- Memory Failure to isolate the error page and even kill impact applications
- VM RAS

UCE: Uncorrected Error
CE: Corrected Error



Local MCE

Backgroud:

- Historically, MCE on Intel x86 processors broadcasts to all logical processors

Issues:

- Broadcasted MCE events may result in fatal event and prevent system recovery.

Actions:

- Intel MCA to allow signaling to only one logical processor.
- No require to perform rendezvous with other logical processors.

Kernel commits: (bc12edb8, 88d53867, 243d657e, 8838eb6c)
Benefit: 1. Enhances MCA recovery-execution path 2.Increases the possibility of recovery

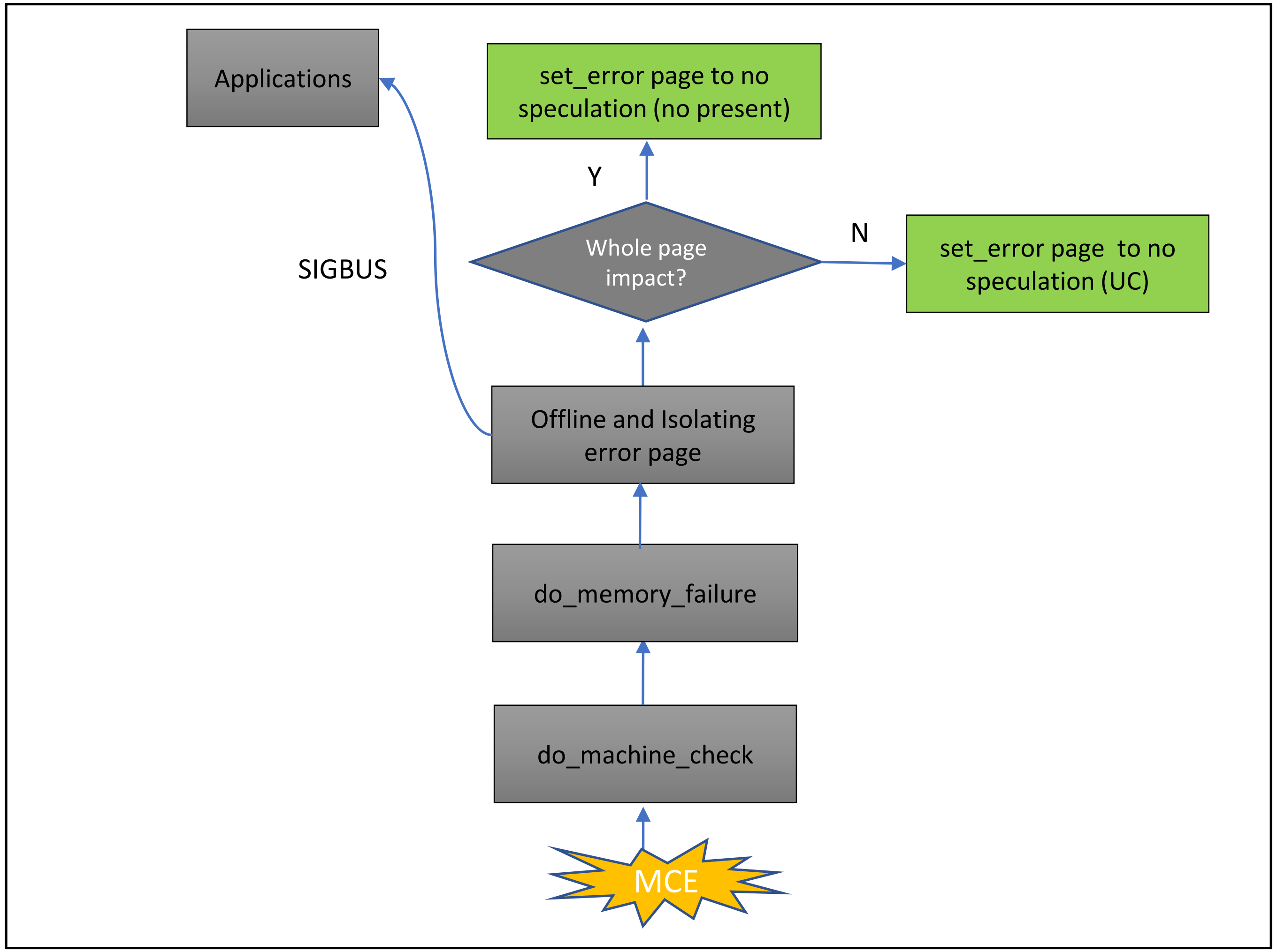
Prevent Speculation Access to Poisoned Data

Problem Statement

-Speculative access log error in MCA bank MSRs.

- Escalation of a subsequent error since the overflow bit set.

- Cause fatal error for the overflow



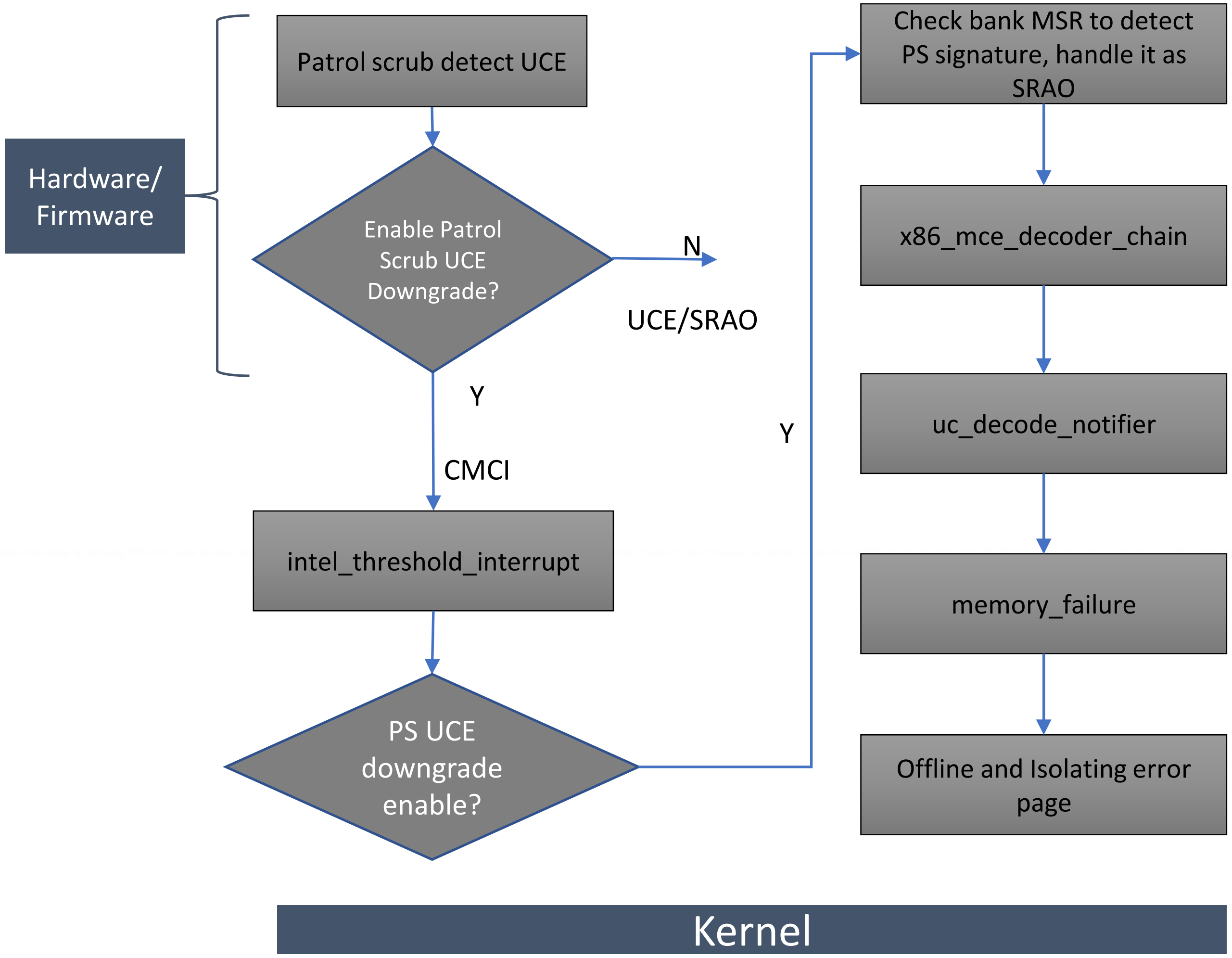
Applications
set_error page to no speculation (no present)
Y
N
Whole page impact?
set_error page to no speculation (UC)
SIGBUS
Offline and Isolating error page
do_memory_failure
do_machine_check
MCE

(kernel commits: ce0fa3e, fd0e786d, 284ce40, c748610, 17fae129)
Result: Injection memory UCE error up to 20,000+ without issue with patched kernel

Patrol Scrub SRAO Downgrade to CE – Mitigate UCE + OVR

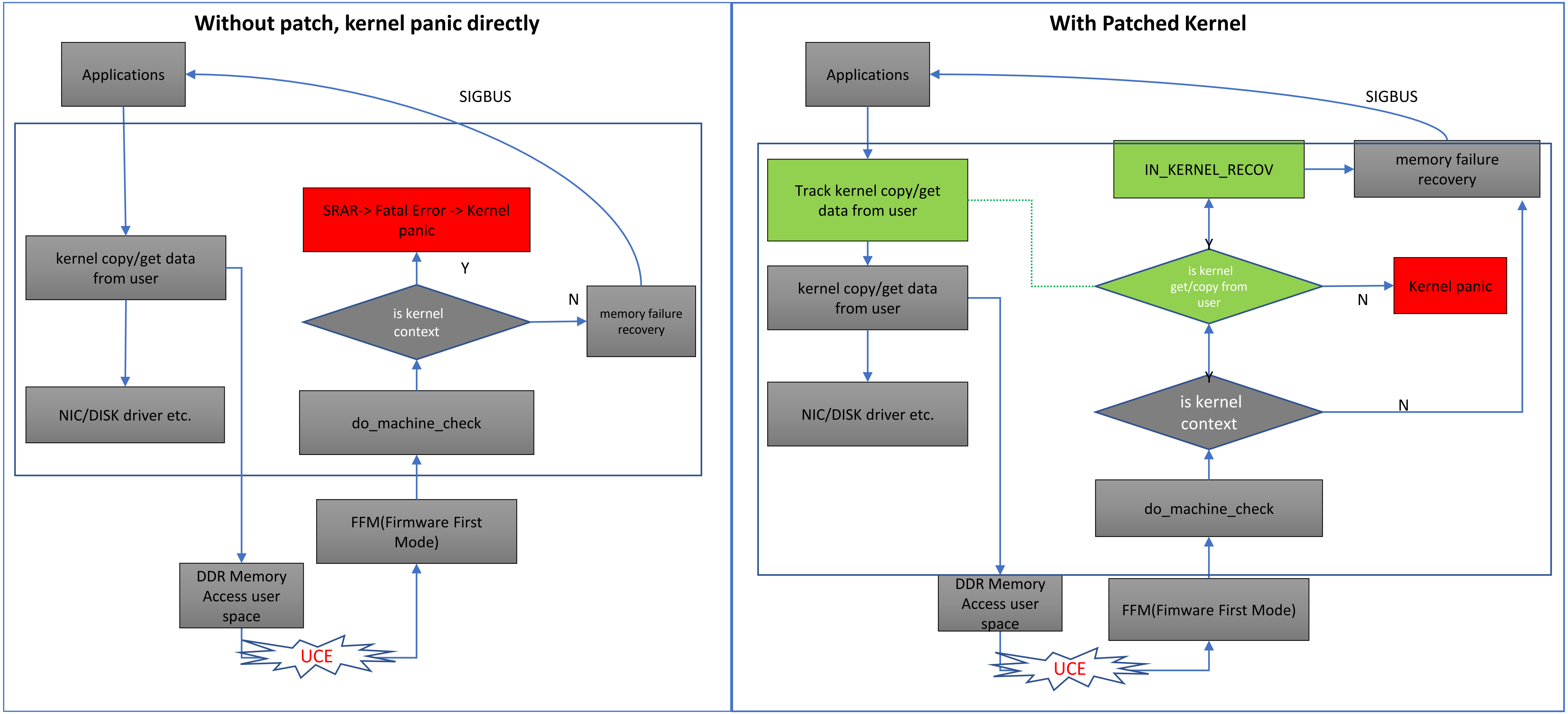
Problem Statement

- Patrol Scrub detected UCE (SRAO) signal as MCE
- Nested MCE thus triggering catastrophic fault (IERR)


Hardware/ Firmware
Patrol scrub detect UCE
Enable Patrol Scrub UCE Downgrade?
N
UCE/SRAO
Y
CMCI
intel_threshold_interrupt
PS UCE downgrade enable?
Y
Check bank MSR to detect PS signature, handle it as SRAO
x86_mce_decoder_chain
uc_decode_notifier
memory_failure
Offline and Isolating error page
Kernel

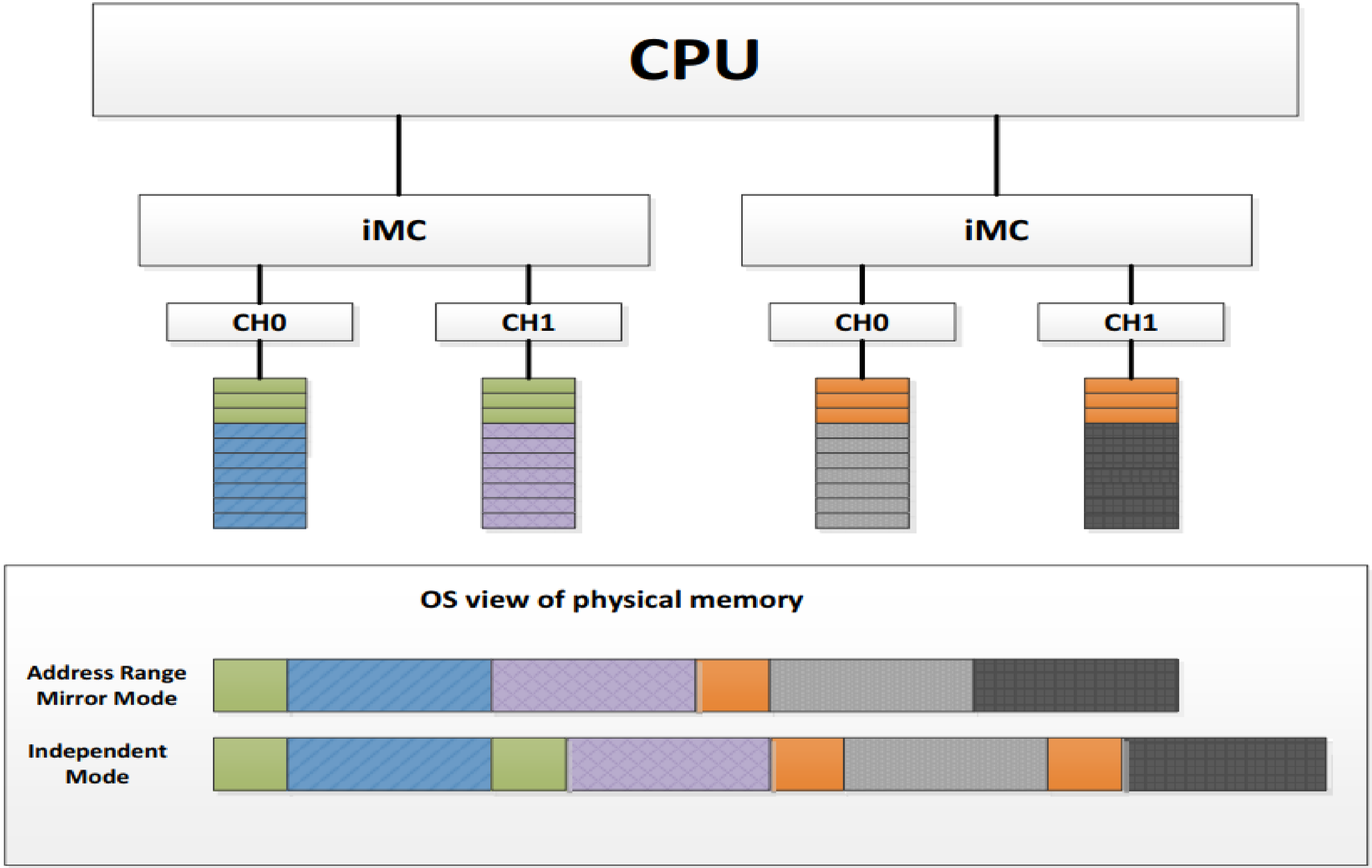
Downgrade Patrol scrub UCE to CE patch merged to v5.10 with kernel commit fd258dc4

MCE Recovery when Kernel Copy from User Space



With pathset, MCE recovery from kernel context when executing copy_user_xxx serial functions.
Patchset is merged to v5.10 now. (kernel commits: 41ce0564, a05d54c4, 278b917f, a2f73400, c0ab7ffc, 30063810)

Address Range Memory Mirror



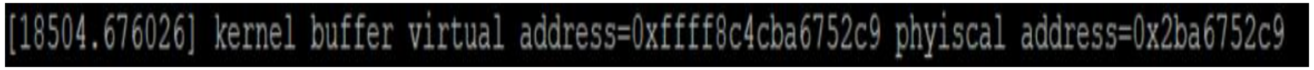
Address Range Memory Mirror – Recovery: kernel data UCE -> CE

# efibootmgr -m 1 -M 11.25

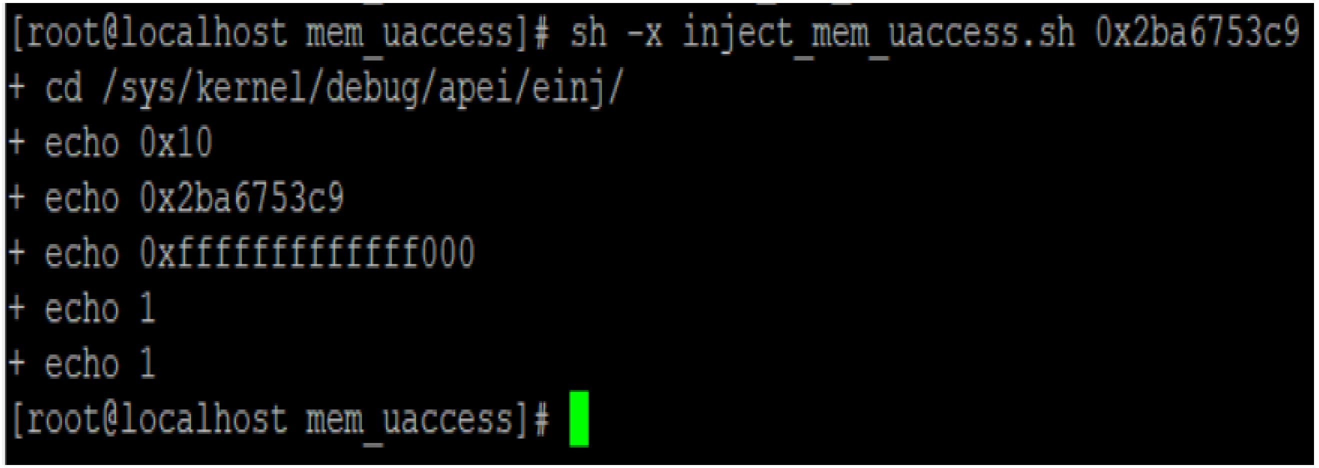
RequestMirroredPercentageAbove4G: 11.25

# dmesg | grep mirror

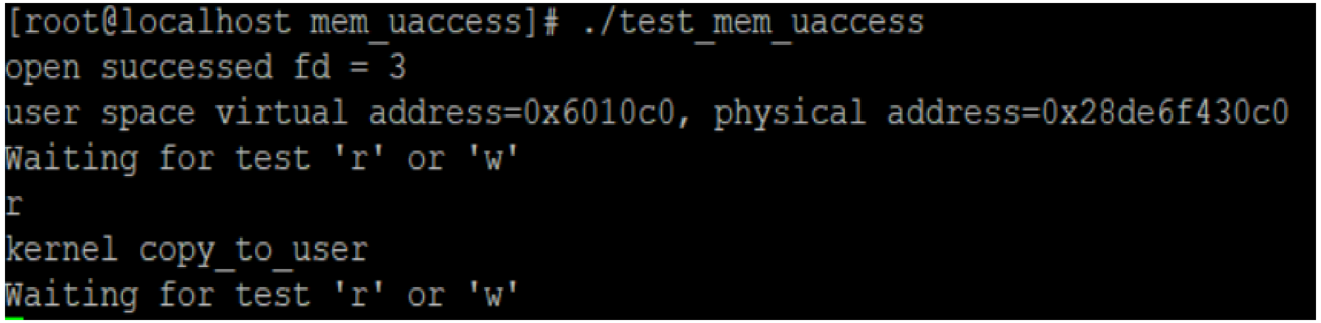
[0.000000] efi: Memory: 38647M/3454208M mirrored memory



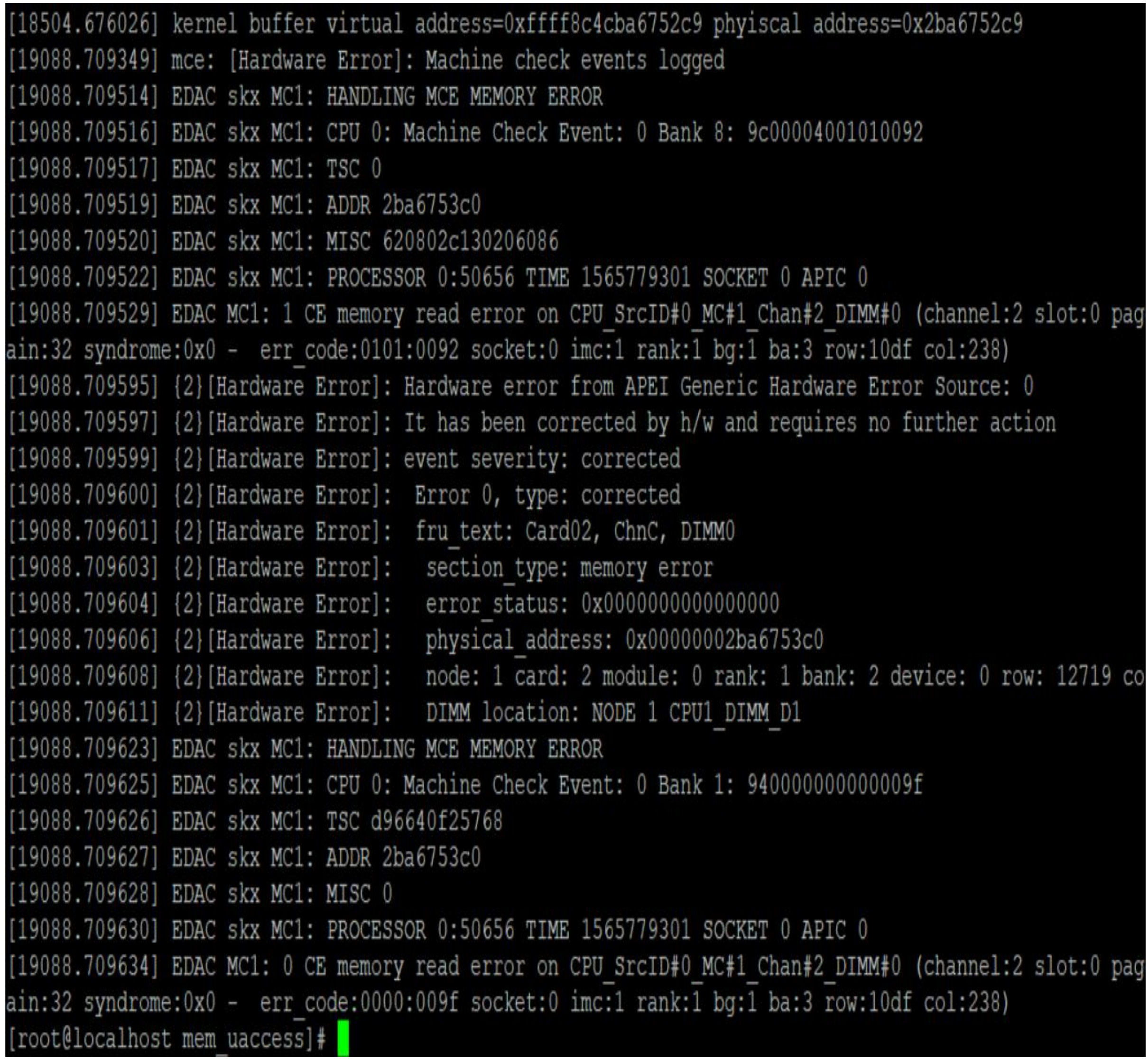
```
[18504.676026] kernel buffer virtual address=0xffff8c4cba6752c9 phyiscal address=0x2ba6752c9
```



```
[root@localhost mem_uaccess]# sh -x inject_mem_uaccess.sh 0x2ba6753c9
+ cd /sys/kernel/debug/apei/einj/
+ echo 0x10
+ echo 0x2ba6753c9
+ echo 0xfffffffffffff000
+ echo 1
+ echo 1
[root@localhost mem_uaccess]#
```



```
[root@localhost mem_uaccess]# ./test_mem_uaccess
open successed fd = 3
user space virtual address=0x6010c0, physical address=0x28de6f430c0
Waiting for test 'r' or 'w'
r
kernel copy_to_user
Waiting for test 'r' or 'w'
```



```
[18504.676026] kernel buffer virtual address=0xffff8c4cba6752c9 phyiscal address=0x2ba6752c9
[19088.709349] mce: [Hardware Error]: Machine check events logged
[19088.709514] EDAC skx MC1: HANDLING MCE MEMORY ERROR
[19088.709516] EDAC skx MC1: CPU 0: Machine Check Event: 0 Bank 8: 9c00004001010092
[19088.709517] EDAC skx MC1: TSC 0
[19088.709519] EDAC skx MC1: ADDR 2ba6753c0
[19088.709520] EDAC skx MC1: MISC 620802c130206086
[19088.709522] EDAC skx MC1: PROCESSOR 0:50656 TIME 1565779301 SOCKET 0 APIC 0
[19088.709529] EDAC MC1: 1 CE memory read error on CPU_SrcID#0_MC#1_Chan#2_DIMM#0 (channel:2 slot:0 pag
ain:32 syndrome:0x0 -  err_code:0101:0092 socket:0 imc:1 rank:1 bg:1 ba:3 row:10df col:238)
[19088.709595] {2}[Hardware Error]: Hardware error from APEI Generic Hardware Error Source: 0
[19088.709597] {2}[Hardware Error]: It has been corrected by h/w and requires no further action
[19088.709599] {2}[Hardware Error]: event severity: corrected
[19088.709600] {2}[Hardware Error]:  Error 0, type: corrected
[19088.709601] {2}[Hardware Error]:  fru_text: Card02, ChnC, DIMM0
[19088.709603] {2}[Hardware Error]:   section_type: memory error
[19088.709604] {2}[Hardware Error]:   error_status: 0x0000000000000000
[19088.709606] {2}[Hardware Error]:   physical_address: 0x00000002ba6753c0
[19088.709608] {2}[Hardware Error]:   node: 1 card: 2 module: 0 rank: 1 bank: 2 device: 0 row: 12719 co
[19088.709611] {2}[Hardware Error]:   DIMM location: NODE 1 CPU1_DIMM_D1
[19088.709623] EDAC skx MC1: HANDLING MCE MEMORY ERROR
[19088.709625] EDAC skx MC1: CPU 0: Machine Check Event: 0 Bank 1: 940000000000009f
[19088.709626] EDAC skx MC1: TSC d96640f25768
[19088.709627] EDAC skx MC1: ADDR 2ba6753c0
[19088.709628] EDAC skx MC1: MISC 0
[19088.709630] EDAC skx MC1: PROCESSOR 0:50656 TIME 1565779301 SOCKET 0 APIC 0
[19088.709634] EDAC MC1: 0 CE memory read error on CPU_SrcID#0_MC#1_Chan#2_DIMM#0 (channel:2 slot:0 pag
ain:32 syndrome:0x0 -  err_code:0000:009f socket:0 imc:1 rank:1 bg:1 ba:3 row:10df col:238)
[root@localhost mem_uaccess]#
```

Linux Memory RAS Status

◆ **UE (Uncorrected Error)**

✓ SRAR/SRAO MCA recovery – Done (v3.14)

✓ Address Range/Partial Mirror - v4.6+

✓ UCNA memory error isolate – v5.6

✓ Downgrade Patrol Scrub UCE to CE - v5.10

✓ Recovery for MCE when kernel copy from user - v5.10

✓ 1GB Hugepage Recovery - ??

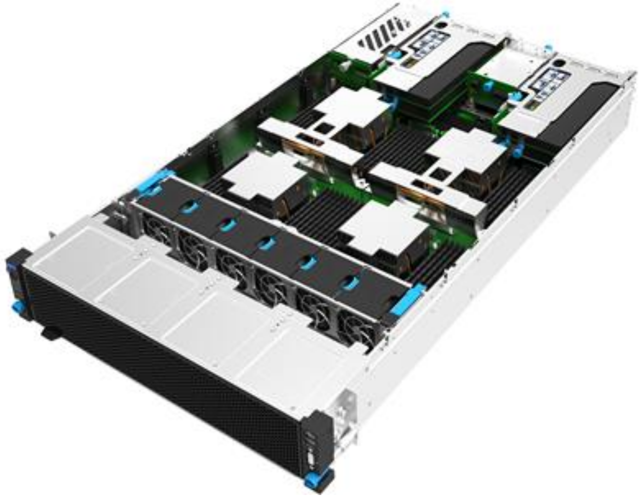
✓ Enhancement/Bug fix

- Speculation to approach UCE page
- SRAO overflow handling
- “Unknown Source MCACOD”

◆ **CE (Corrected Error)**

✓ Memory Failure Prediction/Analysis – user space & kernel support(EDAC)

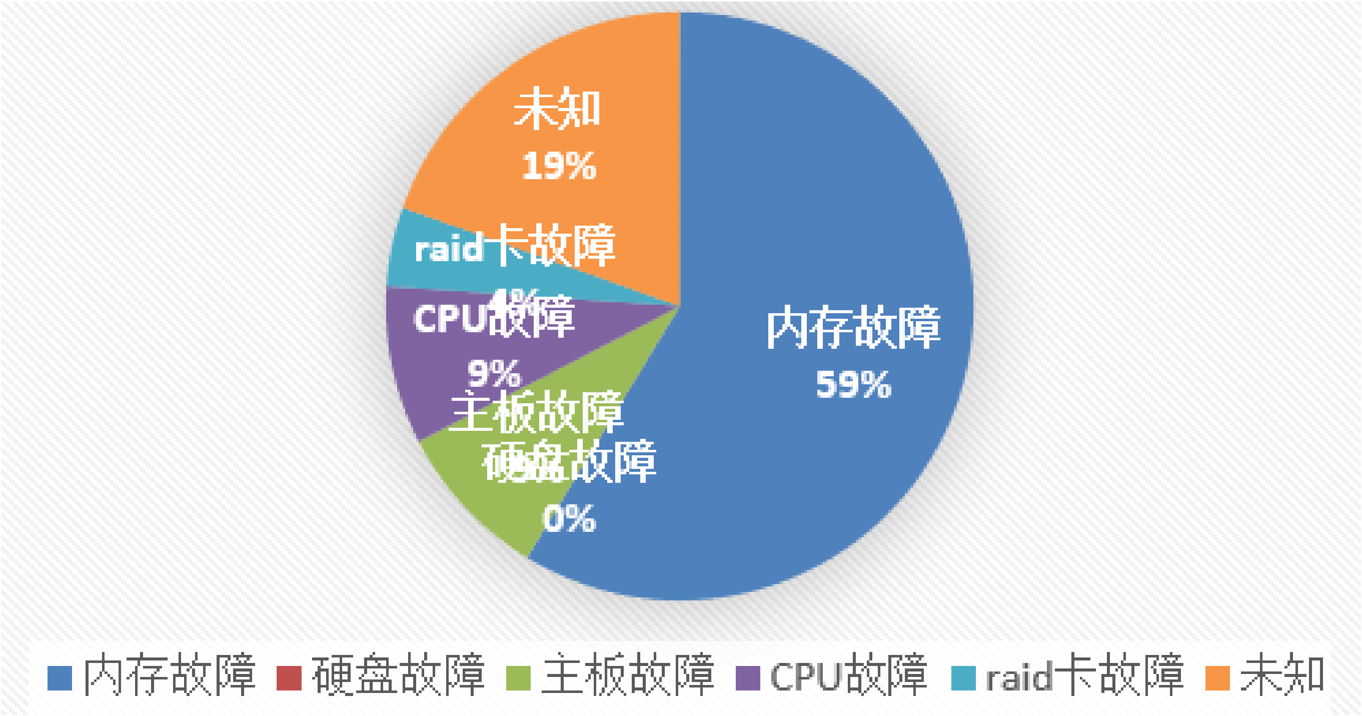
Memory RAS 在腾讯云的应用实践



- 腾讯云星星海首款自研四路服务器
- 基于第三代英特尔® 至强® 可扩展处理器
- 使用第二代英特尔® 傲腾™ 持久内存

背景

- 腾讯云英特尔® 至强® 可扩展平台服务器硬件故障导致的宕机中，内存故障占比很高



原因分析

内存故障多的原因

- 业界难题：DRAM内存颗粒上的Cell容易受环境因素及电气特性影响发生故障
- 内存故障发生后，目前业界采用的解决方案是在CPU的内存控制器上增加ECC算法来进行内存纠错
- ECC算法可以纠正的错误称为CE错误，无法纠正的错误称为UC错误

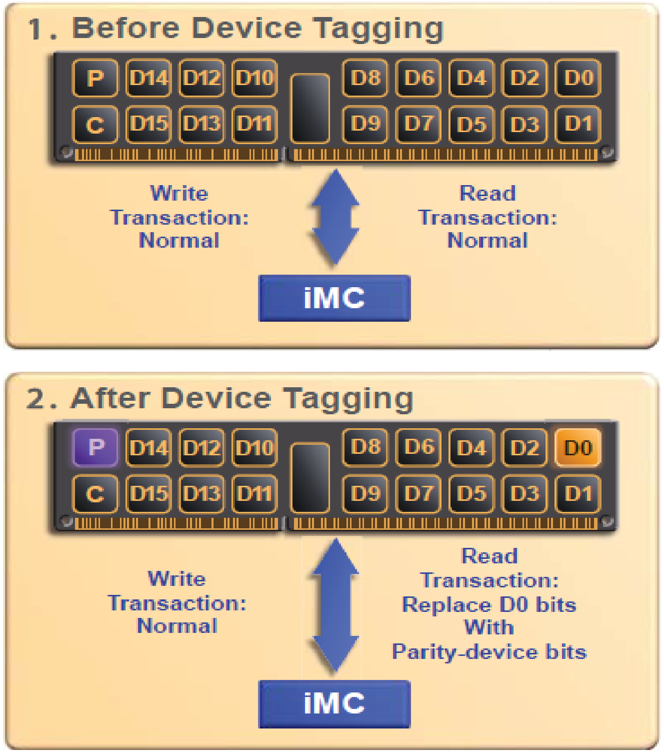
解决方案对比

在英特尔® 至强® 可扩展平台上，业界用于提高内存可靠性的技术有：

- SDDC+1
- ADDDC+1
- Memory Mirroring

解决方案对比

- SDDC+1



1. Normal Memory Write/Read

2. Example: Device D0 hard failure.
   1. Corrected Error Count would reach threshold quickly.

   2. BIOS/SMM detects the failed DRAM Device D0. Triggers Device Tagging.

After Device Tagging

1. Memory Writer operation: Unchanged (Normal).

2. Memory read operation:

   1. D0 device data is replaced with that of Parity Device.

   2. iMC does normal error Checking.

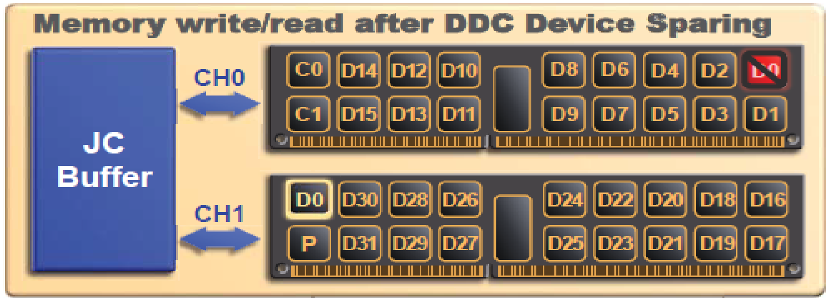
   3. Intel Xeon Scable Processor Family: Upon detecting error, logs error and signal MCE.

优点：由CPU硬件及UEFI固件直接完成，无须OS软件干涉，集成简单

缺点：牺牲了纠错能力，在做了device tagging后容易造成后续UC错误增多，引起CPU IERR

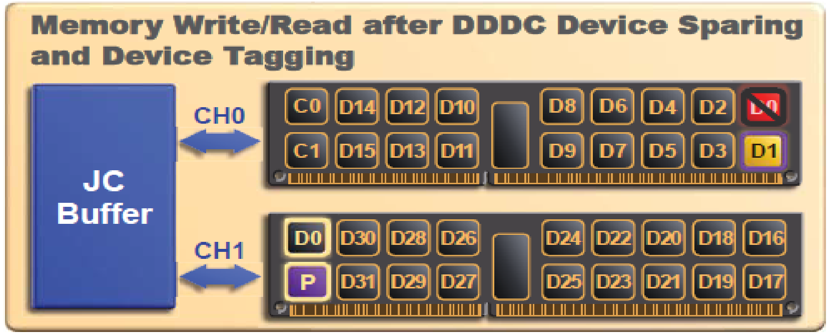
解决方案对比

- ADDDC+1



Memory write/read after DDC Device Sparing

JC Buffer
CH0
CH1
C0 D14 D12 D10 D8 D6 D4 D2 D0
C1 D15 D13 D11 D9 D7 D5 D3 D1
D0 D30 D28 D26 D24 D22 D20 D18 D16
P D31 D29 D27 D25 D23 D21 D19 D17



Memory Write/Read after DDDC Device Sparing and Device Tagging

JC Buffer
CH0
CH1
C0 D14 D12 D10 D8 D6 D4 D2 D0
C1 D15 D13 D11 D9 D7 D5 D3 D1
D0 D30 D28 D26 D24 D22 D20 D18 D16
P D31 D29 D27 D25 D23 D21 D19 D17

1. Normal Memory Write/Read with Spare Device as D0.

2. Example: Device D1 hard failure (second device).

   1. Corrected Error Count would reach threshold quickly.

   2. BIOS/SMM detects the failed DRAM Device D1. Triggers Device Tagging.

After DDDC Device Sparing

1. Memory Write operation: Unchanged. Still using spare device.

2. Memory read operation:

   1. D1 device data is replaced with that of Parity Device.

   2. iMC does normal error checking.

   3. Upon detecting error, logs error and corrects SBE. In case of MBE, logs error and signal MCE.

优点

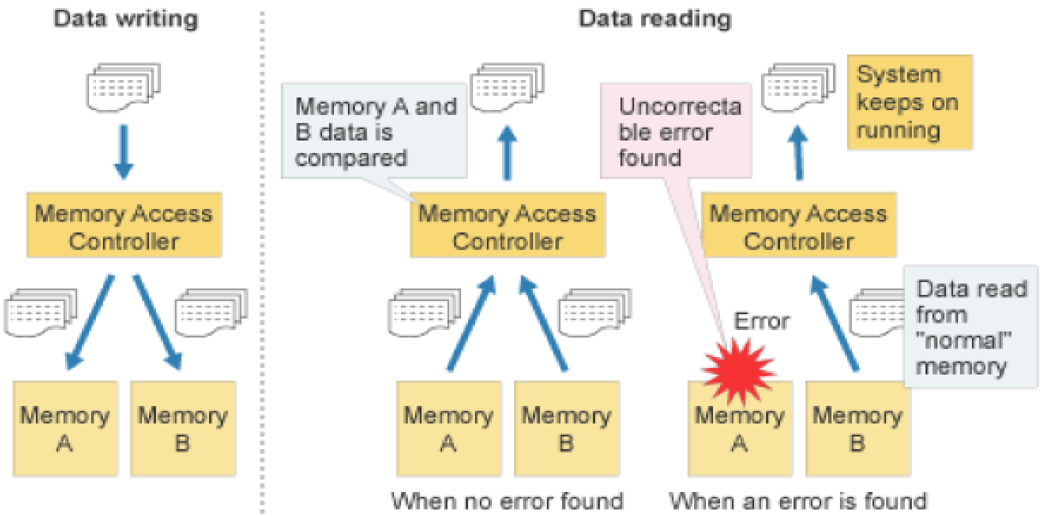
1. 由CPU硬件及UEFI固件直接完成，无须OS软件干涉，集成简单
2. 可以同时覆盖两个Rank上的任意两个故障颗粒

缺点

1. Lockstep模式启动后对系统性能有一定影响
2. 在触发了device sparing后，后续进一步发生的错误容易造成后续UC错误增多，引起CPU IERR

解决方案对比

- Memory Mirroring



优点

1. 作用范围广，容错能力突出
2. 由CPU硬件及UEFI固件直接完成，无须OS软件干涉，集成简单

缺点

增加了服务器内存的成本

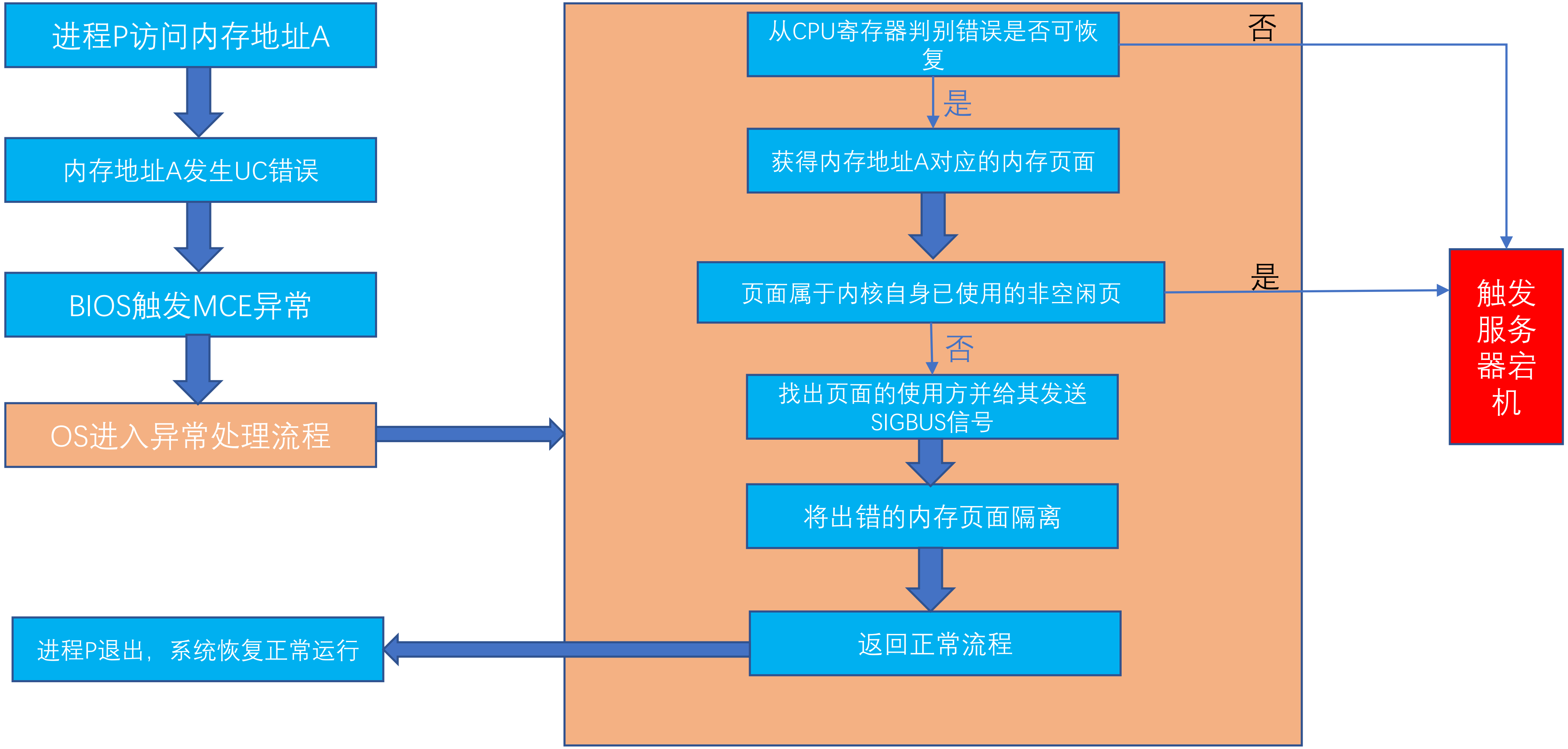
MCA Recovery

- 核心概念：

1. UC错误不直接触发OS的硬件宕机流程，把决定权交给OS
2. 将内存UC错误根据触发场景进一步细分为SRAR、SRAO、UCNA等概念
3. OS根据不同的错误类型以及出现UC错误的内存页面的使用情况，采取不同的恢复策略

MCA Recovery

- SRAR：



进程P访问内存地址A

内存地址A发生UC错误

BIOS触发MCE异常

OS进入异常处理流程

从CPU寄存器判别错误是否可恢复

是

获得内存地址A对应的内存页面

页面属于内核自身已使用的非空闲页

否

找出页面的使用方并给其发送SIGBUS信号

将出错的内存页面隔离

返回正常流程

进程P退出，系统恢复正常运行

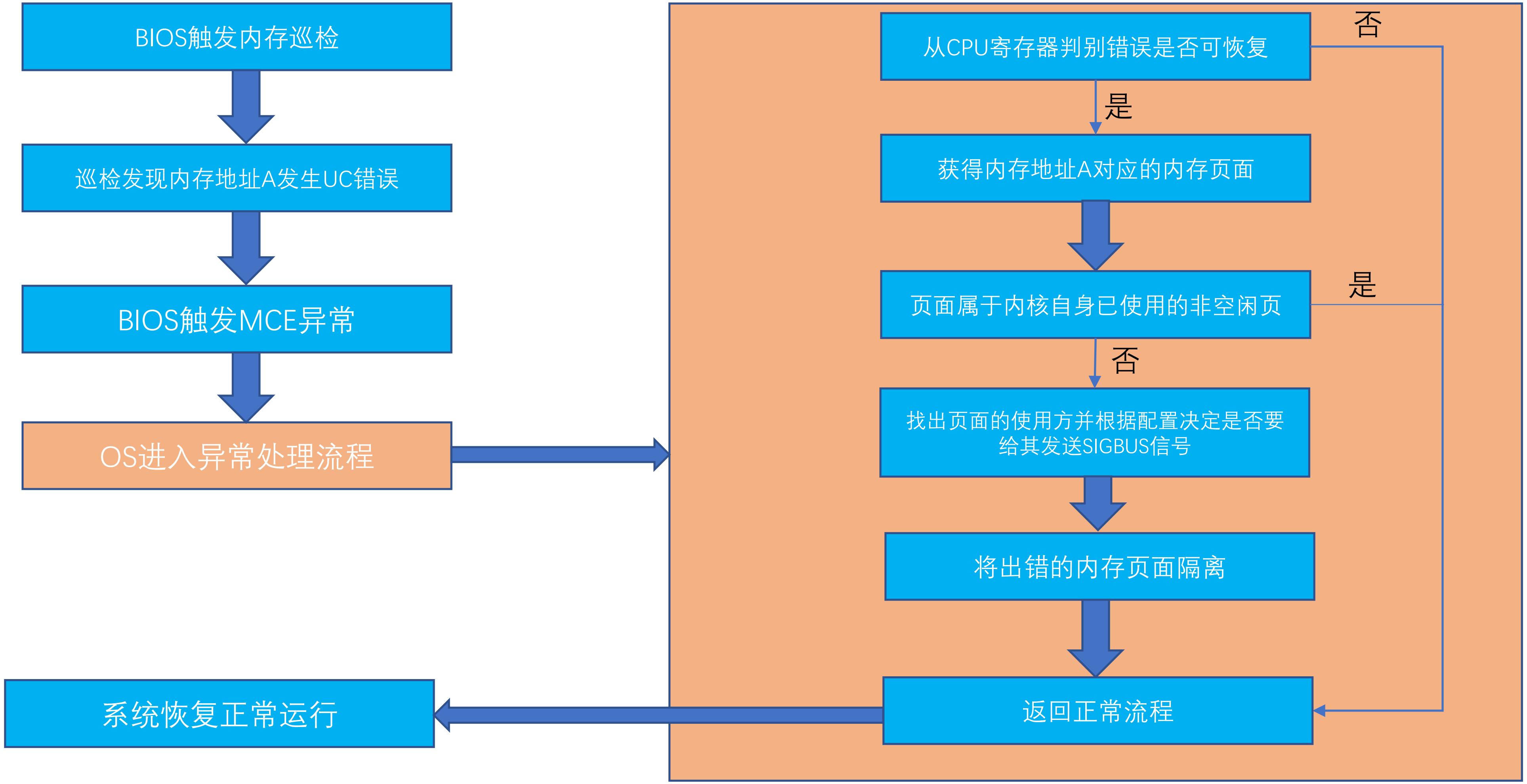
否

是

触发服务器宕机

MCA Recovery

- SRAO：



BIOS触发内存巡检
巡检发现内存地址A发生UC错误
BIOS触发MCE异常
OS进入异常处理流程
从CPU寄存器判别错误是否可恢复
否
是
获得内存地址A对应的内存页面
页面属于内核自身已使用的非空闲页
是
否
找出页面的使用方并根据配置决定是否要给其发送SIGBUS信号
将出错的内存页面隔离
返回正常流程
系统恢复正常运行

面临的挑战

硬件及底层固件层面

- 底层固件和BIOS支持不完善
- 硬件平台设计的缺陷

软件层面

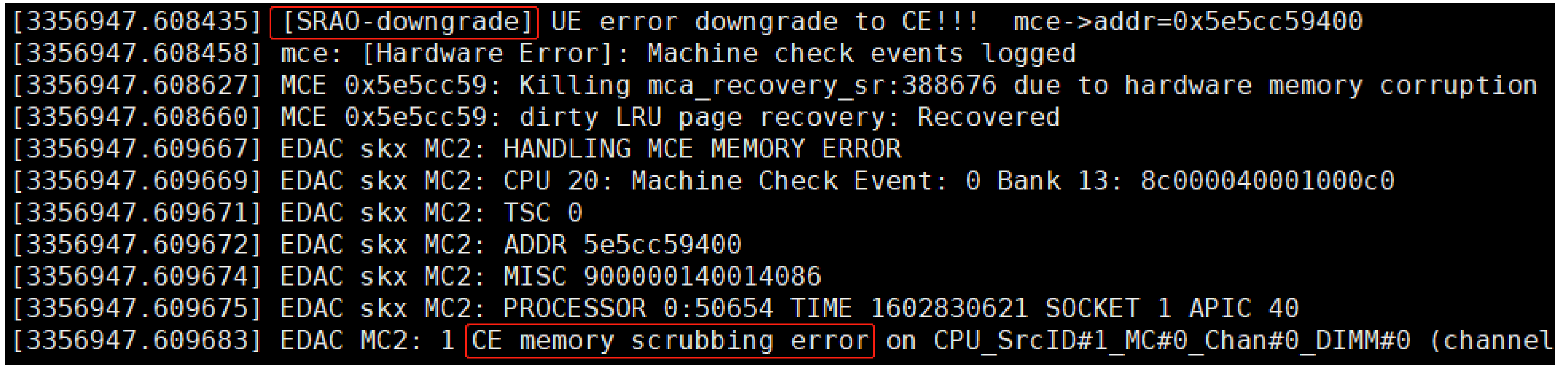
- 缺乏实用的自动化注错工具
- SRAO带了OVERFLOW、UCNA错误忽略不处理导致演变成fatal UCE
- 对1G大页隔离支持不完善
- 错误传递到VM，有些情况下会给客户带来困扰

面临的挑战

- SRAO可能造成MCE嵌套引起服务器挂死

解决方案

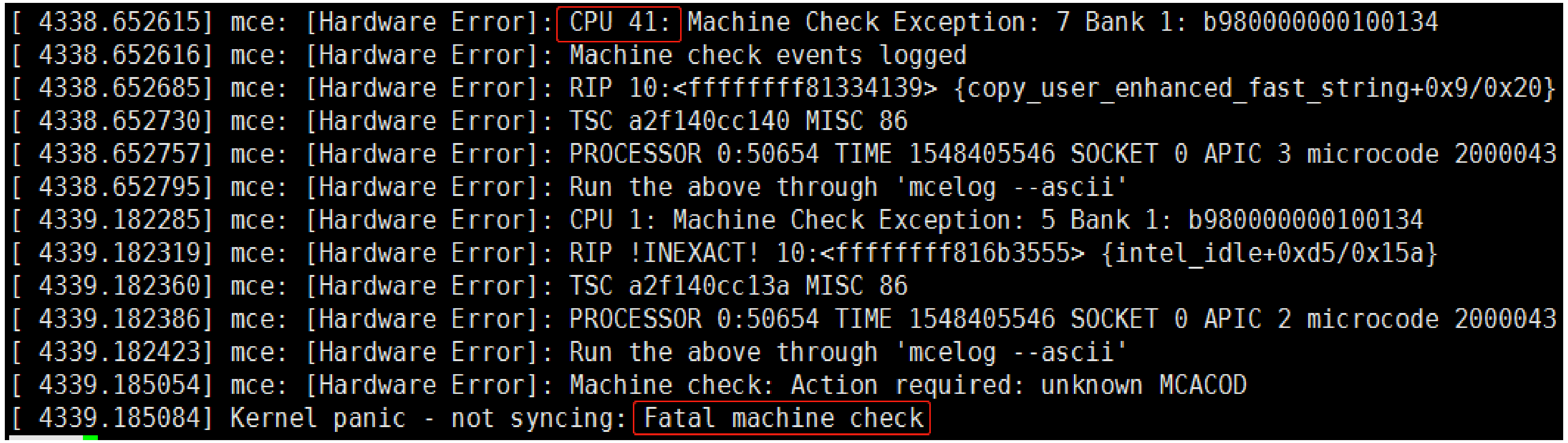
1. SRAO降级为CE，通过CMCI中断上报给OS
2. OS在CMCI中断处理程序里判别降级的情况，实现页面正确隔离



```
[3356947.608435] [SRAO-downgrade] UE error downgrade to CE!!!  mce->addr=0x5e5cc59400
[3356947.608458] mce: [Hardware Error]: Machine check events logged
[3356947.608627] MCE 0x5e5cc59: Killing mca_recovery_sr:388676 due to hardware memory corruption
[3356947.608660] MCE 0x5e5cc59: dirty LRU page recovery: Recovered
[3356947.609667] EDAC skx MC2: HANDLING MCE MEMORY ERROR
[3356947.609669] EDAC skx MC2: CPU 20: Machine Check Event: 0 Bank 13: 8c000040001000c0
[3356947.609671] EDAC skx MC2: TSC 0
[3356947.609672] EDAC skx MC2: ADDR 5e5cc59400
[3356947.609674] EDAC skx MC2: MISC 900000140014086
[3356947.609675] EDAC skx MC2: PROCESSOR 0:50654 TIME 1602830621 SOCKET 1 APIC 40
[3356947.609683] EDAC MC2: 1 CE memory scrubbing error on CPU_SrcID#1_MC#0_Chan#0_DIMM#0 (channel
```

面临的挑战

- 3万次SRAR自动注错测试过程中会概率性发生服务器宕机



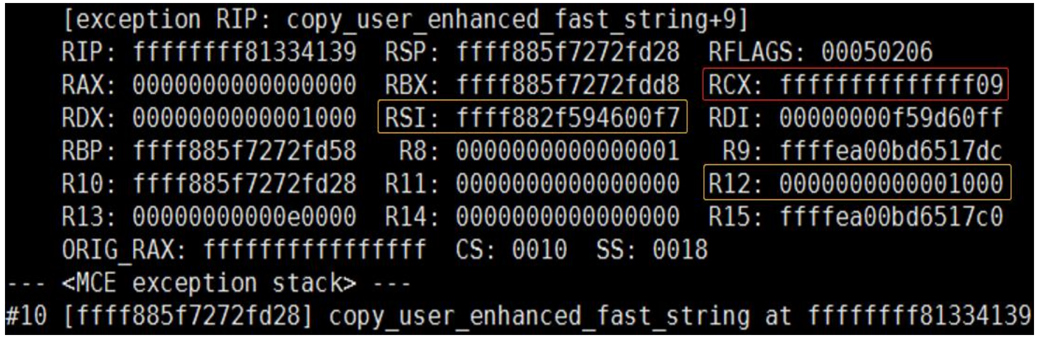
```
[ 4338.652615] mce: [Hardware Error]: CPU 41: Machine Check Exception: 7 Bank 1: b980000000100134
[ 4338.652616] mce: [Hardware Error]: Machine check events logged
[ 4338.652685] mce: [Hardware Error]: RIP 10:<ffffffff81334139> {copy_user_enhanced_fast_string+0x9/0x20}
[ 4338.652730] mce: [Hardware Error]: TSC a2f140cc140 MISC 86
[ 4338.652757] mce: [Hardware Error]: PROCESSOR 0:50654 TIME 1548405546 SOCKET 0 APIC 3 microcode 2000043
[ 4338.652795] mce: [Hardware Error]: Run the above through 'mcelog --ascii'
[ 4339.182285] mce: [Hardware Error]: CPU 1: Machine Check Exception: 5 Bank 1: b980000000100134
[ 4339.182319] mce: [Hardware Error]: RIP !INEXACT! 10:<ffffffff816b3555> {intel_idle+0xd5/0x15a}
[ 4339.182360] mce: [Hardware Error]: TSC a2f140cc13a MISC 86
[ 4339.182386] mce: [Hardware Error]: PROCESSOR 0:50654 TIME 1548405546 SOCKET 0 APIC 2 microcode 2000043
[ 4339.182423] mce: [Hardware Error]: Run the above through 'mcelog --ascii'
[ 4339.185054] mce: [Hardware Error]: Machine check: Action required: unknown MCACOD
[ 4339.185084] Kernel panic - not syncing: Fatal machine check
```

面临的挑战

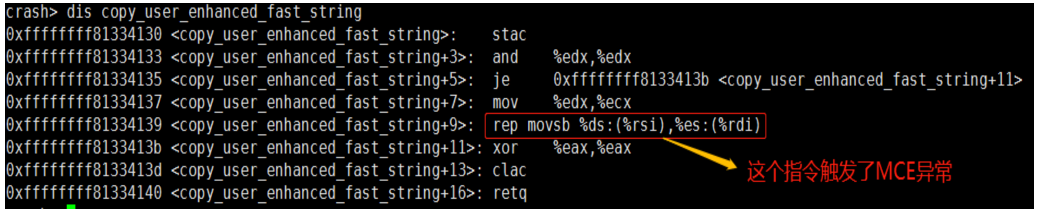
- 3万次SRAR自动注错测试过程中会概率性发生服务器宕机

问题原因

CPU缓存预取后导致底层硬件行为异常



```
    [exception RIP: copy_user_enhanced_fast_string+9]
    RIP: ffffffff81334139  RSP: ffff885f7272fd28  RFLAGS: 00050206
    RAX: 0000000000000000  RBX: ffff885f7272fdd8  RCX: ffffffffffffff09
    RDX: 0000000000001000  RSI: ffff882f594600f7  RDI: 00000000f59d60ff
    RBP: ffff885f7272fd58   R8: 0000000000000001   R9: ffffea00bd6517dc
    R10: ffff885f7272fd28  R11: 0000000000000000  R12: 0000000000001000
    R13: 00000000000e0000  R14: 0000000000000000  R15: ffffea00bd6517c0
    ORIG_RAX: ffffffffffffffff  CS: 0010  SS: 0018
--- <MCE exception stack> ---
#10 [ffff885f7272fd28] copy_user_enhanced_fast_string at ffffffff81334139
```

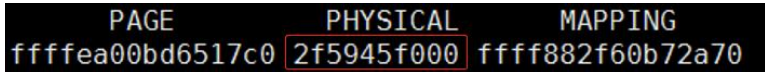


```
crash> dis copy_user_enhanced_fast_string
0xffffffff81334130 <copy_user_enhanced_fast_string>:    stac
0xffffffff81334133 <copy_user_enhanced_fast_string+3>:  and    %edx,%edx
0xffffffff81334135 <copy_user_enhanced_fast_string+5>:  je     0xffffffff8133413b <copy_user_enhanced_fast_string+11>
0xffffffff81334137 <copy_user_enhanced_fast_string+7>:  mov    %edx,%ecx
0xffffffff81334139 <copy_user_enhanced_fast_string+9>:  rep movsb %ds:(%rsi),%es:(%rdi)
0xffffffff8133413b <copy_user_enhanced_fast_string+11>: xor    %eax,%eax
0xffffffff8133413d <copy_user_enhanced_fast_string+13>: clac
0xffffffff81334140 <copy_user_enhanced_fast_string+16>: retq
```

这个指令触发了MCE异常

注错测试程序注入错误的地址是：
inject UC not fatal error to addr = 0x2f59460000

R15是另一个应用程序访问的page结构体地址，该程序访问一个页面的长度（0x1000）



```
      PAGE        PHYSICAL      MAPPING
ffffea00bd6517c0  2f5945f000  ffff882f60b72a70
```

RCX: 0xffffffffffffff09 = -247

RSI: 0xffff882f594600f7 == 物理地址 0x2f594600f7

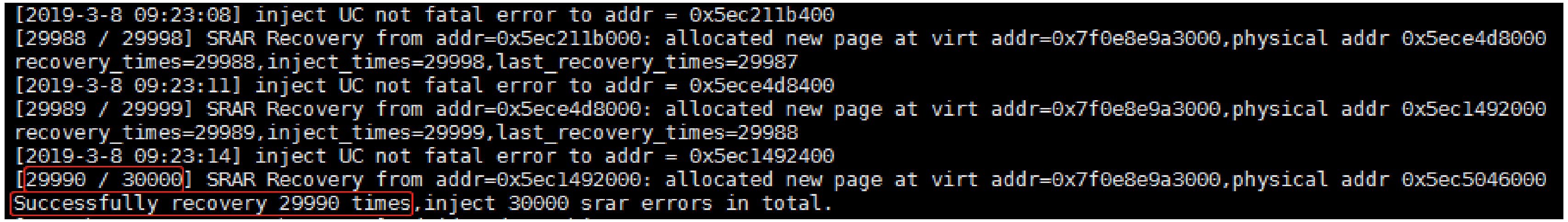
注意到：0x2f5945f000 + 0x1000 + 247 = 0x2f594600f7

解决方案

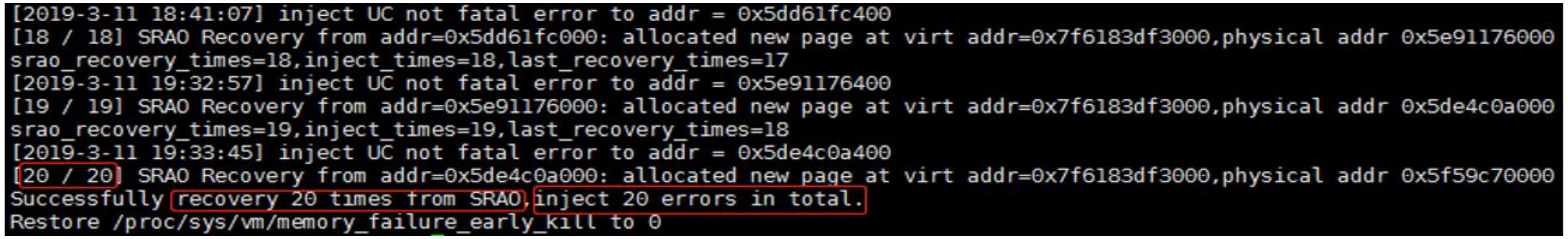
对于hwpoisoned隔离的页面，通过设置页表项PCD位，禁止该页面高速缓存

自动化注错及检测工具

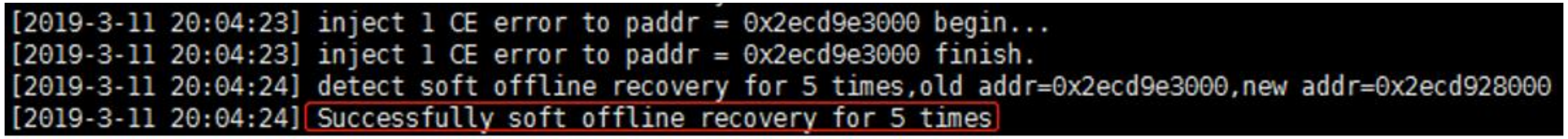
- 可实现多种MCA Recovery相关功能的自动注错和流程是否触发正常的检测



```
[2019-3-8 09:23:08] inject UC not fatal error to addr = 0x5ec211b400
[29988 / 29998] SRAR Recovery from addr=0x5ec211b000: allocated new page at virt addr=0x7f0e8e9a3000,physical addr 0x5ece4d8000
recovery_times=29988,inject_times=29998,last_recovery_times=29987
[2019-3-8 09:23:11] inject UC not fatal error to addr = 0x5ece4d8400
[29989 / 29999] SRAR Recovery from addr=0x5ece4d8000: allocated new page at virt addr=0x7f0e8e9a3000,physical addr 0x5ec1492000
recovery_times=29989,inject_times=29999,last_recovery_times=29988
[2019-3-8 09:23:14] inject UC not fatal error to addr = 0x5ec1492400
[29990 / 30000] SRAR Recovery from addr=0x5ec1492000: allocated new page at virt addr=0x7f0e8e9a3000,physical addr 0x5ec5046000
Successfully recovery 29990 times,inject 30000 srar errors in total.
```

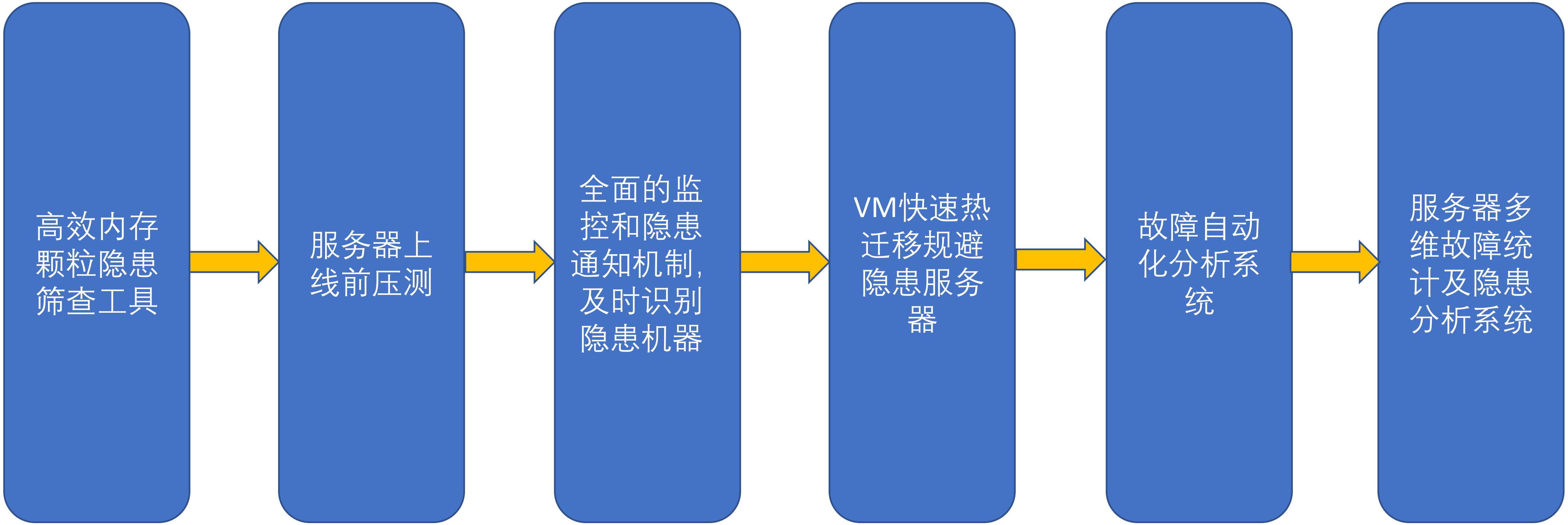


```
[2019-3-11 18:41:07] inject UC not fatal error to addr = 0x5dd61fc400
[18 / 18] SRAO Recovery from addr=0x5dd61fc000: allocated new page at virt addr=0x7f6183df3000,physical addr 0x5e91176000
srao_recovery_times=18,inject_times=18,last_recovery_times=17
[2019-3-11 19:32:57] inject UC not fatal error to addr = 0x5e91176400
[19 / 19] SRAO Recovery from addr=0x5e91176000: allocated new page at virt addr=0x7f6183df3000,physical addr 0x5de4c0a000
srao_recovery_times=19,inject_times=19,last_recovery_times=18
[2019-3-11 19:33:45] inject UC not fatal error to addr = 0x5de4c0a400
[20 / 20] SRAO Recovery from addr=0x5de4c0a000: allocated new page at virt addr=0x7f6183df3000,physical addr 0x5f59c70000
Successfully recovery 20 times from SRAO,inject 20 errors in total.
Restore /proc/sys/vm/memory_failure_early_kill to 0
```



```
[2019-3-11 20:04:23] inject 1 CE error to paddr = 0x2ecd9e3000 begin...
[2019-3-11 20:04:23] inject 1 CE error to paddr = 0x2ecd9e3000 finish.
[2019-3-11 20:04:24] detect soft offline recovery for 5 times,old addr=0x2ecd9e3000,new addr=0x2ecd928000
[2019-3-11 20:04:24] Successfully soft offline recovery for 5 times
```

腾讯云现网运维举措



高效内存颗粒隐患筛查工具

服务器上线前压测

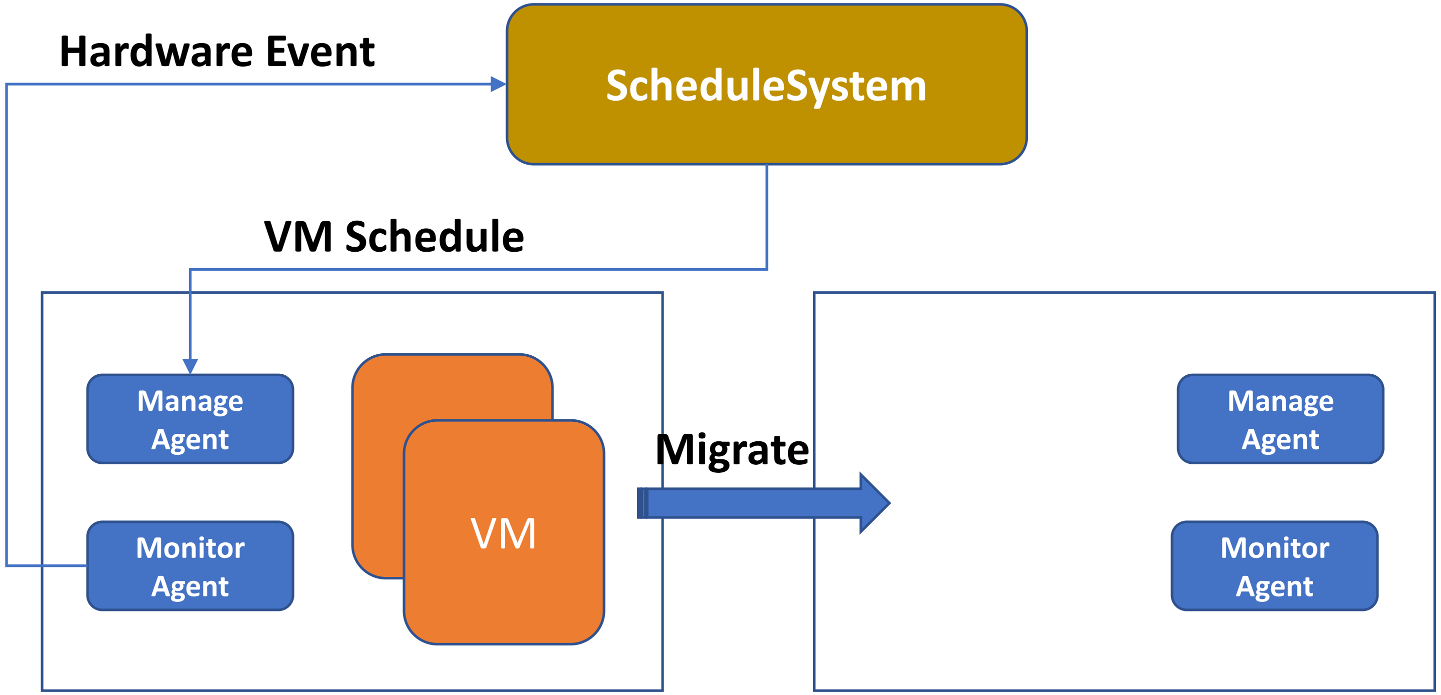
全面的监控和隐患通知机制，及时识别隐患机器

VM快速热迁移规避隐患服务器

故障自动化分析系统

服务器多维故障统计及隐患分析系统

快速热迁移



结合热迁移技术快速隐患规避

1. 监控宿主机硬件事件，识别硬件异常信息
2. 发起VM调度，热迁移主动规避硬件隐患，ms级切换，不影响VM业务

改善数据

- 目前腾讯云英特尔® 至强® 可扩展平台上，内存UCE故障约有50%可以通过MCA Recovery来予以容错避免宕机
- 腾讯云宕机故障中，内存故障的占比从50%以上下降至23%
- 腾讯云服务器月度硬件故障宕机率下降至原来的一半以下

MCA Recovery失效的主要因素

- CPU内部出现了PCC（Processor Context Corruption），导致fatal UCE
- CPU的Cbo模块 Tor Table 发生 3-strike timeout，触发了CPU IERR
- 故障发生在内核自身使用的内存上或不可恢复的内核函数路径上

下一步计划

- 深入挖掘上述失效的因素，进一步提高MCA Recovery生效率
- 内存CE错误的细化解析及分析

Reference

- https://www.intel.com/content/www/us/en/software/reduce-server-crash-rate-tencent-paper.html