# Machine checks on i386/x86-64

Andi Kleen, SuSE Labs
ak@suse.de

# What is a machine check?

- Hardware error
  - Internal errors, Memory, Cache, IO, Busses
  - But users have a hard time to recognize this
- Hardware is (mostly) error correcting
  - 64bit is worse: more DIMMs, more bit flips
  - Smaller/more transistors
  - NMI, Thermal
- Can be panic or a logged event
- Talking only about x86-64 here, a bit i386
  - IA64, S390, PPC64 handle machine checks differently

# The OS is just the messenger

- Caused by hardware
  - But always blamed on the software of course
- Sometimes drivers/BIOS can misprogram hardware
- Challenge is to explain to customers that their hardware is broken

# General classification

- Uncorrected
  - Console log or a jpg
  - Don’t make it to disk
  - Sometimes available in BIOS event log afterwards
  - Sometimes can be recovered by kernel after reboot
  - Run through mcelog --ascii first
- Corrected error but logged
  - Logged in a binary log (/dev/mcelog)
  - mcelog cronjob decodes them into /var/log/mcelog
  - On i386 they just go to /var/log/messages
  - Nothing really bad happened, but if it happens often hardware will likely fail

# A live example

2.4 AMD

```
kernel: Northbridge Machine Check exception = b60ea00100000813 0
ecc error
    link number 0
    err cpu1
    uncorrected ecc error
    processor context corrupt
    error address valid
    error enable
    error uncorrected
    previous error lost
    error address 00000001826ac018
Address: 00000001826ac018
CPU 1: Machine Check Exception: 0000000000000000
Kernel panic: Unable to continue
```

# Another example

2.6 AMD fatal

```
HARDWARE ERROR
CPU 0: Machine Check Exception:                4 Bank 4: b605200100000813
TSC 24291bbb8104 ADDR fbf2c068
This is not a software problem!
Run through mcelog --ascii to decode and contact your hardware vendor
Kernel panic - not syncing: Machine check
```

# Intel example

2.6 Fatal

```
CPU 1: Machine Check Exception: 0000000000000004
Bank 1: b200000000000115
Kernel panic: CPU context corrupt
```

# Another Intel example

Non fatal mcelog entry

```
STATUS 9000020110800e0f MCGSTATUS 2
MCE 0
HARDWARE ERROR. This is *NOT* a software problem!
Please contact your hardware vendor
CPU 21 BANK 4 TSC 20e7a6ec72de3
RIP 00:ffffffff801098e7
MCG status:EIPV
MCi status:
Error enabled
MCA:BUS Generic Generic Generic Other-transaction Request-timeout Error
Model:Pad address glitch
```

# Oops after machine check

(shouldn’t happen)

```
HARDWARE ERROR
CPU 1: Machine Check Exception:                6 Bank 0: be0003001008081f
RIP 00:<ffffffff80175d0a>
TSC 15433f562a4 ADDR 1d5957540 MISC c4000e0f01a60e
This is not a software problem!
Run through mcelog --ascii to decode and contact your hardware vendor
Kernel panic - not syncing: Machine check
 NMI Watchdog detected LOCKUP on CPU 0
CPU 0
Modules linked in: binfmt_misc nfs lockd nfs_acl sunrpc autofs4 edd ipv6 af_packet button battery ac loop dm_mod generic e1000 ide_cd cdrom uhci_hcd ehci_hcd i2c_i801 usbcore i2c_core shpchp pci_hotplug floppy reiserfs fan thermal sg processor ata_piix libata piix sd_mod scsi_mod ide_disk ide_core
Pid: 18329, comm: rpm Tainted: G   M   2.6.16-rc5-git5-20060302131619-smp #1
RIP: 0010:[<ffffffff80116c37>] <ffffffff80116c37>{__smp_call_function+102}
RSP: 0000:ffffffff803bdcb8  EFLAGS: 00000093
RAX: 0000000000000004 RBX: 0000000000000000 RCX: 0000000000000000
RDX: 0000000000000080 RSI: 000000000000007f RDI: 0000000000000040
RBP: 0000000000000000 R08: ffffffff803bdc88 R09: 0000000000000002
R10: 0000000000000008 R11: 0000000000000002 R12: 0000000000000007
R13: ffffffff80116ba8 R14: 0000000000000001 R15: 0000000000000000
FS:  0000000000000000(0000) GS:ffffffff80444000(0063) knlGS:00000000b7db36b0
CS:  0010 DS: 002b ES: 002b CR0: 000000008005003b
CR2: 00000000b7f5b8b0 CR3: 000000007581e000 CR4: 00000000000006e0
Process rpm (pid: 18329, threadinfo ffff810078528000, task ffff8101d51b07f0)
Stack: ffffffff80116ba8 0000000000000000 0000015400000004 ffffffff00000000
       000000000000000d ffffffff802f375c ffffffff8033cfc0 0000015433f5586a
       ffffffff802f375c ffffffff80116c8b
Call Trace: <#MC> <ffffffff80116ba8>{smp_really_stop_cpu+0}
       <ffffffff80116c8b>{smp_send_stop+47} <ffffffff80130a3e>{panic+169}
       <ffffffff802d2444>{_spin_trylock+9} <ffffffff802d2bbf>{oops_begin+93}
       <ffffffff80112562>{mce_log+0} <ffffffff801128cb>{do_machine_check+730}
       <ffffffff8010b9cb>{machine_check+127} <EOE><1>Unable to handle kernel paging request at 00000000c0000000 RIP:
<ffffffff8010bd48>{show_trace+443}
PGD 7f6b3067 PUD 785bf067 PMD 0
```

# User interface I

```
> find /sys | grep machine
/sys/devices/system/machinecheck
/sys/devices/system/machinecheck/machinecheck7
/sys/devices/system/machinecheck/machinecheck7/check_interval
/sys/devices/system/machinecheck/machinecheck7/tolerant
/sys/devices/system/machinecheck/machinecheck7/bank4ctl
/sys/devices/system/machinecheck/machinecheck7/bank3ctl
/sys/devices/system/machinecheck/machinecheck7/bank2ctl
/sys/devices/system/machinecheck/machinecheck7/bank1ctl
/sys/devices/system/machinecheck/machinecheck7/bank0ctl
/sys/devices/system/machinecheck/machinecheck6
/sys/devices/system/machinecheck/machinecheck6/check_interval
...
```

# User Interface II

- Cronjob with mcelog
  - Decodes /dev/mcelog into /var/log/mcelog
- tolerance level
  - 0 always panic
  - 1 panic if deadlock
  - 2 try to avoid deadlock
- oops=panic, panic=timeout, mce=off

# AMD Opteron specific MCEs

- When panic run through mcelog --ascii first
- Banks
  - 0-3 CPU internal (DC, IC, BU, LS)
  - 4 Northbridge:  only really interesting one
- Banks 0-3
  - Ask customer to check cooling or their CPU might be just dying
  - Or power supply/VRM broken

# AMD Opteron Bank 4

Northbridge

- □ 0 / CorrEccEn single bit flip in memory
  - ○ If they get a lot DIMM might be dying or cooling/PS/VRM bad
- □ 1 / UnCorrEccEn
  - ○ Can be triple fault or panic
  - ○ memtest86 and exchange DIMMs/VRMs
- □ 12 Watchdog
  - ○ Could be plugin card/BIOS/driver
  - ○ Or another CPU is dead or mainboard flakey.

# More AMD notes

- Bank 4 GART table walk is harmless
  - SLES10 filters it out
- On some old systems bootup leaves bogus events in the log
  - Newer kernels filter that out
  - Can be overwritten with mce=bootlog to catch old machine checks from before reboot
- RevF: DRAM thresh pseudo events
  - Only if configured using sysfs

# Intel

- Banks depends on the family
  - P4/Xeon
  - P3/P-M
  - Conroe/Woodcrest will have new ones
- Thermal events cause pseudo entry in mcelog
  - On i386 it’s a log entry in the kernel log.
  - CPU is overheating.

# Intel II

- Only BUS errors are interesting normally
  - Non bus is CPU internal
- Don’t report addresses normally
- We don’t get a lot of reports in general
  - Good or bad sign?
- For more details see Appendix E in IA32 manual vol 3

# Tainted bits in Oopses

```
*  ’P’ - Proprietary module has been loaded.
*  ’F’ - Module has been forcibly loaded.
*  ’S’ - SMP with CPUs not designed for SMP.
*  ’R’ - User forced a module unload.
*  ’M’ - Machine had a machine check experience.
*  ’B’ - System has hit bad_page.
*  ’U’ - Unsupported module loaded.
```

# Differences on i386

- Logs to normal /var/log/messages
- Doesn’t decode by default
- Doesn’t have tolerance levels
- Generally less reliable

# Differences on 2.4/x86-64

□ mcelog doesn’t exist

□ Decoded logs go into normal kernel log

  ○ /var/log/messages

□ Doesn’t filter out some bogus one

# Other tools

- mcelog --cputype --ascii
  - cputype: --k8 or --p4
  - Just paste it in
- parsemce for i386

# Tips and tricks

- When you see an oops after the machine check
  - Sometimes it’s a real bug, sometimes it is unavoidable
  - Put it in bugzilla (but it might be not fixable)
  - But explain the customer that fixing the oops won’t fix the original problem

# Tips and tricks II

- mce=off
  - Or disable the event in sysfs
  - Only recommended in emergencies
  - Cause silent data corruption
  - Hardware needs fixing.
- Corrected with PCC
  - Disable the subevent in sysfs
  - Will be still logged,  but no panics
  - BIOS/kernel do that for known offenders
  - If there is a pattern report please

# Tips and tricks III

- Low tolerance level / panic=XXX for clusters

- mce=bootlog

- NMI can have multiple causes
  - User presses button
  - Can be caused by some PCI errors

# mcelog --dmi

New in SLES10

- Run on the same machine and with same configuration
  - As root
  - You need an address (ADDR xxxx)
- Gives label(s) on the mainboard
- Problems
  - Unreliable due to BIOS bugs
  - Often gives multiple DIMMs because of interleaving or bad BIOS

# mcelog --dmi demo

```
# echo "ADDR 0x123456" | mcelog --dmi --ascii
WARNING: with --dmi mcelog --ascii must run on the same machine with the
    same BIOS/memory configuration as where the machine check occurred.
Resolving address 123456 using SMBIOS
WARNING: SMBIOS data is often unreliable. Take with a grain of salt!
DRAM DIMM Synchronous Width 128 Data Width 64 Size 1 GB
Device Locator: H0_DIMM3
Bank Locator: Bank 4
```

# More information

- □ Vendor documentation
  - ○ AMD BIOS and Kernel Developers Guide
  - ○ IA32 Intel Architecture SW Developers manual: Volume 3
- □ parsemce for i386
- □ LinuxKongress paper
  - ○ ~ak/pub/mce.pdf