Crashes, Panics and Other Oddities

Imed Chihi <imed dot chihi at gee mail dot com>
February 2008

Agenda

- "Defining" some terms
- Analogy with User Space
- The BUG() Macro
- Bad Pointer Handling
- The NMI Watchdog
- Machine Check Exceptions
- EDAC
- Other Hardware Reports
- Pseudo Hangs
- The Out of Memory Killer

“Defining” some terms

- **Crash**
  - A very generic term used usually to say that the system has come to halt and no progress is observed. The system seems unresponsive or has already rebooted.
- **Panic**
  - A *voluntary* halt to all system activity when an abnormal situation is detected by the kernel.
- **Oops**
  - Similar to panics, but the kernel deems that the situation is not hopeless, so it kills the offending process and continues.

“Defining” some terms

- **BUG()**
  - Similar to a panic, but is called by intentional code meant to check abnormal conditions.
- **Hang**
  - The system does not seem to be making any progress. System does not respond to normal user interaction. Hangs can be soft or hard.

Analogy with user-space

`signals`	~	`interrupts`
`core`	~	`vmcore`
`segfault`	~	`panic`
`gdb`	~	`crash`

The BUG() macro

- Called by kernel code when an abnormal situation is seen
- Typically indicates a programming error when triggered
- The calling code is intentional code written by the developer
- Calls look like:

```
BUG_ON(in_interrupt());
```

- Inserts an invalid operand (0x0000) to serve as a landmark by the trap handler
- Output looks like:

```
Kernel BUG at spinlock:118
invalid operand: 0000 [1] SMP
CPU 0
```

Bad Pointer Handling

- Typically indicates a programming error
- Typically appear as:

```
NULL pointer dereference at 0x1122334455667788 ..
```

or maybe ..

```
Unable to handle kernel paging request at virtual
address 0x11223344
```

- Detection of those situations is hardware assisted (MMU)
- Typically due to:
  - NULL pointer dereference
  - Accessing a non-canonical address on AMD Opteron
  - Accessing an illegal address on this architecture

NMI watchdog

- A hardware mechanism available on modern hardware (APIC) which detects CPU lockups
- When active, the “NMI” count should keep increasing in `/proc/interrupts`
- When a CPU fails to acknowledge an NMI interrupt after some time, the hardware triggers an interrupt and the corresponding handler is executed. The handler would typically call panic()
- Typically indicates a deadlock situation: a running process attempts to a acquire a spinlock which is never granted

Machine Check Exceptions

- Component failures detected and reported by the hardware via an exception
- Typically looks like:

```
kernel: CPU 0: Machine Check Exception:
    4 Bank 0: b278c00000000175
kernel: TSC 4d9eab664a9a60
kernel: Kernel panic - not syncing: Machine check
```

- To decode, pipe entire line through `mcelog --ascii`
- Always indicates a hardware problem

EDAC

- *Error Detection and Correction* (aka BlueSmoke) is a hardware mechanism to detect and report memory chip and PCI transfer errors
- Introduced in RHEL 4 Update 3 for intel chips and in RHEL 4.5 for AMD chips
- Reported in `/sys/devices/system/edac/{mc/,pci}` and logged by the kernel as:

```
EDAC MC0: CE page 0x283, offset 0xce0, grain 8, syndrome
0x6ec3, row 0, channel 1 "DIMM_B1": amd76x_edac
```

Other Hardware Reports

- Machine Check Architecture. I have never seen this on i386 and x86_64.
- Machine Specific Register
- NMI notifications about ECC and other hardware problems. Typically look like:

```
Uhhuh. NMI received. Dazed and confused, but trying to continue
You probably have a hardware problem with your RAM chips
Uhhuh. NMI received for unknown reason 32. Dazed and confused,
but trying to continue.
Do you have a strange power saving mode enabled?
```

Pseudo Hangs

- In certain situations, the system **appears** to be hang, but some progress is being made
- Those situations include:
  - Thrashing – continuous swapping with close to no useful processing done
  - Lower zone starvation – on i386 the low memory has a special significance and the system may “hang” even when there's plenty of free memory
  - Memory starvation in one node in a NUMA system
- Hangs which are not detected by the hardware are trickier to debug:
  - Use [sysrq + t] to collect process stack traces when possible
  - Enable the NMI watchdog which should detect those situations
  - Run hardware diagnostics when it's a hard hang: memtest86, hp diagnostics

The Out of Memory Killer

- In certain memory starvation cases, the OOM killer is triggered to force the release of some memory by killing a “suitable” process
- In severe starvation cases, the OOM killer may have to panic the system when no killable processes are found:

```
Kernel panic - not syncing: Out of memory and no
killable processes...
```