Long Term Libera System Stability

Michael Abbott

Diamond Light Source
michael.abbott@diamond.ac.uk

Libera Workshop 2011

Long Term Libera Faults

- Fan failures
  - Initial crop of broken fans seems to have been flushed out.
  - Now getting fan failures due to failure of transistor on motherboard. Four fan failures in last couple of months.
  - Easy to to fix: just replace the transistor.

Long Term Libera Faults

- Fan failures
  - Initial crop of broken fans seems to have been flushed out.
  - Now getting fan failures due to failure of transistor on motherboard. Four fan failures in last couple of months.
  - Easy to to fix: just replace the transistor.
- Spontaneous Libera crashes
  - Completely elusive, the topic of this talk.

By “Libera crash” I mean any soft system failure resulting in loss of controller communication, normally associated with complete failure of the embedded Linux system.

Spontaneous Libera Crashes

- Currently seeing approximately one Libera crash every few weeks from a population of around 200 machines.
- System disappears from the Ethernet network, but the communication controller network continues working.
- System synchronisation is eventually lost due to loss of clock synchronisation daemon, so have to restart Libera.
- After power-cycling, the Libera works normally again.
- Need to take precautions to avoid dropping interlock during power-cycling, but this is now a routine procedure.

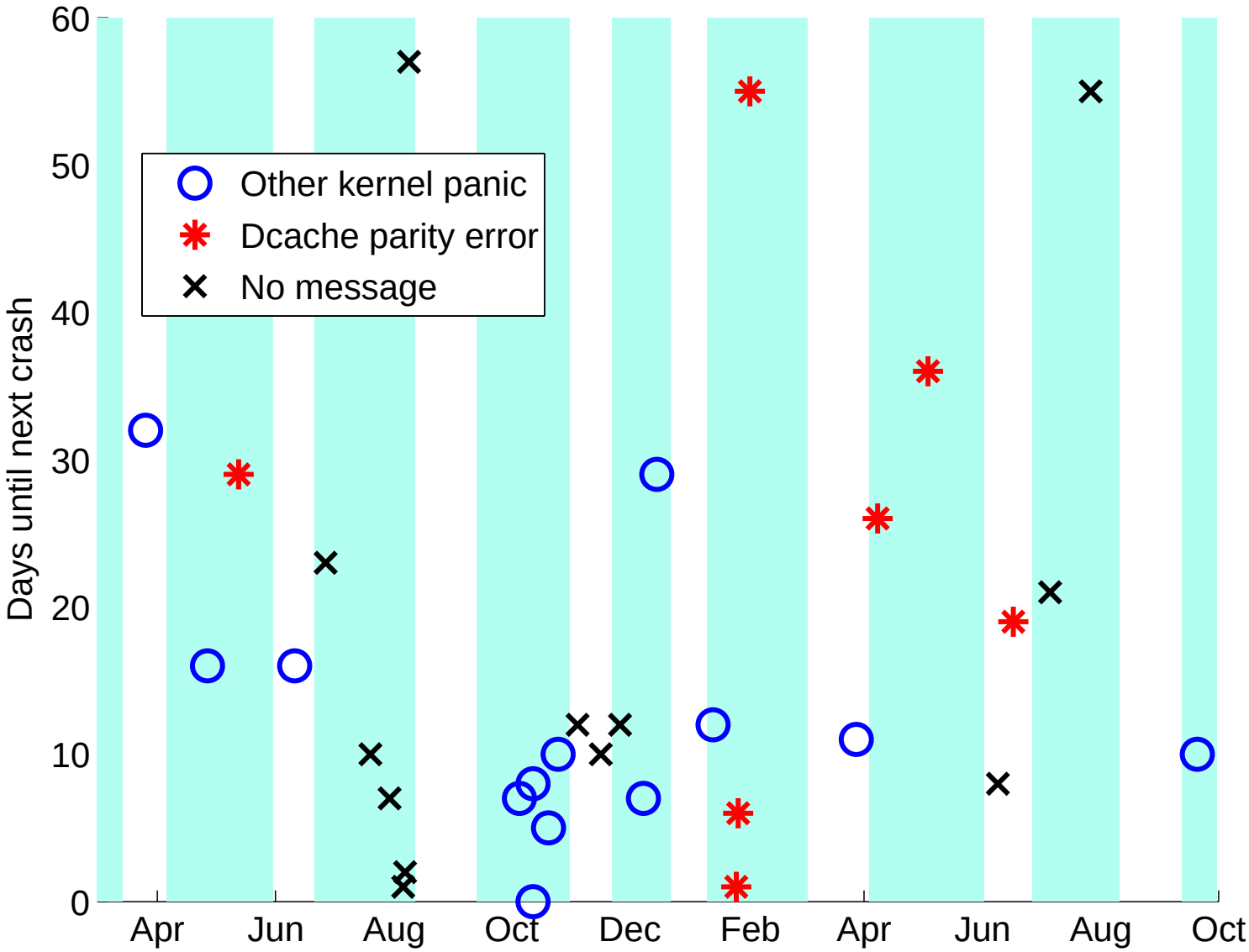
Reports from other Synchrotrons

Institution	No. Liberas	Crashes seen?	I/locks?	Serial Console?
Diamond	210	32 in 1½ y	174	Logged
ESRF	224	10–20/y	8	No
DESY	227	Some	96	No
Taiwan Photon Source	60	PM only	–	No
SINAP	163	10–20/y	all	No
ALBA	162	None	–	No
Soleil	147	2/y	half	No
Australian Synchrotron	98	A couple	limited	No
Elettra	100	None	IDs only	No

Logging Libera System Crashes

- Since early 2010 at Diamond all serial consoles are connected to logging servers and all logging output is redirected to the console, now recording detailed logs of every crash.
- A variety of failure error messages seen:
  - 10 crashes with no message
  - 7 crashes reporting “dcache parity error”
  - 15 miscellaneous kernel oops messages
- Please consider logging your lockups, dropouts or crashes!

Intervals between Crashes (2010–2011)



Searching for Patterns

- Worrying recent flush of “dcache parity error” reports. Other crashes could be due to software errors, but this sounds like a real hardware fault.
- A cluster of crashes in the second half of last year seem to be correlated with “First Turn” processing being enabled . . . but alas we don’t have adequate records.
- Many of the 14 kernel panic crash dumps have been analysed.

Typical console crash dump

```
Unable to handle kernel NULL pointer dereference at virtual address 00000000
pgd = c2d70000
[00000000] *pgd=a2d86031, *pte=00000000, *ppte=00000000
Internal error: Oops: 0 [#1]
Modules linked in: libera msp [last unloaded: msp]
CPU: 0    Not tainted  (2.6.30.9-dls #1)
PC is at 0x0
LR is at 0xfffc0001
pc : [<00000000>]    lr : [<fffc0001>]    psr: 30000013
sp : c2c83f58  ip : 00000018  fp : 00002030
r10: bf3403ac  r9 : 00000400  r8 : 40b52564
r7 : bf340398  r6 : c2c82000  r5 : 40b509b4  r4 : c2c83f80
r3 : 7ffff000  r2 : 81b5256b  r1 : bf341f78  r0 : 7ffff000
Flags: nzCV  IRQs on  FIQs on  Mode SVC_32  ISA ARM  Segment user
Control: 0000397f  Table: a2d70000  DAC: 00000015
Process ioc (pid: 1255, stack limit = 0xc2c82270)
Stack: (0xc2c83f58 to 0xc2c84000)
3f40:                                                   00000000 c00741ac
3f60: ffffffff c2c82000 00000000 00000000 00000000 c38802c0 00002000 c00742e4
3f80: 00000000 00000000 00000000 00000001 00000000 0013d308 40b509b4 00000003
3fa0: c001efc8 c001ee20 00000000 0013d308 00000006 40b509b4 00002000 000195f4
3fc0: 00000000 0013d308 40b509b4 00000003 00128638 001286c8 0012860c 00000000
3fe0: 00000000 40b4c958 400337b4 40033f74 80000010 00000006 00000000 00000000
Code: bad PC value.
```

Oops Log Analysis

- Kernel register dump and traceback quite detailed, can be analysed with corresponding kernel symbols.
  Detailed analysis is very laborious, have done this for nine crashes.
- One hot spot in particular is *very* conspicuous, exactly the same spot occurs in at least five crashes:

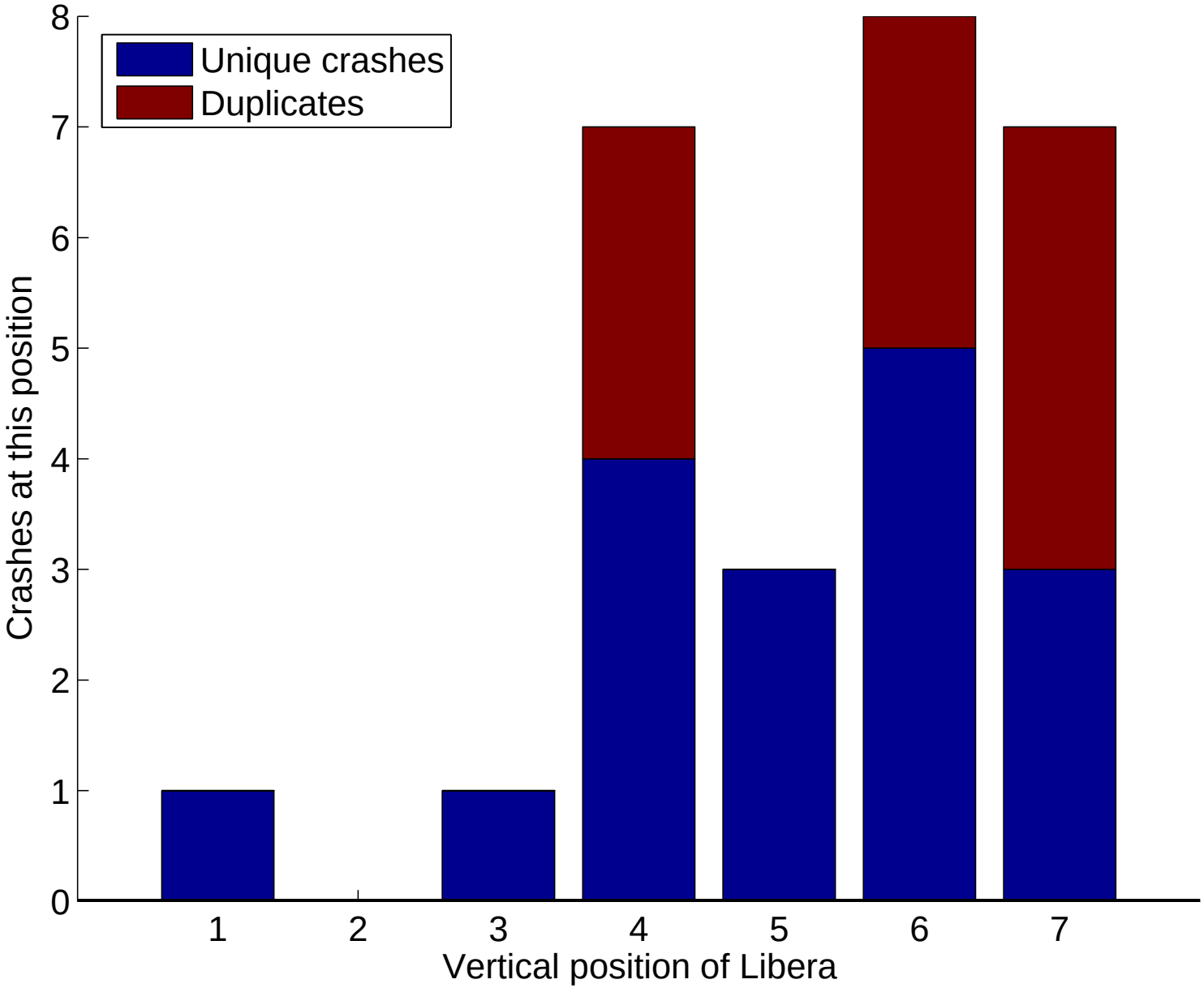
  Copying raw ADC data from kernel driver to user space: `__copy_to_user` called from `libera_adc_read`.

- Also network traffic is a hot spot for crashes.

Differences at Diamond

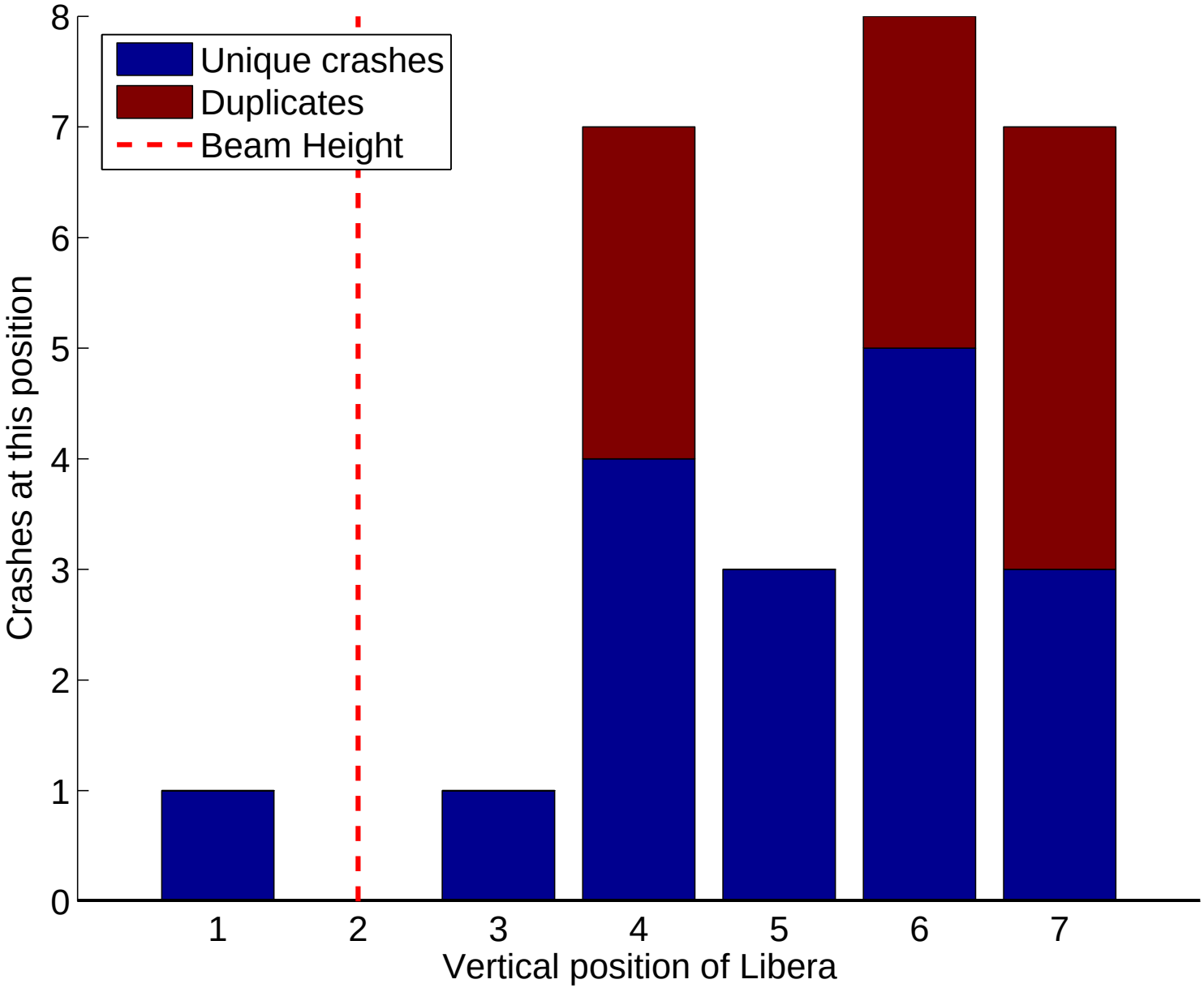
- Diamond specific FPGA. No impact on likely error rate.
- Diamond build of driver. The Diamond version of the driver is only slightly different from the original i-Tech driver, unlikely to account for differences.
- Diamond kernel: running Linux 2.6.30.9 with minimal XCEP patches in comparison with 2.6.20.14 on i-Tech Libera. Can't be ruled out as a contribution.
- Diamond specific EPICS driver. The DLS EPICS driver is continually calling for data from Libera, this high level of system activity is the most likely reason for our high rate of crashes.

Crash Frequency against Position in Rack



amond

Correlated with Beam Radiation?



amond

Possible Reasons for Crashes

- Bugs in Libera device driver.
  Entirely possible . . . but proving remarkably hard to pin down, doesn't explain everything.
- Bug in SMC network driver.
  I have a small amount of money on this one. The code is nasty. But I don't believe it.
- Commodity XScale processor chip showing its limits.
  Just how reliable was the XScale processor designed to be? Maybe we're at the operating limits for this device — the "dcache parity errors" suggest so.
- Something else?