Soft Errors, Silent Data Corruption, and Exascale Computing

Sarah Michalak

Statistical Sciences Group

Los Alamos National Laboratory

michalak@lanl.gov

Collaborators

- Sean Blanchard
- Carolyn Connor
- John Daly
- Andy DuBois
- Dave DuBois
- Kevin Harris
- Nick Hengartner
- Andrea Manuzzato
- Dave Modl
- Lisa Moore
- Heather Quinn
- Bill Rust
- Ruben Salazar
- Andrew Shewmaker
- Curt Storlie
- Bruce Takala
- Steve Wender
- And many others…

LANL HPC Platforms and Users

- **Capability and capacity systems at LANL, Lawrence Livermore National Laboratory, and Sandia National Laboratory are available to LANL scientists**
- **Used for large scientific calculations:**
  - Capability: large calculations (4k-20k processes) that run for weeks to months
  - Capacity: small/medium calculations (10-40k processes) that can run for days
- **Some HPC platforms are at the cutting edge in terms of scale/technology**

Soft Errors & HPC Platforms

- **Soft Error:** An unintended change in the state of an electronic device that alters the information that it stores without destroying its functionality, e.g. a bit flip caused by a cosmic-ray-induced neutron. (Hengartner et al., 2008)
- **LANL Experience: ASC Q Supercomputer (2nd fastest: June 2003)**
  - More single-node (CPU) failures than predicted
  - Hypothesis: caused by cosmic-ray-induced neutrons
  - BTAG SRAM had parity, but not ECC; BTAG parity errors thus lead to node crashes
  - Neutron-beam testing at LANSCE consistent with field data
    - Experimental data (3 datasets): 9.8; 17.4; 22.6 failures/wk*
    - Field data: 24 BTAG parity errors/wk; 27.7 CPU failures/wk*
  - Single node failures can have big impacts in large jobs
    - BTAG parity errors: ~1/2 of all failures; ~2/3 of hardware failures*
  - Successful system use: optimized checkpoint-restart (Daly, 2004 & 2005); spare node in allocation for immediate restart (Daly, 2003)

* Michalak et al., 2005

Silent Data Corruption (SDC)

- **"SDC occurs when incorrect data is delivered by a computing system to the user without any error being logged" Cristian Constantinescu (AMD)**
- **Examples of SDC:**
  - Teraflops Supercomputer [Constantinescu, 2000]
    - Processor ECC disabled (firmware bug)
  - UWI Study [Kola et al (2005)]
    - Faults in storage and in staging/compute nodes lead to SDC
    - Some root causes developed over time, so not observed during initial testing
  - CERN FS Study [Panzer-Steindel (2007)]
    - Disk Errors: write, read, compare 2 GB file
      - Every 2 hrs for 5 weeks on ~3000 nodes ➔ 500 errors on 100 nodes
    - Recalculate and compare checksum for 33,700 files (~8.7 TB)
      - 22 mismatches ➔ one bad file in 1500
  - Storage Stack Study [Bairavasundaram et al (2008)]
    - Checksum mismatches, identity discrepancies, parity inconsistencies

SDC: What We Know or Think We Know

- **SDC can affect networks, nodes, and file systems**
- **SDC has multiple causes:** [Constantinescu 2008]
  - Temperature/voltage fluctuations, particles, manufacturing residues, oxide breakdown, electro-static discharge…
- **SDC will likely be more prevalent in new technologies** [Borkar 2009; Pan et al 2008; Constantinescu 2006]
  - Increased frequency, transistor counts, soft errors, and noise levels
  - Decreased feature sizes and supply voltage
- **For a given device susceptibility, a larger platform is more likely to be affected**
- **SDC could affect scientific desktop computing**
  - Laptops/desktops used for scientific computation may be equivalent to a cluster
- **Applications have differing susceptibility to faults that could lead to SDC**
  - Not all undetected faults will lead to SDC

SDC Research at LANL

Goal:

- **Protect user calculations and data**

Activities:

- **Laboratory Testing (preliminary)**
- **Testing of Platforms**
- **Neutron-Beam Testing**

Laboratory Testing

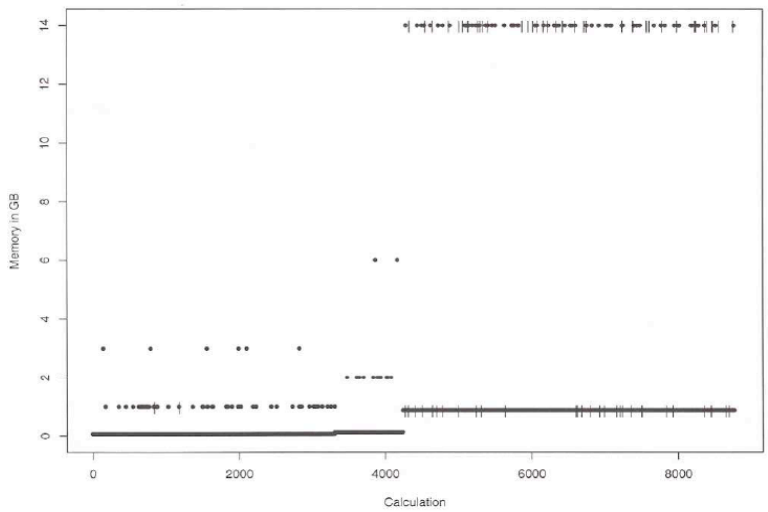
- **Goals:**
  - Understand behavior under diverse operating conditions; inform monitoring programs
  - Verify vendor information
- **Experimental Protocol:**
  - Dual-core 65nm processor on a high-performance overclocking motherboard
    - Easy to manipulate frequency, voltage and fan speeds
  - C version of Linpack sized to run out of L1 and L2 caches
    - Run on both cores simultaneously
  - Manipulate frequency, voltage and fan speeds while running Linpack
    - Limited control of temperature via adjusting fan speed
- **Outcomes:**
  - Multiple forms of SDC: incorrect Linpack results; erroneous timestamps/frequency
  - Other failures: early program exit, program crash, and system crash

Platform Testing

- **Specifics:**
  - OSATSDC Linpack-based software available
    - Systems that use LSF/Moab
    - Configure qos/queue to allow pre-emption on production platforms
    - User-determined Linpack job mix (number of pe's, problem size, …)
  - Crisscross interconnect testing code
    - 6 patterns; all core combinations on nodes under test
  - OSATSDC & Crisscross: parsing and reporting of results with automatic email on failure
  - Goal of no user impact (pre-emptible jobs on otherwise-idle nodes)
- **Other issues detected:** Grep (dynamic lib failing to load); slow nodes (4x typical)
- **Production Platforms:**
  - Rolling out Linpack-based and crisscross-based testing on production platforms
- **Decommissioned Platforms:**
  - Test at nominal or manipulate temperature/voltage
  - SDC observed on three platforms (also incorrect temp readings)

Decommissioned Platform Testing

- **Platform 1: 70 incorrect Linpack calculations; all involve 1 node**



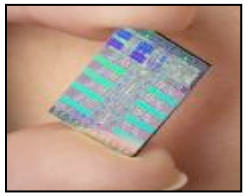
- **Platform 2: 2 incorrect Linpack calculations: one-node jobs on different nodes**
- **Platform 3: 2 incorrect Linpack calculations: two-node jobs on different nodes**
- **Platforms 4 & 5: no SDC**

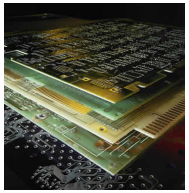
Neutron Beam Testing

- **Background Information:**
  - Cosmic-ray-induced neutrons can cause soft errors that lead to SDC and hard fails
    - Hard fails are important since they affect system reliability
  - LANL is at high altitude where neutrons that can cause SDC and hard fails are more prevalent (~ 6.4x) than at sea level
- **Tested HPC-relevant hardware**
- **Key Research Questions:**
  - Do applications differ in their susceptibility to SDC and hard fails?
    - Understand risks to scientific computation; find efficient test codes
  - How does susceptibility vary among devices of a particular model?
    - We have 10K's of components; need to understand the "susceptibility" distribution
  - How do temperature, voltage and frequency affect the SDC and hard fail rates?
    - Temperature/voltage can be monitored and frequency can be adjusted
    - For future study

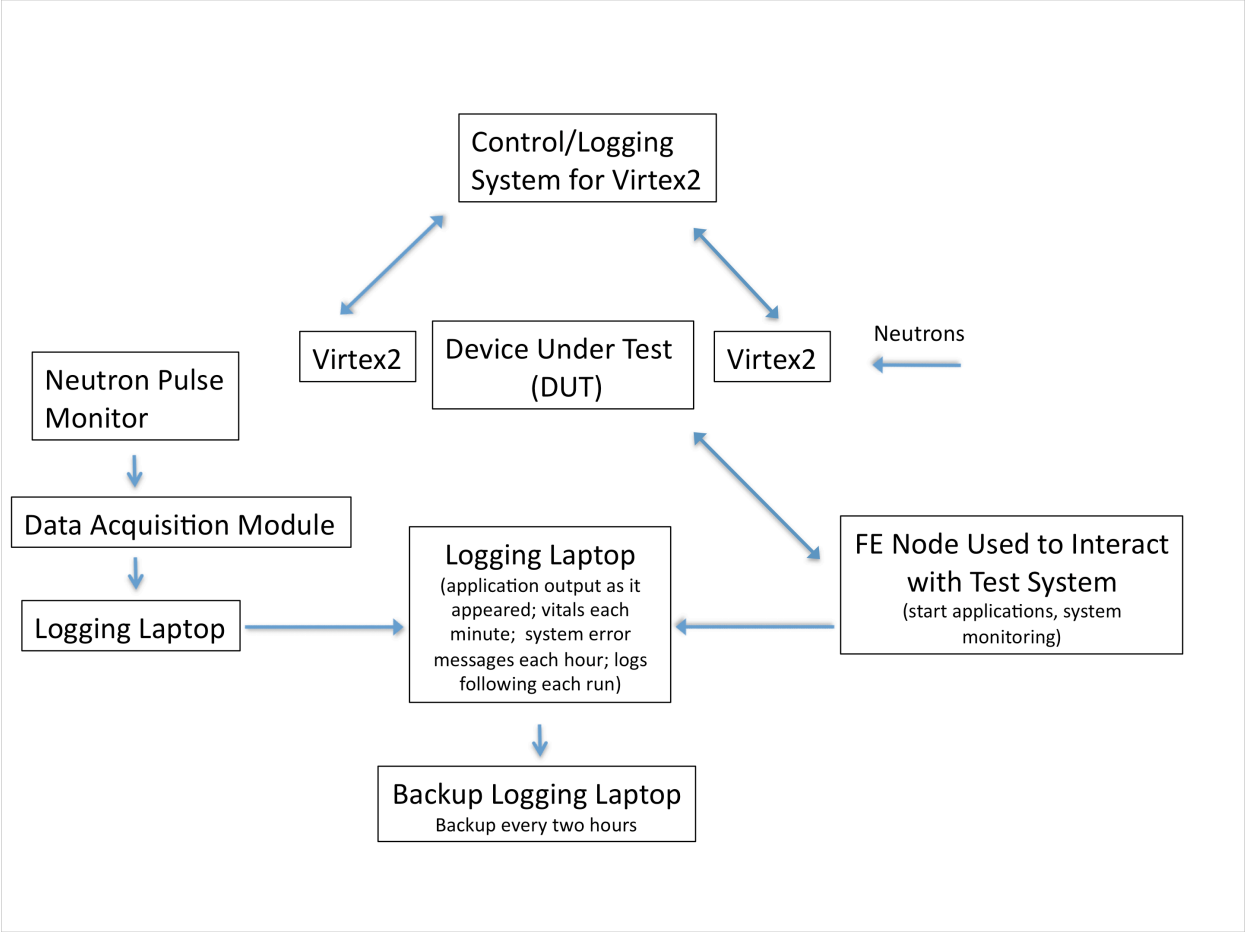
Neutron Beam Testing

- **LANL's LANSCE neutron beam has a spectrum very similar to that at terrestrial and aircraft altitudes**
- **Tested entire nodes rather than hardware on risers**
  - Aim the beam at hardware of interest, but other hardware is irradiated
- **5 applications plus an idle condition**
- **Replicates of hardware**
- **Development of test environment complex**
  - Many details need to be worked out:
    - Electrical requirements, height of table + device, beam aim, ….
  - Scripts for running applications, error logging, heartbeat monitoring, …





Neutron-Beam Test Set-Up

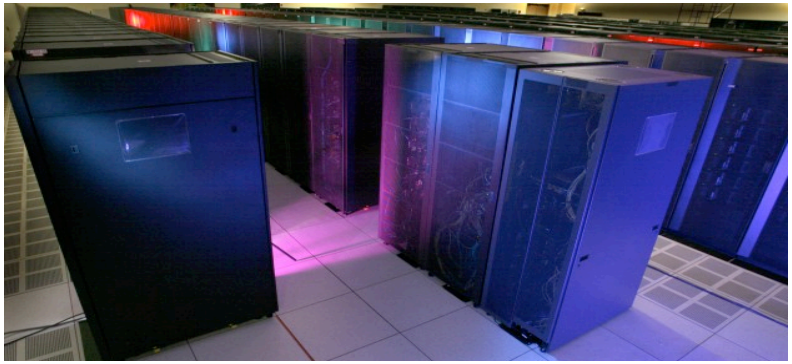


PRELIMINARY Results of Beam Testing

- **Initial modeling completed, but model checking and refinement to be completed**
- **Crashes and SDC observed; failed hardware**
- **No detectable effect of application (doesn't mean there isn't one)**

Possible Future SDC Research

- **E2E testing of HPC resources**
- **Development of test codes**
  - Efficient and effective test codes for different subsystems/components
- **Environmental testing (temperature, particles, …)**
- **Investigation of variability in application and device susceptibility**



Research Challenges I

- **New Technology and Increasing Scale of HPC Platforms**
- **What new reliability concerns emerge?**
  - Scale allows us to detect issues the vendors may not know about
  - How can we quickly identify a small number of susceptible components in a sea of good ones?
- **Can we accurately predict platform reliability *before* procurement/ integration?**
  - Susceptibility distribution for a particular architecture
- **How can we effectively and efficiently test production systems to identify problems *before* they affect users?**
  - Need initial and ongoing E2E testing; balance between testing and production use

Research Challenges II

- **How well can we understand SDC?**
  - What are the causes SDC? How do they change with technology?
  - Can we identify faults that could cause SDC and focus use of fault-mitigation techniques?
  - Can we understand the probability of SDC as a function of environmental/other conditions?
  - How can we learn about SDC rates under nominal conditions?
    - Rare under nominal conditions: need a sound method of extrapolating from extreme conditions to nominal conditions or LOTS of time on test at nominal conditions
    - Beam testing not sufficient
- **Can we predict the SDC rate of a particular application?**
  - Conversion from undetected-error rate
- **How can we ensure that applications keep running to correct answers?**
  - Development of application-specific and generic fault detection and resilience techniques