A Realistic Evaluation of Memory Hardware Errors and Software System Susceptibility

Xin Li[1] Michael Huang[1] Kai Shen[2] Lingkun Chu[3]

[1]Department of Electrical and Computer Engineering
University of Rochester

[2]Department of Computer Science
University of Rochester

[3]Ask.com

2010 USENIX Annual Technical Conference

Memory Hardware Errors: Transient vs Non-transient

- Transient:
  - Completely due to environmental factors
  - Don't cause permanent hardware damage
- Non-transient:
  - Hardware fault plays a role
  - May recur over time

Asymetrical Understanding of Memory Errors

- Transient analysis:
  - Baumann 2004
  - Normand 1996
  - Ziegler et al. 1996
  - O'Gorman et al. 1996
  - Li et al. 2007
- Non-transient error studies:
  - Schroeder et al. 2009
  - Constantinescu 2003
  - No specifics regarding error locations

Importance of Understanding Non-transient Memory Errors

- Non-transient errors
  - Intermittent errors may not be obviously easy to detect
  - System maintenance is not perfect
  - May combine with transient errors to make impact
- The lack of a comprehensive understanding of memory errors
  - High-level studies assume transient errors or resort to synthetic non-transient errors
  - Non-transient errors do happen in practice

A Realistic Evaluation from All Angles

- Collect non-accelerated errors on production computers
  - Detailed per-error address and syndrome
- Simulate how they would manifest with different hardware correction mechanisms
- Observe the end results of software running with these errors

Outline

Outline

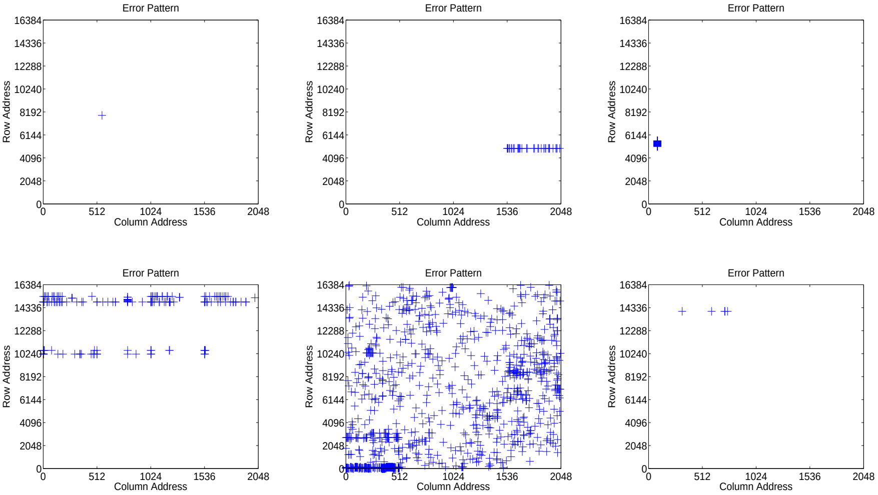
Methodology

- Data primarily from 212 production servers with ECC
  - Monitored for about 9 months
  - Total of 800 GB memory
  - Read error info from ECC registers
  - Enabled hardware scrubbing to help expose errors
- Two other environments are examined
  - 70 PlanetLab geographically distributed testbeds
  - 20 U of Rochester desktops
  - Results reported for transient errors only in *USENIX'07*

Results – Time-line

- 11 machines with errors in the first 2 months
- A new faulty machine after 6 months

Results – Selected Patterns



Results – Patterns

- Summary:
  - 5 cells
  - 3 rows
  - 1 column
  - 1 row-column
  - 2 chip
- Raw data available on our project website
  `http://www.cs.rochester.edu/research/os/memerror`

Outline

Manifestation Overview

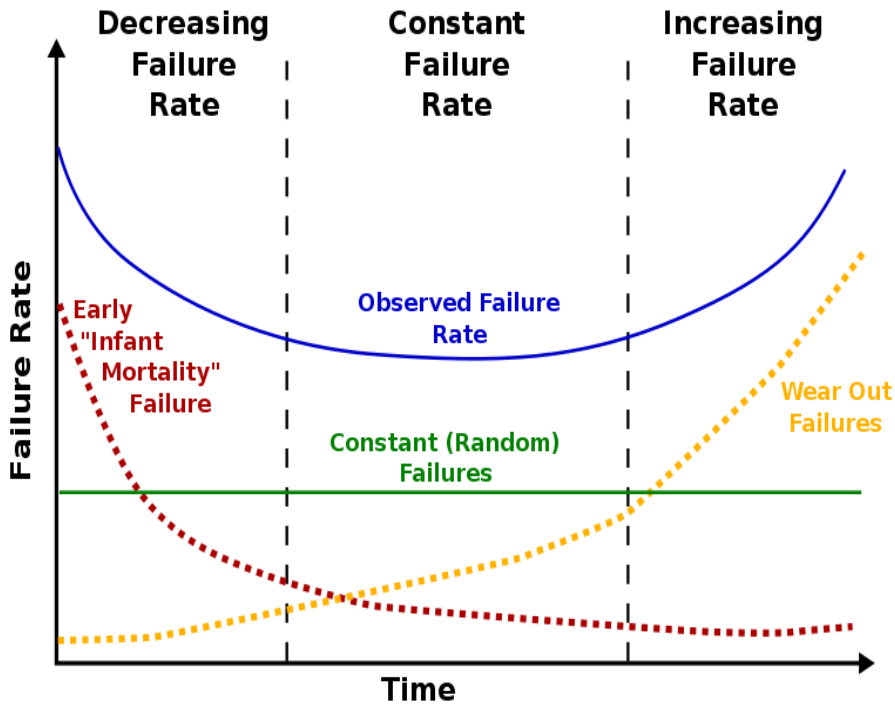
- Countermeasures confine errors inside the memory system
  - ECC correction
  - Preventive maintenance
- Countermeasures at a cost
  - ECC demands extra bits and extra logic
  - Chipkill ECC even requires lock-stepping between channels
- Efficacy is in question

Methodology

- Event-driven Monte Carlo simulation
- Calculate manifestation rates given:
  - Error model (patterns and rates)
  - Countermeasures

Assumptions

- Transient errors
  - Single bit patterns
  - Constant error rates
  - Exponential distribution
- Non-transient errors
  - Patterns based on templates
  - Common belief: bathtub curve
  - Wear-out neglected
  - Weibull distribution (shape parameter < 1)
  - Parameters derived from the raw data

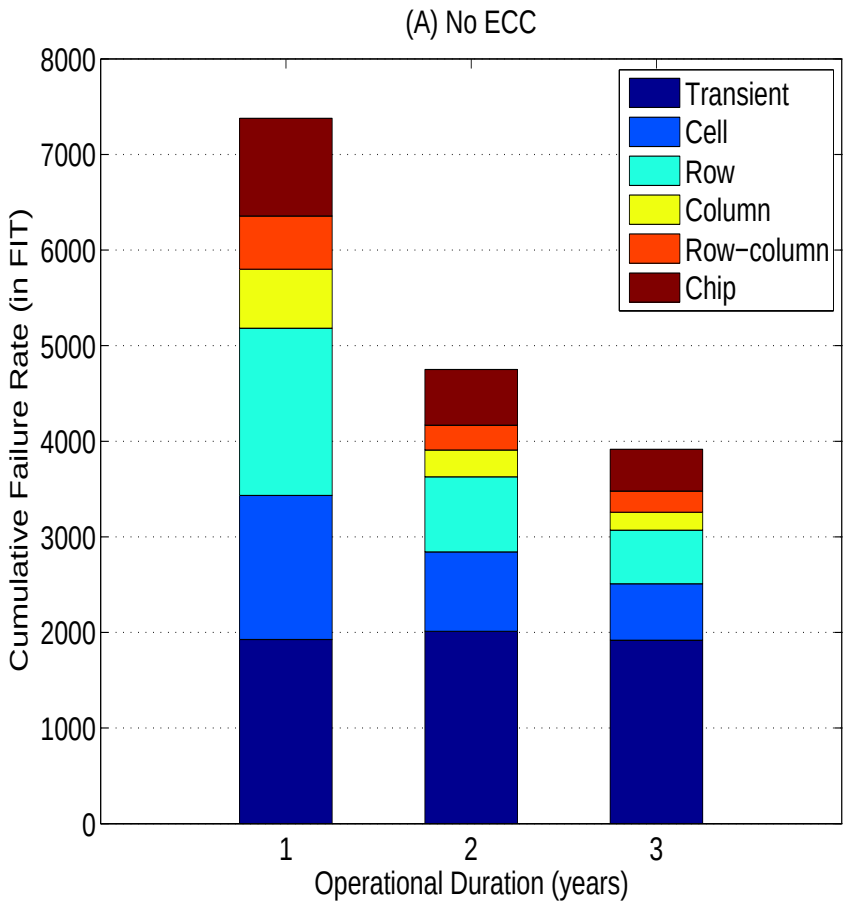


Assumptions Cont'

- ECC
  - SECDED: single bit correction, double bit detection (in a word)
  - Chipkill: correct a whole chip
- Preventive maintenance
  - Not effective in our model
  - Excluded from the results

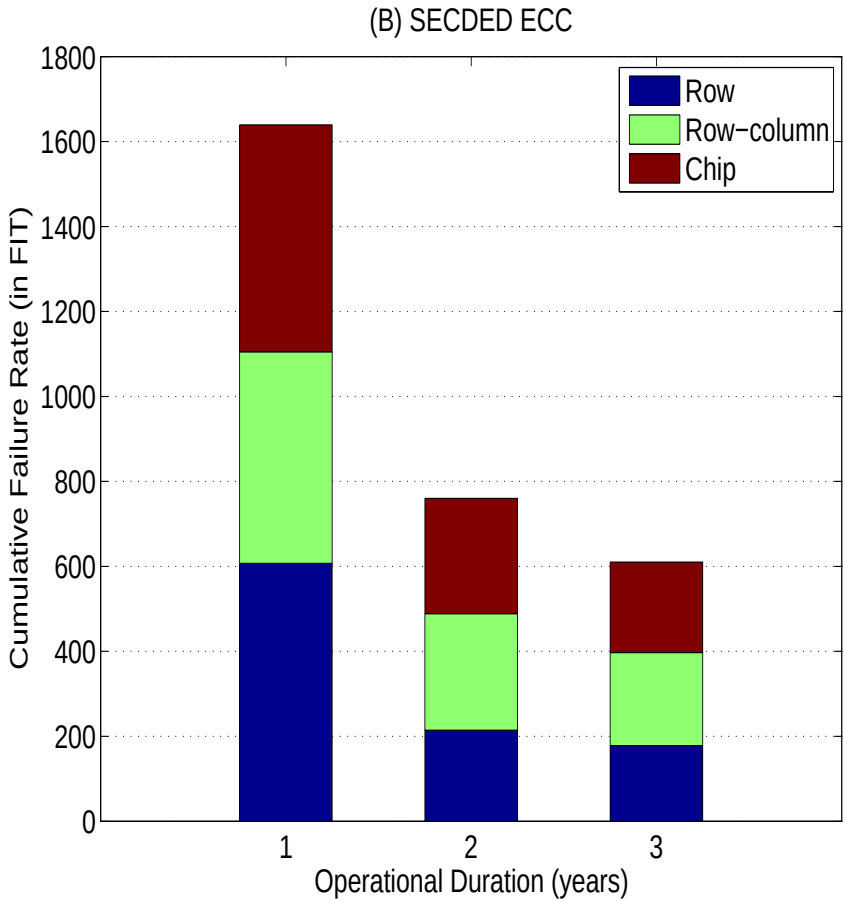
Base Results

- No ECC
  - Transient and non-transient both significant
  - Transient 2000 FIT
  - FIT – Failure In Time (114 FIT – 1000 years MTTF)
  - Non-transient 5000 - 2000 FIT



Base Results (cont')

- SECDED
  - Single-bit errors corrected
  - Eliminated transient / majority of non-transient
- Chipkill
  - No uncorrectable error observed



Bound Estimation and Results

- Estimate rate bounds using statistical methods
- No-ECC and SECDED
  - Non-transient: about 2X difference
- Chipkill
  - Small number of uncorrected errors showing up
  - All caused by transient errors hitting chip error

Outline

Overview

- Software may not be affected by the exposed memory errors
- An investigation of software susceptibility to memory errors
- Root in the realism in the data
- Validate/question conclusions of prior studies

Infrastructure of Injection

- Virtual machine based injection
- Goals
  - Read from faulty locations supplied with erroneous values
  - Write to faulty locations don't overwrite erroneous bits
  - Bookkeeping accesses to faulty locations
- Key challenge: tracking memory accesses

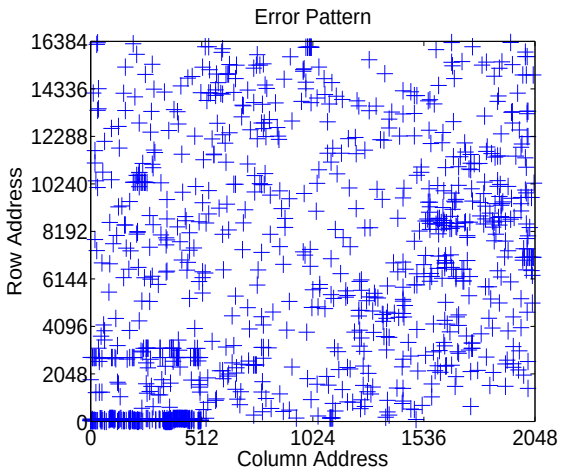
Conventional Tracking Methods

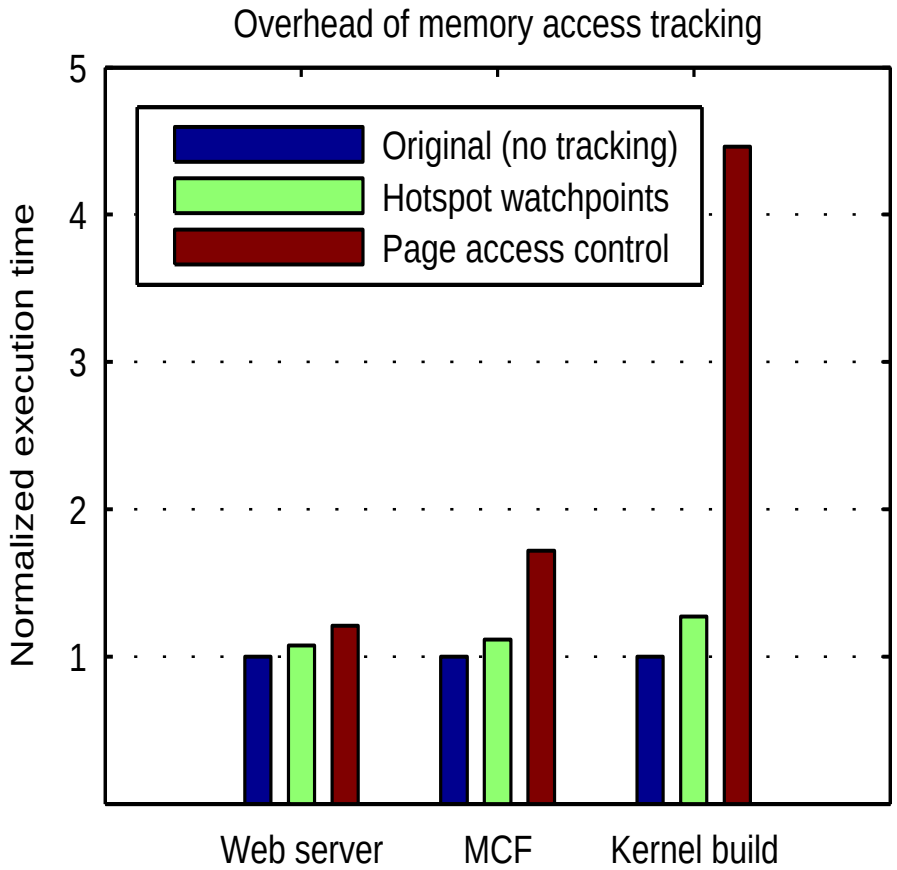
- Hardware watchpoint
- Code instrumentation
- Page access control

Novel Tracking Method

- Observations
  - Error bits spread into different pages
  - Spurious page faults
- Hotspot Watchpoint
  - On access to an error, unprotect the page
  - Set up hardware watchpoint on the error
  - Successive accesses to the error tracked by hardware watchpoints
  - Protect this page again when errors on other pages are accessed

Hotspot Watchpoint Speedup





Evaluation – Non-transient Error Susceptibility

Application	Web server	MCF	Kernel build
No ECC			
M1 (row-col error)	WO	AC	AC
M2 (row error)	OK		
M3 (bit error)	OK		
M4 (chip error)	KC	WO	AC
M5 (row error)	WO	WO	
M6 (row error)	OK		
M7 (bit error)	OK		
M8 (bit error)			
M9 (col error)	WO		
SECDED ECC			
M1 (row-col error)	WO	WO	AC
M5 (row error)	WO	WO	

Table: KC—kernel crash; AC—application crash; WO—wrong output; OK—program runs correctly; blank—not accessed.

Non-transient made transient

Application	Web server	MCF	Kernel build
No ECC			
M1 (row-col error)	WO	AC	OK
M2 (row error)	OK		
M3 (bit error)	OK		
M4 (chip error)	KC	OK	OK
M5 (row error)	WO	OK	
M6 (row error)	OK		
M7 (bit error)	OK		
M8 (bit error)			
M9 (col error)	WO		
SECDED ECC			
M1 (row-col error)	WO	OK	OK
M5 (row error)	WO	OK	

Table: KC—kernel crash; AC—application crash; WO—wrong output; OK—program runs correctly; blank—not accessed.

Additional Discussions

- Miscellaneous validations of prior research in the paper

Contributions

- Memory error data from production systems
  - 212 servers, 800 GB memory, 9 months
  - Detailed information on error addresses and syndromes
  - Substantial non-transient errors (row/column mostly)
- Monte Carlo simulation on error manifestation
  - Simulation on realistic data
  - Significant non-transient errors among manifested
  - Chipkill ECC very effective
- Software susceptibility study
  - A non-transient error injection tool
  - A novel memory tracking approach – Hotspot Watchpoint
  - Software much more susceptible against non-transient
- `http://www.cs.rochester.edu/research/os/memerror`