Terrestrial-Based Radiation Upsets

A Cautionary Tale

Heather Quinn, Paul Graham

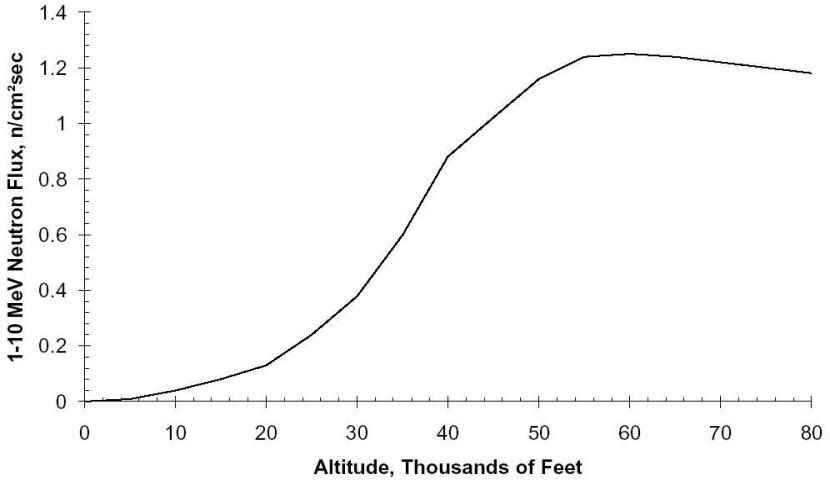
hquinn, grahamp@lanl.gov

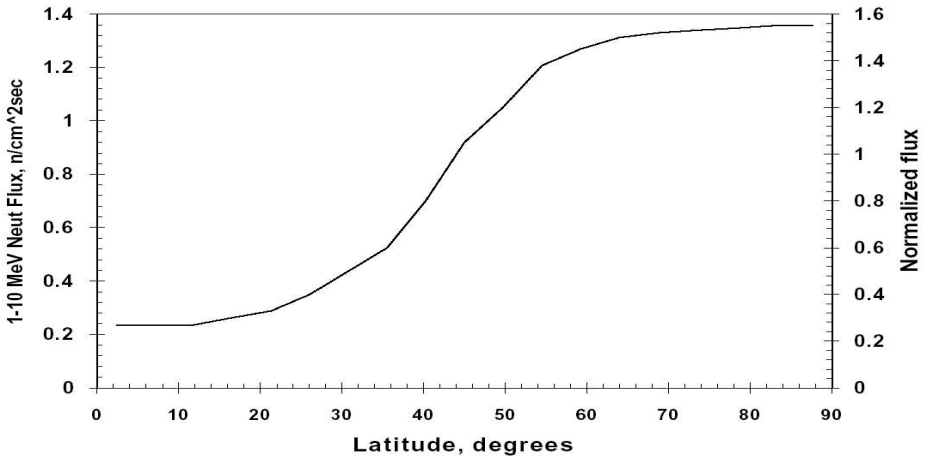
Overview

- Motivation and Background
  - Terrestrial-based Radiation: Coming Soon to a Computer Near You
  - Soft Errors and System Reliability
  - The Vulnerability of FPGA Systems
- Soft Error Rate Estimates
- Low Impact Mitigation Methods
- Conclusions and Future Work

Background: What is Terrestrial-based Radiation?

- Terrestrial-based radiation primarily from neutrons
  - Cause memory upsets
- Flux dependent on longitude, latitude, altitude and geomagnetic rigidity
  - Radiation peaks at high altitudes and near poles
  - Soft errors (SEUs) increase accordingly





Factors in Terrestrial-based Radiation Upsets

- Physics: smaller is not better
  - Smaller transistors are easier to upset (Q_{crit})
  - Denser designs are easier to upset
- System Design: increasing sensitivity (*cross section*)
  - Microscopic: more complex components each generation
  - Macroscopic: larger systems each generation
- System Location: peak neutron radiation levels
  - Multiprocessor and multi-FPGA systems for airborne applications are under research

Soft Errors and System Reliability

- Soft errors are often undetected, unmitigated
- For large-scale, reliable systems unmitigated soft errors are disastrous:
  - Sun Microsystems received bad press for soft error failures in their high end servers
  - System X architect joked they “felt like [they] had not only built the world's third fastest supercomputer, but also one of the world's best cosmic ray detectors.”
  - Q Cluster at LANL experiences 26.1 CPU failures a week due to soft errors
- We are interested in highly available, highly reliable reconfigurable supercomputers with thousands of FPGAs and microprocessors
  - Large cross-sections

Soft Errors in FPGA Systems

- FPGA systems are not exempt from soft errors:
  - The entire system (FPGAs, microprocessors, memory) is sensitive
  - Memory upsets are the root problem
- Memory upsets in FPGAs cause:
  - Changes in intermediate processing values
  - Changes in the state
  - Changes in the configuration

As FPGA systems increase in complexity, soft error rate also increases

Mitigating Soft Errors: Expensive

- Not mitigating soft errors expensive for large-scale, reliable systems
  - Silent data corruption
  - Unreproducible system crashes
- Shielding for neutron radiation is nearly impossible
  - Underground bunkers covered in meters of rock, dirt and water
- Mitigating soft errors through traditional methods is expensive
  - Area, power, speed

Low impact mitigation methods are needed

Analysis Setup

Use vendor test data to estimate soft error rates for untested devices, locations and system size

- Scale reference systems to estimate SER for locations
  - San Jose, Albuquerque, Cheyenne, Los Alamos, Leadville, Mauna Kea, White Mountain
- Scale reference systems to estimate SER for other systems
  - More/larger FPGAs, more memory, more microprocessors

Scaling Soft Error Rates

- SER:

$$SER = flux * \sigma_{dev}$$

- Estimates scaled from reference systems
  - Scaling for locations

$$SER_{loc2} = \left(\frac{flux_{loc2}}{flux_{loc1}}\right) * SER_{loc1}$$

  - Scaling for system size

$$SER_{sys2} = \left(\frac{\sigma_{dev2}}{\sigma_{dev1}}\right) * SER_{sys1}$$

  - Scaling for both

$$SER_2 = \left(\frac{flux_{loc2}}{flux_{loc1}}\right) * \left(\frac{\sigma_{dev2}}{\sigma_{dev1}}\right) * SER_1$$

Derating/Uprating Estimates

- Estimates are worst case scenario, order of magnitude
- Derating Factors:
  - System Utilization: 5-20% of the entire system
  - “Rosetta Factor”: Accelerator test results are about 1.5 times higher than atmospheric test results
- Uprating Factors:
  - Transistor Size: under research

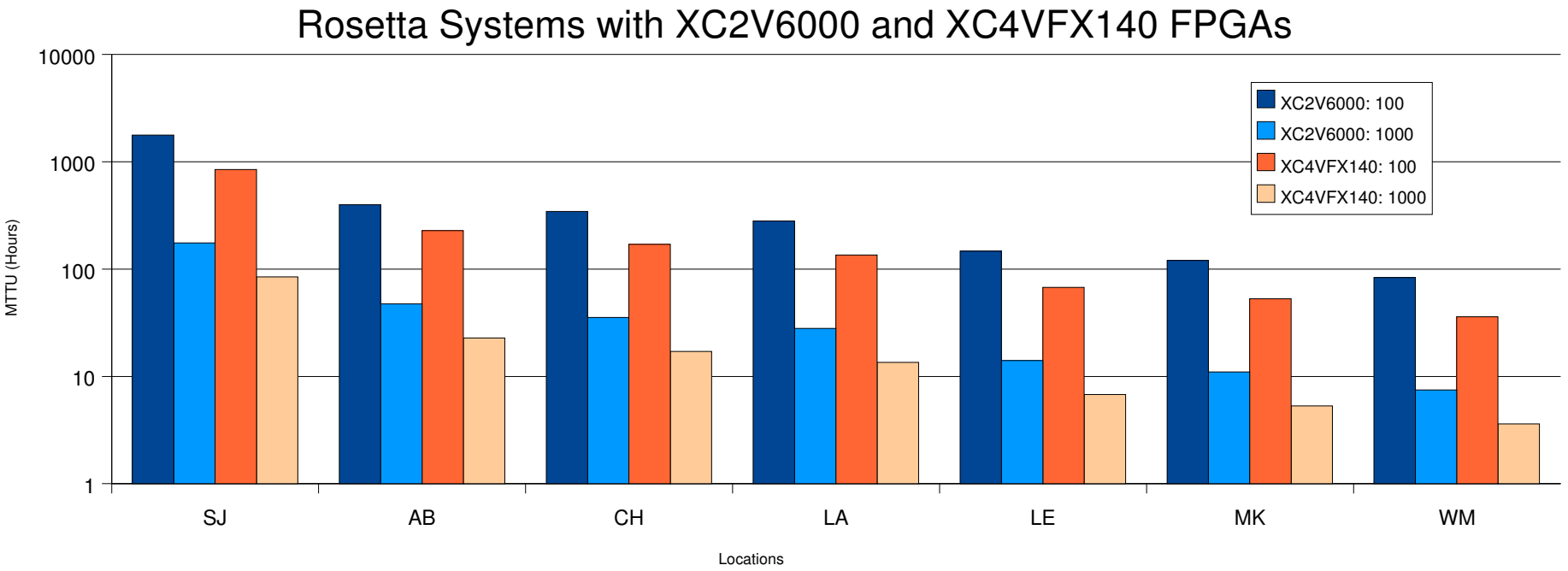
General Trends in Soft Error Rates

- Increasing system size or flux unavoidably increases SER
  - We are not here to beat up on vendors
- With current trends in system design, soft errors will become more prevalent:
  - Microscopic: design components to be less likely to upset
  - Macroscopic: design large systems to be error resistant
- Research and development now while the problem is still manageable
  - Determine the scope
  - Find low impact mitigation methods
  - Change our system design methods

FPGA Estimates

- Estimates were determined from three tests
  - Xilinx Rosetta Test: Atmospheric and accelerator testing of a 100 device XC2V6000 system
  - iRoC/Actel Test: Accelerator testing of Actel, Altera, and Xilinx devices
  - Altera Test: Testing of EP1C6, EP1C20, EP1S25, and EP1S80 devices
- Rosetta test results were used as the reference system
  - Increased flux and system size to determine the change in MTTU
  - Correlated results to the other two tests for accuracy

Xilinx Results

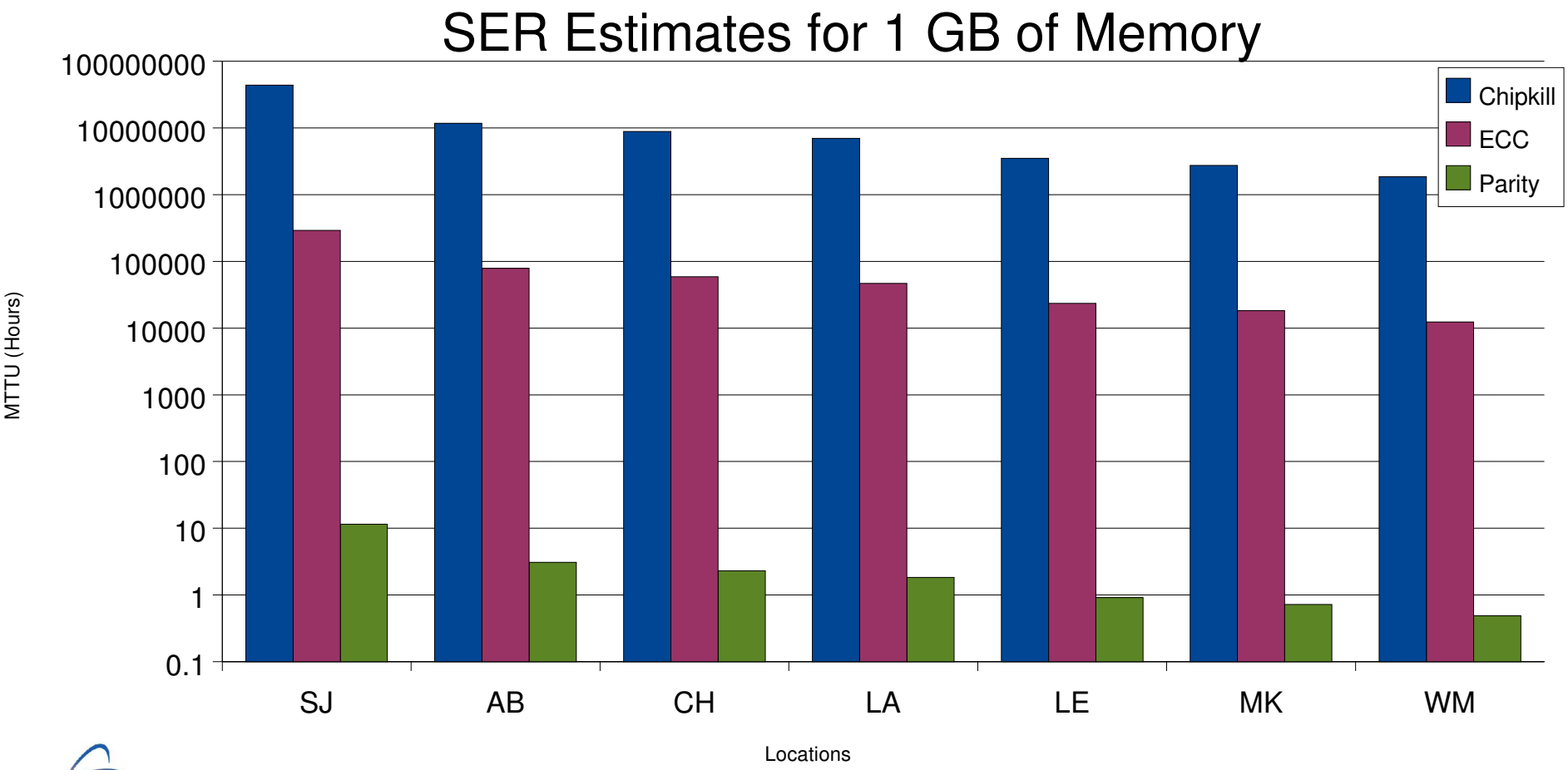


Multi-FPGA systems and high altitude systems need to mitigate soft errors

Memory Estimates

- Estimates were determined from two tests
  - IBM Test: Monte carlo modeling
  - iRoC Test: Atmospheric testing in airplanes
- IBM assumes that 4% of all upsets in DRAM are multi-bit upsets
  - Density, geometry, transistor size important factors
  - Multi-bit upsets break ECC protection

IBM Results


SER Estimates for 1 GB of Memory
100000000
10000000
1000000
100000
10000
1000
100
10
1
0.1
MTTU (Hours)
Chipkill
ECC
Parity
SJ
AB
CH
LA
LE
MK
WM
Locations

Memory Results

- Soft errors in the memory subsystem comparable to FPGAs
- Derating for memory subsystem utilization important
  - How much memory is used
  - How much memory is read

Even with derating, ECC or Chipkill protection is suggested

Microprocessor Estimates

- Much research has been done on preventing soft errors
  - Actual values unreported
- Caches and register files are sensitive to soft errors
  - Upsets to the L2 cache most common
  - Upsets to the L1 cache or register files rare but bad

Microprocessors

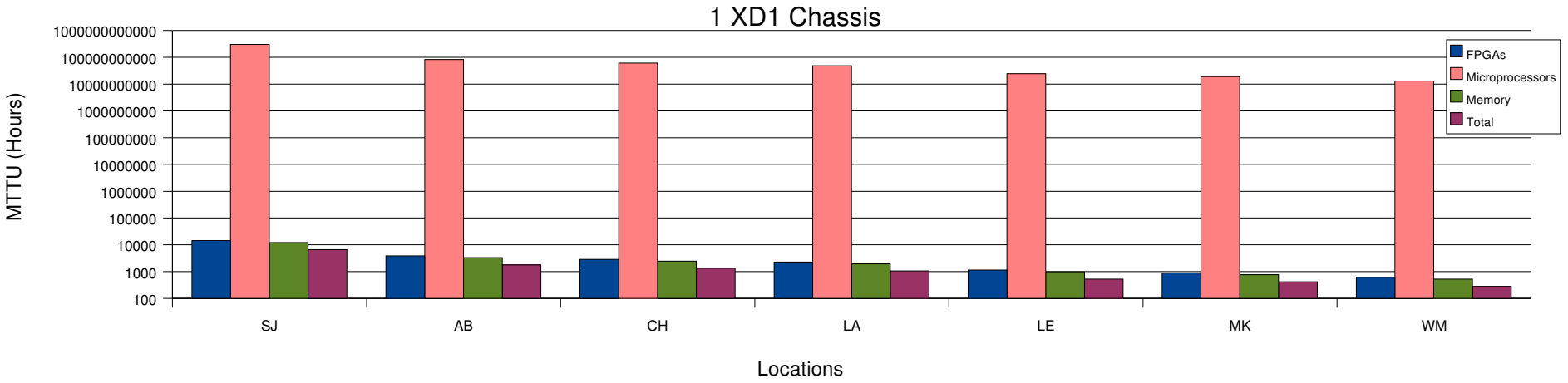
- Most current server-grade microprocessors have ECC protected L2 caches with cache scrubbing, and parity protected L1 cache
- Most older and many current commodity-grade microprocessors have unprotected L1 and L2 caches
  - Used in many computing clusters ($$$)
  - Used in large quantities, soft errors become apparent

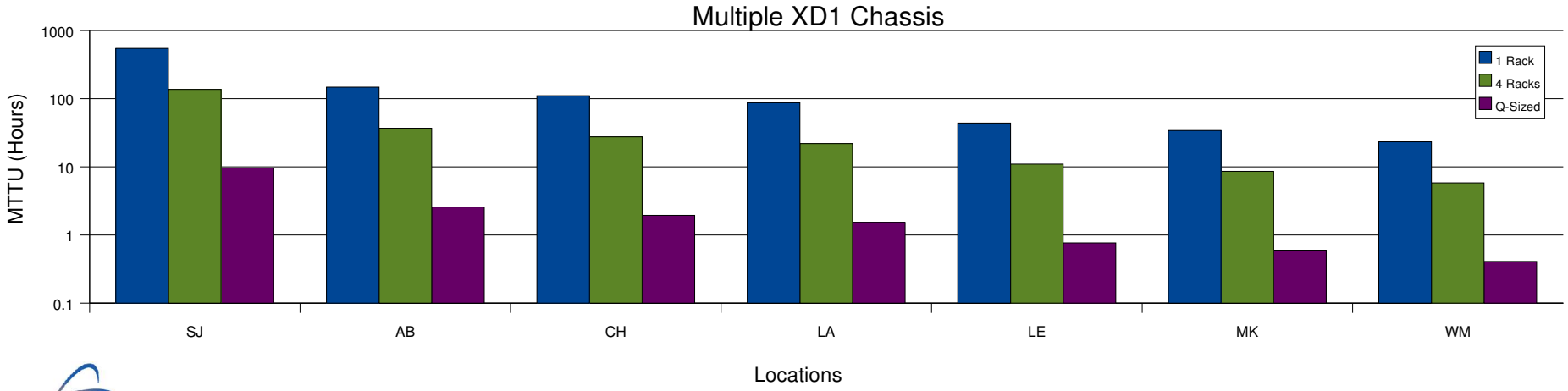
The cost of ad hoc soft error mitigation and crashes is more expensive than server-grade microprocessors

Bringing It All Together: The Cray XD1

- A reconfigurable supercomputer
  - 26 Xilinx FPGAs
  - 12 Opteron microprocessors
  - 24 GB ECC-Protected RAM
- Assume Opterons correct all single bit errors but fail on multi-bit errors

Estimated Soft Error Results



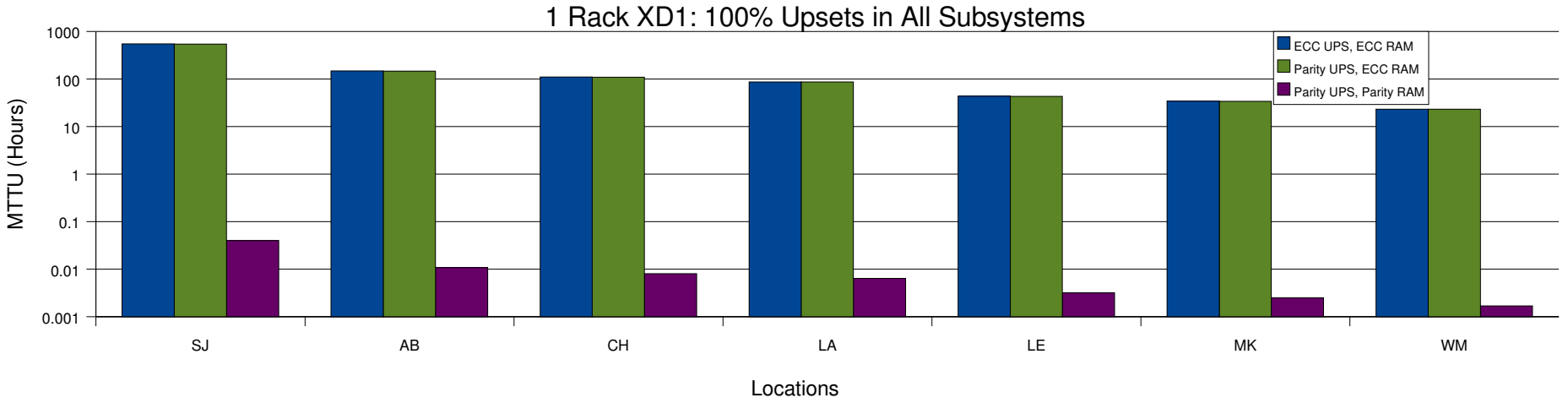


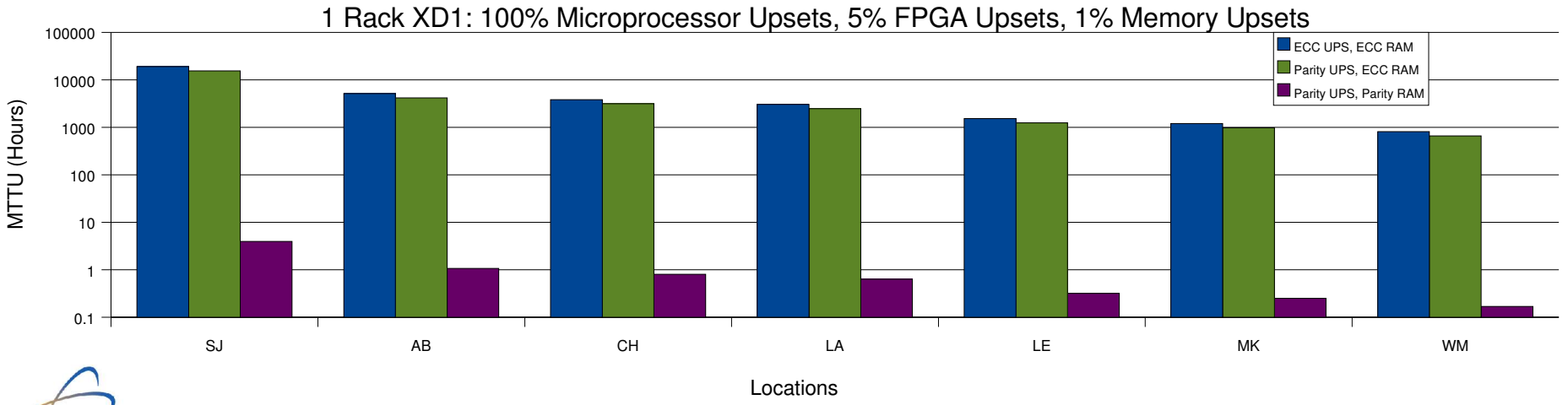
Los Alamos NATIONAL LABORATORY EST. 1943

The World's Greatest Science Protecting America

NNSA

ECC-Protection vs. Parity in Large Systems





Low Impact Mitigation Methods

- Research *now* before soft error problem worsens
- Need methods that:
  - Balance power, speed, and area for reliability
  - Can be tuned to soft error rate
  - Are easy for designers to implement

Mitigating FPGA Soft Errors

- Use CLBs or embedded microprocessors to mitigate soft errors
  - Xilinx SEU Controller for Virtex II Pro: ICAP interface and Power PC 405 core to scan readback for errors
  - Partial Triple Modular Redundancy: TMR critical gates
- Partial configuration methods that rely on readback
  - Single Frame Correction: CRC frame check to detect errors in the readback data
  - Processor-based Detection: host processor detects errors in readback while FPGA processes
  - Scrubbing with CRC: preventively reconfigure device

Mitigation at the Microprocessor Level

- Software must be aware that individual nodes can fail
  - Nodes failure causes only interruption to node computation
  - Software senses when a node has failed
    - Reschedule the computation that the node was processing
    - Reboot the node
    - Reschedule all computations for the node while rebooting
- Software detection of faulty computation
  - Detection methods: checkpointing, information theory, machine learning
  - Software manages reprocessing of faulty computation:
    - Reseting node that the computation originated from
    - Recalculate computation

Conclusions

- With the current trends in technology and system design soft errors will become more noticeable in the next decade
- Recommendations for reconfigurable supercomputers
  - Low impact mitigation methods for FPGAs
  - ECC or Chipkill protected RAM
  - Server-grade microprocessors with protected caches and cache scrubbing
  - Nodes fail independently
  - Software handles node failures

Terrestrial-Based Radiation Upsets

A Cautionary Tale

Heather Quinn, Paul Graham

hquinn, grahamp@lanl.gov