

# High Performance Computing Resilience Testbed

Héctor J. Machín  
University of Turabo

Research Alliance in Science and Math

Computer Science and Mathematics Division, Oak Ridge National Laboratory

Mentors: Stephen L. Scott, Thomas Naughton, Geoffroy Vallée, Anand Tikotekar

## Motivation and Research Goal

- **Probability of failures in high-performance computing (HPC) systems**
    - Systems increase in size
    - Complex scientific computing applications
  - **Software and hardware failures**
    - Increases application interruption
    - Decreases productive use of resources
    - Wastes allocation hours (costs)
  - **Checkpoints: fault tolerance in HPC systems**
    - Application periodically stops useful work
    - Writes checkpoints to disk
  - **Transition to petascale computing †**
    - Multicore size doubling each 30 months
    - System utilization drops to zero by 2013
    - 100% time only writing checkpoints
  - **Develop a resilience testbed**
    - Fault injection techniques
    - Data gathering/monitoring
    - Analyzers for failure evaluation
- † B. Schroeder, G. Gibson – “Understanding Failures in Petascale Computers”

## Resources Used

### Ganglia

- Scalable distributed monitoring system
  - gmond – metric collecting daemon
  - gmetad – storage daemon, round robin databases (RRD)
  - gmetric – tool to inject custom metrics

### Cbench

- Perl-based framework for HPC systems
- Building, running and analyzing outputs of open source codes

### Ovis 1.1

- Real-time monitoring of clusters
  - RRD readers – read the RRD databases
  - RRD starter – start instances of RRD readers for all nodes
  - RRD killer – kills all RRD readers

### HPCC Parallel Benchmarks

- Test suite that examine the performance of HPC architectures
- Kernels with memory access patterns

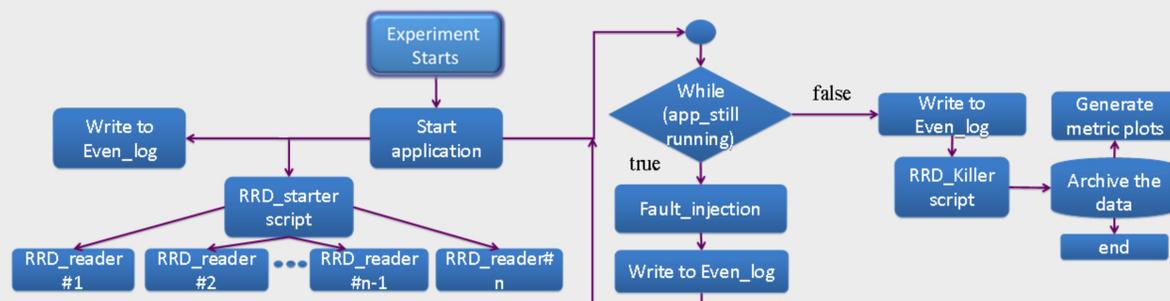


Fig 1. Research Methodology

System Name	# of Nodes	Procs Per Node	CPU Type	CPU Speed	Memory per Node
XTORC	64	1	Pentium IV	1695.7MHz	768MB

Table 1. System Specifications



Fig 2. XTORC Cluster System

## Results and Analysis

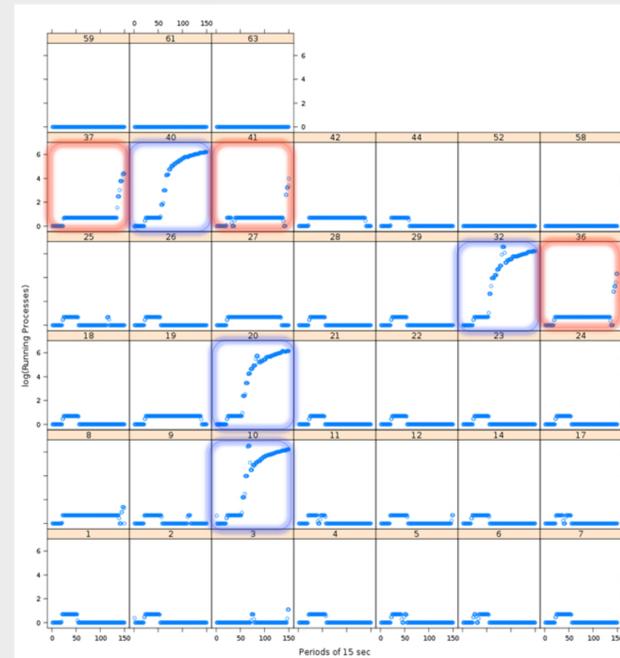


Fig 3. Running processes vs. periods 15 sec.

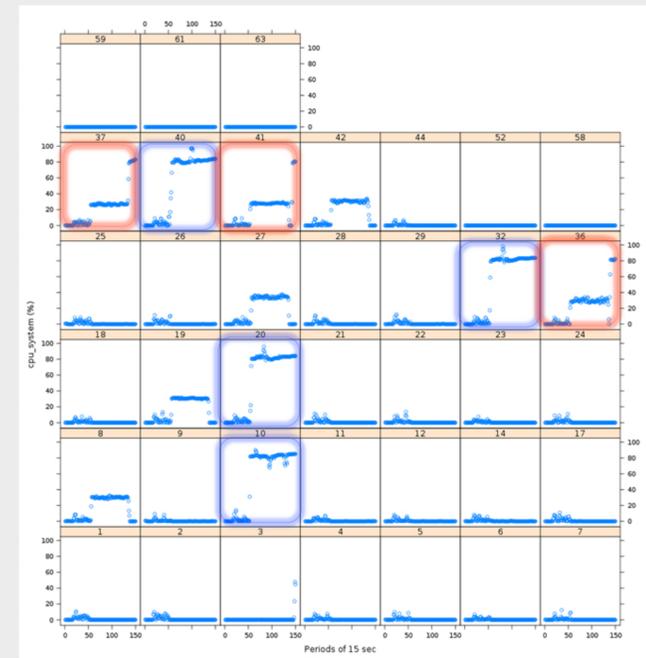


Fig 4. CPU system percent vs. periods 15 sec.

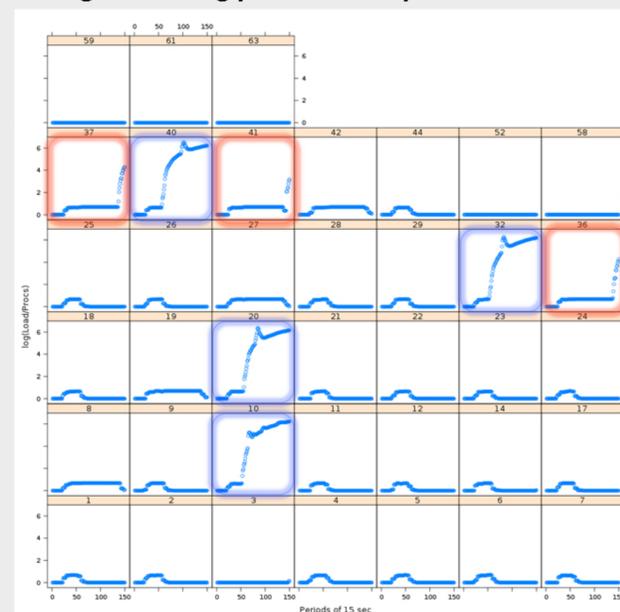


Fig 5. Load/processor vs. periods of 15 sec.

## Summary

- Emphasis on educational process
- Automation processes
  - Fault injection
  - Data gathering/monitoring
  - Data and event archival
  - Failure visualization tools

## Future Work

- Explore more fault generation techniques
- Explore the integrity of the faults injected
- Extended research on more complex HPC systems