Understanding Fault Scenarios and Impacts through Fault Injection Experiments in Cielo

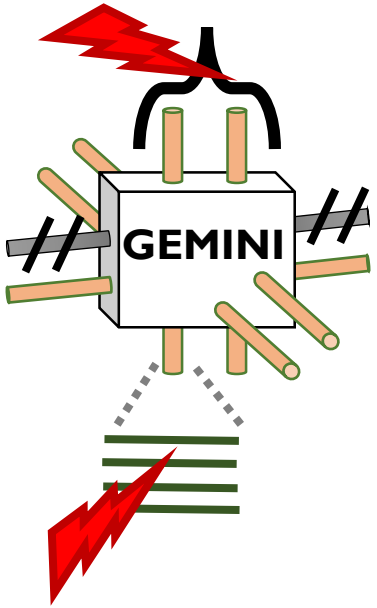
Valerio Formicola, **Saurabh Jha***, Daniel Chen, Fei Deng,
Amanda Bonnie, Mike Mason, Jim Brandt, Ann Gentile,
Larry Kaplan, Jason Repik, Jeremy Enos, Mike Showerman,
Annette Greiner, Zbigniew Kalbarczyk, Ravishankar K. Iyer, and Bill Kramer

***Contact: sjha8@Illinois.edu**

Presented on – May 11, 2017

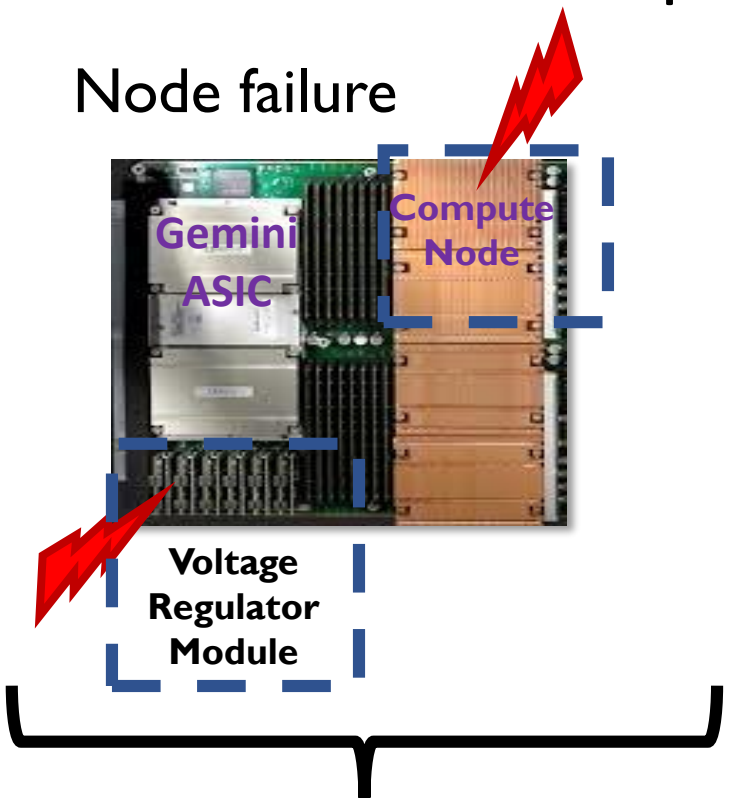
Failures in HPC Compute System

Connection failure

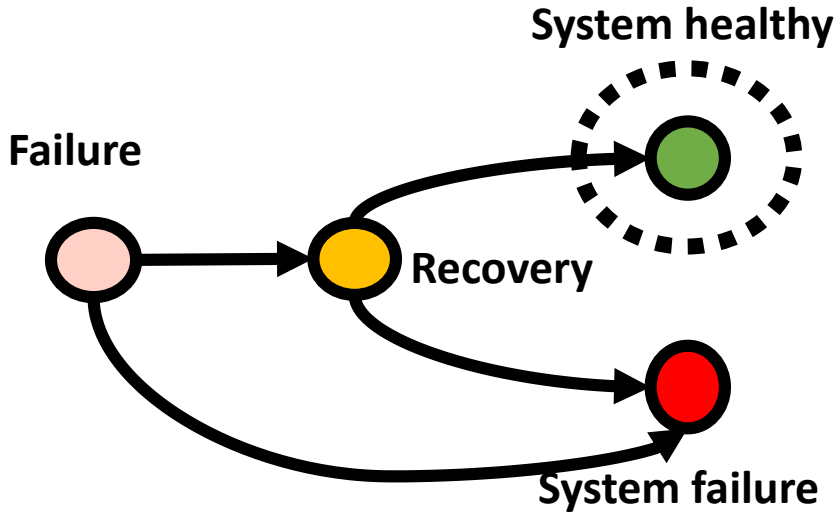


Link failure

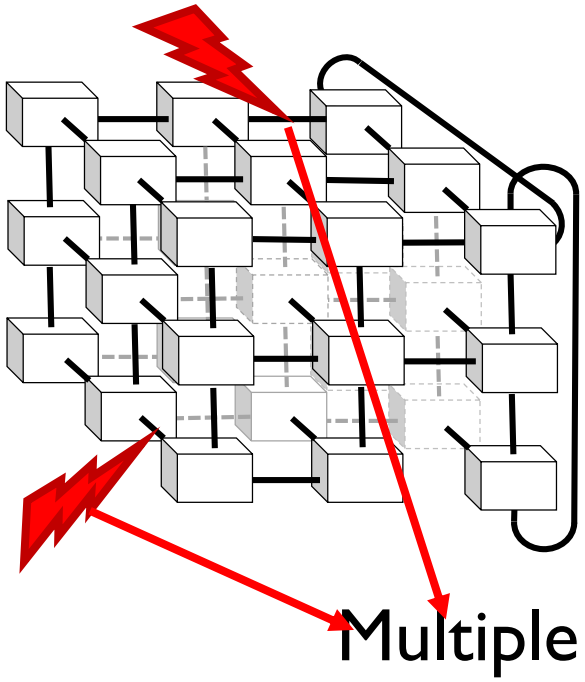
Node failure



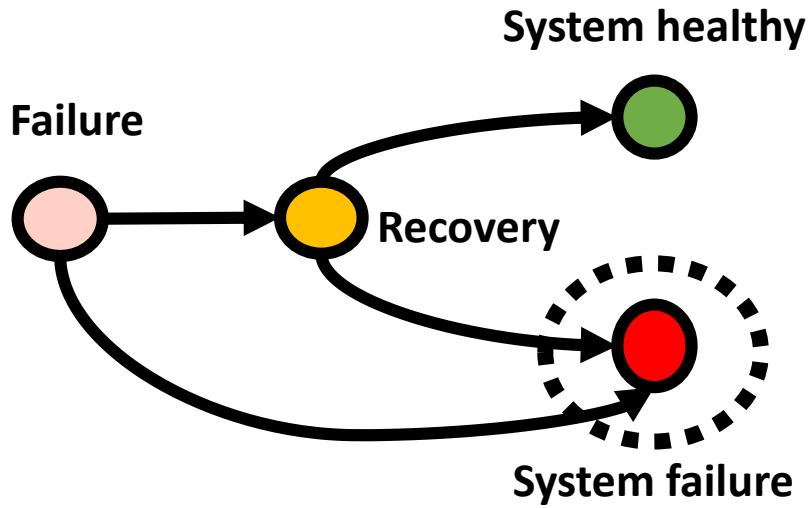
Compute blade failure



Failures in HPC Compute System



Multiple failures*



- Increasing rate of component failures may lead to complex failure scenarios
- Testing system @scale is hard and time consuming
- Challenges @scale
  - Distributed system challenge - Failure detection, detection latency, fault propagation
  - Design Errors – heisenbugs, design robustness

* Multiple failures do not necessarily lead to system failure

Failures in HPC Compute System

- Computing facilities and vendors need to be aware of complex failure scenarios
- Instrumentation and analyses methods that provide early indications of problems may help mitigate the effects of failures

- Decrease in MTBF in future systems will lead to many more complex failure scenarios
- Testing system @scale is hard and too expensive
- Challenges @scale
  - Distributed system challenge - Failure detection, detection latency, propagation delay
  - Software engineering issues are encountered only @scale
    Code bugs, design robustness etc.

HPCArrow: Fault Injector for HPC Interconnection Networks

- Create fault models and failure scenarios to be recreated from field-failure data
  - Provide controlled environment to inject faults
  - Provide ability to conduct experiments in a repeatable way, e.g. malleable scheduling
- Understand fault propagation and recovery mechanisms
  - Fault to failure path models are rarely complete
  - Recovery mechanisms further obscure failure paths
- Develop methods/tools that provide early indication of critical failures, with impact on application success and system continuity
- Assist development of future acceptance test for HPC systems based on system ability to tolerate faults

Contributions

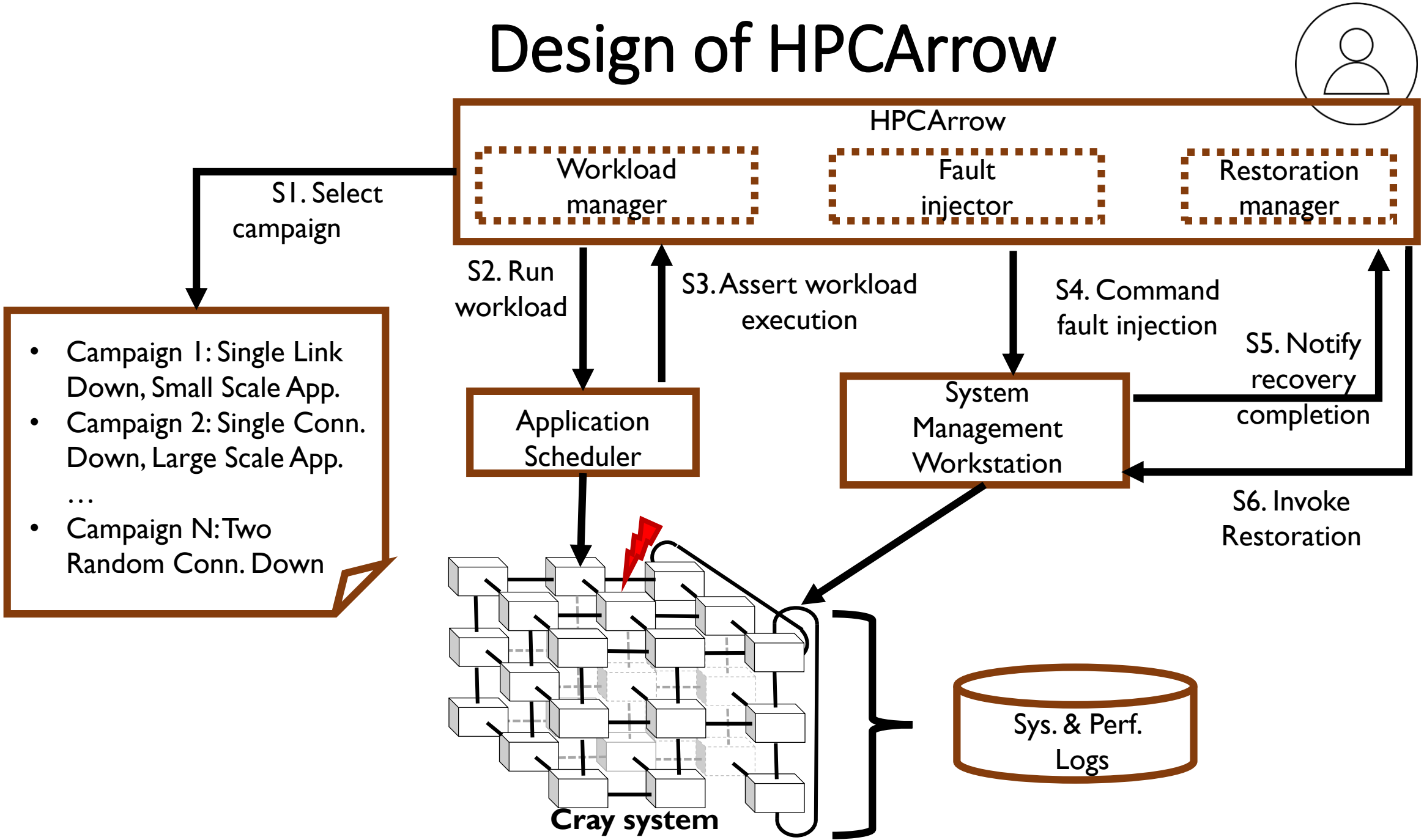
- HPCArrow : A tool for injecting faults into HPC interconnection networks

- Cielo Supercomputer at LANL was used to show the value of HPCArrow @scale
  - Executed 18 fault injection experiments, which led to failures of 54 links, 2 nodes, and 4 blades
  - Characterized the impact of network-related faults on application and system at a granular level

- Identification of critical errors and conditions
  - Detect deadlock and no application progress
  - Characterized network-related critical errors

- Recommendation for notification and instrumentation at application and system levels
  - Feedback to apps about recovery and critical error conditions
  - Opportunity for checkpointing and/or application-specific fault tolerance

Design of HPCArrow

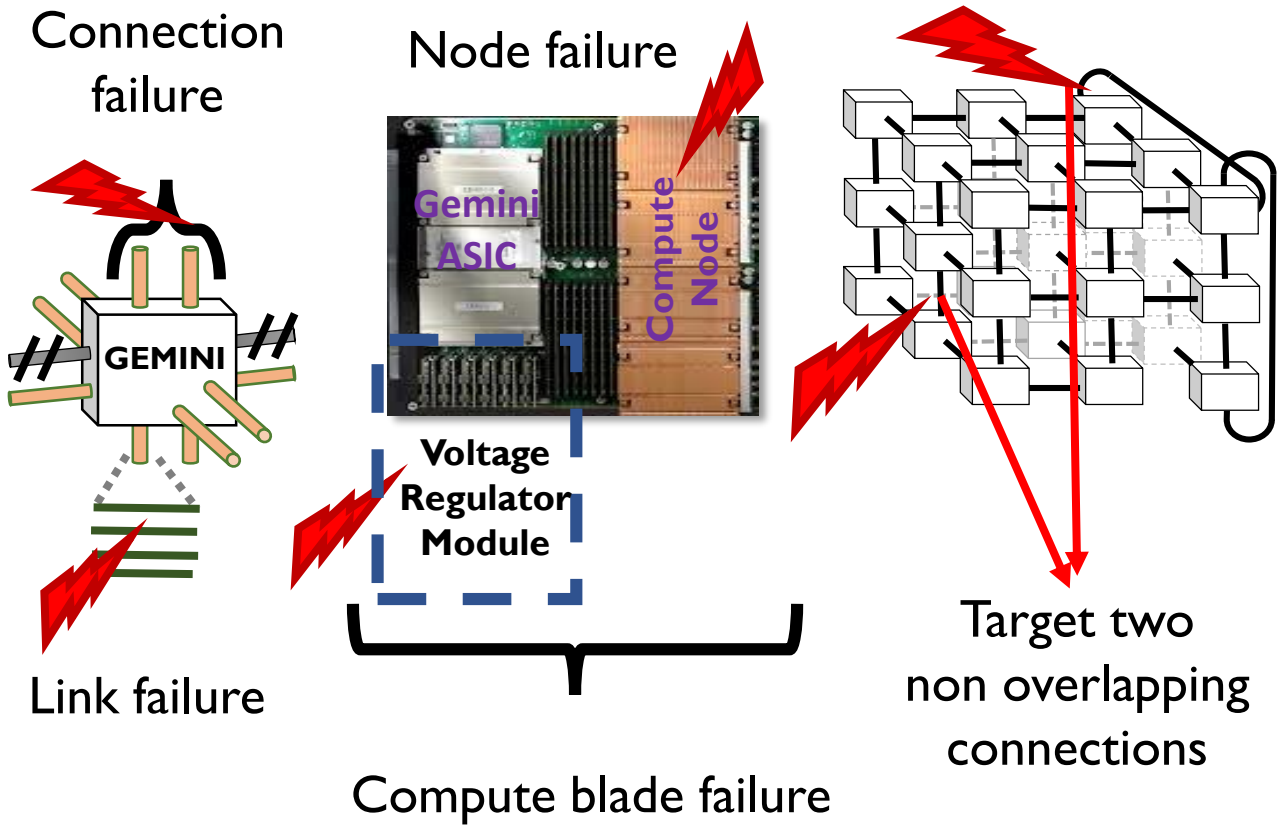


Workload Generation

- Application and scale parameters are specified in the campaign file
- Application scales
  - Nano: occupies < 6.25% of system nodes
  - Small : occupies >= 6.25% nodes and < 12.5 system nodes
  - Medium: occupies >= 12.5% and < 25% system nodes
  - High: occupies >= 25% and < 50% of system nodes
  - Large: occupies >= 50% nodes
- Target Application: Intel MPI Benchmarks
  - Measures point-to-point and global communication operations for a range of message sizes
  - Not a typical HPC representative app but ensures network traffic during fault injection
  - Currently experimenting with Enzo

Failure Scenarios & Models

Failure Scenarios	Target	Method
Link failure	One link	Status flag
Connection failure	All links from one router to another	Status flags
Node failure	One node	Power off
Blade failure	One blade (2 ASICs, 4 nodes)	Voltage Fault
Non overlapping connection failure	All links in two connections with different x,y,z dimension	Status flags



Data Collection and Analysis

Data sources

Aggregated system logs (incl. hardware error logs)

LDMS performance counters

Job commands and output logs

Administrator console screen

Info on the fault Injection experiments: timestamps, targets, workload, errors on the admin screen

Subset data for each experiment

Extracted logs

LogDiver

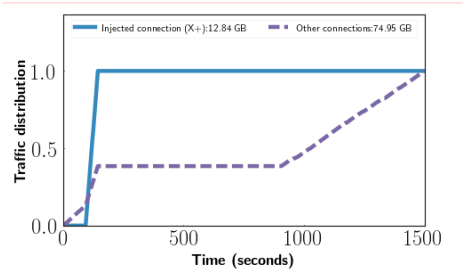
Filtering and reconstruction of network recovery

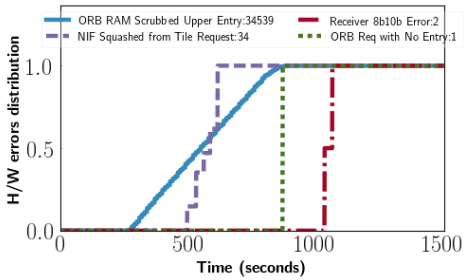
Filter network performance counters on target nodes

Unified Output

- Error counters
- Recovery completion time
- Recovery phases
- Recovery retries, failures and successes
- Application errors (e.g., MPICH2 errors)
- Cumulative Distribution Function of traffic and error logs

plots





Target System

- Target System
  - Cielo, a petaflop Cray XE system at the Advanced Computing at Extreme Scale (ACES) system
  - 8944 compute nodes, 16x12x24 3D torus topology

Experiment Summary

- 18 campaigns launched, faults injected on
  - 54 links
  - 4 blades
  - 2 nodes
- 1.3 GB of hardware error, nlrd logs
  - 461 LogDiver regex patterns found
  - 71 hardware error types found
- ~1 Terabytes of performance logs

Results

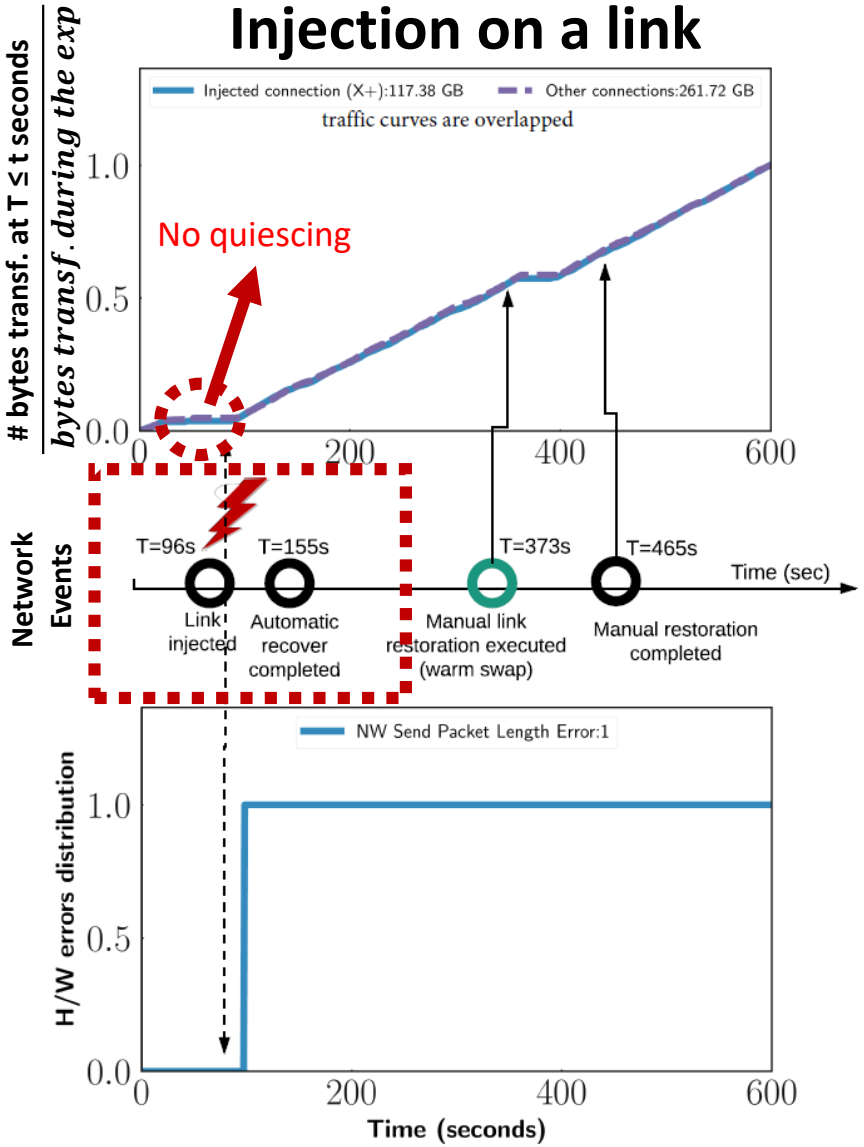
Anomalous Hardware Errors

- Errors were categorized anomalous based on
  - Duration
  - Frequency

Hardware Error Type	Cause/Effects
ORB RAM Scrubbed	Request times out and ORB entry is freed
ORB Request With No Entry	Response packet comes into the receiver response FIFO buffer that does not correspond to a full request entry. App/gnilnd terminates
Receiver 8b10b error	Coding error on link. Results in packet loss
LB Lack of forward progress	All requests destined for NIC will be discarded

Quiescing vs Non-Quiescing Case

Injection on a link

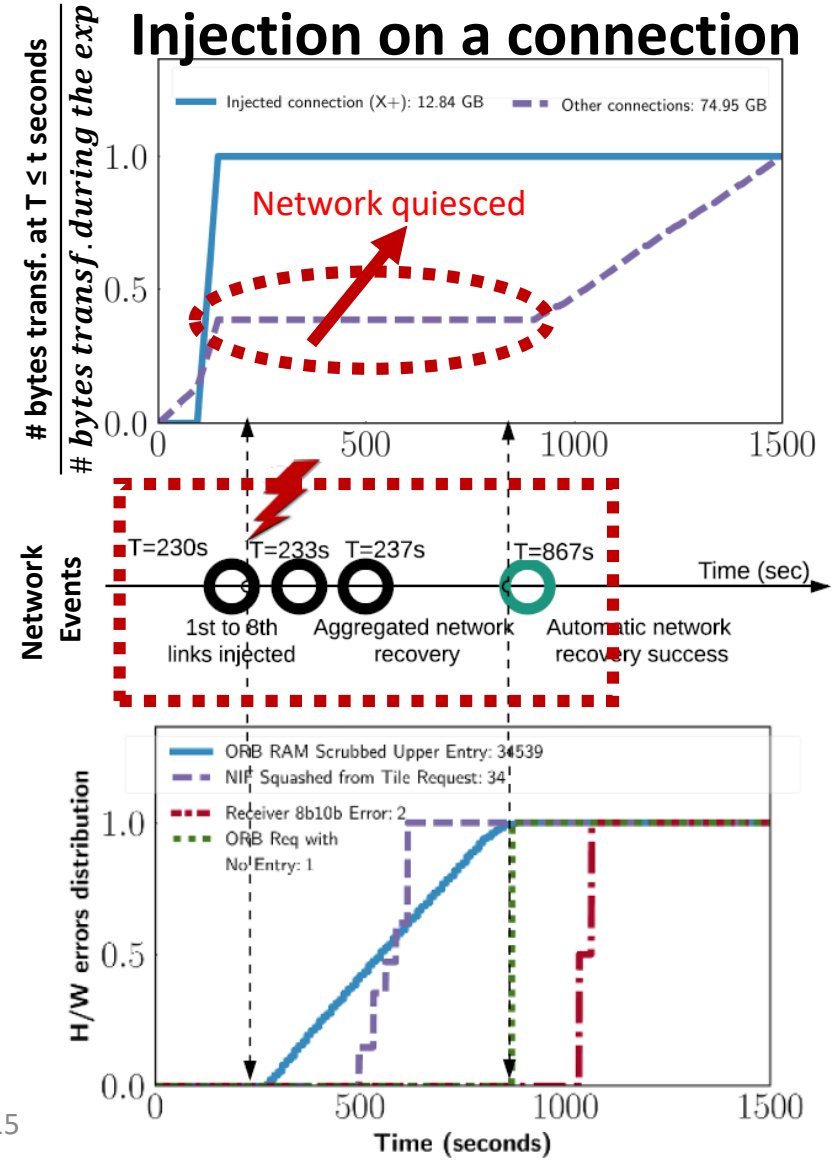


Single link failure is benign and recovery is fast compared to multiple link failure (in conn.) which can take several minutes

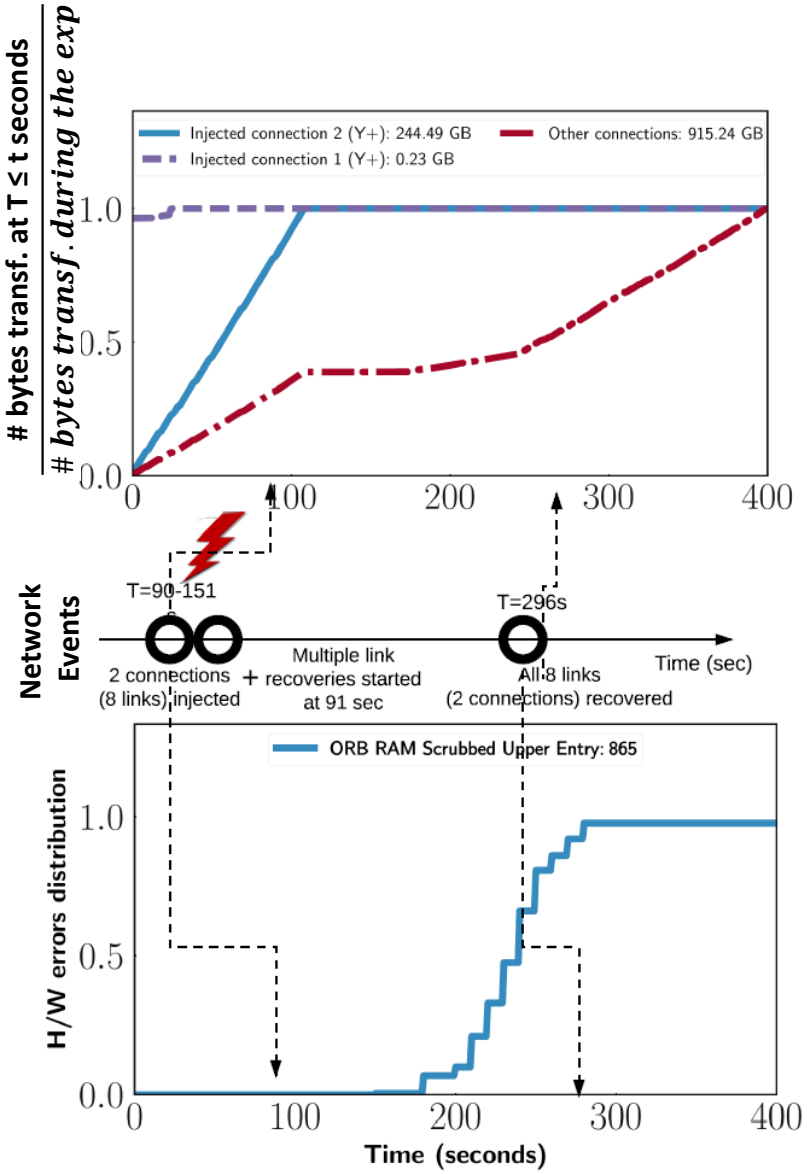
Traffic is paused for 10 minutes

Propagation of errors may lead to critical network errors impacting app

Injection on a connection

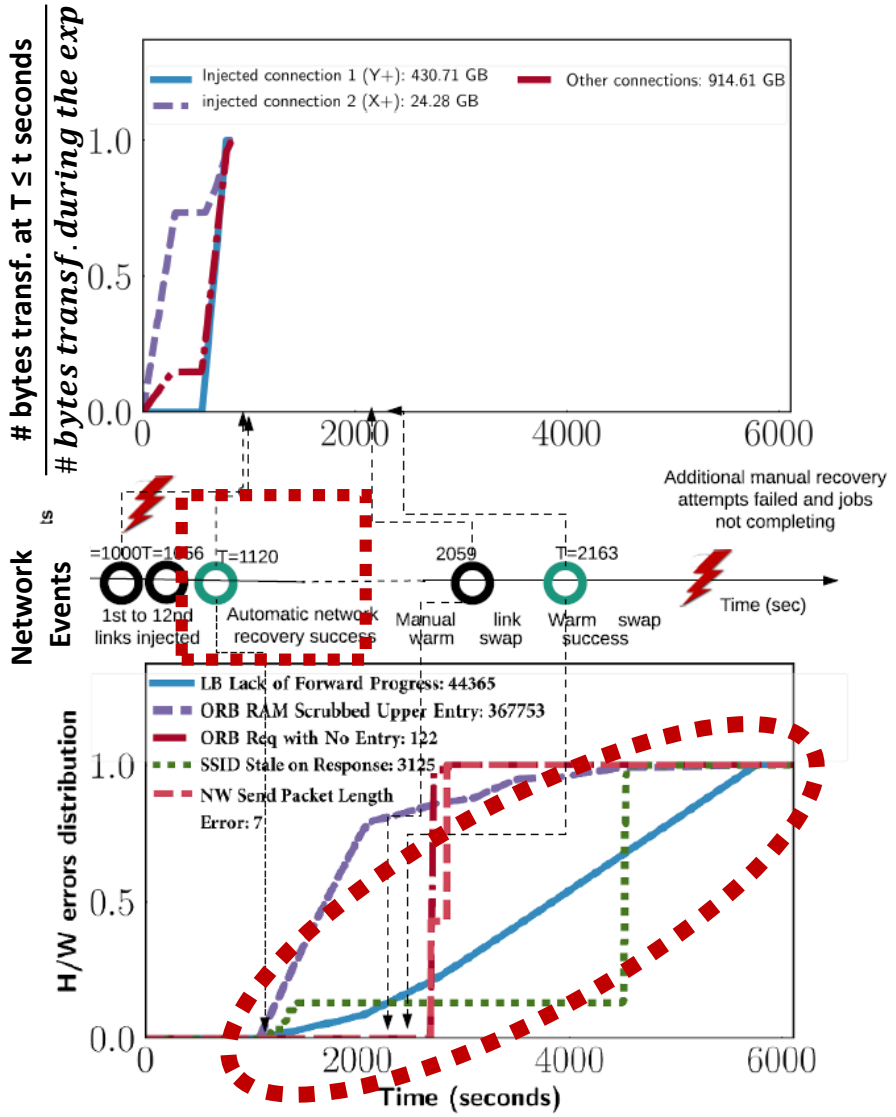


Injecting on Two Non-Overlapping Dimensional Links



Experiments are repeatable but outcome may vary due to non-determinism in the system

Advance analytics can help detect such cases
For example, continuing "ORB Scrubbed"/"Lack of forward progress" errors despite report of successful recovery is an indication of severe network issue



Conclusion

- Proposed and designed HPCArrow to inject network faults
  - Tool tested on Cray XE systems
- Easy to recreate failure scenarios in a repeatable way
- New insights in fault-to-failure propagation with respect to field-failure data analysis that can help build instrumentation and mitigation mechanisms

Future Roadmap: Resiliency Monitoring

- Fault injection on Cray Aries network in Muzea (SNL), Cray Gemini network in Blue Waters (NCSA)
  - Compare resiliency of the two network fabrics
- Extending HPCArrow to infiniband networks
  - Generalizing the tool for testing future systems and network topologies
- Use lessons learned from fault injections to drive detection and monitoring in future systems

Any Questions...
Just Ask!