U.S. DEPARTMENT OF ENERGY | Office of Science

Holistic Measurement Driven Resilience (HMDR)

Combining Operational Fault and Failure Measurements and Fault Injection for Quantifying Fault Detection, Propagation, and Impact

Principal Investigator(s): William Kramer, James Brandt, Ravishankar Iyer, Zbigniew Kalbarczyk, James Lujan, Nicholas J. Wright and Larry Kaplan (Senior Partner)
Senior Staff: Ben Allan, Amanda Bonnie, James Botts, Jeremy Enos, Valerio Formicola, Ann Gentile, Annette Greiner, Mike Mason, Mike Showerman, Cynthia Thackeray
Students: Fei Deng, Daniel Chen, Saurabh Jha

Extreme-scale systems have billions of hardware components and hundreds of millions of lines of software that must all work in perfectly coordinated fashion. Complex failure modes can impact their ability to serve their intended purposes at the quality and reliability desired.

Using data from six centers that house many of the largest high-performance computing (HPC) resources in the world, this project will provide unparalleled insight into the resilience challenges of future systems.

Distinguishing Features:

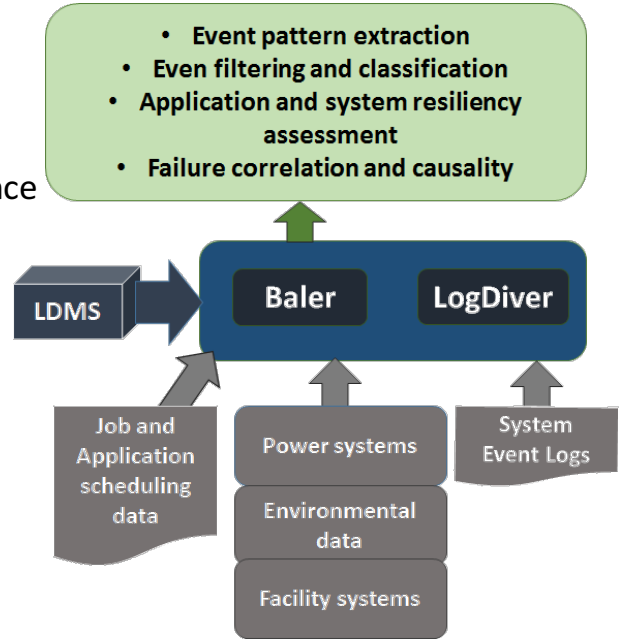
- Use **single vendor multi-site and multi-generation platforms, guided by vendor domain knowledge**, to understand, and accelerate development of, resilience mechanisms and instrumentation
- Develop **scalable processing** (including streaming) and feedback capabilities to enable use of **whole system** data sets (~50TB) from all information sources
- Use **fault injection in combination with field data** analytics to identify fault/error propagation and improve diagnosability and detectability of faults/errors.
- **Accelerate resilience research** through:
  - **Publication of** methods, algorithms, and **datasets**
  - **Building a community** of researchers and practitioners to perform cross-platform data analysis

Data Collection, Exploration and Analysis Tools

- **Problem Statement:**
  - Relationships of raw system and log data with higher level system behaviors are unknown
  - Lack of proven tools, techniques for discovery and analysis of faults in multiple-platform HPC datasets
  - Large data sets (~50TB) require scalable processing
- **Solutions: *Architectural Enhancements to HMDR tools***

  HMDR tools:
  - *Baler* identifies "event patterns" corresponding to failures and performance
  - *LogDiver* processes error signals to identify failure propagation paths
  - *LDMS* samples numeric instrumentation sources

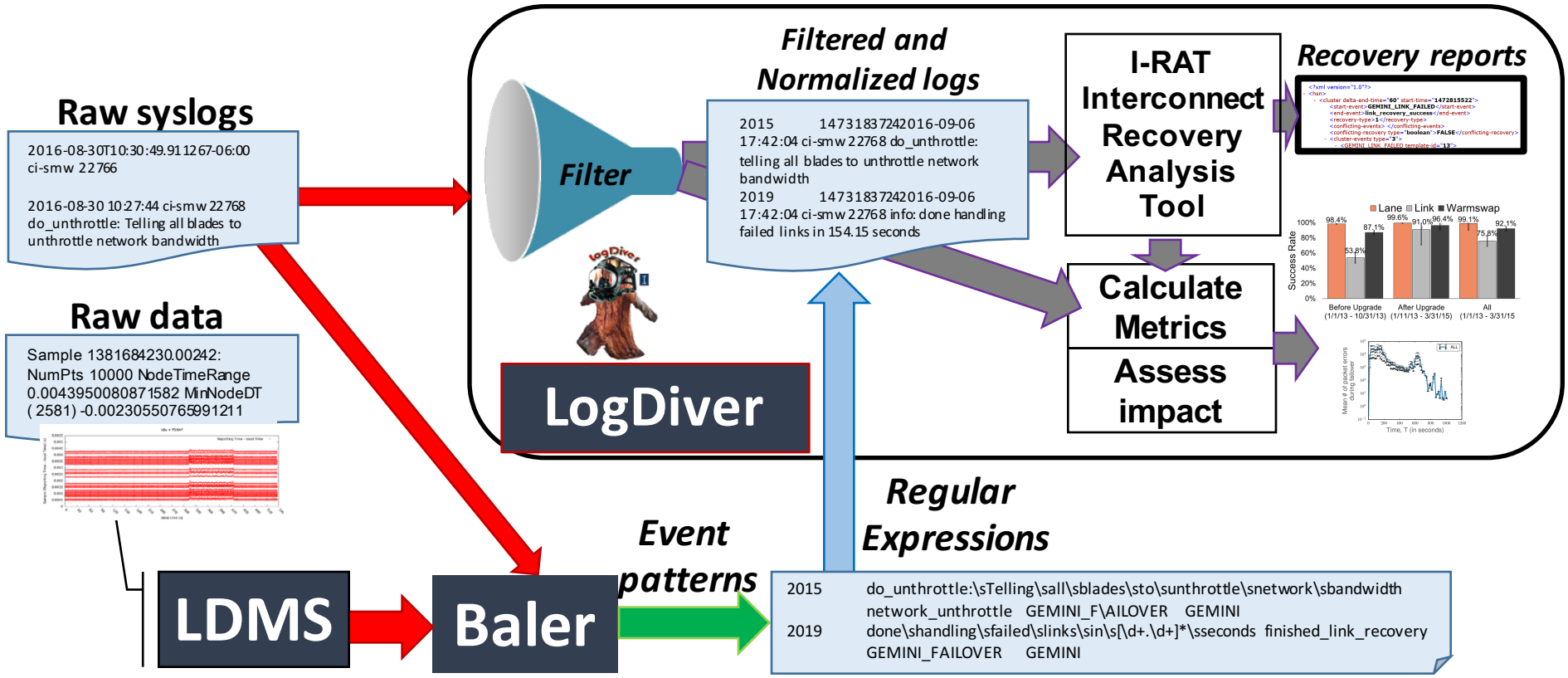
  Enhancements:
  - Faster processing through distributed architectures
  - Easier analysis of different (including cross-generations) systems' data through modular analysis support
  - Higher fidelity system data made available through support for sub-second system data sampling
  - Increased interoperability of the overall toolset



HMDR tools' interaction for data analysis

- **Impacts:**
  - Collecting multiple terabytes/day of failure and performance data from all system levels
  - Demonstrated the developed tools aid understanding of failures on several generations of Cray systems

HMDR Workflow



Lightweight Distributed Metric Service

High Fidelity Data Collection with Minimal Overhead

```
foo1/meminfo:Fri Feb 10 12:50:25 2017
M u64 component_id    1
D u64 MemTotal        1884188
D u64 MemFree         828244
D u64 MemAvailable    1639232
```

```
foo1/meminfo:Fri Feb 10 12:50:25 2017
M u64 component_id    1
D u64 MemTotal        1884188
D u64 MemFree         828244
D u64 MemAvailable    1639232
```

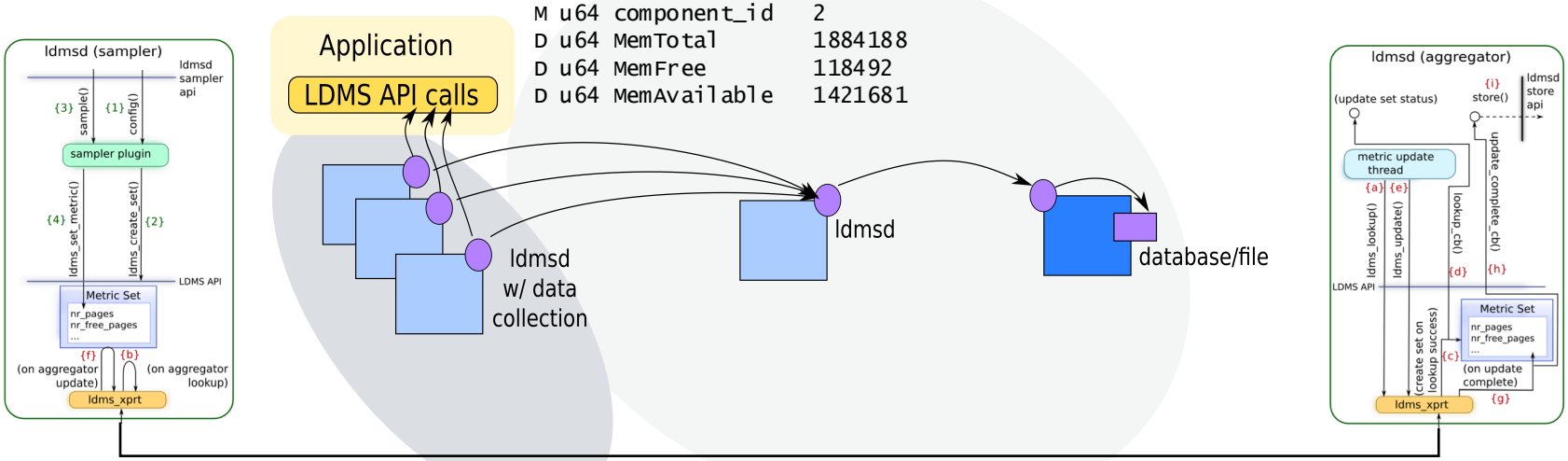
```
foo2/meminfo:Fri Feb 10 12:50:25 2017
M u64 component_id    2
D u64 MemTotal        1884188
D u64 MemFree         118492
D u64 MemAvailable    1421681
```

CSV Store output format:

#Time,Time_usec,ProducerName,component_id, MemTotal,MemFree,...

1486936884.008541,2134,foo1,1,1884188,828244,1639232,...

1486936884.008541,1592,foo2,2,1884188,118492,1421681,...



Compute Nodes

Aggregation Node(s)

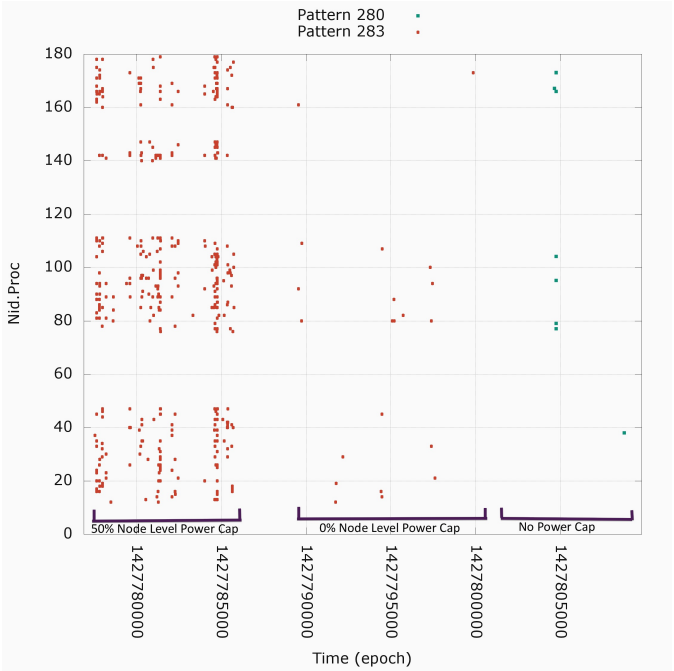
Storage Node(s)

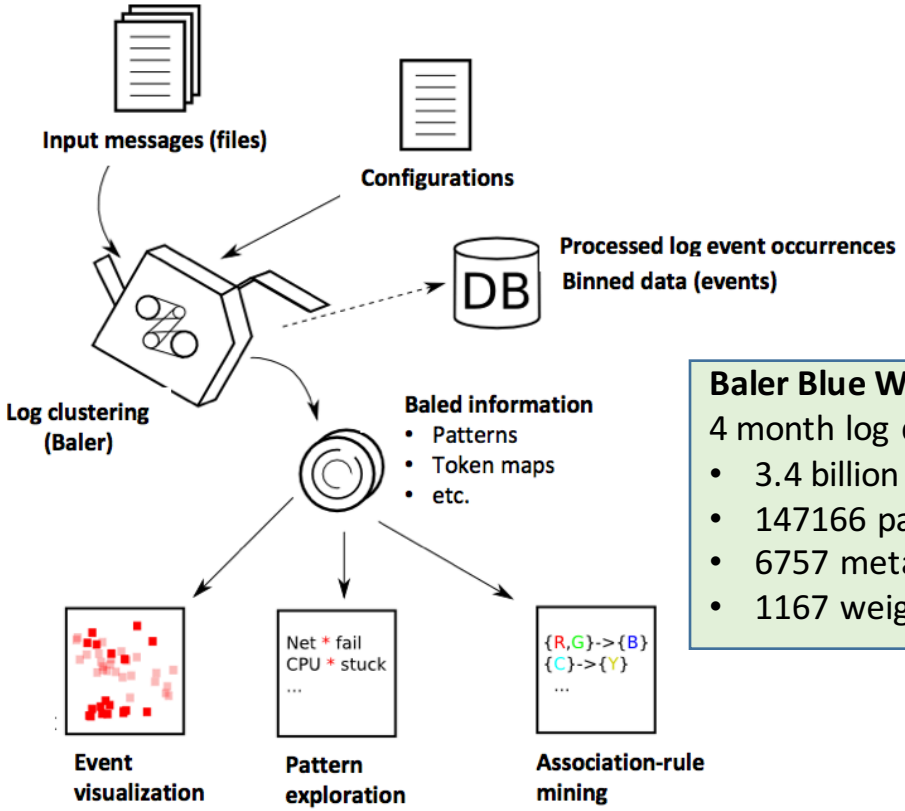
* Only the current data is retained on-node

Baler: Log and Numeric Data Processing Tool

Discovers Log and Numeric Patterns with no user input

Input messages:
280: bcsysd 2080 -- Node 2 interrupt IREQ=0x20000, USRA=0x0, USRB=0x80 USRB[7]: C0_PROCHOT CPU 0 Processor Hot
283: bcpmd 2140 -- Node 2 power budget exceeded! Power=340, Limit=322, Max Correction Time=6

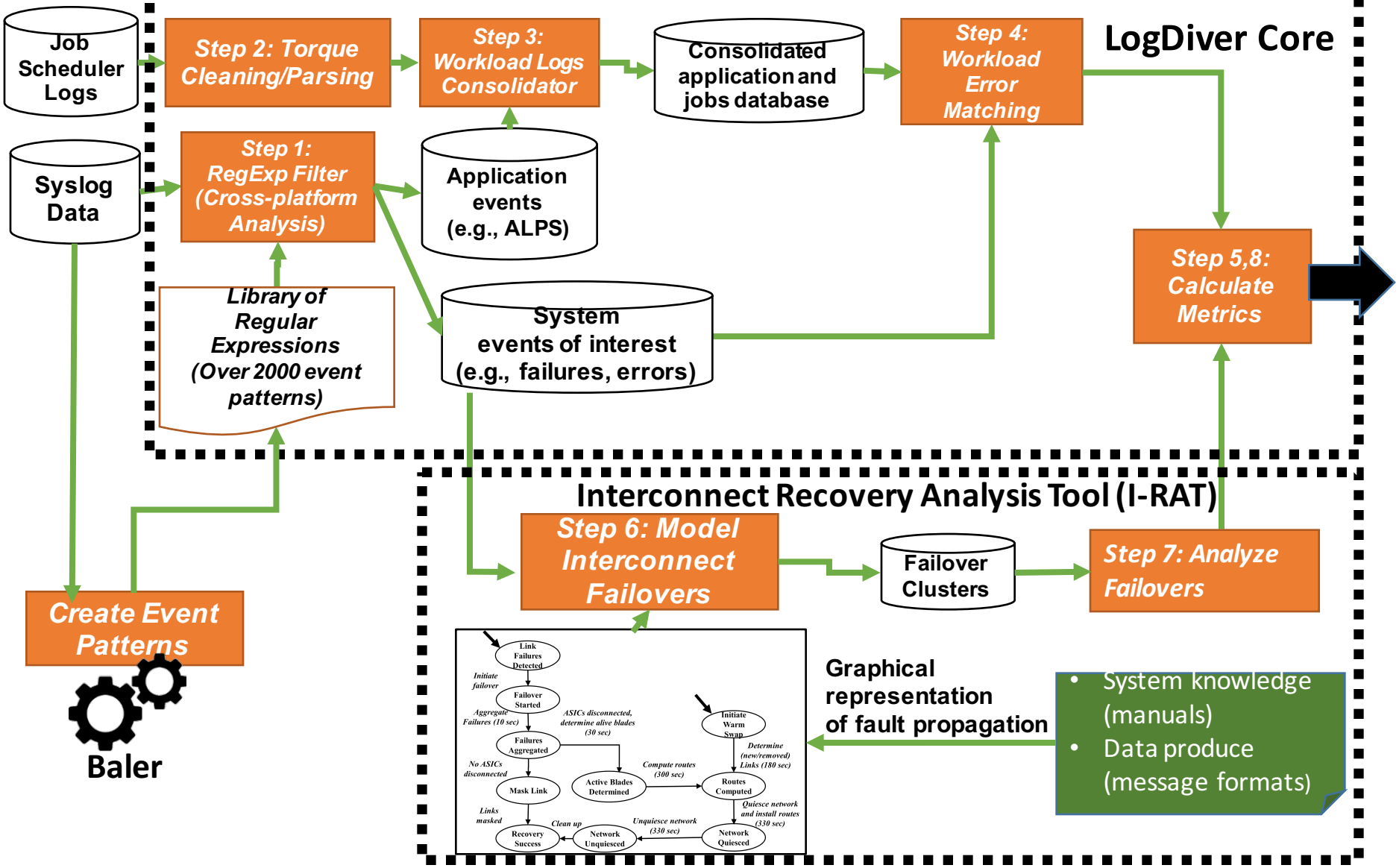




Baler Blue Waters Statistics:
4 month log data =
- 3.4 billion messages
- 147166 patterns
- 6757 meta-patterns
- 1167 weighted meta-patterns

Output patterns:
280: * * -- Node * interrupt *=*, *=*, *=* *[*]: * * * Processor Hot
283: * * -- Node * power budget exceeded! Power=*, Limit=*, * Correction Time=*

LogDiver


Job Scheduler Logs
Step 2: Torque Cleaning/Parsing
Step 3: Workload Logs Consolidator
Consolidated application and jobs database
Step 4: Workload Error Matching
LogDiver Core
Syslog Data
Step 1: RegExp Filter (Cross-platform Analysis)
Application events (e.g., ALPS)
Step 5,8: Calculate Metrics
Library of Regular Expressions (Over 2000 event patterns)
System events of interest (e.g., failures, errors)
Interconnect Recovery Analysis Tool (I-RAT)
Step 6: Model Interconnect Failovers
Failover Clusters
Step 7: Analyze Failovers
Create Event Patterns
Baler
Link Failures Detected
Initiate failover
Failover Started
Aggregate Failures (10 sec)
ASICs disconnected, determine alive blades (30 sec)
Failures Aggregated
Initiate Warm Swap
Determine (new/removed) Links (180 sec)
No ASICs disconnected
Compute routes (300 sec)
Active Blades Determined
Routes Computed
Mask Link
Quiesce network and install routes (330 sec)
Links masked
Clean up
Unquiesce network (330 sec)
Recovery Success
Network Unquiesced
Network Quiesced
Graphical representation of fault propagation
System knowledge (manuals)
Data produce (message formats)

Log File Fault to Failure Path Discovery

Log lines:

kernel - p0-20150625t082211 - LNet: 17821:0:(lib-move.c:851:lnet_post_send_locked()) Dropping message for 12345-202@gni: peer not alive (1441199966 kern 6)

Baler patterns:

MetaPtn 733 386528 occurrences * * •-• * *: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•@*: peer not alive (• * •)

+3761 191009 occurrences - •-• - LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•@gni: peer not alive (• kern •)

+3796 190964 occurrences • • •-• [console@•] LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•@gni: peer not alive (• • •)

+11830 kernel - •-• - LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•.•.•.•@•: peer not alive (• kern •)

+12042 • • •-• [console@•] LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•.•.•.•@•: peer not alive (• • •)

+28238 • • •-• [console@•] <•>LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•.•.•.•@•: peer not alive (• • •)

+28602 • • •-• [console@•] <•>LNet: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•@gni: peer not alive (• • •)

+123970 • • •-• [console@•] •(•: •:•:(lib-move.•:•:lnet_post_send_locked()) Dropping message for •-•@gni: peer not alive (• • •)

Baler Auto-Regex:

3761:

\- [0-9A-Za-z]+\-[0-9A-Za-z]+ \- LNet: [0-9A-Za-z]+:[0-9A-Za-z]+:\(lib\-move\.[0-9A-Za-z]+:[0-9A-Za-z]+:lnet_post_send_locked\(\)\) Dropping message for [0-9A-Za-z]+\-[0-9A-Za-z]+@gni: peer not alive \([0-9A-Za-z]+ kern [0-9A-Za-z]+\)

Log Diver Regex (category annotations added manually)

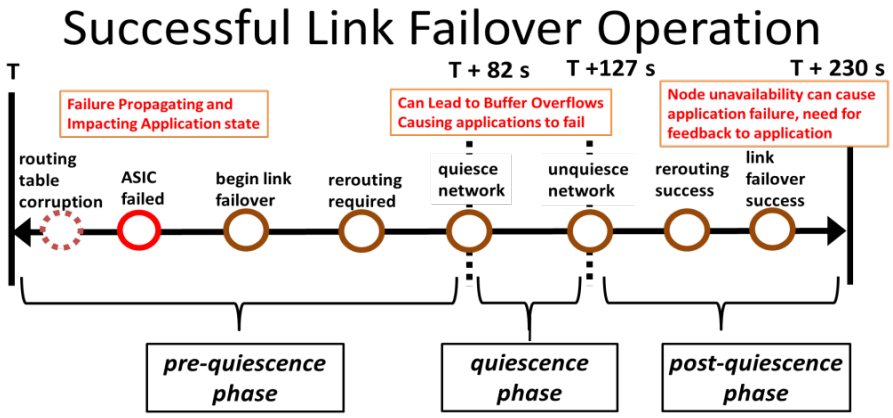
1	ERROR\s*powerSlotDown:	**EPO HARDWARE BLADE/NODE**
2	\@gni:\s*peer\s*not\s*alive	**LNET_PEER_NOT_ALIVE LNET NETWORK**
3	LustreError:.*Got\s*rc\s*\-108\s*from\s*cancel\s*RPC:\s*canceling\s*anyway	**LUSTRE_TRANSPORT_END_SHUTDOWN LUSTRE STORAGE**
4	LOG_ERROR::send_hierarchy,\s*Could\s*not\s*send\s*mom\s*hierarchy\s*to\s*host	**MOAB_MOM_MISCOMMUNICATION MOAB SCHEDULING**
5	Response\s*Protocol\s*Error	**GEMINI_PROTOCOL_ERROR GEMINI NETWORK**
6	\s*failed\s*command:\s*WRITE	**SONEXION_IO_ERROR SONEXION STORAGE**

Fault to failure path discovery:

Log Diver using spatial and temporal information of actual log messages and message categories matched in its Regex library

Fault-to-Failure Characterization

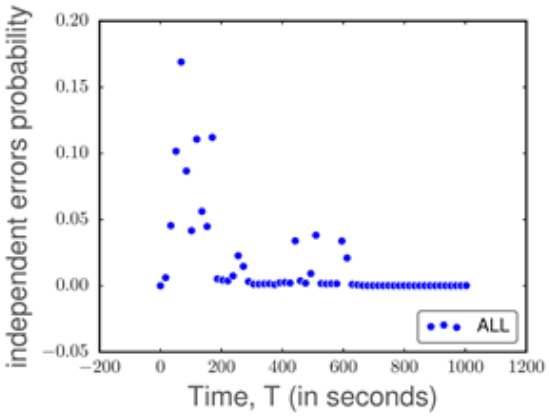
- **Problem Statement:**
  - Fault-to-failure paths largely unknown
  - Difficulty in failure attribution and early intervention
- **Solutions:**
  - Data-driven construction of failure propagation and recovery models to study failure propagation paths in peta-scale systems (e.g., sub-phases of link failovers)

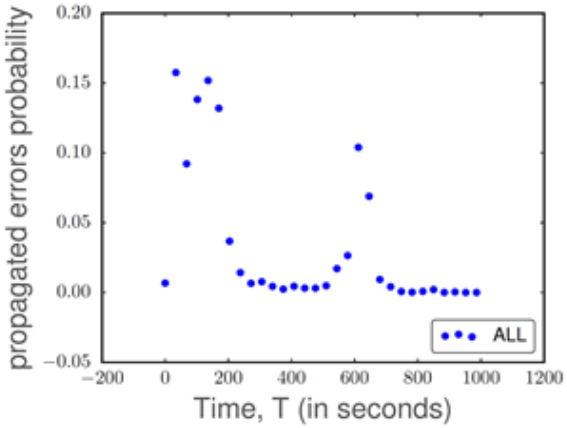


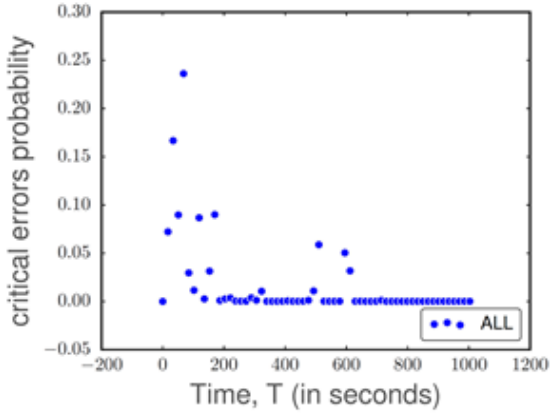
Fault-to-failure models include timing information to drive instrumentation and response options

- **Impacts:**
  - Identified failure modes (based on data) used for design of fault injection experiments for validation and testing
  - Identification of failure characteristics:
    - Lengthy (from 1 to 10s of minutes) multi-phase recovery operations
    - Non-negligible chances of multiple failures during recovery operations
    - Unsuccessful recoveries of critical subsystems (e.g. network fabric) causing system-wide outages
    - Applications fail with a non-negligible probability even during a successful recovery of critical subsystems
    - Length of recovery operations correlating with the breadth and impact of failures
  - Determination of failure models to enable future instrumentation for discovering and mitigating resilience problems

Additional Failures During Failovers







- **Propagated failures (84.4%) occur when other components fail close in time and in the same topological hierarchy of the failure that triggered the recovery in the first place**
- **Independent failures (15.6%) are failures that are not related to the failures that triggered the recovery in the first place**
- **The probability of failure during the failover decreases during quiescence (i.e., when time increases), because quiescence reduces all the system activity and hence it reduces the chance of failure**

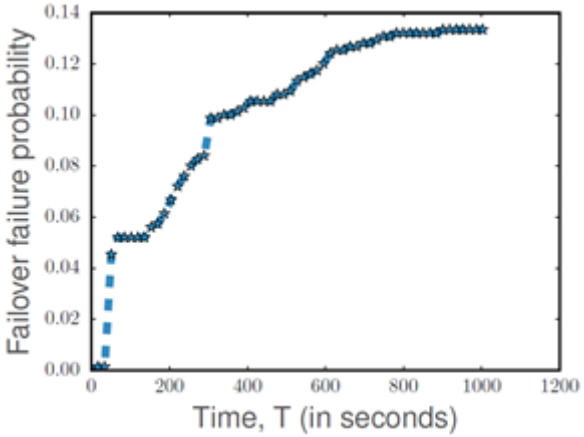
- **Critical events are highly correlated with the failure of recovery. They happen ahead of the end of failover, and can be used for preemptive detection of unsuccessful recovery.**
  - **For example, additional Link failure during the failover precedes the failure of failover on average 90 secs, with Probability 85%**

Time Characterization of Network Recoveries

Longer the duration of failover, higher the chances of experiencing additional failures.

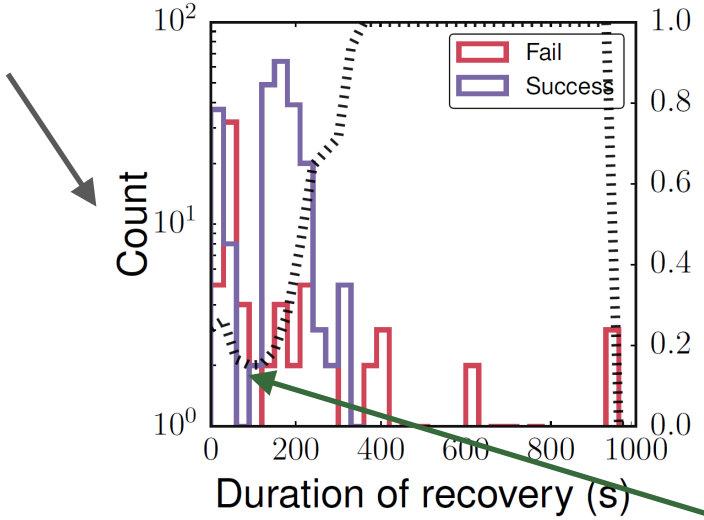
Link Failover

After ~180 sec, probability of failure of the failover is higher than probability of successful link failovers

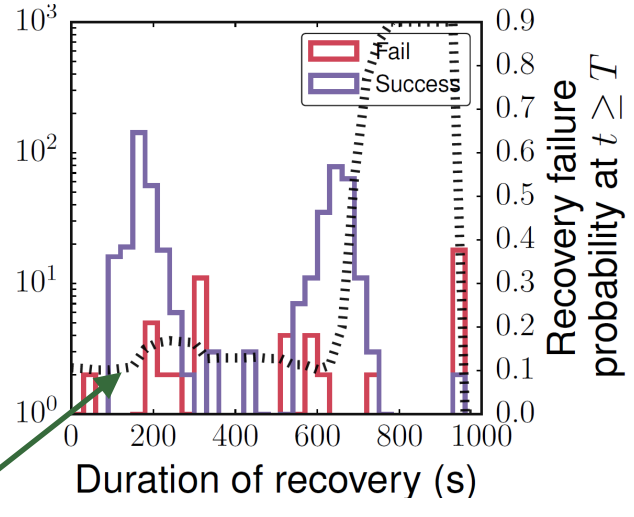


Warm Swap

As the time increases the probability of the failure of a failover increases until ~180 seconds and then after ~600 seconds, the probability that a failover will succeeds becomes zero



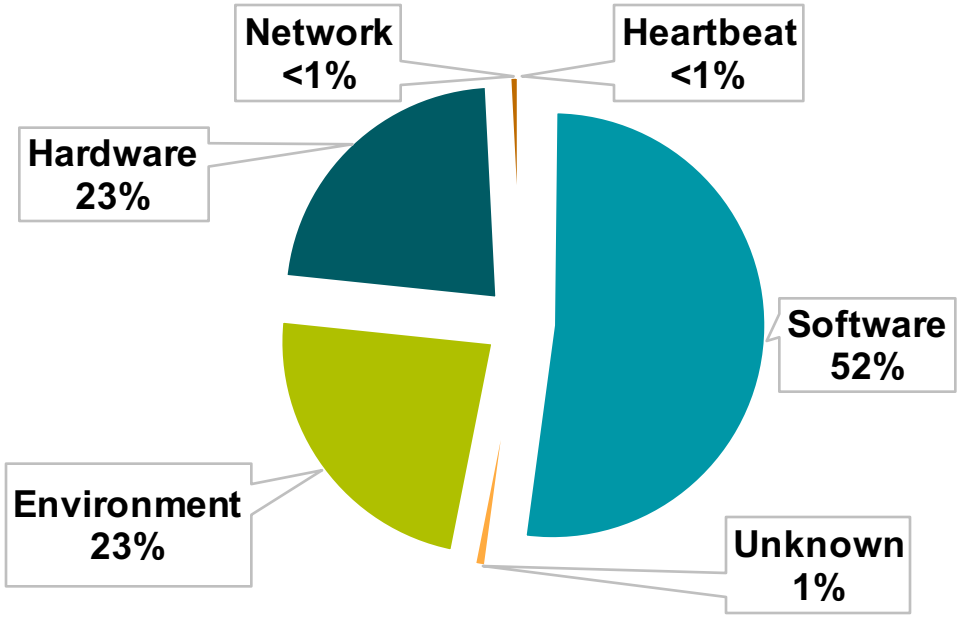
Link



Warm Swap

Optimal Recovery time ~70 – 180 seconds

Repair Hours Breakdown by Category



- Software-related failures frequent in critical failure categories*
  - 23% of all failures and 53% of repair hours compared to 52% of failures and 23% of repair hours for hardware
  - 14% of software errors propagate to more than 1 blade
  - Software Failure rate increased 2x in 9 months

*Breakdown for the Blue Waters supercomputer

Application Resiliency vs. Scale

- Similar behavior of CPU and Hybrid applications:
  - P(application failure) constant (~10^{-3}) from small scale (1 node) to 25% of the system
  - At full scale, P(application failure) increases sharply to 0.18 for CPU based and 0.1 for hybrid apps
- Making applications last longer could make things worse
  - P(application failure for system errors | scale = X node hours)
  - Quadratic function of scale for CPU-based nodes
  - Cubic function of scale for hybrid nodes
  - Need to rethink how to resiliently harness hybrid computing @scale

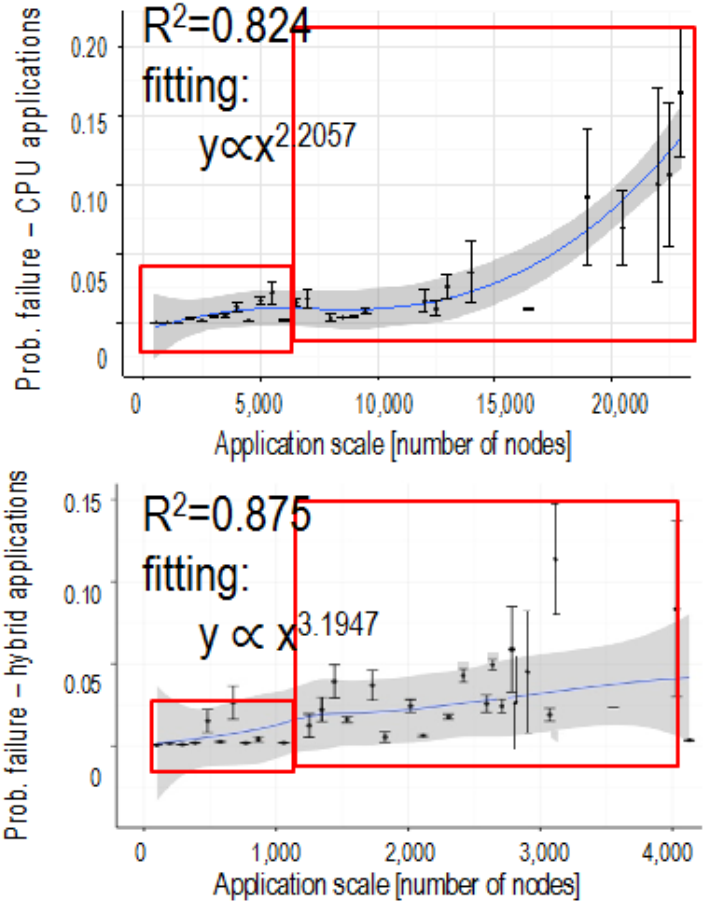
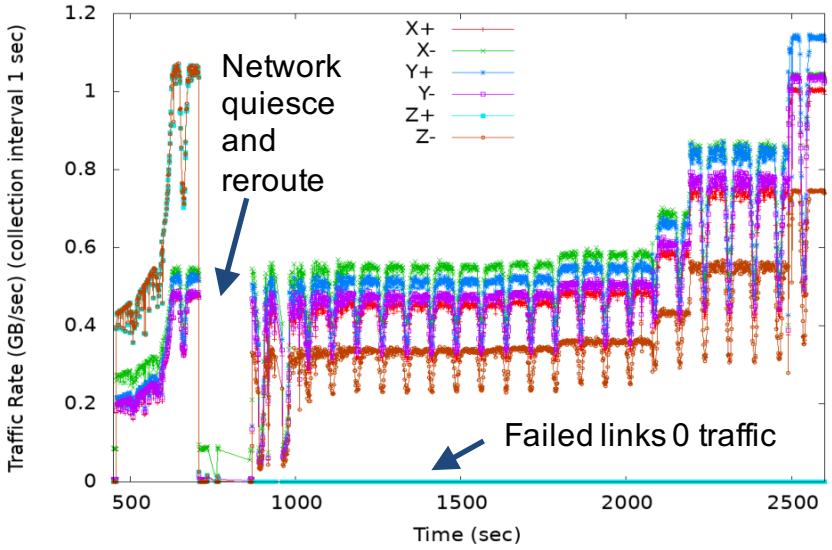


Figure: Probability of application failure due to system errors with increasing scale

Fault Injection (FI)/Experimental Evaluation

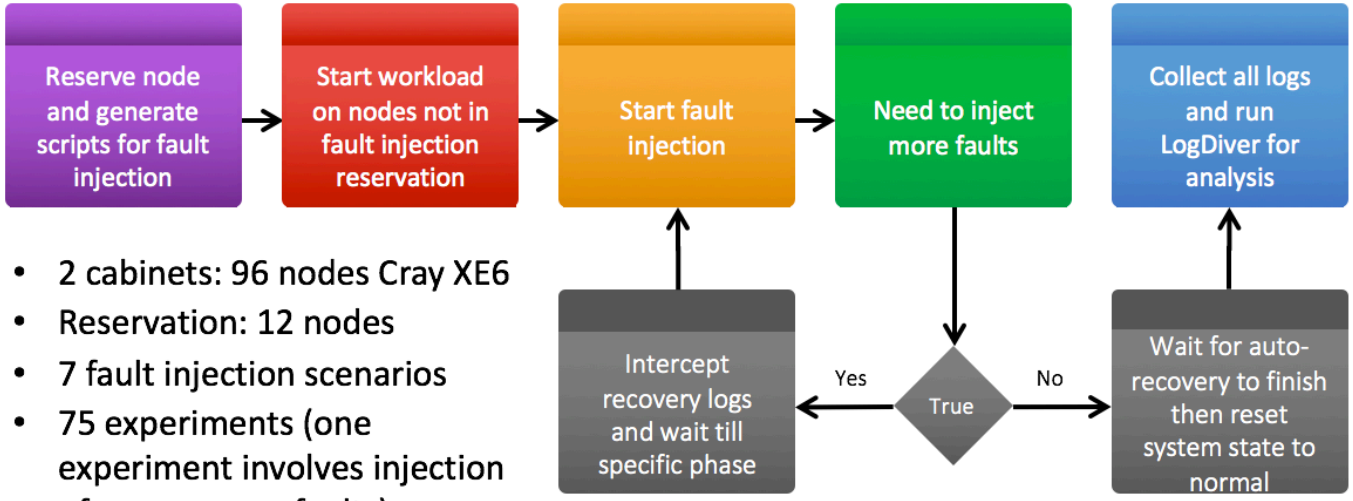
- **Problem Statement:**
  - Fault attribution to initial causes often tenuous due to sporadic occurrence in production
  - More difficult in production, due to lack of control on system state (e.g., workload) and log data volume
  - Hard to assess value of current hardware instrumentation due to lack of holistic measurements
- **Solutions:**
  - Developed, validated, and performed experiments based on observed/inferred fault/failure scenarios from production fault data
  - Developed a toolset that generates failure scenarios of increasing complexity
  - Used a repeatable methodology, on:
    - a large scale (~9,000 nodes, ACES Cielo) system
    - system testbeds with similar hardware
  - Analyzed experiment logs for fault paths/propagations
  - Associated measured observables with failures

Network traffic shows impact of failure and recovery mechanisms

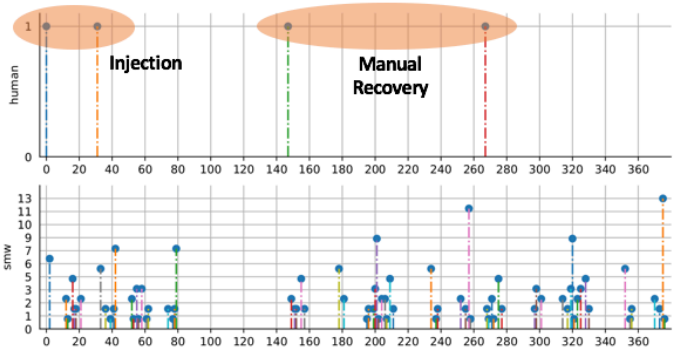


- **Impacts:**
  - Exposed not only failure paths but also failures in the platform's failover/restore mechanisms
  - Validated fault propagation paths observed in the field failure data and observed new faults
  - Developed higher fidelity fault propagation path models than we previously had from log analysis alone
  - Fault propagation timings and observables can drive new instrumentation recommendations
  - Uniform and repeatable methodology for other platforms with respect to both experiments and analysis

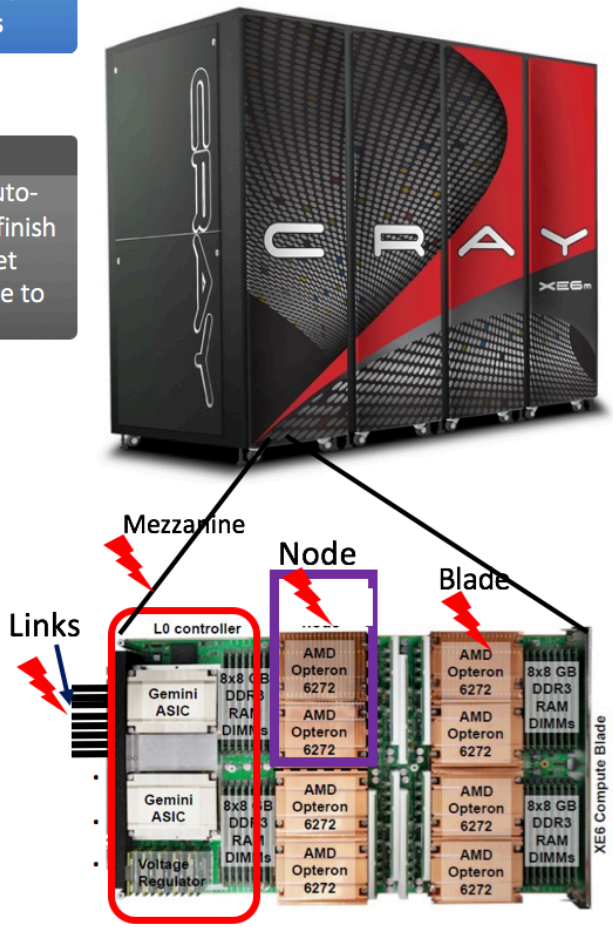
Fault Injection Setup and Workflow



- 2 cabinets: 96 nodes Cray XE6
- Reservation: 12 nodes
- 7 fault injection scenarios
- 75 experiments (one experiment involves injection of one or more faults)
- Workload (10 nodes): Intel MPI Benchmarks (IMB)
- Use LogDiver on raw logs to generate human readable data
  - Use script to collect data from console logs, hardware error logs, network monitors, LDMS, etc.
- Plot data filtered by LogDiver for visualization (sample output to the right)

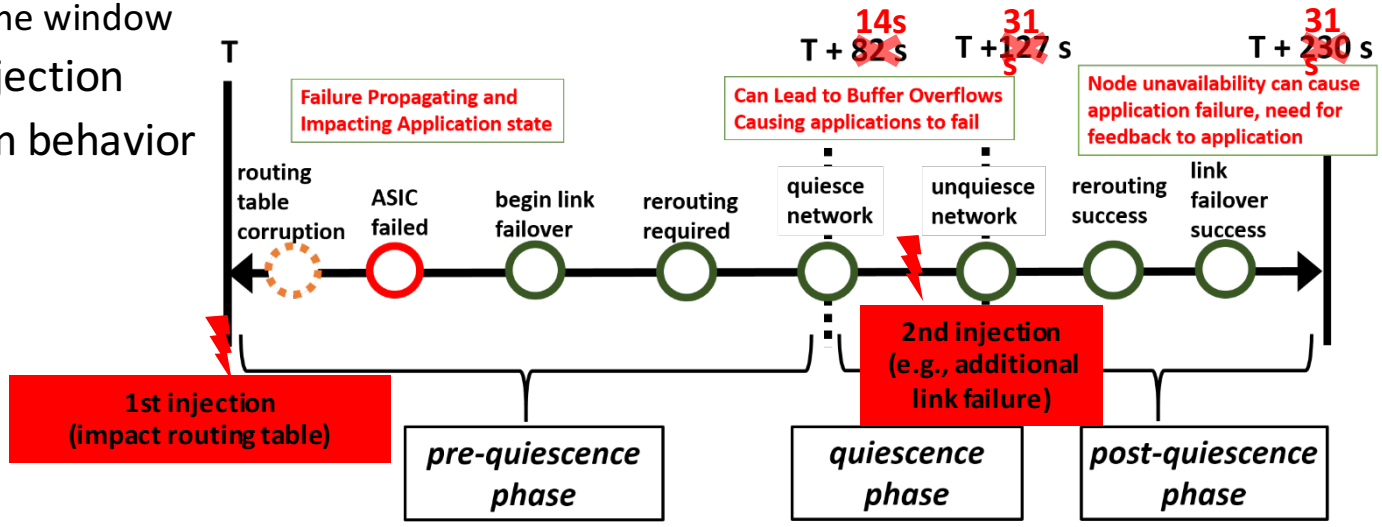


Timeline of injections and corresponding event count observed on the System Monitoring Workstation



Fault Injection Challenges

- Fault injection automation
  - Develop scripts to keep track of different fault injection experiments
  - Self-generation of scripts to initiate and run experiments
- Hard to re-produce faults seen in real data
  - Target system (JYC) much smaller than real machine: 96 nodes vs. ~28K nodes (Blue Waters)
  - 2-D routing in JYC vs 3-D in Blue Waters
  - Much faster speed for the failover procedure to complete (e.g., quiescence phase 17 vs 45 seconds)
  - Need to monitor failover phase in real time for precise control
    - over injection time window
- Visualize the fault injection workflow and system behavior



Fault Injection on Large-Scale System

Cielo ~ 9K Nodes Cray XE6

Single Connection Down: c13-3c0s4 all Z+ links

```
Run FB1 - Random Single connection Down for medium app now? [y/n] y
Tue Sep  6 14:00:31 MDT 2016 *** Running FB2 - Random Single connection Down for medium app ***
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x2000000 2 0 [y/n]y
Tue Sep  6 14:03:03 MDT 2016 Executed link_down c13-3c0s4g1l00: xtmemio -w ::c13-3c0s4g1 0x2000000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x2002000 2 0 [y/n]y
Tue Sep  6 14:03:30 MDT 2016 Executed link_down c13-3c0s4g1l01: xtmemio -w ::c13-3c0s4g1 0x2002000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x2020000 2 0 [y/n]y
Tue Sep  6 14:03:53 MDT 2016 Executed link_down c13-3c0s4g1l10: xtmemio -w ::c13-3c0s4g1 0x2020000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x2022000 2 0 [y/n]y
Tue Sep  6 14:04:01 MDT 2016 Executed link_down c13-3c0s4g1l11: xtmemio -w ::c13-3c0s4g1 0x2022000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x204a000 2 0 [y/n]y
Tue Sep  6 14:04:07 MDT 2016 Executed link_down c13-3c0s4g1l25: xtmemio -w ::c13-3c0s4g1 0x204a000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x204c000 2 0 [y/n]y
Tue Sep  6 14:04:11 MDT 2016 Executed link_down c13-3c0s4g1l26: xtmemio -w ::c13-3c0s4g1 0x204c000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x204e000 2 0 [y/n]y
Tue Sep  6 14:04:16 MDT 2016 Executed link_down c13-3c0s4g1l27: xtmemio -w ::c13-3c0s4g1 0x204e000 2 0
Received all responses.
done with event loop
Proceed with execution of: xtmemio -w ::c13-3c0s4g1 0x206a000 2 0 [y/n]y
Tue Sep  6 14:04:22 MDT 2016 Executed link_down c13-3c0s4g1l35: xtmemio -w ::c13-3c0s4g1 0x206a000 2 0
Received all responses.
done with event loop
End Experiment and attempt to restore now? [y/n]y
```

14:03:05 c13-3c0s4g1l00 c13-3c0s5g1l32
1 Link Inactive

14:03:05 HWERR[c13-3c0s4g1l00]:NW Link Went Inactive

14:03:05 cb_hw_error: failed_component c13-3c0s4g1l00

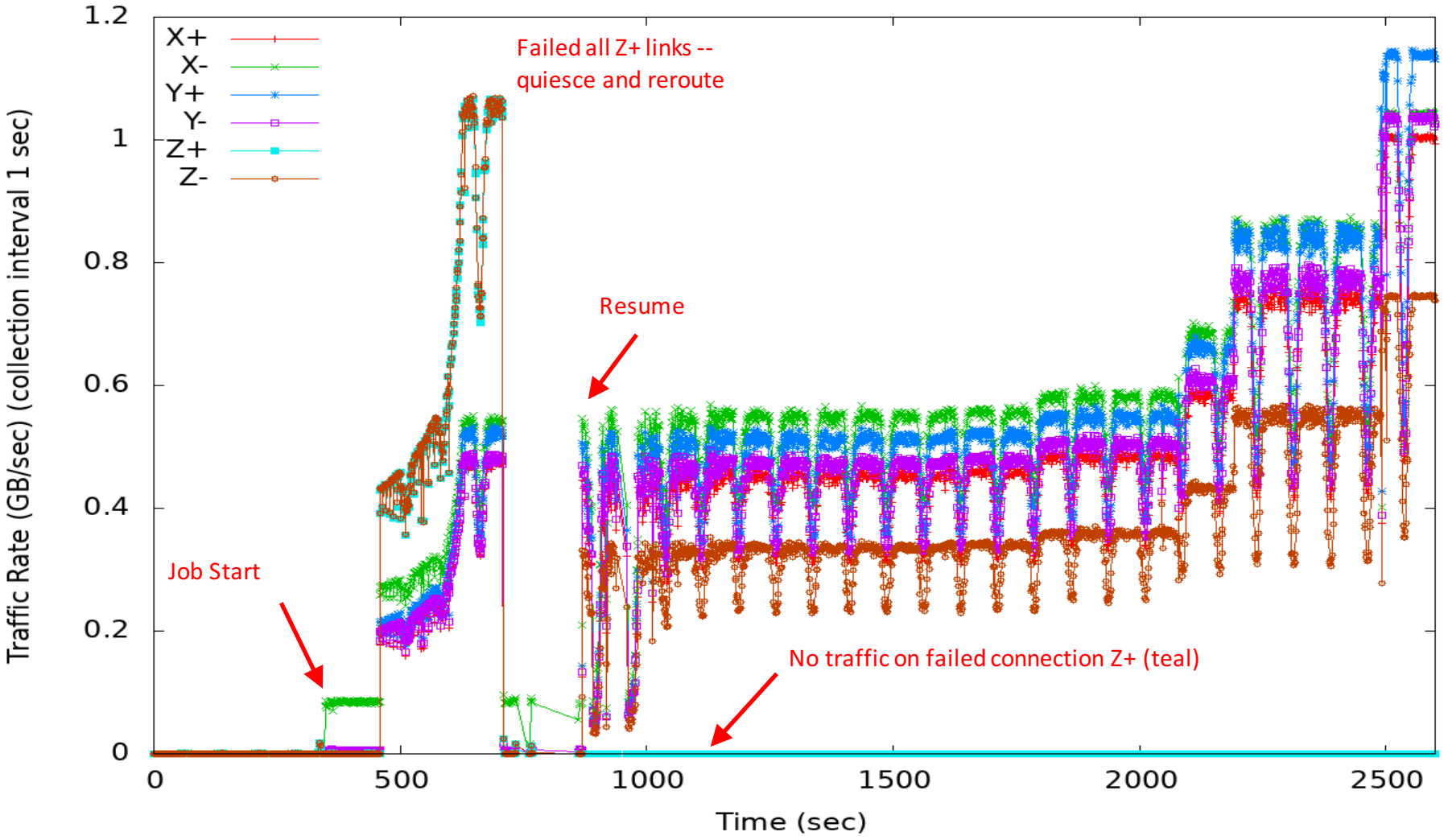
14:03-14:04 cb_hw_error: handling failed link c13-3c0s4gXlXX

14:03-14:05 do_route_compute: generated route_uuid 4415b96b7d2fbbc5f79b12498b86805

14:03-14:05 Executing command '/opt/cray/hss/default/bin/rtr -P -a --id 4415b96b7d2fbbc5f79b12498b86805

14:05:34 ci-smw 22768 do_unthrottle: Telling all blades to unthrottle network bandwidth

Network Traffic on Router with Failed Directional Links



Resilience Datasets for Researchers

- **Problem Statement:**
  - Resilience research hampered by lack of publicly available datasets from modern architectures
  - Difficulty exposing HPC log and operational data to researchers (security and privacy concerns)
  - Curation, analysis, transfer difficult due to size (TBs)
- **Solutions:**
  - Developed log file cleansing tools with multiple strategies
  - Worked with site specific security staff to determine cleansing levels
  - HMDR web portal for publicly released log datasets
    - NERSC-hosted site provides datasets and efficient file transfers
    - Two current ACES Trinity testbed system datasets released
    - Large-scale NERSC Edison dataset in cleansing
    - Additional data sets shared amongst team being prepared for release
- **Impacts:**
  - Meaningful, modern datasets released with 60+ downloads to date
  - Contacted by external researchers interested in collaboration on dataset analysis
  - Datasets enable cross-generational platform studies

The HMDR Project:

Holistic, Measurement-Driven Resilience

Combining Operational Fault and Failure Measurements and Fault Injection for Quantifying Fault Detection, Propagation and Impact

A collaboration between the University of Illinois at Urbana-Champaign (UIUC); Sandia (SNL), Los Alamos (LANL), and Lawrence Berkeley (LBNL) National Laboratories; and Cray Inc.

In HPC systems to date, application resilience to failures has been accomplished by the brute-force method of checkpoint/restart, which allows an application to make forward progress in the face of system faults, errors, and failures independent of root cause or end result. It has remained the primary resilience mechanism because we lack a way to identify faults and project results early enough to take meaningful mitigating action. Because we have not yet operated at scales at which checkpoint/restart cannot help, vendors have had little motivation to provide the instrumentation necessary for early identification. However, as we move from petascale to exascale, mean time to failure (MTTF) will render the existing techniques ineffectual. Instrumentation allowing early indication of problems and tools to enable use of such information by systems, OSes, and applications would offer an alternative, more scalable solution.

In this work, we are building on our experience and expertise developed and accumulated over years of research on design, monitoring, measurement, and assessment of resilient computing systems. Analysis of field data on the current and past generations of extreme-scale systems has revealed several challenges that, if not addressed, may hinder the effectiveness of future exascale computing systems. Specifically, i) file system and interconnect in current-generation large-scale systems already operate at the margins of resiliency and may not scale to larger deployments; ii) automated software-based failover mechanisms are frequently inadequate , such that failures during recovery may lead to system/application failures, including system-wide outages; iii) silent data corruption represents a critical fault mode and will require efficient detection mechanisms if next-generation applications are to take full advantage of exascale hardware; and iv) application-level resilience must be the final arbiter of system effectiveness.

To address the above challenges, we have assembled a team of world-renowned experts in resilient extreme-scale computing from the University of Illinois (Electrical and Computer Engineering, Computer Science, and NCSA), SNL, LANL, NERSC, and Cray. Our team includes representatives from centers that will house many of the largest HPC resources in the world, both today and over the coming years. The team has a unique track record of research in i) system and application failure characterization based on the analysis of field data, ii) data-driven design of fault/error detection mechanisms, and iii) experimental characterization of system/application resiliency. The team includes system owners/operators who can guarantee continuous data collection and access and ensure installation of appropriate analysis tools.

Available Datasets

Note: All datasets are available via Globus for efficient file transfer. We recomend using Globus for datasets more than a few GB in size. To use Globus, point your browser to globus.org, log in, and go to the Transfer Files page. (If you don't have a Globus account, you can sign up for free at globus.org.) In the endpoint selection field, select **NERSC_resilience**, and then select the endpoint to which you wish to transfer the files. You will not need to authenticate to the NERSC endpoint, though you will need to meet the authentication requirements of the destination. Click to select the files you want and click an arrow to move them.

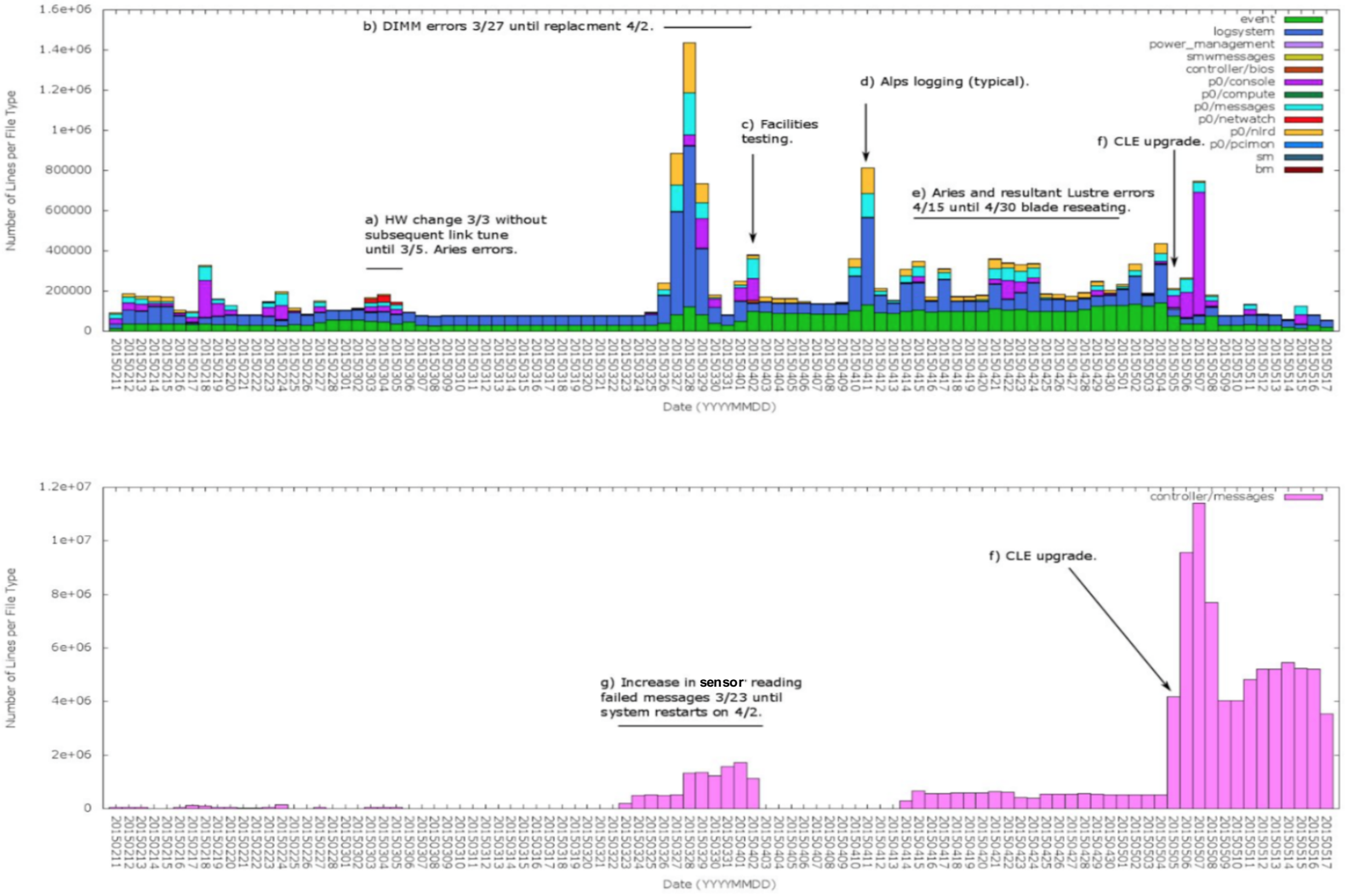
Mutrino 2/15-6/16 Dataset (3.5 GB .tgz file) | About the dataset (80K PDF)

Mutrino 2/15-5/15 Dataset (648 MB .tgz file) | About the dataset (360 K PDF)

Principal Investigators

- William T. C. Kramer, Lead Principal Investigator (PI), University of Illinois
- James Brandt, Institutional PI, Sandia National Laboratory
- Ravi Iyer, Institutional PI, University of Illinois at Urbana-Champaign
- James Lujan, Institutional PI, Los Alamos National Laboratory
- Nicholas Wright, Institutional PI, National Energy Research Scientific Computing Center, Lawrence Berkeley National Laboratory

HMDR Datasets include standup and production operation with both natural and induced network, electrical, thermal, and functional failures



Annotated timeline of events from the Mutrino 2/15-5/15 dataset

From "New Systems, New Behaviors, New Patterns: Monitoring Insights from System Standup", J. Brandt, A. Gentile, C. Martin, J. Repik, and N. Taerat Workshop on Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA) at IEEE CLUSTER 2015.

HMDR Cleansing Tools

<150>1 2016-06-02T09:12:55.983001-06:00 c0-0c1s1n1 aprun 15424 p0-20160403t123157 [alps_msgs@34] apid=547905, Starting, user=24075, batch_id=136836.mutrino-moab, cmd_line="aprun -n 4 -N 4 -S 2 --cc numa_node -d 4 /projects/scs/THERE_IS_A_USER\NAME_HERE/partisn micsn.inp micsn.4.output ", num_nodes=1, node_list=12, cwd="/lscratch1/THERE_IS_A_USER\NAME_HERE/run_pat_4" /other/item/here

Options for handling of user names, paths revealing directory structures, IP addresses, etc.

Options for removal vs. anonymization

Cleanser

<150>1 2016-06-02T09:12:55.983001-06:00 c0-0c1s1n1 aprun 15424 p0-20160403t123157 [alps_msgs@34] apid=547905, Starting, user=24075, batch_id=136836.mutrino-moab, cmd_line="aprun -n 4 -N 4 -S 2 --cc numa_node -d 4 PATHU7P micsn.inp micsn.4.output", num_nodes=1, node_list=12, cwd="PATHU8P" /other/item/here

Cleanse only paths with usernames

path|pathuser -- two options for cleansing. All path options are mutually exclusive. pathuser must also have user. These take one arg: mapout 1) mapout = output file for the map of paths to the cleansed tag

These dynamically discover what to replace. For path, if it is a path independent of user inclusion, it will be replaced. For pathuser, if it is a path with a username in it, then it is replaced. The replacement tags look like PATH100P or PATHU100P.

pathfs -- one option for cleansing. All path options are mutually exclusive. pathfs must also have user.
This takes one arg: fslist
1) fslist = input file for the list of filesystems names starting a path to be retained.

This dynamically discovers what to replace. If it is a path with a username, then it is replaced; otherwise not. If it is going to be replaced it will retain the leading filesystem name if it is in the list; otherwise not. The replacement is generic "PATH" independent of content. There is no output. This option enables path cleansing while still allowing completely parallel runs since there is no dynamic map to be synchronized.

ipaddrunk -- one option for cleansing. All ipaddr options are mutually exclusive.
This takes one arg: ipaddroct
1) list of 1st octet of ipaddr values to keep (e.g., 10, 127). Any ipaddrs starting with a different octect will be replaced with a generic "IPADDRUNK" independent of content. The replaced addr are put into a file of the same name as the input file with ".ipaddrunk" added.
This option enables ipcleansing while still allowing completely parallel runs since there is no dynamic map to be synchronized.

ssh|sfw|secure - three options for totally removing a line. There are no options.
ssh removes a variety of ssh-related lines from the files.
sfw removes a variety of drop packet lines from the files.
ssecure removes a variety of lines matching certain keywords (e.g, firewall) from the files.
These three options will occur before any other checking.

The removed lines will be put into separate files with the same name as the input files, but with ".sshredact" or ".sfwredact" or ".secureredact" added.

<150>1 2016-06-02T09:12:55.983001-06:00 c0-0c1s1n1 aprun 15424 p0-20160403t123157 [alps_msgs@34] apid=547905, Starting, user=24075, batch_id=136836.mutrino-moab, cmd_line="aprun -n 4 -N 4 -S 2 --cc numa_node -d 4 /projects/PATH micsn.inp micsn\.4.output ", num_nodes=1, node_list=12, cwd="/lscratch1/PATH" /other/item/here

Cleanse but retain specified filesystem names

Diversity of Log Formats in Research Datasets

Analysis requires correlation of events from many sources

bootmanager <158>1 2015-02-03T14:17:40.235359-08:00 edismw bootmanager 64922 - [hss_bm@34] Failing node c0-0c0s0n1. Error code = 321 (Node does not accept shutdown events)

boot info [2015-01-13T08:14:11.462009-08:00] All 1439 expected responses were received.

commands 2015-03-24 00:00:11 653 START xtfileio -r -v --type=l1 s0 /var/tmp/DeviceLauncher /tmp/%sxtcheckhss_partLauncher_1427180403 1024

controller/messages <13>1 2016-05-06T11:14:41-06:00 c0-0c0s3 bcbwtd 2107 - - do_node_quiesce: setting SSID transmit credits to 0 for node mask 0x1

event 2015-07-14T00:00:01.642382-07:00 s0 35696 2015-07-14 00:00:01|ec_sedc_warning|src::12:s0|pri:0x0| seqnum:0x8d37|svc:::c2-1c0s13 BC_H_BB_VERT_IVOC0_ECB_FAULT: 0x4 [health fault] <expected:0> |

logsystem <189>1 2015-06-12T00:00:05.536978-07:00 c4-1c2s0n2 xtlog 35376 p0-20150527t162759 [llm_status@34] craylog: message type: ccm_epilogue_debug, connected to port: 514

power management <158>1 2015-12-16T00:00:18.776794-08:00 edismw xtpmd 7587 - [hss_pm@34] MESSAGE: - db_hook (pid 12341) stdout: Finished

sm <158>1 2015-11-10T10:14:09.888703-07:00 edismw state_manager 7730 - [hss_sm@34] liveCompUpdate: 168 modified comps, 0 new in 3 slots, 6 ignored, 0 L0s comp<->service.

smw <22>1 2015-06-03T00:00:10.163285-07:00 edismw postfix 17803 - - connect to fe2.nersc.gov[2001:400:310::1b]:25: Network is unreachable

xtdiscover 2015-09-23 11:39:53 Command line: /opt/cray/hss/default/bin/xtdiscover --warmswap c2-2c1s6,c4-3c2s10,c5-1c1s13

p0/alps 2015-03-17T00:00:04.166686-08:00 21971 EVENT[sync][27]: apid 10988935 uid 61105 cmdName 'pw.x' batchID '2548128.edique02' numNids 5 nids [3099-3102,3119]

p0/netwatch 160906 12:08:54 c9-3c0s3g1l55 c9-1c2s3g0l51 1 Link Inactive

p0/nlrd 2016-09-06T11:19:57.160000-06:00 ci-smw 22766 2016-09-06 11:19:56 ci-smw 22768 cb_hw_error: failed_component c1-5c0s7g0, type 21, error_code 0x0d17, error_category 0x0002

p0/console 2016-05-06T07:25:01.286734-06:00 c0-0c2s8n3 Out of memory: Killed 30 processes sharing cpu group with pid 6158.

p0/messages <6>1 2016-05-06T14:18:41.301167-06:00 c0-0c0s1n1 kernel - p0-20160403t123157 - ata6: SATA link down (SStatus 0 SControl 0)

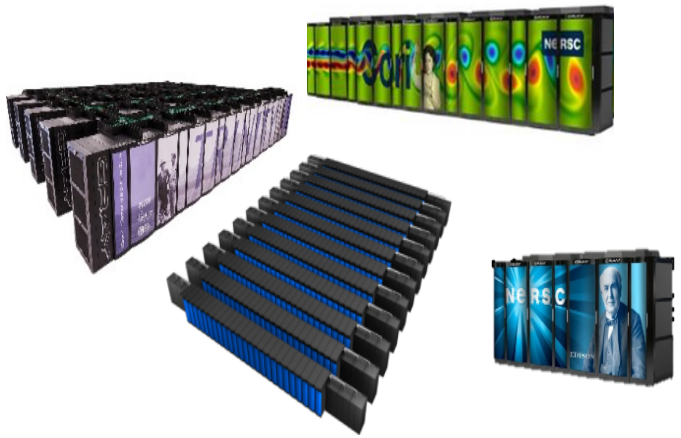
p0/consumer

Fri Sep 2 13:31:51 2016 - rs_event_t at 0x621af0
svcid 0: ::c12-2c0s7n2 = svid_inst=0x0/svid_type=0x0/svid_node=c12-2c0s7n2[rsn_node=0x31dc/rsn_type=0x0/rsn_state=0x6], err code 131276 - Node Heartbeat Fault
ev_data...
00000000: 01 00 00 00 00 00 00 00 00 00 00 00 00 06 dc 31 *.....1*
00000010: 30 20 78 02 00 00 00 00 01 00 00 00 cc 00 02 00 *0 x.....*
00000020: 6e 6f 64 65 20 63 31 32 2d 32 63 30 73 37 6e 32 *node c12-2c0s7n2*
00000030: 20 63 6f 6e 73 69 64 65 72 65 64 20 64 65 61 64 * considered dead*
00000040: 2c 20 6c 61 73 74 20 68 65 61 72 74 62 65 61 74 *, last heartbeat*
00000050: 20 30 30 30 36 33 62 30 31 2c 20 55 6e 61 62 6c * 00063b01, Unabl*

External Interactions, Community Building, Publications, Presentations

- **Problem Statement:**
  - Finding effective solutions to large scale HPC system resilience across the breadth of available hardware and software platforms requires:
    - Involvement of researchers across disciplines
    - Platform diversity to test data processing tools and underlying analytical methods
  - Sharing tools, methodologies, and experiences is often easier than sharing system data

- **Solutions:** ***Encourage collaboration, disseminate information***
  - HMDR hosts and participates in a variety of collaboration and community building activities:
    - Data Analysis Workshop hosted at NCSA: invited participants from other Resilience projects and external collaborators to a multi-day hands-on meeting to share tools and techniques for analyzing HMDR datasets
    - Host HPC Monitoring, Analysis, and Diagnosis BoFs and events at CUG and SC
    - Promoting information sharing and collaborations at our existing HPC community activities
  - HMDR tools and datasets released to collaborators or open source
- **Impacts:**
  - Multiple downloads and requests for collaboration
  - HMDR capabilities for data collection and analysis being applied to investigating instrumentation and metrics for assessing network congestion



Tools and Datasets Access Points

Baler: github.com/ovis-hpc
LDMS: github.com/ovis-hpc
LogDiver: contact for information

- waiting on university approval for Open Source access to community

HMDR log cleansers: contact for information
HMDR dataset access point:
portal.nersc.gov/project/m888/resilience
HMDR community building BoF materials:
sites.google.com/site/monitoringlargescalehpcsystems

HMDR Contact: hmdr@ncsa.illinois.edu

Publications and Presentations

- "Data-Driven Understanding of Fault Scenarios and Impacts Through Fault Injection: Experimental Campaign in Cielo", V. Formicola, S. Jha, F. Deng, D. Chen, A. Bonnie, M. Mason, A. Greiner, A. Gentile, J, Brandt, L. Kaplan, J. Repik, J, Enos, M. Showerman, Z. Kalbarczyk, W. Kramer, and R. Iyer. In *Proc. Cray Users Group (CUG)* 2017.
  - About: Results from fault injection study on ACES Cielo System. Includes elucidation of fault to failure paths; how those are manifested in the log files and in system data; timelines for onset and recovery; and failures of failover paths.
- "Network Performance Counter Monitoring and Analysis on the Cray XC Platform", J. Brandt. E. Froese, A. Gentile, and L. Kaplan, B. Allan, and E. Walsh, In *Proc. Cray User's Group (CUG)* 2016.
  - About: High Speed Network counter collection and analysis from ACES Trinity. Counters indicative of error, performance, and status issues in the Aries network.
- "Design and Implementation of a Scalable Monitoring System for Trinity", A. DeConinck, A. Bonnie, K. Kelly, S. Sanchez, C. Martin, M. Mason, J. Brandt, A. Gentile, and B. Allan, In *Proc. Cray User's Group (CUG)* 2016.
  - About: Architecture to enable scalable data collection and analysis for resilience applied to ACES Trinity system.
- "Design and Implementation of a Scalable HPC Monitoring System", S. Sanchez, A. Bonnie, G. Van Heule, C. Robinson, A. DeConinck, K. Kelly, Q. Snead, and J. Brandt, In *Proc IEEE Int'l. Parallel and Distributed Processing Symposium Workshops (IPDPSW)* 2016.
  - About: Architecture to enable scalable data collection and analysis for resilience.

- "Understanding Gemini Interconnect Failovers on Blue Waters", S. Jha, V. Formicola, C. Di Martino, Z. Kalbarczyk, W. Kramer, and R. Iyer, In *Proc. Cray User's Group* 2016.
  - About: Presents methodology and tools to understand and characterize the recovery mechanisms of the Gemini interconnect system from raw system logs. The tools can assess the impact of these recovery mechanisms on the system and user workloads.
- "Attacking supercomputers through targeted alteration of environmental control: A data driven case study.", C. Keywhan, V. Formicola, Z. Kalbarczyk, R. Iyer, A. Withers, and Adam J. Slagell. In *Communications and Network Security (CNS), 2016 IEEE Conference on*, pp. 406-410. IEEE, 2016.
  - About: Demonstrates that the control systems of chilled water in Blue Waters' facilities, can be used as entry points by an attacker to indirectly compromise the computing functionality.
- "Resiliency of HPC Interconnects: A case study of interconnect failures and recovery in Blue Waters.", S. Jha, V. Formicola, C. Di Martino, M. Dalton, W. Kramer, Z. Kalbarczyk, and R. Iyer. Under review to *IEEE Transactions on Dependable and Secure Computing.*
  - About: Characterizes the recovery procedures of the Blue Waters Gemini interconnect network, the largest Gemini network built by Cray.
- ***Datasets:*** "Mutrino Dataset 2/15-6/16 (12/16 Release)", J. Brandt, A. Gentile, J. Repik, SAND2016-12310 O, 2016 and "Mutrino Dataset 2/15-5/15", J. Brandt, A. Gentile, and J. Repik, SAND2016-2449 O, 2016 [Online]: *at HMDR access portal.*
  - About: Logs from the ACES Trinity testbed system Mutrino, including production, standup, and induced network, electrical, thermal, and functional failures.
- ***Keynote Presentation:*** "Failure and Resiliency in the Shadow of Exascale – Will our our Current Assumptions Take us in the Right Direction?", B. Kramer, Workshop on High Performance Computing Systems Plus Applications (HPCMASPA) @ IEEE IPDPS, 2016.
- ***Presentation:*** "Measuring Resiliency of Extreme-Scale Computing Systems", Z. Kalbarczyk, S. Jha, V. Formicola, C. Di Martino, R. Iyer, B. Kramer, Seminar at Advanced Digital Sciences Center (ADSC), 2016.

Future Work

Continued failure path characterization across all available platform data

- Intel KNL systems using log and numeric data from Cori and Trinity phase 2
  - Implement new samplers for KNL specific performance counters
- Deploy high fidelity implementation of Cray's "System Environment Data Collections" system

Comparison of fault types between Blue Waters (Cray XE/XK), Edison (Cray XC30), and Trinity/Cori (Cray XC40)

- Investigate if/how models have changed, due to differences in technology and instrumentation, where there is overlap

Continued development and tighter integration of Baler and LogDiver

- Baler patterns to LogDiver Regex tagging in Baler database
- LogDiver analysis using Baler database for interactive exploration

Incorporation of numeric data as "binned events" to explore "symptom to failure" path analysis

- Identify behavioral characteristics in numeric data that have strong correlation to fault and failure associated events in log files

Make recommendations for future instrumentation and notifications to the system based on fault-to-failure paths, timings, characterizations (e.g., frequency of occurrence, severity of impact)

Further release of cleansed log and numeric data from large scale machines

- Include more meta-data (e.g., times of replacement parts and software upgrades)
- Include details of what types of materials were cleansed and the process