Enabling Dependability-Driven Resource Use and Message Log-Analysis for Cluster System Diagnosis

Edward Chuah, Arshad Jhumka, Samantha Alt, Theo Damoulas, Nentawe Gurumdimma, Marie-Christine Sawley, William L. Barth, Tommy Minyard, James C. Browne
Outline

- Motivation
- Related research
- Problem specification
- CORRMEXT framework
- Case study: Ranger supercomputer
- Conclusion and future work
Mean-time between failures on large supercomputer systems are decreasing

➢ Ranger supercomputer at TACC:
   ▪ 4,048 nodes, ~64,000 cores.
   ▪ Generates ~10.5 million messages in one month.
   ▪ Reports more than 1,000 O/S hang-up messages in one month.

➢ Challenges in analyzing supercomputer message logs to diagnose failures:
   1. Messages are ambiguous and unstructured.
   2. Messages exhibit wide variations in their structures.
   3. Messages relevant to a failure are interleaved among thousands of other messages.
   4. Messages missing from the logs, e.g., not logged.
Summary of contributions

- A new framework that combines resource use data and message logs for detailed dependability-oriented system diagnosis.

- Shows that the earliest times of change in system behaviour can only be identified by analysing both patterns of resource use and events.

- Shows that all the correlations are statistically significant by applying the Bonferroni correction.

- Applied CORRMEXT on real data (resource use data and message logs) collected on the Ranger supercomputer to:
  - Identify error propagation paths leading to failure,
  - Explain instances of error recovery,
  - Explain causes of failed error recovery mechanisms.
Related research

- Error detection, failure prediction and failure diagnosis using message logs alone:
  - Error detection --> [A. Oliner et. al., 2008], [J. Stearly et. al., 2008], [W. Xu et. al., 2009].
  - Failure prediction --> [Y. Liang et. al., 2007], [E.W. Fulp et. al., 2008], [F. Salfner et. al., 2008], [Z. Lan et. al., 2010], [A. Gainaru et. al., 2012], [I. Fronza et. al., 2013], [A. Pelaez et. al., 2014].
  - Failure diagnosis --> [T. Reidmeister et. al., 2009], [A. Oliner et. al., 2010], [S.P. Kavulya et. al., 2012].

- Anomaly detection, failure prediction and error recovery using resource use data alone:
  - Anomaly detection --> [Z. Lan et. al., 2010], [Q. Guan et. al., 2010].
  - Failure prediction --> [F. Salfner et. al., 2009].
Related research (cont’d)

- Error detection and failure diagnosis using both resource use data and message logs:
  - Error detection --> [N. Gurumdimma et. al., 2015 & 2016].
  - Failure characterization & diagnosis --> [Z. Zheng et. al., 2011 & 2012].

Our work focuses on other dependability-related issues such as error propagation and error recovery.
Related research: 1\textsuperscript{st}, 2\textsuperscript{nd} generation FDiag

- Correlates event messages by time-bins of 24-hours.
- Identified the dates of event sequences, correlated events for one time period.
- Did not identify the nodes, did not identify new causes of system failures.
- Analyses was limited to identifying correlated events for one time period.

Reports where (the nodes), when (dates of event sequences), and why (the events involved) the system failure occurred.

Did not identify new causes of system failures.

Analyses was limited only on the Linux syslog messages.
Related research: 3rd generation FDiag

- Identifies additional causes of failures not found by the previous versions of FDiag and of which the systems administrators were not previously aware.
- Added anomaly (outlier) detection and enabled processing of Rationalized message logs.
- Analyses was limited only on the Linux syslog messages and Rationalized log messages.

**Table: FDiag Features**

<table>
<thead>
<tr>
<th>Ver.</th>
<th>Outlier Extraction</th>
<th>Correlation (Pearson)</th>
<th>Event Sequence Construction</th>
<th>Standard Linux syslogs</th>
<th>Rationalized message logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

11th EDCC 2015

FDiag available online

http://diag-toolkits.github.io/FDiag/
Related research: ANCOR (ANomaly CORrelator)

- Identifies causes of system failures that are due to resource utilization anomalies based on combining resource use data by job with rationalized message logs.
- Evaluated multiple anomaly extraction algorithms and integrated anomaly analysis and correlation analysis, and assessed the impact of resource utilization anomalies on system failures.
- Correlations were limited to errors that follow a linear function; uses only Pearson correlation.
Related research: CRUMEL (Correlating Resource Use Data and MEssage Logs)

- Performs failure diagnosis by identifying patterns of events and resource use.
- The workflow shows that more events correlated with failures can only be identified by applying different correlation algorithms.
- Demonstrates that correlations of Lustre I/O activity can be used to monitor Lustre errors which eventually lead to a system failure.

The analyses produced by the workflow alleviates the challenge of missing messages in the logs.

- Analysis of patterns of events and resource usage is limited to errors which are correlated only to system failures.
Analysing only the log messages for cluster system diagnosis has low coverage ...

- Two categories of information: **Constants, Variables**
  - **Constants** – words only.
  - **Variables** – alpha-numerical words.

• Extracting Constants alleviates the challenge of variations in message structure.

**Jan 1 01:21:23** i121-308 kernel: [1427074.562395] **LustreError: 167-0:** This client was evicted by scratch-OST0100; in progress operations using this service will fail.

**Messages missing from the logs, i.e., not logged.**
Resource use data collected by TACC_Stats provide the necessary resolution …


- TACC_Stats is a job-oriented and logically structured version of the conventional Sysstat system performance monitor.

- In addition to information collected by Sysstat, TACC_Stats records a comprehensive range of metrics that spans all system resources including hardware performance monitoring data, Lustre file-system operation counts, Infiniband & Omni-Path usage, energy consumption, vectorisation, I/O and network activity.

- Collection of resource use data requires no cooperation from the job owner, and requires minimal overhead (~1.1%) and automatically collects its data for every job executed on TACC systems.
Message logs that CORRMEXT currently processes in its workflow

Syslog message
Jan 1 01:21:20 oss30 kernel: LustreError: 0:0
(ldlm_lockd.c:305:waiting_locks_callback()) ##
lock callback timer expired after 51s: evicting client at *.*.*.*@o2ib ns:
filter-scratch-OST0100_UUID lock:
00000102347cbb80/0x344dcb188d1dcc4a lrc: 3/0,0 mode: PW/PW res:
30607838/0 rrc: 2 type: EXT [0->18446744073709551615] (req 0->4095)
flags: 0x20 remote: 0x5275d77aa11d3abd expref: 4 pid: 7941 timeout
4438041253

Rationalized message

time:1293552419
host:i101-403
jobid:1743653
prog:kernel
0:<3>BUG: soft lockup detected on CPU#%d, pid:%d, uid:%u, comm:%s
1:14
2:5656
3:0
4:ldlm_bl_04

The Rationalized message logs alleviates the challenge of message ambiguity & structure.

Dependability-Oriented System Diagnosis

- State-of-the-art techniques on failure diagnosis correlate a given error event with system failure, i.e., the technique returns the most likely cause of the failure.

- This information may not provide enough details to the system administrator.

- For example, information such as "An inode failure led to communication errors, which were not recovered, and which led to failure" provides more information than "system failure is due to communication errors".

- The recovery can now target inode failures rather than communication errors as we shall see in the subsequent slides.
CORRMEXT (CORrelating Resource use data and Message logs and EXtracting Time-bins) framework

PROBLEM STATEMENT

Given (i) a set of resource use data, (ii) a set of message logs, (iii) a list of resource use counters, (iv) a list of message types, and (v) a date, then:

• Identify groups of resource use counters that are *linearly* or *monotonically* correlated on the specified date,

• Identify groups of errors that are *linearly* or *monotonically* correlated on the specified date,

• Identify the time-bin(s) that are associated with the correlated resource use counter groups and correlated error groups on the date specified.
The process workflow of CORRMEXT is composed of three modules:

- Data Type Extraction.
- Correlation.
- Time-bin Extraction.

The output of each module are sets of reports which can be used for diagnosis.
CORRMEXT: Data Type Extraction module

- Extracts resource use counters from TACC_Stats resource use data.
- Currently process up-to 410 resource use counters.
- The data matrix $DR_{time-bins}$ contains histograms of the resource use counters by hourly time-bins.
- Steps for extracting the resource use counters and generating the data matrix:

1. Split the resource usage logs into individual hours by the given date.
2. For each log entry, extract the resource use counter and store it in a list.
3. Identify the unique resource use counter name in the list and obtain the resource use counter type.
4. For each resource use counter type in a resource use log which match the given list of resource use counter names, if the values associated with the resource use counter types of two consecutive log entries are different, obtain the difference and add the difference to the value obtained in the preceding operation and store the value.
CORRMEXT: Data Type Extraction module

- Extracts message types from the message logs.
- English-only words called *Constants* are extracted, and repetitions are then removed.
- The data matrix $DM_{\text{time-bins}}$ contains histograms of the message types by hourly time-bins.
- Steps for extracting the message types and generating the data matrix:

1. Split the message logs into logs of individual hours by the given date.
2. For each message log, extract the message *constant* part and store it in a list.
3. Identify the unique message *constants* in the list and obtain the message types.
4. Given a list of message types, count the number of message types by hour for the given date in the logs.
CORRMEXT: Correlation module

1. X-axis represents one production hour.
2. Y-axis represents the occurrence count of a message type and a resource use counter.
3. Automatically finds a correlation coefficient threshold which is then used to extract a list of strongly correlated events and strongly correlated resource use counters.
CORRMEXT: Correlation module (cont’d)

- Apply Fisher’s Z-score to test the significance of all correlation coefficients.
- We define null hypothesis: $H_{0r}$: that a pair of resource use counters are very weakly positive correlated. $H_{0e}$: that a pair of events are very weakly positive correlated.
- We define alternate hypothesis: $H_{ar}$: that a pair of resource use counters are strongly positive correlated. $H_{ae}$: that a pair of events are strongly positive correlated.

**Bonferroni correction:**
- The probability of observing at least one significant result just due to chance is:
  \[
  1 - (1 - P)^d
  \]
  where $P$ is the $p$-value to be tested, $d$ is the number of hypotheses to test.
- To account for inflation in false positive as a result of applying multiple independent tests, we apply a standard technique called *Bonferroni correction*.
- Apply the Bonferroni correction on the unadjusted $p$-value to obtain the adjusted $p$-values.
- The adjusted $p$-values are obtained by multiplying the unadjusted $p$-values by $d$. 

**Diagram Description:**

- The diagram illustrates the correlation module process, starting with the collection of data from resource use counters and message logs.
- The data is then extracted and unique resource use counters and message types are identified.
- Correlation analysis is performed on both resource use counters and events, with validation steps included.
- The adjusted $p$-values are obtained by applying the Bonferroni correction to the unadjusted $p$-values.
Corrmext: Time-bin Extraction

- Identifies the times of change in system behaviour during the day.
- Computes the variance of both the correlated resource use counters and correlated events at every hour.
- Identifies the earliest times in the correlated resource use counters and correlated events using the following process:
  - Store the variance for each time-bin in a list \( L_{\text{var}} \).
  - Obtain the difference in the variance between two consecutive time-bins and store the difference in a list \( L_{\text{vardiff}} \).
  - Sort \( L_{\text{vardiff}} \) in descending order with the first element, the largest different in variance between two time-bins.
Case study: Ranger supercomputer

- A Linux-based cluster comprising of 4,048 nodes.
- Each node is equipped with its own Linux operating system kernel.
- All the nodes are linked together via a high-speed Infiniband network.
- Job scheduling and resource management are provided by the Sun Grid Engine.
- The Lustre file-system provides high-speed file storage and access on Ranger.
- UDP is used as the main transport protocol on all the nodes.
Data collection on Ranger supercomputer

- Collected three months of resource use data and rationalized message logs.

<table>
<thead>
<tr>
<th>Month</th>
<th>Resource use data</th>
<th>Rationalized message logs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Size</td>
<td>Qty. logs</td>
</tr>
<tr>
<td>June 2011</td>
<td>120.9 GB</td>
<td>603,024,456</td>
</tr>
<tr>
<td>July 2011</td>
<td>124.1 GB</td>
<td>637,860,203</td>
</tr>
<tr>
<td>August 2011</td>
<td>125.4 GB</td>
<td>633,396,685</td>
</tr>
</tbody>
</table>

- 26 dates when soft lockup failures were reported in the rationalized message logs on June, July and August 2011.

<table>
<thead>
<tr>
<th>Month</th>
<th>Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 2011</td>
<td>3, 5, 14, 15, 16, 21, 22</td>
</tr>
<tr>
<td>July 2011</td>
<td>5, 6, 7, 11, 18, 19, 23, 24, 25, 26, 27, 31</td>
</tr>
<tr>
<td>August 2011</td>
<td>3, 4, 11, 22, 24, 30, 31</td>
</tr>
</tbody>
</table>
Dependability use cases on Ranger

- Identified three dependability use cases.

<table>
<thead>
<tr>
<th>Issue</th>
<th>System</th>
<th>Error</th>
<th>No. of dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recovery attempt</td>
<td>Application &amp; process memory allocation</td>
<td>Memory allocation &amp; memory leaks</td>
<td>25</td>
</tr>
<tr>
<td>Error propagation</td>
<td>Lustre file-system I/O &amp; Infiniband</td>
<td>Communication &amp; file-system I/O errors</td>
<td>24</td>
</tr>
<tr>
<td>Recovery attempt</td>
<td>Chipset &amp; system memory</td>
<td>Chipset &amp; memory errors</td>
<td>26</td>
</tr>
</tbody>
</table>
Capturing Recovery Attempt and its Impact: Application & process memory allocation

**Phase 1: Correlated NUMA and Process counters.**

Correlations of NUMA miss, NUMA foreign, ps processes & ps ctxt can be used to infer error recovery.
Capturing Recovery Attempt and its Impact: Application & process memory allocation

Phase 2: Correlated Segmentation faults and General Protection Errors.

- segfault and general protection error events are strongly positive correlated to soft lockup events on July 06 and July 23, 2011.
- segfault and general protection error events are weakly positive correlated to soft lockup events on eight dates.

On eight of ten dates, the O/S removed the faulty application, representing a recovery rate of 80%.
On two of ten dates, the O/S did not catch the general protection faults, represents a failure rate of 20%.
Capturing Recovery Attempt and its Impact: Application & process memory allocation

**Phase 3: Earliest times of change.**

Both correlated NUMA & process resource use counters and correlated segfaults & general protection errors are required to identify the earliest times of change in system behaviour.
Capturing Recovery Attempt and its Impact: Application & process memory allocation

**Validation: Test of statistical significance**

<table>
<thead>
<tr>
<th>Correlated groups</th>
<th>June 2011</th>
<th>July 2011</th>
<th>August 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUMA &amp; Process resource use counters ( (n = 24) )</td>
<td>( z_r = 10.68 )</td>
<td>( 6.15 \leq z_r \leq 10.68 )</td>
<td>( z_r = 10.68 )</td>
</tr>
<tr>
<td>Segmentation faults &amp; general protection errors ( (15 \leq n \leq 24) )</td>
<td>( z_e = 10.68 )</td>
<td>( 9.08 \leq z_e \leq 10.68 )</td>
<td>( 9.08 \leq z_e \leq 10.68 )</td>
</tr>
</tbody>
</table>

Under the null hypothesis, \( z_{or} = 2.64 \) and \( z_{oe} = 2.64 \). Hence, we reject the null hypothesis in favour of the alternate hypothesis.

**Probability of rejecting the null hypothesis when it is true**

- We use the significance level alpha = 0.01.
- Lowest z-score is 6.15. \( p \)-value is equal to the probability of observing a value greater than 6.15 in the standard normal distribution, or 0.00001 for a one-sided test.
- Adjusted \( p \)-value is 0.00025. The adjusted \( p \)-value is less than 0.01, indicating it is highly unlikely this result would be observed under the null hypothesis.
Error propagation: Lustre file-system I/O & Infiniband

**Phase 1: Correlated Infiniband and Lustre I/O counters.**

Strong positive correlations between Infiniband and Lustre file-system I/O resource use counters.
Error propagation: Lustre file-system I/O & Infiniband

Phase 2: Correlated communication and file-system errors.

<table>
<thead>
<tr>
<th>Error event</th>
<th>Failure event</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>error occurred while communicating with failure inode</td>
<td>soft lockup</td>
<td>June 21</td>
</tr>
<tr>
<td>failure inode</td>
<td>soft lockup</td>
<td>June 21</td>
</tr>
<tr>
<td>error occurred while communicating with failure inode</td>
<td>soft lockup</td>
<td>July 23</td>
</tr>
<tr>
<td>failure inode</td>
<td>soft lockup</td>
<td>July 23</td>
</tr>
</tbody>
</table>

On nine of eleven dates, the file-system recovered from inode failures, rep. a recovery rate of 81%.

On two of eleven dates, the file-system failed to recover from inode failures, rep. a failure rate of 18%.
Error propagation: Lustre file-system I/O & Infiniband

Phase 3: Identify earliest times of change.

Both correlated Infiniband & Lustre I/O counters and correlated communication & file-system errors are required to identify the earliest times of change in system behaviour.
Error propagation: Lustre file-system I/O & Infiniband

Validation: Test of statistical significance

<table>
<thead>
<tr>
<th>Correlated groups</th>
<th>June 2011</th>
<th>July 2011</th>
<th>August 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infiniband &amp; Lustre I/O resource use counters</td>
<td>$3.71 \leq z_r \leq$</td>
<td>$3.58 \leq z_r \leq$</td>
<td>$6.15 \leq z_r \leq$</td>
</tr>
<tr>
<td>($n = 24$)</td>
<td>10.68</td>
<td>10.68</td>
<td>10.68</td>
</tr>
<tr>
<td>Communication &amp; file-system errors</td>
<td>$3.71 \leq z_e \leq$</td>
<td>$5.29 \leq z_e \leq$</td>
<td>$z_e = 10.68$</td>
</tr>
<tr>
<td>($n = 24$)</td>
<td>10.68</td>
<td>10.68</td>
<td></td>
</tr>
</tbody>
</table>

Under the null hypothesis, $z_{or} = 2.64$ and $z_{oe} = 2.64$. Hence, we reject the null hypothesis in favour of the alternate hypothesis.

Probability of rejecting the null hypothesis when it is true

- We use the significance level alpha = 0.01.
- Lowest z-score is 3.58. $p$-value is equal to the probability of observing a value greater than 3.58 in the standard normal distribution, or 0.000172 for a one-sided test.
- Adjusted $p$-value is 0.0041. The $p$-value is less than 0.01, indicating it is highly unlikely this result would be observed under the null hypothesis.
Capturing Recovery Attempt and its Impact: Chipset and Memory system

Phase 1: Correlated CPU and Memory counters.

Strong positive correlations between CPU and Memory resource usage activities.
Phase 2: Correlated chipset and ECC errors.

- Correlations of northbridge error with ECC error and correlations of northbridge error with ECC error occurred on all dates when CPU and memory activities are correlated.
- northbridge error, ECC error and core events are weakly positive correlated to soft lockups on all 26 dates.

Represents a recovery rate of 100%.
Capturing Recovery Attempt and its Impact: Chipset and Memory system

**Phase 3: Identify earliest times of change.**

Both correlated CPU & memory resource usage and correlated chipset & memory errors are required to identify the earliest times of change in system behaviour.
Capturing Recovery Attempt and its Impact: Chipset and Memory system

Validation: Test of statistical significance

<table>
<thead>
<tr>
<th>Correlated groups</th>
<th>June 2011</th>
<th>July 2011</th>
<th>August 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU &amp; memory use counters</td>
<td>$3.74 \leq z_r \leq 5.86$</td>
<td>$3.74 \leq z_r \leq 8.16$</td>
<td>$4.17 \leq z_r \leq 6.18$</td>
</tr>
<tr>
<td>($n = 24$)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chipset &amp; ECC errors</td>
<td>$z_e = 10.68$</td>
<td>$z_e = 10.68$</td>
<td>$z_e = 10.68$</td>
</tr>
<tr>
<td>($23 \leq n \leq 24$)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Under the null hypothesis, $z_{or} = 2.64$ and $z_{oe} = 2.64$. Hence, we reject the null hypothesis in favour of the alternate hypothesis.

**Probability of rejecting the null hypothesis when it is true**

- We use the significance level alpha = 0.01.
- Lowest z-score is 3.74. $p$-value is equal to the probability of observing a value greater than 3.74 in the standard normal distribution, or 0.00008 for a one-sided test.
- Adjusted $p$-value is 0.002. The $p$-value is less than 0.01, indicating it is highly unlikely this result would be observed under the null hypothesis.
Summary

• Demonstrated that a tool chain comprising of standard techniques can identify patterns of resource usage and events associated with error recovery and error propagation.

• Showed that CORRMEXT can yield accurate dependability-oriented diagnosis of errors in a large cluster system.

• Showed that both the resource usage data and message logs are required to identify the earliest times of change in system behaviour.
Future work

- Apply CORRMEXT to analyse patterns of error recovery and error propagation other than memory, communication and file-system I/O errors.

- Test and evaluate CORRMEXT on the HPC clusters (Stampede, etc.) at TACC.
To rank the pair of samples, we applied a standard technique called *tied rank average* method.

**PROCESS:**

- Rank order the values in the dataset $x$ with the smallest value getting a rank of 1.
- If more than one value has the same rank in dataset $x$, assign the average rank to these values.
- Rank order the values in the dataset $y$ with the smallest value getting a rank of 1.
- If more than one value has the same rank in the dataset $y$, assign the average rank to these values.

**Example:**

<table>
<thead>
<tr>
<th>Count</th>
<th>Rank</th>
<th>Count</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>7.5</td>
<td>26</td>
<td>6.5</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>50</td>
<td>8</td>
</tr>
<tr>
<td>33</td>
<td>5</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>7.5</td>
<td>23</td>
<td>5</td>
</tr>
<tr>
<td>35</td>
<td>6</td>
<td>26</td>
<td>6.5</td>
</tr>
<tr>
<td>24</td>
<td>3</td>
<td>11</td>
<td>3</td>
</tr>
</tbody>
</table>
Selected Publications


