Pin-pointing Node Failures in HPC Systems
Anwesha Das, Frank Mueller (North Carolina State University)
Paul Hargrove, Eric Roman (Lawrence Berkeley National Lab)

Motivation


Goal - less effective for node resilience, Correlation extraction is hard.

Challenges ? Detecting faults independently without pin-pointing node failures is less effective for node resilience. Correlation extraction is hard. Goal - Can automated Machine Learning Techniques help us? What features are required to extract node failures? Study logs to extract required patterns.

Solution Approach

- Node Status and Events
- Torque (Job) & System Data Correlation

What information is needed from Cray logs?

- Bootinfo
- Node Health
- Console
- HostInfo

Node Status and Error

- Job Id X
- Node Id (c1-2c2x1n1)
- ResID, appID, pageID
- Job X State

Allocate nodes for X (exit, re-run)

Correlate Torque logs with Cray system logs

- Torque Logs
- Alps Logs
- Job Id
- Hatch Id

Interesting Phrases (helpful)?

Run Topics over Time Algorithm (frequency based estimation)

Integrate separate documents into a single document term matrix

Extract those patterns.

What caused it? (Internal or external events?)

Was it normal shutdown or abnormal failure?

Which node exactly failed?

Contributions

Identification of patterns for indicating node failures distinguishing from mass service shutdown for maintenance based on size and time.

Leveraged TOT - Topics Over Time (continuous time based LDA - Latent Dirichlet Allocation algorithm) to estimate dynamic change in log messages.

Derivation of ways to correlate Torque (Job) logs and Cray system logs to pin-point node failures.

Table 1: Some Typical Node Failures

<table>
<thead>
<tr>
<th>NodeId</th>
<th>Node-Id-Type</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>c0-c0</td>
<td>Service</td>
<td>Node BIOS communication error</td>
</tr>
<tr>
<td>c0-c0</td>
<td>Service</td>
<td>Node BIOS communication error</td>
</tr>
<tr>
<td>c0-c0</td>
<td>Service</td>
<td>NMU Fault</td>
</tr>
<tr>
<td>e2-c0</td>
<td>Service</td>
<td>Disk Queue Fatal Error</td>
</tr>
<tr>
<td>c0-c0</td>
<td>Service</td>
<td>Lustre Error</td>
</tr>
<tr>
<td>e6-c0</td>
<td>Service</td>
<td>LNET Runtime Error</td>
</tr>
</tbody>
</table>

Insights and Findings

- Normal node shutdowns very frequent for maintenance (multiple chassis going down in groups of 4) & periodic reborts, compute node failures relatively infrequent compared to service nodes (Job failures & error logs)
  - Service nodes – 12 times a month
  - Compute Nodes – 3 per week

- Some key phrases of interest for node failures:
  - Failing node c1-0c8x1n2, node_unavailable, node status down, Errors, Fatal, exit codes, allocated nodes for Jobs, etc.

- Leverage Job Id & state coupled with node Id & state to correlate Job logs and Cray system logs for pin-pointing node failures.

(2013-04-26T00:41.948135-05:00, AER_BAD_TLP, 0.064),
(2013-04-26T00:41.948135-05:00, ac_hw_error, 0.56)

TOT provides the dynamic phrase distribution over continuous time-series data.

Results

Log Data Details

- Edison, Hopper, Cori based Cray logs – Approx 1698961 files, more than 600 GB data.
- Factorie toolkit, scikit-learn python packages for various libraries.
- LogDiver Tool for high-level data analysis.

Acknowledgments:

Dr. Frank Mueller, Dr. Paul Hargrove, Dr. Eric Roman for insightful guidance. Nick Wright, Tina Butler, James Botts and Annette Greimere from NERSC for helping with data and valuable information. This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.