Failure Prediction with Cray Log

Ziming Zheng\textsuperscript{a}, Zhiling Lan\textsuperscript{a}, Byung H. Park\textsuperscript{b}, and Al Geist\textsuperscript{b}

\textsuperscript{a} Illinois Institute of Technology, \{zzheng11, lan\} @iit.edu
\textsuperscript{b} Oak Ridge National Laboratory, \{parkbh,gst\}@ornl.gov

Cray Log Analysis Contest at WASL’08
An Overview of Cray Log

- There are 6 top-level files and 8 directories
- The **EVENTLOGS** folder accumulate occurrences of all event types that are strictly ordered by the timestamps

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>CRMS Event Type</th>
<th>SRC</th>
<th>SVC</th>
<th>Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007-08-01</td>
<td>ec_mesh_link_failed</td>
<td>src::c2-2c0s4</td>
<td>svc::c2-2c0s4s0</td>
<td>c2-2c0s4s0l5=S</td>
</tr>
</tbody>
</table>

- The **SEDC_FILES** folder contains environmental data, e.g. temperature of L0 controller, etc.
  - However, the logging time is not matched with **EVENTLOGS**

- Our analysis is solely based on the **EVENTLOGS**
Our Effort – Failure Prediction

Data Preprocessor
- Categorizer
- Filter

Clean Log

Raw Logs

Predictor without Location Info
- Decision Tree
- Assoc. Rules
- Effective Rule Set
- Predictor
- Result

A failure will occur in the near future (e.g., 5 min-1 hr)!

Predictor with Location Info
- Decision Tree
- Assoc. Rules
- Effective Rule Set
- Predictor
- Result

A failure will occur in c0-0c0s0n3 and c3-0c0s5n0 in the near future!
Log Preprocessing

- Event categorizing
  - Five event types are identified as interested failures:
    1. node heartbeat fault (NHF)
    2. node failed fault (NFF)
    3. seastar heartbeat fault (SHF)
    4. VERTY health check fault (VHC)
    5. L0 voltage fault (L0V)

- Event filtering
  - Temporal and spatial filtering is used to remove the redundant events
  - The clean log keeps --- event start and end time, event count and event location

- Totally, there are 18 failures in the log
Prediction W/O Location Info

- First, we have discovered the following rules by using decision tree and association rule methods
  1. uPacket squash fault occurred more than 506 times → A failure will occur
  2. Lustre PTL timeout fault occurred → A failure will occur
  3. Segmentation Fault occurred more than 6 times → A failure will occur

- uPacket squash fault, Lustre PTL timeout fault and Segmentation Fault are always reported from different locations from that of the failure
  - These rules can only forecast that a failure will occur in the near future, without pinpointing the location

- Result: 70% of failures are predicted with 14% of false alarms
We have also identified the following rule set

- “no more processes left in this runlevel” in c0-0c0s1n3 → A failure will occur in c0-0c0s1n3
- `ec_console_log` occurred in c1-0c1s2n1 more than 200 times → A failure will occur in c1-0c1s2n1

- Out of 18 failures, 8 do not have any precursor events from the same location

- Result: 49% of failures are predicted with 30% of false alarms
Discussion

- Prediction accuracy could be improved via meta-learning
  - It is improper to expect a single method to capture various failure patterns alone!
  - Both DT and AR are limited by the proportion of fatal events without any precursor warning
  - Suggest the use of meta-learning to combine different methods [ICPP07 & ICPP08]
    - An alternative method like probability distribution can be combined with DT or AR to boost prediction
    - However, the limited size of the RAS log prevents us from performing this alternative method
Prediction could be improved by including other data sources (i.e., in addition to RAS events)
  • The environmental data might be helpful to identify failure location
  • Further, our previous study shows that environmental/performance data could be used to pinpoint failure location [Cluster’07]
    • Pattern recognition techniques like PCA and ICA
We have also analyzed the Cray XT system *jaguar* at ORNL

- 45G data are collected from 2007-05-05 09:32:55 to 2007-11-27 03:14:14
- Result: 87% of failures are predicted with 13% of false alarms (without location info)
Reference