A detailed analysis of fault prediction results and impact for HPC systems.

Ana Gainaru, Franck Cappello, Bill Kramer
Motivation

- Log files give useful information
  - Systems generate events about hardware, application, user actions
- Classic data mining workflow:
  1) Group events of same type in clusters
  2) Filter redundant events (in space and time)
  3) Correlation analysis (explore time or/and space dependencies)
  4) Event or failure prediction.
- Best result so far: Good precision (80%) and recall (70%)
  - From failure logs (not event logs) + Resource Usage log
  - Very long training phase (predicting 1 month from 9 months of training)
- Observation
  - Different error types present different distributions
  - Analyze behavior differences in all events
Signal analysis

- Occurrences of each event types are considered as time series
  - Different event types become different signals
- Variation in signal's normal behavior identifies suspicious events that could represent failures
- Easy to shape and characterize different behaviors

Data mining

- Optimized on finding patterns
  - Correlation extraction
  - Fast online outlier detection
Silent signal characteristic of error events. PBS errors

Noise signal typically Warning messages: Memory errors corrected by ECC

Periodic signals daemons, monitoring

The sixth workshop of the Joint Laboratory
Table of contents

- Signals analysis
  - Pre-process
  - Analysis methodology
- Results
  - Correlation
  - Location analysis
  - Prediction
  - First step: Prediction’s impact on checkpointing
• Looking for frequently occurring messages with similar syntactic patterns
  • HELO: Hierarchical Event Log Organizer
• Signal extraction
  • Use a sampling rate (different depending on the signal)
  • Map number of events for each sample

<table>
<thead>
<tr>
<th>Template</th>
<th>Event type</th>
</tr>
</thead>
<tbody>
<tr>
<td>failed to configure resourcemgmt subsystem err = 10</td>
<td>Processor cache error</td>
</tr>
<tr>
<td>psu failure</td>
<td>PSU failure</td>
</tr>
<tr>
<td>component state change: component * is in the * state *</td>
<td>State change in a component</td>
</tr>
</tbody>
</table>

Table II

EXAMPLES OF TEMPLATES AND THEIR EVENT TYPES
1) Outlier detection

- Causal moving data window
  - Each observed data point is compared to the median
  - Based on the distance between the points we detect outliers

- Thresholds
  - 2 months time window
  - Different distance values automatically selected

- Advantages
  - Decrease the influence of severe outliers on signals
2) Signal correlation

- Data mining algorithm
  - Filter out the normal behavior
  - Apply a algorithm based on the GRITE data mining algorithm
  - Tree-based exploration – multi-set correlations

3) Location correlation

- Set of locations that events propagate on
Experiments on BlueGene/L
  - Offers information about event severity
  - Analyze location prediction
  - Additional experiments on Mercury

Correlations
  - Patterns, breakdown on different components

Prediction
  - Precision, recall, time between prediction and occurrence
Examples of correlations

- Memory errors
  - Starting: *correctable error detected in directory*
  - Ending: *parity error in read queue*
  - Time delay: around one minute
- Node card errors
- Multi-line messages
- Component restart sequences

- 23% cannot be used for prediction
  - Refer only to informational messages
  - Eliminate all events with INFO severity
**Dissection**

- Distribution of the events that compose a sequence

- Time delay between correlations

  **Pairs of correlations**
  - 33.6% have at least 10 seconds
  - 7.8% have at least 1 minute and
  - 2.1% over 10 minutes

  **Complete sequences**
  - 54.1% have at least 10 seconds
  - 19.4% have at least 1 minute and
  - 5.7% over 10 minutes
Propagation behavior

- Around 20-25% events propagate
- Cumulative distribution of the number of different locations per chain

Example: NFS (network file system) on Mercury

- Propagation break down
  - Initial pair of correlated events
  - 76.92% show no propagation
  - 2.16% expend outside the midplane
• Prediction process overview
  • Visible prediction window
Prediction results

• Analysis window: avg 2 seconds, worst case 8.43
• Signal analysis - many sequences, small length
  • Higher analysis window
• Data mining approach
  • Loses correlations between signals of different types
• Location prediction results
  • 93% to 86% precision
  • 43% to 40% recall
• Force best precision
Appearances of different error types

- reported to all errors in the system
- dark: correctly predicted cases out of total occurrences

**Best case:** nodecard
- Over 80% predicted

**Worst case:**
- Around 10%
Other observations

- **Sequence use for prediction**
  - 23.5% of sequences are used in over 90% of the cases
  - 2.5% of sequences are never used for prediction

- **Visible prediction window**
  - 85% of the prediction offer more than 10 seconds
  - 36.6% of the total failures on BlueGene/L are seen with more than 10 seconds in advance
Impact on checkpointing

Analytical model

• Start with the formula from describing the waste for a checkpointing strategy from [1]
  • Accounts for checkpointing time, recovery time and the lost due to faults
  • Include a prediction with a precision and recall
    • Changes the mttf, adds the waste of taking checkpoints when predicting an error, adds the waste when the prediction is wrong

• For $C = R = 5$, $D = 1$ minutes

<table>
<thead>
<tr>
<th>Mttf</th>
<th>Precision</th>
<th>Recall</th>
<th>Waste</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 day</td>
<td>92</td>
<td>36</td>
<td>10.4%</td>
</tr>
<tr>
<td>5 h</td>
<td>90</td>
<td>50</td>
<td>22%</td>
</tr>
</tbody>
</table>

• **Signal analysis**
  - Different event types may have different normal behaviors
  - Faults affect event types in a different way

• **Data mining**
  - Correlations
  - Location prediction

• **Apply the model for fault prediction**
  - Precision of 90% and recall of 40%
  - Prediction over 10 seconds before event
Future work

- Noise in each analysis step
  - Optimize the steps that influence the results

- Breakdown on components
  - Shows uneven distribution on precision
  - Better understanding for certain error types

- Fault distribution after prediction
  - Better analytical model

- Combine the prediction module with a checkpointing strategy
A detailed analysis of fault prediction results and impact for HPC systems.

Ana Gainaru (againaru@illinois.edu)