Predicting Computer System Failures Using Support Vector Machines

Errin W. Fulp\textsuperscript{a}  \hspace{1cm}  Glenn A. Fink\textsuperscript{b}  \hspace{1cm}  Jereme N. Haack\textsuperscript{b}

\textsuperscript{a}Wake Forest University  \hspace{1cm}  \textsuperscript{b}Pacific Northwest National Laboratory

Department of Computer Science  \hspace{1cm}  Richland WA, USA

Winston-Salem NC, USA

USENIX Workshop on the Analysis of System Logs

December 7, 2008

High-Performance Computing Trends

- Expected that computing will continue to double each year
  - \textit{Petaflop systems listed on top500.org}
  - However CPU clock rates will see limited increases
- Computing improvements achieved with more processors
  - IBM Blue Gene at LLNL has 212,992 processors
  - System failures will become more problematic
System Events

- There are several critical system events
  - Hardware failure, software failure, and user error
  - Frequency will increase as systems become larger (cluster)
  - Resulting in lower overall system utilization
- *Cannot easily improve failure rates, can we manage failure?*
  - Smarter scheduling of applications and services
  - Minimize the impact of failure
- Accurate event predictions are key for event management
  - *Are predictions possible? How accurate?*
  - Need system status information to make predictions

System Status Information

- *Almost* every computer maintains a system log file
  - Provide information about system events
  - syslog is actually general-purpose logging facility [Lon01]
- An event represents a change in *system state*
  - Include hardware failures, software failures, and security

<table>
<thead>
<tr>
<th>Host</th>
<th>Facility</th>
<th>Level</th>
<th>Tag</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.129.8.6</td>
<td>kern</td>
<td>alert</td>
<td>1</td>
<td>1171062692</td>
<td>kernel raid5: Disk failure on sde1, disabling device</td>
</tr>
</tbody>
</table>

- Entries contain information such as: time, message, and tag
  - Time identifies when the message was recorded
  - Message describes the event, typically natural language
  - Tag represents criticality, low values are more important
Log Files

<table>
<thead>
<tr>
<th>Host</th>
<th>Facility</th>
<th>Level</th>
<th>Tag</th>
<th>Time</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>198.129.8.6</td>
<td>local7</td>
<td>notice</td>
<td>189</td>
<td>1171061732</td>
<td>sysstat</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>kern</td>
<td>info</td>
<td>6</td>
<td>1171061732</td>
<td>kernel md: using maximum available idle CPU bandwidth</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>cron</td>
<td>info</td>
<td>78</td>
<td>1171061733</td>
<td>crond 2500 (root) CMD (/usr/lib/sa/sal 1 l)</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>auth</td>
<td>info</td>
<td>38</td>
<td>1171062445</td>
<td>rsh(pam_unix) 2215 session opened for user by (uid=0)</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>auth</td>
<td>info</td>
<td>38</td>
<td>1171062445</td>
<td>in.rshd 2216 <a href="mailto:root@hpcs2.cs.edu">root@hpcs2.cs.edu</a> as root: cmd=/root/temps</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>daemon</td>
<td>info</td>
<td>30</td>
<td>1171062590</td>
<td>smartd 88 Device: /dev/twe0 SMART Prefailure Attribute</td>
</tr>
<tr>
<td>198.129.8.18</td>
<td>syslog</td>
<td>info</td>
<td>46</td>
<td>1171062590</td>
<td>syslogd restart.</td>
</tr>
<tr>
<td>198.129.7.282</td>
<td>daemon</td>
<td>info</td>
<td>30</td>
<td>1171062590</td>
<td>ntpd 2555 synchronized to 198.129.149.218, str</td>
</tr>
<tr>
<td>198.129.7.222</td>
<td>daemon</td>
<td>info</td>
<td>30</td>
<td>1171062590</td>
<td>ntpd 2555 synchronized to 198.129.149.218, str</td>
</tr>
<tr>
<td>198.129.7.238</td>
<td>daemon</td>
<td>info</td>
<td>30</td>
<td>1171062590</td>
<td>ntpd 2555 synchronized to 198.129.149.218, str</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>auth</td>
<td>notice</td>
<td>37</td>
<td>1171062590</td>
<td>sshd(pam_unix) 12430 auth failure; logname=el-fork-o</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>kern</td>
<td>info</td>
<td>6</td>
<td>1171062590</td>
<td>kernel md: using 512k, over a total of 12287936 blocks.</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>cron</td>
<td>info</td>
<td>78</td>
<td>1171062601</td>
<td>crond 2500 (root) CMD (/usr/lib/sa/fork-it 1 l)</td>
</tr>
<tr>
<td>198.129.8.6</td>
<td>kern</td>
<td>alert</td>
<td>1</td>
<td>1171062692</td>
<td>kernel raid6: Disk failure on sde1, disabling device</td>
</tr>
</tbody>
</table>

- Log file is a list of messages, can be analyzed for
  - Auditing, determine the cause of an event (*past*)
  - Predicting important events (*future*)

Example System Event to Predict

- An interesting event is *disk failure*
  - By 2018 [large systems] could have 300 concurrent reconstructions at any time [SG07]
  - Predicting disk failure is important
    - *Easy to identify event in the log...*
- Predict failure as *early as possible*
  - \( n \) messages \( M = \{ m_1, m_1, ..., m_n \} \)
  - Assume \( m_n \) is the event
  - Min depth \( d \) and max lead \( l \)
- *Are all messages the same?*
SMART

- Self-Monitoring Analysis & Reporting Technology (SMART)
  - SMART disks monitor their health and performance
  - Attributes describe current state, each attribute has unique ID
- Many different types of messages (Attribute and Value)

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Raw_Read_Error_Rate changed to ( x )</td>
</tr>
<tr>
<td>190</td>
<td>Airflow_Temperature changed to ( x )</td>
</tr>
<tr>
<td>2</td>
<td>Throughput_Performance</td>
</tr>
<tr>
<td>8</td>
<td>Seek_Time_Performance</td>
</tr>
<tr>
<td>201</td>
<td>Soft_Read_Error_Rate changed to ( x )</td>
</tr>
</tbody>
</table>

- Pinheiro et al. investigated Google hard drive failure [PWB07]
  - Some SMART parameters do correlate with drive failure
  - Conclude SMART messages alone may not be sufficient

Disk Failure Prediction

- What features (information) should be considered?
  - A message contains criticality, message, and time
  - *Is there a series of messages that tend to be a precursor?*
- Consider a sequence of messages arriving (ordered by time)
  - *Is it possible to classify into failure and non-failure classes?*
  - Other approaches have considered Bayesian Nets and HMM
Support Vector Machines

- Support Vector Machine (SVM) is a classification algorithm
  - Consider a set of samples from two different classes
  - Each vector consists of features describing the sample
  - SVM finds a hyperplane separating the classes in hyperspace
  - The vectors closest to the plane are the support vectors
- Great for aggregate statistics, what about series?
  - Interested in using sequences of messages as features

Spectrum Kernel

- A spectrum kernel considers \( k \) length sequences as features
  - The frequency of the sequence is the feature value
- Assume two symbols \( \{A, B\} \) and sequence length \( k = 2 \)
  - There are \( 2^k \) possible sequences (features) \( (AA, AB, BA, BB) \)
  - Value of a feature is the number of occurrences
  \[
  M = \{A, A, B, A, A, B, B, A\}
  \]
  - \( AA: 2 \)
  - \( AB: 2 \)
  - \( BA: 2 \)
  - \( BB: 1 \)
  - There are \( b^k \) possible sequences, were \( b \) is number of symbols
- How does this work for syslog messages?
tag Sequences

- Each message has a tag that indicates criticality
  - Sequence of messages represented by sequence of tag values

- Need to reduce number of symbols, assume three levels
  - high (tag < 10), medium (10 < tag < 140), low (tag > 140)

- Given a series of messages \( M \), process using a sliding window
  - Count the number of occurrences of \( k \)-length sequences

Example tag Processing

- Let \( M = \{148, 148, 158, 40, 158, 188, 188, 88, 158, 188\} \)

- Assume \( b = 3 \) and \( k = 5 \), then \( 3^5 = 243 \) possible features

<table>
<thead>
<tr>
<th>tag</th>
<th>Encoding (e)</th>
<th>Sequence</th>
<th>( f ) (base 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>148</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>148</td>
<td>2</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>158</td>
<td>2</td>
<td>222</td>
<td>222</td>
</tr>
<tr>
<td>40</td>
<td>1</td>
<td>2221</td>
<td>2221</td>
</tr>
<tr>
<td>158</td>
<td>2</td>
<td>22212</td>
<td>239</td>
</tr>
<tr>
<td>188</td>
<td>2</td>
<td>22122</td>
<td>233</td>
</tr>
<tr>
<td>188</td>
<td>2</td>
<td>21222</td>
<td>215</td>
</tr>
<tr>
<td>88</td>
<td>1</td>
<td>12221</td>
<td>160</td>
</tr>
<tr>
<td>158</td>
<td>2</td>
<td>22212</td>
<td>239</td>
</tr>
<tr>
<td>188</td>
<td>2</td>
<td>22122</td>
<td>215</td>
</tr>
</tbody>
</table>

- Feature number is \( f_{t+1} = \text{mod} (b \cdot f_t, b^k) + e \)

- Vector for \( M \) would be (160:1, 215:2, 233:1, 239:2)
System Data used for Experiments

- About 24 months of syslog files from 1024 node Linux cluster
  - Averaged 3.24 messages an hour (78 a day) per machine
  - Observed 120 disk failure events

Prediction Experiments

- Sets of $M = 1200$ messages (15 days) collected per machine
  - From first message, processed $d =\{400, 600, 800, 1000, 1100\}$
- One SVM considered aggregate features occurring within $d$
  - Number of occurrences for each tag value
- Another SVM also considered tag sequences occurring within $d$
  - Sequences consisted of 5 messages, there were 19 tag ranges
Prediction Results

- Accuracy, precision, recall, and ROC recorded per experiment
  - Where $acc = \frac{TP + TN}{P + N}$, $prec = \frac{TP}{TP + FP}$, and $recall = \frac{TP}{P}$

- More messages improved prediction results
- Combined were better, 73% accuracy with 200 message lead

- ROC curve can be used to compare classifiers/predictions [Faw06]
  - Closer to the *north-west*, the better the performance
  - Some issues with false negatives
- Combined features performed better, typically 4% to 5% increase
Feature Weights

- Use of a linear kernel for the SVM allows for feature analysis
  - Larger weight (positive or negative) indicates a feature useful

- Of 2,476,289 features, only 2,251 were useful
  - Of the useful features 22 were aggregate, remaining were sequences

Runtime Performance

- For the combined feature experiments
  - Training time averaged 7 minutes 38 seconds
  - Testing time averaged 0.21 seconds
Conclusions and Future Work

- Using syslog data to predict disk failures
  - Spectrum-kernel SVM predicted with 73% 100 msg lead
  - Message sequences did improve performance
- Several areas for improvement
  - determine $k$ and $b$, add new features, ...
  - *How does message rate impact performance?*
  - Need more and different data
- Consider other *interesting* events
  - Other failures, since disk failure $\neq$ node failure
  - *Can this be useful for security?*
  - Multi-system analysis
- Possible to create a *reduced message system?* [YM05]

References

### Other Prediction Stats

<table>
<thead>
<tr>
<th>Metric</th>
<th>M = 400</th>
<th>600</th>
<th>800</th>
<th>1000</th>
<th>1100</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Accuracy</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Agg</td>
<td>64</td>
<td>65</td>
<td>65</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>Comb</td>
<td>67</td>
<td>69</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Agg</td>
<td>64</td>
<td>66</td>
<td>67</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>Comb</td>
<td>67</td>
<td>69</td>
<td>72</td>
<td>73</td>
</tr>
<tr>
<td><strong>Recall</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Agg</td>
<td>62</td>
<td>63</td>
<td>63</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>Comb</td>
<td>63</td>
<td>66</td>
<td>68</td>
<td>69</td>
</tr>
<tr>
<td><strong>F-score</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Agg</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>67</td>
</tr>
<tr>
<td></td>
<td>Comb</td>
<td>66</td>
<td>68</td>
<td>71</td>
<td>71</td>
</tr>
</tbody>
</table>