Automated Monitoring of Large-scale Computing Systems
A Light-weight and Unsupervised Method for Near Real-time Behavioral Analysis Using Operational Data Measurement

Motivation
Operational data of computing systems are correlated. Changes in the value of this data and the balance between them is a reflection of the system’s behavior. Detection of deviation in system behavior in early stages would prevent major failures which otherwise could cause additional damages. This work proposes a light-weight and fully unsupervised method for near real-time behavioral analysis using energy consumption and temperature measurements of large computing systems.

Proposed method
- Collection and streaming of temperature and energy consumption data using MetricQ²
- Preprocessing, normalization and down sampling of collected operational data
- Progressive training of a purposely designed light-weight LSTM Autoencoder
- Continues update of the trained model to reflect current systems behavioral pattern
- Automatic adjustment of threshold value based on most current normal behavior
- Identifying divergence from normal behavior

Main characteristics
- Minimal training data and iterations (4 hours and 50 epochs)
- Accurate prediction of systems behavior (approx. 96%)
- Near real-time behavioral analysis (< 2 min)
- System users behavioral pattern awareness
- Adaptive threshold calculation
- Identification of failures type/category
- Further optimization of hyper parameters
- Definitive decision based on the identified anomalies

Future work
- Adaptive threshold calculation
- Identification of failures type/category
- Further optimization of hyper parameters
- Definitive decision based on the identified anomalies

Data Preparation

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Avg. power</th>
<th>CPU0 power</th>
<th>CPU1 power</th>
<th>CPU0 temp.</th>
<th>CPU1 temp.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022-07-13 00:00:07</td>
<td>131.826</td>
<td>33.27999</td>
<td>26.91096</td>
<td>58.0</td>
<td>57.0</td>
</tr>
<tr>
<td>2022-07-13 00:00:17</td>
<td>114.7102</td>
<td>24.99287</td>
<td>20.10629</td>
<td>60.0</td>
<td>57.0</td>
</tr>
<tr>
<td>2022-07-13 00:00:27</td>
<td>136.3787</td>
<td>36.36461</td>
<td>27.7025</td>
<td>60.0</td>
<td>57.0</td>
</tr>
<tr>
<td>2022-07-13 00:00:37</td>
<td>129.5102</td>
<td>31.33605</td>
<td>25.80199</td>
<td>60.0</td>
<td>57.0</td>
</tr>
<tr>
<td>2022-07-13 00:00:47</td>
<td>118.3239</td>
<td>26.30085</td>
<td>21.37567</td>
<td>60.0</td>
<td>57.0</td>
</tr>
</tbody>
</table>

Pattern identification [Autoencoder]

Calculating threshold

Predictions & Unexpected behaviours

1. https://doc.zih.tu-dresden.de/jobs_and_resources/hardware_overview/
2. https://github.com/metricq/metricq