ML Analysis of Workflow Data

Cong Wang

Research Scientist
RENCI, UNC Chapel Hill

Feb 25, 2021
Machine learning (ML) Methods for Performance Data

Panorama 360 framework
- State-of-the-art testbeds
- Production HPC environments

Collect data with Pegasus WMS
- Data stored in internal database
- Open source, flexible API

Online data collection and analysis
- Workflow-level
- Task-level
- Infrastructure-level
Workflow- and Task-Level Analysis: Anomaly Detection

- Multivariate techniques, particularly Machine Learning (ML) algorithms provide the appropriate theoretical foundation.
  - Use workflow-level performance analysis to characterize overall behavior of running workflow by clustering statistically similar workflows.
  - Task-level analysis is triggered to detect faults and bottlenecks using task-level metrics.
  - This talk shows workflow level analysis
Leverage existing Pegasus monitoring API to collect workflow-level metrics.
  - https://pegasus.isi.edu/documentation/rest-api-monitoring.php
  - Exposes a REST API that provides data about workflows running on the system.
    - Eg. curl --insecure --request GET --user adamant:<passwd>
      https://localhost:5000/api/v1/user/adamant/root/14/workflow/1/job/6/job-instance?pretty-print=true
  - A way to get data from the underlying Stampede database
    - Stampede schema: https://pegasus.isi.edu/documentation/images/stampede_schema_overview-small.png
    - Workflow → Job → Job instance → Exit code
    - Workflow → Job → Job instance → Local duration
Classifier Setup

• Workflow-level features
  • Feature vector collected for workflow: \((J_s, J_f, t_s, t_f, o_J_s)\)
    \[
    \frac{\#\text{job\_instances\_succeeded}}{\#\text{job\_instances\_done}} \quad \frac{\#\text{job\_instances\_failed}}{\#\text{job\_instances\_done}} \\
    \frac{\text{Sum(\text{local\_duration(successful\_job\_instances)})}}{\#\text{job\_instances\_succeeded}} \quad \frac{\text{Sum(\text{local\_duration(failed\_job\_instances)})}}{\#\text{job\_instances\_failed}} \\
    \frac{\#\text{job\_instances\_succeeded}}{\#\text{total\_workflow\_jobs}}
    \]

• Collected ~170 workflow runs

• K-means classifier
  • Unsupervised clustering algorithm to partition the input feature vectors into k clusters
Machine learning (ML) Methods for Performance Data

• Ran 1000Genome Pegasus workflows on dedicated slice of resources on ExoGENI testbed
• Cluster consisted of 5 VMs:
  • 1 master, 3 workers, 1 data node
  • Each node: 4 vCPU, 10GB RAM
• Various synthetic anomalies
  • Failure injection with misconfigurations
  • Stress on CPU, RAM, I/O and HDD
• Use clean runs as training, anomalous runs as testing data
Performance Analysis

Clustering results with samples including **Clean runs and runs with anomaly injection with “stress”**;
Using three key features to constitute the feature vector: $J_s$, $t_s$, $t_f$

Finding optimal number of clusters for the data set

- **Elbow at 3 clusters**

Workflow clustering (clean AND “stress” samples) with $J_s$, $t_s$ and $t_f$

- Different levels of “stress” create different clusters;
- Only depends on $t_s$

k-means Clustering with optimal number of clusters;
- $x$, $y$, $z$ axes represent value ranges for scaled features
Conclusion

- Workflow anomaly detection using Pegasus monitoring and data collection capabilities
  - Workflow level anomaly detection
  - Sub-workflow (task) level
- Light weight machine learning techniques
  - K-means
- Promising results
  - >0.7 for Normalized Mutual Information (NMI)
  - >0.7 for Completeness score
Our Pegasus Feedback

- Pegasus interface to make custom REST API calls (e.g., call resource provisioning services)
- Fine-grained monitoring capabilities
Thank You!