PANORAMA 360: Performance Data Capture and Analysis for End-to-End Scientific Workflows

George Papadimitriou[1], Rafael Ferreira da Silva[1], Karan Vahi[1], Rajiv Mayani[1], Mats Rynge[1], Ewa Deelman[1], Anirban Mandal[2], Cong Wang[2], Mariam Kiran[3], Vickie Lynch[4], Jeffrey Vetter[4]

[1]University of Southern California – Information Sciences Institute [2]Renaissance Computing Institute [3]Lawrence Berkeley National Laboratory [4]Oak Ridge National Laboratory

PANORAMA 360

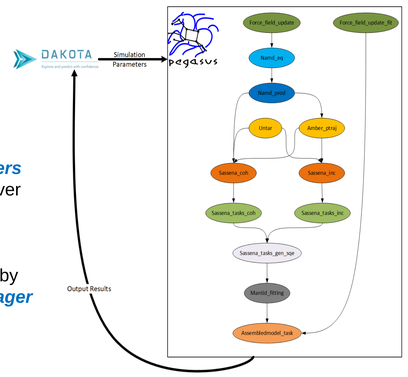
Overview of the Project Description

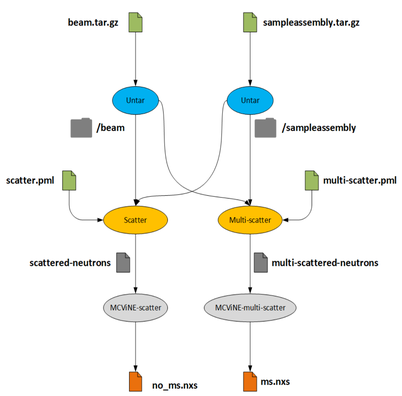
The Panorama 360 aims to further the understanding of the behavior of scientific workflows as they are executing in heterogeneous environments. Panorama 360 collects and correlates workflow performance data into a comprehensive view, that can characterize the **end-to-end** workflow performance on today's systems and drive the design of the future generation systems.
The Panorama 360 architecture collects data from these individual data sources: the ***Pegasus WMS***, the ***Globus Online*** service, the *TCP STatistics and Analysis Tool (**Tstat**)* and the ***DARSHAN*** HPC I/O Characterization Tool.
Our approach for correlating the real time application and infrastructure monitoring data can be used to verify application behavior, perform anomaly detection and diagnosis, and support adaptivity during workflow execution, in an **online** manner. This can lead to improved performance and stability of scientific workflows and benefit the DOE-relevant applications.
Ultimately, by having all these data and analysis tools in our disposal, we envision the creation of a workflow performance data repository and a collection of tools that will be publicly available and can drive science forward.

IMPACT ON DOE SCIENCE

Nanodiamond and MCViNE workflows

- We have extended the SNS-Nanodiamond workflow with (a) parameter sweep, and (b) feedback loop capability, using ***Dakota*** (https://dakota.sandia.gov).
- Dakota based on an internal model generates the ***Simulation Parameters*** and triggers a ***Pegasus*** analysis driver that initiates a new workflow.
- Dakota can instantiate multiple simulation workflows in parallel and by using the ***Pegasus Ensemble Manager*** we can have more control over their execution.





MCViNE (http://www.mcvine.org) is a Monte Carlo neutron ray-tracing program for computer modeling and simulations that mirror real neutron scattering experiments.

MCViNE is used to study data from many SNS instruments, and some of them are ARCS and SEQUOIA.

During the past summer we collaborated closely with SNS scientists and explored ways to support MCViNE workflows with Pegasus.

LEARN MORE

Website https://panorama360.github.io

GitHub Repository https://github.com/Panorama360

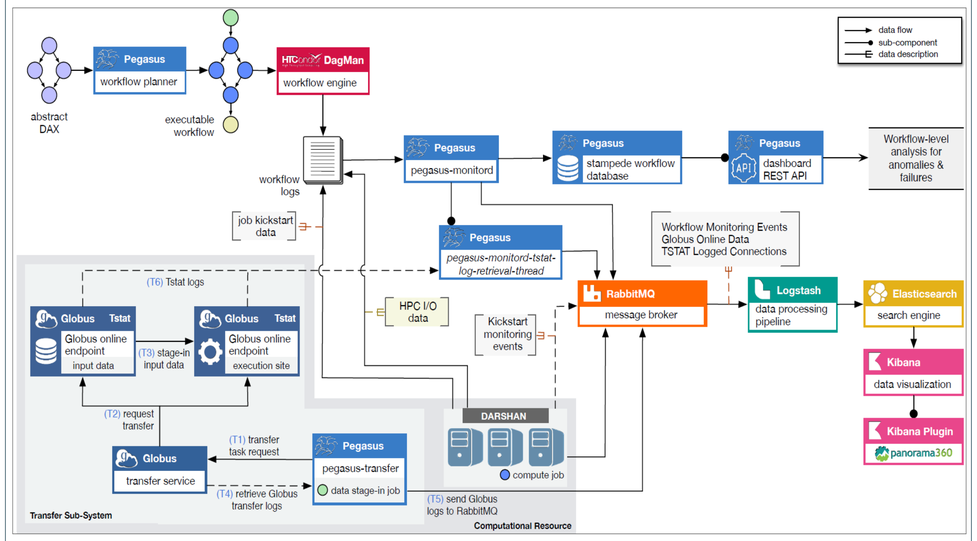
Panorama 360 is funded by the US Department of Energy under Grant #DE-SC0012636M

PERFORMANCE DATA CAPTURE

Overview of the Data Capture Architecture

Panorama 360's data capture architecture can be divided into **5 entities:**

- The workflow management system (Pegasus)
- The data sources (Pegasus, Globus, Tstat, Darshan)
- The search (ElasticSearch) and visualization (Kibana) engines
- The message broker (RabbitMQ)
- The data processing pipeline (Logstash)

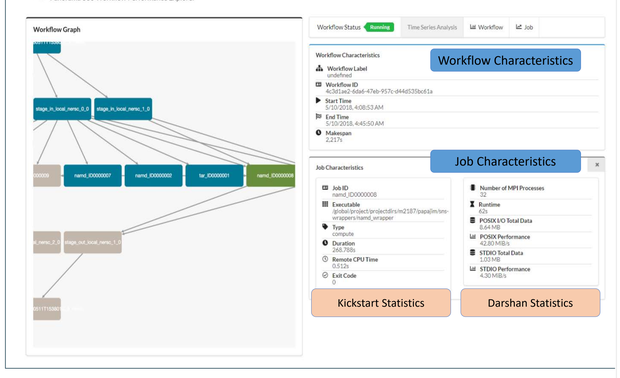


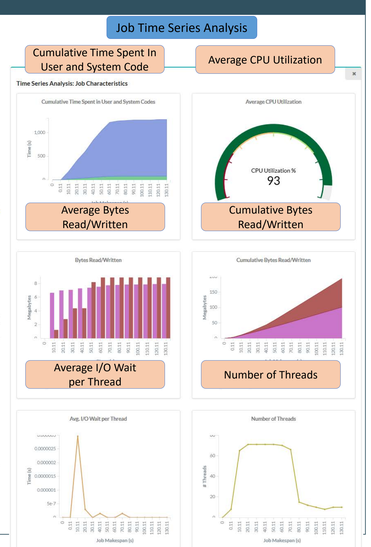
- Pegasus, apart from executing the workflow, is the entity that orchestrates the data movement between the data sources and the analysis framework.
- Pegasus-monitord reports events about the status of the workflow, and extracts Darshan reports from the job's stdout.
- Pegasus-kickstart wraps the execution of jobs and provides aggregated execution statistics (duration, I/O, memory usage), after job completion. Additionally it has an **online** monitoring feature that allows us to collect refined traces with an interval, as low as 1 second.
- Darshan provides POSIX and MPI I/O file access statistics. It tracks the application's file access patterns and the performance of the underlying file system.
- Globus reports back overall transfer statistics and information about each individual transfer (throughput, files transferred, errors etc.)
- Tstat captures low level TCP statistics and apart from revealing network issues, it can help us understand the reason behind an underperforming network connection.

DATA VISUALIZATION

Panorama 360 Kibana Plugin

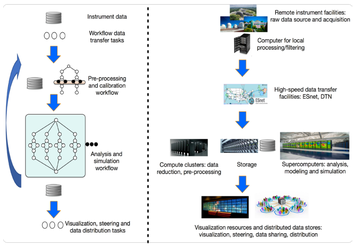
In order to visualize the collected data in a *meaningful* and *compact* way, we have created a **Kibana Visualization Plugin** that correlates the information provided by pegasus-monitord with the monitoring data and compiles a custom dashboard with **per workflow** and **per job** level information, that is updated automatically during the workflow execution.



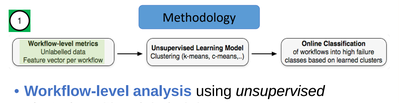


PERFORMANCE ANOMALY DETECTION

Workflow Performance

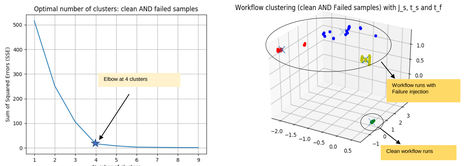


Complexity of end-to-end workflow enactment brings increased potential for performance problems that need to be detected and mitigated



- **Workflow-level analysis** using *unsupervised clustering* with unlabeled data
  - Cluster similar workflow runs and associate them with behavioral classes using k-means clustering with randomized initial centers and Euclidean distance metric
- Detection approach: Clustering algorithm will produce a set of classes. This will be used during online classification of an incoming feature vector
  - Compute degree of membership for each cluster for new workflow instances
- Feature vector collected for each workflow: (J_s, J_f, t_s, t_f, o_i_s)
  - J_s = #job_instances_succeeded/#job_instances_done
    J_f = #job_instances_failed/#job_instances_done
    t_s = Sum(local_duration(successful_job_instances))/#job_instances_succeeded
    t_f = Sum(local_duration(failed_job_instances))/#job_instances_failed
    o_i_s = #job_instances_succeeded/#total_workflow_jobs

Montage Workflow: Clustering results with samples including **Clean runs and runs with Failure injection**;
Using three key features to constitute the feature vector: J_s, t_s, t_f



Evaluation (clean AND Failed Samples)

Manual labeling of samples with **4 clusters**
(i) clean samples, (ii) samples with low failure injection, (iii) samples with medium failure injection, (iv) samples with high failure injection

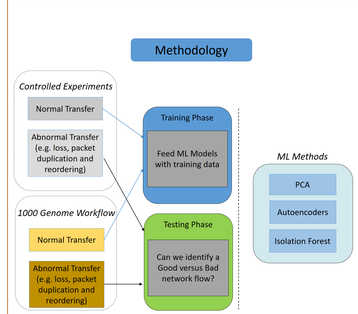
Qn: How does the optimal clustering compare to the ground truth derived from manual labeling of the samples ?

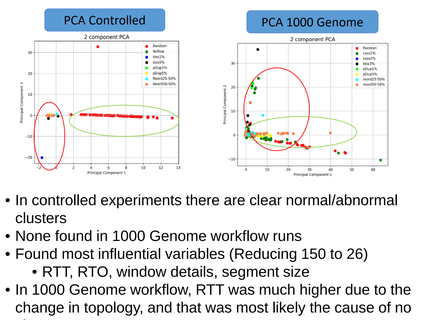
Adjusted Rand Score = 0.957839506631 [-1,1]
Adjusted Mutual Info Score = 0.887212796429 [0,1]
Normalized Mutual Info Score = 0.891753607867 [0,1]
Homogeneity Score = 0.893408890238 [0,1]
Completeness Score = 0.890101392356 [0,1]
Fowlkes Mallows Score = 0.979488868432 [0,1]
Calinski Harabaz Score = 1657.56790411

* Scores closer to 1 are better; For CH Score, higher is better

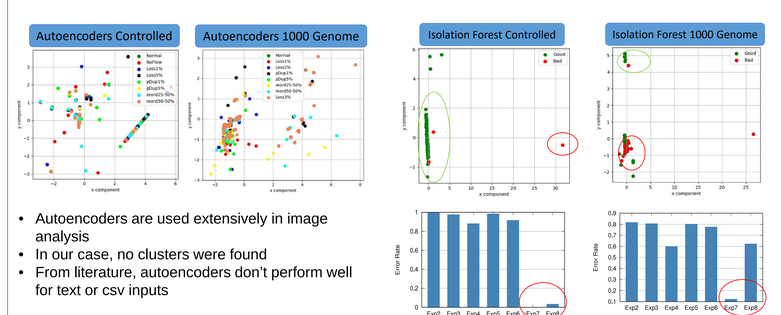
PERFORMANCE ANOMALY DETECTION

Infrastructure - Network Performance





- In controlled experiments there are clear normal/abnormal clusters
- None found in 1000 Genome workflow runs
- Found most influential variables (Reducing 150 to 26)
  - RTT, RTO, window details, segment size
- In 1000 Genome workflow, RTT was much higher due to the change in topology, and that was most likely the cause of no clusters



- Autoencoders are used extensively in image analysis
- In our case, no clusters were found
- From literature, autoencoders don't perform well for text or csv inputs

Table I: Scenario 1: Controlled Experiments

No.	Experiment type (runs for 1 hour)	Total Sample
1	Random Traffic (iperf sends traffic at random)	152
2	No flow	120
3	1% Loss	120
4	5% Loss	120
5	1% Packet Duplication	60
6	5% Packet Duplication	60
7	25% - 50% Packet reordering	50
8	50% - 50% Packet reordering	60

Table II: Scenario 2: 1000 Genome Workflow Experiments

No.	Experiment type	Total Sample
1	Workflow runs	1475
2	1% Loss	1563
3	2% Loss	1636
4	5% Loss	1948
5	1% Packet Duplication	1574
6	5% Packet Duplication	1527
7	25% - 50% Packet reordering	1491
8	50% - 50% Packet reordering	1502

The tables refer to the error rate tests of the Isolation Forest classifier

- Isolation Forest identified good and bad flows more successfully than the other classifiers
- But it has high error rates !!!
- In the case of the workflow, the error rate for Exp8 improves significantly