



ENABLING DATA ANALYTICS WORKFLOWS USING NODE-LOCAL STORAGE

Tu Mai Anh Do¹, Ming Jiang², Brian Gallagher², Albert Chu², Cyrus Harrison², Karan Vahi¹, Ewa Deelman¹

¹University of Southern California – Information Sciences Institute ²Lawrence Livermore National Laboratory

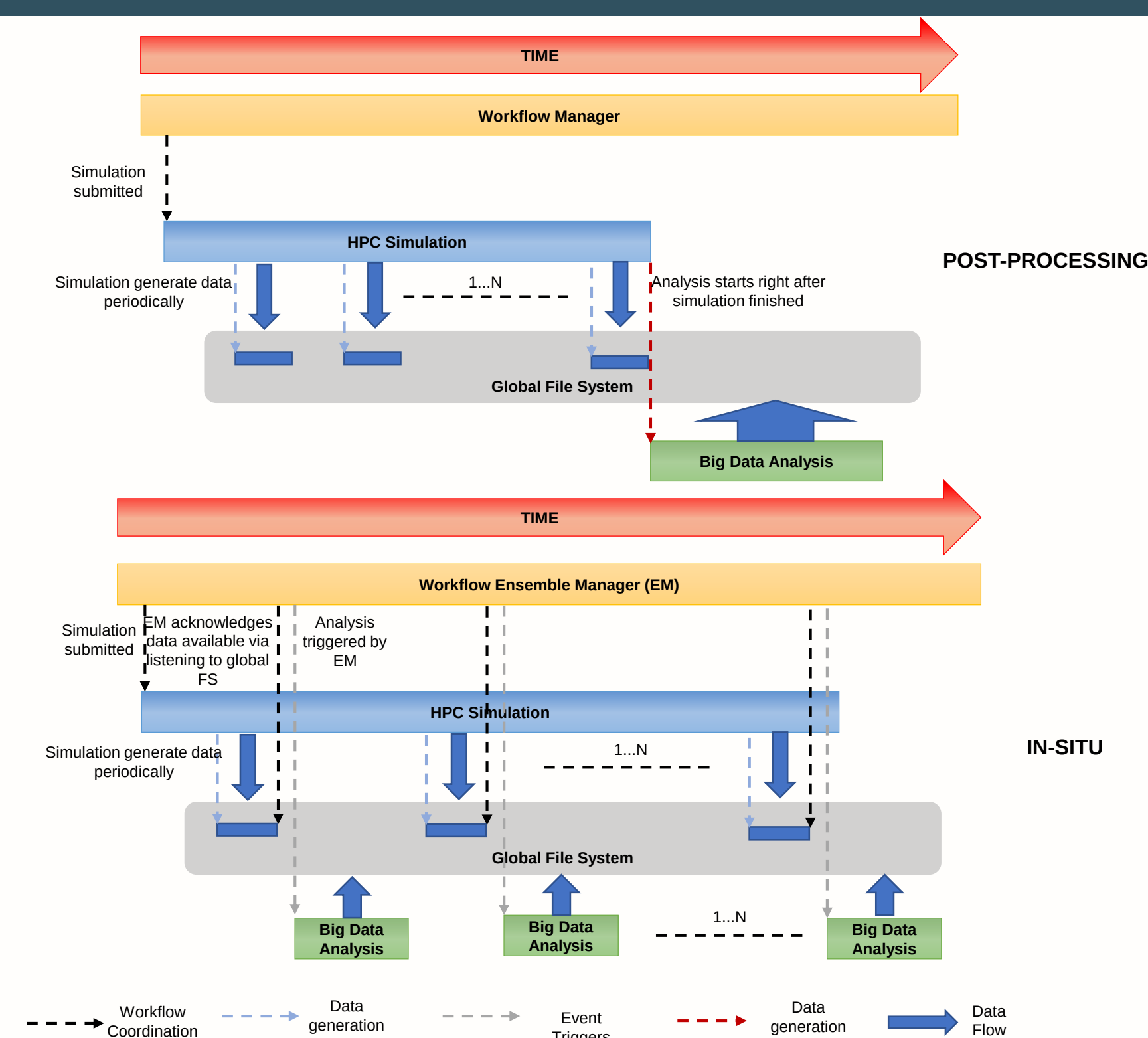
INTRODUCTION

HPC + Big Data

- Advances in **High Performance Computing** enable us to run simulations on ever increasing larger scale, while the use of **Big Data** frameworks allow for large datasets to be analyzed in an effective fashion.
- The **mismatch in data representation** between traditional HPC based POSIX processing, and HDFS-based access employed by frameworks such as Spark makes the coupling between the two frameworks difficult
- Simulation outputs need to be written out to a file system, and then ingested by Spark for processing, which **increases the time to run the complete end to end workflow**
- We propose a **node-local approach** that leverages **non-volatile random-access memory (NVRAM)** to enable data analytics workflows

POST-PROCESSING AND IN-SITU

Coupling between HPC simulation and Big Data Analyses



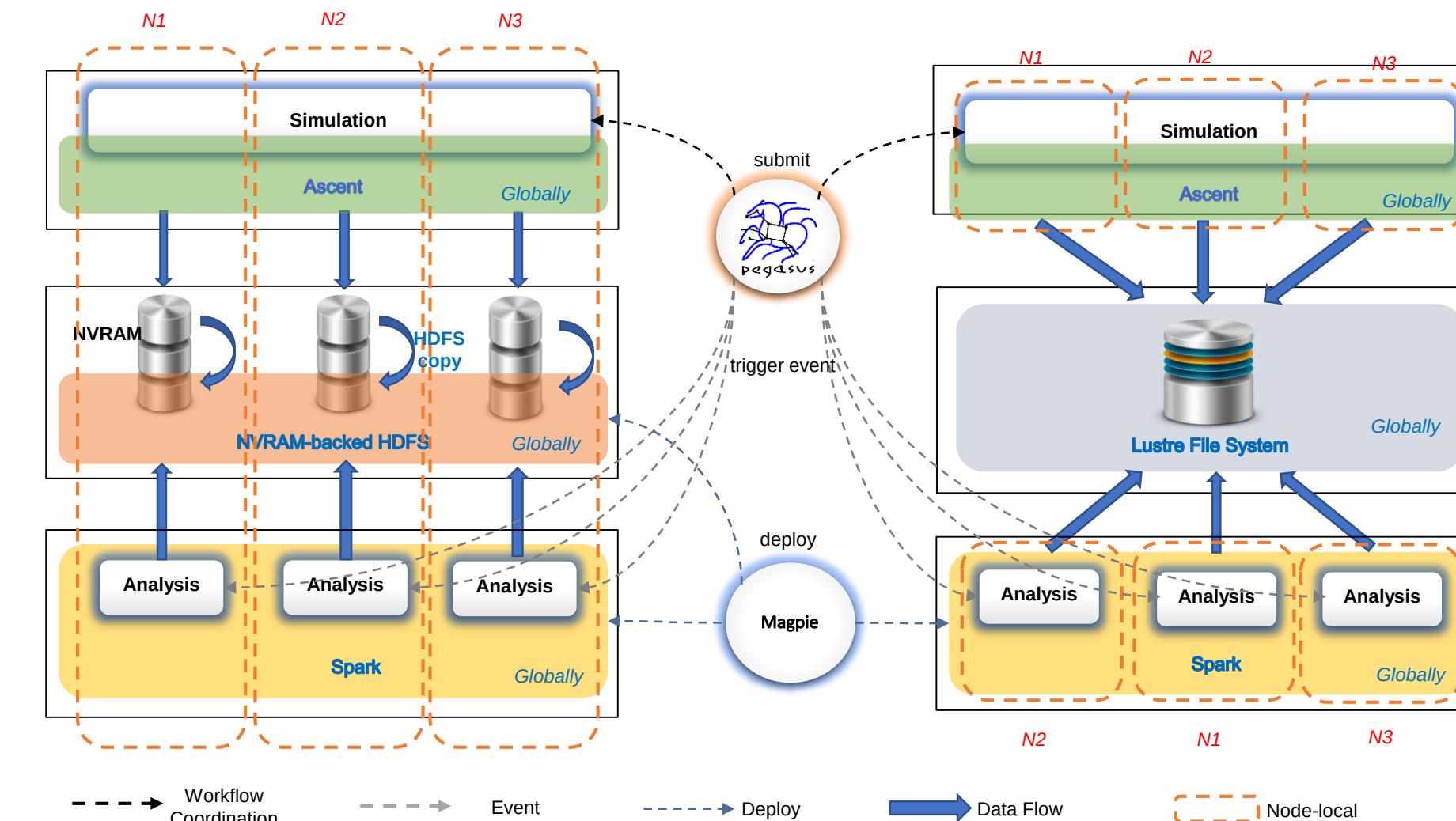
- Post-processing** is a traditional type of coupling that the analysis only starts when the simulation finishes generating entire data. The workflow manager submits the jobs in **sequential order**
- To reduce the expensive storage needs, the analytics might not be required to delay starting the execution until all iterations finish. We plan to analyze simulation data as generated in a few time-steps, called **in-situ**, using **event-based triggers** for launching workflows for future work.

IN COLLABORATION WITH



NODE LOCAL APPROACH

Data placement on global file system



- We proposed a **node-local approach** that leverages **NVRAM** as intermediate storages of partial data generated from the simulation to enable data analytics workflows
- Instead of writing out to a global file system, data is **written locally to local NVRAM** by the simulation, and then immediately ingest data into **NVRAM-backed Hadoop Distribute File System (NHDFS)** using in built hooks provided by the **Ascent** framework. NHDFS is built over node-local NVRAMs on the compute nodes.
- Magpie** (<https://github.com/LLNL/magpie>) facilitates the deployment of a multiple-node cluster with Hadoop and Spark
- Three main advantages:
 - Reduces the latency** once data is generated to node-local NVRAM
 - The analysis can be performed locally to **avoid off-node data transfer**
 - NHDFS **resolves the data representation mismatch** between the HPC simulations and the Big Data analytics.

ASCENT

An in-situ framework on HPC platforms

- We use **Ascent** (<https://alpine-dav.github.io/ascent>), an *in-situ framework on HPC simulation* to interfere the simulation and **extract data to node-local NVRAMs** at a certain frequency (an interval in a number of cycles)
- Cloverleaf3D**, a 3D Lagrangian-Eulerian hydrodynamics benchmark, that is integrated to Ascent to test out this work



RANDOM FORESTS SPARK ANALYSIS

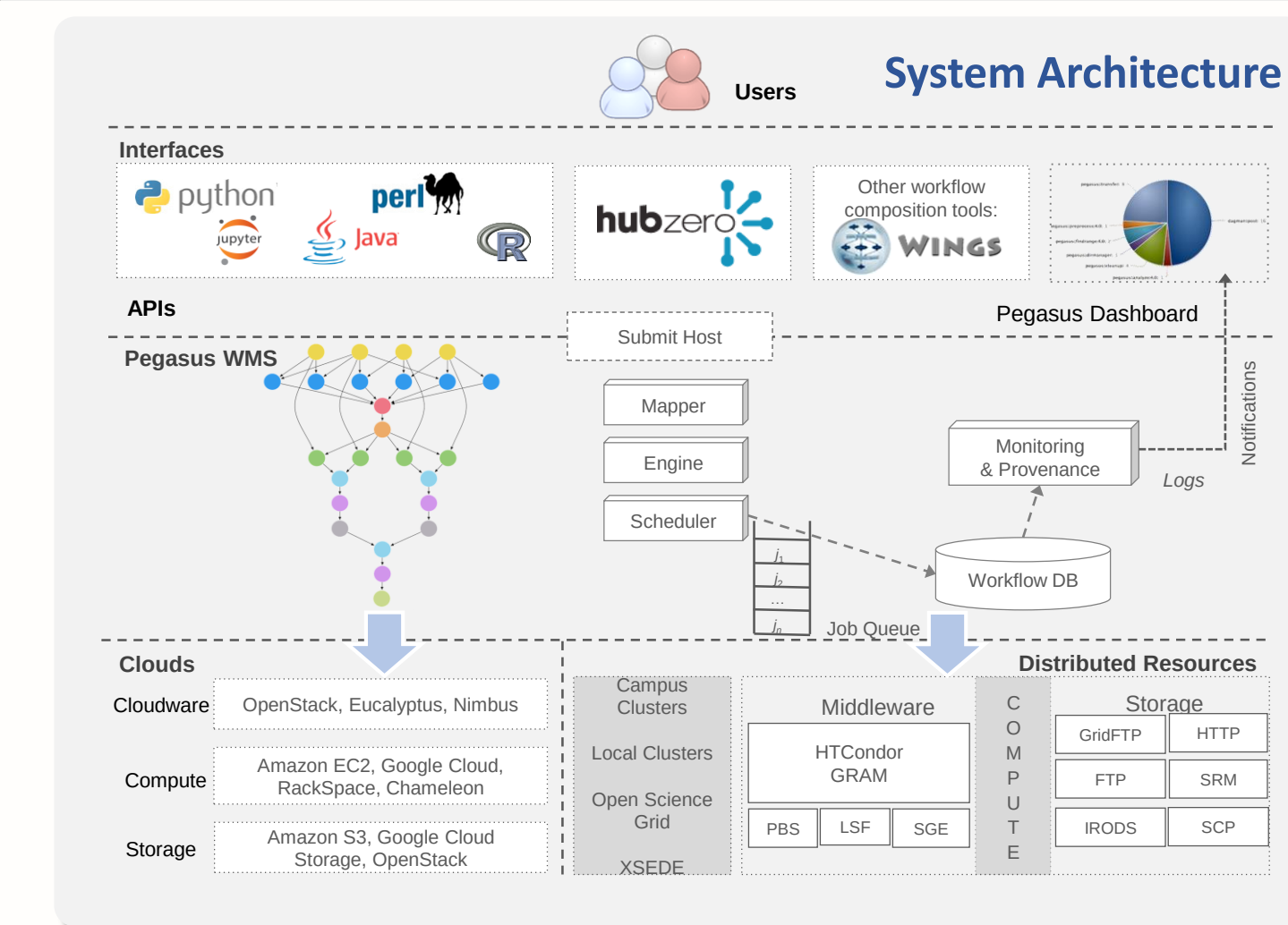
Big Data Analytics

- Random Forests** is a machine learning model for classification and regression
- We opt to use the random forests implementation in **Spark MLlib** as a representative analytics on Big Data

PEGASUS WORKFLOW MANAGEMENT SYSTEM

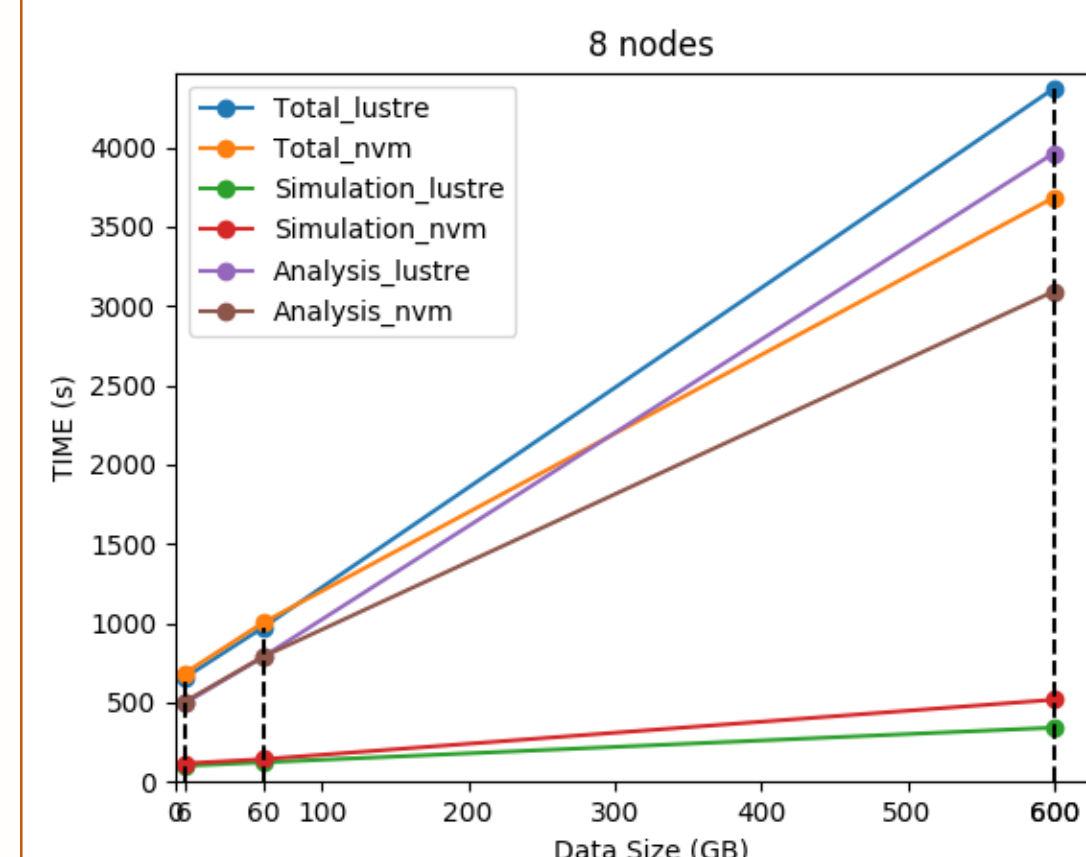
Overview of the Pegasus WMS

- Application **coordination at large-scale** demands to put more effort. Scientists employ the use of **scientific workflows** to **manage and automate** the large-scale simulation and associated analysis of the output datasets.
- Pegasus** (<https://pegasus.isi.edu>) is a **Workflow Management System** for mapping and executing abstract **application workflows** over a range of execution environments
- We utilize **Pegasus** for the coordination on entire pipelines. Pegasus plan a workflow indicating a dependency between the simulation and the analytics job



EVALUATION

Preliminary experimental results at LLNL Catalyst System



- The preliminary results on show approximately **28%** improvement in *time of the Spark analysis on NVRAM over Lustre at 600GB data*
- The simulation suffers 34% slowdown for copying data from local NVRAMs
- For the *total time*, the overhead is concealed and NRAM wins on approximately **19%** at large data compared to Lustre

CONCLUSION

Diamonds that deliver!

- A **node-local approach on NVRAM** to couple the HPC simulations with the Big Data analytics with the help of Pegasus
- The benefits of the approach are confirmed by the experiment on **HPC Catalyst** system with the **realistic simulation Cloverleaf3D** in Ascent and the representative **Random Forests** analytic in Spark MLlib