Event-Based Triggering and Management of Scientific Workflow Ensembles

Suraj Pandey
University of Hawaii
Honolulu, Hawaii
surajp@hawaii.edu

Karan Vahi
USC Information Sciences Institute
Marina Del Rey, California
vahi@isi.edu

Rafael Ferreira da Silva
USC Information Sciences Institute
Marina Del Rey, California
rafsilva@isi.edu

Ewa Deelman
USC Information Sciences Institute
Marina Del Rey, California
deelman@isi.edu

Ming Jiang
Lawrence Livermore National Lab
Livermore, California
jiang4@llnl.gov

Cyrus Harrison
Lawrence Livermore National Lab
Livermore, California
cyrush@llnl.gov

Al Chu
Lawrence Livermore National Lab
Livermore, California
chu11@llnl.gov

Henri Casanova
University of Hawaii
Honolulu, Hawaii
henric@hawaii.edu

ABSTRACT

As the scientific community prepares for extreme-scale computing, Big Data analytics are becoming an essential part of the scientific process enabling new insights and discoveries. This poster describes how we utilized workflow ensembles to model the next generation of computational workflows, where a long-running simulation job periodically generates data that needs to be analyzed concurrently on high-performance computing resources. In this work, we developed extensions to the open source Pegasus Ensemble Manager service to enable support for event-based triggering that can be used to add workflows to an existing running ensemble.

KEYWORDS

scientific workflows, workflow ensembles, distributed computing, Big Data analytics, Apache Spark

ACM Reference format:
Suraj Pandey, Karan Vahi, Rafael Ferreira da Silva, Ewa Deelman, Ming Jiang, Cyrus Harrison, Al Chu, and Henri Casanova. 2018. Event-Based Triggering and Management of Scientific Workflow Ensembles. In *Proceedings of International Conference for High Performance Computing in Asia-Pacific Region, Tokyo, Japan, January 2018 (HPC Asia 2018),* 4 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Scientists increasingly use scientific workflows to connect and manage distinct, individual pieces of scientific codes into larger scientific pipelines. These pipelines are often composed of a variety of different tasks ranging from bag of tasks, to large monolithic parallel codes. The execution and coordination of the scientific workflows are typically done using scientific workflow management systems [8]. These systems facilitate the execution of the user pipelines on a variety of supported cyberinfrastructure, ensuring that the tasks composing the pipelines are launched in the right order and executed reliably. The majority of workflow systems express workflows as a directed acyclic graph (DAG) of jobs, whereby nodes represent workflow tasks that are linked via dataflow and control flow edges.

With the push towards extreme-scale computing, it has become possible to run traditional high-performance computing (HPC) scientific application codes at an ever bigger scale utilizing tens of thousands of cores for computing. Running application codes at this scale also necessitates the ability to analyze the generated outputs at regular intervals without waiting for the complete output datasets to be generated. This is particularly useful because it gives the scientist a window into how their computations are progressing, and an ability to identify any errors in the computation in early steps, allowing them to kill or modify their workflows without wasting too many computing cycles. At the same time, newer techniques for analyzing the large amounts of generated output data are gaining traction, such as the use of Big Data analytics framework [7]. Examples of such computational workflows already exist, such as the recent work on integrating machine learning algorithm with Arbitrary Lagrangian-Eulerian simulations for failure prediction [4]. In order to scale up the analysis for large-scale simulations, one would need to utilize a Big Data framework, such as Apache Spark [9], to enable distributed machine learning.

However, leveraging Big Data analytics with HPC simulations in an effective fashion remains a major challenge for the current generation of scientific workflow management systems (WMS) [2]. This primarily stems from the fact that in a DAG-based framework, a job can only run when all its parent jobs have successfully finished. Obviously, this approach has two clear disadvantages: 1) the amount of simulation data that must be saved for post-processing—all the output data needs to be generated before the analysis jobs can be launched; and 2) the increase in overall runtime by not overlapping the simulation job with the analysis job.

HPC Asia 2018, January 2018, Tokyo, Japan

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

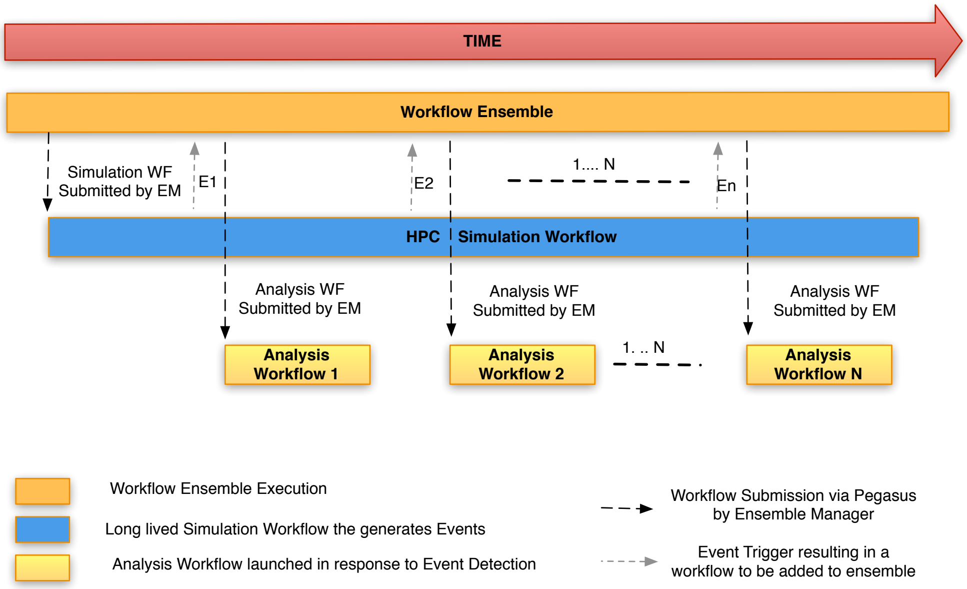
TIME
Workflow Ensemble
Simulation WF Submitted by EM
E1
E2
1.... N
En
HPC Simulation Workflow
Analysis WF Submitted by EM
Analysis WF Submitted by EM
Analysis WF Submitted by EM
Analysis Workflow 1
Analysis Workflow 2
1. .. N
Analysis Workflow N
Workflow Ensemble Execution
Long lived Simulation Workflow the generates Events
Analysis Workflow launched in response to Event Detection
Workflow Submission via Pegasus by Ensemble Manager
Event Trigger resulting in a workflow to be added to ensemble

Figure 1: Execution timeline for event based workflow ensemble

The goal is to have a coupling of simulation and analytics, where a long-running simulation job periodically generates data and that data is passed to analytics, such as those supported by Apache Spark, while the remainder of the simulation is progressing. This poster explores the use of workflow ensembles with event-based triggers to manage this type of computational workflows. As the simulation progresses, new data analytics workflows are created and launched.

2 APPROACH

In this work, instead of trying to model the computation pipeline (HPC simulation jobs, followed by one or more analytics job) in a single workflow, we decided to model it as a collection of workflows that can be launched and executed in tandem based on certain triggers. To this end, we utilize a service called Ensemble Manager, which has been developed as part of the open-source Pegasus WMS [3]. It manages collections of workflows called ensembles and is useful for managing a set of workflows that need to be executed over a long period of time. The ensemble manager allows users to issue commands to dynamically add (or remove) a workflow to an existing ensemble.

In order to utilize the Ensemble Manager for automatically launching analysis workflows consisting of Big Data analytics jobs at regular intervals, we developed an extension that enables users to specify event-based triggers. These triggers result in the generation and addition of new workflows to an existing running workflow ensemble. This extension can support two types of triggers that are specified in a JSON formatted configuration file (one per ensemble):

(1) *File-based*: Triggers an event based on the modification of a file.
(2) *Directory-based*: Triggers an event based on the modification or presence of files inside a directory. The number of files to look for can be specified as a parameter.

The JSON event trigger format is illustrated below in Listing 1.

```
{
  "event-dir": "/dir/keep-track",
  "event-content": "*",
  "event-type": "file-dir",
  "event-cycle": 10,
  "event-size": 0,
  "event-numfiles": 1,
  "pegasus-args": "/dir/workflow-invocation
      -script",
  "event-script": "/dir/workflow-generation
      -script",
  "event-dax-dir": "/dir/pegasus/workflows"
}
```

Listing 1: Ensemble Manager JSON event trigger format.

The attributes shown in Listing 1 are described below:

- *event-dir* tells where to seek for the trigger files;
- *event-content* tells where to seek for any file names;
- *event-type* tells whether the content to watch over is a directory or a file;
- *event-cycle* is the number of times such triggers will occur;
- *event-size* is the size of trigger files to look for;
- *event-numfiles* is the number of files after which trigger will occur;
- *pegasus-args* is the path to a shell script containing the invocation for planning the generated workflow (i.e., a call for the pegasus-plan command);

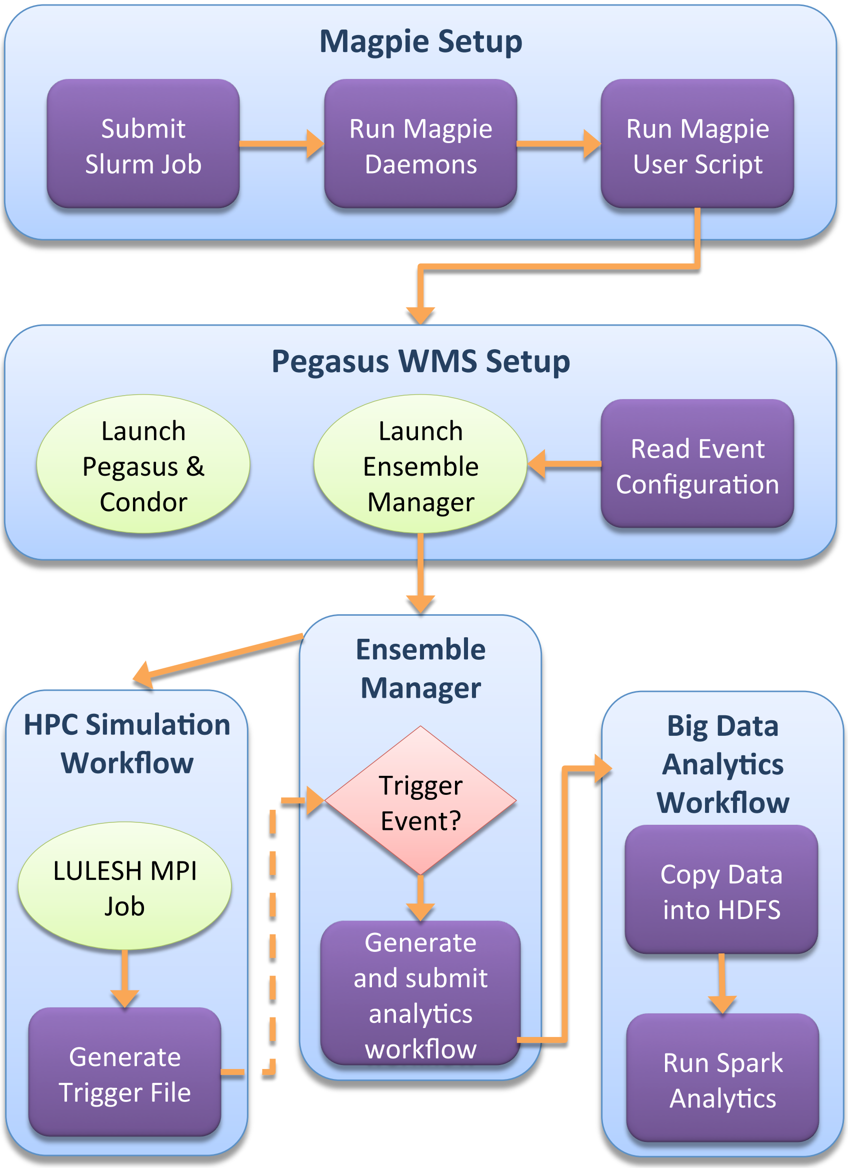

Figure 2: Experimental Setup at LLNL Catalyst Cluster

- *event-script* is the script that is launched when an event is detected;
- *event-dax-dir* is the directory where the Pegasus workflows are generated.

With these extensions, the problem of automatically launching data analytic jobs alongside a long running HPC simulation job can be modeled as follows. Initially, the workflow ensemble contains a single workflow that can generate triggers during its execution. This workflow consists of the traditional HPC simulation jobs that generate output data at regular intervals and write to disk. The generation of the output data to a particular directory is the event that the workflow ensemble is configured with. As the event specified in the configuration file occurs, a user specified script, pre-specified in the configuration file, is executed to create a new Pegasus workflow consisting of a Big Data analytics job, and added to the running ensemble. The ensemble manager then submits this newly added workflow for execution through Pegasus. A generic timeline illustrating how the event based-workflow ensemble execution works is shown in Figure 1.

We deployed and tested this approach using data generated from the LULESH proxy application [6] on an HPC cluster (Catalyst [1]) at the Lawrence Livermore National Laboratory (LLNL). Catalyst is designed specifically for experimentation with HPC and Big Data analytics. It is a 150 teraFLOP/s system with 324 nodes, each with 128 gigabytes of dynamic random access memory and 800 gigabytes of non-volatile memory. We tested a long-running MPI instance of LULESH periodically producing output for analysis using Apache Spark. Since we were primarily interested in creating and executing the child analysis job at the right point of time rather than actual analysis of the simulation output, we created an example Spark job and executed it as the specified events occurred. The experimental setup is illustrated in Figure 2.

We used Magpie [5], an open-source project developed at LLNL for running Big Data software in HPC environments, including Hadoop and Spark. For our experiments, we submit a Magpie SLURM job. After nodes are allocated, Magpie does the following steps (Run Magpie Daemons box in Figure 2):

- Determines which nodes will be “master” nodes, “slave” nodes, or other types of nodes.
- Sets up, configures, and starts appropriate Big Data daemons to run on the allocated nodes. In our setup, we used the Magpie SPARK template to setup a dynamic Spark cluster on the allocated nodes.
- Reasonably optimizes the configuration for the given cluster hardware that it is being run on.
- Magpie then executes a user specified script to give control back to the user.

In the Magpie’s user specified script (Run Magpie User Script box in Figure 2), we do the following:

- Start Pegasus + HTCondor on the Magpie master node.
- Start the Ensemble Manager Service
- Submit the experiment ensemble for execution.

The ensemble submitted initially consists of a single workflow that launches the LULESH MPI application. As LULESH executes, it periodically (in our case every 10 simulation cycles) writes out outputs to a directory on the shared filesystem that is tracked by the Ensemble Manager as part of the event trigger specified, when the ensemble is first submitted. As the new files are generated, the Ensemble Manager invokes a script that generates the data analytics workflow on the newly generated datasets. The data analytics workflow consists of two jobs:

1. *HDFS copy job*: This job takes the newly created output files by LULESH and puts them into the Hadoop Distributed File System (HDFS) to facilitate Spark analysis.
2. *Spark Data Analysis job*: The example Spark analysis job.

The key insight behind this work is the creation of the ensemble, which allows us to associate it with a configuration file, where we can specify the trigger events to look for. We spawn a separate process from the Ensemble Manager to look for those events. As the specified event occurs, a new workflow comprising Spark analysis is created and added to the running ensemble. It is important to note that our approach is neither dependent on the type of HPC application used nor on the Big Data framework used, as evident in the execution timeline of the event based workflow ensemble shown in Figure 1. The same approach can be used for any user application that is represented as a Pegasus workflow without any changes to the Ensemble Manager code.

3 CONCLUSION

In this poster, we described a new approach for automatically launching Big Data analytics jobs in tandem with long running HPC simulation jobs, via the use of workflow ensembles. We tested our approach by running both the LULESH application and a sample Spark analysis job on the same set of nodes on Catalyst at LLNL. LULESH produced simulation output along with the trigger files in a directory every 10 simulation cycles, which the Ensemble Manager tracked. Using our event-based triggering approach, new Spark analysis workflows, consisting of a HDFS copy job and Spark job were automatically submitted as part of the overall workflow.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL) under contract DE-AC52-07NA27344. This work was partially funded by the Laboratory Directed Research and Development Program at LLNL under project 16-ERD-036. Pegasus is currently funded by the National Science Foundation(NSF) under the OAC SI2-SSI grant #1664162. Previously, NSF has funded Pegasus under OCI SDCI program grant #0722019 and OCI SI2-SSI program grant #1148515.

REFERENCES

[1] Catalyst 2016. https://computation.llnl.gov/computers/catalyst. (2016).

[2] Rafael Ferreira da Silva, Rosa Filgueira, Ilia Pietri, Ming Jiang, Rizos Sakellariou, and Ewa Deelman. 2017. A characterization of workflow management systems for extreme-scale applications. *Future Generation Computer Systems* 75 (2017), 228 – 238. https://doi.org/10.1016/j.future.2017.02.026

[3] Ewa Deelman, Karan Vahi, Gideon Juve, Mats Rynge, Scott Callaghan, Philip J Maechling, Rajiv Mayani, Weiwei Chen, Rafael Ferreira da Silva, Miron Livny, and Kent Wenger. 2015. Pegasus: a Workflow Management System for Science Automation. *Future Generation Computer Systems* 46 (2015), 17–35. https://doi.org/10.1016/j.future.2014.10.008

[4] M. Jiang, B. Gallagher, J. Kallman, and D. Laney. 2016. A Supervised Learning Framework for Arbitrary Lagrangian-Eulerian Simulations. In *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 977–982. https://doi.org/10.1109/ICMLA.2016.0176

[5] Magpie 2016. http://github.com/LLNL/magpie. (2016).

[6] M. B. Gokhale R. D. Hornung, J. A. Keasler. 2011. *Hydrodynamics Challenge Problem, Lawrence Livermore National Laboratory.* Technical Report LLNL-TR-490254. 1–17 pages.

[7] Daniel A. Reed and Jack Dongarra. 2015. Exascale Computing and Big Data. *Communications of ACM* 58, 7 (2015), 56–68.

[8] Ian J. Taylor, Ewa Deelman, Dennis B. Gannon, and Matthew Shields. 2014. *Workflows for e-Science: Scientific Workflows for Grids.* Springer Publishing Company, Incorporated.

[9] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. 2010. Spark: Cluster Computing with Working Sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud’10)*. USENIX Association, Berkeley, CA, USA, 10–10. http://dl.acm.org/citation.cfm?id=1863103.1863113