panorama360

Workflow Submit Nodes as a Service on Leadership Class Systems

George Papadimitriou, Karan Vahi, Jason Kincl, Valentine Anantharaj,
Ewa Deelman, Jack Wells

Practice and Experience in Advanced Research Computing – PEARC20
July 30th, 2020

This work was funded by the US Department of Energy under Grants DE-SC0012636M, DE-AC02-06CH11357, DE-AC05-00OR22725

U.S. DEPARTMENT OF ENERGY | Office of Science

Background – Pegasus Workflow Management System

Automates complex, multi-stage processing pipelines

Enables parallel, **distributed computations**

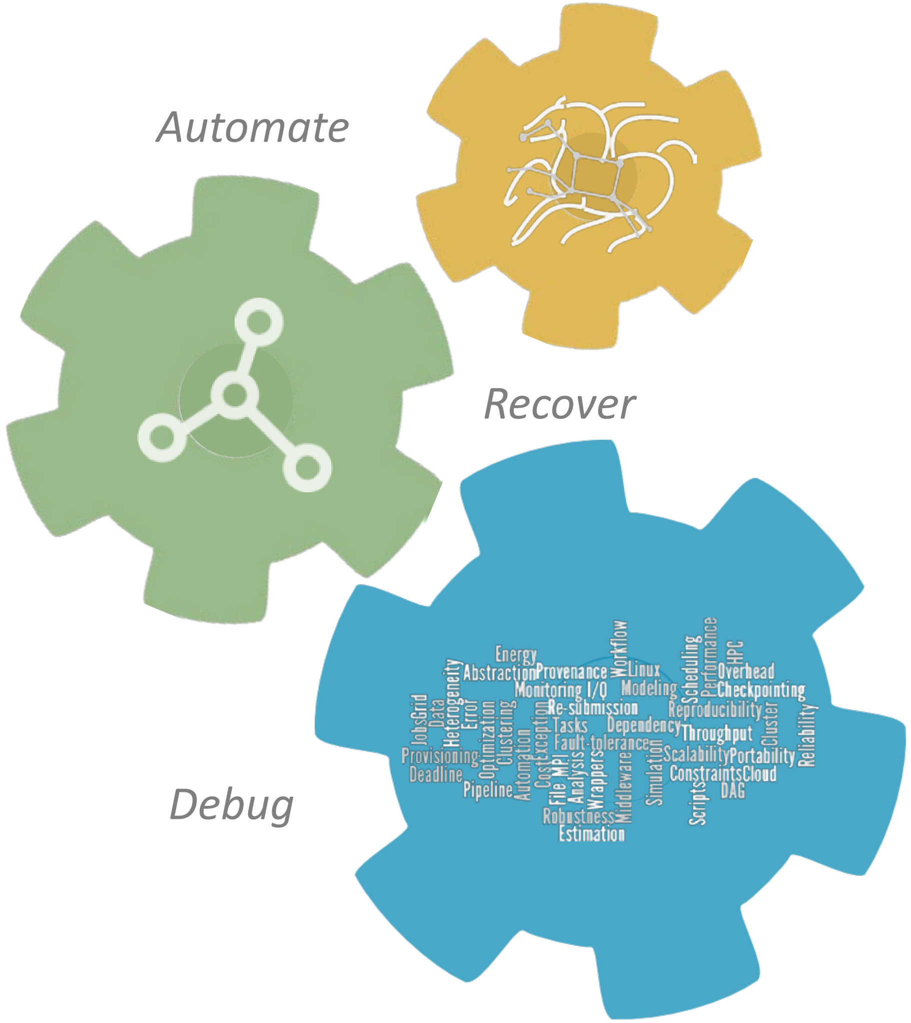
Automatically executes data transfers

Reusable, aids **reproducibility**

Records how data was produced (**provenance**)

Handles **failures** with to provide reliability

Keeps track of data and **files**



NSF funded project since 2001, with close collaboration with HTCondor team

Background – Oak Ridge Leadership Computing Facility (OLCF)

- OLCF is supported by the US Department of Energy and offers leadership-class computing resources to researchers

- It provides to users HPC resources, ranging from traditional x86 systems to systems with GPUs and ARM based processors.

- IBM AC922 Summit is OLCF’s flagship system with capability of 200 petaFLOPS.

- Strict security is enforced on the systems and 2-factor authentication processes has been adopted.

Reference: https://www.olcf.ornl.gov/

Motivation

- We want to make Pegasus workflows more **accessible** by OLCF users.
- Provide an easy way to target the **Summit** supercomputer and other OLCF's resources.

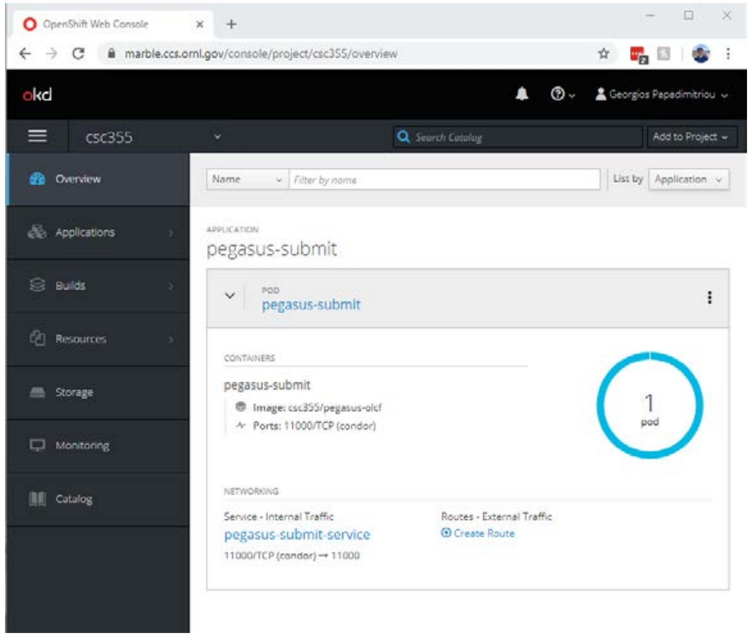
Ideally: A solution that doesn't require **debugging** and that **works out of the box** for all users.

Challenges

- Simple to use.
- 2-factor authentication.
- In HPC systems, workflow environments usually require expertise to build, setup and deploy.
- For every system that becomes available re-deployment is needed.
- Long-running processes might be prohibited by system administrators.

Approach – Kubernetes at OLCF

- **Kubernetes** is an open-source platform for running and coordinating containerized application across a cluster of machines.
- OLCF has deployed OpenShift, a distribution of Kubernetes developed by RedHat



marble.ccs.ornl.gov/console/project/csc355/overview

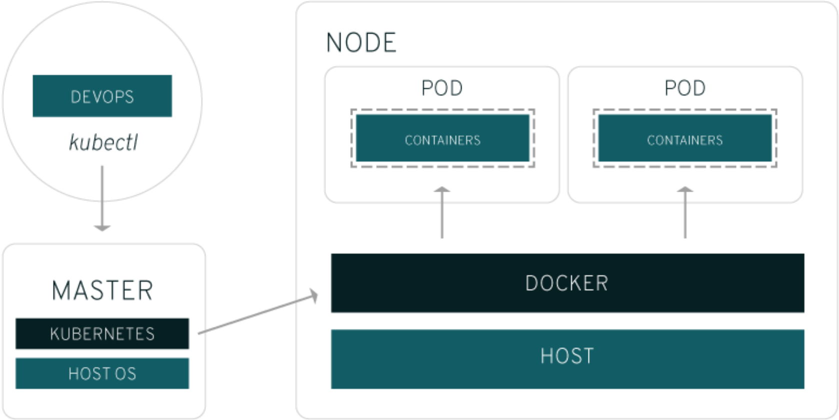
APPLICATION
pegasus-submit

POD
pegasus-submit

CONTAINERS
pegasus-submit
Image: csc355/pegasus-olcf
Ports: 11000/TCP (condor)

NETWORKING
Service - Internal Traffic
pegasus-submit-service
11000/TCP (condor) → 11000

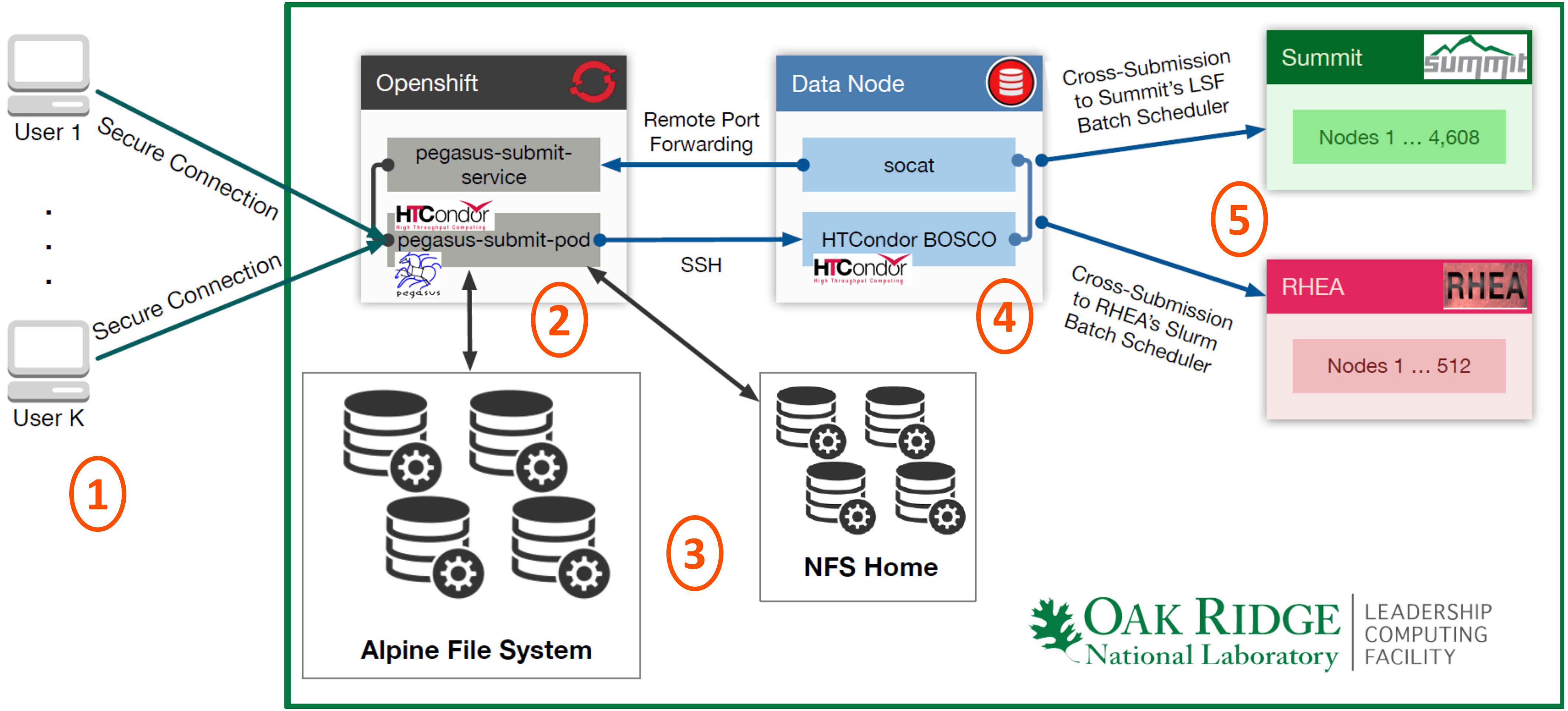
Routes - External Traffic
Create Route



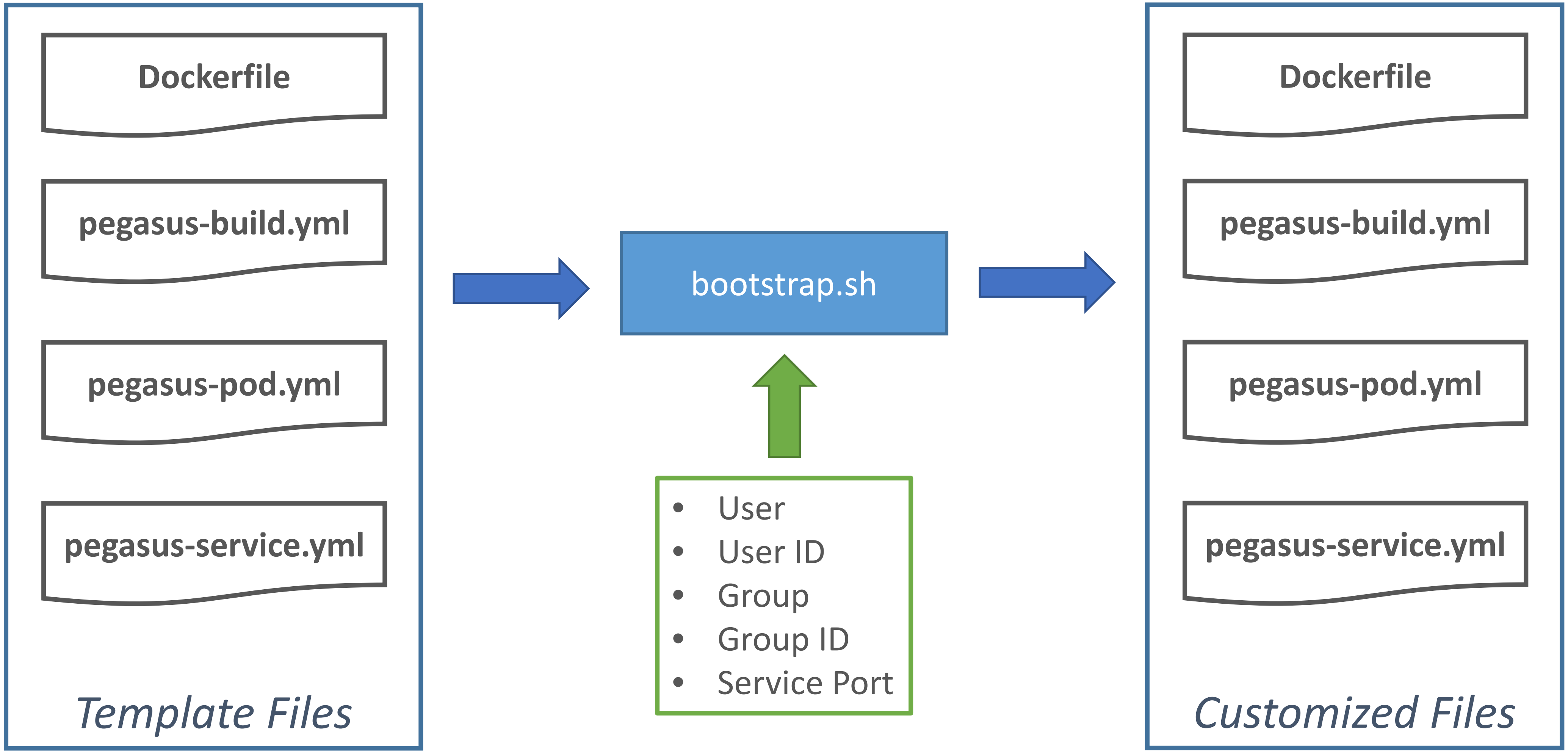
- OpenShift provides a command line and a web interface to manage your Kubernetes objects (pods, deployments, services, storage etc.)
- OLCF's deployment has automation mechanisms that allow users to submit jobs to the batch system and access the shared file systems (NFS, GPFS)
- All containers run as an automation user that is tied to a project

References:
https://www.redhat.com/en/topics/containers/what-is-kubernetes
https://www.olcf.ornl.gov/wp-content/uploads/2017/11/2018UM-Day3-Kincl.pdf

Approach – Pegasus Workflow Submit Nodes as a Service



Approach – Pegasus Kubernetes Templates



Dockerfile

pegasus-build.yml

pegasus-pod.yml

pegasus-service.yml

Template Files

bootstrap.sh

- User
- User ID
- Group
- Group ID
- Service Port

Dockerfile

pegasus-build.yml

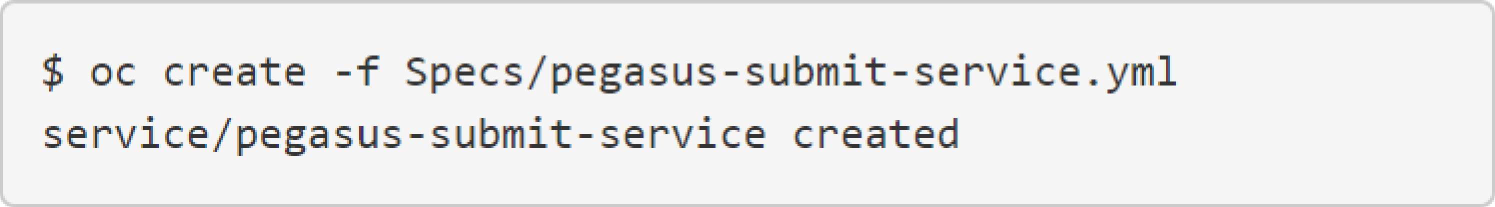
pegasus-pod.yml

pegasus-service.yml

Customized Files

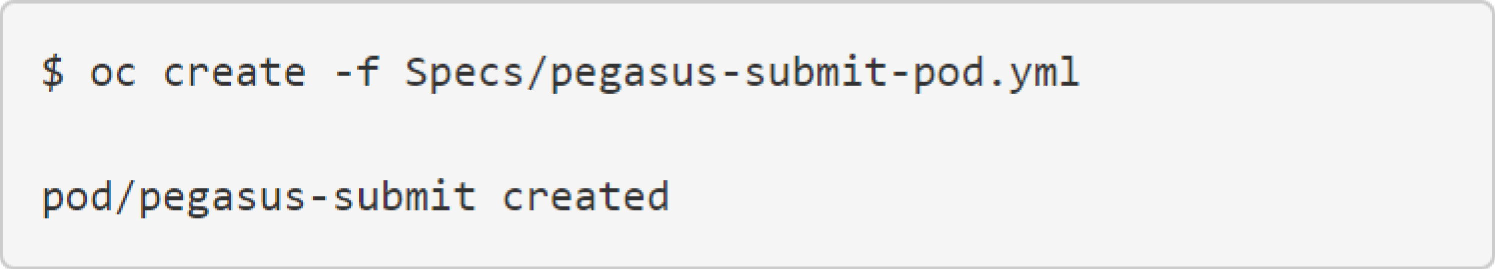
Approach – Simplicity

Start a Kubernetes Service that will expose Pegasus’ submit pod’s services:



```
$ oc create -f Specs/pegasus-submit-service.yml
service/pegasus-submit-service created
```

Start a Kubernetes Pod with Pegasus and HTCondor:



```
$ oc create -f Specs/pegasus-submit-pod.yml

pod/pegasus-submit created
```

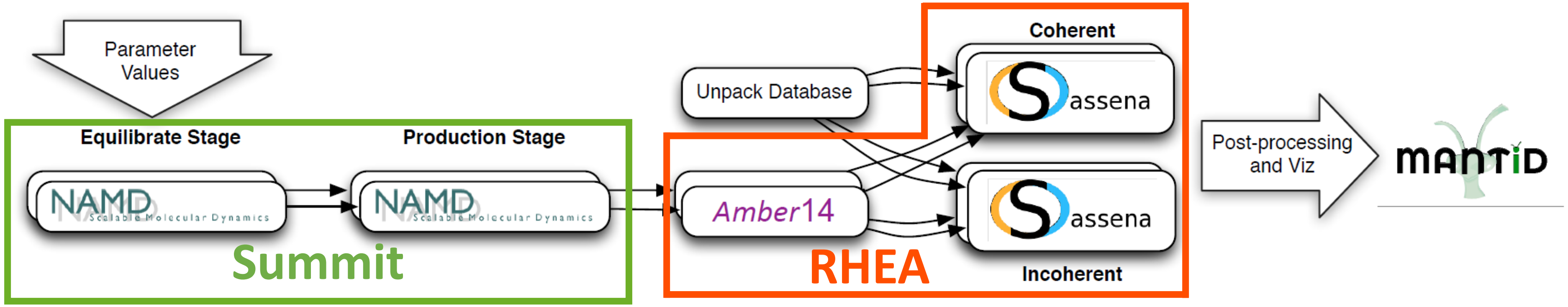
Approach – Benefits

- Containers and Kubernetes deployment templates **simplify** Pegasus workflow **environments** at OLCF.
- Pegasus submit nodes can be deployed as a service withing a few seconds.
- HTCondor’s BOSCO SSH style submissions on the DTNs achieve submissions to both the SLURM and LSF batch schedulers.
- A single workflow can access **all** OLCF’s compute resources.
- No long running processes on OLCF’s Login nodes !

Evaluation – Method

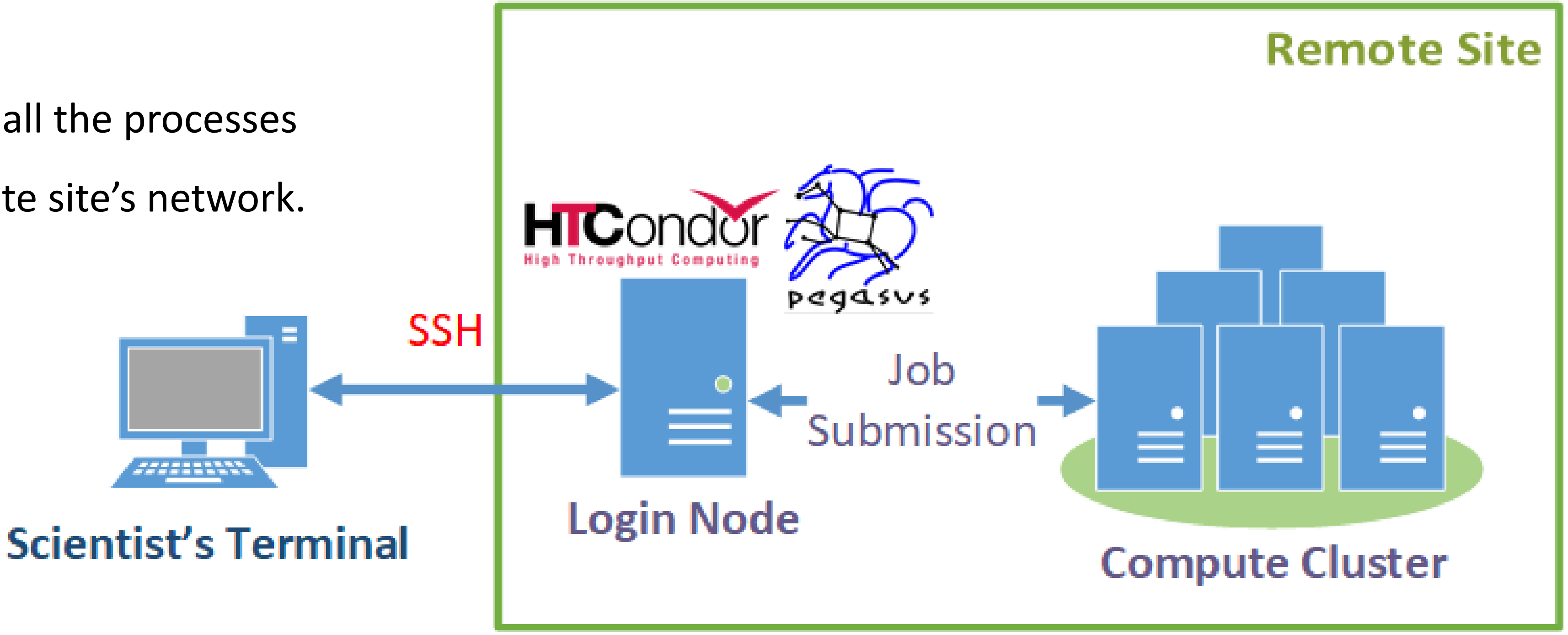
- We compare the Kubernetes solution with the “Login Node” and “rvGAHP” solutions
  - How much time do they need to be deployed ?
  - Does the Kubernetes solution introduce any submission delays ?
  - What limitation does the Kubernetes solution have ?

- To evaluate submission delays we used the *Nanodiamond Pegasus workflow* created by spallation neutron source (SNS) scientists at ORNL.
  - Each workflow contains 11 compute jobs (8 MPI and 3 single core jobs).
  - We submitted 300 workflows and 990 compute jobs in total.



Evaluation – Pegasus on Login Nodes

- The Login Node deployment is a commonly used approach.
- A submission environment must be set up on the Login Nodes of the HPC system.
- Users need to execute long running processes on the Login Nodes.
- Security is not an issues since all the processes communicate inside the remote site's network.

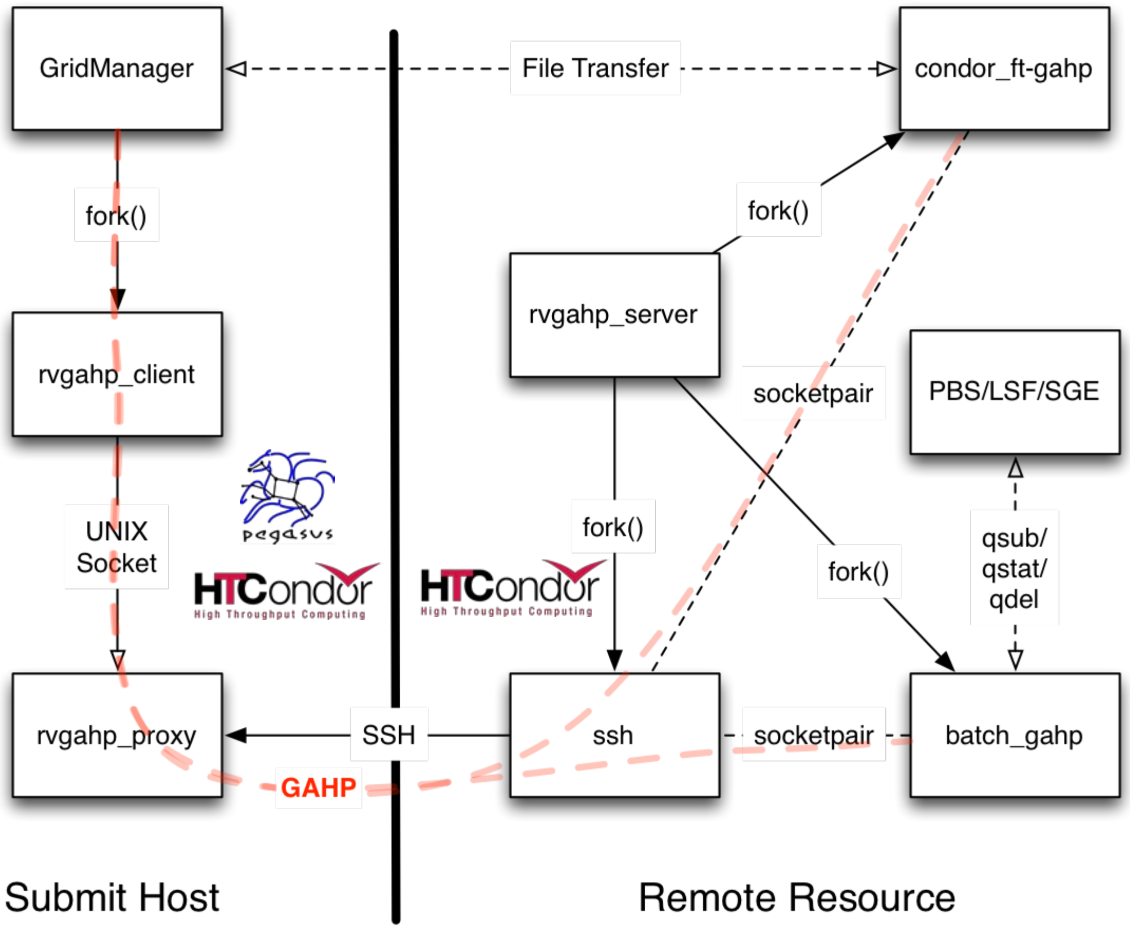


Evaluation – Pegasus using rvGahp

- The Reverse GAHP enables remote job submissions.
- It doesn't require the remote resource to run services that accept incoming network connections or to accept SSH connections **without** 2-factor authentication.
- An SSH connection is established from the Remote Resource to the Submit Host allowing communications between the two ends with Unix sockets.

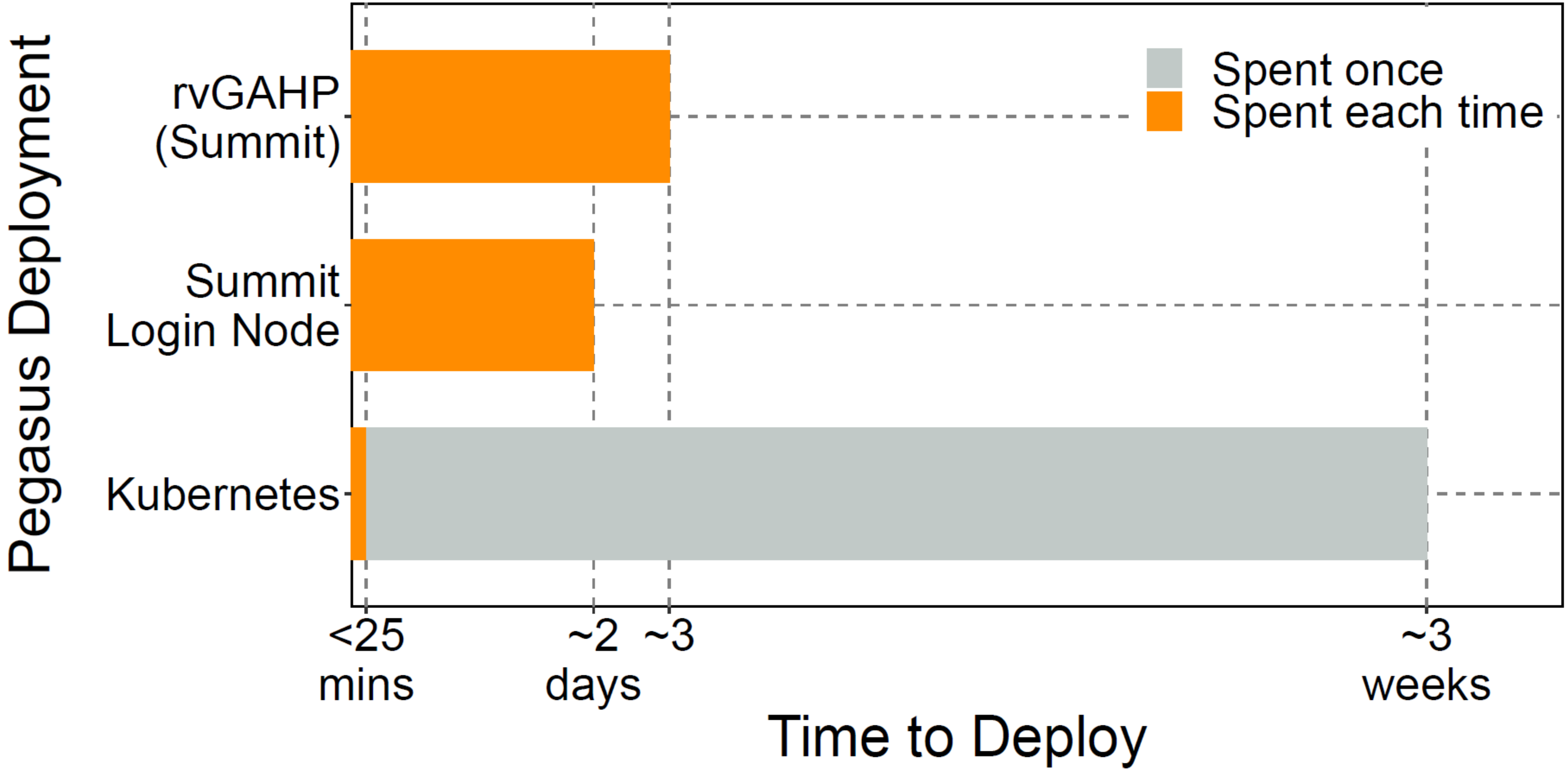
Southern California Earthquake Center

https://www.scec.org/article/522



Reference: https://github.com/pegasus-isi/rvgahp

Evaluation – Time to Deploy



Evaluation – Queue Delays, Overhead

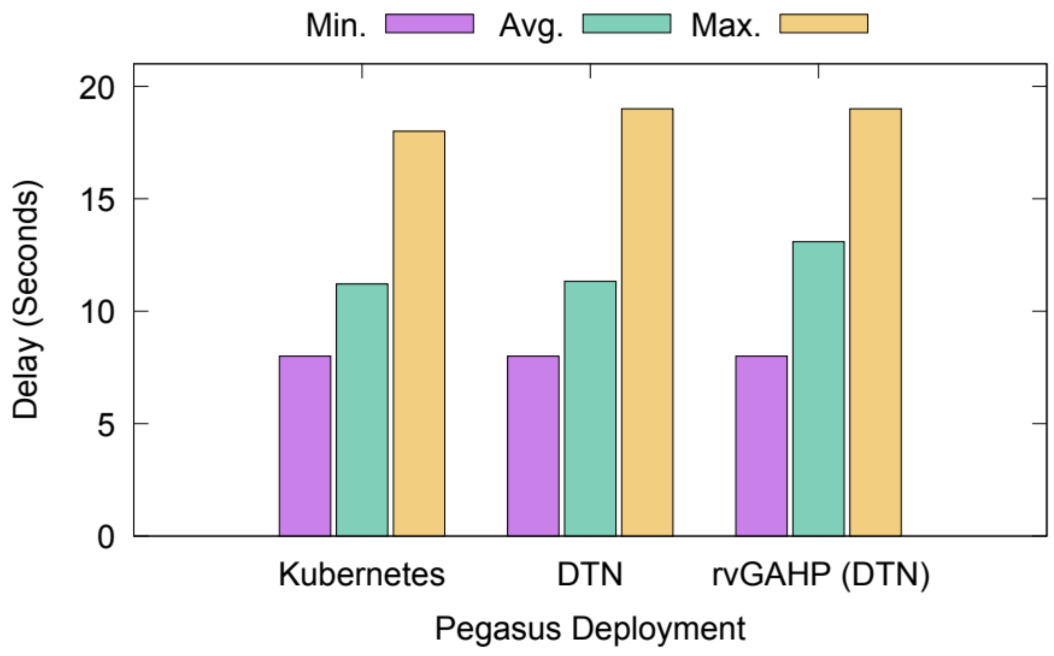


Figure 4: HTCondor Queue Time to Submission

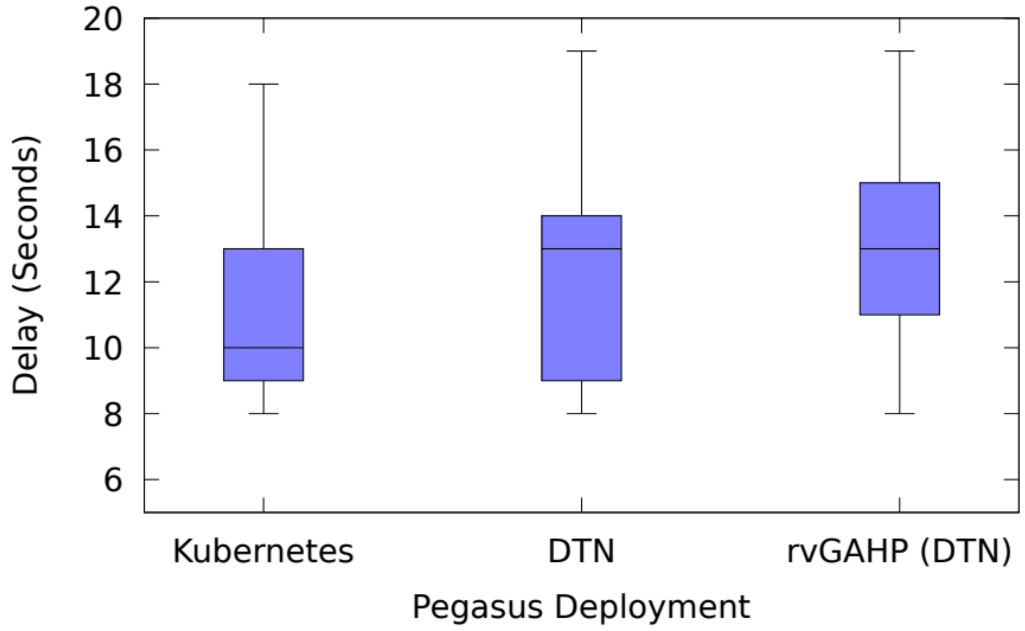


Figure 5: HTCondor Queue Time Distribution

Statistics from 990 compute jobs to the batch queues at OLCF !

Evaluation – Limitations

- Unlike “rvGAHP” the Kubernetes solution can only be used to submit workflows from within OLCF’s DMZ. (no remote submissions)
- It imposes restrictions on the number of running jobs a single project allocation can have.
  - OLCF applies per-user limits
  - Since all the workflows are submitted under the automation user, the number of jobs a project can run via the Kubernetes deployment is limited by the automation user limits.

Conclusion

- First time Pegasus users at OLCF can now deploy a working submission environment following a few well documented steps in less than 30minutes.
- Users can maintain their own long running services without creating headaches for the system administrators.
- The deployment relies on OLCF’s automations and abstractions, providing access to the shared filesystem, Summit, RHEA and the DTNs.
- In comparison to “Login Node” and “rvGAHP” deployments, no overhead was added to submissions.

Try it out, it’s easy!

GitHub

https://pegasus.isi.edu/tutorial/summit/

https://github.com/pegasus-isi/pegasus-olcf-kubernetes panorama360

pegasus

Pegasus 5.0 will be released soon!

Pegasus Online Office Hours

https://pegasus.isi.edu/blog/online-pegasus-office-hours/

Thank you!

George Papadimitriou
Computer Science PhD Student
University of Southern California
email: georgpap@isi.edu

USC Viterbi
School of Engineering
Department of Computer Science