rvGAHP – Push-Based Job Submission Using Reverse SSH Connections

Scott Callaghan, Gideon Juve, Karan Vahi, Philip J. Maechling,
Thomas H. Jordan and Ewa Deelman

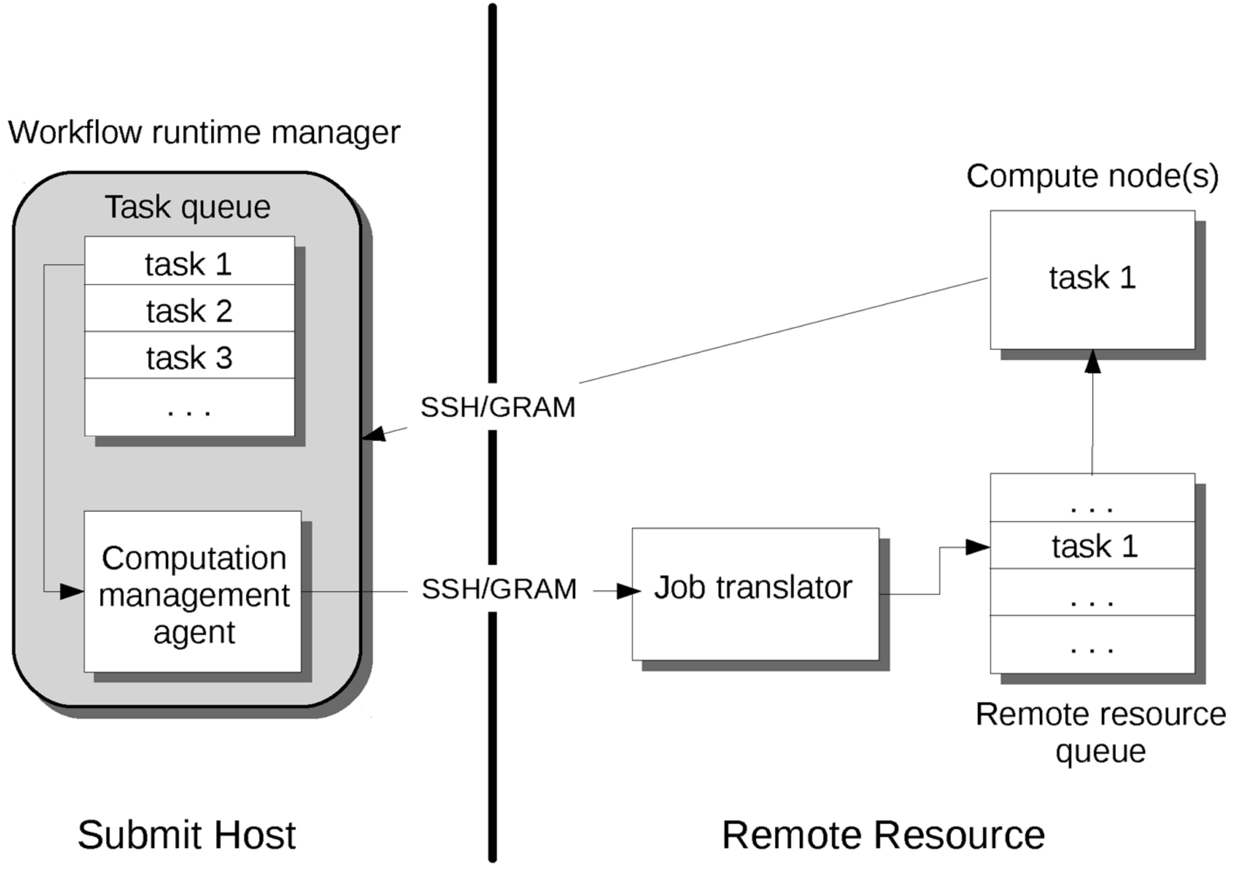
WORKS ‘17
November 13, 2017

Motivation

- Scientific workflows continue to scale up
- Require automated resource provisioning on largest available clusters
- Two main approaches:
  - Push-based: requests initiate from workflow queue
  - Pull-based: requests initiate from remote resource
- Both have limitations
  - Push-based: not viable on systems with two-factor authentication
  - Pull-based: risk of high overhead with heterogeneous workloads
- rvGAHP: alternative approach

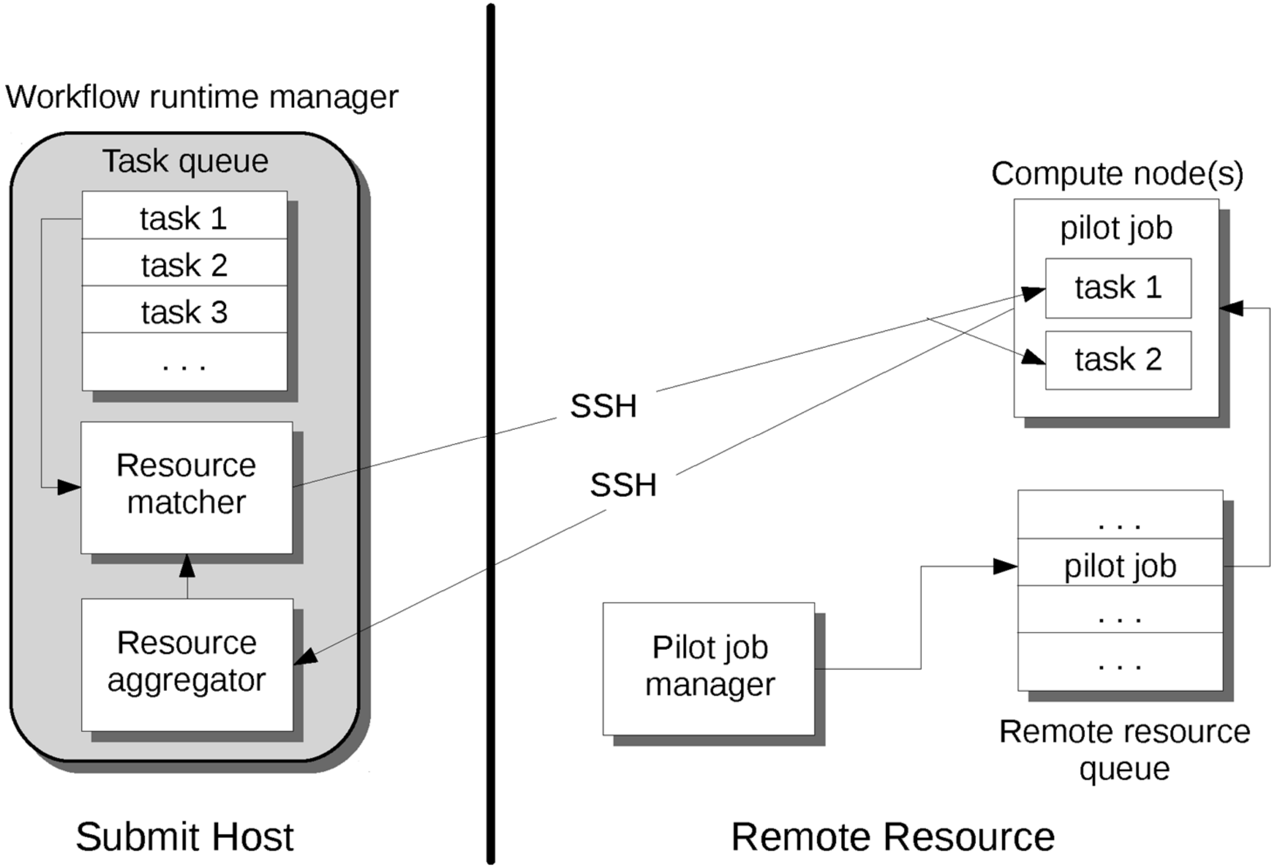
Push-Based Approach

- Resource requests are on-demand, initiating from runtime manager
- Request waits in remote queue, starts up, runs workflow task
- Efficient use of remote resources: nodes only used when workflow tasks are being run
- Requires automated authentication to remote system



Pull-Based Approach

- Resource requests initiate from remote resource ('pilot job')
- Resources advertised to workflow submission host for task scheduling
- Requires authentication from remote system to workflow host
- Can incur overhead
  - Waiting for workflow task
  - Mismatch of pilot job to task size and length



Summary of Approaches

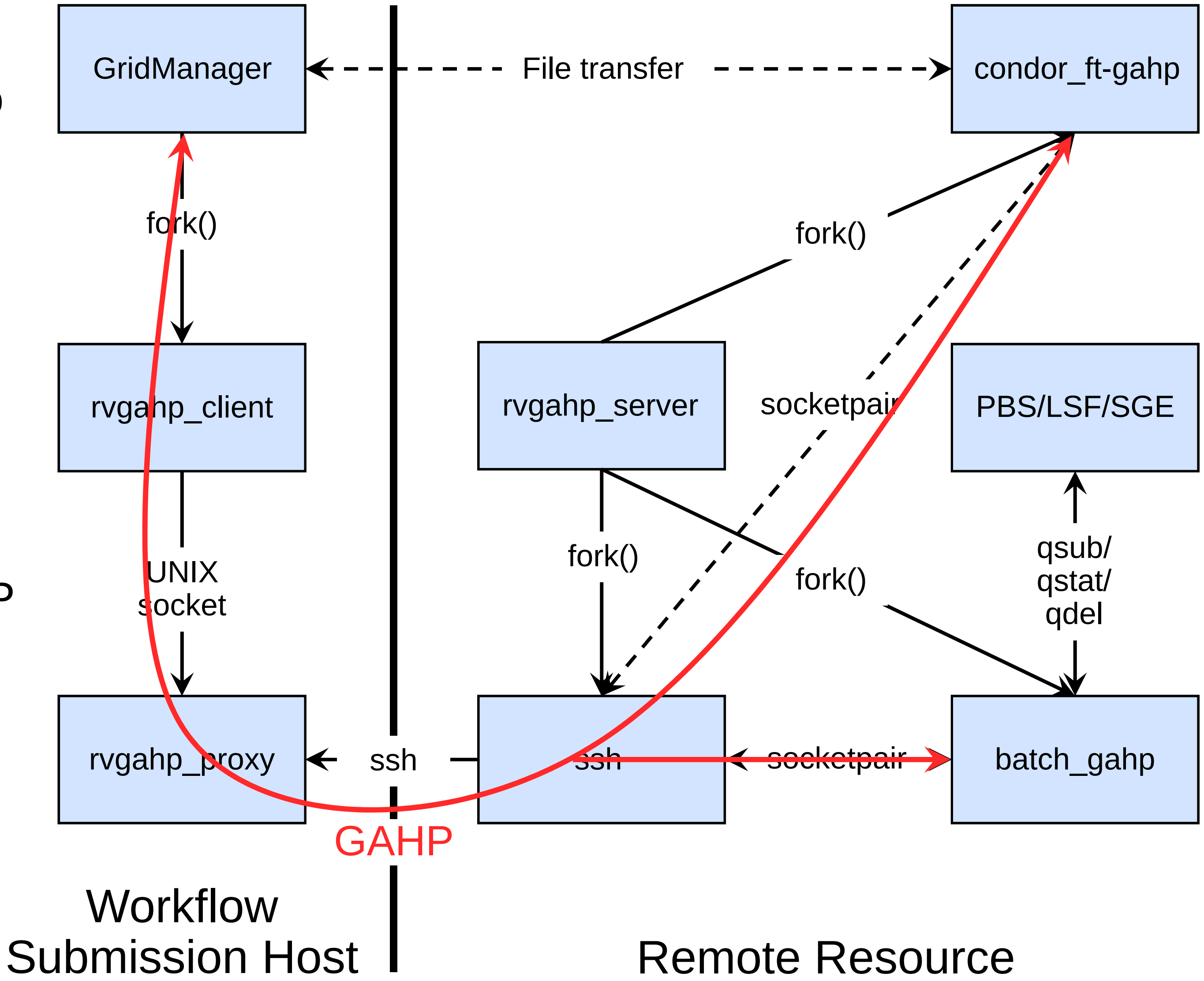
- Push-based
  - Potential issues with automated authentication
  - Little overhead: node-hours burned are used for workflow work
- Pull-based
  - Authentication performed from remote system to workflow host
  - Risk of high overhead, especially for heterogeneous workloads
- Would like solution with overhead characteristics of push-based, but with authentication flexibility of pull-based

rvGAHP Overview

- Developed new solution, reverse GAHP (rvGAHP)
  - Implementation of text-based GAHP protocol
- Push-based approach
- Does not require automated authentication to remote resource, just to workflow submission host
- Does require daemon running on remote resource
- Integrates cleanly with HTCondor

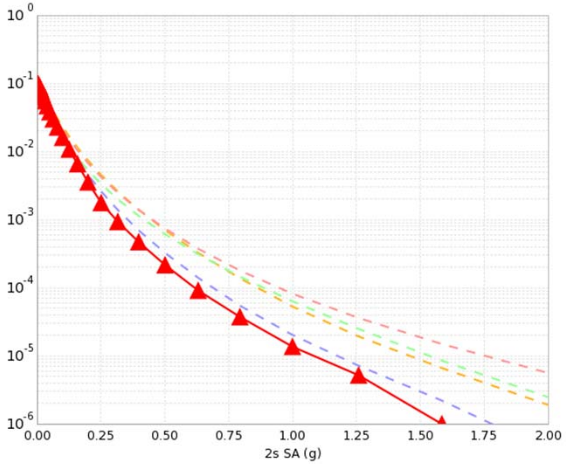
rvGAHP Details

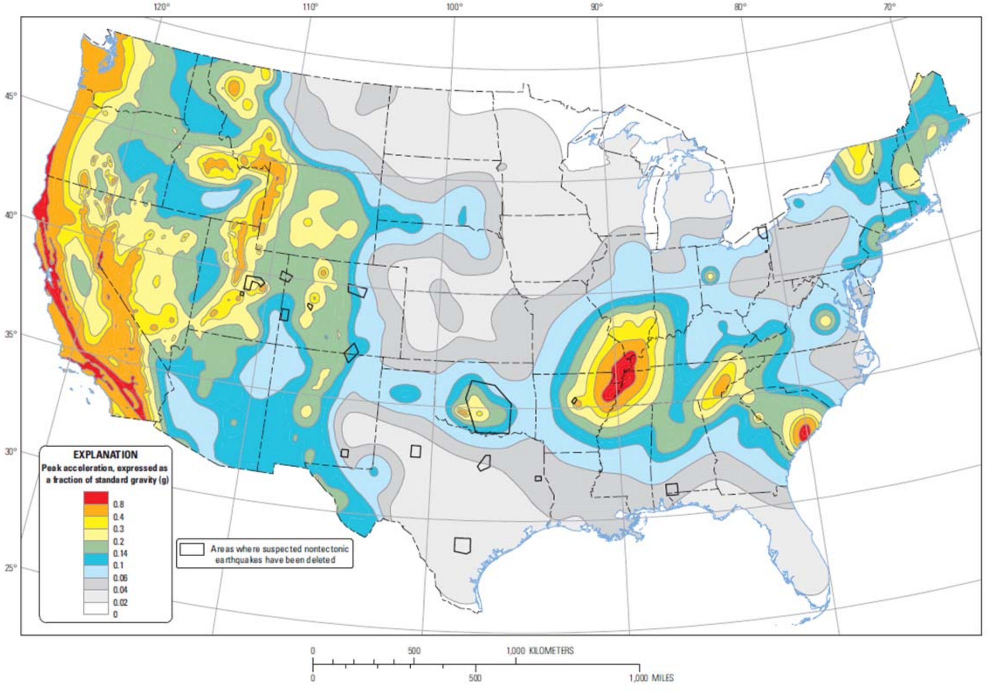
1. rvgahp_server is started on remote resource
2. rvgahp_server establishes ssh connection to workflow submission host and starts rvgahp_proxy
3. When remote GAHP job is submitted, GridManager launches rvgahp_client
4. rvgahp_client, via rvgahp_proxy, starts appropriate GAHP process on remote resource
5. A socketpair is set up between remote GAHP process and ssh process, establishing communication between GridManager and remote GAHP process
6. Another ssh process and rvgahp_proxy are started by the rvgahp_server (not shown)



Application Overview

- Probabilistic Seismic Hazard Analysis (PSHA)
- What will ground motion be in the next 50 years?
  - Building engineers
  - Insurance companies
  - Disaster planners





Two-percent probability of exceedance in 50 years map of peak ground acceleration

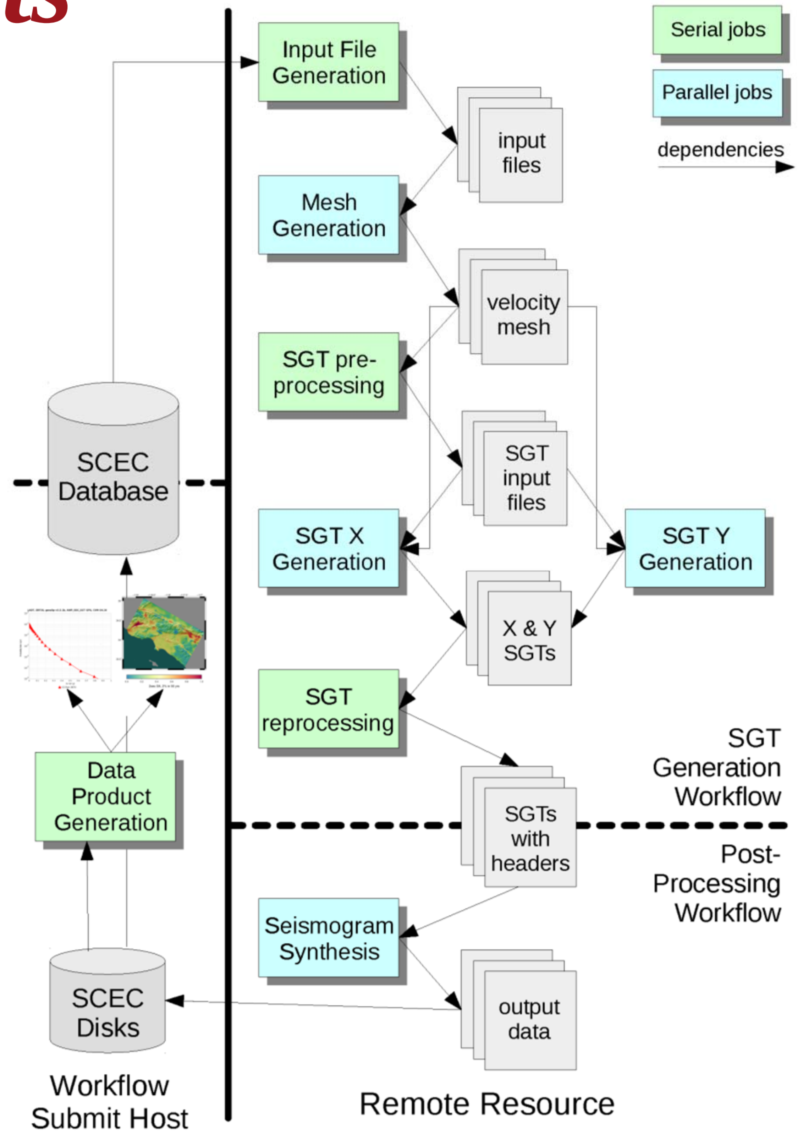
- PSHA is performed by
  1. Assembling a list of earthquakes
  2. Calculating the shaking of each
  3. Combining shaking with probability
- SCEC CyberShake project – simulation-based approach to step 2

Resource Requirements

- Computational requirements per site

Stage	Code Type	Nodes	Walltime
Mesh Creation	MPI CPU	240	0.4 hrs
SGT Generation	MPI GPU	800	1.0
Seismogram Synthesis	MPI CPU	240	10
Other	Sequential	1	0.1 – 2

- 200-400 sites required to complete study
- Parallelism dominated by SGT generation
  - 2 jobs of 800 GPU nodes x ~1 hour
- Cost dominated by seismogram synthesis
  - 240 nodes for ~10 hours



CyberShake workflow schematic

CyberShake Workflows

- Each ‘run’ for a single site consists of a workflow
- Software stack:
  - Pegasus-WMS to create workflow description, plan for execution
  - HTCondor for runtime execution
  - Globus GRAM for communication between workflow host and remote system
- CyberShake study consists of running hundreds of sites
  - Site parallelism of 10-30
- Able to split CyberShake studies to run on multiple resources

Current Solutions for CyberShake

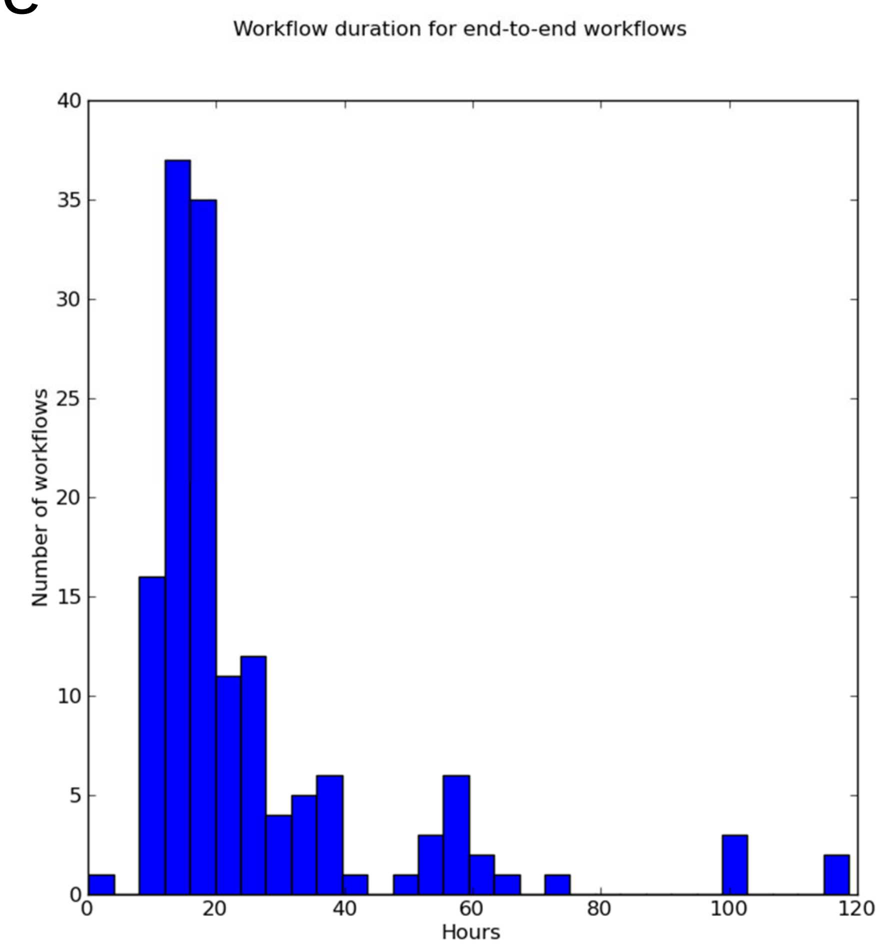
- CyberShake requires 800-node GPU jobs
  - 6.5x faster than CPU version
  - Targeted NCSA Blue Waters and OLCF Titan
- For Blue Waters: push solution works well
  - When jobs appear in HTCondor queue, GRAM submits them over the network to be translated and scheduled in Blue Waters queue
- For Titan: push solution unavailable due to authentication policy
  - Two-factor with fob
  - Proxies only issued for data transfer nodes
  - Decided to try pull-based approach with HTCondor pilot jobs

CyberShake with Pull-Based Approach

- In 2015, used pull solution using HTCondor pilot jobs
- Pilot daemon running on login node queried HTCondor queue on workflow submission host for jobs
- When idle jobs found, pilot job submitted to Titan queue
- Because of variability in task sizes, 4 different sizes of pilot jobs
- Introduced complexity and possibility of errors
- Pull solution resulted in only 68% resource utilization
  - Scheduling overhead
  - Tradeoff between pilot jobs terminating before workflow tasks, and pilot jobs sitting idle

CyberShake with rvGAHP

- In 2017, performed 31-day CyberShake study on Titan using rvGAHP
- rvgahp_server daemon ran on Titan login node
- 13,334 jobs submitted through rvGAHP
  - 10 minutes – 9 hours walltime
  - 1 – 240 nodes wide
  - 450,000 node-hours total
- Increased resource utilization from 68% to 97%
  - Savings of ~130,000 node hours
- Decreased delay per job from 9.2 to 0.6 hrs



Conclusion

- rvGAHP provides new approach for remote resource provisioning
  - Reduced overhead of push-based approaches
  - Runs on systems requiring two-factor authentication
  - Requires little of application developers already using HTCondor
- Very effective for real-world seismic hazard workflow application
- Increases efficiency on current systems
- Opens up new systems for scientific workflows

Thanks!

SCEC AN NSF+USGS CENTER

SDSC SAN DIEGO SUPERCOMPUTER CENTER

U.S. DEPARTMENT OF ENERGY

NCSA

BLUE WATERS SUSTAINED PETASCALE COMPUTING

ISI Information Sciences Institute

USGS science for a changing world

TACC

U.S. DEPARTMENT OF ENERGY INCITE LEADERSHIP COMPUTING

XSEDE Extreme Science and Engineering Discovery Environment

Pegasus

NSF National Science Foundation

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

USCGRID