Data Integrity Use Cases for Community Feedback

Scientific Workflow Integrity for Pegasus Project

April 13th, 2017

Ilya Baldin[3], Ewa Deelman[2], Randy Heiland[1], Anirban Mandal[3],
Steve Myers[1], Mats Rynge[2], Rafael Ferreira da Silva[2], Karan Vahi[2], and Von Welch[1]
[1]Indiana University
[2]ISI/USC
[3]RENCI

Feedback to https://goo.gl/forms/qAo5qpNvz6VFO5R02

About SWIP

https://cacr.iu.edu/projects/swip/

The Scientific Workflow Integrity with Pegasus (SWIP) project improves the security and integrity of scientific data by integrating cryptographic integrity checking and provenance information into the Pegasus workflow management system (https://pegasus.isi.edu/). Complex workflows are commonplace in computational science and engineering applications and Pegasus is a popular WMS used in numerous scientific domains, e.g., astronomy, bioinformatics, earthquake science, gravitational wave physics, ocean science and neuroscience. For example, one project using Pegasus is LIGO (the Laser Interferometer Gravitational-Wave Observatory), which announced in early 2016 the first direct detection of gravitational waves.

The overarching goal of this project is to help enable more trustworthy science. One important facet of that is to ensure data has not been altered - either maliciously or accidentally. Digital signatures are one type of cryptographic support offered in SWIP that can help. By digitally signing data that is run through Pegasus, it is possible to assert the data has not changed. This helps ensure the integrity of the data and verify its provenance, providing greater trust in results. It is expected that solutions implemented for this project will be generic enough to apply to other workflow systems and applications, thereby helping a broad range of research with concerns about data integrity.

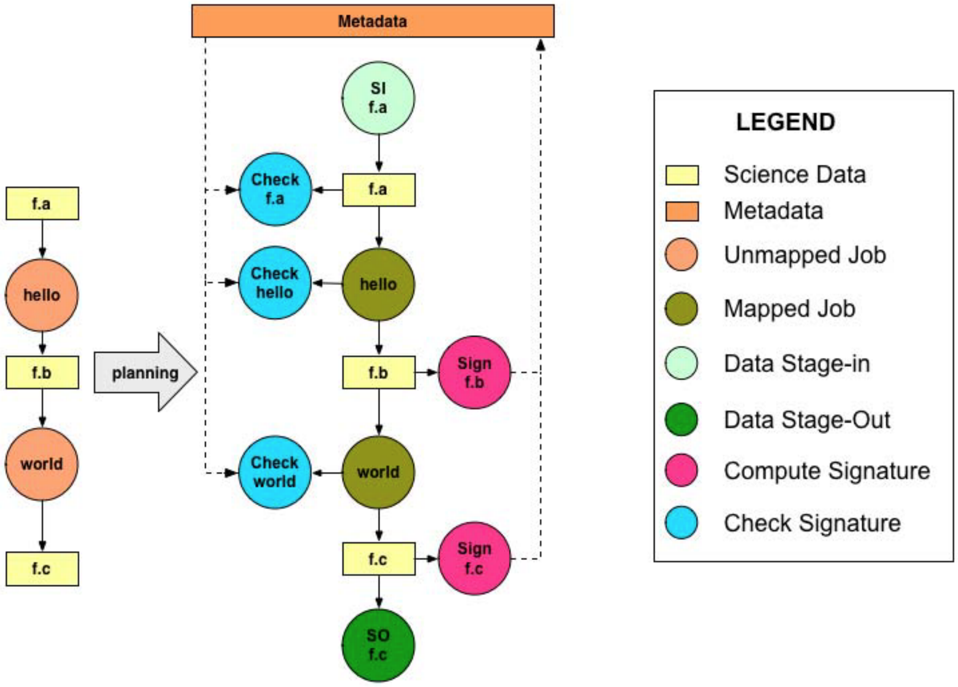
SWIP will be capable of operating across distributed cyberinfrastructure (CI), including dynamically-configurable CI. Pegasus has been integrated with the Open Resource Control

Architecture (ORCA) to add assurances based on Pegasus' dynamic provisioning capabilities for isolation and confidentiality.

The three-year project is funded by a $1 million grant from the National Science Foundation (NSF) as part of its Cybersecurity Innovation for Cyberinfrastructure (CICI) program. Von Welch (IU CACR) is the PI with Co-PIs Dr. Steven Myers (IU SOIC), Dr. Ilya Baldin (RENCI), and Dr. Ewa Deelman (USC/ISI).

The primary goals of the SWIP work are to (1) provide additional assurances that a scientific workflow is not accidentally or maliciously tampered with during its execution, and (2) allow for detection of modification to its data or executables at later dates to facilitate reproducibility. More specifically, in terms of implementation, our goals are to:

1. Provide assurance that any changes to input data, executables, and output data associated with a given workflow can be efficiently detected.
2. Provide provence data from the execution of a given set of tasks that can be compared against future workflow executions to ensure replicability, and provide an audit trail to determine deviation from the workflow in the case of measurement mismatch.
3. Provide assurance that a workflow was executed at a given time, by a given user, and on a given system to the degree of accuracy provided by the underlying systems.
4. Provide assurance that any changes to a workflow specification as communicated to all computational endpoints in a workflow are detected.



SWIP will augment a normal Pegasus workflow (on left) with with data integrity checks (on right).

For more information on SWIP, please see: Ilya Baldin, Ewa Deelman and Von Welch. Scientific Workflow Integrity with Pegasus. IU Booth Talk at SC16, November 2016. https://dx.doi.org/10.6084/m9.figshare.4282499

The SWIP Project is supported by the National Science Foundation under Grant 1642070, 1642053, and 1642090. The views expressed do not necessarily reflect the views of the National Science Foundation or any other organization.

Goal of this Document

This document captures the planning of the SWIP project to-date, showing what use cases we have identified from the community that we believe are important for us to support. We are circulating it in order to solicit feedback on:

1. Additional use cases that the community feels are important.
2. Attributes of our identified use cases that are important to the community - e.g. performance constraints, expected data sizes/number of files involves in a use case, expected behavior on integrity failures.

Feedback is requested by May 5th, 2017. Feedback may be provided via a Google form (https://goo.gl/forms/qAo5qpNvz6VFO5R02), comments to this document, or via email to any or all of the project PIs: Ilya Baldin (ibaldin@renci.org), Ewa Deelman (deelman@isi.edu), Steve Myers (samyers@indiana.edu), Von Welch (vwelch@iu.edu)

Table of Contents

About SWIP 1

Goal of this Document 3

Table of Contents 4

FreeSurfer 5

SPLInter 7

1000Genome Pegasus Workflow using ORCA/ExoGENI 9

SCEC 13

FreeSurfer

FreeSurfer is a software package for the analysis and visualization of structural and functional neuroimaging (human brain) data from cross-sectional or longitudinal studies. The software is developed and supported by the Laboratory for Computational Neuroimaging at the Athinoula A. Martinos Center for Biomedical Imaging at Massachusetts General Hospital. To make this software more widely available to the user community, the University of Chicago provides software to let authorized users upload data to the Open Science Grid and run a Pegasus workflow that provides FreeSurfer analyses.

A typical workflow is shown below (Figure 1). A researcher will upload a compressed MRI data file (about 10-40MB) and submit a job to run a Pegasus workflow that consists of a diamond-shaped directed acyclic graph (DAG). The DAG will perform three basic steps: 1) prepare the input data (autorecon1), 2) normalize the data and generate brain regions' surfaces (autorecon2), 3) identify and label regions (autorecon3). Step 2 can be parallelized across two processes and work on each brain hemisphere independently. A final output, compressed file will be about 200-300MB.

A revised workflow, showing the SWIP enhancements, is shown in Figure 2. Note that the signature creation/checking in steps 1 and 5 (involving the remote user) will be outside of Pegasus's control. The intent would be to have that *FSurf* service[1] incorporate the same, basic solution to signing that is used in SWIP.

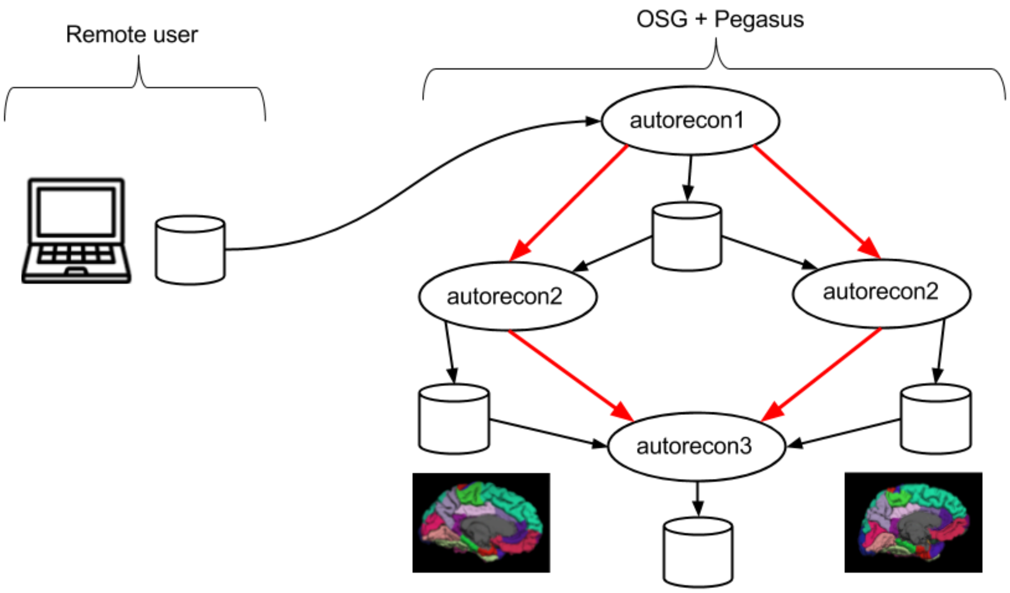


Figure 1. Traditional workflow. Red arrows indicate process flow; black represent data flow.

[1] https://support.opensciencegrid.org/support/solutions/folders/12000002373

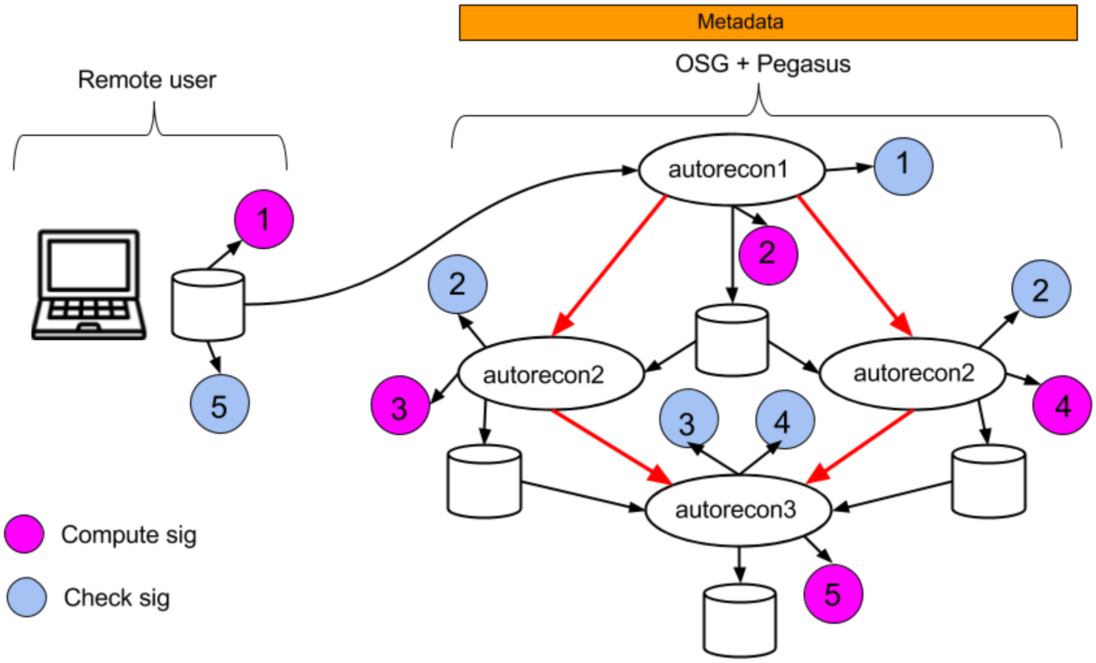


Figure 2. Workflow with SWIP enhancements. Note that there is an implied data flow between steps 2-4 and the Metadata.

SPLInter

The science of SPLInter is well described in the ScienceNode article “Virtual screening powered by high-throughput computing” (https://sciencenode.org/feature/virtual-screening-powered-high-throughput-computing.php):

> *Meroueh's laboratory has created SPLInter (Structural Protein Ligand Interactome), an online interactome that predicts the interactions of thousands of small organic molecules with thousands of proteins, through structure-based molecular docking and scoring. The site also ranks proteins for active compounds against individual targets. Users can also identify potential off-targets of a compound of interest for further experimental validation in cells or in vivo.*
>
> *"In SPLInter, we're doing the same thing we would traditionally do in rational drug design, but doing it at the cellular level. Instead of one focused target, we screen for all proteins of the human proteome whose structure has been solved by x-ray crystallography or NMR. We take a protein, find its structure, and through its structure we do virtual screening for all the targets in the cell."*
>
> *"Proteins evolve to adopt a specific shape. That shape and how the protein associates with other things are critical. If you want a drug to bind to a protein it has to fit into a cavity; the small molecules like pockets." explains Meroueh.*

The workflow is a large bag of independent tasks trying to do the docking. A typical workflows starts with around 100 pockets and 50,000 ligands.

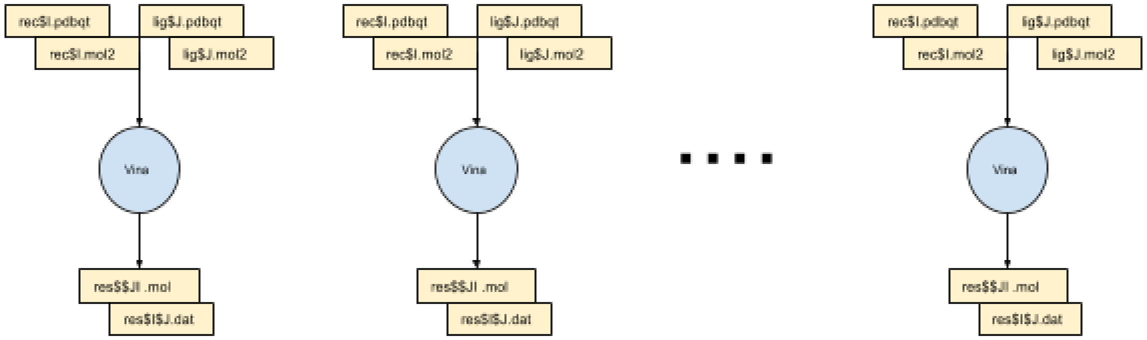
Each task takes 4 main input files, 2 files for receptor description (~500KB mol2 file and ~500KB pdbqt file) and and 2 files for ligand description (~5KB mol2 file and ~5KB pdbqt file). There are a few other input files such as the binary and sometimes config files. Each task outputs 2 files describing the docking, a mol2 file ~5KB mol2 file and ~0.5KB dat.

What makes SPLInter an interesting use case is the number of tasks (~5M) and that the workflow is running the distributed environment provided by OSG. That means that each workflow has millions of data transfers (can very depending on task clustering).

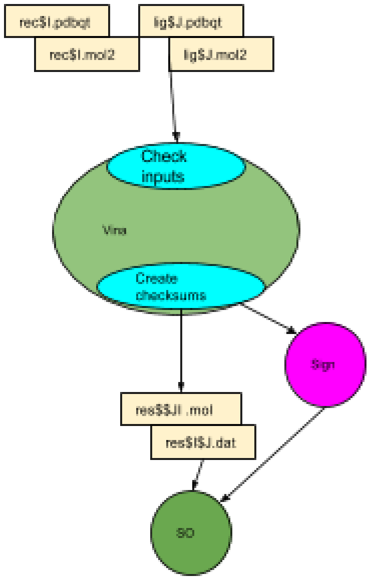
SWIP Enhancements for SPLInter

Due to the distributed execution environment in this case, it will be important to be able to verify the inputs as close to the execution as possible, preferably as a part of the job itself. Similarly, we want to generate potential checksums as part of the job, before any transfers takes place.

Abstract:



Planned with SWIP enhancements:



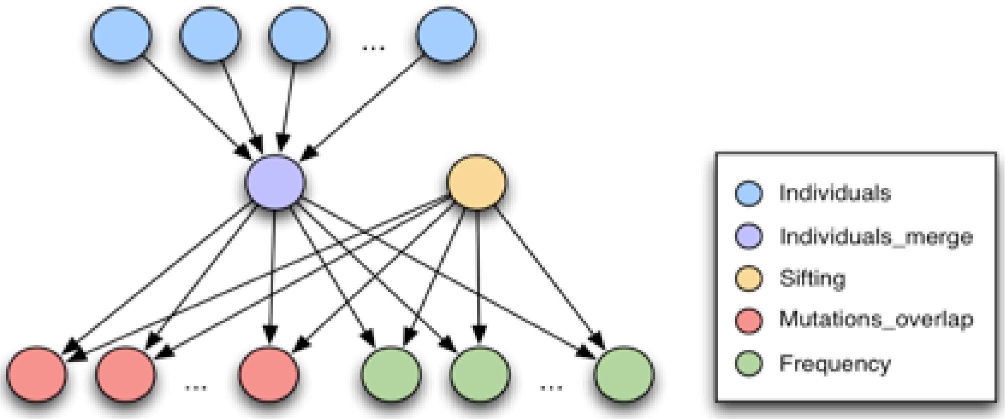
Also consider the amount of metadata entries for this workflow. If we only keep one checksum and one signature, we will end up with 20M entries in the meta data catalog (5M tasks * 2 output files * 2 entries per file)

1000Genome Pegasus Workflow using ORCA/ExoGENI

1000 Genome Workflow and Infrastructure Description

The 1000 genomes project provides a reference for human variation, having reconstructed the genomes of 2,504 individuals across 26 different populations to energize these approaches. This workflow identifies mutational overlaps using data from the 1000 genomes project in order to provide a null distribution for rigorous statistical evaluation of potential disease-related mutations. The workflow fetches, parses, and analyzes data from the 1000 genomes project (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/release/). It cross-matches the extracted data (which person has which mutations), with the mutation's sift score (how bad it is). Then it performs a few analyses, including plotting. The workflow and description is available at https://github.com/pegasus-isi/1000genome-workflow .

The figure below shows a branch of the workflow for the analysis of a single chromosome.



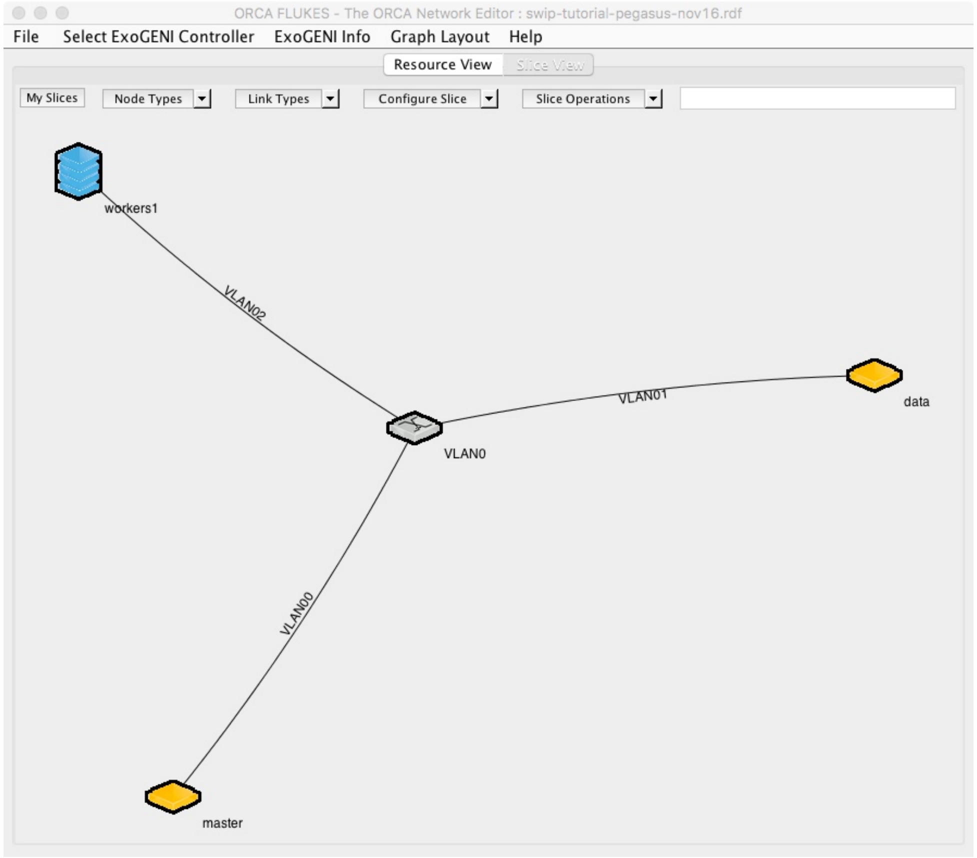
Individuals: This task fetches and parses the Phase3 data from the 1000 genomes project by chromosome. These files list all of Single Nucleotide Polymorphisms (SNPs) variants in that chromosome and which individuals have each one. An individual task creates output files for each individual of ‘rs numbers 3’, where individuals have mutations on both alleles. The *Individuals_merge* task collates the outputs of the *Individuals* tasks.

Sifting: This task computes the SIFT scores of all of the SNPs variants, as computed by the Variant Effect Predictor (VEP). For each chromosome, the sifting task obtains data from 1000 genome project, processes the corresponding VEP, and selects only the SNPs variants that has a SIFT score, recording in a file (per chromosome) the SIFT score and the SNPs variants ids.

Mutations_overlap: This task measures the overlap in mutations (also called SNPs variants) among pairs of individuals by population and by chromosome.

Frequency: This tasks measures the frequency of overlapping in mutations by selecting a number of random individuals, and selecting all SNPs variants without taking into account their SIFT scores.

For the version of the workflow running on ExoGENI, we limited the workflow to the execution of 2 populations. It included 22 *Individuals* tasks, 2 *Sifting* tasks, 14 *Mutations_overlap* tasks, and 14 *Frequency* tasks. The figure below shows the ExoGENI infrastructure used to run the workflow.



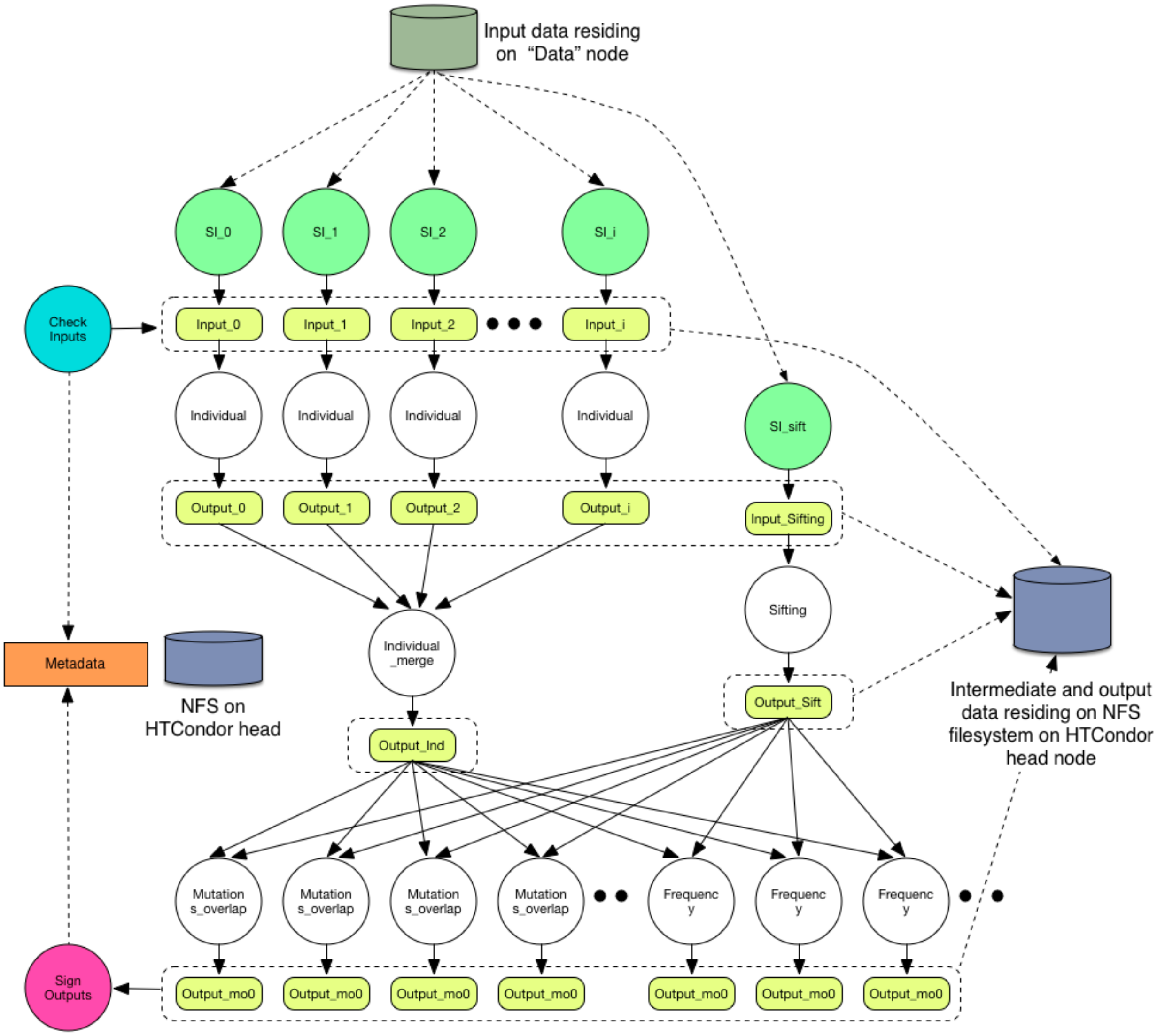
The infrastructure consists of

- 1 ‘master’ VM that runs the HTCondor master, NFS server, and Pegasus software.

- A set of VMs called ‘workers1’, each of which is a HTcondor worker. This request has 4 HTCondor worker VMs, each with 4 cores. That makes this setup a HTCondor pool with 16 cores. The HTCondor workers and the HTCondor master share the NFS filesystem.
- 1 ‘data’ VM that hosts the input data from 1000 genomes project required by the workflow. The *Individuals* and *Sifting* jobs obtain the data using http from a http server running on the ‘data’ VM. The ‘data’ VM does not share the above NFS filesystem.
- VLAN0 represents a broadcast network used by all the VMs. A bandwidth is set on VLAN0 and can be changed as desired.

SWIP Enhancements for 1000 Genome Workflow

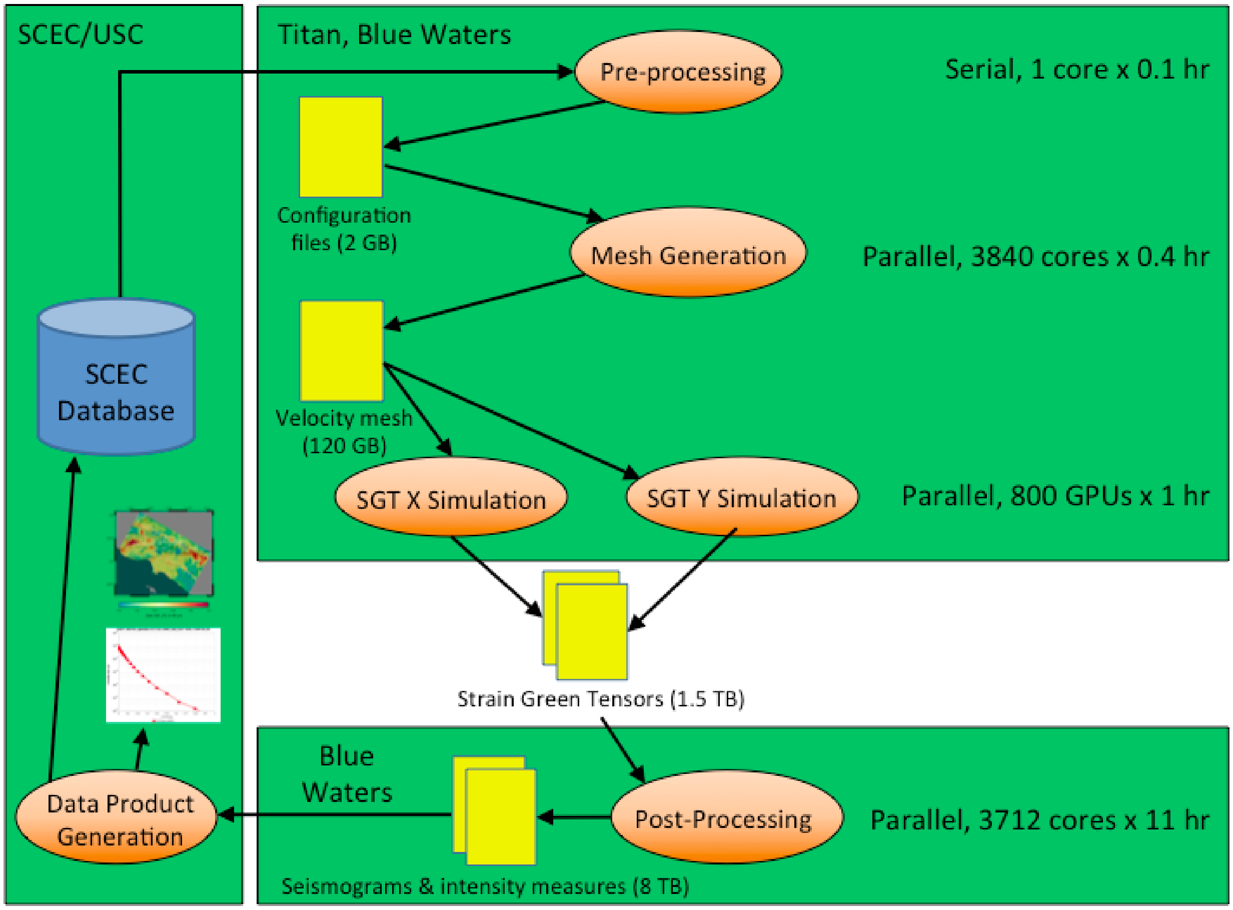
The following figure captures proposed enhancements for the 1000 Genome workflow to ensure data integrity. The stage-in jobs pull input data from the “Data” node. This data might already have some existing MD5 checksums, as per the quality control protocols of the 1000 Genomes project (http://www.nature.com/nmeth/journal/v9/n5/extref/nmeth.1974-S1.pdf (p7)). The “Check Inputs” integrity check can leverage that. We might also sign the outputs of the “Mutations_ovelap” and “Frequency” tasks. The metadata corresponding to integrity checking will be stored on the NFS server on HTCondor head node running Pegasus.



SCEC

The Southern California Earthquake Center (SCEC) uses Pegasus-WMS extensively to manage the execution of CyberShake, a physics-based probabilistic seismic hazard analysis (PSHA) application. PSHA quantifies the peak ground motions from all possible earthquakes, which might affect geographic locations in Southern California and establishes probabilities that these locations will experience a given level of ground motion over a time interval. Through the CyberShake project, SCEC has developed simulation-based PSHA that captures the physics of earthquakes more effectively than alternative approaches not explicitly based on physical models. PSHA results more closely model the real world and have the potential to lead to more resilient design of structures and infrastructure networks, reducing seismic risk and improving community resilience.

To provide broad impact to users of CyberShake data, such as seismologists, utility companies, and building code engineers, SCEC performs CyberShake studies, last of which was ran from Dec 2015 - April 2016, to calculate PSHA estimates of improved accuracy. The calculations are distributed between the NSF Track 1 system NCSA Blue Waters, the DOE Leadership-class system OLCF Titan, and USC’s center for High Performance Computing, using Pegasus-WMS.



Most of the big codes in the above workflow: mesh generation, SGTs, and post-processing are MPI. There are other, smaller, codes which we run which are serial, like writing input files and performing checksums

The work was split up so that 150 TB of intermediate data was staged from Titan to Blue Waters for post-processing, which was automatically handled as part of the Pegasus workflows. This study was 16 times as large as previous CyberShake studies, and used 1.1 million node-hours over 5 weeks of wallclock time. On average, 54 Pegasus workflows ran concurrently, on an average of 1962 nodes across the three systems. At peak, CyberShake calculations were running on 20% of Blue Waters and 80% of Titan, running both GPU and CPU jobs. Pegasus managed over a petabyte of data, of which 8 TB was automatically staged back to SCEC storage as part of the workflows.

Study 15.4 produced an urban seismic hazard map for Los Angeles at a seismic frequency of 1 Hz, twice what was possible previously, and a goal that the SCEC computational team had been working towards for several years. Following on this success, SCEC began Study 15.12, which combines the deterministic physics-based results from Study 15.4 with stochastic high-frequency seismograms produced using software from the SCEC Broadband Platform.

Current Integrity Checks in the workflow

SCEC currently adds MD5 checksum jobs to specifically check the SGT files. Since the files are fairly large, the check jobs take a long time to execute. In order, to do the production runs as fast as possible, SCEC does not put these checksum jobs in the critical path of the workflow. Instead they are a separate branch in the workflow, and on failure are triggered to fail the complete workflow. To achieve this they rely on a DAGMan feature called ABORT-DAG-ON , that triggers a failure for the whole workflow if a particular node fails.