Accelerating Data-Intensive Seismic Research Through Parallel Workflow Optimization and Federated CI

M. Adair, I. Rodero, M. Parashar, D. Melgar

Research Question

- How can high throughput computing from federated cyberinfrastructure be leveraged and optimized to accelerate and streamline FakeQuakes (FQs) earthquakes simulations?

Background

- Synthetic data from simulated large earthquakes (MW 7.5+) is valuable for training AI-based **earthquakes early warning (EEW)** models to predict earthquake magnitudes
- **MudPy,** a Python-based simulation framework, can be used for simulating the necessary data
- **Open Science Grid (OSG)** provides users with fair access to processing & storage capacity contributed by university campuses, government-supported supercomputing institutions, and more

Motivation

- The MudPy earthquake simulation framework has limitations: **(1)** lengthy simulations, **(2)** lack of user-friendliness, and **(3)** no current platform for discovery and sharing generated data

Implementation

- Leveraged OSG to create a parallel workflow for enhancing the speed and user experience of FakeQuakes simulations: **FakeQuakes DAGMan Workflow (FDW)**
- Utilized HTCondor DAGMan workflows to automate the steps of FQs
- Containerized MudPy and its dependencies with Singularity to deploy across OSG nodes
- Developed a system to monitor the progress of running and completed DAGMans,
- Created system for remotely launching DAGMans simultaneously
- Implemented a VDC bursting simulation framework in Python to simulate execution times and associated costs of offloading OSG jobs to the Cloud
- Leveraged submission, execution, and termination times from real OSG workloads
- Developed OSG-tailored job bursting policies to address low throughput, congested queues, and gaps in job submissions

Experimental Evaluation

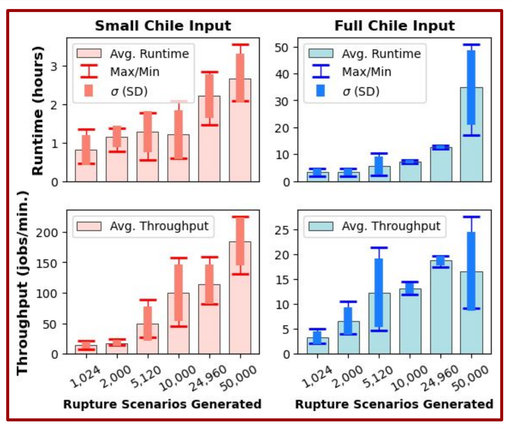
1. Increasing Earthquakes Simulation Quantities

2. Concurrent HTCondor DAGMans

3. Simulated VDC Bursting

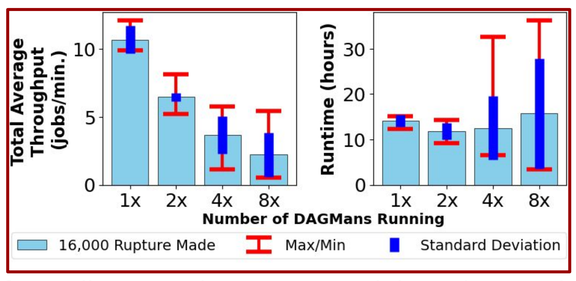
Experimental Results

1.

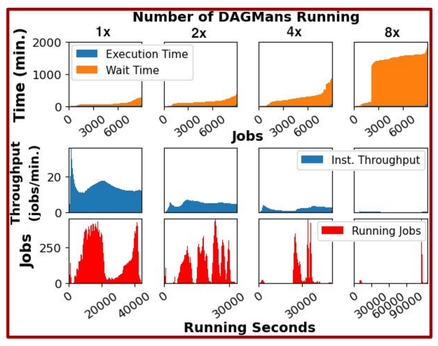


These graphs illustrate the average total runtime and throughput of the FDW simulating varying amounts of earthquake scenarios using two different-sized input lists (2 and 121 stations).

2.

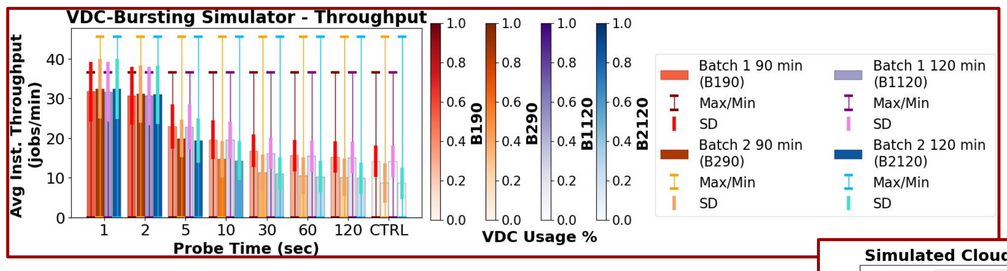


This figure illustrates the average total throughput and runtime of DAGMan workflows that ran parallel to create 16,000 ruptures using the full Chilean input.



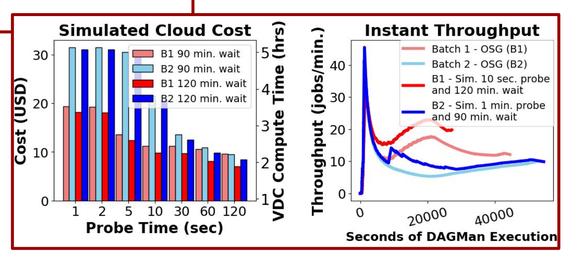
These graphs showcase examples of individual jobs' execution and wait times (sorted by duration) and instant throughput and running job count (for every running second of individual workflows) in various numbers of concurrently running DAGMans.

3.



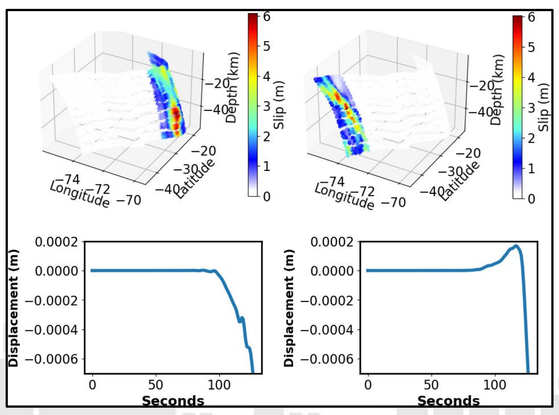
The above graph illustrates the average instant throughputs and VDC utilization percentages while simulating supplemental job bursting for OSG jobs. We investigated two policies: the first evaluated against a throughput threshold with varying probe times for bursting, and the second examined different maximum allowed queue times until VDC bursting jobs.

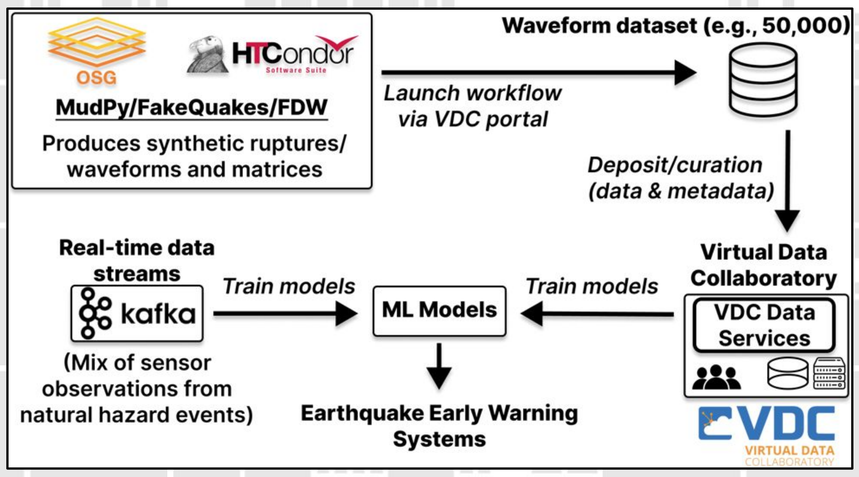
The below graphs showcase the simulated cost (left) for supplemental VDC bursting on two real OSG DAGMans examples and their instant throughput over time (right).



Discussion

- FDW significantly reduced execution tme and increased throughput for generating earthquake waveforms compared to running FQs sequentially on a single host
- Experimentation revealed that OSG performs best with a single running DAGMan
- The bursting simulator demonstrated varying performance, leaving room for improvement
- Future work includes experimenting with regions beyond Chile, helping researchers create new datasets/leverage existing ones, explore and expand on the cloudbursting model





THE UNIVERSITY OF UTAH

NSF

SCI www.sci.utah.edu