Creating A Tier-3 Compute Cluster leveraging Amazon Web Services Infrastructure

SUPER Grant Recipient
William Barden
California State University, Fresno

Premise

- **Existing Compute Clusters**
  - Large Hardware Investment
    - Hardware exists as a ‘snapshot’
    - Requires large amount of real estate
    - Ongoing maintenance costs
    - Difficult to upgrade
- **Cloud Infrastructure**
  - Modular
  - Dynamically Scalable
  - Virtualization makes hardware upgrades trivial

Amazon Web Services

- **Flexibility**
  - Provides the ability to create virtual machines and networks
    - Virtual Machine instances (‘EC2’s) can be networked
    - Creation of a ‘bastion’ or gateway for security purposes
    - Machines and processing cores can be spun up on demand
    - Deploying EC2s is trivial
- **Cost and Labor Reductions**
  - Amazon handles Layers 1 and 2
  - Current project implementation implies a “pay as you go” model
  - No up-front hardware investment.

EC2: An Introduction

- **Amazon’s Virtual Machine Service**
  - Each Virtual Machine is referred to as an EC2 instance
  - Can be instantiated relatively quickly ~5 minutes
  - Supports most major operating systems and Linux Distributions
  - Can be configured as remote CLI workstations or as servers depending on choice of operating system

Project Goals

- **Develop and Deploy a Working Tier-3 Compute Cluster in a virtualized environment for us by US ATLAS Group members**
  - Replicate LXPlus Functionality
  - Accept and complete compute jobs
  - Provide Robust computational options for various research institutions
  - Implment CERN’s Virtual Machine File System (CVMFS)
    - Configure to allow for setupATLAS and lsetup root commands from the GRID
  - Must be a viable alternative to in-house compute clusters from both a financial and workflow perspective

Project Progress and Evolution

- **Settle on an Operating System**
  - Initial testing done on Ubuntu and Debian-based working environments
  - Rapidly pivoted to CentOS to better mesh with existing documentation for CVMFS, GRID
  - Utilize existing free and Open-Source Amazon Machine Image (AMI) for CentOS7
  - Developed and Documented deployment of CVMFS and related dependencies

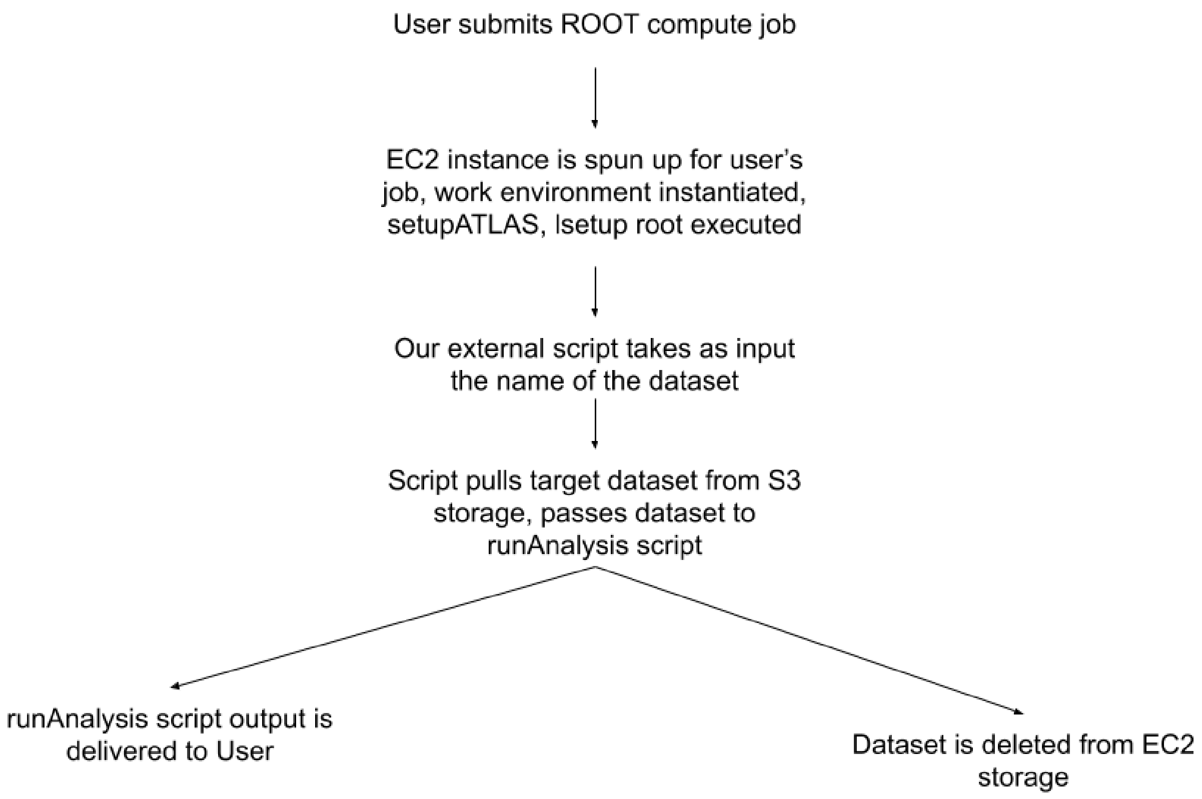
COVID-19 Impact

- Project Suffered minimal interruption (~two to three weeks) due to to Coronavirus
- Campus closure forced work to continue at home
- Work continues after adjustment, project team able to communicate via email, slack, and video conferencing.

Storage Needs

- **Data Sets**
  - Vary wildly in size, will need to be able to handle large (100 GB to 1TB) BLOBs of Data
  - Multiple researchers will be working off of the same dataset, does not make sense to force users to provide their own dataset
  - Resolve to utilize Amazon's S3 storage service to create a centralized repository for available datasets
  - Implement usage of S3 'bucket' and modified ROOT scripts to call S3 objects into EC2

CVMFS → ROOT→ S3 Workflow



User submits ROOT compute job

EC2 instance is spun up for user’s job, work environment instantiated, setupATLAS, lsetup root executed

Our external script takes as input the name of the dataset

Script pulls target dataset from S3 storage, passes dataset to runAnalysis script

runAnalysis script output is delivered to User

Dataset is deleted from EC2 storage

Dependency Issues and CentOS7

- In order to implement automated usage of S3 storage solutions, additional packages are required
- The Dependencies for these packages, including gcc are either woefully out of date or non-existent in CentOS7 repositories
- Accordingly, we have now shifted focus to CentOS8, with minimal friction.

Documentation

- **All Project work is being Documented**
  - Rapidity of EC2 deployment allows for quick and easy testing of virtual machines as testbeds
  - Virtualization allows for rapid replication of both blockers and solutions.

Continuing Work

- **Implement AWS-CLI and AWS SDK to automate S3 storage dataset utilization**
- **Other team members are working on implementation of other systems incuding:**
  - HTCondor
  - Pandas
  - Virtual Private Cloud Infrastructure
  - Web Interface for job submission

Accomplishments Thus Far

- The Project has implemented a secure Virtual Private Cloud with a bastion on AWS Infrastructure
- Deployed CentOS based EC2 instances
- Deployed CVMFS/GRID implementation
- SetupATLAS and lsetup root commands working
- Able to execute ROOT commands and analyses on AWS virtual machines
- Implemented S3 “bucket” storage for holding datasets

Thanks to:

- US ATLAS Group for their ongoing support and the SUPER Grant, making this project possible
- CERN
- All support staff
- Fresno State's Technology Services Department
- Our friends at AWS
- Professors Harinder Bawa and Yongsheng Gao for their guidance and support