Travis Miller
Kristi Nikolla
Rob Reyes
Xingyu Wu

Radiology on the Cloud

Project Description Outline

Vision and Goals of the Project

Radiology in the Cloud will use software called ChRIS to provide Radiologists a means to do cutting edge image processing on MRI and other medical scans. The image processing required is computationally intensive and therefore requires a cluster of computers to complete in a convenient amount of time. With the current implementation of ChRIS, the software is customized specifically for the Boston Children's Hospital computer cluster. This makes it difficult for ChRIS to be modular and be used by radiologists elsewhere. The aim for this project is to make use of the OpenStack API and build an interface for the ChRIS software to provision hardware from an Open Cloud and set up the nodes to do the required computations. This will require:

- using the OpenStack hardware to provision the required hardware easily and with little setup from the user
- installing the appropriate OS and uploading the required Python scripts and data
- setting up a remote scheduler to handle the processing the jobs as they come in.

Users/Persona of the Project

Radiology in the Cloud will be used by radiologists using the ChRIS software to convert data and process medical scans. It will also used by the network administrators of the hospital who will be setting up the Cloud Computing network. Because doctors are using the software and have little time to set it up, it must be a quick and easy interface to the cloud.

Radiology Cloud does not target patients whose data the software will be processing. It is only up to the doctor to take patient data and process it.

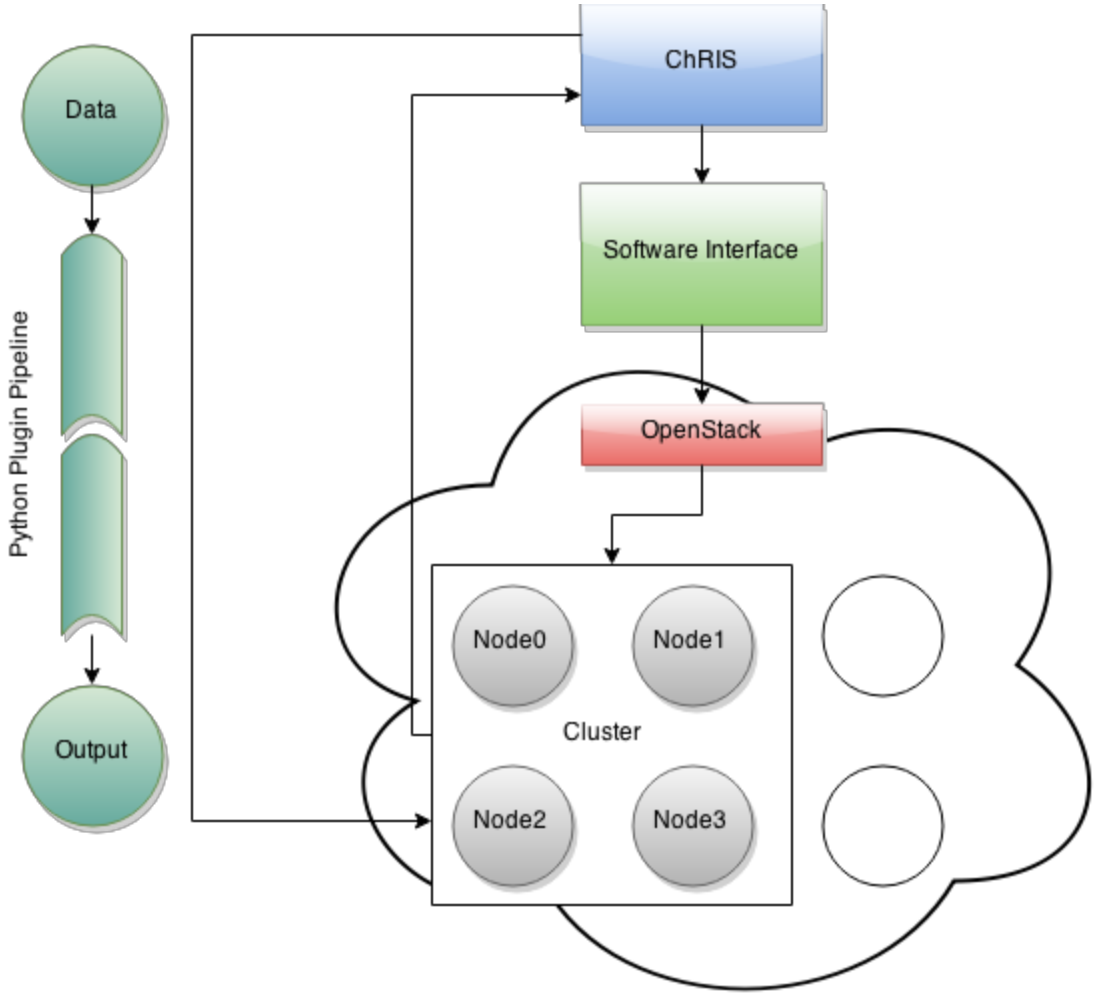
Scope and Features of the Project

- Create a software interface between ChRIS and OpenStack using the OpenStack SDK
    - software interface will configure the nodes, using the appropriate disk images as required by ChRIS.
    - software interface will configure the network of the nodes, ensuring they are correctly connected and can see each other.
    - software interface will configure the cluster of nodes, installing the appropriate scheduler.
    - software interface will give access to the cluster to ChRIS, who will use it to perform computation on the data it will provide through its plugins, which may be either in Python or C/C++.
- Create a disk image from which to provision the nodes:
    - Based on Ubuntu 14.10 server 64-bit.
    - Containing all the required dependencies for the Python and C/C++ to run without problems.

Solution Concept

We will make use of the Python SDK of OpenStack to develop a software interface that connects ChRIS to OpenStack.

- Make use of the OpenStack API to build a software interface using Python, between ChRIS and OpenStack.
- The software interface will instruct OpenStack to launch the virtual machines.
- The software interface will instruct OpenStack to create a networked cluster with these virtual machines, which we will call nodes.
- Nodes will be running Ubuntu 14.10 Server 64-bit, so part of our work will be to create and pre-configure the required images which OpenStack will later instantiate.
- The cluster information will be returned to ChRIS by the software interface.
- ChRIS will use the cluster of nodes to execute the plugin pipeline to convert the data into the required output.



Acceptance Criteria

At the minimum there should be a script which can be called from the ChRIS software to form a cluster of nodes running the desired OS of Ubuntu 14.10.

High level goals:

- include a remote scheduler for the requested processing jobs that will run on the nodes
- upload the python scripts and data from ChRIS and process medical data

Release Planning

Release #1 (Week 5):

Become more familiar with the OpenStack and its API
Setup a simple OpenStack disk image.
Create Python scripts which provisions a virtual machine using a disk image of Ubuntu 14.10.

Release #2 (Week 7):

Release #3 (Week 9):

Release #4 (Week 11):

Release #5 (Week 13):

Setup cluster and scheduler for ChRIS.