Hybrid Cloud Services Project Proposal

Mentors:

- Xu (Simon) Chen:Xu.Chen@twosigma.com

Members:

- Akash Singh: singh.aka@husky.neu.edu
- Jaison Babu: babu.j@husky.neu.edu
- Surekha Jadhwani: jadhwani.s@husky.neu.edu
- Vignesh Shanmuganathan: shanmuganathan.v@husky.neu.edu

1. Vision and Goals of the Project:

Hybrid cloud model has a foreseeable future, because two types of clouds offer drastically different properties that offers combination of benefits from private and public cloud (security, performance, bursting capability, features, etc.)

Hybrid model is hard to build and manage since it involves lot of factors such as network connectivity, security, user interaction, etc. The aim of this hybrid cloud services to lay foundation for creating a completely seamless world that allows developers just focus on building application logic, and the infrastructure layer would automatically handle scaling, placing and managing application and its different components on either or both clouds.

The primary goal of this project would be to create a simple scaled down version of hybrid cloud by setting up an private cloud hosting sensitive data in cloud such as AWS (Amazon web services) and a public cloud hosting non-sensitive data in cloud such as MOC (Massachusetts Open Cloud) and establish communication between these two cloud Hadoop clusters allowing to run the analytics either on private cloud or public cloud or on both based on the performance and data considerations.

Our system will enable tagging of data by the user and hence user needn't be worried about the transfer of secure data to a public cloud later. The data will be segregated such that the secure data would reside in the private cloud and unsecure data would be transferred to the public cloud without user knowledge. The processing of data as mentioned previously on either clouds work consider the complete data set through implementation mechanism discussed later in this report. We will even produce benchmark analysis which would be useful to the user in making appropriate decisions.

Some configuration policies will be set for the user so that the user would be able to provide basic configurations (e.g.: Secure data tags) for the system. The final system will hence be supporting a hybrid cloud model by making the use of public and private clouds as per the needs.

Users/Personas of the Project:

The project is aimed at the following audience:

a. Companies with private cloud wishing to incorporate public cloud infrastructure without compromising data security.
b. Software developers who can make use of the combined public and private cloud.
c. Researchers of hybrid cloud computing model.

2. Scope and Features of the Project:

Scope of this project includes:

a. Creating a public cloud in MOC that contains non-sensitive data.
b. Creating a private cloud in AWS that contains sensitive data.
c. Set-up of Hadoop clusters in AWS and MOC.
d. Establishing communication between AWS and MOC for data transfers.
e. A spark streaming service running on AWS will gather real time data from twitter and partition data according to user configuration for classifying data as sensitive/non-sensitive. The proposed data classifiers include hash tag mechanism and linguistics.
f. Based on the tags, linguistics, the created classifier will distribute data across AWS and MOC.
g. Running jobs in the following setups:
    i. Running data analysis job on public cloud spanning the complete dataset (sensitive and non-sensitive data).
    ii. Running data analysis job on private cloud spanning the complete dataset.
    iii. Running data analysis job on non-sensitive data on public cloud and on sensitive data on private cloud, in parallel.
h. Performing benchmark analysis on the above proposed configurations.

Features of this project are:

a. Easy user configuration and abstraction of data segregation from the user.
b. Establishing communication between the cloud clusters.
c. Ability to run data analysis jobs on either/both clouds.
d. Analyzing real world data (twitter feed) on real world load on the hybrid setup.
e. Configuring the setup to sustain load of such load without major failures spanning both clouds.

Stretch Goals:

a. Adding an auto-scaling mechanism that can be enabled by the user as a part of configuration. If true, for high amount of loads, the job will be run on the public cloud cluster automatically in an abstracted manner to the user.
b. Adding a security layer by using one of the mechanisms like VPN, Firewalls, etc.
c. Providing a UI to the user for easing out the configuration process.

3. Solution concept/architecture:

We will use existing APIs to consume live twitter feed. We will develop a parser to segregate the data into different locations – sensitive on local and non-sensitive on public cloud. The parser will be loaded in the private cloud and it will store the sensitive data and route the non-sensitive data to public data. The data has been separated using the tag mechanism, linguistics defined by the user.

The clusters are connected in a network; hence the transfer to public cloud is possible. We will run our data analytics jobs using Spark that will span both clouds thus giving the hybrid cloud features. In auto-scaling mechanism, the cloud will make decisions on whether to run the job on the public cluster or private cluster. A log of files of twitter data will be stored differentiating the loads and the developed program will take these things into considerations to give auto scaling mechanism of hybrid cloud.

The following architecture is proposed:

Hybrid Cloud Service Architecture:

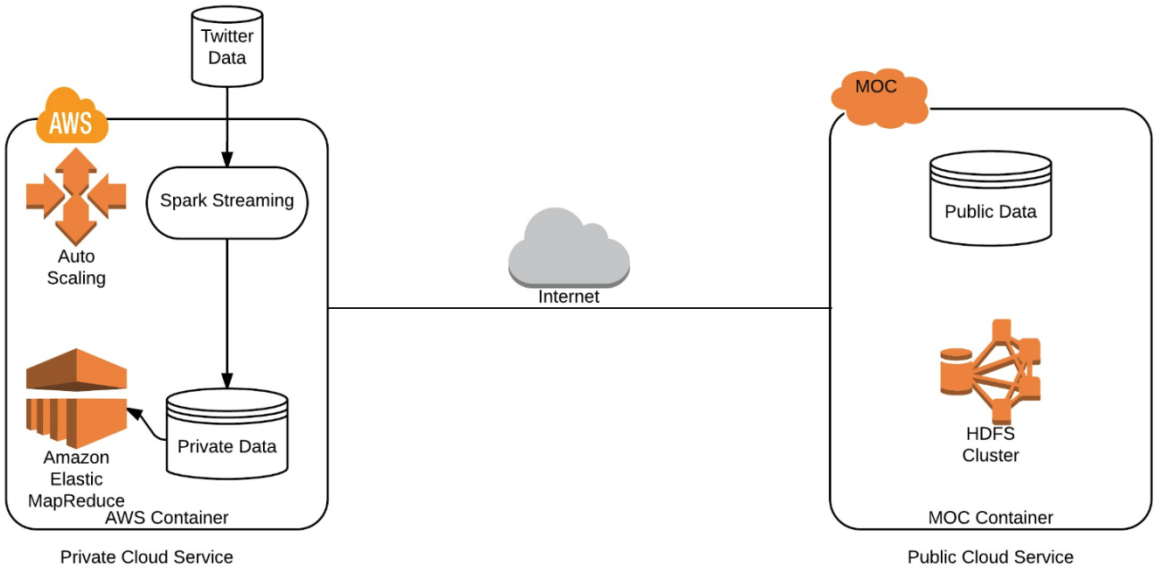

Figure 1: Hybrid cloud architecture

4. Design implications:

Our implementation will make use of the following technology:

- Java, Bash scripting, XML, Spark
- Amazon AWS, EC2, S3, EMR
- MOC, OpenStack APIs
- Twitter Feed APIs, IP, TCP and others as and when needed for integration into the software.

The program will be Java based. The main considerations is that we are dealing with are data segregation, running of data analysis on one cloud which will span both clouds, handling real world data load, scaling mechanism.

5. Acceptance Criteria:

- Segregation of data on the basis of tag and linguistics mechanism.
- Running of parallel processing Spark jobs on both clusters.
- Performing bench mark analysis of analytics jobs on different clouds.
- Giving easy user configurations and abstracting as much information from user as possible.

6. Release Planning:

We are doing bi-weekly Sprint planning with weekly updates to the mentors.

Release #1: (Feb 12, 2016):

i. Study Hadoop on AWS cluster and other technologies to be used in the project.
ii. Set-up Hadoop cluster on AWS.
iii. A functioning parser which could segregate data as sensitive or non-sensitive on a local machine.

Release #2: (Feb 23, 2016):

i. Set-up Hadoop cluster on MOC.
ii. Establish communication between MOC cluster and AWS cluster.
iii. Setup the parser running such that it can send data to AWS cluster. This would use the communication channel to transmit file splits to AWS cluster.

Release #3: (Mar 15, 2016):

i. Running Spark job on twitter feed on public cluster. Job could be as basic as counting tweets of different hashtags. (Running one job successfully will pave way for running other parallel jobs on clusters)
ii. Running Spark job on twitter feed on private cluster.

Release #4: (Mar 29, 2016):

i. Running Spark job in parallel on both clouds on the live twitter feed.
ii. Exploring ideas and implementing them to gain advantage of both clouds. (One thought idea: Running Map jobs on both clouds which will output the result of map job as is. Transfer Map output on public cloud to private cloud. Run a Map job which will read the map outputs of public and private cloud and run the final reduce job)
iii. Explore configuration enhancements and basic bug fixes.

Release #5: (Apr 12, 2016):

i. Perform benchmarking of different setups for analysis.
ii. Bug fixing and deployment.
iii. Exploring auto-scaling mechanism and adding one auto-scaling feature.
iv. Configuration UI for the user.

Final Demo: (Apr 26, 2016):

i. Final deployment, documentation and release management.
ii. Video demo of working model of the proposed software.