Assignment 2: Virtual resources management in OpenNebula

Prerequisites:

Familiarity with Unix command line

Assignment 2.1: VM Contextualization

Objectives:

In the lectures you have been presented with Cloud Computing concepts. To help you better understand and get a hands–on experience with these concepts, we have defined a simple set of assignments in which you will have to familiarize yourself with OpenNebula an open-source cloud computing toolkit for managing heterogeneous distributed data center infrastructures. You will learn how to use OpenNebula API and understand how your applications can benefit from cloud computing.

Background:

There are two aspects of OpenNebula that need to be understood:

OpenNebula API

> The OpenNebula API allows you to manipulate virtual resources. You will learn how to use this API via a simple pre-assignment. The latter consists in creating one VM image description and one virtual network description.

VM Contextualization

> OpenNebula offers contextualization hooks. These are similar to hooks used by other (commercial) cloud middleware, e.g. Amazon EC2 API. You will learn how to use the contextualization mechanism via a simple pre-assignment.

TODO:

1. **Prepare** the Lab by following the tutorials [1][2]. We advise you to follow the tutorials and download and configure all the software needed for the lab.
2. **Contextualize a VM** such that it automatically deploys the URL Shortner Web service (developed for Assignment 1.1) at start up. The service should be available (visible) outside the VM.
3. **How to get your assignment GRADED:**

a. Show a working prototype to the Lab assistant

b. **Submit via Blackboard a** tar file containing

   - The VM template (which should include the contextualization)
   - The network template
   - A short report describing your experience with ONE API, as well as how you contextualized the VM (at most 1 A4 page)
   - The tar file should be named: <group-number>_VM_context_2_1.tar

Assignment 2.2: Big data analytics [optional + Bonus]

Objectives:

In the bonus assignment you will learn how to create a small cluster using VMs in which you will deploy Apache spark to perform some simple data analytics. This will help you to understand the idea behind Big Data platforms/frameworks

Background:

Apache Hadoop

Apache Hadoop[3] is a collection of open-source software utilities that facilitate using a network of many computers to solve problems involving massive amounts of data and computation. It provides a software framework for distributed storage and processing of big data using the MapReduce programming model.

Apache Spark

Apache Spark[4] is an open source big data processing framework built around speed, ease of use, and sophisticated analytics. It was originally developed in 2009 in UC Berkeley's AMPLab, and open sourced in 2010 as an Apache project.

We are going write a simple Spark application to compute the PageRank[5] for a number websites in a small cluster. PageRank is an algorithm used by Google Search to rank websites in their search engine results.

TODO

1. **Prepare** a small cluster with at least 3 VMs. One is the master, and the others are slaves [6][7].

2. **Install Hadoop Distributed File System** (HDFS) in your cluster [8]. Please make sure it is working properly.

3. **Install Spark** to your cluster [7]. At this point, ensure that the relevant Spark and HDFS daemons are running.

4. **Write a spark application** to implement PageRank Algorithm. PageRank is an algorithm that is used by Google Search to rank websites in their search engine results. This algorithm iteratively updates a rank for each document by adding up contributions from documents that link to it. The algorithm can be summarized in the following steps –

   1) Start each page at a rank of 1.
   2) On each iteration, have page p contribute $rank(p)/|neighbors(p)|$ to its neighbors.
   3) Set each page's rank to 0.15 + 0.85 X contributions.
   For detailed information regarding PageRank please read [5].

5. For the purpose of this assignment, we will be using the Berkeley-Stanford web graph [9] and execute the algorithm for a total of 10 iterations. Each line in the dataset consists of a URL and one of its neighbours. You are required to **put this file to HDFS**.

6. **How to get your assignment GRADED:**
   - Show a working prototype to the Lab assistant
   - **Submit via Blackboard a** tar file containing
     - A short report describing your experience with Hadoop and Spark.
     - The top 10 rank websites in your computation.
     - How the performance of the application varies with respect to the resources of VMs and the number of VMs.
     - The tar file should be named: <group-number>_VM_context_2_2.tar

References:

1. http://wiki.cs.vu.nl/greenclouds/index.php/OpenNebula-User-4.4
2. https://archives.opennebula.org/documentation:archives:rel3.8
3. http://hadoop.apache.org/
4. https://spark.apache.org/
5. https://en.wikipedia.org/wiki/PageRank
6. https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/ClusterSetup.html
7. https://spark.apache.org/docs/latest/spark-standalone.html
8. https://linode.com/docs/databases/hadoop/how-to-install-and-set-up-hadoop-cluster/
9. https://snap.stanford.edu/data/web-BerkStan.html

IMPORTANT NOTE:

- Lab Teachers and the rest of the team coordinating the course will support you in doing your assignment during Lab Session or by reacting to your emails
- Lab Teachers have extensive experience in developing in Java and Python and will be able to provide you with High quality support if you develop in these two languages.
- Lab Teachers will provide you with **best Effort support** if you decide to develop in another language or use another environment than the one suggested in the assignment.