T-110.5150 Applications and Services in Internet

Assignment 2

Amazon Web Services

Contact: t-110.5150@tkk.fi

Deadline: 23.59 (Helsinki time) 30, Nov.,2011

V1.0
Aalto University
Data Communications Software Lab
2011

Goal

Introduce yourself to the Amazon Web Services (AWS) on cloud technology, specifically, Amazon Elastic Compute Cloud (EC2), Amazon Simple Storage Service (S3) and Amazon Elastic MapReduce.

Exercise summary

1) Get practical experience in Amazon EC2 and other Amazon Web Services.
2) Implement an experimental program to explore the scheduling behaviors between virtual machines in one single EC2 host; analyze the measurement results.
3) Implement an experimental program to test the network throughput between instances within a single region, and analyze the measurement results.

Usage

For this course, each student will get 100 USD credits from Amazon. To redeem the credit, you first need to obtain an Amazon account by following this link: http://aws.amazon.com/ec2/. After you have created the account at Amazon, you can redeem the credit by following this link: http://aws.amazon.com/awscredits. The code for the credit will be sent to you by email.

Note that for creating an account, Amazon needs your credit card information and phone number. You will not be charged before the 100$ is depleted, as they just want certain alternate payment method for this. If you feel uncomfortable of giving your credit card information to Amazon, please let us know so we could give you some other assignments instead.

Note: The 100 USD credit is more than enough for finishing the exercise work.

Caution: Monitor your account activity carefully, make sure the credit is enough for your usage and turn off the instances before you log out Amazon management console. Last year some students were charged some thousands of USD (even though we figured out some way to solve that problem eventually) as they forgot to terminate the instances. Clear your account, i.e. empty the Amazon S3, terminate all the instances in Amazon EC2, and close down the other related Amazon Web Services, after you have finished the assignment.

Basic features

1. Amazon S3 (10 pts)

Using Amazon S3 as the storage, upload files with different sizes (1KB, 10KB, 100KB, 1MB, 10MB etc), and test throughput from your local machine to the **Ireland** region for the bucket. Use s3cmd (http://s3tools.org/s3cmd) as the benchmark tool, or any similar tool you are familiar with can be used as well.

2. Amazon EC2 and scheduling behavior (30 pts)

The purpose of this experiment is to measure whether your Virtual Machine (VM) is sharing the physical processor with other VMs or not. One way to achieve this goal is to get the system time continuously, e.g. using the ***gettimeofday()*** function in C language (Posix compatible system). If the VM occupies the processor solely, the system time should be continuous without any noticeable gap. If the VM is sharing the processor with other VMs, there supposed to be some gap (during which period your VM is scheduled off and given way to other VMs running concurrently on the same physical machine) in the system time.

In this experiment, you are required to launch two types of Amazon EC2 instances: **Micro** and **Small**, by using the Basic 32-bit Amazon Linux Amazon Machine Image (AMI) 2011.09 (AMI Id: ami-31814f58). Then you are required to test the CPU utilization ratio of your VM, i.e. how much percentage of CPU time it gets during the whole process. Here is the link for getting started: http://docs.amazonwebservices.com/AWSEC2/latest/GettingStartedGuide/.

You are required to implement a small program, which retrieves current system time as fine-grained as possible, i.e. calling the aforementioned function ***gettimeofday()*** in C or similar functions in other programming languages, 1 million times in a loop. The system time can be recorded in the memory firstly, and then write through to the local file. Run this program on both the **Micro** and the **Small** instance and collect the data, then analyze the results carefully, you will notice the difference between the **Micro** and **Small** instances in terms of CPU utilization ratio. Use your data to draw a graph and explain your findings, then calculate the CPU utilization ratio for your **Micro** and the **Small** instances.

There is no any limitation on which tool you use for graph drawing. Some potential choices can be Microsoft Excel, Matlab, or GNU Plot (http://www.gnuplot.info/).

3. Amazon EC2 network throughput (30 pts)

Establish two pairs of instances (**Micro-Micro**, and **Small-Small**) in the **Ireland** region using the AMI mentioned before, then implement a TCP and UDP throughput program to measure the fine-grained network throughput between the pairs of instance.

Regardless of using TCP or UDP, a throughput measurement program consists of two parts, the client and the server. The client side acts as a sender while the server is a receiver. In the client side, a file is read and transmitted. In the server side, you calculate the throughput for every 256KB data received, record the finish time of this 256KB data and the throughput as a line of data.

In order to boost the performance of TCP, the sending buffer in the client and the receiving buffer in the server are needed to be enlarged to 256KB (enlarge these buffers will expand the sending and receiving window size in TCP). In the C language, this is achieved by **setsockopt()** function with option **SO_SNDBUF** and **SO_RCVBUF**; In Java, check the **setSendBufferSize()** and **setReceiveBufferSize()** methods in the class **java.net.Socket**. Other languages should also have similar interfaces.

For the UDP client and the server, each UDP packet should use a payload of 1024 bytes.

Use your data to draw two figures, i.e. one for TCP throughput and one for UDP throughput, then calculate the average TCP and UDP throughput for the whole period, compare the difference between the **Micro-Micro** pair and the **Small-Small** pair, and explain your findings, how this is related to the results from the previous CPU utilization experiments.

Similar to the basic feature 2, the tool for graph drawing can be chosen freely.

Additional features (25 pts each)

1. MapReduce

Test Amazon Elastic MapReduce as shown in this example: http://s3.amazonaws.com/awsVideos/AmazonElasticMapReduce/AmazonElasticMapReduce.html. Report the completion time for the task. Some benchmark tool for recording the completion time should be used.

2. Bundling image service

Test the bundling image service, either Windows OS or Linux OS of the original image.

3. P2P integration

If you implement a full feature p2p application in assignment 1, here you can deploy the application to more number of nodes (for example, 5 nodes in each instance, 5 instance) in Amazon EC2. Analyse the topology and search behaviors in your p2p network by using different parameters in the p2p application (Max number of peers, max number of PONG entries in one PONG message, different algorithms to select peers to be filled in a PONG message etc.); State your experiment setup and show your analysis in the report.

4. Amazon API and your own idea

Do you have your own ideas? Use Amazon API to develop your own service.

Survey (10 pts)

Complete two surveys, which includes some general questions regarding cloud computing and Amazon Web Services. The surveys should be finished after you have completed the exercise work and the final report. The link of this survey will be sent to you by email. The motivation for this survey is to express your opinions after being exposed to some real-life commercial services.

Note: The surveys need to be completed by every student who attend the second assignment, even you are in a group of two, you still need to finish them separately.

Submissions

The final report should include the following elements:
1. Report the process of the experiment, results and analysis for each basic feature;
2. Report the process of the experiment, and other related contents (mentioned in the requirement) for the additional features.

For the second assignment, please create a new directory called "ex2" to your repository's root directory and submit the final report to this repository.

Evaluation

Experimenting with basic features is enough, but by trying out some additional features you can get more points.The basic features are rather fixed, but you have free hands to try out additional features. With freedom comes responsibility, you should report your experience with detailed documentation.
* Basic features: 70 pts

* Additional features: 25 pts each(max. 2)
* Final report: 15 pts
* Finishing the surveys (10 pts)

Total: 155 Points
0 - failure (<70 points)
1 - (>=70 points)
2 - (>=90 points)
3 - (>=110 points)
4 - (>=125 points)
5 - (>=140 points)

The assignment 1and the assignment 2 each accounts for 20% in the whole course grading, in other words, the percentage of the whole assignment part is 40%, another 60% is the final exam.