Example: How to Set Up Parallel SECONDO on AWS EC2

Jiamin Lu, April 2013

SECONDO was built by the SECONDO team

In order to help users to set up Parallel SECONDO on their own clusters without purchasing any hardware, the Amazon Machine Image (AMI) of Parallel SECONDO is publicly provided on Amazon Web Services (AWS) EC2.

At present, the Parallel SECONDO AMI 1.2 is available on the zone US-East (located at Northern Virginia) and its AMI id is: ami-f3167d9a. It is derived from the public AMI ami-3d4ff254, using the operating system Ubuntu-Server12.04 64bit. Within the image, a SECONDO 3.3.2 is installed, and a Data Server example of Parallel SECONDO is prepared.

Of course it is possible that the user creates his/her own virtual clusters based on other Linux AMI in EC2, and installs Parallel SECONDO on them like common physical clusters. If so, it may take some time for the user to get familiar with the installation scripts of Parallel SECONDO and also with the usage of the EC2 services. Hence, the Parallel SECONDO AMI 1.2 is proposed to simplify the preparation work.

In the following, the steps of setting up Parallel SECONDO on either a single AWS EC2 instance or a virtual cluster consisting of multiple EC2 instances are introduced, respectively. Most of the steps are generally used by various Amazon EC2 users and are presented with abundant details in the User Guide of Amazon EC2. Therefore, we highly recommend the user to consult the *EC2 User Guide* when he/she needs more information about certain steps. At the same time, in this document the positions of some EC2 related steps in the *User Guide* are marked as blue and underlined, based on the guide version of 2013-02-01.

Setting Up Parallel Secondo on a Single AWS EC2 Instance

AWS EC2 provides 18 types of instances, see Available Instance Types (p.87). It is possible to set up Parallel SECONDO on a single EC2 instance of any type, just like starting a normal EC2 instance, by completely following the steps introduced in Getting Started with Amazon EC2 Linux Instances (p.8):

1. Sign up for Amazon EC2. An AWS account is required for using AWS services. This account must be combined with a credit card, but the user is only charged with the resources that he/she takes. The prices of EC2 services are listed on Amazon EC2 Pricing.
2. Launch an EC2 instance based on the Parallel SECONDO AMI 1.2. Amazon provides a browser-based console dashboard to manage EC2 resources, and the user can start a single instance Parallel Secondo by simply clicking the **Lauch Instance** button on the start page. Nevertheless, a few options should be specifically set for Parallel SECONDO.
    - The user should select the classic wizard to start the instance, in order to find and indicate the Parallel SECONDO AMI 1.2, ami-f3167d9a.

- Up to now, the Parallel SECONDO is only published in the region of US-East, hence the user must start an instance also in that region, or else it is impossible to find the image.
- In the security group setting, ports used by Hadoop include at least 49000, 49001, 50010, 50030, 50070 and 50075. The mini-SECONDO of the Data Server needs the port 11234. At last, the port 22 must be opened for SSH connection. The user can set this security group with any name that he/she likes, e.g PSGroup in this document.
- A key-pair is created and its private key file is downloaded to the local computer. Keep this file in an appropriate place, and change its permission to be read-only by the owner. The name of the key-pair can be set as any value, but the downloaded private key should not be renamed.

  ```
  $ mv myKey.pem $HOME/.ssh
  $ chmod 400 $HOME/.ssh/myKey.pem
  ```

  Here we name the key-pair as "myKey", and move the downloaded private key file myKey.pem to the $HOME/.ssh, where normally all SSH related files are kept.

3. In order to connect a started instance, the user can simply right click the instance on the console dashboard, and choose "Connect". Hereby, the instance is connected with the browser-based terminal MindTerm. Note here the User name is `ubuntu`, also the user should indicate the location of the downloaded private key file and the name of the security group, like the myKey.pem and PSGroup that are created before.

4. When the instance is logged in for the first time, two procedures are started. Note they are automatically executed with two scripts that have been encapsulated inside the Parallel SECONDO AMI, and the user doesn't have to manually carry them out. These two scripts process the following two tasks, respectively.

   - The first script is the $HOME/.parasecrc. It prepares the Data Server of the started instance based on the encapsulated DS example. If the instance is micro type, then the Data Server is set on the boot volume. Otherwise it is set to the temporary instance store, which has a much larger disk space.
   - The second script is the $HOME/secondo/Algebra/Hadoop/clusterManagement/_ps-ec2-initialize. It sets various parameters in the local Data Server according to the IP address of the instance, since each time an EC2 instance is assigned with a different IP address.

5. After initializing the instance, the Parallel SECONDO has already been set up. Format the namenode of Hadoop and start it, then start the monitor of the mini-SECONDO in the local instance. At last, the user can start the text interface of Parallel SECONDO and input parallel queries.

   ```
   $ hadoop namenode -format
   $ start-all.sh
   $ ps-startMonitors
   $ ps-startTTYCS -s 1
   ```

After setting up the single instance Parallel SECONDO, the user can retrieve the instance's public DNS address from the Instances panel of the console dashboard. Afterward, the user can monitor the started Hadoop's HDFS system by visiting the DNS address' 50070 port. At the same time, Hadoop's JobTracker can also be monitored by visiting the DNS address' 50030 port.

Although it is possible to set a single instance Parallel SECONDO on any type EC2 instances, it is quite difficult to process parallel queries on a single micro type instance, since the provided memory is too small. If the user prefers to use micro instances, since they are free for new EC2 users, we highly recommend him/her to set up Parallel SECONDO on a micro instance cluster, by simply following the subsequent steps.

If the user does not need the instance anymore, he/she can stop or terminate the instance on the console dashboard, see Stopping Instances (p.379).

Argument Name	Description	Type	Default Value
-h	Help information		
-i	AMI ID	string	ami-f3167d9a
-n	Number of instances	int	1
-m	Master node's Name tag	string	Master
-s	Slave nodes' Name tag	string	Slaves
-g	Security group name	string	
-k	The position of the downloaded private key file	string	
-t	Instance type	string	t1.micro
-z	Availability zone	string	us-east-1a
-o	additional options master-is-slave	 yes/no	 yes

Table 1: Arguments in ps-ec2-startInstances

Setting Up Parallel Secondo on a AWS EC2 cluster

In order to easily create a virtual EC2 cluster and set up a Parallel SECONDO on it, a bash script is provided, being named `ps-ec2-startInstances`. This script is independent from other Parallel SECONDO scripts, therefore it can be run without installing SECONDO. However, it uses many EC2 Command Line Interface Tools (CLI Tools), which thus the user must set up first. Setting up CLI Tools is quite easy, and the detailed steps are described in Setting Up the Amazon EC2 Command Line Interface Tools(p. 562). The CLI Tools are available on all Linux/Unix/MacOSX platforms, and so is the `ps-ec2-startInstances` script.

After installing CLI Tools, creating an EC2 cluster and setting up a Parallel SECONDO on it require the following steps:

1. Create the key-pair and the security group on EC2. More details are given in Getting a Key Pair (p.274) and Creating Your Own Security Groups (p.422), respectively. If both have already been done in the last section prepared for a single instance, then they can be repeatedly used on all EC2 clusters created by the current AWS account.

2. Download the `ps-ec2-startInstances` script from our website. The parameters required by this script are shown in the Table 1. Most of them have been set with default values. The default value of the AMI ID is set for the Parallel SECONDO AMI 1.2, and the instances by default are built on the free micro instances on the us-east-1a region. In particular, the position of the private key file can be pointed out with an environment variable named **`EC2_KeyPair`**, so that the user does not have to specially indicate it for the script. For example, the following command

   ```
   $ export EC2_KeyPair=$HOME/.ssh/myKey.pem
   $ ps-ec2-startInstances -n 5 -g PSGroup
   ```

   creates a virtual cluster consisting of five micro type EC2 instances in the us-east-1a region. The myKey.pem is the private key file that was downloaded when we set up the single instance Parallel SECONDO. The security group "PSGroup" is also created at that time, opening all required ports by Hadoop, mini-SECONDO and SSH.

   Among the five started instances, one is viewed as the master node, and its Name tag is marked as "Master". The other four instances are then marked as "Slaves" on their Name tags. In the mean time, since the option *master-is-slave* is set as the default value "yes", the master node is also used as a slave in the Parallel SECONDO. Thereby, the cluster in total has one master node and five slave nodes.

Usually the start-up procedure takes several minutes and finishes when all started instances are reachable. Sometimes, especially when the user wants to create a large-scale cluster consisting of tens or even hundreds of instances, it happens that several instances cannot be started anyway. In this case, it prompts like this:

```
Till now, ? instances are started, and there are still ? instances pending.
Would you like to:
  1) Keep waiting for another ten seconds.
  2) Terminate all unstarted instances, and start new ones.
  3) Abort. (Note!! All started instances are not stopped.)
```

The user can select different options based on his/her own situation.

When the start-up is about to finish, it may prompt something like:

```
The authenticity of host '... (...)' can't be established.
RSA key fingerprint is .......
Are you sure you want to continue connecting (yes/no)?
```

Here the user is asked to add the master instance to the known host list, and he/she can simply type "yes". This prompt shows up every time when a new cluster is created. If the user wants to avoid this, the following lines can be added into the $HOME/.ssh/config file

```
Host *
StrictHostKeyChecking   no
UserKnownHostsfile      /dev/null
```

3. The output of the last script looks like this:

   ```
   The initialization is finished,
   you can log into the master node with command:
   ssh -i *.pem ubuntu@ec2-*.amazonaws.com
   ```

   The user can access the master node of the cluster by using the given ssh command directly.

4. When the user is logged on the master node for the first time, the initialization of the Parallel SECONDO starts automatically. It takes another several minutes, and then the user can start to use Parallel SECONDO directly.

   ```
   $ hadoop namenode -format
   $ start-all.sh
   $ cd $HOME/secondo/Algebras/Hadoop/clusterManagement
   $ ps-start-AllMonitors
   $ ps-startTTYCS -s 1
   ```

The cluster can also be stopped or terminated in the console dashboard of AWS EC2 when the user doesn't need it any more. Besides, the user can also use the CLI Tools to stop or terminate all instances at the same time. Note that when the cluster is not terminated and eliminated from the dashboard of AWS EC2 system, the Name tags used for its instances can not be used on the instances of other clusters, in order to help the user to distinguish clusters that are started at different times. Not only that, even when a cluster is already terminated, the instances will not be removed from the user's account in a certain time. Therefore, it is better for the user to set different Name tags on each cluster.