HamaStreaming

Hama Streaming

This article focuses on the usage of Hama Streaming with Python.

Setup

We hope that you have installed the latest version of Apache Hama, Streaming is available since 0.6.0.

If you haven't yet installed Hama, please go through the manual in the GettingStarted article.

For the Python version, you need Python 3.2, if you are not running it, have a look at the various tutorials to install it. So verify that you run the latest python version, a very quick way is to check if there is a python3.2 command, or the normal python interpreter tells you the correct version number.

Now you should start your HDFS deamons.

So for the first step, please change into the directory of your Hama installment. If you see the bin/conf and lib folder and a couple of jars, you are probably right.

Running example in Streaming

Now let's start the Hama cluster:

```
bin/start-bspd.sh
```

Once started, you can get yourself familiar with the shell submitter of pipes and streaming jobs:

```
bin/hama pipes
```

Now a good way to start is to retrieve the Hama Streaming for Python from github by executing

```
git clone git://github.com/thomasjungblut/HamaStreaming.git
```

If you don't have git installed, no problem: you can download a zip file from the https://github.com/thomasjungblut/HamaStreaming.

In any case you should now find a "HamaStreaming" folder in your Hama home directory which contains several scripts.

Now we have to upload these scripts to HDFS:

```
hadoop/bin/hadoop fs -mkdir /tmp/PyStreaming/
hadoop/bin/hadoop fs -copyFromLocal HamaStreaming/* /tmp/PyStreaming/
```

Let's start by executing the usual Hello World application that already ships with streaming:

```
bin/hama pipes -streaming true -bspTasks 2 -interpreter python3.2 -cachefiles /tmp/PyStreaming/*.py -output /tmp
/pystream-out/ -program /tmp/PyStreaming/BSPRunner.py -programArgs HelloWorldBSP
```

This will start 2 bsp tasks in streaming mode. In streaming a child process will be forked from the usual BSP Java task. In this case, this would yield to a new task that starts with python3.2, with the py files from HDFS. The noteworthy thing is actually, that you pass a runner class that takes care of all the protocol communication. Your user program is passed as the first program argument. This works because python will start the runner py in a work directory from the cache files. So they are implicitly included and the whole computation can work, this is why you don't have to provide a path with the HelloWorldBSP (note the py is not needed, because of the reflective import).

Hopefully you should see something along these lines:

```
12/09/17 19:06:31 INFO pipes.Submitter: Streaming enabled!
12/09/17 19:06:33 INFO bsp.BSPJobClient: Running job: job_201209171906_0001
12/09/17 19:06:40 INFO bsp.BSPJobClient: Job complete: job_201209171906_0001
12/09/17 19:06:40 INFO bsp.BSPJobClient: The total number of supersteps: 15
12/09/17 19:06:40 INFO bsp.BSPJobClient: Counters: 8
12/09/17 19:06:40 INFO bsp.BSPJobClient:   org.apache.hama.bsp.JobInProgress$JobCounter
12/09/17 19:06:40 INFO bsp.BSPJobClient:     LAUNCHED_TASKS=2
12/09/17 19:06:40 INFO bsp.BSPJobClient:   org.apache.hama.bsp.BSPPeerImpl$PeerCounter
12/09/17 19:06:40 INFO bsp.BSPJobClient:     SUPERSTEP_SUM=15
12/09/17 19:06:40 INFO bsp.BSPJobClient:     COMPRESSED_BYTES_SENT=3310
12/09/17 19:06:40 INFO bsp.BSPJobClient:     TIME_IN_SYNC_MS=2805
12/09/17 19:06:40 INFO bsp.BSPJobClient:     COMPRESSED_BYTES_RECEIVED=3310
12/09/17 19:06:40 INFO bsp.BSPJobClient:     TOTAL_MESSAGES_SENT=60
12/09/17 19:06:40 INFO bsp.BSPJobClient:     TOTAL_MESSAGES_RECEIVED=30
12/09/17 19:06:40 INFO bsp.BSPJobClient:     TASK_OUTPUT_RECORDS=28
```

And now you can view the output of your job with:

```
hadoop/bin/hadoop fs -cat /tmp/pystream-out/part-00001
```

in my case this looks like this:

```
Hello from localhost:61002 in superstep 0
Hello from localhost:61001 in superstep 0
Hello from localhost:61001 in superstep 1
Hello from localhost:61002 in superstep 1
[...]
Hello from localhost:61001 in superstep 14
Hello from localhost:61002 in superstep 14
```