

Installation and Running GoFFish-2.6.1 on Cluster

Prerequisites:

Total number of nodes = 3

Number of head nodes = 1

Number of worker nodes = 3 (head node is also worker node)

There can be any number of worker nodes.

Configurations:

Head node: (Worker node 1)

Server1 = server number 58

IP: 10.10.1.58

Configuration: 1 TB Hard Disk, 16 GB RAM

Software installed: Java 1.7+ , Maven 3.0+ , Cmake 2.7+ , metis , python 2.7+

Worker node 2:

Server2 = server number 59

IP: 10.10.1.59

Configuration: 1 TB Hard Disk, 16 GB RAM

Software installed: Java 1.7+

Worker node 3:

Server3 = server number 60

IP: 10.10.1.60

Configuration: 1 TB Hard Disk, 16 GB RAM

Software installed: Java 1.7+

Step 1: Enable password less SSH between the 3 nodes

A. Making password less connection from server1 to server2 and server3:

```
[user1@server1 ~]$ ssh-keygen -t rsa
```

```
[user1@server1 ~]$ ssh-copy-id user1@server2
```

```
[user1@server2's password: [Give password for user1]
```

```
[user1@server1 ~]$ ssh-copy-id user1@server3
```

```
[user1@server3's password: [Give password for user1]
```

Similarly, follow the steps in worker node 2 and worker node 3.

Step 2: Software installation

A. Install the following software on the head node:

- Java 1.7+ - [\[http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html\]](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)
- Maven 3.0+ - [\[http://maven.apache.org/download.cgi\]](http://maven.apache.org/download.cgi)
- Cmake 2.7+ - [\[http://www.cmake.org/download/\]](http://www.cmake.org/download/)
- metis - [\[http://glaros.dtc.umn.edu/gkhome/views/metis\]](http://glaros.dtc.umn.edu/gkhome/views/metis)
- python 2.7+ - [\[https://www.python.org/download/releases/2.7/\]](https://www.python.org/download/releases/2.7/)

B. Install the following software on the worker nodes:

- Java 1.7+ - [\[http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html\]](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)

Step 3: Edit the goffish-conf.json file in goffish-2.6.1/install

```
{
  "machines" :{
    "headnode": {
      "id" : "node1",
      "address" : 10.10.1.58
    },
    "nodes" : [
      {
        "id" : "node1",
        "address" : 10.10.1.58
      },
      "nodes" : [
        {
          "id" : "node1",
          "address" : 10.10.1.59
        },
        "nodes" : [
          {
            "id" : "node1",
            "address" : 10.10.1.60
          }
        ]
      }
    ],
    "username" : {
      "default" : "hduser"
    },
    "paths" : {
      "default" : {
        "source" : "/home/hduser/goffish/goffish-deploy/source",
        "bin" : "/home/hduser/goffish/goffish-deploy/bin",
        "data" : "/home/hduser/goffish/goffish-deploy/data",
        "client" : "/home/hduser/goffish/goffish-deploy/client",
        "config" : "/home/hduser/goffish/goffish-deploy/config",
        "sample" : "/home/hduser/goffish/goffish-deploy/sample"
      }
    },
    "common-home" : "true",
    "source" :{
      "type" : "file",
      "url" : "~/goffish/goffish-2.6.1/"
    }
  }
}
```

Step 4: Deployment

For deploying GoFFish folder run the following command in your prompt:

```
$python goffish_install.py DEPLOY
```

Once GoFFish is deployed, a new folder called goffish will be created in the home directory. Inside this folder is another folder called goffish-deploy which contains various other folders among which client folder contains the scripts to run the sample graph.

Running Sample GoFFish job:

Here we describe step by step instruction for running the sample program (vertex count) on the sample graph (facebook graph)

To run a goffish job from scratch we need to do the following steps:

- 1) Start a GoFS server
- 2) Format the GoFS File System
- 3) Load the graph in to GoFS
- 4) Load the graph in to Gopher
- 5) Run Gopher job
- 6) Checking the result

Following steps describes how to perform each step:

Before starting the GoFS server give all the necessary permissions to gofs and gopher folders located inside client of goffish-deploy folder inside goffish folder.

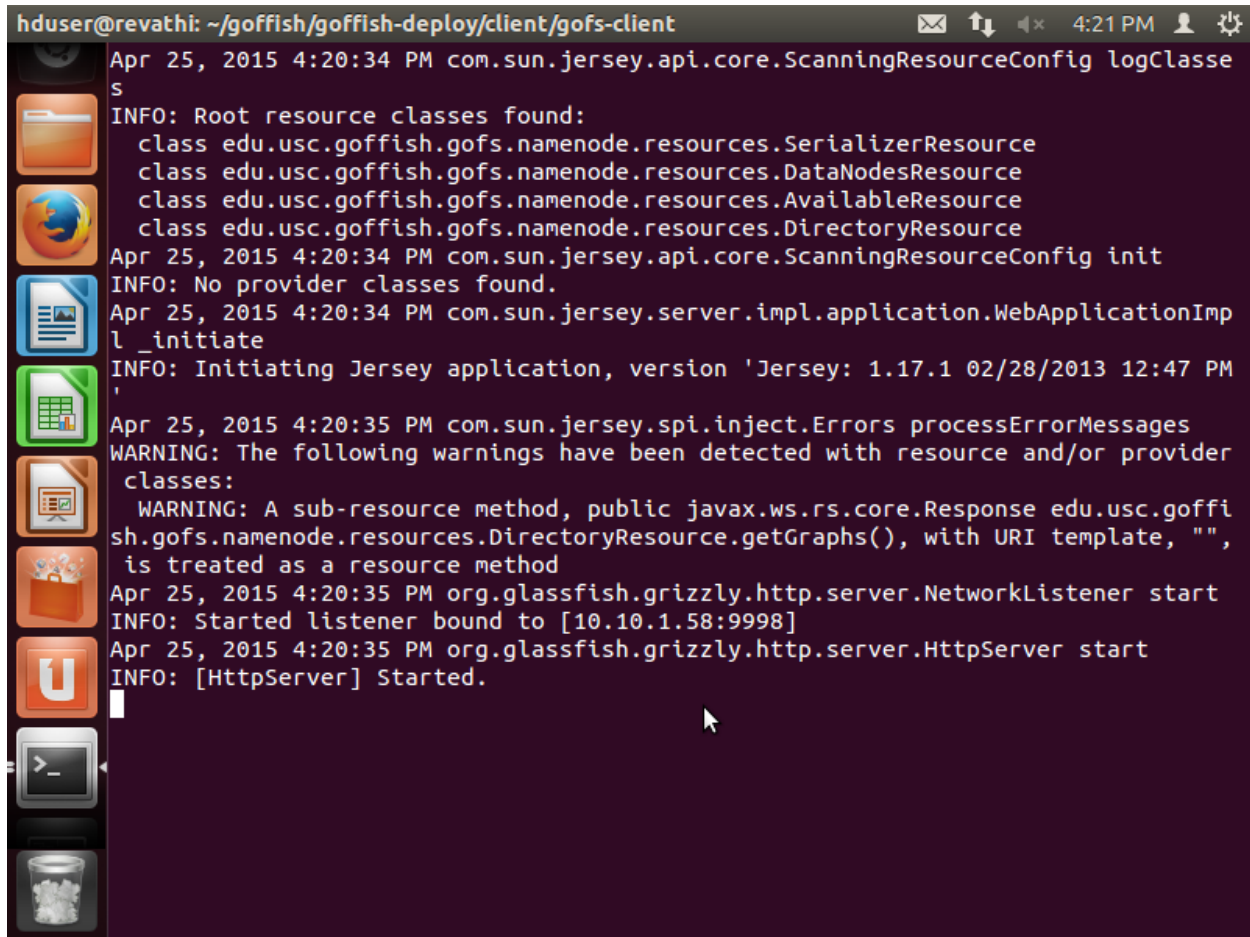
1. Start GoFS server

Go to ~/goffish/goffish-deploy/client/gofs-client

Now run the script "gofs-client.py"

```
$python gofs-client.py START
```

It will start the GoFS server

A terminal window titled 'hduser@revathi: ~/goffish/goffish-deploy/client/gofs-client'. The window shows a series of log messages from a Java application. On the left side of the terminal, there is a vertical dock with several application icons: a folder, a globe, a document, a spreadsheet, a presentation, a folder with a magnifying glass, a 'U' logo, a terminal icon, and a trash can. The log messages include timestamps and class names, indicating the initialization of Jersey and the starting of a Grizzly HTTP server.

```
hduser@revathi: ~/goffish/goffish-deploy/client/gofs-client
Apr 25, 2015 4:20:34 PM com.sun.jersey.api.core.ScanningResourceConfig logClasses
INFO: Root resource classes found:
class edu.usc.goffish.gofs.namenode.resources.SerializerResource
class edu.usc.goffish.gofs.namenode.resources.DataNodesResource
class edu.usc.goffish.gofs.namenode.resources.AvailableResource
class edu.usc.goffish.gofs.namenode.resources.DirectoryResource
Apr 25, 2015 4:20:34 PM com.sun.jersey.api.core.ScanningResourceConfig init
INFO: No provider classes found.
Apr 25, 2015 4:20:34 PM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.17.1 02/28/2013 12:47 PM'
Apr 25, 2015 4:20:35 PM com.sun.jersey.spi.inject.Errors processErrorMessages
WARNING: The following warnings have been detected with resource and/or provider
classes:
WARNING: A sub-resource method, public javax.ws.rs.core.Response edu.usc.goffi
sh.gofs.namenode.resources.DirectoryResource.getGraphs(), with URI template, "",
is treated as a resource method
Apr 25, 2015 4:20:35 PM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [10.10.1.58:9998]
Apr 25, 2015 4:20:35 PM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
```

2. Format the GoFS File System

Now open another terminal and in the same folder run

```
$python gofs-client.py FORMAT
```

It will format the gofs file system

```
hduser@revathi: ~/goffish/goffish-deploy/client/gofs-client
# repeat for each data node to include
gofs.datanode =file://10.10.1.58/home/hduser/goffish/goffish-deploy/data
gofs.datanode =file://10.10.1.59/home/hduser/goffish/goffish-deploy/data
gofs.datanode =file://10.10.1.60/home/hduser/goffish/goffish-deploy/data
# full class name of the serializer to use at every data node
# gofs.serializer = edu.usc.goffish.gofs.slice.JavaSliceSerializer
gofs.serializer = edu.usc.goffish.gofs.slice.KryoSliceSerializer

Parsing config...
Contacting name node...
Contacting data nodes...
executing: "ssh 10.10.1.58 cd /home/hduser/goffish/goffish-deploy/data;true;"
executing: "ssh 10.10.1.59 cd /home/hduser/goffish/goffish-deploy/data;true;"
executing: "ssh 10.10.1.60 cd /home/hduser/goffish/goffish-deploy/data;true;"
Formatting data node file://10.10.1.58/home/hduser/goffish/goffish-deploy/data/g
ofs/...
executing: "scp -B -q -r -p /tmp/gofs_format3155667094509836025/gofs 10.10.1.58:
/home/hduser/goffish/goffish-deploy/data/"
Formatting data node file://10.10.1.59/home/hduser/goffish/goffish-deploy/data/g
ofs/...
executing: "scp -B -q -r -p /tmp/gofs_format3155667094509836025/gofs 10.10.1.59:
/home/hduser/goffish/goffish-deploy/data/"
Formatting data node file://10.10.1.60/home/hduser/goffish/goffish-deploy/data/g
ofs/...
executing: "scp -B -q -r -p /tmp/gofs_format3155667094509836025/gofs 10.10.1.60:
/home/hduser/goffish/goffish-deploy/data/"
GoFS format complete
/home/hduser/goffish/goffish-deploy/bin/gofs-bin/bin/GoFSFormat
hduser@revathi:~/goffish/goffish-deploy/client/gofs-client$
```

3. Loading Graph in GoFS

In the same folder run

\$python gofs-client.py LOAD

It will deploy the Facebook graph on GoFS

4. Loading graph in Gopher

Go to ~/goffish/goffish-deploy/client/gopher-client

Now run the script "gopher-client.py"

\$python gopher-client.py LOAD

This will load the gopher jar in all the machines in goffish cluster as well as start the manager and coordinator (part of floe framework) which supervises the execution

```
hduser@revathi: ~/goffish/goffish-deploy/client/gopher-client
PRINTING CLASSPATH /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-client/
Manager Co-ordinator and Container loaded
Running command on hduser@10.10.1.58
ssh hduser@10.10.1.58 -tt "ps -ef|grep edu.usc.pgroup.floe.startup.Coordinator|grep -v grep|awk {'print \$2'} >> /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-server/bin/kill_list"
Connection to 10.10.1.58 closed.
Running command on hduser@10.10.1.58
ssh hduser@10.10.1.58 -tt "ps -ef|grep edu.usc.pgroup.floe.startup.Manager|grep -v grep|awk {'print \$2'} >> /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-server/bin/kill_list"
Connection to 10.10.1.58 closed.
Running command on hduser@10.10.1.58
ssh hduser@10.10.1.58 -tt "ps -ef|grep edu.usc.goffish.gopher.impl.Main|grep -v grep|awk {'print \$2'} >> /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-server/bin/kill_list"
Connection to 10.10.1.58 closed.
Running command on hduser@10.10.1.59
ssh hduser@10.10.1.59 -tt "ps -ef|grep edu.usc.goffish.gopher.impl.Main|grep -v grep|awk {'print \$2'} >> /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-server/bin/kill_list"
Connection to 10.10.1.59 closed.
Running command on hduser@10.10.1.60
ssh hduser@10.10.1.60 -tt "ps -ef|grep edu.usc.goffish.gopher.impl.Main|grep -v grep|awk {'print \$2'} >> /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-server/bin/kill_list"
Connection to 10.10.1.60 closed.
All the process id logged in kill_list (will be used for killing gopher)
hduser@revathi:~/goffish/goffish-deploy/client/gopher-client$ python gopher-client.py RUN
```

5. Running sample gopher job

In the same folder run

\$python gopher-client.py RUN

This will run the gopher job on goffish. Currently this script will keep on running even when the gopher job has stopped. Click ^C to close the gopher job

```
hduser@revathi: ~/goffish/goffish-deploy/client/gopher-client
{u'bin': u'/home/hduser/goffish/goffish-deploy/bin', u'config': u'/home/hduser/goffish/goffish-deploy/config', u'sample': u'/home/hduser/goffish/goffish-deploy/sample', u'source': u'/home/hduser/goffish/goffish-deploy/source', u'client': u'/home/hduser/goffish/goffish-deploy/client', u'data': u'/home/hduser/goffish/goffish-deploy/data'}
/home/hduser/goffish/goffish-deploy/bin /home/hduser/goffish/goffish-deploy/data
Bin: /home/hduser/goffish/goffish-deploy/bin Data: /home/hduser/goffish/goffish-deploy/data
Id : node1 address: 10.10.1.58 user: hduser path: Bin: /home/hduser/goffish/goffish-deploy/bin Data: /home/hduser/goffish/goffish-deploy/data
Id : node1 address: 10.10.1.58 user: hduser path: Bin: /home/hduser/goffish/goffish-deploy/bin Data: /home/hduser/goffish/goffish-deploy/data
Id : node1 address: 10.10.1.59 user: hduser path: Bin: /home/hduser/goffish/goffish-deploy/bin Data: /home/hduser/goffish/goffish-deploy/data
Id : node1 address: 10.10.1.60 user: hduser path: Bin: /home/hduser/goffish/goffish-deploy/bin Data: /home/hduser/goffish/goffish-deploy/data
{"jar_path": "$GOFFISH_SAMPLE/gopher-apps/vert-count-2.6.jar", "class_path": "edu.usc.pgroup.goffish.gopher.sample.VertCounter", "graph_id": "fbgraph1", "araguments": "NILL", "iterations": "0"}
/home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-client/bin/GopherClient.sh /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-client/gopher-config.xml /home/hduser/goffish/goffish-deploy/bin/gopher-bin/conf/gofs.config fbgraph1 vert-count-2.6.jar edu.usc.pgroup.goffish.gopher.sample.VertCounter NILL 0
Using HOME DIR /home/hduser/goffish/goffish-deploy/bin/gopher-bin/gopher-client
Using java version:
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
Sending Init message : vert-count-2.6.jar,edu.usc.pgroup.goffish.gopher.sample.VertCounter,3,fbgraph1,http://10.10.1.58:9998
```

6. After you have completed the gopher job run following command

\$python gopher-client.py KILL

This will kill the manger and coordinator. If not next time you load a gopher job you will get a MIME type exception

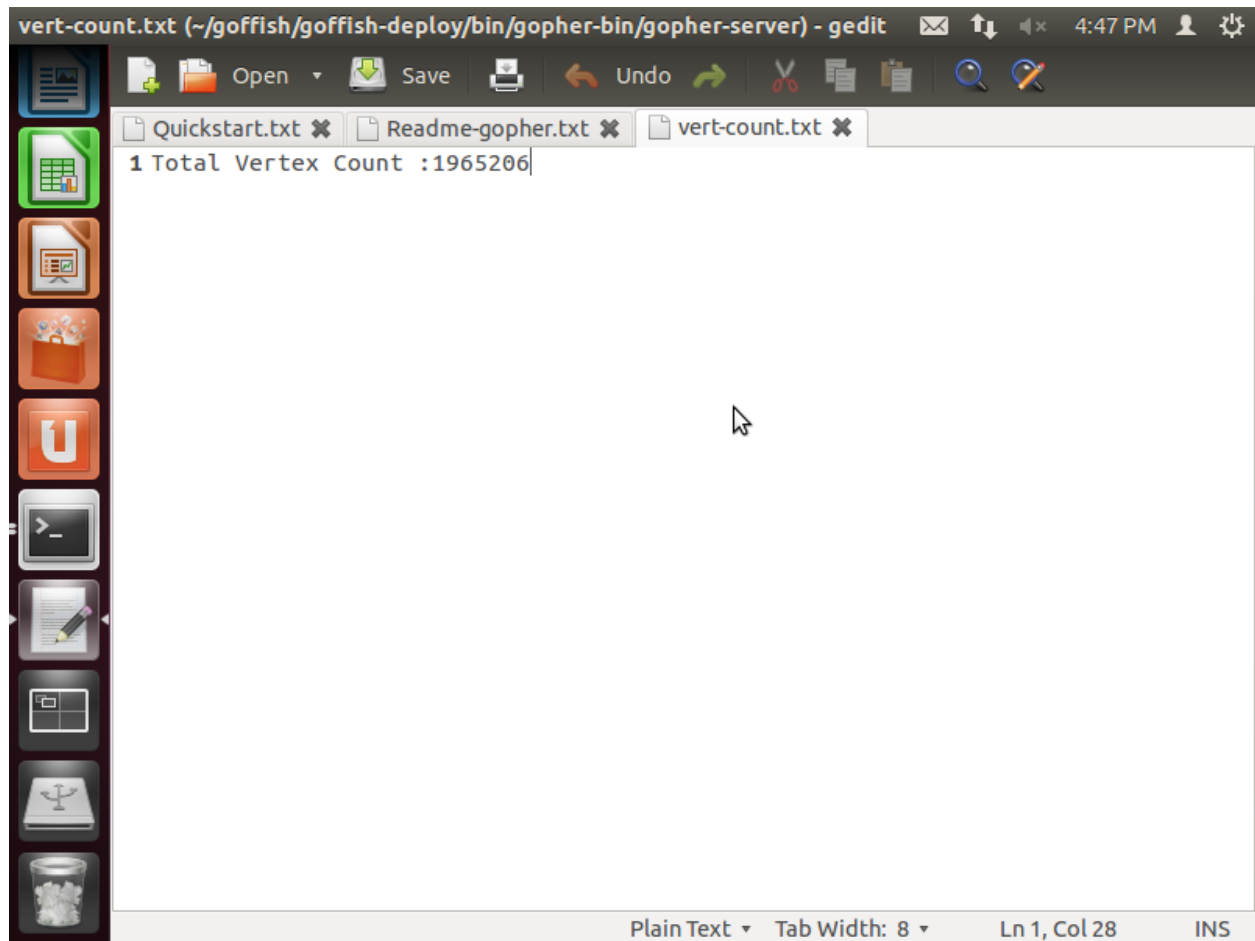
7. Checking the result

The above script will keep running.

To check the result go to \$goffish/goffish-deploy/bin/gopher-server-2.0

We will see a file "vert-count.txt" having output

Total Vertex Count: 4039



This documentation is all about installing and running a sample graph (vertex count program on Facebook graph) on GoFFish platform. We can also run other sample graphs on this platform.

Log Files:

The log files created are:

- Container.log
- Partition.log
- Subgraph.log

These files contain the logs from which the computation time, IO time and total app time can be observed.