Introduction

MapReduce is a software framework for data intensive applications provides the parallel and distributed platform. Google developed and uses this MapReduce framework for computing several hundreds of applications using MapReduce[3]. The drawbacks with this Google MapReduce framework are it is not open source, and it is tied to Google file systems to store and process the data[3]. However, there are other open source applications like Yahoo's Hadoop MapReduce but tied to Hadoop filesystem[4]. SAGA MapReduce is aimed to implement MapReduce framework in heterogeneous environments that are supported by SAGA (Simple API for Grid Applications). However, extracting performance is difficult using the SAGA MapReduce because of the overhead in providing interoperability [2].

In depth details of MapReduce can be found at:

http://saga.cct.lsu.edu/publications/papers/confpapers/Application-Level-Interoperability-between-Clouds-and-Grids

http://saga.cct.lsu.edu/publications/papers/journals/understanding-application-level-interoperability-scaling-out-mapreduce-over-high-performance-grids-and-clouds

http://labs.google.com/papers/mapreduce.html

http://saga.cct.lsu.edu/publications/papers/confpapers/Programming-Abstraction-for-Data-Intensive-Computing-on-Clouds-and-Grids

For SAGA-MapReduce implementation and working details
Please go through User Manual and Programming Manual provided in the svn.

https://svn.cct.lsu.edu/repos/saga-projects/applications/MapReduce/branches/MapReduce.2009/

Prerequisites

- SAGA-MapReduce is works over boost-1.40, 1.42, 1.43 version.

- Download and install latest version of SAGA from
  http://saga.cct.lsu.edu/software/cpp/download

For Installation instructions you may want use the quick start guide from here
http://saga.cct.lsu.edu/software/cpp/documentation

- Access to a SAGA-Advert server is mandatory. It is used for communication between master and worker.

- Install Google Protocol Buffers.

To run SAGA-MapReduce distributed different machines:

- SAGA SSH adaptor is mandatory.
- Fuser mount support is also required the input output locations across the machines.

Installing SAGA MapReduce

SAGA environment Variables you may need to set:
export SAGA_LOCATION=/path/to/SAGA
export BOOST_LOCATION=/path/to/BOOST
export LD_LIBRARY_PATH=/path/to/SAGA /lib:path/to/postgresql /lib:/path/to/googlebuffers/lib:$LD_LIBRARY_PATH
export PATH=/path/to/SAGA /bin:$PATH

To begin, download SAGA-MapReduce code from SVN Repository here

```
$ svn co https://svn.cct.lsu.edu/repos/saga-projects/applications/MapReduce/branches/MapReduce.2009/
```

Compile the code as follows but make sure that you have set the SAGA environment

```
$ cd /path/to/MapReduce.2009
$ make
```

If you run into errors try(some time this may be mandotory)

```
$ make libclean
$ make
```

Once the code is compiled you are good to go.

Setting up the Environment:
Either use put the MapReduce libraries in environment.

```
$ export LD_LIBRARY_PATH=/path/to/MapReduce.2009/source/mapreduce
```

For mac:

```
$ export DYLD_LIBRARY_PATH=/path/to/MapReduce.2009/source/mapreduce
```

or run

```
$ make install
```

To run the word count example:

```
$cd /path/to/MapReduce.2009/examples/wordcount/
```

Edit the LocalTest.xml file

1. Set the advert server location to xml file.
   (user name and password for advert server should handled in $SAGA_LOCATION/saga/share/saga_default_advert.ini)
   Example:

```
<!-- The orchestrator host (AdvertDB) -->
<OrchestratorDB>
<Host>
advert://advert.cct.lsu.edu/8080
</Host>
</OrchestratorDB>
```

2. Set the Binary image path

Example:

```
<!-- Application binaries for different platforms/architectures -->
<ApplicationBinaries>
<BinaryImage arch="i686" OS="Linux" extraArgs="">
        /path/to/MapReduce.2009/examples/wordcount/wordcount
</BinaryImage>
</ApplicationBinaries>
```

3. Set the out put location.

   Example:

   ```
   <!-- Output file system schema and base location -->
   <OutputBase>file://localhost//home/username/workerop/</OutputBase>
   ```

4. If you want to add multiple workers you can add it in these

```
<TargetHosts>
     <Host arch="i686" OS="Linux"> fork://localhost</Host>
     <Host arch="i686" OS="Linux"> ssh://localhost</Host>
     <Host arch="i686" OS="Linux"> ssh://hostname.com/</Host>
</TargetHosts>
```

5. If you want to change the intermediate files which are created during execution SAGA-MapReduce you can change it here.

   Example

   Edit MapReduce.2009/source/mapreduce/output/FileOutputFormat.cpp

```
std::string FileOutputFormat::GetOutputBase(const JobDescription& job) {
        return job.get_attribute(JOB_ATTRIBUTE_FILE_OUTPUTBASE,
        "file://localhost//home/uame/op");}

std::string FileOutputFormat::GetOutputPath(const JobDescription& job) {
return job.get_attribute(JOB_ATTRIBUTE_FILE_OUTPUTPATH,"")}
```

6. Set the input location in MapRedcue.2009/examples/wordcount/wordcount.cpp

   Example

   ```
   FileInputFormat::AddInputPath(job, "file://localhost//path/to/test-data/")
   ```

7. You can run the wordcount using this command line

   ```
   $ cd /MapRedcue.2009/examples/wordcount/
   $ ./wordcount –c LocalTest.xml
   ```

8. To run distributed using ssh
   - Mount the input and output temporary file locations, across machine in the same path(use sshfs command)
   - Make sure you set the same locations for intermiditate files in the MapReduce framework too.
   - Make sure to change the locations to appropriate path and the lines in this file

     /MapReduce.2009/source/mapreduce/master/DistributedJobRunner.cpp

     ```
     std::vector<std::string> env;
     ```

```
env.push_back("SAGA_LOCATION=/work/smaddi2/saga15");
env.push_back("LD_LIBRARY_PATH=/work/smaddi2/saga15//lib:/work/smaddi2/mapreduc
e/source/mapreduce/: LD_LIBRARY_PATH ");
```

- Make sure ssh file adaptor for saga is installed and password less login between machines
- Make sure you have changed the Master location to the node address Like this in the xml config file
  <MasterAddress>tcp://cyd01.cct.lsu.edu:80011</MasterAddress>
- Update the path to executable in configuration file. If the path of executables on multiple machines are different

9. To Run on a Cluster like queen bee
   - Make sure MapReduce using SSH is working.
   - Use this python script

```
from socket import gethostname; print gethostname()
hnm= gethostname()
file = open("/work/smaddi2/ndfile.txt","rb")
sseq = "test"
list = ""
r=0
lstr=""
while (sseq != ''):
    sseq = file.readline()
    kl= str(sseq)
    lst= kl.rstrip()
    if not (sseq == ''):
     if (lstr == lst):
       if (r < 3):      # change this number according to the number of workers on each node
          list = list + '<Host arch="x86_64" OS="redhat"> ssh://%s/ </Host>\n' %lst
          r=r+1
          lstr=lst
       elif(r==7):
          r=0
       else:
          r=r+1
     else:
         list = list + '<Host arch="x86_64" OS="redhat"> ssh://%s/ </Host>\n' %lst
         lstr = lst
         r=r+1

file.close()
print list
nfile = open("/work/smaddi2/mapreduce/examples/terasort/pbs_use.xml","rb")
sseq = "test"
newxml = ""
while (sseq != ''):
    sseq = nfile.readline()
    if ("replaceme" in sseq): #place replace me accordingly

       newxml = newxml + list
    elif ("replace1me" in sseq):
#place replace1me accordingly in file/work/smaddi2/mapreduce/examples/terasort/pbs_use.xml

       newxml = newxml + " <MasterAddress>tcp://%s:80011/</MasterAddress>"%hnm
    else:
       newxml = newxml + sseq
nfile.close()
FILE = open("/work/smaddi2/mapreduce/examples/terasort/pbs.xml", "w", 0)
FILE.writelines(newxml)
```

   - Use this qsub script

```
#!/bin/bash
#PBS -N maplog
#PBS -l walltime=15:00:00,nodes=32:ppn=8
#PBS -q workq
echo Job $PBS_JOBNAME is executing in $PBS_O_WORKDIR
cat $PBS_NODEFILE > /work/smaddi2/ndfile.txt
cd /path/to/mapreduce/examples/terasort/
python xml.py
./terasort /work/smaddi2/mrinput/ -c pbs.xml
```