# Temp

```
NOTE:The Hama version is 0.6.1

Q1: At org.apache.hama.bsp.GroomServer.java (line:343~349)
   If the GroomServer is reinitialized (at line:901) then the rpcPort and rpcAddr will be constant values
anyway that always cause throwing IllegalArgumentException at line 406.So I think the line 346~349 should
follow behind the line 344 which out of the ‘if’ statement.

Q2: At org.apache.hama.bsp.GroomServer.java (line:1149)
   I’m sure it is just a slip of the pen.  “Master: ”here should be “GroomServer: ”

Q3: At org.apache.hama.bsp.SuperstepBSP.java (line:74)
   Why reassigned startstep to zero in each loop? Personally believe that this line is redundant since the step
counter is recorded at each Barrier Synchronization automatically(line:73).

Q4:As far as I know that the Hama framwork exist single master of failure problem, which means once the master
shuts down the program will report error.Compare with namenode play a role of namespace manager in HDFS ,the
BSPMaster play the key role in bussiness logic layer. It is more important to keep the reliability of master.So
I’m wondering if we can do the following:
Add a private variable as a tag in BSPMaster.  private boolean isBackup=false
  Add hardware parameter to GroomServerStatus like CPU frequency and available memory.
  Add a new state for GroomServer static enum State.BLOCK .

Once the BSPMaster is initialized, chooses a optimal GroomServerStatus from groomServers map by using some kind
of dynamic election algorithm.Then send a SetupBSPMasterAction order with master mirror image file which
include the  current master status, job file and so on to the assigned GroomServer.

 Class DynamicElect{
   public static GroomServerStatus MasterBackup
   (Collection<GroomServerStatus>  ServerStatus){
     //get each hardware parameter from the Collectin by using a ‘for’loop
     //sort status by ascending or descending
     //choose a optimal one like available memory maximum and cpu rank top class
  }
}

All the GroomServer receive and handle action in its Instructor thread. When the program execute if(action
instanceof SetupBSPMasterAction) statement of handle() method, the groom will constructe a new instance of
BSPMaster in the local node.
BSPMaster MasterMirror=new BSPMaster(...,isBackup=true,...)

After the MasterMirror is built up then establish the communication protocol to BSPMaster. So that they can
send heartbeat to each other to make sure the opposite side is alive.Meanwhile the MasterMirror will create all
groom proxy and job client proxy with the initialized information.
As long as the BSPMaster is alive, it will send the latest groom status and job info to the MasterMirror ASAP
when data updated. In this way we can ensure the data consistency between BSPMaster and MasterMirror
dynamically to avoid repetitive work in case the master shuts down.
//BSPMaster obtain proxy of its mirror do synchronization when job status updated
mirrorClient=(masterMirrorProtocol)RPC.waitForProxy(...);
mirrorClient.doSync(BSPMasterStatus);

During the job lifecycle if BSPMaster lose the connection to its mirror then elects another MasterMirror as
substitution by repeating the above procedure, else if the MasterMirror lose the connection to its original
then take over as a new BSPMaster  to continue BSPJob working and elect another new MasterMirror and reset
master property in hama configuration.
//MasterMirror init its RPC server waiting for GroomServers request
masterServer = RPC.getServer(this, host, port, conf);
masterServer.start();
//continue the Scheduler work according to the current data maps

When BSPMaster shuts down the groom will lost ping to the master, that means the GroomServer throw an error
exception at doReport method. Since we add a new state for groom,we can return State.BLOCK instead of the error
exception.Once State.BLOCK appears, the groom pause current tasks waiting for the connection from the
MasterMirror within the stipulated time.
pull out the try-catch statement from internal of doReport method to the outside
try{
  doReport(taskStatuses);
 }catch(IOException ioe){
```

```
   LOG.error("Fail to communicate with BSPMaster for reporting.", ioe);
   LOG.info(“Waiting for a heartbeat from the back up BSPMaster...”);
   return State.BLOCK;
 }
...
if (osState == State.BLOCK) {
  // establish the communication link to the back up BSPMaster
  // enroll in the new BSPMaster
}
```