

GOS 3. 2 Install Guide

版本：1.1.0

作者：中国科学院计算技术研究所

时间：2009年4月14日

文档修改历史

版本号	编辑人及单位	编辑时间	备注
0.9	程伯群	2008-1-28	修改了“3.8 证书准备”，添加了2张系统证书需求准备
0.9.1	程伯群	2008-2-29	修改添加了 javadoc 和 tutorial 以及 gosclient 在安装之后的位置. 本版本适用于 GOS 3.0.0
0.9.2	程伯群	2008-3-17	修改了“3. 安装准备”部分描述, 本版本适用于 GOS 3.0.0
1.0.0	吴莎, 乔健, 程伯群	2008-4-10	1. 修订了作业配置管理 2. 在“3.8.1 证书名及目录结构”章节中添加了“证书的文件目录结构” 本版本适用于 GOS 3.0.0
1.0.1	程伯群, 邹永强	2008-5-8	1. 修改证书准备 2. 修改安装配置和安装过程 3. 根据 gsh 的最新修改 review 安装过程. 此次修改适用于 0508 版本。
1.0.2	鲁小亿、熊飞、 林健	2008-12-30	1. 针对 GOS3.2 版的安装包, 进行修改。
1.1.0	鲁小亿, 程伯群, 刘杰, 龙新征, 吴莎, 彭娟, 阚博文等	2009-4-14	1. 更新 HPCG 相关配置 2. 更新了 GOS Core 的相关配置

目录

1. 摘要	6
2. 概述	6
3. 安装准备	7
3.1 MySQL	7
3.2 Ant.....	7
3.3 ftp server	7
3.4 jdk	8
3.5 C 语言开发工具.....	8
3.6 Tomcat.....	8
3.7 本地机群管理系统.....	8
3.8 证书准备.....	8
3.8.1 证书名及目录结构.....	9
3.8.2 FAQ.....	11
4. 安装	12
4.1 安装前确认.....	12
4.2 解压安装包.....	12
4.3 修改配置文件.....	13
4.4 GOS 系统软件安装.....	15
4.4.1 执行安装命令.....	15
4.4.2 拷贝 jdk 和 tomcat 到运行目录.....	15
4.4.3 安装前拷贝.....	16
4.4.4 设置相关密码.....	17
4.4.5 安装拷贝.....	19
4.4.6 自动安装 C 版 gosClient 及部署 C 版 gosShell.....	19
4.5 手动安装 gosclient	22
4.6 检验.....	24
4.6.1 准备.....	24
4.6.2 检验 1: 启动验证.....	24
4.6.3 验证 2: 服务列表查看.....	27
4.6.4 验证 3: 核心验证.....	31
4.6.5 验证 5: HPCG Portal 界面查看.....	32
5. HPCG 配置.....	34
5.1 作业管理配置.....	35
5.1.1 前提.....	35

5.1.2 批作业配置.....	35
5.1.3 关于 LSF.....	36
5.2 文件配置.....	37
5.2.1 FTP 服务器配置.....	37
5.2.2 文件异步传输配置.....	38
5.3 记账配置.....	39
5.3.1 作业日志配置文件 accounting.conf	39
5.3.2 调度器配置文件 quartz_job.xml.....	40
5.3.3 安装节点取消记帐功能模块配置方法.....	44
5.3.4 使用 rsync 同步机群作业管理系统的日志文件，以支持基于作业日志的记帐功能实现.....	44
6. 停止与卸载	46
6.1 停止.....	46
6.2 卸载.....	46
7. 多节点互联	46
7.1 创建网络.....	47
7.2 各节点加入网络.....	47
7.3 各节点退出网络.....	47
7.4 HPCGAgora 创建	47
7.4.1 登录网络.....	48
7.4.2 创建 HPCGAgora 社区	48
7.4.3 检查 HPCGAgora 社区信息同步 daemon 的状态.....	49
7.4.4 停止 HPCGAgora 社区信息同步 daemon.....	49
7.4.5 启动 HPCGAgora 社区信息同步 daemon.....	49
7.4.6 删除已经创建的 HPCGAgora 社区	50
7.4.7 添加一个已存在的社区为 HPCGAgora	50
7.5 Hpcg va	51
7.6 Hpcg msg	51
7.6.1 配置文件 MsgClientConf.xml.....	51
7.6.2 配置文件 MsgClientTopicConf.xml.....	52
7.6.3 Topic or Queue 的添加与删除	53
7.6.4 消息的订阅与发布	53
7.6.5 消息退订	54
8. 记账数据的备份与恢复.....	54
8.1 记账数据的备份.....	54
8.1.1 备份机制和目前功能.....	54
8.1.2 备份数据格式、位置及配置说明	54
8.2 记账数据的恢复.....	54
9. 文档位置	55
9.1 javadoc	55

9.2 manual.....	55
9.3 gosclient.....	56
9.4 GOSTutorial.....	56
10. 附录	57
10.1 附录 1: 安装准备.....	57
10.1.1 硬件需求.....	57
10.1.2 网络需求.....	58
10.1.3 软件需求.....	58
10.1.4 其他需求.....	59
10.1.5 安装规划.....	59
10.2 附录 2: 发布包内容.....	59
10.2.1 安装包大小.....	59
10.2.2 安装包目录结构.....	59
10.2.3 HPCG 安装包包含的组件	60
10.3 附录 3: 部署方式.....	60
10.3.1 HPCG 节点类型	60
10.3.2 HPCG 部署类型	61
10.3.3 关于 IPv6 支持	63

1. 摘要

作为 GOS 的安装手册，本文档描述了 GOS（包括组件 HPCG 和 ignite）的安装、配置、启动、停止、互联等操作的具体步骤和方法。

Note: 一般可以遵照本文档的顺序，依次执行安装过程即可，且每一步安装是否成功都可以检验。

2. 概述

本文档介绍了与 HPCG 系统安装相关的内容。预期读者为 HPCG 本地安装用户以及网络管理员。读者可在本文档中找到以下相关内容：

- 安装准备
- 安装
- 配置
 - 作业管理配置
 - 文件管理配置
 - 记账配置
 - 诊断工具配置
- 启停与卸载
- 多节点互联
 - 消息系统配置
- 附录：
 - 安装准备
 - 发布包内容
 - 部署方式

本文首先描述单个节点的安装、配置、安装检验、运行维护工具的使用以及启停与卸载，然后介绍与多节点互联相关的相关问题。最后在附录中，将详细介绍安装准备、发布包内容以及各种部署方式等。

提示:

在安装过程中，请严格按照本文的叙述顺序按步骤执行。如果哪一步操作失败，请立即停止后续的安装配置操作，将当前出现的问题解决之后，再行后续的操作。

3. 安装准备

详细的安装准备详细信息见附录：《[附录 1：安装准备](#)》

本章节简要描述安装前的准备工作，包括：需要安装配置 MySQL,ant,ftp server，作业管理系统等；需要数据库 root 权限和本机 root 身份等。

本文后续部分将以 GOS3_*_*_*_linux_yyyymmdd.tar.gz 安装包为例

3.1 MySQL

1. 版本：4.1.12 及以上
2. 数据库已经安装并且正常启动
3. 安装过程中需要 mysql 的 root 权限

3.2 Ant

1. 版本：1.6.5 及以上
2. path 设置：设置 ANT_HOME，并且将\$ANT_HOME/bin 目录放到 PATH 路径下

3.3 ftp server

HPCG 的服务端要启动 ftp server。在 RedHat as3 中，配置启动 vsftpd 的参考操作如下：

1. 切换到 root 身份
2. 修改默认的配置文件的配置文件/etc/vsftpd/vsftpd.conf
 - a) 在配置文件的后面加上一行：text_userdb_names=YES,
 - b) 检查是否设置 chroot_local_user=YES:

如果该项没有设置，需要在文件/etc/vsftpd/vsftpd.conf 中作如下设置：

```
chroot_list_enable=YES.
```

```
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
```

同时，需要将 HPCG 网格用户所对应的各个本地用户依次添加到配置文件/etc/vsftpd/vsftpd.chroot_list 中。可以用 vi 编辑保存，也可以通过下面的命令进行操作：

```
echo 本地用户名 >> /etc/vsftpd/vsftpd.chroot_list
```

3. 重启 vsftpd: service vsftpd restart.

3.4 jdk

要求 jdk1.5.0 及以上版本, 设置 JAVA_HOME, 并且将 \$JAVA_HOME/bin 添加到系统 PATH 路径下。

3.5 C 语言开发工具

如果要安装 C 版的 GOS 客户端库, C 版的 gosClient 和 gosShell, 那么要求安装 C 语言开发工具: gcc、yacc、autoconf、make、dos2unix (仅针对 linux 版)。

3.6 Tomcat

安装包自带 tomcat 5.0.28, 安装时自动安装, 启动时自动设置相关环境变量, 无须其他工作。

3.7 本地机群管理系统

HPCG 的服务端必需已经安装有适当的本地机群管理系统 (如 PBSPro、OpenPBS 或 LSF), 或者是已有机群的一个访问节点 (Access Node)。并且, 通过本地安装用户通过机群作业管理系统的本地命令能够提交作业。

3.8 证书准备

GOS 安装包中, 不带有任何证书。如果要安装 GOS, 需要先把证书准备好, 才能安装。

安装一份完整的 GOS 需要 25 张证书, 其中 4 张用户证书, 4 张社区证书, 14 张服务证书, 3 张系统证书。它们分别是:

1. 3 张系统证书
 - gripcontainerscert
 - agorascert
 - resourcescert

2. 14 张服务证书
 - a) Core (6 张服务证书)
 - naming
 - agoraservice
 - userservice
 - GripServerInAxis
 - AxisPing (无需证书)

dds 和 dds:deployer （两张）

b) HPCG（8 张服务证书）

服务：

HPCGMetaInfoService

HPCGFileManageService

resource-usage-service

StatisticAccounting

HPCGAgoraService

以下三个服务在 batchService/services 下：

HPCGJobService

BESService

batchService

3. 4 张用户证书

节点管理员

普通用户

两个测试用户

4. 4 张社区证书

默认社区

两个测试社区

HPCGAgora

3.8.1 证书名及目录结构

如果不要求对服务进行认证，则 14 张服务证书可以省略，只需要 10 张（运管中心需要 11 张）证书。

证书开始必须按照以下要求准备好，放到某个目录下，并且在在 `/${gos.home}/installBase/gosconf.properties` 文件中配置其正确的路径 `certsBaseDir`。

必须的 10 张证书分别描述如下：

1. 3 张系统服务证书（包括 cert 和 key，共六个文件，并且不能有密码）（此处不需要生成代理证书）

要保证以下路径下的以下 cert 文件和 key 文件

`/${certsBaseDir}/resourcescert.pem`

`/${certsBaseDir}/resourceskey.pem`

`/${certsBaseDir}/agorascert.pem`

`/${certsBaseDir}/agoraskey.pem`

```

${certsBaseDir}/gripcontainerscert.pem
${certsBaseDir}/gripcontainerskey.pem

```

2. 1 张节点管理员证书，1 张默认用户证书以及和 1 张默认社区证书
这三张证书先要通过工具生成代理证书。

这三张证书 proxy 同样不能设置密码。

```

${certsBaseDir}/node/admin_proxy.pem
${certsBaseDir}/node/user_proxy.pem
${certsBaseDir}/node/agora_proxy.pem

```

3. 4 张安装测试证书

需要有以下路径下的以下四个名字的代理证书（文件路径和文件名不能改变，代理证书需要预先通过工具生成好，同时也不能有密码设置）

```

${certsBaseDir}/test/agora01_proxy.pem
${certsBaseDir}/test /agora02_proxy.pem
${certsBaseDir}/test /user01_proxy.pem
${certsBaseDir}/test /user02_proxy.pem

```

4. 运管中心要多一张证书，将其生成代理证书之后，用于创建 HPCGAgora

生成代理证书的工具可以从网上找，也可以从 GOS 站点下载：

- <http://vega.ict.ac.cn/gos>
- 从右边导航条“网格系统软件”-->“下载”-->“下载 Proxy 工具”

5. 证书根目录的文件目录结构如下：

```

${certsBaseDir} (目录)
..... agorascert.pem (证书文件，名称和相对路径不能更改)
..... agoraskey.pem (证书文件，名称和相对路径不能更改)
..... gripcontainerscert.pem (证书文件，名称和相对路径不能更改)
..... gripcontainerskey.pem (证书文件，名称和相对路径不能更改)
..... resourcescert.pem (证书文件，名称和相对路径不能更改)
..... resourceskey.pem (证书文件，名称和相对路径不能更改)
.....node(目录)
..... admin_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... user_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... agora_proxy.pem (代理证书文件，名称和相对路径不能更改)
.....test(目录)
..... agora01_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... agora02_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... user01_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... user02_proxy.pem (代理证书文件，名称和相对路径不能更改)
..... (其它代理证书文件)

```

3.8.2 FAQ

- 问：我收到了一批运管中心发过来的一批证书，我应该怎么处理？是不是放到 /home/gos/GOS3_linux_yyyymmdd/all/certs 就可以了？还需要重启容器么？为便于管理，我能否在/home/gos/GOS3_linux_yyyymmdd /all/certs 目录下建立一个目录，把这些证书丢进去。thks!
 - ◆ 此外，这些证书的名字是***_user1.key, ***_user2.key, ***_user3.key 等等，是不是我要在 backend 上面建立名为***_user1, ***_user2,等用户，同时在机群的子节点也要对应建立用户；是否还要把这些用户加入到 GOS-portal 的管理网关中的用户管理里面去？一堆问题啊，呵呵，不好意思，谢谢
- 答：
 - ◆ 收到的这批证书在安装时可以放到 installBase/certs 下，当然可以保留原来的各个证书。
 - ◆ 安装时指定节点管理员、默认用户、默认社区的证书路径时，就可以填 installBase/certs 的绝对路径，这样就会使用新的这些证书来作为上面用户、社区的证书，安装中要求填证书路径时也 用 installBase/certs 的绝对路径，安装后会拷贝到 /home/gos/GOS3_linux_yyyymmdd /all/certs 中。
 - ◆ 其他的多的证书是供测试用的。比如 testInstall 必须要求有两个用户证书和两个社区证书，默认用原来的证书就可以。
 - ◆ 多的证书可以在测试用户和社区时用，因为建用户或者社区都需要新的证书。
 - ◆ 一旦有网格用户想要在本本地机群中提交作业，则必须在机群的子节点也要对应建立本地用户，并登录 hpcgmt 中将此网格用户建立正确的对本地用户的映射，这个在“用户管理”那块来做。
- 问：我现在已经安装好了，如果想换成这批证书，应该怎么办呢？需要重新安装么？或者拷贝到 all/certs 而不是 installBase/certs 是吧？此外，这些证书的名字是***_user1.key, ***_user2.key, ***_user3.key 等等，是不是我要在 backend 上面建立名为***_user1, ***_user2,等用户？谢谢
- 答：如果要换证书确实需要重装，因为不同证书的 DN 不一样，而 GOS 中不允许修改已注册网格用户的 DN。
- 问：这些证书的名字是***_user1.key, ***_user2.key, ***_user3.key 等等，是不是我要在 backend 上面建立名为***_user1, ***_user2,等用户？谢谢
- 答：网格用户名一般要用 email，和证书中的 email 不强制要求对应，但如果对应的应该是一种比较好的命名习惯。本地用户名和证书名也不强制对应。本地用户名的命名可以根据本地管理员的习惯取一些有意义的名字。
- 问：那我应该如何建立证书和用户之间的映射呢？如果不是通过证书名和 email 不强制对应的。我了解底层的运作，以后也好分析问题，谢谢
- 答：
 - ◆ 一个 gos 用户对应一张证书。
 - ◆ 一个 gos 用户要想到某个 site 的机群上交作业的话，需要将该 gos 用户映射到一个本

地用户

- ◆ 在做 gos 用户映射到机群本地用户操作时，不同的 gos 用户必须只能映射到一个唯一的本地用户。具体请参见《HPCG 1.0 Admin Manual》

4. 安装

4.1 安装前确认

- 下载正确的安装包；
- 确认节点满足HPCG软硬件安装需求；
- 确定安装用户对下载和安装HPCG的目录有写的权限；
- 确定是安装为前端机、后端机还是前后端在一台机器上（前后端一体）；
- 对相应机器配置好正确的网络；
- 申请新的证书（共21张证书，其中4张用户证书，4张社区证书，13张服务证书），并存放在HPCG安装用户能访问到的待安装HPCG的节点上；
- 准备两个有效的email，作为网络管理员和默认用户的用户名；
- 成功安装ant 1.6.5，并设置ANT_HOME及Path环境变量
- 如果安装包中没有jdk的话，需要成功安装jdk1.5或以上，并设置JAVA_HOME及Path环境变量
- 安装C语言开发工具：gcc、yacc、autoconf、make、dos2unix（见3.5节所述）
- 如果安装包中没有tomcat的话，需要成功安装tomcat 5.0.28，并设置TOMCAT_HOME或CATALINA_HOME及Path环境变量
- 安装好Mysql，把mysql\bin添加到系统path下并已经启动；
- 如果要使用批作业系统，要预先安装好本地批作业管理系统，并能通过本地命令成功提交作业。比如OpenPBS或者LSF。

如果要安装多个节点并互联，则还需要注意以下事项：

- 确定消息中心的位置
- 各个节点之间按附录1：安装准备的网络要求完成网络互通，尤其注意防火墙的设置；
- 每个节点必须申请和使用不同的证书；
- 每个节点使用的有效email必须不同；

4.2 解压安装包

解压缩安装包到目标路径（该路径即作为系统变量 $\${HPCG.home}$ ）

注意：此路径不能有中文字符，也不能有空格字符

4.3 修改配置文件

修改`{HPCG.home}/installBase/gosconf.properties` 文件。配置文件中有很多配置项，每个配置项是一个 key/value 对。

```
# these lines begin with '#' are comments.
# these configs are 'key = value' pair. commented pair will prompt to user to
input value when install.
# You can change the value as your wish.

# the name of the site, it will be used to identify the site when multiple sites
form a grid.
# It can be not equal to the hostname.
gos_site_name=ict_218

gos_host=10.61.0.218
# the port that GOS Core will listen.
gos_port=38080
# the port that GOS system will listen.
gos_system_port=8080
# You machine's hostname. It should never contains '-' and other special
characters.
gos_hostname=cngrid18

# front machine host information
# The IP address of front machine
gos_front_host=10.61.0.218
# the GOS Core port of front machine that GOS will listen.
gos_front_port=8080
# the GOS System port of front machine that GOS will listen.
gos_front_system_port=8080
# front machine's hostname. It should never contains '-' and other special
characters.
gos_front_hostname=cngrid18

# database infomation
# the database user for GOS
db_user=gos
# set a new password for database user.
#db_pass=gos2
# database root, no need to change
db_super=root
```

```
# please input the database root's current password.
#db_superpass=
# database host IP address
db_host=127.0.0.1
# the port that the database is listening.
db_port=3306

# whether you want to backup database.
opt_backupdb=false
# whether you want to create new database data for GOS.
opt_createdb=true

# set grid admin infomation
# please input an email as grid admin's user name.
gridAdminUsername=Admin_218@cngrid.org
# set a new password for Grid Admin.
#gridAdminPassword=GridAdminPassword

# set default user infomation, Default user is an open user for public use.
#please input an email as default user name.
defaultUsername=User_218@cngrid.org
# set a new password for default user.
defaultUserPassword=pwd

# please set the default agora's name. It's the first part of the agora name,
It must not contain ':'.
defaultAgoraName=Agora_218

#set the certs base direcotry path, including certs directory name
certsBaseDir=/opt/certs
```

注意:

- 1, 如果要把 gos 核心和 hpcg 分开在两个 tomcat 容器下, 则 gos_port 和 gos_system_port 一定不能一样。
- 2, 如果要把 gos 核心和 hpcg 安装在一起, 以 ‘all’ 模式安装时 gos_port, gos_system_port, gos_front_port and gos_front_system_port 必须一样。
- 3, 如果需要多节点互联, 需要保证各个 gos 的管理员用户名和默认用户名不能雷同。
- 4, gos_hostname 的配置项的值必须是运行 hostname 后得到的值。比如:

```
[gos@tu130019 ~]$ hostname
tu130019.ip.tsinghua.edu.cn
```

```
[gos@tu130019 ~]$
```

则此处，“gos_hostname”的值必须设置成“tu130019.ip.tsinghua.edu.cn”

4.4 GOS 系统软件安装

目前HPCG仅支持命令行安装模式。

特别提示：前后端分开安装的时候，`${HPCG.home}/installBase/gosconf.properties` 配置文件必须完全一致。

以下针对 linux 版安装包的安装过程进行解析。

4.4.1 执行安装命令

```
cd $HPCG.home/installBase
```

GOS安装方式有多种（详细见10.3 附录3：部署方式），其命令格式如下：

- 1, 前后端一体
 - a) Gos核心和hpcg分别各放一个tomcat: `./install.sh split`
 - b) Gos核心和hpcg放一个tomcat: `./install all`
- 2, 前后端分开（前后端分开安装，不分先后次序，先装前端机或者后端机器皆可）
 - a) 安装gos前端: `./install all frontend`
 - b) 安装gos后端
 - i. Gos核心和hpcg分别各放在一个tomcat: `./install split backend`
 - ii. Gos核心和hpcg放一个tomcat: `./install all backend`

注意：

- 1, Windows系统下，执行install.bat，Linux系统下，执行./install.sh。
- 2, 运行./install.sh split frontend系统也会按照./install.sh all frontend方式安装。在运行该命令之后，后续将执行验证/文件拷贝/数据库初始化/设置相关密码等操作。
- 3, Linux系统下安装包会默认安装C版gosShell，若希望登录以往Java版的gosShell，需将`${HPCG.home}/gosShell/binary`目录下的文件“gsh”改为其他任意名字，然后按以往方式登录，启动的gosShell既是Java版gosShell。

4.4.2 拷贝 jdk 和 tomcat 到运行目录

无论前后端机器安装，首先将根据用户输入的安装选项，进行jdk和tomcat的拷贝，配置文件的拷贝，tomcat端口的修改以及ant版本的检查。对于安装目录下如果还存在老的证书，还将进行先行删除操作：

```
[gos@paiguzhu installBase]$ ./install.sh all backend
install mode : all backend
base dir is /home/gos/GOS3
```

```
copy and generate gosconf.properties and gosfrontendconf.properties
install MODE is all
installBase is /home/gos/GOS3
    gos use config file : gosconf.properties
the source file /home/gos/GOS3/installBase/gosconf.properties
generate
                                                                    file
/home/gos/GOS3/installBase/install/gos/gosfrontendconf.properties
store it: /home/gos/GOS3/installBase/install/gos/gosfrontendconf.properties
properties file modified OK!

install GOS core and system in the same container
make the /home/gos/GOS3/all folder

no folder /home/gos/GOS3/all exist, make it!
check install, jdk and tomcat folders...

check install files...
copy install files...
check jdk...
copy jdk files...
check tomcat...
copy tomcat files...

port is 8080 system port is 8080
change file /home/gos/GOS3/all/jakarta-tomcat-5.0.28/conf/server.xml's main
port to 8080

ant path is /opt/apache-ant-1.6.5/bin/ant
Apache Ant version 1.6.5 compiled on June 2 2005
check ant success.
begin...
os type:Linux
deleting old certs...
modules/base/src/gosComm/certs
```

4.4.3 安装前拷贝

在正确设置当前日期之后，安装脚本先检查数据库 **MySQL** 是否启动，版本是否满足，就进行

一系列的安裝前拷貝：

```

。。。 以上省略
mysql version info:/usr/bin/mysql Ver 14.7 Distrib 4.1.14, for pc-linux-gnu
(i686) using readline 4.3
db version:4.1.14
JAVA_HOME:/home/gos/GOS3/all/jdk1.5.0_06
TOMCAT_HOME:/home/gos/GOS3/all/jakarta-tomcat-5.0.28
***** THE VERIFICATION IS DONE! *****

***** BEGIN COPY FILES *****
project file:/home/gos/GOS3/all/install/projects.xml
project path:/home/gos/GOS3/all/install/src/gos/gosCompiler
call ant gosCompiler
build file:/home/gos/GOS3/all/install/src/gos/gosCompiler/build.xml
command:ant -buildfile
/home/gos/GOS3/all/install/src/gos/gosCompiler/build.xml
-Djava.home="/home/gos/GOS3/all/jdk1.5.0_06"
-Dgos_webserver.home="/home/gos/GOS3/all/jakarta-tomcat-5.0.28"
-Ddhpcg_webserver.home="/home/gos/GOS3/all/jakarta-tomcat-5.0.28"
-Dgos.home="/home/gos/GOS3/all" -Ddhpcg.home="/home/gos/GOS3/all"
-Dcommonlib.home="/home/gos/GOS3/all/install/commonjar" distribute
Buildfile:
/home/gos/GOS3/all/install/src/gos/gosCompiler/build.xml

distribute:
[copy] Copying 1 file to
/home/gos/GOS3/all/install/gos/modules/base/src/gosComm/conf

BUILD SUCCESSFUL
Total time: 1 second
project path:/home/gos/GOS3/all/install/src/gos/rc
call ant rc
。。。 以下省略很多

```

此过程根据机器的性能以及文件系统的性能，通常会持续大约2分钟。

4.4.4 设置相关密码

在完成安装前文件拷贝之后，系统会提示用户设置或者输入相关密码：

1. SET PASSWORD for Grid Admin xxx：设置网格管理员密码。（其中xxx为配置中填写的管理员用户名）
2. SET PASSWORD for DATABASE user gos2：设置数据库gos2用户的密码

3. CURRENT PASSWORD for DATABASE user root: 输入当前数据库root用户的密码。

- 在配置文件中没有设置管理员和默认用户的密码，那么在安装后端机的时候，会提示输入管理员和默认用户的密码（下图中，默认用户的密码因为设置在配置文件中，所以没有提示输入）：

```

<init>("Load arguments") {
    argument 0: install
    argument 1: backend
}

<init>("Setup class path") {
}

<init>("Load configuration") {
    use config file : gosconf.properties
    java_version : 1.5.0_06
}

installProduct("Install product linux") {
    linux
    x86
base, initStatus, db, frontdb, hotfile, portal
    base("Config GOS2 base") {
    }
SET PASSWORD for Grid Admin Admin_218@cncgrid.org : pwd

    You select Grid Admin's DN = EMAILADDRESS=cbq@software.ict.ac.cn,
CN=node02_Admin, OU=GOS, O=ICT.AC.CN, L=Haidian, ST=Beijing, C=CN
    You select default user's DN = EMAILADDRESS=cbq@software.ict.ac.cn,
CN=node02_User, OU=GOS, O=ICT.AC.CN, L=Haidian, ST=Beijing, C=CN

    You select default agora's DN = EMAILADDRESS=cbq@software.ict.ac.cn,
CN=node02, OU=GOS, O=ICT.AC.CN, L=Haidian, ST=Beijing, C=CN

generated guid for grid admin is A2FD27B6C7443E68C0284BF3139876D7A917AC53
generated guid for default user is A181CCE4FAF60BD41C344D1D156C0F86142ADE8A
generated guid for default agora is 29DE2DFFFA9DE6A879E47B5311754828C42E21B5
generated guid for default group is 3824AF2DB0C3C0A8174AB68A1C93A270FA3188FE
local site url http://10.61.0.63:8080/axis/services/naming
generated guid for local site is 89BDF02F96053533564495EB6928F26BC893A1EB
generate guid for core and system services.

default guids generated.

```

- 如果在配置文件中没有设置数据库相关用户的密码，那么在安装后端机的时候，还会提示输

入数据库用户gos2密码和root密码

```
SET PASSWORD for DATABASE user gos: gos
CURRENT PASSWORD for DATABASE user root: 123456
```

4.4.5 安装拷贝

在设置完所有的配置项值，并且正确完成了安装前拷贝之后，最后将进行真正的安装拷贝。

```
base("Install GOS base") {
    gos_home      =/home/gos/GOS3/all
    java_home     =/home/gos/GOS3/all/jdk1.5.0_06
    catalina_home =/home/gos/GOS3/all/jakarta-tomcat-5.0.28
    axis_home
= /home/gos/GOS3/all/jakarta-tomcat-5.0.28/webapps/axis
    gos_host      =10.61.0.63
    gos_front_host =10.61.0.214
    Copy gos files
    Copy tomcat files
    Install axis
    Copy axis files
    Copy java files
}
```

在安装后端机的时候，还会进行数据库初始化工作：

```
SQL home = /home/gos/GOS3/all/jakarta-tomcat-5.0.28/bin/sql
SQL ...
SQL scheme
SQL grant
SQL data
}
```

4.4.6 自动安装 C 版 gosClient 及部署 C 版 gosShell

上述安装操作执行完成后，接着便进入对C版gosClient的安装，以及C版gosShell的部署。

注意：若linux系统没有安装C语言开发工具：`gcc`、`yacc`、`autoconf`、`make`、`dos2unix`，则不会执行安装C版本gosClient及部署C版gosShell的任务；因此，Windows安装包不会执行这一部分操作。

如果Windows用户需要使用gosClient，那么可以按照4.5节所述，手动安装gosClient工具。Linux安装用户也可以按照4.5节所述在别的机器，或者同一台机器不同目录下另行安装gosClient。

若linux系统已经安装好C语言开发工具，接下来的安装步骤及输出信息如下：

```
==== Install gosclient ====
~/GOS3/installBase ~/GOS3/all/install/gos
~/GOS3/all/install/gos
Using MODE :          all
Using END :
Using BASE :          /home/gos/GOS3
Using INSTALL_BASE : /home/gos/GOS3/installBase/install
Using ALL_HOME :     /home/gos/GOS3/all
Using CORE_HOME :    /home/gos/GOS3/core
Using SYS_HOME :     /home/gos/GOS3/system
Now installing gosclient for all ...
~/GOS3/all ~/GOS3/all/install/gos
Unzipping gosclient ...
Writing 'gosconf.properties' for gosclient ...
Starting gosclient installer ...
/home/gos/GOS3/all/gosclient/installBase
/home/gos/GOS3/all/gosclient/installBase
base dir is /home/gos/GOS3/all/gosclient

copy and generate gosconf.properties and gosfrontendconf.properties
properties file modified OK!

Unzipping gsh files ...
/home/gos/GOS3/all/gosclient/installBase

check install files...
copy install files...

<init>("Load arguments") {
    argument 0: install
}

<init>("Setup class path") {
}

<init>("Load configuration") {
    use config file : gosconf.properties
    java_version : 1.5.0_06
}
```

```

installProduct("Install product linux") {
    linux
    x86
base
    base("Config GOS2 base") {
    }
    base("Install GOS2 base") {
        gos_home      =/home/gos/GOS3/all/gosclient
        java_home     =/home/gos/GOS3/all/jdk1.5.0_06
        log_path      =/home/gos/GOS3/all/logs
        Copy gos files checking whether gcc needs -traditional... no
checking for a BSD-compatible install... /usr/bin/install -c
checking for ar... ar
    ..... 省略
/bin/sh ./support/mkversion.sh -b -S . -s release -d 3.2 -o newversion.h \
&& mv newversion.h version.h
gcc      -DPROGRAM=""bash"    -DCONF_HOSTTYPE=""i686"    -DCONF_OSTYPE=""linux-gnu"
-DCONF_MACHTYPE=""i686-pc-linux-gnu"    -DCONF_VENDOR=""pc"
-DLOCALEDIR="/home/gos/gsh/share/locale"    -DPACKAGE=""bash"    -DSHELL
-DHAVE_CONFIG_H    -I.    -I.    -I./include    -I./lib    -I /home/gos/GOS3/all/jdk1.5.0_06/include    -I
/home/gos/GOS3/all/jdk1.5.0_06/include/linux    -g    -DBUILDTOOL -c -o buildversion.o ./version.c
gcc      -DPROGRAM=""bash"    -DCONF_HOSTTYPE=""i686"    -DCONF_OSTYPE=""linux-gnu"
-DCONF_MACHTYPE=""i686-pc-linux-gnu"    -DCONF_VENDOR=""pc"
-DLOCALEDIR="/home/gos/gsh/share/locale"    -DPACKAGE=""bash"    -DSHELL
-DHAVE_CONFIG_H    -I.    -I.    -I./include    -I./lib    -I /home/gos/GOS3/all/jdk1.5.0_06/include    -I
/home/gos/GOS3/all/jdk1.5.0_06/include/linux    -g    -o    bashversion    ./support/bashversion.c
buildversion.o
    *****
    *
    * GNU bash, version 3.2.0(1)-release (i686-pc-linux-gnu)
    *
    *****

rm -f shell.o
gcc      -DPROGRAM=""bash"    -DCONF_HOSTTYPE=""i686"    -DCONF_OSTYPE=""linux-gnu"
-DCONF_MACHTYPE=""i686-pc-linux-gnu"    -DCONF_VENDOR=""pc"
-DLOCALEDIR="/home/gos/gsh/share/locale"    -DPACKAGE=""bash"    -DSHELL
-DHAVE_CONFIG_H    -I.    -I.    -I./include    -I./lib    -I /home/gos/GOS3/all/jdk1.5.0_06/include    -I
/home/gos/GOS3/all/jdk1.5.0_06/include/linux    -g    -O2 -c shell.c
rm -f eval.o
gcc      -DPROGRAM=""bash"    -DCONF_HOSTTYPE=""i686"    -DCONF_OSTYPE=""linux-gnu"
-DCONF_MACHTYPE=""i686-pc-linux-gnu"    -DCONF_VENDOR=""pc"
-DLOCALEDIR="/home/gos/gsh/share/locale"    -DPACKAGE=""bash"    -DSHELL

```

```
-DHAVE_CONFIG_H -I. -I. -I./include -I./lib -I /home/gos/GOS3/all/jdk1.5.0_06/include -I
/home/gos/GOS3/all/jdk1.5.0_06/include/linux -g -O2 -c eval.c
make[1]: Entering directory `/home/gos/GOS3/all/gosclient/gosShell/gsh-src/builtins'
rm -f mkbuiltins.o
..... 省略
```

在完成以上操作之后，还将删除安装临时文件：

```
remove install files to save disk space...
GOS Client install success!

Copying 'gsh' from 'gosclient/gosShell' to 'gosShell' ...
~/GOS090108/all/install/gos
rst=0

remove install files to save disk space...
```

出现以下提示，表示安装过程全部结束：

```
GOS install success!
```

注：以上安装信息是在linux下安装GOS××× linux版本的信息。

提示：

出现以上提示信息，即表示本地安装拷贝文件以及系统数据库初始化成功完成。但是，是否真正可用，还必须先进行试启动，通过严格的五步检验，完全通过之后，方可进入后续的安装配置过程。

4.5 手动安装 gosclient

安装好 GOS 系统软件后，会在 $\${HPCG.home}/all$ [core]中生成 `gosclient.tar.gz` 和 `gosclient.zip` 两个压缩包，这两个压缩包 GOS 系统软件提供的客户端程序，用户在没有安装 GOS 系统软件的机器上，安装 `gosclient`，并修改好相关配置项后，利用 `gosclient` 登录到某台已经安装了 GOS 系统软件。并启动了 GOS 的节点，执行操作。其中 `gosclient.tar.gz` 是针对 linux 平台的安装包，`gosclient.zip` 是针对 windows 平台的安装包。以下是安装 windows 平台上 `gosclient` 的步骤：

1. 安装工具软件，配置`gosclient`安装环境（与安装GOS系统软件相同）。
2. 在linux机器上，解压缩包`gosclient.tar.gz`；在windows机器上，解压缩包`gosclient.zip`
3. 进入 $\${gosclient}/installBase$ 文件夹，修改`gosconf.properties`配置文件；配置文件中有

四项: `gos_host`: `gosclient`登录到的GOS节点IP; `gos_port`: 登录到GOS节点Core对应的port; `log_path`: `gosclient`运行时log存放位置(默认被注释掉了,那么会在安装`gosclient`过程中要求用户输入日志路径)。 `gos_system_port`: 登录到GOS节点System对应的port。如果GOS节点是split方式(即Core和System是分开安装)安装,那么`gos_port`和`gos_system_port`对应的值不一样,如果GOS节点是all方式(即Core和System是安装在一起)安装,那么`gos_port`和`gos_system_port`对应的值一样。

4. 修改配置好`gosconf.properties`文件后,保存退出执行命令: windows平台能执行 `./install.bat`, linux平台执行 `./install.sh`, `gosclient`便开始安装进程

5. `Gosclient`开始安装后的运行情况如下所示:

```
GRIPMAPDIR ok!
JAVA_HOME ok!
C:\gosclient\installBase
base dir is C:\gosclient
copy and generate gosconf.properties and gosfrontendconf.properties
已复制      1 个文件。
properties file modified OK!
check install files...
copy install files...
C:\gosclient\installBase\install\gos\gos.bat
C:\gosclient\installBase\install\gos\gos.sh
C:\gosclient\installBase\install\gos\gosconf.properties

。。。省略若干行
C:\gosclient\installBase\install\gos\scripts\commands\fixCrLf.bsh
C:\gosclient\installBase\install\gos\scripts\commands\fmpp.bsh
C:\gosclient\installBase\install\gos\scripts\commands\installModule.bsh
C:\gosclient\installBase\install\gos\scripts\commands\isNeededModule.bsh
C:\gosclient\installBase\install\gos\scripts\commands\rmDir.bsh
C:\gosclient\installBase\install\gos\scripts\commands\runModule.bsh
C:\gosclient\installBase\install\gos\scripts\commands\sqlExec.bsh
复制了 257 个文件
<init>("Load arguments") {
    argument 0: install
}

<init>("Setup class path") {
}

<init>("Load configuration") {
    use config file : gosconf.properties
    java_version : 1.5.0_06
SET LOG_PATH for GOS3CLIENT ( ATTENTION: THE PATH MUST BE READABLE AND WRITABLE!
):c:\windows\temp\
```

```
}  
  
installProduct("Install product win32") {  
    windows  
    x86  
base  
    base("Config GOS2 base") {  
    }  
    base("Install GOS2 base") {  
        gos_home      =C:/gosclient  
        java_home     =C:/GOS3-Env-Win32/jdk1.5.0_06  
        log_path      =c:/windows/temp  
        Copy gos files  
    }  
}  
  
remove install files to save disk space...  
GOS Client install success!
```

4.6 检验

4.6.1 准备

启动前请先检查是否有程序占用端口，如果是，需要停掉相关服务。

特别强调：在linux环境下不要以root身份启动HPCG

4.6.2 检验 1：启动验证

一，检查启动进程配置文件\${HPCG.home}/gosprocess.conf

该文件在安装后生成，不同方式安装的情况下，前端机和后端机该文件内容不同，一般无须更改。

以下是 gos 核心和 hpcg 分开在两个 tomcat 安装方式下，后端机的配置文件。key 等于 on 表示在执行 ./gos.sh start 时会启动该进程

```
gos-system=on  
gos-core=on  
gos-activemq=off  
gos-msghelper=off
```

二、启动 gos

进入目录 “\${HPCG.home}”

Windows系统下，执行gos.bat start；Linux系统下，执行gos.sh start

./gos.sh start 会有如下几个过程，过程视不同安装方式不同而不同：

```
[gos@paiguzhu GOS3]$ ./gos.sh
Usage: gos.sh [start/stop/restart/status]
[gos@paiguzhu GOS3]$
```

启动系统的方法为：

```
[gos@paiguzhu GOS3]$ ./gos.sh start
```

运行上表中的启动命令之后，系统将依次启动如下进程：

1. 启动gos core

```
starting gos core
now start up container ...
Using CATALINA_BASE: /opt/GOS3/core/jakarta-tomcat-5.0.28
Using CATALINA_HOME: /opt/GOS3/core/jakarta-tomcat-5.0.28
Using CATALINA_TMPDIR: /opt/GOS3/core/jakarta-tomcat-5.0.28/temp
Using JAVA_HOME: /opt/GOS3/core/jdk1.5.0_06
Please wait while start container ...
now start up naming ...
local naming url is http://10.61.0.218:18080/axis/services/naming
the naming started.
GOS started successfully!
```

2. 启动gos system

```
staring gos sytem
now start up container ...
Using CATALINA_BASE: /opt/GOS3/system/jakarta-tomcat-5.0.28
Using CATALINA_HOME: /opt/GOS3/system/jakarta-tomcat-5.0.28
Using CATALINA_TMPDIR: /opt/GOS3/system/jakarta-tomcat-5.0.28/temp
Using JAVA_HOME: /opt/GOS3/system/jdk1.5.0_06
Please wait while start container ...
now start up naming ...
GOS started successfully!
```

3.启动activemq

```
[hpcg@paiguzhu bin]
starting activemq
$ ACTIVEMQ_HOME: /home/gos/GOS3/all/apache-activemq
```

```

ACTIVEMQ_BASE: /home/gos/GOS3/all/apache-activemq
Loading message broker from: xbean:activemq.xml
INFO BrokerService - Using Persistence Adapter:
AMQPersistenceAdapter(/home/gos/GOS3/all/apache-activemq/data/localhost)
INFO BrokerService - ActiveMQ 5.0.0 JMS Message Broker
(localhost) is starting
INFO BrokerService - For help or more information please
see: http://activemq.apache.org/
INFO AMQPersistenceAdapter - AMQStore starting using directory:
/home/gos/GOS3/all/apache-activemq/data/localhost
INFO ManagementContext - JMX consoles can connect to
service:jmx:rmi:///jndi/rmi://localhost:1099/jmxrmi
INFO KahaStore - Kaha Store using data directory
/home/gos/GOS3/all/apache-activemq/data/localhost/kr-store/state
INFO AMQPersistenceAdapter - Active data files: []
INFO KahaStore - Kaha Store using data directory
/home/gos/GOS3/all/apache-activemq/data/localhost/kr-store/data
INFO TransportServerThreadSupport - Listening for connections at:
tcp://paiguzhu:61616
INFO TransportConnector - Connector openwire Started
INFO BrokerService - ActiveMQ JMS Message Broker
(localhost, ID:paiguzhu-44211-1199597765215-0:0) started

```

如果出现上述提示信息，则表明本步骤通过。

若出现某进程启动失败，可以重新运行./gos.sh start (已启动进程，将不会再重启)

若core启动失败，可以cd core/jakarta-tomcat-5.0.28/bin/目录下运行./gosstart.sh

若system启动失败，可以cd system/jakarta-tomcat-5.0.28/bin/目录下运行./gosstart.sh

若activemq启动失败，可以cd system/apache-activemq/bin目录下运行./start_activemq.sh

或者全部重启:

```
[gos@paiguzhu GOS3]$ ./gos.sh restart
```

三，查看gos状态

具体内容因gosprocess.conf文件内容不同而不同

```

[gos@cngid18 GOS3]$ ./gos.sh status
gos.core is alive
gos.system is alive
activemq is dead
[gos@cngid18 GOS3]$

```

或者通过ps命令查看进程状态

```
[gos@cngid0 bin]$ ps -Af|grep java
```

```

gos          5046          1    5  00:42  ?                00:39:43
/home/gos/GOS3/core/jdk1.5.0_06/bin/java -Dthis.name=GOS.CORE -Xms256m
-Xmx1024m -XX:MaxPermSize=128m -Dgos.home=/home/gos/GOS3/core
-Djava.library.path=/home/gos/GOS3/core/jakarta-tomcat-5.0.28/bin
-Dlog4j.configuration=log4j.
gos          5195          1    0  00:43  ?                00:06:39
/home/gos/GOS3/system/jdk1.5.0_06/bin/java -Dthis.name=GOS.ACTIVEMQ
-Xmx512M -Dorg.apache.activemq.UseDedicatedTaskRunner=true
-Dderby.system.home=/home/gos/GOS3/system/apache-activemq/data
-Dderby.storage.fileSyncTransactionLog=true -Dcom.su
gos          5642          1    0  00:48  ?                00:00:41
/home/gos/GOS3/system/jdk1.5.0_06/bin/java -Dthis.name=MsgHelper
-Dgsh.home=/home/gos/GOS3/system/gosShell
-Djava.library.path=/home/gos/GOS3/system/jakarta-tomcat-5.0.28/bin
org.gos.system.msg.util.LoginGrid org.gos.system.msg
gos          14600         1    5  01:20  pts/7           00:45:08
/home/gos/GOS3/system/jdk1.5.0_06/bin/java -Dthis.name=GOS.SYSTEM -Xms256m
-Xmx1024m -XX:MaxPermSize=128m -Dgos.home=/home/gos/GOS3/system
-Djava.library.path=/home/gos/GOS3/system/jakarta-tomcat-5.0.28/bin
-Dlog4j.configuratio
gos          3477 20883    0 13:56  pts/3           00:00:00  grep java
[gos@cngrid0 bin]$

```

以上两种检验方法只能是从形式上检查进程是否启动。具体启动之后能不能正常工作，还需要做进一步的检验。

4.6.3 验证 2：服务列表查看

一旦通过./gos.sh start完成启动之后，就可以通过浏览器，分别输入网址[http://\\${gos_host}:\\${gos_port}/axis/services](http://${gos_host}:${gos_port}/axis/services)和[http://\\${gos_host}:\\${gos_system_port}/axis/services](http://${gos_host}:${gos_system_port}/axis/services)以及[http://\\${gos_host}:\\${gos_system_port}/batchService/services](http://${gos_host}:${gos_system_port}/batchService/services)，分别出现如下三个表所示的GOS核心服务，HPCG系统服务以及作业服务，则表明HPCG服务器启动成功：

And now... Some Services

- logManage
 - getLog
 - listLogs
 - readfile
 - readfile
 - readfile
 - listfiles
 - removeLog

- o removefile
- vsPingServiceNoSec
 - o ping
- GosCAService
 - o createGosUserFromCA
- AdminService
 - o AdminService
- naming
 - o register
 - o start
 - o …略过
- agoraservice
 - o add
 - o remove
 - o …略过
- pwdservice
 - o addpassword
 - o deletepassword
 - o …略过
- PingServiceSec
 - o ping
- vsPingServiceSec
 - o ping
- userservice
 - o add
 - o remove
 - o …略过
- PingServiceNoSec
 - o ping
- va
 - o parseVAPath
- Version
 - o getVersion
- AxisPing
 - o ping
- GripServerInAxis
 - o control
 - o …略过
- dds
 - o start
 - o …略过

GOS 核心服务列表

And now... Some Services

- va ([wsdl](#))
 - parseVAPath
- AdminService ([wsdl](#))
 - AdminService
- HPCGMetaInfoService ([wsdl](#))
 - searchCertainQueue
 - searchCertainCluster
 - searchLicenseInfo
- Version ([wsdl](#))
 - getVersion
- resource-usage-service ([wsdl](#))
 - extractUsageRecords
 - deleteUsageRecords
 - insertUsageRecords
 - modifyUsageRecords
 - deleteUsageRecordsOthers
 - listExtractNum
 - CCExtractURofSite
- dds ([wsdl](#))
 - start
 - stop
 - renew
 - isStarted
- pwdservice ([wsdl](#))
 - addpassword
 - deletepassword
 - updatepassword
 - inquirepassword
 - inquirepassword
- StatisticAccounting ([wsdl](#))
 - getTotalUsage
 - getStatByTime
 - getStatByTimeS
 - getTotalUsageS
 - getUserStatByTime
 - getUserStatByTimeS
 - getJobDistStatByTime
 - getJobDistStatByTimeS
 - getUserList
 - getUserListS
 - getSiteListInGrid
 - getSiteListInGridS

- getDownloadUrl
- HPCGAgoraService ([wsdl](#))
 - add
 - remove
 - status
 - startHPCGAgoraDaemon
 - stopHPCGAgoraDaemon
 - getAgoraList
-

HPCG除作业服务和文件服务之外的其他系统服务列表

And now... Some Services

- HPCGJobService ([wsdl](#))
 - submitJob
 - queryJob
 - manageJob
- fileService ([wsdl](#))
 - zip
 - copy
 - tail
 - unzip
 - chmod
 - submitAFTJob
 - searchAFTJobs
 - resumeAFTJob
 - pauseAFTJob
 - deleteAFTJob
 - cancelAFTJob
 - move
- BESService ([wsdl](#))
 - CreateActivity
 - GetActivityStatuses
 - TerminateActivities
 - GetActivityDocuments
 - GetFactoryAttributesDocument
- batchService ([wsdl](#))
 - getJobStatus
 - submitJob
 - terminateJob
 - startJob

- Subscribe

HPCG作业服务列表

提示:

如果上述服务列表呈现出现问题，后续步骤将无需继续下去，待问题解决之后再行继续。

4.6.4 验证 3: 核心验证

- 进入目录 “`${gos.home}\gosShell\binary`”。（如果是split方式将core与system安装在不同的container，则进入`${gos.home}/core/`中的gosShell；如果是all方式，则进入`${gos.home}/all/`中的gosShell）
- 在Linux系统下，执行 “`./grun.sh -u XxxAdmin@cngid.org -p password`”；在Windows系统下，执行`grun.bat`，启动grun程序。其中XxxAdmin@cngid.org换成实际的节点管理员的用户名，password换成节点管理员的密码。登录成功后，会进入gosShell的使用界面，看到[gos]提示符。新安装包安装成功后，默认情况下启动的是C语言版本的gosShell，若用户希望进入老版本安装包的Java版gosShell，可以将“`${gos.home}\gosShell\binary`”下的文件“gsh”改为其他任意名字，再执行“`./grun.sh -u XxxAdmin@cngid.org -p password`”登录后，便启动的是Java版的gosShell,看到的提示符为[gos]。
- 在[gos]提示符后，执行命令“`testInstall`”。该运行脚本会对核心的各项功能进行检查，必须所有测试都能运行成功，才能保证系统正常。
- 如果在输出中看到以下提示信息，则表示安装拷贝成功。

```
[gos@cngid15 binary]$ ./grun.sh -u Admin_218@cngid.org -p pwd
log in successfully
'local' into native mode, 'quit' back
[gos]testInstall
.add a agora agora1
.add a user user1@ict.ac.cn
.add ext service OK. resource is AxisPing
.add a agora agora2
.add a user user2@gmail.com
.add ext service OK. resource is AxisPing2
.to check ownership ...
check ownership ok!
。。。此处省略很多
.to remove agora agora1
.to remove agora agora2

Time: 34.862

OK (38 tests)
```

```
[gos]
```

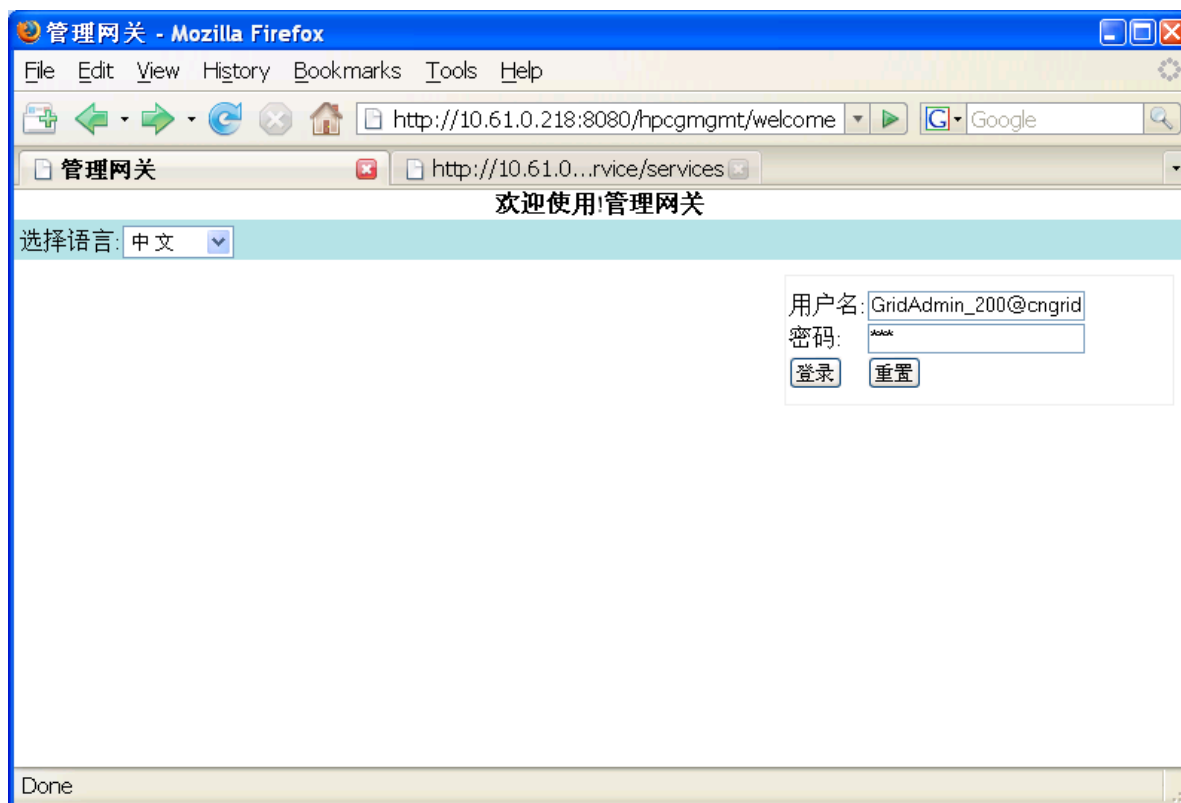
注意：核心测试是否成功关键看最后是否出现 OK(xx tests)字样，其中 xx 表示核心进行的测试项数目。测试中屏幕输出中会有很多异常信息，一般后面都有紧跟着 exception occurred as needed，这个是核心测试时故意引入的错误操作，用以检验核心对各种异常情况的处理是否正确。

4.6.5 验证 5: HPCG Portal 界面查看

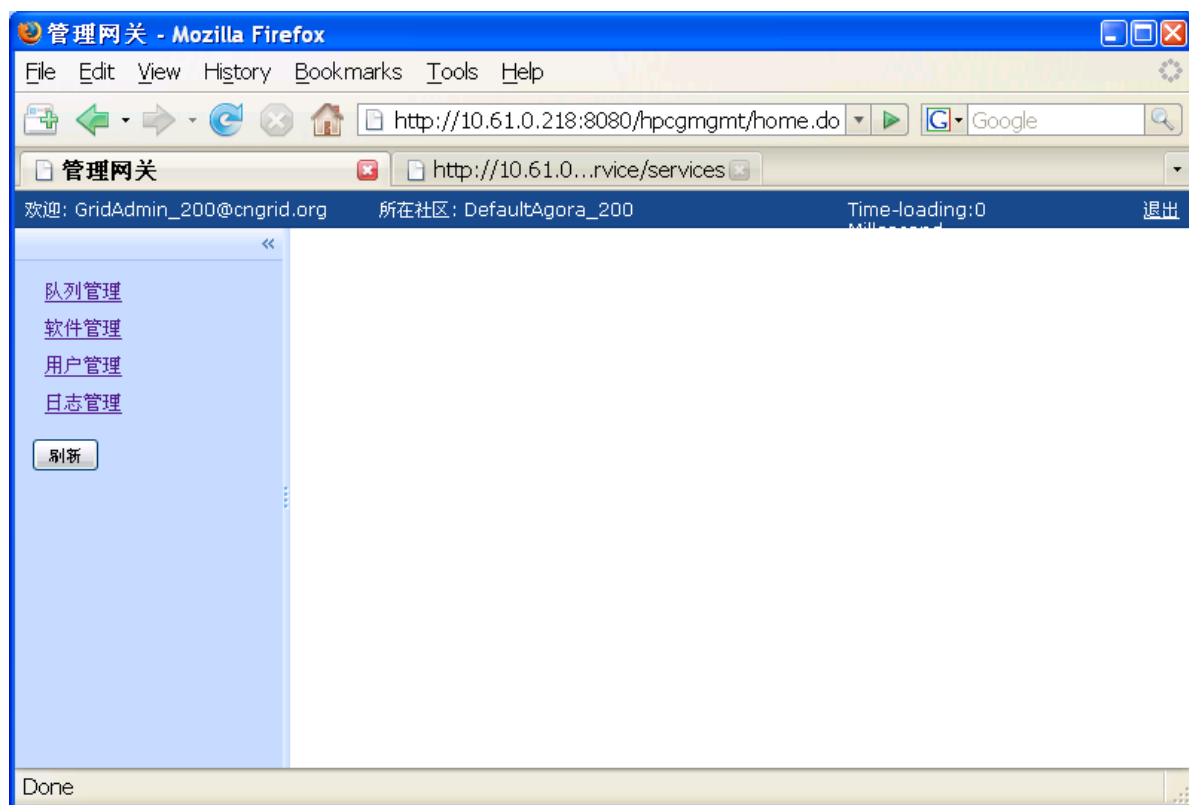
为了进一步检验系统能否正常工作，还需要在安装完前端机并全部正常启动之后，登录 portal 界面查看。

4.6.5.1 HPCG 管理 portal 界面登录查看：

从浏览器打开页面：[http://\\${ gos_front_host}:\\${ gos_front_system_port}/hpcmgmt](http://${ gos_front_host}:${ gos_front_system_port}/hpcmgmt)，其中，[\\${ gos_front_host}](#)和[\\${ gos_front_system_port}](#)分别表示前端机的ip和port

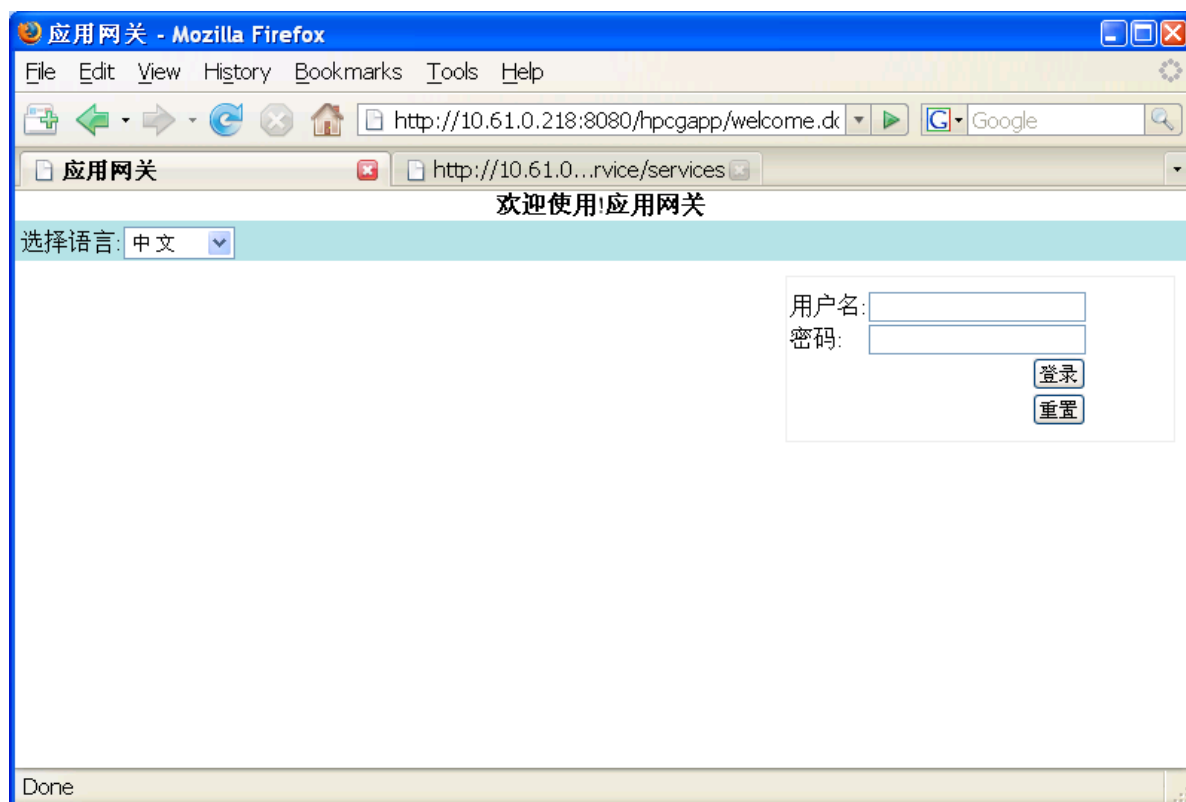


输入网络管理员的全局用户名和密码
见到如下的管理界面，则管理门户通过：



4.6.5.2 HPCG 应用 portal 界面登录查看:

从浏览器打开页面: <http://10.61.0.63:8080/hpcgapp/>,



输入某个全局网络用户的用户名和密码，见到如下的终端用户使用界面，则网关应用门户通过：

提示：

1. 如果本节检验没有通过，则无需继续安装，待检查问题的根源并解决之后，才能进入下一步配置操作。
2. 以上步骤仅仅是完成了基本模块的copy安装和数据库的初始化，对于hpcg的各个功能模块还需要作若干的配置，然后重启，经过进一步检验之后，才能完成整个hpcg的安装配置工作。

5. HPCG 配置

提示：

在进入本章配置之前，建议运行`${hpcg.home}/gos.sh stop`，将 HPCG 先关闭。

5.1 作业管理配置

5.1.1 前提

检查 OpenPBS/LSF 是否已经安装，并且能正常提交和运行作业。具体请参考 OpenPBS 相关使用手册。

5.1.2 批作业配置

在完成后端机的安装之后，若要开启批作业功能，需要在后端机执行以下操作：

1、安装完后进入目录`${HPCG.home}/work/batch/bin/`

- ◆ 将所有其中的.sh脚本赋予执行权限

```
chmod ugo+rx *.sh
```

- ◆ 将`${HPCG.home}/work/batch/bin`的全路径上各个目录都加上读和执行的权限。

eg. HPCG安装在`/opt/HPCG`，则需要执行以下语句

```
chmod ugo+rx /opt
chmod ugo+rx /opt/gos
chmod ugo+rx /opt/gos/work
chmod ugo+rx /opt/gos/work/batch
chmod ugo+rx /opt/gos/work/batch/bin
```

【请注意】：

若您把HPCG安装在了某个用户的home目录下，譬如`/home/gos/HPCG_20080127`，则请特别注意必须为其home目录（如`/home/gos`）添加`ugo+rx`权限。

2、配置 sudoer

- a) 首先切换为 root 身份：

```
su - root
```

- b) 然后使用 visudo 命令编辑/etc/sudoer 配置文件：

```
visudo
```

添加内容如下（user 为安装用户，请在配置时替换为实际的 GOS/HPCG 安装用户）：

```
user ALL=(ALL) NOPASSWD: /bin/cat
user ALL=(ALL) NOPASSWD: /bin/chmod
user ALL=(ALL) NOPASSWD: /bin/sed
user ALL= (ALL) NOPASSWD: /GOS Driver 所在目录/pbs_driver_function.sh *
user ALL= (ALL) NOPASSWD: /GOS Driver 所在目录/file_driver_function.sh *
```

其中：

file_driver_function.sh 为负责文件操作的 driver；而 pbs_driver_function.sh 是负责批作业系统操作的 driver。若您的本地批作业系统是 OpenPBS 之外的其他系统，则需要根据您的系统情况对 pbs_driver_function.sh 进行定制。

【请留意】：linux 系统要求/etc/sudoers 文件的 umask 必须为 440，否则 sudo 将无法正常工作。

3、修改配置文件：gos.home/conf/batch.conf

修改其中如下内容：

- driver.script=pbs_driver_sudo.sh 改成针对自己机群的 driver 脚本
- defaultUserName=hpcg 改成系统上实际存在的用户名
- defaultJobName=DefaultHPCGJob 按自己喜爱修改
- defaultQueueName=small 改成机群上实际存在的一个队列名
- AllowUseHomeDir=true 配置是否使用用户的 home 目录为作业的工作目录
- workingDir=/home/hpcg/1216/all/work/batch/workingDir 改成机群上某个 NFS 共享目录，并保证都有写权限。如果上面的 AllowUseHomeDir 配置项设置为 false，那么工作目录将采用该配置项的内容；如果上面的 AllowUseHomeDir 配置项设置为 true，该配置项的配置将忽略。
- AllowSendJobRecords=true 如需全局作业记录汇总功能，则在配置好消息模块后将其设置为 true。默认将是 false
- AllowConsistencyCheck=true 如需一致性保障机制，则将其设置为 true，默认是 false
- ConsistencyCheckPeriod=300000 设置一致性检查的周期，默认是 1800000 毫秒（半小时），有效范围是 300 秒~3600 秒（5 分钟到 1 小时）

4、若机群部署方式比较特别（譬如没有安装在机群上，或者机群 NFS 目录在各个节点的路径不同，etc），则还需要定制 driver 脚本。具体定制方式请参考《HPCG Driver Development Manual》

5、完成上述配置后，必须重启 GOS 才会生效。

【注意】

对于 OpenPBS 机群系统，在部署之前，请在安装节点上通过终端使用 tracejob 命令查看某个历史 PBS 作业的详情。若 tracejob 命令不可用，则应当配置使用 pbs_driver_function_withoutTracejob.sh 的 driver。具体方法为：将 pbs_driver_function_withoutTracejob.sh 改名并覆盖原有 pbs_driver_function.sh 脚本，并赋予其执行权限。

5.1.3 关于 LSF

前面假设本地作业管理系统是用 OpenPBS，如果作用管理系统是 LSF，则要做如下修改：

- 修改 \${HPCG.home}/conf/batch.conf，修改其中的 driver.script 和 driver.home，指向

```

${HPCG.home}/work/batch/bin/lsf_driver.sh

```

修改前可能如下所示：

- driver.script=pbs_driver_sudo.sh
- driver.home=/home/HPCG2/HPCG/work/batch/bin/

修改后则如下所示：

- driver.script=lsf_driver_sudo.sh
- driver.home=/home/HPCG2/HPCG/work/batch/bin/

完成修改后，返回开始执行从“5.1.2 批作业配置”开始的步骤，并将其中对 pbs_driver 的操作换成对 lsf_driver 操作即可。

5.2 文件配置

当部署方式为 HPCG 后端机或者是 HPCG 前后端一体机的方式时，需要进行文件服务端的配置。服务端的配置内容包括：FTP 服务器的启动和异步文件传输配置两个方面的内容。

5.2.1 FTP 服务器配置

- ◆ 检查 FTP 服务器是否启动，是否能够使用操作系统本地用户进行登录。在 Linux 系统下，只需要启动默认的 vsftp 配置即可。在 Windows 环境下，需要将 FTP 的根目录设置为工作目录的路径根目录，例如工作目录在 F:/HPCG/Workspace/，需要在 F:/ 目录启动 FTP 服务。在 Linux 系统中需要允许本地用户可以切换到其主目录之外的其他目录。

例如在 vsftp 的配置文件 vsftpd.conf 中，如果限制用户主目录的选项开启，即当配置选项：

```
chroot_local_user=YES;
```

需要设置：

```
chroot_list_enable=YES.
```

```
chroot_list_file=/etc/vsftpd/vsftpd.chroot_list
```

将 HPCG 使用的用户放入配置文件/etc/vsftpd/vsftpd.chroot_list。

这样使得该用户能够从/home/开始查找文件路径。

- ◆ FTP 服务指定用户的权限为：读写，删除，文件和目录的建立和删除。在 Linux 默认的 FTP 服务器 vsftp 中添加配置项：

```
text_userdb_names=YES,
```

这样使得获得文件的属主为用户名，而不是 UID。

- ◆ FTP 使用被动模式，使用默认端口 21；

- ◆ IPv6 支持：在 IPv6 的情况下，请使用主动模式。

注：启动 FTP 服务器，修改配置内容需要和机器管理员权限，并配置防火墙。

5.2.2 文件异步传输配置

对于使用用户权限限制的文件异步传输功能，需要配置脚本的权限和 `sudo` 设置以及配置一个权限为 777 的临时工作目录。默认的安装方式只需要设置 `sudo` 就可以了。但是为了避免可能出现的其他方面的问题，建议检查每个配置文件。

5.2.2.1 `sudo` 权限的配置

需要将 HPCG 启动用户配置相应的 `sudo`，例如安装用户为 `gos`，相应的配置为：

```
gos    ALL=(ALL) NOPASSWD: $gos_home /work/batch/bin/ fms_driver_function.sh *
```

5.2.2.2 `$gos_home /work/batch/bin/ fms_driver_function.sh` 脚本执行权限

需要对 `gos` 的运行用户有可执行权限

5.2.2.3 配置文件 `fms-config.conf`

配置文件的位置：

```
$gos_home/conf/fms-config.conf
```

you should not to modify the following four lines!!

1. 文件管理脚本，提供压缩和解压缩等功能；

```
fileDriverPath=/opt/GOS3/system/work/batch/bin/fms_driver_sudo.sh
```

2. 文件异步传输的相关配置：管理员可以设定重传次数和每次的间隔时间的缺省值，其中间隔时间的单位是毫秒

```
org.icenigrd.gridsam.core.plugin.transfer.http.delay=60000
```

```
org.icenigrd.gridsam.core.plugin.transfer.http.retry=0
```

以上两项为 http 的重传次数和重传间隔时间的配置。

```
org.icenigrd.gridsam.core.plugin.transfer.ftp.delay=1000
```

```
org.icenigrd.gridsam.core.plugin.transfer.ftp.retry=10
```

以上两项为 ftp 的重传次数和重传间隔时间的配置。

```
org.icenigrd.gridsam.core.plugin.transfer.sftp.delay=1000
```

```
org.icenigrd.gridsam.core.plugin.transfer.sftp.retry=0
```

以上两项为 sftp 的重传次数和重传间隔时间的配置。

```
org.icenigrd.gridsam.core.plugin.transfer.local.delay=1000
```

```
org.icenigrd.gridsam.core.plugin.transfer.local.retry=0
```

以上两项为本地文件传输的重传次数和重传间隔时间的配置。

```
org.icenigrd.gridSam.core.plugin.transfer.gsiftp.delay=1000
org.icenigrd.gridSam.core.plugin.transfer.gsiftp.retry=0
org.icenigrd.gridSam.core.plugin.transfer.local.security=true
```

注意：脚本请设置在\$gos_home /work/batch/bin/目录下，需要可执行的权限。

5.3 记账配置

5.3.1 作业日志配置文件 accounting.conf

记账模块分为服务端，和web界面前端。其中记账服务部署在HPCG后端机或前后端一体机上。
修改\${tomcat.home}/webapps/axis/WEB-INF/classes/accounting.conf:

1. 把logFileType改为您的批作业管理系统类型（目前支持PBS和LSF）
 2. 把logFileDir改为您的批作业系统的记账日志所在的目录（提醒：该目录中的日志文件应该是HPCG安装用户开放读的权限）
 3. 如果您想包括记账日志目录下的子目录，请把searchSubDirs设为“true”
 4. 如果您希望同步其他节点的记账记录，请把ConsistencyHandlerTag改为“false”
 5. ConsistencyThreshold指同步其他节点的记账记录时，单次同步过程中同步的记录数（在每个同步周期中在每个节点上进行10次这样的单次同步，即一次同步周期之后，同步的记录数为ConsistencyThreshold*10）
 6. 如果您希望节点即时清理垃圾数据，请把autoDeleteTag改为“true”
- 该配置文件示例如下：

```
#the type of logFile, values PBS or LSF
logFileType=PBS
#logFileType=LSF

#the directory contains log file
#logFileDir=/opt/1218/GOS3_linux_20071218/all/jakarta-tomcat-5.0.28/web
apps/axis/pbslog
#logFileDir=/var/spool/torque/server_priv/accounting
logFileDir=/usr/spool/PBS/server_priv/accounting

#if search sub dirs
searchSubDirs=false
```

```
#use a regular expression to shield specific log file(s)
#eg: ^20070* means log file named 20070101-20070930 will be ignored
shield=^20070*

#the DB_URI,user,password
DB_URI=xmldb:exist://localhost:9080/exist/xmlrpc/db
DB_USER=admin
DB_PASSWD=admin

#the mysql
MYSQL_DB_DRIVER_CLASS ,MYSQL_DB_URL,MYSQL_DB_USER,MYSQL_DB_PWD,MY_SQL_D
B_BAK_FILEPATH
MYSQL_DB_DRIVER_CLASS=com.mysql.jdbc.Driver
MYSQL_DB_URL=jdbc:mysql://${db_host}:${db_port}/accounting
MYSQL_DB_USER=${db_user}
MYSQL_DB_PWD=${db_pass}
MY_SQL_DB_BAK_FILEPATH=${gos_home}/mysqlBak

#the max size of sub collection
Threshold=5000

# whether process ConsistencyHandler
ConsistencyHandlerTag=false

#the name of this site
SiteName = ${gos_site_name}

#tha max size of usageRecords pulled from other sites in consistencyCheck
ConsistencyThreshold=50

#config items prepared beforehand
globalUserTag=true

#config items prepared beforehand
autoDeleteTag=false

#config items prepared beforehand
deteleTolerance=
```

5.3.2 调度器配置文件 quartz_job.xml

记账服务定时的去解析批作业系统产生的日志，默认是2分钟进行一次解析。如果您想修改记账服务解析批作业日志的周期，请修改配置文件

`${tomcat.home}/webapps/axis/WEB-INF/classes/quartz_job.xml`，例如：

`<cron-expression>0 0/2 * * * ?</cron-expression>`：每2分钟解析一次

`<cron-expression>0 0/5 * * * ?</cron-expression>`：每5分钟解析一次

`<cron-expression>0 15 10 ? * *</cron-expression>`：每天10: 15启动解析。

现在记账服务中包括有如下几个job，都可以根据相似的方法来修改这些job的启动周期：

```
<?xml version="1.0" encoding="UTF-8"?>
<quartz>
  <job>
    <job-detail>
      <name>BackupJob</name>
      <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.BackupJob</job-class>
      <job-listener-ref>BackupJobListener</job-listener-ref>
    </job-detail>
    <trigger>
      <cron>
        <name>BackupTrigger</name>
        <group>group1</group>
        <job-name>BackupJob</job-name>
        <job-group>group1</job-group>
        <cron-expression>0 0 4 * * ?</cron-expression>
      </cron>
    </trigger>
  </job>
  <job-listener
class-name="org.gos.system.hpcg.accounting.ur.quartz.BackupJobListener"
name="BackupJobListener" />

  <job>
    <job-detail>
      <name>ConsistencyCheckJob</name>
      <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.ConsistencyCheckJob</j
ob-class>
      <job-listener-ref>ConsistencyCheckJobListener</job-listener-ref>
    </job-detail>
    <trigger>
      <cron>
        <name>ConsistencyCheckTrigger</name>
```

```
<group>group1</group>
  <job-name>ConsistencyCheckJob</job-name>
  <job-group>group1</job-group>
  <cron-expression>0 0/11 * * * ?</cron-expression>
</cron>
</trigger>
</job>
<job-listener
class-name="org.gos.system.hpcg.accounting.ur.quartz.ConsistencyCheckJobLi
stener" name="ConsistencyCheckJobListener" />

<job>
  <job-detail>
    <name>AccJob</name>
    <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.AccountingJob</job-cla
ss>
  </job-detail>
  <trigger>
    <cron>
      <name>AccTrigger</name>
      <group>group1</group>
      <job-name>AccJob</job-name>
      <job-group>group1</job-group>
      <cron-expression>0 0/2 * * * ?</cron-expression>
    </cron>
  </trigger>
</job>

<!-- Aggrementation job for statistics data. -->
<!-- -->
<job>
  <job-detail>
    <name>AggreDayToMonthJob</name>
    <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.StatAggeDayToMonthJob<
/job-class>
  </job-detail>
  <trigger>
    <cron>
      <name>AggreDayTrigger</name>
      <group>group1</group>
```

```
<job-name>AggreDayToMonthJob</job-name>
<job-group>group1</job-group>
<cron-expression>0 5 0 1 * ?</cron-expression>
</cron>
</trigger>
</job>
<job>
  <job-detail>
    <name>AggreHalfToYearJob</name>
    <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.StatAggeHalfyearToYear
Job</job-class>
  </job-detail>
  <trigger>
    <cron>
      <name>AggreHalfyearTrigger</name>
      <group>group1</group>
      <job-name>AggreHalfToYearJob</job-name>
      <job-group>group1</job-group>
      <cron-expression>0 23 3 1 1 ?</cron-expression>
    </cron>
  </trigger>
</job>
<job>
  <job-detail>
    <name>AggreMonthToSeasonJob</name>
    <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.StatAggeMonthToSeasonJ
ob</job-class>
  </job-detail>
  <trigger>
    <cron>
      <name>AggreMonthTrigger</name>
      <group>group1</group>
      <job-name>AggreMonthToSeasonJob</job-name>
      <job-group>group1</job-group>
      <cron-expression>0 11 1 1 1,4,7,10 ?</cron-expression>
    </cron>
  </trigger>
</job>
<job>
  <job-detail>
```

```

    <name>AggreSeasonToHalfyearJob</name>
    <group>group1</group>

<job-class>org.gos.system.hpcg.accounting.ur.quartz.StatAggeSeasonToHalfye
arJob</job-class>
  </job-detail>
  <trigger>
    <cron>
      <name>AggreSeasonTrigger</name>
      <group>group1</group>
      <job-name>AggreSeasonToHalfyearJob</job-name>
      <job-group>group1</job-group>
      <cron-expression>0 17 2 1 1,7 ?</cron-expression>
    </cron>
  </trigger>
</job>
<!-- -->
  </quartz>

```

5.3.3 安装节点取消记帐功能模块配置方法

HPCG 1.0中，支持记帐功能的可配置，安装节点可以通过配置去掉记帐功能模块的服务端和客户端。具体操作方法如下：

修改`$(tomcat.home)/webapps/axis/WEB-INF/classes/quartz_job.xml`，注释其中所有内容

5.3.4 使用 rsync 同步机群作业管理系统的日志文件，以支持基于作业日志的记帐功能实现

记帐信息是通过解析机群作业管理系统的日志文件获取的。当HPCG运行在非PBS 或者LSF server主机上时，为了保证记帐信息的实时获取，可以使用rsync同步作业管理系统的日志文件。

Rsync是远程同步（Remote Sync）的简称,是一个提供快速增量式文件同步功能的开源软件。它的特点：

- 1, 可以镜像保存整个目录树和文件系统
- 2, 很容易做到保持原来文件的权限，时间，属主，软硬连接等
- 3, 增量式的同步
- 4, 优化的流程，文件传输速率较高
- 5, 可使用 rcp,ssh 等方式来传输文件，也可直接通过 socket 来进行连接
- 6, 支持匿名传输，方便网站镜像。

下面以 PBS 作业管理系统为例，说明如果使用 rsync 来进行作业日志的同步。配置匿名的文件同步步骤如下：

- 1.在 <http://rsync.samba.org> 下载 rsync 最新版本, 例如 rsync-2.6.9.tar.gz
- 2.分别在 PBS server 所在的主机和 HPCG 所在的主机安装

```
#tar zxf rsync-2.6.9.tar.gz
#cd rsync-2.6.9
#./configure
#make
#su -
#make install
```

- 3.在 PBS server 主机上配置 rsync daemon

```
#su -
# vi /etc/rsyncd.conf
```

```
max connections = 25
syslog facility = local3
```

[PBS]

```
comment = PBS logs
path = /var/spool/torque/server_priv/accounting
read only = yes
list = yes
uid = nobody
gid = nobody
```

- 4.在 PBS server 主机上启动 rsync daemon

```
#rsync --daemon
```

或配置为自动启动:

编辑 /etc/rc.local

添加行

```
rsync --daemon
```

- 5.在 HPCG 运行的主机上配置 Cron 任务, 定时获取同步数据

```
#crontab -e
```

```
#run rsync job every 5 minutes
```

```
* /5 * * * * rsync -a [PBS server host]::PBS/ /home/gos/pbslogs
```

这样, HPCG运行主机就能同步地获取到PBS server上的日志文件了。

6. 停止与卸载

6.1 停止

HPCG前端机，后端机或前后端一体机的停止方式相同，如下所示：

进入`/${gos.home}`，执行`gos.sh stop`，将会停止所有GOS相关进程。

6.2 卸载

在卸载HPCG之前，需要先停止其上HPCG的运行。

删除HPCG的安装目录即可。

如果需要，还要删除HPCG所用的数据库文件。

7. 多节点互联

多个节点可以互联成一个网格。互联必须从某一个节点发起，此节点首先建立网格，并为网格确定一个名字，此节点成为运管中心，此节点的节点管理员即成为网格的拥有者，即网格管理员。其它节点依次加入网格，由系统自动形成拓扑。

互联相关的所有操作都由`gnm`命令完成。

进入目录“`/${gos.home}\gosShell\binary`”。（如果是`split`方式将`core`与`system`安装在不同的`container`，则进入`/${gos.home}/core/`中的`gosShell`；如果是`all`方式，则进入`/${gos.home}/all/`中的`gosShell`），执行`grun.sh`进入`gosShell`界面，此时可以看到`[gos]`提示符。

执行 `gnm` 可以启动此命令。

命令的一般格式为：

```
usage: gnm command parameters [-n namingUrl] [-l] [-h]
```

```
-h print help for the command
```

```
-l give a detailed information
```

```
-n specify the naming URL
```

其中 `command` 表示要执行的具体命令，可以是 `createNetwork`, `join`, `leave`, `list` 等，`parameters` 是特定于此命令的参数，`-u` 指定要操作的 `naming` 系统，如果不指定则使用本地 `naming` 系统。`-l` 表示长格式，会显示比较详细的输出，默认是短格式。

7.1 创建网络

选择一个节点作为运管中心，通过 `createNetwork` 动作创建一个网络，其中填入网络的名字。如下所示，例子中网络的名字是 `CNGrid`。

```
[gos]gnm createNetwork CNGrid
```

如果执行成功，会看到如下信息：

```
successfully create a network CNGrid!
```

7.2 各节点加入网络

主节点创建网络后，其他节点通过 `join` 动作，选择一个已经在网络中的节点（一般就是主节点），与之联系以加入网络。示例命令如下：

```
[gos]gnm join http://10.61.0.216:8080/axis/services/naming
```

例子中选择与 <http://10.61.0.216:8080/axis/services/naming> 联系以加入网络。

一旦加入网络，在一定的时限后，此节点会被网络上的全部节点所感知。

可以用 `list` 命令查看网络上的全部节点，示例如下：

```
[gos] gnm list
```

```
local naming url is http://10.61.0.215:8080/axis/services/naming
```

```
There are totally 2 sites.
```

```
guid = 08663898683FFC90A52D778694D12A56751D107D, name = cngrid16, url =  
http://10.61.0.216:8080/axis/services/naming, status = active
```

```
guid = 17A4E3FA3F26B494C5DE05D2411B8EFEA428CFD6, name = cngrid15, url =  
http://10.61.0.215:8080/axis/services/naming, status = active
```

7.3 各节点退出网络

执行 `leave` 命令离开网络，示例如下：

```
[gos]gnm leave
```

当网络中所有节点都退出后，此网络也就自动销毁。

7.4 HPCGAgora 创建

HPCG 在多节点互连之后，可以创建 HPCGAgora+“节点名”的社区，支持基于 web 的高性能计算。创建过程如下：

- 1.cd \$gos.home/all(system)/gosShell/binary
- 2.用节点管理员的身份登录网格: ./grun.sh
- 3.运行: ghpcga

多个节点互连之后, HPCGAgora 最终用户登录应用 portal, 都会自动进入 HPCGAgora 社区, 完成所要进行的作业、记帐和文件操作。

用法如下:

```
init
destroy
start
stop
status
add agoraName
remove agoraName
```

具体操作步骤示例如下:

7.4.1 登录网格

输入本节点的网格管理员用户名和密码, 出现如下情况后, 表明登录成功。

```
[gos@cngid0 binary]$ ./grun.sh
please input your name: GridAdmin_200@cngid.org
please input password: pwd
log in successfully
'local' into native mode, 'quit' back
[gos]
```

7.4.2 创建 HPCGAgora 社区

在创建过程中需要输入一个有效的社区代理证书文件的绝对路径。

```
[gos]ghpcga init
find file:
/home/gos/GOS3_linux_20080106/system/gosShell/binary/hpcgAgora.1.gsh
exec>> /home/gos/GOS3_linux_20080106/system/jdk1.5.0_06/bin/java
-Xms256m -Xmx512m -Dgos.home=/home/gos/GOS3_linux_20080106/system
-Djava.library.path=/home/gos/GOS3_linux_20080106/system/jakarta-tomcat
-5.0.28/bin org.gos.system.hpcg.hpcgagora.client.HPCGAgoraMain
input agora certs:
/home/gos/GOS3_linux_20080106/installBase/certs/agora/agora50_proxy.pe
m
```

```
add      a      agora      HPCGAgoracngrid0      guid      is
8F833F2E9472D0887F8F2DAA68FFF89A2E664496
Logging is working
Date: Sun Jan 06 18:55:00 CST 2008
Version: Apache-XML-Security-J 1.0.4
[gos]
```

7.4.3 检查 HPCGAgora 社区信息同步 daemon 的状态

在 init 之后，link daemon 是同时启动的。

```
[gos: binary] ghpcga status
GOSAgoraDeamon is alive.
agora in daemon :
      gostest
[gos: binary]
```

7.4.4 停止 HPCGAgora 社区信息同步 daemon

运行 “*ghpcga stop*” 命令，正常情况下，就可以完成 daemon 的停止操作：

```
[gos]ghpcga stop
find                                     file:
/home/gos/GOS3_linux_20080106/system/gosShell/binary/hpcgAgora.l.gsh
exec>>      /home/gos/GOS3_linux_20080106/system/jdk1.5.0_06/bin/java
-Xms256m    -Xmx512m    -Dgos.home=/home/gos/GOS3_linux_20080106/system
-Djava.library.path=/home/gos/GOS3_linux_20080106/system/jakarta-tomcat
-5.0.28/bin org.gos.system.hpcg.hpcgagora.client.HPCGAgoraMain
Logging is working
Date: Sun Jan 06 19:04:33 CST 2008
Version: Apache-XML-Security-J 1.0.4
[gos]
```

7.4.5 启动 HPCGAgora 社区信息同步 daemon

如果想要重新启动 daemon，运行 “*ghpcga start*”，就可以了。

```
[gos]ghpcga start
find                                     file:
/home/gos/GOS3_linux_20080106/system/gosShell/binary/hpcgAgora.l.gsh
exec>>      /home/gos/GOS3_linux_20080106/system/jdk1.5.0_06/bin/java
```

```
-Xms256m -Xmx512m -Dgos.home=/home/gos/GOS3_linux_20080106/system
-Djava.library.path=/home/gos/GOS3_linux_20080106/system/jakarta-tomcat
-5.0.28/bin org.gos.system.hpcg.hpcagora.client.HPCGAgoraMain
Logging is working
Date: Sun Jan 06 19:06:15 CST 2008
Version: Apache-XML-Security-J 1.0.4
[gos]
```

7.4.6 删除已经创建的 HPCGAgora 社区

在 gshell 中，运行 “*ghpcga destroy*”，就可以删除创建的 HPCGAgora 社区。

```
[gos]ghpcga destroy
find file:
/home/gos/GOS3_linux_20080106/system/gosShell/binary/hpcgAgora.1.gsh
exec>> /home/gos/GOS3_linux_20080106/system/jdk1.5.0_06/bin/java
-Xms256m -Xmx512m -Dgos.home=/home/gos/GOS3_linux_20080106/system
-Djava.library.path=/home/gos/GOS3_linux_20080106/system/jakarta-tomcat
-5.0.28/bin org.gos.system.hpcg.hpcagora.client.HPCGAgoraMain
Logging is working
Date: Sun Jan 06 19:08:25 CST 2008
Version: Apache-XML-Security-J 1.0.4
remove agora: guid is 8F833F2E9472D0887F8F2DAA68FFF89A2E664496
[gos]
```

7.4.7 添加一个已存在的社区为 HPCGAgora

在 gshell 中，运行 “*ghpcga add agoraname*”，就可以删除创建的 HPCGAgora 社区。

```
[gos: binary] ghpcga add gostest
add agora gostest to GOSAgoraDaemon successfully.
[gos: binary]
```

删除 HPCGAgora demon 中的社区

```
[gos: binary] ghpcga remove gostest
remove agora gostest from GOSAgoraDaemon successfully.
[gos: binary]
```

7.5 Hpcg va

参见文档：《HPCG-VA User Manual》

7.6 Hpcg msg

消息系统在 HPCG 中提供基于 JMS 的实时信息 pub 与 sub 功能。HPCG 多点互连之后，其全局功能提供，将有赖与消息系统的正常配置与正常运行。

配置消息系统，首先要确定消息中心的位置，然后需要进行一系列的操作：

- 1.消息中心的启动
 - 2.添加 topic 到 GOS 元信息系统中
 - 3.订阅 topic
 - 4.发送消息到 topic
- 其他的还有取消订阅等。

消息系统安装在后端机上：

1. all 方式安装：\$gos.home/all/bin/msgclient
2. split 方式安装：\$gos.home/system/bin/msgclient

7.6.1 配置文件 MsgClientConf.xml

文件\${gos.home}/all(system)/bin/msgclient/conf/MsgClientConf 的内容如下：

```
<org.gos.system.msg.util.MsgClientConf>
  <namingurl>http://10.61.0.63:8080/axis/services/naming</namingurl>
  <desinfos>
    <org.gos.system.msg.util.DesInfo>
      <brokerurl>tcp://10.61.0.212:61616</brokerurl>
      <username>system</username>
      <password>mX5anag4er2P</password>
      <name>hpcg.metainfo</name>
      <topic>true</topic>
      <clientid>636363</clientid>
    <listener>org.gos.system.hpcg.metainfo.monitorinfo.msg.ReceiveMsg</listener>
  </org.gos.system.msg.util.DesInfo>
  <org.gos.system.msg.util.DesInfo>
    <brokerurl>tcp://10.61.0.212:61616</brokerurl>
    <username>system</username>
```

```

        <password>mX5anag4er2P</password>
        <name>hpcg.JobRecords</name>
        <topic>>true</topic>
        <clientid>636363</clientid>
<listener>org.gos.system.hpcg.job.records.HPCGJobRecordsMessageListener
</listener>
    </org.gos.system.msg.util.DesInfo>
    <org.gos.system.msg.util.DesInfo>
        <brokerurl>tcp://10.61.0.212:61616</brokerurl>
        <username>system</username>
        <password>mX5anag4er2P</password>
        <name>hpcg.accounting</name>
        <topic>>true</topic>
        <clientid>636363</clientid>
    </org.gos.system.msg.util.DesInfo>
</desinfos>
<interMilli>20000</interMilli>
</org.gos.system.msg.util.MsgClientConf> >

```

- Namingurl: gos 的 naming 系统的位置，一般无须修改
- desinfos 项指的是将要添加的 queue 或者 topic 的信息
 - ◆ brokerurl 指的是要添加的 queue 或者 topic 的目的地
 - ◆ username/password 指的是消息系统的用户名和密码
 - ◆ Name 是 queue 或者 topic 的名字
 - ◆ topic 为 false 指的是这是一个 queue，否则是 topic。
- interMilli: 如果消息中心暂时不在线，本地消息 daemon 会在间隔设置的时间（单位为毫秒）之后尝试再次连接

7.6.2 配置文件 MsgClientTopicConf.xml

文件\${gos.home}/all(system)/bin/msgclient/conf/MsgClientTopicConf 的内容如下:

```

<org.gos.system.msg.util.MsgClientConfTopic>
    <namingurl>http://${gos_host}:${gos_port}/axis/services/naming</namingurl>
    <desinfos>
        <org.gos.system.msg.util.DesInfo>

            <name>hpcg.metainfo</name>
            <topic>true</topic>
            <clientid>11111</clientid>

        <listener>org.gos.system.hpcg.metainfo.monitorinfo.msg.ReceiveMsg</listener>

```

```
</org.gos.system.msg.util.DesInfo>
<org.gos.system.msg.util.DesInfo>

    <name>hpcg.JobRecords</name>
    <topic>true</topic>
    <clientid>11111</clientid>

    <listener>org.gos.system.hpcg.job.records.HPCGJobRecordsMessageList
ener</listener>
    </org.gos.system.msg.util.DesInfo>

</desinfos>
<interMilli>20000</interMilli>
<ip>${gos_host}:${gos_port}</ip>
</org.gos.system.msg.util.MsgClientConfTopic>
```

7.6.3 Topic or Queue 的添加与删除

本地运行脚本 `$gos.home/all[or system]/bin/msgclient/msgGnode.sh add` 或者 `delete`, 命令来添加上面配置文件中指定的 queue 或 topic,

运行 `msgGnode.sh remove` 命令来删除上面配置文件中指定的 queue 或 topic

本地运行 `msgGnode.sh list` 命令可以列出所有注册到 GOS 的 topic 或者 queue 的名称

本地运行 `msgGnode.sh detail` 命令可以列出注册到 GOS 的 topic 或者 queue 的详细信息

本地运行 `msgGnode.sh -h` 命令可以打印出帮助信息

本地运行 `msgGnode.sh -l` 命令可以列出所有注册到 GOS 的 topic 或者 queue 的名称, 同 `msgGnode.sh list` 命令.

7.6.4 消息的订阅与发布

`MsgHelper.sh` :即消息的订阅与发布.

本地运行脚本 `$gos.home/all[or system]/bin/msgclient/MsgHelper.sh`, 完成 `$gos.home/all[or system]/bin/msgclient/conf/MsgClientTopicConf.xml` 文件中所配置的所有 topic 和 queue 的基于缓存的发布, 以及 `MsgClientTopicConf.xml` 文件中所配置的所有 topic or queue 的订阅.

7.6.5 消息退订

本地运行脚本 `$gos.home/all[or system]/bin/msgclient/unsubHelper.sh`，完成 `$gos.home/all[or system]/bin/msgclient/conf/MsgClientTopicConf.xml` 文件中所配置的所有消息的退订。

8. 记账数据的备份与恢复

8.1 记账数据的备份

8.1.1 备份机制和目前功能

记账数据包括两个部分：记账详细记录数据和统计数据，这两部分数据直接关联。为了防止系统运行过程中出现错误导致数据库停止运行，而造成的数据不一致的问题，我们提供了一套记账数据的自动备份机制。

目前在系统中，记账数据会于每天的凌晨 4:00 去进行自动备份操作，在进行自动备份操作的同时会停止解析日志、往 mysql 里面写数据的操作，备份过程中会自动备份 mysql 里面的数据。

8.1.2 备份数据格式、位置及配置说明

mysql 的备份数据的格式为 `yyyy-MM-dd-THHmss.sql` 格式，备份数据的默认放置位置为 `${gos.home}/mysqlBak`，也可以在 `${gos.home}/jakarta-tomcat-5.0.28/webapps/axis/WEB-INF/classes/accounting.conf` 中更改 `MY_SQL_DB_BAK_FILEPATH` 配置项。

在 `${gos.home}/jakarta-tomcat-5.0.28/webapps/axis/WEB-INF/classes/accounting.conf` 中关于备份还可以配置下列各项：

`DB_URI`—链接 exist 数据库的地址，默认为

`xmldb:exist://localhost:9080/exist/xmlrpc/db`

`DB_USER`—登录 exist 数据库的用户名，默认为 `admin`

`DB_PASSWD`—登录 exist 数据库的密码，默认为 `admin`

`MYSQL_DB_DRIVER_CLASS`—链接 mysql 数据库的驱动名，默认为 `com.mysql.jdbc.Driver`

`MYSQL_DB_URL`—mysql 数据库的链接，默认为 `jdbc:mysql://localhost:3306/accounting`

`MYSQL_DB_USER`—登录 mysql 的用户名 默认为 `gos2`

`MYSQL_DB_PWD`—登录 mysql 的密码，默认为 `gos2`

8.2 记账数据的恢复

备份数据在上一小节中已经详细叙述，用户可根据需要自行选择需要恢复的数据版本，这里记账模块的两部分数据都需要用户去自行手动恢复。

我们利用 mysql 命令的导入功能来恢复数据库，在命令行模式下输入下列命令：

```
>mysql -ugos2 -pgos2 databasename < yyyy-MM-dd-THH:mm:ss.sql
```

注意：记账模块在 mysql 中的数据一般都存在 accounting 数据库里面，所以 databasename 一般为"accounting"，请用户不要随便更改，如果确信要导入另外一个数据库，可根据上述命令的格式更改 databasename。

另外，指定的 databasename 必须是当前系统 mysql 数据库中已经存在的一个数据库名，如果 databasename 不存在的话，此命令将不会成功。

9. 文档位置

下面的\$install_dir 指的是 GOS 的解压根目录,\$tomcat_home 指的是 tomcat 的根目录。

9.1 javadoc

	前后端分开		前后端一体
	前端	后端	
All	<ul style="list-style-type: none"> ● \$install_dir/all/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc 	<ul style="list-style-type: none"> ● \$install_dir/all/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc 	<ul style="list-style-type: none"> ● \$install_dir/all/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc
Split	<ul style="list-style-type: none"> ● \$install_dir/all/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc 	<ul style="list-style-type: none"> ● \$install_dir/core/docs/javadoc ● \$install_dir/system/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc 	<ul style="list-style-type: none"> ● \$install_dir/core/docs/javadoc ● \$install_dir/system/docs/javadoc ● \$tomcat_home/webapps/docs/javadoc

9.2 manual

	前后端分开		前后端一体
	前端	后端	
All	<ul style="list-style-type: none"> ● \$install_dir/all/do 	<ul style="list-style-type: none"> ● \$install_dir/all/do 	<ul style="list-style-type: none"> ● \$install_dir/all/docs/manual

	<ul style="list-style-type: none"> ● cs/manual ● \$tomcat_home/webapps/docs/manual 	<ul style="list-style-type: none"> ● cs/ manual ● \$tomcat_home/webapps/docs/manual 	<ul style="list-style-type: none"> ● \$tomcat_home/webapps/docs/manual
Split	<ul style="list-style-type: none"> ● \$install_dir/all/docs/ manual ● \$tomcat_home/webapps/docs/manual 	<ul style="list-style-type: none"> ● \$install_dir/core/docs/ manual ● \$install_dir/system/docs/ manual ● \$tomcat_home/webapps/docs/manual 	<ul style="list-style-type: none"> ● \$install_dir/core/docs/ manual ● \$install_dir/system/docs/ manual ● \$tomcat_home/webapps/docs/manual

9.3 gosclient

gosclient 提供 zip 包和 tar 包: gosclient.zip, gosclient.tar.gz

	前后端分开		前后端一体
	前端	后端	
All	● 无	● \$install_dir/all/	● \$install_dir/all
Split	● 无	● \$install_dir/core	● \$install_dir/core

9.4 GOSTutorial

GOSTutorial 提供 zip 包和 tar 包: GOSTutorial.zip, GOSTutorial.tar.gz

	前后端分开		前后端一体
	前端	后端	
All	● 无	● \$install_dir/all/docs	● \$install_dir/all/docs
Split	● 无	● \$install_dir/core/docs	● \$install_dir/core/docs

10. 附录

10.1 附录 1：安装准备

1. 软件
 - a) Linux 或者类 linux 系列机器
 - b) 有并且能运行 sun jdk 1.5.*系列
 - c) Mysql: 4.1.12 及以上
 - d) Apache Ant: 1.6.5 及以上
 - e) ftp server: 标准 ftp server, 如 vsftpd。
2. 硬件
3. 网络:
 - a) 后端机:
 - i. 对参与公测的各个网格节点的前后端机进行 ip 授权
 - ii. 对各个节点的前后端机双向开放 4 个端口: 8080, 18080, 61616, ftp
 - b) 前端机:
 - i. 对外开放端口: 8080, ftp
4. 后端机与机群的关系: 是机群的一个 login server
 - a) 用户相通
 - b) 文件相通
 - c) 能运行 qsub 或者 bsub 等机群作业管理软件的命令

10.1.1 硬件需求

1. 后端机:
 - a) 推荐双 P4 或者同级别 AMD, 支持 sun jdk 1.5.*运行
 - b) 推荐 4GB 内存
 - c) 安装磁盘空间要求: <700MB
 - d) 运行磁盘空间要求: 日志文件, 数据库文件
 - e) (100Mbps), 支持 LAN 和 WAN
2. 前端机:
 - a) 推荐 P4 或者同级别 AMD, 支持 sun jdk 1.5.*运行
 - b) 推荐 4GB 内存
 - c) 安装磁盘空间要求: <700MB
 - d) 运行磁盘空间要求: 日志文件, 通过 portal 提交的临时文件
 - e) (100Mbps), 支持 LAN 和 WAN

10.1.2 网络需求

10.1.2.1 IP

- 如果需要被外网访问，都保证HPCG节点具有外部IP。

10.1.2.2 Proxy

- 不推荐使用NAT防火墙，否则可能引起网络问题。

10.1.2.3 防火墙设置

- 后端机或者前后端一体机对互连节点（具体参见7. 多节点互联）的前后端机器开放5个端口：两个http端口,ftp 端口，消息61616端口以及ssh端口
- 前端机对终端用户完全放开2个端口：一个http端口,ftp 端口
- 如果需要被外网访问，前后端之间要求能互相按外网地址访问，或者说，必须要按照外网ip来部署。

10.1.3 软件需求

10.1.3.1 预安装软件

- 操作系统：RedHat Linux 7.3, RedHat Linux 9.0, RedHat Linux Enterprise Linux Advance Server 3 或者RedHat Linux Enterprise Linux Advance Server 4， 或 win32系统（要获得好的IPv6的支持，最好使用Windows Server 2003）
- 安装ant 1.6.5及以上版本，并设置ANT_HOME及Path环境变量
- 需要预先安装jdk1.5或以上，并设置JAVA_HOME及Path环境变量
- 安装C语言开发工具gcc、yacc、autoconf、make、do2unix（仅针对linux版安装）
- 后端机或前后端一体机需要安装数据库：Mysql Version 4.1.12或以上。
- Linux系统的后端机需要安装OpenPBS（推荐版本为：2.3.16）或者其他机群作业管理系统
- 前端机需要提供ftp服务。
- 后端机需要开放ftp服务

10.1.3.2 发布包自带软件

- Apache Axis 1.2RC2

- Apache-activemq-5.0.0
- 其他需要的第三方开发库

10.1.4 其他需求

- 22张证书及部分代理证书（安装包中不带证书）；
- 准备两个有效email地址，分别作为网络管理员和默认用户的用户名。

10.1.5 安装规划

- 确定整个网络互联的拓扑结构，选定主节点；
- 确定整个网络的网络协议：（1）纯IPv4协议（2）纯IPv6协议（3）IPv4 和IPv6混合；
- 确定每个网络节点的安装类型：（1）前端机（2）后端机（3）前后端一体机；
- 确定后端机的安装模式：（1）split（2）all
- 确定每个网络节点的网络协议；
- 确定消息中心节点或者消息中心的Master/Slave节点；
- 确定每个节点的安装用户，安装路径等。

10.2 附录 2：发布包内容

10.2.1 安装包大小

目前只提供 linux 安装包，包名的命名方式为：

GOS3_linux_yyyymmdd.tar.gz

压缩包大小接近 200M

解压之后大小：

安装完之后，按照后端机的 split 方式安装，占用磁盘空间大小约为：700M

10.2.2 安装包目录结构

HPCG 解压后，可以得到如下目录：

```
gos.bat//windows 启动脚本
gos.sh//linux 启动脚本
gosprocess.conf//启动进程配置文件
installBase//安装操作目录
-----gosconf.properties//安装配置文件
-----install//安装包主要内容
-----install.bat//windows 安装脚本
-----install.log//安装日志
-----install.sh//linux 安装脚本
```

```

-----jakarta-tomcat-5.0.28//tomcat 容器
-----jdk1.5.0_06//jdk
-----splitutil//安装脚本

```

图 1. HPCG的发布包目录结构

10.2.3 HPCG 安装包包含的组件

- apache-activemq-5.0.0
- Apache Axis 1.2RC2 and the third-party supporting libraries
- HPCG core, system and application service implementations and common libraries
- Grid Portal implementation on above
- SQL scripts for database initialization and install scripts
- configuration files
- other directories for runtime and work

HPCG 软件发布包自带一系列第三方开发库(如 axis 1.2 RC2 等) 以避免软件版本冲突可能带来的问题。

10.3 附录 3：部署方式

在安装HPCG系统之前，必须事先根据用户的实际需要确定好网格的整体拓扑结构。HPCG整体采用松耦合结构设计和开发，能够非常灵活的部署。

10.3.1 HPCG 节点类型

HPCG标准安装方式是每个站点采用两台机器，分别安装HPCG的前端部分和后端部分，这两台机器相应叫做前端机和后端机。前端机使用一个tomcat容器作为web server，后端机根据安装选项，支持GOS core和system是否是安装在同一个tomcat容器中还是安装在不同的tomcat容器中，以此分担负载，隔离错误和做更细粒度的防火墙设置。

安装前需要对每个站点规划好前端机和后端机，可以用不同的机器做前后端；也可以用同一台机器，但安装在不同的位置，并使用不同的端口。这两种方式安装后会有两个物理的tomcat 容器存在。还可以把前后端功能合并在一起安装，这样会安装在同一个位置，使用同一个tomcat容器，相同端口，这种情况叫做前后端一体机。

HPCG系统部署所在的节点按照功能区分，分为以下三种：

10.3.1.1 前端机

前端机上仅部署tomcat服务器，对外提供HPCG系统的管理portal和应用portal。前端机必须与一个特定的后端机相连，对外提供HPCG服务。

10.3.1.2 后端机

后端机上部署HPCG服务，包括社区（agora）、网程（grip）、naming等核心服务以及作业、记账、文件等系统服务。

用户要通过portal调用后端机上部署的HPCG服务，必须通过与该后端机相连的前端机访问入口，前端机将把调用请求转向后端机。

一个后端机可以与多个前端机相连，通过这些前端机均可以调用这台后端机上的HPCG服务。

10.3.1.3 前后端一体机

前端机和后端机可以部署在同一个节点上，这时称为前后端一体机。HPCG的所有组件，包括核心服务、系统服务和HPCG portal，都部署在一台节点上。

10.3.2 HPCG 部署类型

HPCG系统有两种部署方式：独立部署和互联部署

10.3.2.1 独立部署

一个独立的HPCG后端机或前后端一体机，除了前端机，不与其他任何HPCG后端机或前后端一体机相连，独立对外提供服务，这种方式称为独立部署方式。

网络内的所有前端机和后端机共同对外提供完整的HPCG组件和服务，包括：（1）社区（Agora）服务、网程（Grip）服务、路由（Router）服务、安全模块等核心模块；（2）网络文件系统FMS、批作业系统等系统模块和应用；（3）HPCG Portal。

注意需要在HPCG后端机或前后端一体机上安装防火墙软件保证该节点的安全。

独立部署方式简单，快捷。部署结构如下图所示：

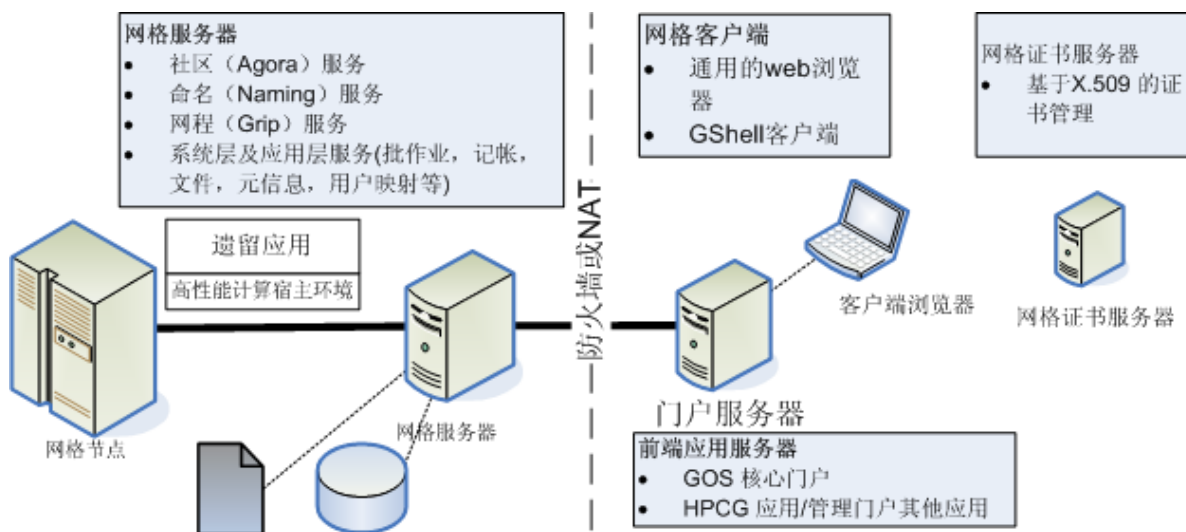


图 2. 独立部署方式结构图

10.3.2.2 互联部署

当系统信息和数据量很大，单个后端机或前后端一体机的性能和存储不足以提供所有服务支持时，可以采用互联部署方式。

互联部署方式是将原有的多个独立的HPCG后端机或前后端一体机互联起来，并选取其中一个充当网络运行中心或者叫主节点，其余所有节点充当从节点，并将其上的部分数据通过HPCG自带的消息服务汇总至主节点。另外，每个后端机需要与一个或多个前端机相连，对外提供portal。通过这种方式部署HPCG系统，称为互联部署方式。这样组成的HPCG系统作为一个整体，共同对外提供服务。

任何已安装好的独立的HPCG后端机或前后端一体机可以通过简单的加入网络命令与指定的主节点相连，成为一个从节点。任何一个已经与主节点相连的从节点也可以通过简单的命令，随时退出网络，成为一个独立的HPCG后端机或前后端一体机。

一个完整的网络系统中有至少一个后端机或前后端一体机用来提供HPCG服务。每一个后端机需要至少有一个前端机与之相连，前端机对外提供HPCG portal，供用户访问部署在其上的HPCG服务和应用。

互联部署结构如下图所示：

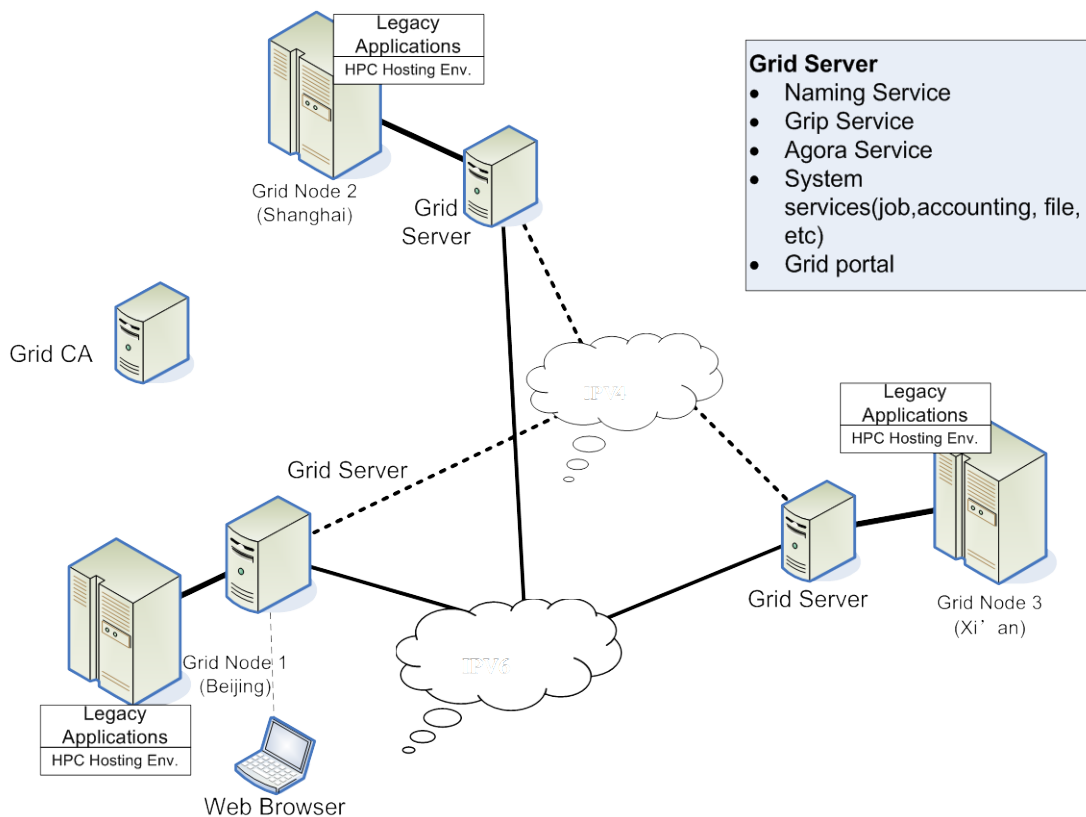


图 3. 互联部署方式

10.3.3 关于 IPv6 支持

HPCG 支持 IPv6，即网络节点可以处于纯 IPv4 网络，或者同时处于 IPv4 与 IPv6 网络。

网络节点都同时处于 IPv4 与 IPv6 网内，这种网络节点的前后端机器都需要双网卡，两个网卡分别有 IPv4 地址和 IPv6 地址，分别连接 IPv4 和 IPv6 网络，走 IPv4 和 IPv6 协议。前端机和后端机都需要双网卡，如图所示。

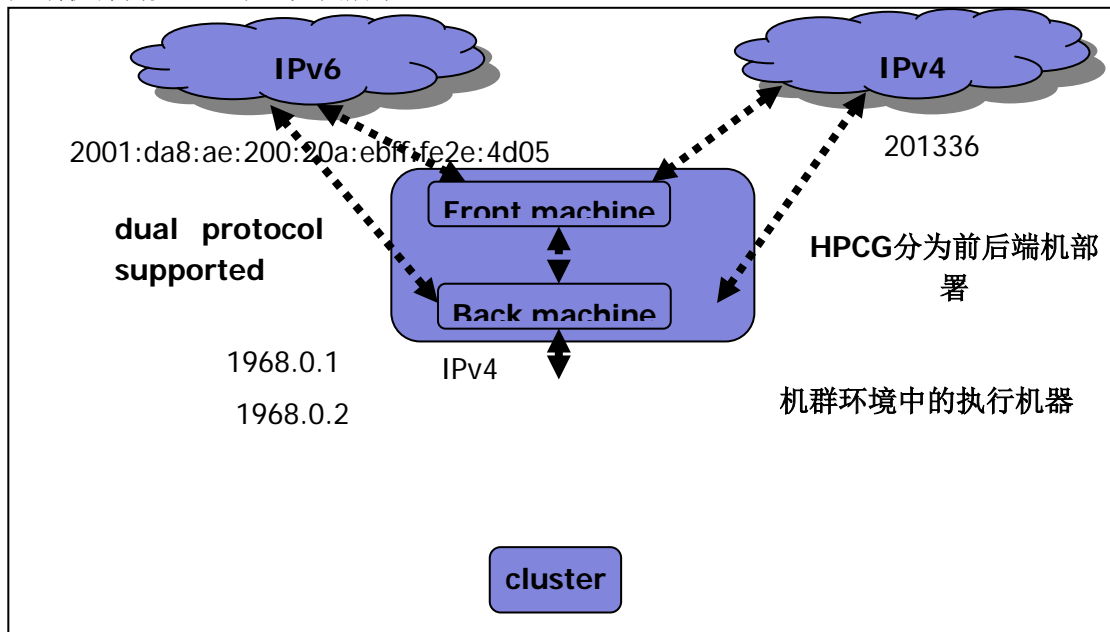


图 4. 单个网络节点的网络配置

图中的节点对于 Windows 和 Linux 网络相关配置方法分别如下。

10.3.3.1.1 Windows 下 IPv6 网络配置方法

Windows Server 2008 和 Windows Vista 默认支持 IPv6。对 Windows Server 2003 和 Windows XP，需要对网络连接添加 IPv6 协议。

以 Windows Server 2003 为例，可以从开始菜单→控制面板→网络连接，选中要修改的连接，点“属性”，在弹出的“本地连接 属性”窗口中，安装协议 Microsoft TCP/IP 版本 6。安装后可以同时看到 Internet 协议(TCP/IP)，即 IPv4；以及 Microsoft TCP/IP 版本 6，即 IPv6。如果 IPv4 和 IPv6 协议前的复选框都处于选中状态，则此网络连接（即网卡）使用双栈。

分别去掉 IPv4 或 IPv6 前的复选框，可以禁用 IPv4 或 IPv6 协议，从而获得纯 IPv6 或纯 IPv4 的协议栈。

当使用双栈时，原来有 IPv4 地址的网络连接会有一个匹配的 IPv6 地址。比如原来有 IPv4 的

地址 10.61.0.45，配成双栈后会有一个 IPv6 地址 fe80::213:72ff:fe2e:391e%6，其中%6 和本机相关，使用时会有诸多问题，所以一般不用这个地址。如果要连接 IPv6 网络，需要给网络连接新设置一个 IPv6 地址(无论是作为客户端还是服务器都需要新设置 IPv6 地址，不能使用 IPv4 地址对应的 IPv6 地址)。

在 Windows 上，可以用以下命令设置连接的 ipv6 地址：

```
netsh interface ipv6 add address 本地连接 1:2:3:4:5:6:7:45
```

其中“本地连接”是用 ipconfig 看到的连接名字。比如 ipconfig 看到这句

```
Ethernet adapter 本地连接:
```

则 Ethernet adapter 之后的为连接名字，如果有空格，在 netsh 命令中需要用””括起来。

其中 1:2:3:4:5:6:7:45 是想要设置的地址值。

同理，可以用 netsh interface ipv6 del address 本地连接 1:2:3:4:5:6:7:45 删除相关地址。

配置成功后，可以用 ping 1:2:3:4:5:6:7:45 测试。

10.3.3.1.2 Linux 下 IPv6 网络配置方法

Linux 下 IPV6 设置方法：

以 root 身份加载 IPV6 模块：

```
#insmod ipv6
```

添加成功后，可以用 lsmod 查看到有 ipv6 模块。

给网卡添加地址，例如 eth0：

```
# ifconfig eth0 add 2001:250:1800:1::88/64
```

修改文件，使其重启机器后自动启动 ipv6：

(1) # vi /etc/rc.local

添加如下两行：

```
ifconfig eth1 add 2001:250:1800:1::88/64 # ipv6 地址
```

```
route -A inet6 add default gw 2001:250:1800:1::1 #ipv6 网关
```

(2) #vi /etc/sysconfig/network

添加如下一行

```
NETWORKING_IPV6=yes
```

配置好后，可以用 ping6 2001:250:1800:1::88 测试是否能正常通信。

10.3.3.2 多个网络节点的网络配置

图中假设有清华、北航、计算所三个节点，其中北航节点按照前后端方式安装。

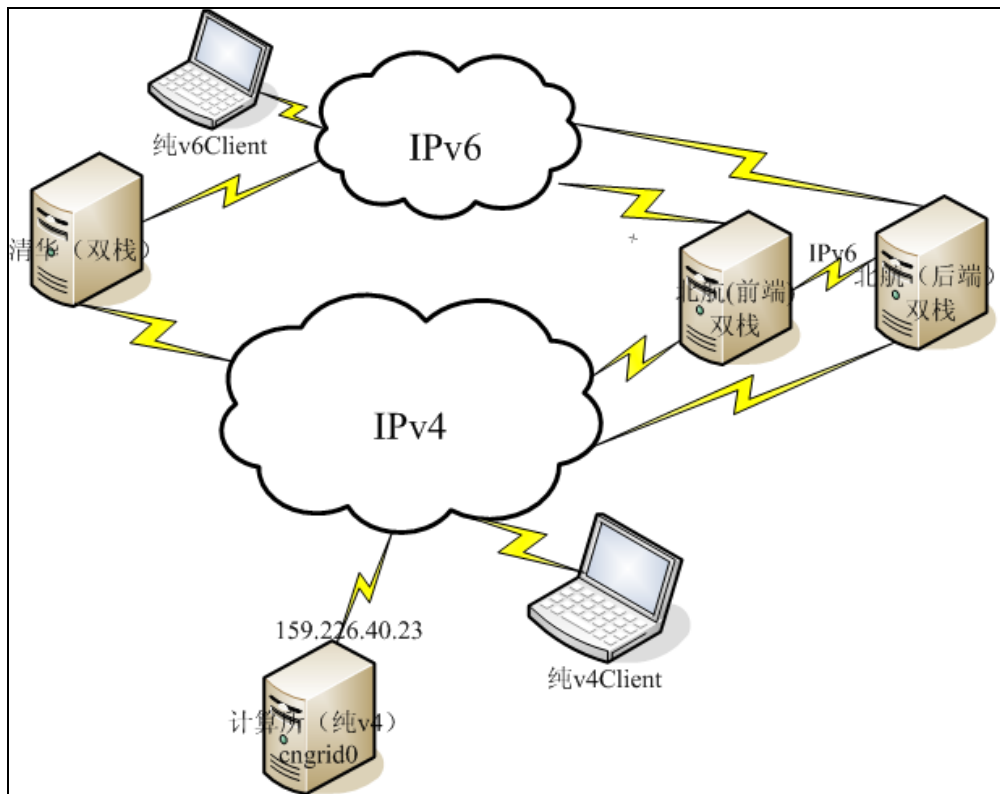


图 5. 支持ipv6的网络配置

每个节点按单节点 IPv6 方式配置好后，互联起来的网络系统的网络拓扑大致如上图所示。
注意：允许纯 IPv4 的机器做网络节点。