

QuickStart: Deploying Platform Symphony 3.1 on HP Clusters



1 Overview.....	2
2 Platform Symphony: Quick Overview	2
2.1 Platform Symphony	3
2.2 Platform EGO	3
2.3 Platform Management Console	3
3 Requirements.....	4
4 Installation	5
4.1 Master Host Installation.....	5
4.2 Management Host Installation.....	9
4.3 Compute Host Installation.....	9
4.4 Optional: Developer's Edition Installation	11
5 Testing	12
5.1 Web Interface	12
5.2 symping.....	13
5.3 SampleApp	15
6 Troubleshooting.....	16
6.1 Symphony/EGO do not start on compute nodes	16
6.2 Web Interface Does Not Load	16
6.3 derbydb never enters STARTED state	17
6.4 SampleApp throws a strange error when running the client	17
6.5 Using ssh Instead of rsh	17
References	18
For More Information	19

1 Overview

This white paper describes the installation and testing of Platform Symphony 3.1 [1] and Platform EGO 1.2.1 [2] on HP Clusters. It does not describe how to manage or use Platform Symphony clusters. We refer the reader to the Symphony documentation [3] for details on how to manage and use the Symphony Grid.

Platform Symphony is a grid scheduler used primarily in the Financial Services Industry [4]. By grid-scheduler we mean that it schedules jobs and tasks as well as assigns resources within a service oriented environment. The jobs are encapsulated in web-services.

Jobs that take advantage of Platform Symphony's capabilities are jobs which can be decomposed into a set of embarrassingly parallel tasks. These tasks are most commonly short pieces of approximately 1-2 seconds of computing but can be longer as well. Developers can grid-enable their applications by having their application call the Symphony API [5] to access services. Symphony jobs and LSF batch jobs can co-exist in the EGO environment.

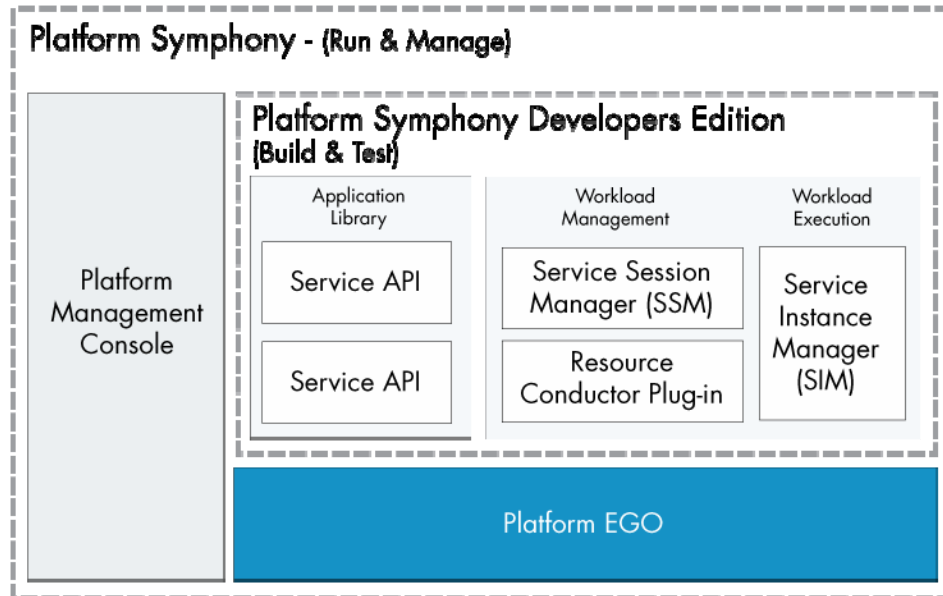
Platform Symphony can improve the performance of individual jobs and decrease overall response time, by distributing long compute applications to the grid.

This white paper is organized as follows: [Section 2](#) gives a short overview of Platform Symphony. We encourage the reader to also read the Symphony [3] and EGO documentation [2]. [Section 3](#) describes the requirements necessary to successfully install Platform Symphony. [Section 4](#), [Section 5](#) and [Section 6](#) present respectively the installation, the testing and the troubleshooting of Platform Symphony and Platform EGO on HP Clusters. The reader is encouraged to look at the EGO documentation [2, 6, 7, 8, 9], the Symphony documentation [1, 3] and to use the support@platform.com mailing list when debugging a problem.

2 Platform Symphony: Quick Overview

The text in this section is taken from the Platform Symphony Architecture documentation [3]. To understand Platform Symphony scheduler, we need to understand how Symphony interacts with other Platform Symphony products such as Platform EGO [2], Platform Management Console [2, 3], and the Platform Symphony Developer Edition [10]. Figure 1 shows how the various components interact.

Figure 1. Symphony 3.1 built on Platform EGO (courtesy of Platform Computing)



2.1 Platform Symphony

Platform Symphony is a service-oriented grid middleware for application orchestration and workload management and runs on top of Platform EGO to carry applications.

2.2 Platform EGO

Platform EGO (Enterprise Grid Orchestrator) is a service-oriented grid infrastructure for resource orchestration and dynamic resource sharing. It provides a central point for managing resource allocations according to hierarchical resource ownership and sharing policies. It provides various built-in system services for grid computing and facilities to run any application services, middleware systems, and workload management systems reliably on grid resources with failover and guaranteed service-level agreements.

2.3 Platform Management Console

The Platform Management Console is a Web interface for monitoring and controlling a cluster and workload. The console includes a dashboard, which provides a quick summary of cluster health and early warning for problems that may occur.

3 Requirements

We suggest performing the following steps before installing Platform Symphony on a cluster or system:

1. Plan a cluster [7, 9]. These Platform documents provide a summary of the steps involved in planning a cluster installation.
2. Check that Symphony supports the systems in the cluster [11].
3. Get the appropriate Platform Symphony 3.1 software suitable for the systems. Download sites as well as temporary license keys are available through a Platform Computing representative.
 - Note that since Symphony supports heterogeneous environments it is necessary to get the appropriate package for each different version of the operating system and system architecture in the target cluster.
 - On Linux, check the operating system, kernel version, system libraries by entering:

```
# uname -a
Linux motie.zko.hp.com 2.6.9-42.EL.lnxhpc.1smp #1 SMP Sun Aug 20
12:49:42 IST 2006 x86_64 x86_64 x86_64 GNU/Linux
```

```
# rpm -aq | grep glibc
glibc-devel-2.3.4-2.25
glibc-kernheaders-2.4-9.1.98.EL
compat-glibc-2.3.2-95.30
glibc-2.3.4-2.25
glibc-devel-2.3.4-2.25
compat-glibc-headers-2.3.2-95.30
glibc-common-2.3.4-2.25
glibc-utils-2.3.4-2.25
glibc-headers-2.3.4-2.25
compat-glibc-2.3.2-95.30
glibc-2.3.4-2.25
glibc-profile-2.3.4-2.25
```

4. On an x86_64 system with the Linux 2.6 kernel the following five files are required to install Platform EGO and Platform Symphony. The files listed below are the latest version of the software at the time the document was published.
 - egocomputehost-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
 - symphony-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
 - ego-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
 - symphonySetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin
 - symcomputehostSetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin
5. The Symphony Developer Edition [10] and the Symphony Client package are located in different rpm files. See [Section 4.3](#) for more information.
6. Obtain a valid Platform Symphony 3.1 License. Platform Computing provides free Symphony demo licenses.
7. Install the operating systems supported by Symphony [11] and assign IP addresses and host names to all the hosts in the cluster. Symphony is a scheduler and does not do bare-metal provisioning.
8. Get root access to all of the systems in the cluster.

9. Make sure communication is enabled between the nodes in the cluster through either `rsh` or `ssh`. The systems must be set up so `rsh` or `ssh` can communicate from one node to the other without being password challenged. If `rsh` is disabled and `ssh` are enabled on the cluster then the directions provided in [Section 6.5](#) will resolve the problem.

4 Installation

We suggest that you create an `egoadmin` user account on all the systems before starting the install procedure so that that user can be used to administer Symphony.

As mentioned in [Section 2](#), a Symphony cluster has two types of nodes: the master node and compute nodes.

[Section 4.1](#) describes how to install Symphony on one x86_64 Linux master host. If a second master host is required for redundancy purposes the instructions in [\[6, 8\]](#) describe the steps to follow. [Section 4.2](#) shows how to create a management host. [Section 4.3](#) shows the installation of Symphony for one x86_64 Linux compute node.

The installation procedures described in these sections are explained in greater detail in:

- *Installing a Single-Host Cluster on Linux* [\[6\]](#)
- *Planning and Installing Your Cluster on Linux* [\[7\]](#)
- *Installing a Single-Host Cluster on Windows* [\[8\]](#)
- *Planning and Installing Your Cluster on Windows* [\[9\]](#)

These documents are available on the installed kit from the `/opt/ego/docs/ego/1.2.1/` directory.

4.1 Master Host Installation

The following steps show how to set up Symphony on the master host [\[6, 7\]](#).

1. Log in to the designated master host as `root`.
2. Copy the installation files (including the `license.dat` license file) onto the host.
 - Placing the installation files in an NFS exported directory makes future install easier so that approach is recommended. This is not necessary for Symphony to successfully install and work but is more convenient since it is not necessary to copy files from one place to another.
3. Check the ports. Symphony requires the following ports:
 - 7869
 - 7870
 - 7871
 - 7872
 - 8080
 - 8005
 - 8009
 - 9090
 - 53
 - 1527

Check for port conflict by using `netstat -a | grep port`; if nothing returns, then the port is free. For more information see the “Check ports” section in *Installing a Single-Host Cluster on Linux* [\[6\]](#) and *Planning and Installing Your Cluster on Linux* [\[7\]](#).

4. By default Symphony is installed in the `/opt` directory. If this is a problem see the “Resolving installation directory conflicts” section in *Installing a Single-Host Cluster on Linux* [6] and *Planning and Installing Your Cluster on Linux* [7].
5. Enter `cd` to get to the subdirectory containing these files.
6. Set the needed environmental variables by entering the following shell commands (making sure to substitute for the correct cluster and hostnames):

```
export CLUSTERNAME=HPCLUSTER /* Define cluster name */
export DERBY_DB_HOST=typhon16 /* Define the database */
```

- Note that `DERBY_DB_HOST` should be set to the master host. It will not work correctly if it is set to a compute host.
- The *Installing a Single-Host Cluster on Linux* [6] states that `DERBY_DB_HOST=hostM` can be set in the cluster properties configuration file. It is our experience that this procedure does not work.

7. Install the Platform Symphony and Platform EGO packages by entering these commands:

```
rpm --prefix /opt/ego -ivh ego-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
rpm --prefix /opt/ego -ivh symphony-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
```

To use the predefined settings for Platform Symphony, install EGO and Symphony with the `symphonySetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin` installation program. The automatic setup will place EGO and Symphony in `/opt/ego` and will use a set of predefined defaults. The setup can be customized by manually installing the `rpm` files.

```
bash-3.00# ./symphonySetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin
```

This program will use following command to install EGO and Symphony RPM to the system.

```
rpm --prefix /opt/ego -ivh ego-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
rpm --prefix /opt/ego -ivh symphony-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
```

Do you want to continue?(yes/no)yes

Extracting files... done!

Start install...

```
Preparing... #####
[100%]
```

The installation will be processed using the following settings:

```
Cluster Name: HPCLUSTER
Cluster Administrator: egoadmin
Connection Base Port: 7869
Embedded Derby DB host: typhon12
Installation Directory: /opt/ego
```

```
1:ego-linux2.6-glibc2.3-x#####
[100%]
```

Platform EGO 1.2.1 is installed at `/opt/ego`.

Remember to use the `egoconfig` command to complete the setup process.

```
Preparing... #####
[100%]
```

```
1:symphony-linux2.6-glibc#####
[100%]
```

Platform Symphony 3.1 is installed at `/opt/ego`.

Once the install is done, it is recommended that readers check for the presence of a file called `ego.cluster.HPCLUSTER`, where `HPCLUSTER` is the specified `CLUSTERNAME` in `/opt/ego/kernel/conf`.

- Invoke the EGO profile by entering this command from the bash shell to set up the EGO/Symphony environment:

```
. /opt/ego/profile.platform
```

- Set proper login permissions for the egoadmin account by issuing this command:

```
egosetsudoers.sh
```

This command is located in

`/opt/ego/1.2/linux2.6-glibc2.3-x86_64/etc/egosetsudoers.sh` but should work without the path since the environment was set up with the `lprevious` source command.

- Optional: Configure EGO to run on system startup by entering this command:

```
/opt/ego/1.2/linux2.6-glibc2.3-x86_64/etc/egostrc.sh
```

- Optional: Configure the system to utilize the EGO profile settings by entering these commands:

```
cp /opt/ego/profile.platform /etc/profile.d/ego.sh  
chmod +x /etc/profile.d/ego.sh
```

- Switch to the egoadmin user account:

```
su egoadmin
```

- Invoke the EGO profile by entering the following command from the bash shell:

```
. /opt/ego/profile.platform
```

- Create the cluster by entering the command below, where `typhon16` is the master host:

```
bash-3.00$ egoconfig join typhon16  
You are about to create a new cluster with this host as the master  
host. Do you want to continue? (y/n)y  
A new cluster HPCLUSTER has been created. The host typhon16 is the  
master host.  
You should run egoconfig setlicense <licensefile> before using the  
cluster.  
bash-3.00$
```

- Configure the licensing for the cluster by entering the following command:

```
egoconfig setlicense license.dat
```

```
bash-3.00$ egoconfig setlicense license.dat  
The license file has been configured.  
bash-3.00$
```

16. Start the software services by entering:
egosh ego start

At this point the master node should be running. It can take a couple of minutes for the necessary services to come up. To check the services on the master node, enter the following command:

egosh service list

```
[egoadmin@typhon16 log]$ egosh service list
SERVICE      STATUS      ALLOC  CONSUMER  RGROUP  RESOURCE  SLOTS  SEQ_NO  STATE  ACTI
derbydb       STARTED     1      /Managem  Managem  typhon16  1      1      RUN    156
plc           STARTED     9      /Managem  Managem  typhon16  1      1      RUN    159
purger        STARTED    10      /Managem  Managem  typhon16  1      1      RUN    160
WEBGUI        STARTED    11      /Managem  Managem  typhon16  1      1      RUN    155
RS            STARTED     5      /Managem  Managem  typhon16  1      1      RUN    157
WebServiceG   DEFINED     /Managem  Managem
SD            STARTED     6      /Managem  Managem  typhon16  1      1      RUN    158
ServiceDire  ALLOCATI   20     /Managem  Managem
[egoadmin@typhon16 log]$
```

To start or stop services it is necessary to be logged in as the Platform Symphony Admin. EGO is not going to let root or egoadmin manage its services. In the following example, egoadmin tries to stop all the services:

```
bash-3.00$ egosh service stop all
Cannot stop service, <Not logged on as cluster admin>
bash-3.00$
```

It is necessary to log in as Admin with password Admin to be able to manage the services:

egosh user logon -x Admin -u Admin

```
bash-3.00$ egosh user logon -x Admin -u Admin
Logged on successfully
bash-3.00$
```

```
bash-3.00$ egosh service stop all
Service [derbydb] has been stopped successfully
Service [plc] has been stopped successfully
Service [purger] has been stopped successfully
Service [WEBGUI] has been stopped successfully
Service [RS] has been stopped successfully
Service [WebServiceGateway] has been stopped successfully
Service [SD] has been stopped successfully
Service [ServiceDirector] has been stopped successfully
bash-3.00$
```

Some of the EGO commands have flags that allow commands to be broadcast to all the nodes. Example of such commands are:

```
egosh ego start all
egosh ego stop all
egosh ego restart all
```

On a cluster with many nodes it is convenient to restart, start, or stop EGO services on all the nodes from one command. See [Section 6.5](#) if you have any issues using the egosh {start, stop, restart} all command on HP clusters as well as if nodes are not showing up on the egosh resource list.

4.2 Management Host Installation

The install procedure for the management host is the same as for the master host (see [Section 4.1](#)). A host with a management role should be set up as a management host and will need access to a shared directory, which contains system files. These files should be accessible to all management hosts. When setting up a management host more setup steps are required, as described in the Install Management Host section in *Installing a Single-Host Cluster on Linux* [6] and *Planning and Installing Your Cluster on Linux* [7].

If multiple management hosts are not needed and no shared directory is required a host can be defined to be a management host by editing the

`/opt/ego/kernel/conf/ego.cluster.HPCLUSTER` file.

Before making `typhon16` a management host:

```
Begin Host
HOSTNAME model type rlm mem swp RESOURCES #Keywords
typhon16 ! ! - - - (linux)
#lemon PC200 LINUX86 3.5 1 2 (linux)
#plum ! NTX86 3.5 1 2 (nt)
End Host
```

After making `typhon16` a management host:

```
Begin Host
HOSTNAME model type rlm mem swp RESOURCES #Keywords
typhon16 ! ! - - - (linux mghost)
#lemon PC200 LINUX86 3.5 1 2 (linux)
#plum ! NTX86 3.5 1 2 (nt)
End Host
```

4.3 Compute Host Installation

The procedure described below should be repeated for each of the compute nodes. If many nodes are involved, scripting the following process is recommended. We recommend that readers also follow the steps in *Installing a Single-Host Cluster on Linux* [6] and *Planning and Installing Your Cluster on Linux* [7].

1. Log in to the compute node as `root`.
2. `cd` to the location where the installation files are located.
3. Check the ports.
The default base connection port is 7869. EGO uses four consecutive ports starting from this base port (7869-7872).
4. Set the needed environmental variable by entering the following command, making sure to substitute for the correct CLUSTERNAME:
export CLUSTERNAME=HPCLUSTER
5. Install the program files by entering the following commands:

```
rpm --prefix /opt/ego -ivh egocompuesthost-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
rpm --prefix /opt/ego -ivh symphony-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
```

- To use the predefined default settings for Platform Symphony, install EGO and Symphony with the `symcomputehostSetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin` installation program. The automatic setup will place EGO and Symphony in `/opt/ego` as well as use a set of predefined defaults. The setup can be customized by manually installing the `rpm` files.

```
[root@typhon12 sym_install]# ./symcomputehostSetup3.1.0_linux2.6-glibc2.3-x86_64-82259.bin
```

```
This program will use following command to install EGO and Symphony RPM to the system.
rpm --prefix /opt/ego -ivh egocomputehost-linux2.6-glibc2.3-x86_64-1.2.1-82259.rpm
rpm --prefix /opt/ego -ivh symphony-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
Do you want to continue?(yes/no)yes
Extracting files... done!
Start install...
Preparing... ##### [100%]
```

```
The installation will be processed using the following settings:
Cluster Name: cluster1
Cluster Administrator: egoadmin
Connection Base Port: 7869
Installation Directory: /opt/ego
```

```
1:egocomputehost-linux2.6##### [100%]
```

```
Platform EGO 1.2.1 (compute host package) is installed at /opt/ego.
Remember to use the egoconfig command to complete the setup process.
Preparing... ##### [100%]
```

```
1:symphony-linux2.6-glibc##### [100%]
```

```
Platform Symphony 3.1 is installed at /opt/ego.
```

- Once the installation is completed, it is recommended to, check for the presence of a file called `ego.cluster.HPCLUSTER`, where `HPCLUSTER` is the specified `CLUSTERNAME` in `/opt/ego/kernel/conf`.
- Invoke the EGO profile by entering this command to set up the EGO/Symphony environment (e.g. the bash shell):

```
. /opt/ego/profile.platform
```

- Set proper login permissions for the `egoadmin` account by entering this command:

```
egosetsudoers.sh
```

This command is located in `/opt/ego/1.2/linux2.6-glibc2.3-x86_64/etc/egosetsudoers.sh` but should work without the path since the environment was set up with the last command.

- Optional: Configure EGO to run on system startup by entering this command:
`/opt/ego/1.2/linux2.6-glibc2.3-x86_64/etc/egosetrc.sh`
- Optional: Configure the system to utilize the EGO profile settings by entering these commands:
`cp /opt/ego/profile.platform /etc/profile.d/ego.sh`
`chmod +x /etc/profile.d/ego.sh`
- Switch to the `egoadmin` user account.
- Invoke the EGO profile by entering (e.g. in the bash shell):
`. /opt/ego/profile.platform`

14. Create the cluster by entering these commands:

```
egoconfig join typhon16 where typhon16 is the master node
```

```
bash-3.00$ egoconfig join typhon16
```

```
You are about to join this host to cluster HPCLUSTER. Do you want to
continue? (y/n)
```

```
y
```

```
The host typhon12 has joined the cluster HPCLUSTER.
```

15. Start the software services by entering the following command:

```
egosh ego start
```

16. At this point the compute node should be running. The following command checks the state of the service on the node:

```
bash-3.00$ egosh service list
```

SERVICE	STATUS	ALLOC	CONSUMER	RGROUP	RESOURCE	SLOTS	SEQ_NO	STATE	ACTI
derbydb	STARTED	1	/Managem	Managem	typhon16	1	1	RUN	30
plc	STARTED	2	/Managem	Managem	typhon16	1	1	RUN	31
purger	STARTED	3	/Managem	Managem	typhon16	1	1	RUN	32
WEBGUI	STARTED	4	/Managem	Managem	typhon16	1	1	RUN	33
RS	STARTED	5	/Managem	Managem	typhon16	1	1	RUN	34
WebServiceG	DEFINED		/Managem	Managem					
SD	STARTED	6	/Managem	Managem	typhon16	1	1	RUN	28
ServiceDire	DEFINED		/Managem	Managem					

```
bash-3.00$
```

17. By looking in the status column, we can see that all the services are up and running.

18. Enter `egosh resource list` to see the nodes in the cluster:

```
bash-3.00$ egosh resource list
```

NAME	status	mem	swp	tmp	ut	it	pg	rlm	r15s	r15m	ls
typhon12	ok	15G	16G	26G	0%	0	0.7	0.0	0.0	0.0	2
typhon16	ok	14G	6142M	4924M	0%	0	15.3	0.0	0.0	0.1	4

```
bash-3.00$
```

19. Some of the EGO commands have flags that let commands be broadcast to all of the nodes.

Example of such commands are:

```
egosh ego start all
```

```
egosh ego stop all
```

```
egosh ego restart all
```

On a cluster with many nodes it is convenient to restart, start, or stop EGO on all the nodes from one command. See [Section 6.5](#) if problems are encountered using the `egosh {start, stop, restart} all` command on HP clusters or if the nodes are not showing up on the `egosh resource list`.

4.4 Optional: Developer's Edition Installation

In order to compile and run applications on the Symphony grid, it will be necessary to install the Developer's Edition of Symphony [10] so that necessary headers and sample programs are available. The Developer's Edition is only needed on the node where programs are compiled and not on all the nodes within the cluster. To install the Developer's Edition, follow these steps:

- Log in as the root user on the master node, navigate to the directory containing the installation files and enter the following command:

```
bash-3.00# export CLUSTERADMIN=egoadmin
```

```
bash-3.00# rpm -ivh symphonyDE-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm
```

```
Preparing... ##### [100%]
```

```
1:symphonyDE-linux2.6-gli##### [100%]
Platform Symphony Developer Edition is installed. To test your installation
and run a sample client and service, refer to "Testing your installation"
in /opt/symphonyDE/docs/symphonyde/3.1/index.html.
bash-3.00#
```

- To restart EGO, enter:
egosh ego restart all.

The Developer's Edition contains excellent code examples that test the deployment and the enabling and disabling of services. We decided to focus on the C++ examples but similar examples are available for Java and .NET. The following examples and tutorials are highly recommended and included in the kit:

- /opt/symphonyDE/3.1/samples/CPP
- /opt/symphonyDE/3.1/index.html (Platform Knowledge Center)
- C++ tutorials are very helpful

The default EGO/Symphony `symping` test program is of limited interest but it can be used to confirm that the cluster is working correctly. The `symping` testing is described in [Section 5.2](#). The testing done with one of the sample applications, `SampleApp`, is described in [Section 5.3](#).

5 Testing

At this point the EGO and Symphony packages should be installed on the systems. The following sections describe some of the tests completed to verify the expected functionality of the grid.

5.1 Web Interface

First verify that the Web interface is accessible to the outside world. From a workstation, open a web browser to `http://<cluster_hostname>:8080/platform/`. The login screen is similar to the one shown in Figure 2.

Figure 2. Platform Management Console



If the page does not load, review [Section 6.1](#).

Log in to the system using the user and password combination `Admin` and `Admin`. Note that this is case-sensitive. Disable the pop-up blocker in the browser, or enable an exception for this website. Once logged in, a variety of system settings and view reports can be examined. For more information on the Platform Management Console and on how to setup Symphony see [6, 7].

5.2 symping

The second recommended test is to run the `symping` example application that comes with Symphony. Its purpose is to ensure that there is communication between the master and compute hosts.

Log in to the master host and enter the following command:

symping

`symping` returns a screen-full of information. In the following output, we can determine where the job ran by looking in the last table for a field titled `ExecHostName`. Note that the jobs were run on hosts different from the master/management host.

```
[egoadmin@typhon16 3.1]$ symping  
Symphony Ping Utility.
```

Use this utility to test and verify that components are working and responsive.

Vary the workload to try out different Symphony workload conditions. This client/service was developed using Symphony Developer Edition APIs.

```
-----  
Client connecting to Symphony...  
Connected to Session Director.  
Connection information:  
User: Admin  
Application: symping
```

Connected to Session Manager.
Creating 1 session...
Session created. ID: 2

Test Run 1.

Send 20 tasks and retrieving replies...

ExecHostName	SI-PID	TaskID	SubmitTime	StartTime	EndTime	RecvTime	
typhon12	19132	1	16:20:26.896	20:26.896	20:26.948	20:26.950	53
typhon12	19132	2	16:20:26.897	20:26.949	20:27.001	20:27.002	105
typhon12	19132	3	16:20:26.897	20:27.002	20:27.054	20:27.055	158
typhon12	19132	4	16:20:26.898	20:27.055	20:27.107	20:27.108	210
typhon12	19132	5	16:20:26.898	20:27.107	20:27.159	20:27.160	261
typhon12	19132	6	16:20:26.899	20:27.160	20:27.212	20:27.213	314
typhon12	19132	7	16:20:26.899	20:27.213	20:27.265	20:27.266	367
typhon12	19132	8	16:20:26.900	20:27.266	20:27.318	20:27.319	419
typhon12	19132	9	16:20:26.900	20:27.319	20:27.371	20:27.372	471
typhon12	19132	10	16:20:26.901	20:27.371	20:27.423	20:27.424	523
typhon12	19132	11	16:20:26.901	20:27.423	20:27.475	20:27.476	574
typhon12	19132	12	16:20:26.902	20:27.476	20:27.528	20:27.529	627
typhon12	19132	13	16:20:26.902	20:27.529	20:27.581	20:27.582	680
typhon12	19132	14	16:20:26.902	20:27.582	20:27.634	20:27.635	732
typhon12	19132	15	16:20:26.903	20:27.635	20:27.687	20:27.687	784
typhon12	19132	16	16:20:26.903	20:27.687	20:27.739	20:27.740	836
typhon12	19132	17	16:20:26.904	20:27.739	20:27.791	20:27.792	887
typhon12	19132	18	16:20:26.904	20:27.792	20:27.844	20:27.846	941
typhon12	19132	19	16:20:26.905	20:27.845	20:27.897	20:27.898	993
typhon12	19132	20	16:20:26.905	20:27.898	20:27.950	20:27.952	

1046Number of tasks completed:20

Connection to compute hosts OK.

SIs and SIMs OK.

Test Run 1 done. Adding to summary and detail reports...

First task submit time: 2007-07-25 16:20:26.896
Last task end time: 2007-07-25 16:20:28.029

Efficiency: 91.9%
[(Task run time * # of tasks)/(Elapsed run time * # of SIs used)]

Elapsed run time: 1.133sec
[Last task end time - First task submit time]

Number of ExecHost in use: 1

Number of service instances in use: 1

Item	ResponseTime(ms)	TaskID	ExecHostName
Minimum	53	1	typhon12
Maximum	1046	20	typhon12
Average	549		

Efficiency: 91.9%
[(Task run time * # of tasks)/(Elapsed run time * # of SIs used)]

Elapsed run time: 1.133sec
[Last task end time - First task submit time]

Number of ExecHost in use: 1

Number of service instances in use: 1

Item	ResponseTime(ms)	TaskID	ExecHostName
Minimum	53	1	typhon12

Maximum	1046	20	typhon12
Average	549		

Compute Host	SI-PID	Number of Tasks
typhon12	19132	20

The responses times below are high because of the initial “cold-start” and because we executed all tasks on a single host.

Check the `symping.summary.*` and `symping.date*` files for results.

5.3 SampleApp

The third test performed to validate the Symphony cluster is to compile and run a customized application `SampleApp`. The Developer’s Edition is needed to test this application. Section 4.4 and the Developer’s Edition documentation [10] describe how to install the Developer’s Edition.

The version of the Developer’s Edition packages ending in 82259 (`symphonyDE-linux2.6-glibc2.3-x86-3.1.0-82259.rpm` and `symphonyDE-linux2.6-glibc2.3-x86_64-3.1.0-82259.rpm`) do not work.

We tested the Developer’s Edition version ending in 89969 (`symphonyDE-linux2.6-glibc2.3-x86-3.1.0-89969.rpm` and `symphonyDE-linux2.6-glibc2.3-x86_64-3.1.0-89969.rpm`), and after a little tweaking could successfully run the `SampleApp` program. In order for `SampleApp` to work correctly we needed to modify `SampleApp.xml`. We changed the `<env name="SOAM_HOME">` parameter to reflect the path for Symphony rather than to SymphonyDE in two locations in the file. The default location for Symphony is `/opt/ego/soam`. We then registered the updated version, and the `SampleClient` ran as expected.

The steps necessary are to compile and run the application on a Symphony cluster:

1. Log in to the master node as the `egoadmin` user. Navigate to the `/opt/symphonyDE/3.1/samples/PHP/SampleApp/` directory. Begin the compilation process by entering the following command:

```
make
```

2. Go to the `output` subdirectory and create the `SampleService` service package with the following commands:

```
tar -cvf SampleService.tar SampleService  
gzip SampleService.tar
```

3. Deploy the newly created service package with the following command:

```
soamdeploy add SampleService -p SampleService.tar.gz -c  
/SampleApplications/SOASamples
```

4. Verify that the service has been successfully deployed with the following command:

```
soamdeploy view -c /SampleApplications/SOASamples
```

5. Register the application with the following command:

```
soamreg ../SampleApp.xml
```

6. Check to make sure the newly registered application is visible with the following command:

```
soamview app
```

7. Disable the `symping` application with the following command:

```
soamcontrol app disable symping
```
8. Run the client for the sample application with the following command:

```
./SyncClient
```
9. Assuming the execution was successful, `symping` can be re-enabled with the following command:

```
soamcontrol app enable symping
```

6 Troubleshooting

The following sections discuss how to troubleshoot some potential Symphony/EGO issues.

6.1 Symphony/EGO do not start on compute nodes

We encountered problems when trying to use Symphony and EGO on the compute nodes within HP Clusters. The installation went smoothly and everything looked like it should have worked. When using the `egosh resource list` command only the master node's status was `ok`. All other nodes were listed as status unknown.

We noticed that on the compute node only the `lim` process was started. The `pem` and the `pim` processes were missing. When looking in the `/opt/ego/kernel/log` directory we noticed that the `lim.log.<nodename>` log file was the only log file available and its content was pretty sparse.

This problem can be resolved by making sure that the firewall is not blocking the ports needed by EGO and Symphony as follows:

- On non-XC clusters, the system administrator can determine whether the problem is a communication problem caused by the firewall settings by entering the following command.

```
service iptables stop
```

This is only recommended as a debugging procedure and should not be used as a permanent solution.
- On HP clusters like XC, system administrators should use the `openiptables` tool, which opens an IP port in the firewall. The `openiptables` tool is located in `/opt/hptc/bin`. For example:

```
openiptables --port 7869 --protocol tcp --interface Admin
```

6.2 Web Interface Does Not Load

If the Web interface does not load, check to see that the WEBGUI service is running. This can be done by entering the following command:

```
[egoadmin@typhon16 log]$ egosh service list
SERVICE      STATUS    ALLOC CONSUMER RGROUP  RESOURCE  SLOTS SEQ_NO STATE ACTI
derbydb        STARTED  1      /Managem Managem typhon16  1        1      RUN   156
plc            STARTED  9      /Managem Managem typhon16  1        1      RUN   159
purger        STARTED  10     /Managem Managem typhon16  1        1      RUN   160
WEBGUI        STARTED  11     /Managem Managem typhon16  1        1      RUN   155
RS            STARTED  5      /Managem Managem typhon16  1        1      RUN   157
WebServiceG   DEFINED   /Managem Managem
SD            STARTED  6      /Managem Managem typhon16  1        1      RUN   158
ServiceDire  ALLOCATI 20     /Managem Managem
[egoadmin@typhon16 log]$
```

If the `WEBGUI` entry under the `SERVICE` column does not have a corresponding `STARTED` entry under the `STATUS` column, then the service simply is not running. Restarting EGO by using the command `egosh ego restart` will often fix the problem.

If the `WEBGUI` entry has in fact `STARTED`, then it may be blocked by another service. Check for listening services on its port by entering the following command:

```
netstat -an | grep 8080
```

If this is not the issue, then check to make sure that the machine has not been firewalled. On HP clusters like `XC`, system administrators should use the `openiport` tool, which opens an IP port in the firewall. `openiport` is located in `/opt/hptc/bin`. For example:

```
openiport --port 8080 --protocol tcp --interface External
```

6.3 derbydb never enters `STARTED` state

The `derbydb` database is designed to automatically perform all database creation and management steps. However, in our experience it will only do this on the master host. All attempts to configure `derbydb` to use a compute host as the database server failed. As a result, we recommend configuring the `EGO` installation to use the master host as the `derby` server by setting the `DERBY_DB_HOST` variable to the master host. This has to be done on installation otherwise the software will need to be reinstalled for the `derbydb` database to work and to change the Symphony internal setting.

Although the *Installing a Single-Host Cluster on Linux* [6] and *Planning and Installing Your Cluster on Linux* [7] documentation mentions that `DERBY_DB_HOST=hostM` can be set in the cluster properties configuration file. In our experience, this does not work.

6.4 SampleApp throws a strange error when running the client

If, upon running the `SyncClient` portion of `SampleApp`, the error is received is:

```
"exception caught ... Failed to start session manager for consumer
/SampleApplications/SOASamples. An operation timed out. Please make sure
resources are available for session manager and try again."
```

This indicates that the source code has not been compiled correctly. To compile the application correctly, it is required to make sure that the Developer's Edition of Symphony is installed. This package includes files that are critical to the proper functioning of the application and places them in the correct directories on the master node. Once this has taken place, the procedure described in [Section 5.3](#) and the *Planning and Installing Your Cluster on Windows or Planning and Installing Your Cluster on Linux* documents [7, 9] should be followed.

6.5 Using `ssh` Instead of `rsh`

Most of the `EGO` commands have options that allow broadcasting commands to all the nodes. Examples of such commands are:

```
egosh ego start all
egosh ego stop all
egosh ego restart all
```

At this point Symphony does not explicitly support `ssh`.

The following example shows the output after entering the `ego start all` command:

```
bash-3.00$ egosh ego start all
Do you really want to start up LIM on all hosts ? [y/n]y
Start up LIM on <typhon16> ..... typhon16: No route to host
rsh failed; please ensure correct operation of rsh.
bash-3.00$
```

If `rsh` is disabled on the cluster, the lack of `ssh` support can be overcome by creating a link from `rsh` to `ssh` in `/usr/bin`:

```
mv rsh rsh.old
```

```
ln -s ssh rsh
[egoadmin@typhon16 ~]$ egosh ego start all
Do you really want to start up LIM on all hosts ? [y/n]y
Start up LIM on <typhon16> ..... done
Start up LIM on <typhon12> ..... done
[egoadmin@typhon16 ~]$
```

References

- [1] Platform Symphony, <http://www.platform.com/Products/Platform.Symphony/>
- [2] EGO 1.2.1 documentation, Platform EGO Developer's Guide, /opt/ego/docs/ego/1.2.1
- [3] Symphony 3.1 documentation, docs/symphony/3.1/running_managing/index.html
- [4] Platform Symphony, Enterprise Grid Solutions for Financial Services, white paper, Platform Computing
- [5] Symphony API, Developer's Edition, /opt/symphonyde/docs/symphonyde/3.1/
- [6] Installing a Single-Host Cluster on Linux, /opt/ego/docs/ego/1.2.1/install_single-host_unix/
- [7] Planning and Installing Your Cluster on Linux, /opt/ego/docs/ego/1.2.1/planning_installing_linux/
- [8] Installing a Single-Host Cluster on Windows, /opt/ego/docs/ego/1.2.1/install_single-host_windows/
- [9] Planning and Installing Your Cluster on Windows, /opt/ego/docs/ego/1.2.1/planning_installing_windows/
- [10] Developer's Edition, /opt/symphonyDE/docs/symphonyde/3.1/
- [11] System requirements, <http://www.platform.com/Products/Platform.Symphony/Product.Information/System.Requirements/>

For More Information

For questions regarding Platform Computing EGO and Platform Computing Symphony, send email to support@platform.com or refer to Platform Computing's website: <http://www.platform.com>

For questions regarding installing Platform Computing EGO and Platform Computing Symphony on HP Clusters, send email to pdlhpcdgrid@hp.com or refer to the section about Grid and AI on the website: www.hp.com/go/collaboration

© 2007 Hewlett-Packard Development Company, L.P. The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

4AA1-6096ENW, October 2007

Platform[™]

