Jaryba® SmartSuspend™

Integration with Oracle Grid Engine

Version 2.1.1

August 2012

Jaryba, Inc.

2068 Tesuque CT Reno, Nevada 89511

Disclaimer

The information contained herein is subject to change without notice. Jaryba, Inc. is not liable for errors or damages, incidental or consequential, in connection with the furnishing, performance, or use of this material.

Jaryba SmartSuspend software users are bound by the terms and conditions of the Jaryba, Inc. license agreement.

Copyright

Trademarks

Jaryba, the Jaryba logo and Jaryba SmartSuspend are trademarks or registered trademarks of Jaryba, Inc. or its subsidiaries in the United States and/or other countries.

All other company and product names mentioned may be trademarks of the respective companies with which they are associated.

Contents

Preface

This guide provides:

- Instructions for integrating Jaryba SmartSuspend software with Oracle Grid Engine
- Instructions for submitting jobs to OGE using SmartSuspend

This guide is intended for SmartSuspend administrators and users.

Prerequisites

To use Jaryba SmartSuspend, you should have working knowledge of the operating system and shell for the machines on which you are performing the operations described in this guide.

In addition, you should have in-depth knowledge of any third-party applications you are using in conjunction with the SmartSuspend software.

Conventions Used in This Guide

Commands, URLs, and any text that you enter appear in **`this bold font`**.

Computer output, file names, and locations appear in `this font`.

Elements on the user interface appear in **bold like this**.

Important new terms appear in *italics like this*.

Variables, where you need to substitute site- or installation-specific details, appear in brackets and italics <*`like_this`*>.

Contacting Jaryba

For product operation and support inquiries send email to:
`support@jaryba.com`

The following information is needed when you contact Jaryba, Inc. for support:

- Company name, contact name, email address, and phone number
- Software version number: execute the software library name to obtain the version number, for example: `/lib/libssr.so` or `/lib64/libssr.so`
- Platform details: hardware, operating system type, and version

- Problem background information: describe the problem in as much detail as possible and any actions you have attempted to resolve the problem. In addition, be sure to include available log files and error message text.

You can also visit our support Web site at:
http://support.jaryba.com/

For general inquiries, send email to:
info@jaryba.com

For sales inquiries, send email to:
sales@jaryba.com

For general information about Jaryba and Jaryba SmartSuspend visit our Web site at:
http://jaryba.com/

1 Integrating SmartSuspend with OGE

Overview

Integrating Jaryba SmartSuspend with Oracle Grid Engine allows you to leverage the powerful suspend and resume capabilities of SmartSuspend, while using the familiar Oracle Grid Engine interface to submit and monitor jobs.

This document provides instructions for integrating SmartSuspend with OGE, including configuration details and working with an example OGE integration wrapper script supplied by Jaryba. This documentation assumes that you have prerequisite and working knowledge about configuring, administering, and using both the Jaryba SmartSuspend and the Oracle Grid Engine software.

Configuring the SmartSuspend Integration Script

The OGE integration script is included in the SmartSuspend distribution. After installing SmartSuspend according to the instructions in the *Using SmartSuspend* documentation, you will find the integration script, `ssroge.sh`, in `/usr/share/smartsuspend`.

The `ssroge.sh` integration script defines default values for the variables listed below:

```
SSR_LOGDIR_ROOT_DEFAULT=<specify directory>
SSR_SAVE_PATH_DEFAULT=/tmp
SSR_SIG_RESUME_DEFAULT=58
SSR_SIG_SUSPEND_DEFAULT=60
```

- `SSR_LOGDIR_ROOT_DEFAULT` specifies the default path for `SSR_LOGDIR_ROOT`, which is the base directory where `ssroge.sh` creates the subdirectory `<SSR_LOGDIR_ROOT>/<JOB_ID>`. This subdirectory contains the SmartSuspend status directory (required for storing job state communication data and process IDs (PIDs)), in addition to various log files. Ideally, `SSR_LOGDIR_ROOT_DEFAULT` is a location on a shared file system, because the directories for jobs started on all machines are then kept in one location. A location on a local file system may also be used if desired.

Note: There is no valid default set for this variable; you must specify a valid path to which all users have read, write and execute permissions.

For instance, if:

```
SSR_LOGDIR_ROOT_DEFAULT=/path/to/ssroge/logdir_root
```

use the following commands as root:

```
$ mkdir -p /path/to/ssroge/logdir_root
$ chmod 1777 /path/to/ssroge/logdir_root
```

- SSR_SAVE_PATH_DEFAULT sets the default location in which SmartSuspend stores the memory for suspended jobs. The default location is set to /tmp (which is also the default location for ssrcmd), but you can specify a different location by setting this variable. Override this default by passing the "-d <pathname>" argument to ssroge.sh, or by setting the SSR_SAVE_PATH environment variable.

- SSR_SIG_RESUME_DEFAULT and SSR_SIG_SUSPEND_DEFAULT variables define the signals that ssroge.sh sets to instruct SmartSuspend to use to suspend and resume jobs. Note that the values used for the defaults in ssroge.sh may differ from the default values that ssrcmd assumes (10 for suspend, and 12 for resume). Although you would not normally need to, you can change the signals that SmartSuspend uses, if one of your applications uses these signal values internally for another purpose.

Note: Be aware that $SSR_SIG_RESUME_DEFAULT and $SSR_SIG_SUSPEND_DEFAULT will not override any values already specified for $SSR_SIG_RESUME or $SSR_SIG_SUSPEND in your shell/application startup script or set from the command line through the -a (--sig_suspend) or -b (--sig_resume) options. If you encounter problems suspending a job, make sure you do not have variable specification conflicts.

Configuring OGE to Use SmartSuspend

The standard OGE product provides a preemptive scheduling feature that allows pending high-priority jobs to take resources away from running jobs of a lower priority. To keep the integration as simple and transparent as possible, we take advantage of OGE's subordinate queue configuration, and use SmartSuspend technology for handling the underlying job suspension and resumption operations.

You can specify an OGE queue as:

- Preemptive—Jobs in a *preemptive* queue can suspend jobs in any subordinated queue.

- Preemptable—Jobs in a *preemptable* queue can be suspended by any job in a preemptive queue on the same host.

Add SmartSuspend Preemptive Scheduling Queues

Add a preemptive queue called "highp.q" and a preemptable queue called "lowp.q" using the following queue configurations.

highp.q configuration:

```
qname                 highp.q
starter_method        /usr/share/smartsuspend/ssroge.sh
suspend_method        /usr/share/smartsuspend/ssroge.sh -s
resume_method         /usr/share/smartsuspend/ssroge.sh -r
subordinate_list      lowp.q
```

lowp.q configuration:

```
qname                 lowp.q
starter_method        /usr/share/smartsuspend/ssroge.sh
suspend_method        /usr/share/smartsuspend/ssroge.sh -s
resume_method         /usr/share/smartsuspend/ssroge.sh -r
```

Configuring the highp.q queue to use the smartsuspend methods for starting, suspending, and resuming is entirely optional. Jobs in this queue will not be automatically suspended by OGE but can manually suspended and resumed by the administrator or user to these jobs with the `qmod` command.

Submitting Your Job to OGE Using SmartSuspend

`ssroge.sh` takes the same arguments, in the same format, that `ssrcmd` does. However, any `-n` argument passed to `ssroge.sh` will be ignored. Similarly, any value set for SSR_STATUS_PATH will be overwritten by `ssroge.sh`. Refer to the *Using SmartSuspend* documentation for more information about `ssrcmd`.

Submit your job to either the preemptive queue (highp.q) or the preemptable queue (lowp.q) that you created for SmartSuspend jobs, for example:

```
qsub –q highp.q script.qsub
```

If there are more OGE job slots needed than are available, OGE automatically uses SmartSuspend to suspend enough jobs submitted to the lowp.q queue to make room for jobs in the highp.q queue. When enough job slots become available, OGE uses SmartSuspend to resume any suspended jobs in the lowp.q queue. OGE's overall scheduling policies and resource policies determine which jobs are suspended or resumed.

FLEXnet Licenses

In your `qsub` script or in your queue configuration, prepend either the `-f <`*`license_file`*`>` or `-l <`*`license_host`*`>` option, but not both. Note that if `ssroge.sh` is already in your execution path, you don't need to specify the full path.

- `<license_file>` specifies the FLEXnet $LM_LICENSE_FILE environment variable as the license file.
- `<license_host>` specifies the FLEXnet license server host, or multiple hosts in a colon-separated list

For example:

```
/usr/share/smartsuspend/ssroge.sh -f <license_file> -- <application_name>
<application_arguments>
```

Parallel Applications Using MPI

To submit parallel applications using MPI, add the 'mpirun' command and the appropriate MPI option to your `bsub` script:

SmartSuspend MPI options are:

Option	Description
`-H, --hpmpi`	Enables HP-MPI support.
`-M, --mpich`	Enables MPICH MPI support.
`-O, --openmpi`	Enables OpenMPI support.

In this example, an MPICH application <`mpi_app`> is submitted with `mpirun`:

```
/usr/share/smartsuspend/ssroge.sh --mpich -- mpirun <mpi_app> <mpi_app_args>
```

You can choose to add this option to your queue configuration or remove the startup_method altogather, and call the ssroge.sh integration script from each qsub script individually.

Controlling Your OGE Job Using SmartSuspend

Your job can be suspended, resumed or killed with the standard OGE commands, but instead of using the default OGE operations, the commands use the underlying SmartSuspend operations:

- `qmod -s <job_id>`— OGE uses SmartSuspend to suspend CPU usage, page out memory on demand, and free up licenses if required.
- `qmod -us <job_id>`— OGE uses SmartSuspend to restore CPU and memory usage and allow the application to obtain any needed licenses as required.

Use the OGE command `qdel` to remove/kill the job; no action by SmartSuspend is necessary:

`qdel` <*job_id*>—OGE sends its default signals to kill your job.