SIEMENS EDA

Calibre® Dynamic Resource Allocator (DRA) User’s Manual

Software Version 2021.2

SIEMENS

Unpublished work. © 2021 Siemens

This material contains trade secrets or otherwise confidential information owned by Siemens Industry Software, Inc., its subsidiaries or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this information is strictly limited as set forth in Customer's applicable agreement with Siemens. This material may not be copied, distributed, or otherwise disclosed outside of Customer's facilities without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This document is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made. Siemens disclaims all warranties with respect to this document including, without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement of intellectual property.

The terms and conditions governing the sale and licensing of Siemens products are set forth in written agreements between Siemens and its customers. Siemens' **End User License Agreement** may be viewed at: www.plm.automation.siemens.com/global/en/legal/online-terms/index.html.

No representation or other affirmation of fact contained in this publication shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

TRADEMARKS: The trademarks, logos, and service marks ("Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' trademarks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Support Center: support.sw.siemens.com
Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction to Calibre Dynamic Resource Allocator (DRA)

This section introduces Calibre® Dynamic Resource Allocator (DRA). It begins with an overview of the product, followed by the flow, some key concepts, the requirements, the syntax conventions, and how to invoke this interactive software tool.

Calibre DRA Overview

Calibre DRA is a tool for managing job resources that enables you to adjust the number of CPU cores assigned to a Calibre job that is already running. The Calibre DRA command line interface also gives you access to the job status, which includes information about active and allocated remote resources, current operations, and resource demand.

Calibre DRA allows you to make dynamic resource changes in a currently running Calibre job as well as obtain information about the resources on which the job is running.

The Calibre DRA product is designed for system administrators and end-users who want the ability to optimize resource utilization for a job, but also want to minimize the setup overhead and complexity that can be associated with a full cluster management tool such as Calibre® Cluster Manager (CalCM). However, unlike CalCM, the Calibre DRA user needs to manually initiate the communications to adjust the resources and collect information for the job.

Calibre DRA runs in the Calibre® MTflex™ process environment and is invoked simultaneously with the launch of the Calibre job. Calibre DRA can then be used throughout the run to access, control, and query information about the job.

Calibre jobs and operations supported for running on CalCM are also supported for Calibre DRA. This includes the Calibre® FullScale™ post-tapeout platform. See the *Calibre Post-Tapeout Flow User's Manual* for more information on using Calibre FullScale. For support

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

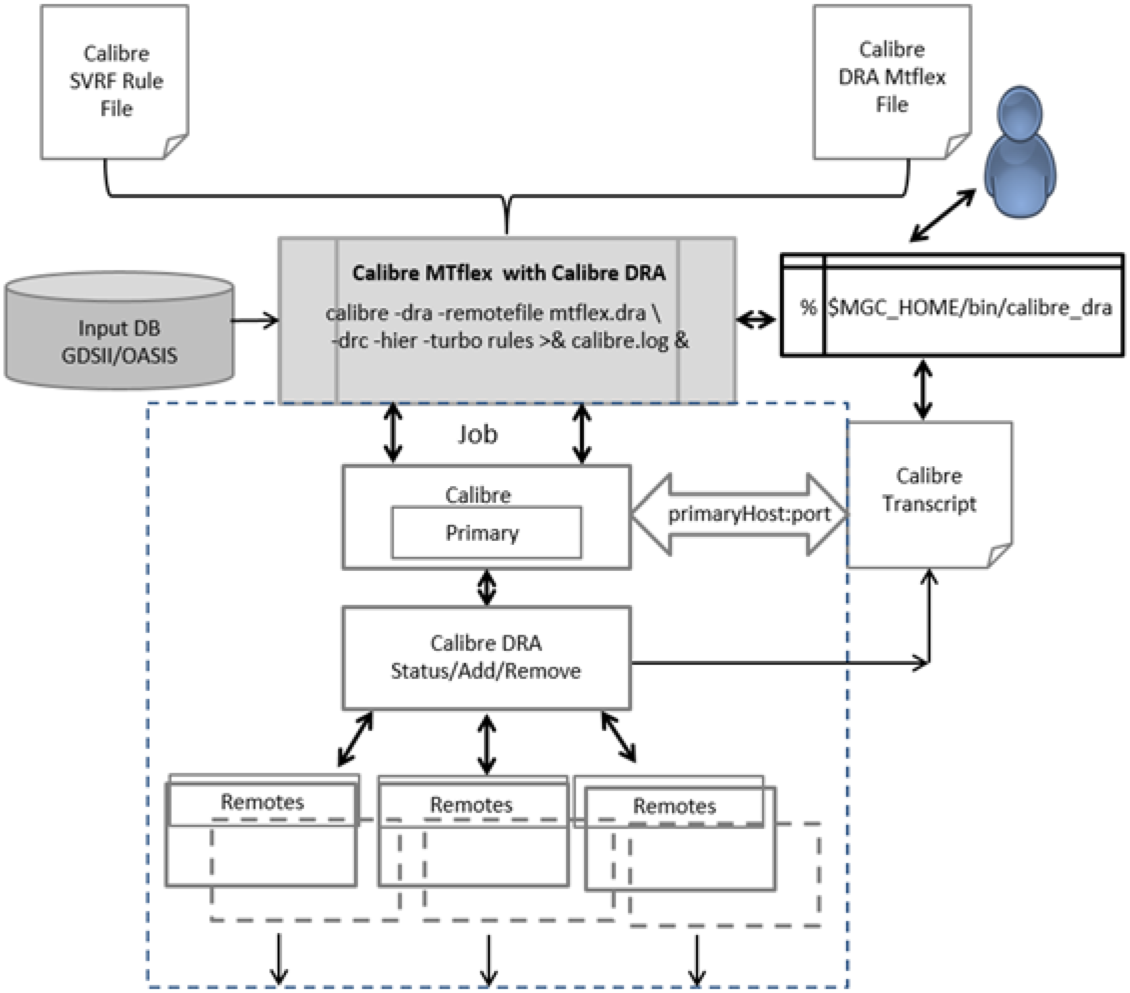
questions about running specific Calibre tools and operations with Calibre DRA, consult your Siemens representative.

Calibre DRA Flow

The Calibre DRA flow encompasses the job's run. While the job is running, you can use Calibre DRA to change the job's resources by initiating a DRA event. The Calibre DRA flow ends upon completion of the Calibre job.

The following figure illustrates how Calibre DRA is used to adjust job resource allocations. Calibre DRA includes a set of command line statements that can be issued during the Calibre job to update CPU resource allocations and get real-time status for remotes and job operations.

Figure 1-1. Interaction of Calibre DRA with Calibre Job


Calibre SVRF Rule File
Calibre DRA Mtflex File
Calibre MTflex with Calibre DRA
calibre -dra -remotefile mtflex.dra \
-drc -hier -turbo rules >& calibre.log &
Input DB GDSII/OASIS
% $MGC_HOME/bin/calibre_dra
Job
Calibre
Primary
primaryHost:port
Calibre Transcript
Calibre DRA Status/Add/Remove
Remotes
Remotes
Remotes

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

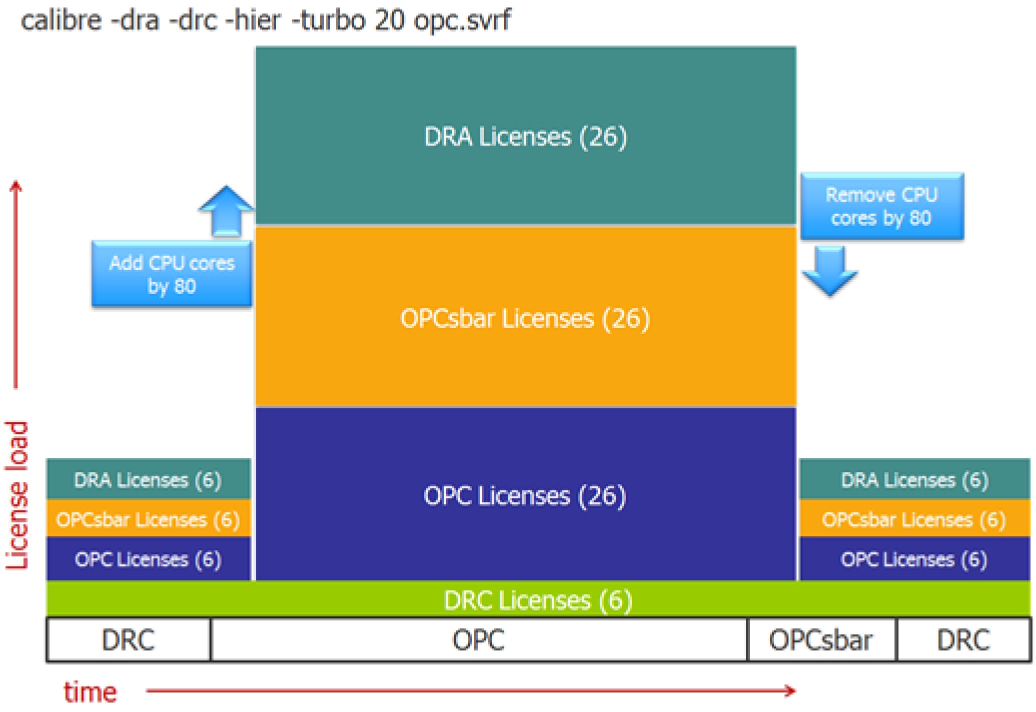
Calibre DRA Licensing Behavior

The licensing behavior of Calibre DRA coincides with the dynamic changes in the CPU core allocation during a job.

When a Calibre run is invoked with Calibre DRA, job resources are added and removed during the run triggered by user-initiated DRA events. Calibre DRA licenses are secured at the beginning of the run and released at the end. As job resources are added and removed during the run, the number of licenses acquired and released is adjusted accordingly. The standard Calibre license consumption and substitution schemes apply to Calibre jobs running with Calibre DRA; see the *Calibre Administrator's Guide* for more details and complete licensing information.

In the following figures, the dynamic licensing behavior of Calibre DRA is compared to the static licensing behavior of a standard Calibre run (without Calibre DRA), where licenses are held throughout the run.

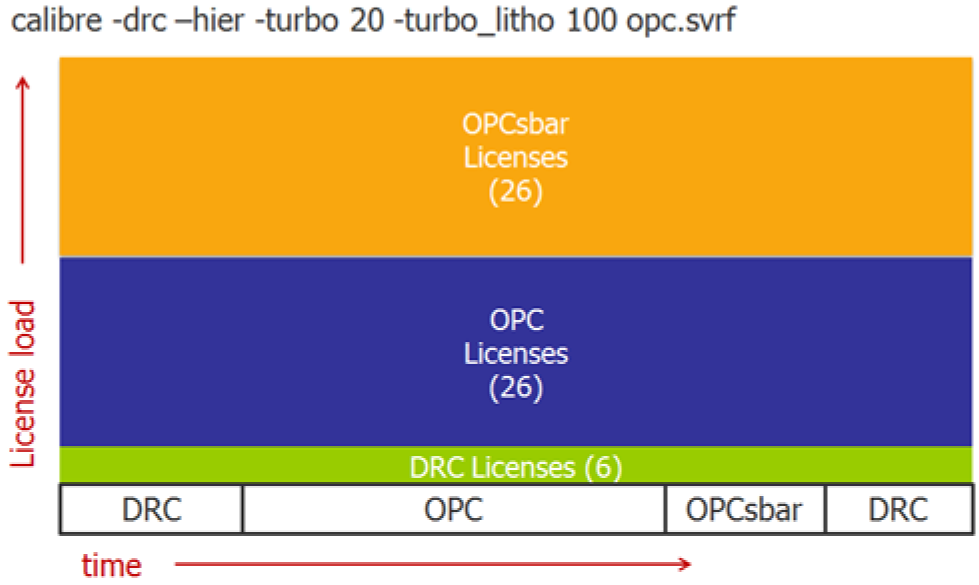
Figure 1-2. Dynamic Licensing Under Calibre DRA



```
calibre -dra -drc -hier -turbo 20 opc.svrf
```

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Figure 1-3. Static Licensing Under Calibre



For Calibre jobs running with Calibre DRA, you can check the real-time license usage from your Mentor Graphics Licensing System (MGLS) license server at the command line:

```
$MGC_HOME/bin/lmstat -a
```

Note

Adding and removing a large number of CPUs quickly can be slower for the MGLS license server compared to a static job. If this is the required use model, Calibre Cluster Manager (CalCM) is recommended instead of Calibre DRA, because the License Broker Daemon used by CalCM can be faster than using the MGLS license server.

Starting the job on 20 remote CPUs (as shown in the previous figures) results in a total of six licenses to start the job. See “Licensing Requirements for Distributed Calibre” in the *Calibre Administrator’s Guide* for computation details. The Calibre DRC licenses remain fixed at six during the job due to the -turbo 20 option, which instructs Calibre to limit DRC operations to 20 CPUs even if LITHO operations can scale higher.

Note

The -turbo_litho option is not supported for Calibre DRA.

Calibre DRA Key Concepts

The definitions are provided for some key terms and concepts used when discussing Calibre DRA.

- **Dynamic Resource Allocation** — The ability to add or remove a run’s assigned CPUs during execution per variations in the workload. It is also the name of a Calibre utility.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

This approach addresses the issue of over or under allocating resources, as in the case of static systems, and can be used to efficiently distribute the workload across multiple compute resources.

- **Primary** — The primary is the host machine from which you invoke a Calibre job. The primary host launches and connects to remote hosts and handles tasks not assigned to the remote hosts. The primary host is sometimes referred to in commands, environment variables, and transcripts as a “local” or “master” host.
- **Remote** — The remote hosts run a Calibre task that is assigned by the Calibre job running on the primary host. The remote hosts must be able to access the MGC_HOME tree for the respective platform. For Linux®[1] environments using multiple Linux hardware and operating systems, the remote hosts may need to access different MGC_HOME trees for their respective platforms.

  Note

  When using advanced Calibre platform options such as Calibre Remote Data Server (-remotedata) and remote hyperscaling (-hyper remote), or Calibre FullScale (-pto), the remote hosts must also communicate with each other and have access to the same file system as the primary host.

- **Distributed Processing** — This type of processing, provided by Calibre MTflex functionality, is a parallel processing data architecture for running multithreaded Calibre tools on distributed networked computers. For complete information, see “Calibre MTflex Processing” in the *Calibre Administrator’s Guide*.

Calibre DRA Requirements

There are certain requirements for running Calibre DRA in your job environment.

Hardware and Networking

All remote hosts must be able to communicate with all primary hosts in your hardware system and operating environment. All other requirements for running Calibre must also be supported. Refer to your Siemens representative for information specific to your environment.

Licensing

A specific license is required for running Calibre DRA in addition to the necessary product licenses for your Calibre jobs. The standard Calibre license consumption formula applies to Calibre DRA and all product licenses. Licenses are acquired and released according to the number of CPUs added or removed during the Calibre run. See also “Calibre DRA Licensing Behavior” on page 11. For complete information on all Calibre product licensing, refer to the *Calibre Administrator’s Guide*.

[1] Linux® is a registered trademark of Linus Torvalds in the U.S. and other countries.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Calibre Version Support

Calibre DRA is supported as of Calibre version 2018.4.

Operating Systems

Calibre tools run on various operating system combinations and hardware configurations. For complete information, see "Supported Operating Systems and Hardware" in the *Calibre Administrator's Guide*.

Environment Variables

If you do not use any Siemens EDA products other than Calibre, you can set the MGC_HOME environment variable to the location of your Calibre software tree. If you do use other Siemens EDA products in addition to Calibre, then set the CALIBRE_HOME environment variable to the location of your Calibre software tree and use the MGC_HOME environment variable for other Siemens EDA products. See "CALIBRE_HOME Environment Variable" in the *Calibre Administrator's Guide* for more information.

Configuration Files

A Calibre MTflex configuration file is required with your remote host names and CPU numbers for the Calibre run with Calibre DRA. When invoked, Calibre DRA uses certain required Calibre MTflex file configuration statements for the run. See "Configuring the Calibre MTflex File for Calibre DRA" on page 17 for detailed information.

Job Input Files

The input data and files for your Calibre run includes a GDS or OASIS®[2] database, model files, and a Standard Verification Rule Format (SVRF) file.

Related Topics

Calibre Administrator's Guide [Calibre Administrator's Guide]

Syntax Conventions

The command descriptions use font properties and several metacharacters to document the command syntax.

Table 1-1. Syntax Conventions

Convention	Description
Bold	Bold fonts indicate a required item.

[2] OASIS® is a registered trademark of Thomas Grebinski and licensed for use to SEMI®, San Jose. SEMI® is a registered trademark of Semiconductor Equipment and Materials International.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Table 1-1. Syntax Conventions (cont.)

Convention	Description
Italic	Italic fonts indicate a user-supplied argument.
`Monospace`	Monospace fonts indicate a shell command, line of code, or URL. A bold monospace font identifies text you enter.
Underline	Underlining indicates either the default argument or the default value of an argument.
UPPercase	For certain case-insensitive commands, uppercase indicates the minimum keyword characters. In most cases, you may omit the lowercase letters and abbreviate the keyword.
[ ]	Brackets enclose optional arguments. Do not include the brackets when entering the command unless they are quoted.
{ }	Braces enclose arguments to show grouping. Do not include the braces when entering the command unless they are quoted.
‘ ’	Quotes enclose metacharacters that are to be entered literally. Do not include single quotes when entering braces or brackets in a command.
| or ||	Vertical bars indicate a choice between items. Do not include the bars when entering the command.
…	Three dots (an ellipsis) follows an argument or group of arguments that may appear more than once. Do not include the ellipsis when entering the command.
Example: **DEVice** {*element_name* [‘(’*model_name*‘)’]} ***device_layer*** {***pin_layer*** [‘(’*pin_name*‘)’] …} [‘<’*auxiliary_layer*‘>’ …] [‘(’*swap_list*‘)’ …] [BY NET | BY SHAPE]	

Calibre DRA Mode of Operation

Calibre DRA uses a Calibre command line.

Calibre DRA is specified with the -dra switch at the Calibre command line along with typical options for running a Calibre run with Calibre MTflex processing. For example,

```
$MGC_HOME/bin/calibre -dra -remotefile mtflex.dra -drc -hier -turbo \
    rules.svrf >& calibre.log &
```

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

During the Calibre job, Calibre DRA command line options are used interactively through the calibre_dra command to get status information and to dynamically adjust the number of resources for the job.

Calibre DRA can be used stand-alone with the remote hosts manually specified, or alternatively, it can be used with a platform like Platform LSF™ (LSF) or Univa® Grid Engine® with the remote hosts coordinated through a job scheduler. See "Configuring the Calibre MTflex File for Calibre DRA" on page 17 and "Calibre DRA Stand-Alone Configuration Example" on page 24 for more information.

The Calibre DRA Direct Connect mode can be used to connect extra remote resources to a running Calibre job using the calibre_dra_remote command. This mode provides an interface to an existing external system or job scheduler to connect the extra remote resources. The job status information from "calibre_dra -status" helps the user determine the resource adjustment. See "Calibre DRA Direct Connect Mode" on page 26 and "calibre_dra_remote" on page 36.

Related Topics

calibre -dra

calibre_dra

calibre_dra_remote

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Chapter 2
Running Calibre DRA

This section covers how to create a customized Calibre MTflex file, invoke Calibre DRA, and then use it to access the resources and status of your running Calibre job.

Configuring the Calibre MTflex File for Calibre DRA

This procedure includes the additional statements and arguments used to customize your Calibre MTflex configuration file for executing Calibre jobs with Calibre DRA.

Refer to the following sections of the *Calibre Administrators’s Guide* for the exact syntax and a complete description of the Calibre MTflex configuration file:

- Distributed Calibre: Creating a Configuration File — Information on creating the Calibre MTflex configuration file.
- Distributed Calibre: Executing and Monitoring— Information and scenarios for running Calibre MTflex in different environments and launch modes.
- -remotefile — Information on the argument specifying the path to the Calibre MTflex configuration file.

Prerequisites

- The MGC_HOME environment variable must be set to the location of a Calibre software tree. This software tree must be accessible by the remote hosts and the same version of Calibre must be used throughout the network. See “Environment Variables” on page 14 for more information.
- LM_LICENSE_FILE or MGLS_LICENSE_FILE must be set to the location of a license file. See “Mentor Standard Licensing (MSL)” in the *Calibre Administrator’s Guide* for more information.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Procedure

1. In an ASCII editor, add statements to your Calibre MTflex configuration file to enable system level monitoring of the local and remote hosts, for example,

   ```
   //Monitoring
   MONITOR UTILITY
   MONITOR SYSTEM INTERVAL 300 HEARTBEAT 4 PRINT
   MONITOR LOCAL LOAD INTERVAL 120
   ```

   These statements are used to monitor the system for unresponsive remote hosts at a calculated time period (INTERVAL value x HEARTBEAT value) and print to standard output (stdout) when no response is received. If the time period is exceeded, the connection between the local and the remote host is dropped.

   These settings can be overridden with certain Calibre DRA command line arguments. Refer to "calibre_dra" on page 34.

   Note

   The MONITOR UTILITY statement is mandatory for correct Calibre DRA usage. It instructs Calibre to compute a CPU demand for the job, which is reported when using the calibre_dra -status command — see "Allocating Resources with Calibre DRA", Step 2.

2. Add a statement to your Calibre MTflex configuration file to set the parameters for the connection of the local host to the remote hosts, for example,

   ```
   LAUNCH AUTOMATIC COUNT 4 MAXCOUNT 1000
   ```

   This statement includes the COUNT and the MAXCOUNT parameters. The MAXCOUNT <*m*> parameter is mandatory and is used to specify the maximum number of remote processes that can be added during the job. The COUNT <*n*> parameter specifies the *initial* number of remote processes needed to launch the job. The COUNT parameter is required when LAUNCH CLUSTER is used and not required when individual REMOTE HOST specifications are used with the LAUNCH AUTOMATIC mode.

   See LAUNCH CLUSTER, REMOTE HOST, and LAUNCH AUTOMATIC in the *Calibre Administrator's Guide* for additional parameter descriptions.

3. Add a statement to your Calibre MTflex configuration file to provide the information for launching the remote host, for example,

   ```
   REMOTE HOST hostxyz 5
   ```

4. Add the following statement to specify the method to launch the Calibre MTflex server on a remote host. This statement also specifies the path to the file on the local host containing the commands for invoking a remote server on the remote host, for example,

   ```
   REMOTE COMMAND DYNAMIC remote_launch_script ARGUMENTS [%N %H:%P]
   ```

The REMOTE COMMAND DYNAMIC statement is specific to Calibre DRA usage and is required in order to add or remove CPU resources. This statement has similar parameters as specified for the REMOTE COMMAND statement in the *Calibre Administrator’s Guide*.

See “Using the LAUNCH CLUSTER and REMOTE COMMAND Statements” for examples of an rsh *remote_launch_script* and the ARGUMENTS specification.

Note

When you run Calibre DRA stand-alone, you specify the remote hosts for dynamic allocation in a *remote_launch_script*. Depending on your environment, it can be preferable to use the LSF *bsub* or Grid Engine *qsub* commands to handle the dynamic aspect of Calibre DRA.

See “Calibre DRA Stand-Alone Configuration Example” on page 24 for an example of using Calibre DRA stand-alone.

Results

You have now added statements to configure the Calibre MTflex file for executing your Calibre job with Calibre DRA.

Examples

Example 1

Here is an example of specifying the Calibre MTflex configuration file that sets up the Calibre MTflex run for a Calibre job with Calibre DRA.

```
calibre -dra -remotefile mtflex.dra -drc -hier -turbo rules.svrf >& \
    calibre.log &
```

Example 2

Here is an example of a Calibre MTflex configuration file that connects initially to named remote hosts and then uses the LSF *bsub* command to choose which additional remote hosts are made available for dynamic CPU allocation for Calibre DRA during the Calibre job. The rcalibre utility is used to initiate the Calibre MTflex run on the remote hosts as described in the *Calibre Administrator’s Guide*.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
// -----------------------------------------------
// Monitoring
// -----------------------------------------------
MONITOR UTILITY
MONITOR SYSTEM INTERVAL 300 HEARTBEAT 4 PRINT
MONITOR LOCAL LOAD INTERVAL 120
// -----------------------------------------------
// Launch Mode
// -----------------------------------------------
LAUNCH AUTOMATIC MAXCOUNT 1000
REMOTE HOST node1 4
// -----------------------------------------------
// DRA Setup
// -----------------------------------------------
REMOTE COMMAND DYNAMIC bsub ARGUMENTS [-J calibre_dra[1-%C] \
    -o calibre_dra_%J[%I].log <MGC_HOME>/bin/rcalibre /tmp 0 \
    -mtflex %H:%P] -f
```

Example 3

Here is an example of a Calibre MTflex configuration file that uses the LSF *bsub* command to choose initial remote hosts and additional remote hosts for dynamic CPU allocation for Calibre DRA during the Calibre job. The REMOTE COMMAND INITIAL statement is specific to Calibre DRA usage and is used to launch the initial remote hosts from the start of the dynamic jobs. This statement has similar parameters as specified for the REMOTE COMMAND statement in the *Calibre Administrator's Guide*. The rcalibre utility is used to initiate the Calibre MTflex run on the remote hosts and is also described in the *Calibre Administrator's Guide*.

```
// -----------------------------------------------
// Monitoring
// -----------------------------------------------
MONITOR UTILITY
MONITOR SYSTEM INTERVAL 300 HEARTBEAT 4 PRINT
MONITOR LOCAL LOAD INTERVAL 120
// -----------------------------------------------
// Launch Mode
// -----------------------------------------------
LAUNCH CLUSTER COUNT 4 MAXCOUNT 1000
REMOTE COMMAND INITIAL bsub ARGUMENTS [-J calibre_dra[1-%C] \
-o calibre_dra_%J[%I].log <MGC_HOME>/bin/rcalibre /tmp 0 \
-mtflex %H:%P] -f
// -----------------------------------------------
// DRA Setup
// -----------------------------------------------
REMOTE COMMAND DYNAMIC bsub ARGUMENTS [-J calibre_dra[1-%C] \
-o calibre_dra_%J[%I].log <MGC_HOME>/bin/rcalibre /tmp 0 \
-mtflex %H:%P] -f
```

The number of initial CPUs launched is 4, specified by the "COUNT 4" option in the LAUNCH CLUSTER statement. The value of 4 is passed through the %C variable to the REMOTE

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

COMMAND INITIAL statement. During the job, the user can decide whether to launch additional CPUs from the command line. For example:

```
$MGC_HOME/bin/calibre_dra -add 25
```

The value of 25 is also passed through the %C variable to the REMOTE COMMAND DYNAMIC statement.

Other variable substitutions used in the example, including %H (hostname) and %P (port number), are available and specified in the *Calibre Administrator's Guide*.

Invoking Calibre DRA

You invoke Calibre DRA from the command line when you execute your Calibre job.

Prerequisites

- The Calibre MTflex file is set up for your run. Refer to "Configuring the Calibre MTflex File for Calibre DRA" on page 17.
- The proper licenses are available for Calibre DRA and the operations to be executed during your run. Refer to the *Calibre Administrator's Guide* for complete product licensing information.

Procedure

1. In a terminal window, specify the -dra switch and the -remotefile argument with the path to your configured Calibre MTflex file along with other typical arguments for executing your Calibre job, for example,

   ```
   $MGC_HOME/bin/calibre -dra -remotefile mtflex.dra -drc -hier \
       -turbo rules.svrf >& calibre.log &
   ```

   Note

   The -turbo_litho option is not supported with the -dra switch.

2. Examine the generated Calibre log file and verify that the Calibre MTflex file is read in and the remotes are launched and connected with the expected number of CPUs.

Results

You have launched your Calibre job with Calibre MTflex processing and specified the Calibre DRA switch, which allows you to access your job resources and information during the run.

Related Topics

calibre -dra

Allocating Resources with Calibre DRA

Calibre DRA enables you to manually interact with your active Calibre job to get status information and adjust resources using the calibre_dra command.

```
$MGC_HOME/bin/calibre_dra
```

Prerequisites

- The Calibre MTflex file is set up for your run. Refer to “Configuring the Calibre MTflex File for Calibre DRA” on page 17.
- You have started a Calibre job with Calibre MTflex processing and Calibre DRA. Refer to “Invoking Calibre DRA” on page 21.

Procedure

1. Run calibre_dra on a command line with no arguments to show the options available for Calibre DRA.

   ```
   $MGC_HOME/bin/calibre_dra
   ```

   The usage specification, commands, and flags are printed to standard output (stdout).

   ```
   Usage: calibre_dra LOG [COMMAND] [FLAG]

   The log should be specified.
     -log <fname>        specifies Calibre transcript to connect.

   The possible commands are listed as follows.
     -status                 prints current Calibre remotes status.
     -add [<host>#]<n>...    adds <n> remotes from the specified
                             <host>.
     -remove [<host>#]<n>... revokes <n> remotes from the specified
                             <host>.

   The command line flag.
     -nowait             don't wait for cpu adjustable or completion
                         of add/remove.
     -timeout <sec>      timeout for the completion of command. The
                         default is 240 seconds.
    -waittime <sec>  wait time for the cpu adjustable. The default
   is 43200 seconds.
   ```

2. Run calibre_dra with the -status argument to show the job status information, for example,

```
$MGC_HOME/bin/calibre_dra -log calibre.log -status
Scanning hostname and port from calibre.log...
Connecting to primary1:<port number>...
REPORT REMOTES
  NAME: node1  CPU: 4/4  IP: <ip address>
  PENDING: 0/0
  TOTAL: 4/4
END
REPORT BRIEF
  TIMESTAMP: 405
  OPS COMPLETE: 19(95%) of 20
  MODULE NAME: EXECUTIVE MODULE  MTFlex HDB(0)
    OPNAME: metal1_opc  OP-TIMESTAMP: 357  OP-ECP:  0.0% OP-CLASS:
LITHO:70
    ACTIVE REMOTE: 4  CELL QUEUE LENGTH: 78  STRIPE QUEUE LENGTH: 59
TOTAL ACTIVE REMOTE: 4/4
END
REPORT UTILITY
  CHANGE: 1
END
```

The status for the Calibre job shows the remote hosts that are connected, the SVRF operations (OPS) running, the number of active remote CPUs, and the CPU demand through the last report parameter. A positive CPU demand means that the Calibre job can accept additional CPUs for the current operation. In this example, CHANGE: 1 means that there is work for one additional CPU. A negative CPU demand means that the current operation is not scaling enough to use all assigned remote CPUs, or the operation is almost finished.

3. Run calibre_dra with the -add argument to add five additional remote CPUs, for example,

```
$MGC_HOME/bin/calibre_dra -log calibre.log -add 5
Scanning hostname and port from calibre.log...
Connecting to primary1:<port number>...
Waiting CPU adjustable... true
Checking current remotes... 4 remotes.
Sending command, "TRIGGER 5"...
Command accepted.
Checking current remotes... 4 remotes.
Checking current remotes... 9 remotes.
Added 5/5 (@ 65s)
```

The Calibre job log file shows the information for the connected CPUs on the remote hosts, the acquired licenses, and the <DCA> string that describes the dynamic CPU allocation event, for example,

```
//  Connected to CPU on remote host node2
//  Connected to CPU on remote host node2
//  calibrehdrc license acquired.
//  calibredrc license acquired.
//  calresourceman license acquired.
//  calmtopcpro license acquired.
<DCA 492 0 1310 N node2 R 2 0 1 1 1 T 5 5 4 A 4 5 30893(P)>
//  Connected to CPU on remote host node3
//  Connected to CPU on remote host node3
//  Connected to CPU on remote host node3
<DCA 493 0 1310 N node3 R 2 0 1 2 1 T 6 6 6 A 6 6 30892(P)>
//  calibrehdrc license acquired.
//  calibredrc license acquired.
//  calresourceman license acquired.
//  calmtopcpro license acquired.
```

When the five additional CPUs are requested, the command specified in the Calibre MTflex configuration file to add dynamic CPUs is executed. The remote Calibre processes launch and connect to the primary, which adds them to its remote pool. The primary distributes computational tasks on the additional resources and reduces its runtime through the usage of more remote CPUs.

The exact format of the <DCA> string is described in "CALCMD_REMOTE_COMMAND_DYNAMIC_LOG_LEVEL" in the *Calibre Cluster Manager (CalCM) User's Manual*.

Results

You have used the calibre_dra command to get the status information and adjust the resources for your active Calibre job with Calibre DRA.

Calibre DRA Stand-Alone Configuration Example

This example shows the configuration for using Calibre DRA stand-alone. In this case, the user is responsible to provide the exact host names as specified in REMOTE COMMAND DYNAMIC or the status output, including full qualification, the CPU counts, and the commands for interacting with the running Calibre job.

Note

These are the minimum requirements to configure Calibre DRA for stand-alone use. Refer to the *Calibre Administrator's Guide* for more information on Calibre MTflex statements and remote launch scripts.

Calibre DRA MTflex File

In the following example configuration file, the MONITOR UTILITY statement is mandatory for the Calibre CPU demand computation. The MAXCOUNT and the REMOTE COMMAND DYNAMIC statements are required for Calibre DRA.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
MONITOR UTILITY
LAUNCH AUTOMATIC MAXCOUNT 1000
REMOTE HOST node123 5
REMOTE COMMAND DYNAMIC <full_path>/dra_launch_remotes.tcl \
    ARGUMENTS [%N %H:%P]
```

Calibre DRA Launch Remotes Script

The following example script is used by the REMOTE COMMAND DYNAMIC statement to allocate the hosts for Calibre DRA. This Tcl script uses the rcalibre utility, which is described in more detail in "rcalibre" in the *Calibre Administrator's Guide*.

```
#!/usr/bin/tclsh
# This script will be executed by Calibre MTFlex REMOTE COMMAND
# DYNAMIC to add new remote CPUs. Syntax:
# REMOTE COMMAND DYNAMIC dra_launch_remotes.tcl ARGUMENTS [%N %H:%P]
# %N, %H, %P are MTFlex variables that Calibre will replace with:
# %N    is the list of remote hosts and number of processes
# %H:%P is the primary host and port for primary/remotes communication
array set hostnameCount {}
foreach part [split [lindex $argv 0] ":"] {
  set host_parts [split $part "#"]
  set name [lindex $host_parts 0]  ;# Example: node123
  set count [lindex $host_parts 1] ;# Example: 5
  if {$count == ""} { set count 1 }
  set hostnameCount($name) $count
}
set primary_communication [lindex $argv 1] ;# Example: primary123:40665
set mgc_home "<full_path_to_MGC_HOME_used_by_primary>"
foreach name [array names hostnameCount] {
  set num $hostnameCount($name)
  for {set x 0} {$x<$num} {incr x} {
    # Launch a remote Calibre process
    # Syntax is documented in Calibre Admin Guide, "rcalibre utility"
    # $MGC_HOME/bin/rcalibre [rundir] \
    #    env_var_count {[env_var_1] [env_var_value]} ... \
    #    -mtflex primaryhost:port_number [-v] [-f]
    set    cmd "${mgc_home}/bin/rcalibre /tmp "
    append cmd "1 MGC_HOME $mgc_home "
    append cmd "-mtflex $primary_communication "
    exec rsh $name -n $cmd &
  }
}
```

Add Resources Command

Specifying the following command adds two remote CPU resources to the running Calibre job, for example,

```
calibre_dra -log calibre.log -add node456#2
```

The Calibre transcript shows the effect of the add command.

```
//  Connected to CPU on remote host node456.<domain_name>,
//  logfile = /tmp/CalibreRemoteLog.primary123….node456.<domain_name>…
//  Connected to CPU on remote host node456.<domain_name>,
//  logfile = /tmp/CalibreRemoteLog.primary123….node456.<domain_name>…
//  Applying licensing policy...
//  calibrehdrc license acquired.
//  calibredrc license acquired.
//  calresourceman license acquired.
//  calmtopcpro license acquired.
//  <DCA 52 0 60 N node456.<domain_name>
//  R 4 0 2 2 2 T 10 6 0 A 0 0 16251(P) 16253(P)>
```

Status Command

Specifying the status command shows information for the connected remote hosts, for example,

```
calibre_dra -log calibre.log -status
```

The information is printed to the terminal window.

```
REPORT REMOTES
  NAME: node123  CPU: 4/8  IP: xxx.xxx.xx.xx
  NAME: node456.<domain_name>  CPU: 2/4  IP: xxx.xxx.xx.xx
  PENDING: 0/0
  TOTAL: 4/12
END
```

Remove Resources Command

Specifying the following command removes two remote CPU resources from the running Calibre job, for example,

```
calibre_dra -log calibre.log -remove node456.<domain_name>#2
```

The removal of the two remote CPU resources can be confirmed by checking the Calibre transcript and rerunning the status command.

Calibre DRA Direct Connect Mode

The Calibre DRA Direct Connect mode is similar to Calibre DRA with the exception that the Calibre job is not responsible to launch extra remote processes. The Calibre job simply accepts the remote connections. An external system, such as a job scheduler or user, decides when and how to launch the extra remotes.

Direct Connect MTflex Example

In the following example from the configuration file, the Calibre process no longer launches the remote processes. This task is handled by the external system.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Refer to the *Calibre Administrator’s Guide* for the exact syntax and a complete description of the Calibre MTflex configuration file

```
// MTflex file statements.
// Protects against unresponsive hosts.
MONITOR SYSTEM INTERVAL 900 HEARTBEAT 4 PRINT
// Shows CPU load in log.
MONITOR LOCAL LOAD INTERVAL 20
// For DRA, computes CPU demand.
MONITOR UTILITY
// DRA Direct Connect mode: Don’t launch any remote hosts.
// This action is performed externally.
// Calibre job waits for external connections on specified ports.
// Connect on specified ports while job is running.
// Start computing when one remote connects (it has 60 seconds.)
// Job can be scaled up to 20000 remote processes.
// Make sure port numbers are available before using them.
LAUNCH MANUAL PORT 1234 WAIT 60 COUNT 1 MAXCOUNT 20000 CTRLPORT 5678
```

The two port numbers written in the MTflex configuration file must be available on the primary host when the Calibre job starts. In the Calibre DRA Direct Connect mode, the user is responsible to choose these port numbers. The ports must be specified in the calibre_dra_remote command launched from the remote hosts. This specification tells the remotes where to connect on the primary host.

Note

The specified LAUNCH MANUAL keyword options are mandatory for Calibre DRA Direct Connect mode usage. The LAUNCH MANUAL keywords, MAXCOUNT and CTRLPORT, are only used for the Calibre DRA Direct Connect mode.

The following example shows the Calibre command line arguments for -remotefile *mtflex*, calibre_dra_remote, and calibre_dra.

```
$MGC_HOME/bin/calibre -drc -hier -turbo -dra -remotefile mtflex \
rules.svrf > calibre.log &
# Connect one remote to start the job;
# LAYOUT DATA INPUT MODULE doesn't need many remotes.
ssh <remotehost> "$MGC_HOME/bin/calibre_dra_remote \
-master <primaryhost> -ports 5678:1234"
# Check job status and scale up the job when needed.
$MGC_HOME/bin/calibre_dra –log calibre.log –status
```

Effect of DRA Direct Connect in Calibre Transcript

In the following Calibre transcript example, the Calibre tool waits 60 seconds for a remote to be launched externally. The initial connection is for one remote only. During the job, the external system or user can scale up the job (up to MAXCOUNT) using calibre_dra_remote. The calibre_dra -status command can be used to help determine the resource adjustment.

```
// Mentor Graphics software executing under x86-64 Linux
// Running Mgc_home/pkgs/icv/pvt/calibre -dra -remotefile
// mtflex_dra_direct_connect -drc -hier -turbo rules.svrf
// Reading remote host configuration file:
// mtflex_dra_direct_connect
// Necessary for DRA to compute CPU demand.
MONITOR UTILITY
// Protection against unresponsive remotes.
MONITOR SYSTEM INTERVAL 300 HEARTBEAT 4 PRINT
MONITOR LOCAL LOAD INTERVAL 120
LAUNCH MANUAL PORT 1234 WAIT 60 COUNT 1 MAXCOUNT 20000 CTRLPORT 5678
// Performing manual launch of remote hosts: WAIT 60
// Connected to CPU on remote host <remote_hostname>
// List of connected remote cpus:
// REMOTE CPU <remote_hostname> (<ip_address>) : pid = …
// Calibre Controller Connection <remote_hostname>:<5678>
--- CALIBRE::DRC-H CALIBRE REMOTE PROTOCOL INITIALIZATION MODULE
COMPLETED.  CPU TIME = 0  REAL TIME = 13
...
_im10_ = LENGTH _im1_ >= 0.012 <= 1.4
-------------------------------------
// Connected to CPU on remote host <remote_hostname>, logfile = …
// Connected to CPU on remote host <remote_hostname>, logfile = …
// Connected to CPU on remote host <remote_hostname>, logfile = …
// Applying licensing policy...
<DCA 41 0 12 N <remote_hostname> R 20 20 2 6 6 T 7 4 0 A 0 0
14972(V)[14959]
 …>
_im10_ (HIER-FMF TYP=2 CFG=1 HGC=4788723 FGC=12774154 HEC=4788723
FEC=12774154 IGC=18992 VHC=F VPC=F)CPU TIME = 1 + 17  REAL TIME = 6  LVHEAP
= 51/715/715  OPS COMPLETE = 8 OF 20  ELAPSED TIME = 46
```

Direct Connect Scale-Up Example

In this example, a shell script is used to implement an “external system” that adjusts the remote resources based on information from a calibre_dra -status command

For example, calibre_dra -status for a Calibre job running a large LITHO simulation shows the following information.

```
REPORT BRIEF
  TIMESTAMP: 405
   OPS COMPLETE: 19(95%) of 20
   MODULE NAME: EXECUTIVE MODULE MTFlex HDB(0)
   OPNAME: metal1_opc OP-TIMESTAMP: 357
           OP-ECP: 0.0% OP-CLASS: LITHO:70
   ACTIVE REMOTE: 4 CELL QUEUE LENGTH: 78
   STRIPE QUEUE LENGTH: 59 TOTAL ACTIVE REMOTE: 4/4
 END
```

To scale the job up, if 500 remote hosts are available with 40 CPUs each, the following shell script is implemented.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

```
#!/bin/bash
i=1
while [ $i -le 40 ]
do
  foreach remotehost (`./list_remote_hosts.sh`)
     ssh $remotehost "$MGC_HOME/bin/calibre_dra_remote \
                     -master <primaryhost> -ports 5678:1234 &"
  end
  i=$(( $i + 1 ))
end
```

The shell script is a manual implementation. In a production environment, a job scheduler is recommended to control and optimize the utilization of jobs with large-scale CPU clusters.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Chapter 3
Calibre DRA Command Reference

Certain command line options are used when invoking and accessing a Calibre job with the -dra switch and executing the calibre_dra command.

This section is limited to the typical options and statements used in a Calibre run with Calibre DRA.

For a complete list of Calibre command line options, refer to the “Calibre nmDRC and Calibre nmDRC-H Command Line” section of the *Calibre Verification User’s Manual*.

calibre -dra

Initiates a Calibre run with Calibre DRA using typical command line options for launching a Calibre job with design rule checking and Calibre MTflex processing, or alternatively, launches a Calibre post-tapeout run with Calibre DRA and Calibre FullScale.

Usage

calibre -dra {**-drc -hier** | **-pto**} **-turbo** [*number_of_CPUs*] **-remotefile** ***filename***
 rule_file_name

Arguments

- **-dra**

  Required switch that specifies to run Calibre with Calibre DRA to manually access job status and adjust resource allocations during the run.

- **-drc**

  Required argument that specifies to perform design rule checking. Specifying this option without -hier performs flat design rule checking.

- **-hier**

  Required argument that specifies to perform design rule checking hierarchically.

- **-pto**

  Required argument that specifies to initiate a Calibre run using the Calibre FullScale engine.

- **-turbo** [*number_of_CPUs*]

  Required argument that specifies to use multithreaded parallel processing. The *number_of_CPUs* argument is optional; if not specified, Calibre runs on the maximum number of CPUs for which you have licenses.

- **-remotefile** ***filename***

  Required argument that specifies the path to a configuration file containing Calibre MTflex settings. Required argument for Calibre runs with Calibre DRA.

- ***rule_file_name***

  Required argument that specifies the path to the rule file.

Description

The -dra switch initiates Calibre DRA to run in conjunction with the specified options for a Calibre MTflex run. Calibre DRA enables you to manually adjust the job CPU resources from the command line and obtain status information during the Calibre run.

The following limitations apply when specifying the Calibre DRA -dra option:

- The -turbo_litho option is *not* supported in Calibre DRA.

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

- The -calcm option is *not* supported in Calibre DRA.
- The -remotedata option, that triggers the Calibre Remote Data Server (RDS) technology, is supported. However, Calibre DRA cannot add or remove a remote data server, because the Calibre RDS technology is not dynamic. Therefore, the initial number of RDS launched must be large enough to provide the necessary memory. The memory can be checked in the transcript as shown:

  ```
  //  MTflex RDS resources: 4GB, with recovery off
  ```

For more information on Calibre command line options and Calibre RDS, see "Command Line Options Reference Dictionary" and "Remote Data Server" in the *Calibre Administrator's Guide*.

Calibre DRA is supported for Calibre post-tapeout runs using the Calibre FullScale engine.

Contact your Siemens representative for more information on the tools and operations that are supported for running with Calibre DRA.

Examples

Example 1

Here is an example of typical command line options used for a Calibre run with Calibre MTflex processing and Calibre DRA.

```
$MGC_HOME/bin/calibre -dra -remotefile mtflex.dra -drc -hier -turbo \
    rules.svrf >& calibre.log &
```

Example 2

Here is an example of a typical command line invocation for a Calibre MTflex run with Calibre DRA and Calibre FullScale. Note that -pto option replaces the -drc and -hier options.

```
calibre -dra -pto -turbo -remotefile mtflex_pto.dra rules.svrf \
   >& calibre.log
```

See "Calibre FullScale Platform" and "Remotefile Contents" in the *Calibre Post-Tapeout Flow User's Manual* for setup information and supported operations for Calibre FullScale.

Related Topics

Running Calibre DRA

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

calibre_dra

You use the calibre_dra command to obtain status information and to adjust the resources for a Calibre job that is running.

Usage

calibre_dra -log *filename* [-status | {-add [*host#*]*count*} |
{-remove [*host#*]*count*}] [-controllertimeout *seconds*] [-nowait | -waittime *seconds*]
[-timeout *seconds*]

Arguments

- **-log *filename***

  Required argument that specifies the transcript file to connect to for the Calibre job.

- -status

  Optional argument that specifies to print the current remote status and other information for the Calibre job to the terminal.

- -add [*host#*]*count*

  Optional argument that specifies to add a specific number (*count*) of remote CPUs to the Calibre job.

  If the *host#* option is specified, the specified number (*count*) of CPUs is added from the named remote host, for example,

  ```
  calibre_dra -log dra.log -add hostxyz#3
  ```

  The exact host names as specified in REMOTE COMMAND DYNAMIC or the status output, including full qualification, must be specified. IP addresses are also accepted starting with Calibre 2020.4, for example,

  ```
  calibre_dra -log dra.log -add 123.456.78.910#3
  ```

  Multiple [*host#*]*count* specifications are separated by a colon (:), for example,

  ```
  calibre_dra -log dra.log -add host123#5:host456#3
  ```

  If the *host#* option is not specified, the specified number (*count*) of CPUs is added from the remote host(s) selected by the Calibre primary host, for example,

  ```
  calibre_dra -log dra.log -add 4
  ```

- -remove [*host#*]*count*

  Optional argument that specifies to remove a specific number (*count*) of remote CPUs from the Calibre job.

  If the *host#* option is specified, the specified number (*count*) of CPUs is removed from the named remote host, for example,

  ```
  calibre_dra -log dra.log -remove hostxyz#3
  ```

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

The exact host names as specified in REMOTE COMMAND DYNAMIC or the status output, including full qualification, must be specified. IP addresses are also accepted starting with Calibre 2020.4, for example,

```
calibre_dra -log dra.log -remove 123.456.78.910#3
```

Multiple [*host#*]*count* specifications are separated by a colon (:), for example,

```
calibre_dra -log dra.log -remove host123#5:host456#3
```

If the *host#* option is not specified, the specified number (*count*) of CPUs is removed from the remote host(s) selected by the Calibre primary host, for example,

```
calibre_dra -log dra.log -remove 4
```

- -controllertimeout *seconds*

  Optional argument that specifies a wait time in seconds to keep looking for controller information in the log file before exiting. This can be used to avoid showing an error when a Calibre job has just started and is not ready yet to communicate with calibre_dra. The default is to not wait.

- -nowait

  Optional argument that specifies *not* to wait for the completion of the status or the CPU add or remove command.

- -waittime *seconds*

  Optional argument that specifies a wait time in seconds for the completion of the CPU addition or removal. The default is 43200 seconds (12 hours).

- -timeout *seconds*

  Optional argument that specifies the timeout period in seconds for each individual communication with the Calibre job. The default is 240 seconds (4 minutes).

Description

Specify the calibre_dra command with arguments to interact with an already running Calibre job. The Linux terminal window used for the command line interactions must be able to connect and access the Calibre job primary and the local and remote resources on your system.

See “Allocating Resources with Calibre DRA” on page 22 for information on using the calibre_dra command.

Examples

Here is an example that uses the calibre_dra command with the -status argument.

```
calibre_dra -log calibre.log -status
```

calibre_dra_remote

You can use calibre_dra_remote to connect available remotes launched through an external system to a Calibre primary process running with Calibre DRA.

Usage

calibre_dra_remote -master *primary_hostname* -ports *control_port:data_port*
additional_options [-h]

Arguments

- **-master *primary_hostname***

  Specifies the name of the primary host for the Calibre job running with Calibre DRA.

- **-ports *control_port:data_port***

  Specifies the port numbers to connect to on the primary process.

- ***additional_options***

  Specifies the job-scheduler options for your external system and environment.

- [-h]

  Displays command usage information in standard output format.

Description

Initiates Calibre DRA Direct Connect mode to interface with your existing job-scheduler environment. In this mode, extra CPUs launched by your external system can connect directly to a running Calibre job specified with calibre -dra. You can use the calibre_dra_remote command along with the calibre_dra -status option to help you decide when and how to adjust the job resources. See "Calibre DRA Direct Connect Mode" on page 26 for more information.

Examples

Example 1

This example shows how to integrate calibre_dra_remote in the Univa Grid Engine environment to scale up a Calibre job running with calibre -dra by *n* CPUs.

```
qsub -cwd -b y -j y -V -N DRA_calibre -o DRA_calibre.log -t 1-<n>
 $MGC_HOME/bin/calibre_dra_remote -master <hostname> -ports 5678:1234
```

Example 2

This example shows how to integrate calibre_dra_remote in the LSF environment to scale up a Calibre job running with calibre -dra by *n* CPUs.

```
bsub -J "DRA_calibre[1-<n>]" -o DRA_calibre.log
 $MGC_HOME/bin/calibre_dra_remote -master <hostname> -ports 5678:1234
```

Note - Viewing PDF files within a web browser causes some links not to function. Use HTML for full navigation.

Index

Third-Party Information

Details on open source and third-party software that may be included with this product are available in the *<your_software_installation_location>/legal* directory.