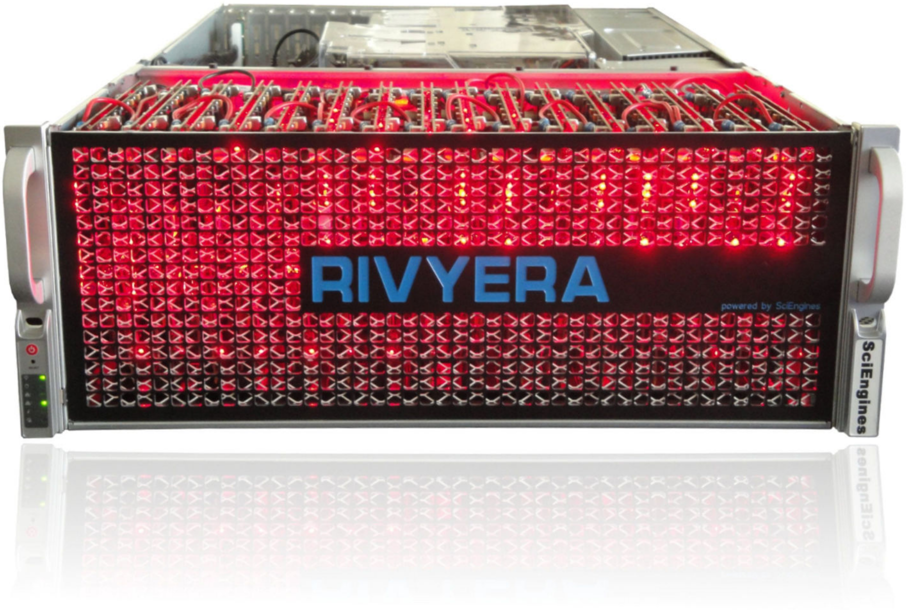
SciEngines
massively parallel computing

se_cluster

SciEngines Cluster Management Utility

Application User Guide

Version 1.95.03



May 24, 2023

SciEngines
Software Division

Released version

SciEngines Cluster Management Utility

Application User Guide
Version 1.95.03

SciEngines GmbH
Am Kiel-Kanal 2
24106 Kiel
Germany

Public

Released version

Contents

1 General information

1.1 Overview

This guide introduces you to the basic features of the SciEngines cluster management utility *se_cluster*.
The primary task of *se_cluster* is to administrate and use a cluster of SciEngines RIVYERA computers from a central node. While the term *FPGA cluster* refers to a single RIVYERA computer (multiple FPGAs interconnected inside one computer), the term *RIVYERA cluster* refers to multiple RIVYERA computers interconnected by network.
The SciEngines cluster management utility, *se_cluster* allows to configure multiple RIVYERA computers to a RIVYERA cluster. It makes these RIVYERA computers available for applications from a single point of use.

1.2 Scope

The *se_cluster Application User Guide* gives a general overview of the *se_cluster* features. A brief overview of the *se_cluster* command line options is provided.

1.3 Main features of se_cluster

Main features of *se_cluster* include:

- Remote management of RIVYERA computers from a central host
- Adding and removing RIVYERA computers from or to a cluster
- Remote starting, stopping and restarting of RIVYERA computers from a central instance
- Providing a clear overview of the status of all RIVYERA computers
- Making all RIVYERA computers accessible remotely

1.4 Master- and remote machines

In general, every RIVYERA computer in a RIVYERA cluster can be reached and used from any other RIVYERA computer within the same network and subnet. To do so, one RIVYERA computer (client) has to be configured to connect to another RIVYERA computer (server). The server will be visible on the client as a remote device.

A typical use case would define one RIVYERA computer as the cluster master and all others as remote machines. The master then may be used as a central place to manage and use all other RIVYERA computers that are part of the cluster. These other RIVYERA computers are referred to as *remote machines* or *slaves*.

Within this document the terms *master* and *remote machines* refer to the typical use case described above.

1.5 Communication between Master- and remote machines

For communication between RIVYERA computers in the cluster a system service called *se_cluster.service* needs to be enabled on all remote machines. On boot of the RIVYERA computer, the service automatically starts a daemon called *se_clusterServer* allowing clients to connect to the RIVYERA computer. The service enables remote management by the *se_cluster* utility, as well as remote utilization of the FPGAs by user applications.

To make a RIVYERA computer a remote device, you need to make sure *se_cluster.service* is running on that RIVYERA computer and at least one local machine (usually the FPGAs physically connected by PCIe) must be set to *enabled* and *shared*. Also please make sure that the *se_cluster* port (usually 7365/tcp) is not blocked by a firewall on the RIVYERA computer or in the network.

After initial configuration of the RIVYERA cluster, usually all operations are done from the master node only. The initial configuration of such a setup is described in detail in section Initial configuration of RIVYERA clusters.

2 Getting started using se_cluster

2.1 Executing se_cluster from the command line

To use *se_cluster*, log in to any RIVYERA computer and type `se_cluster` on the command line. This will execute *se_cluster* showing the usage help. The `se_cluster` binary is installed to `/opt/sciengines/current/bin/se_cluster`. A symlink in `/bin` allows execution of *se_cluster* from any working directory.

2.1.1 Accessing the built-in help

By typing `se_cluster`, `se_cluster --help`, or `se_cluster -h` on the command line, *se_cluster* is launched showing the built-in help, which lists all possible command line options, including a brief description for each option.

Example of accessing the built-in help:

```
$ se_cluster --help
Usage: se_cluster [OPTIONS] <COMMAND> [MACHINE_IDENTIFIER]

"OPTIONS":
   -h --help                       Print this help and exit.
   -v --version                    Print product version and exit.
      --nocolor                    Do not print any colors.
   -p --port                       Define the port used for local connection when reloading
        machines
      --compact                    Print a more compact output ("info" command, only).

"COMMAND":
   list    [<MACHINE_IDENTIFIER>]  Print a list of all machines available in this cluster.
                                   Optionally, a machine identifier may be provided.
   status  [<MACHINE_IDENTIFIER>]  Print the status of all machines
                                   Optionally, a machine identifier may be provided.
   info    <MACHINE_IDENTIFIER>    Print detailed information for given machine(s).
   start   <MACHINE_IDENTIFIER>    Start   given machine(s).
   stop    <MACHINE_IDENTIFIER>    Stop    given machine(s).
   restart <MACHINE_IDENTIFIER>    Restart given machine(s).
   share   <MACHINE_IDENTIFIER>    Share   given machine(s).
   unshare <MACHINE_IDENTIFIER>    Unshare given machine(s).
   enable  <MACHINE_IDENTIFIER>    Enable  given machine(s).
   disable <MACHINE_IDENTIFIER>    Disable given machine(s).
   serials <MACHINE_IDENTIFIER>    Print serial numbers for given machine(s).
   force-free <MACHINE_IDENTIFIER> Force to free given machine(s).
   swap    <IDX1,IDX1>             Swap given machine(s) with indices IDX1 and IDX2.
   reload  [<MACHINE_IDENTIFIER>]  Reload the machine list by allocating all devices.
                                   Optionally, a machine identifier may be provided
   add     <HOST[:PORT]>           Add all shared machines given by host address and port to the
        cluster.
   add     <DRIVER{DRIVER_PARAMS}> Add a device using driver DRIVER and parameters DRIVER_PARAMS.
   remove  <MACHINE_IDENTIFIER>    Remove the given machine(s) from the cluster.

"MACHINE_IDENTIFIER":

   The machine identifier can be one of the following values:
   * a single machine index                                     e.g. "0" / "1" / "2"
   * the word "all" to address all machines                     e.g. "all"
   * the word "local" to address all local machines             e.g. "local"
   * the word "pcie" to address all local pcie machines         e.g. "pcie"
   * the word "remote" to address all remote machines           e.g. "remote"
   * the word "simulation" to address all simulation machines   e.g. "simulation"
   * a comma separated list of the values named above           e.g. "0,1,2"

If machines on the local host are restarted, the machine list displayed by "list" will be updated
making machines available for use.
This can be done explicitly using the reload command and will happen automatically,
when starting, stopping or restarting remote machines, or if any local machine is given
by the machine identifier.

Examples:
   List machines:                  se_cluster list
   Show status of all machines:    se_cluster status
   Start all machines:             se_cluster start all
   Stop machines 0 and 1:          se_cluster stop 0,1
   Share local machines:           se_cluster share local
```

3 Initial configuration of RIVYERA clusters

This section describes how to initially setup a cluster of RIVYERA computers. All the steps listed here only need to be performed once, or when more RIVYERA computers are added to or removed from the RIVYERA cluster.

This section describes the recommended configuration for the typical use case of one master and multiple remote machines. In this setup a number of RIVYERA computers connected to the same network form a cluster, while one of them (the so called master) will act as the central node for managing and using all RIVYERA machines that are part of the cluster. The non master RIVYERA computers will be referred to as *remote machines* or *slaves*.

3.1 Configuration of RIVYERA Slaves

The following steps need to be performed once on every slave machine:

1. Log in to the slave machine
2. Verify that the locally connected FPGA cluster (PCIe) is present and idle:

```
$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | false  | pcie{0xaaa}
      1     |    1   |    1   |    1   | true   | false  |
           isim{XC6SLX150-3FGG676/0x499602d2}
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | false  | < idle >
      1     |    1   |    1   |    1   | true   | false  | < idle >
```

3. Set the local machine to `shared`, to allow the master to connect

```
$ se_cluster share 0
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | true   | pcie{0xaaa}
      1     |    1   |    1   |    1   | true   | false  |
           isim{XC6SLX150-3FGG676/0x499602d2}
```

   Notice that the output of the `shared` field has changed to `true`.

4. Make sure the shared machine is also `enabled`. If it is not enabled (see output of step 2) run the command:

```
$ se_cluster enable 0
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)
```

5. Make sure the firewall is not blocking the port used by *se_cluster*. This is port 7365 by default. The firewall is not part of the *se_cluster* utility, so the administration of the firewall may vary depending on the operating system used. On a CentOS 7 system, the following commands will open the port needed for *se_cluster* to work:

```
$ firewall-cmd --add-port 7365/tcp
$ firewall-cmd --add-port 7365/tcp --permanent
```

6. make sure the se_cluster service is running:

```
$ systemctl status se_cluster.service
  se_cluster.service - SciEngines cluster management utility
   Loaded: loaded (/usr/lib/systemd/system/se_cluster.service; enabled; vendor preset:
        disabled)
   Active: active (running) since Mon 2023-05-22 09:30:59 CEST; 1 day 18h ago
 Main PID: 1627 (se_clusterServe)
    Tasks: 1 (limit: 101472)
   Memory: 3.8M
   CGroup: /system.slice/se_cluster.service
           `-1627 /opt/sciengines/current/bin/se_clusterServer --daemon
```

7. In case the service is not running, start the service and enable the automatic starting at boot time using the following commands:

```
$ systemctl start se_cluster.service
$ systemctl enable se_cluster.service
```

3.2 Configuration of RIVYERA Master

On the Master, the following steps need to be performed once:

1. Log in to the master machine

2. Verify the local connected FPGA cluster is present and idle:

```
$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | false  | pcie{0xaaa}
      1     |    1   |    1   |    1   | true   | false  |
           isim{XC6SLX150-3FGG676/0x499602d2}
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | false  | < idle >
      1     |    1   |    1   |    1   | true   | false  | < idle >
```

3. Add all slaves to the cluster. To do so, run the following command on the master for each slave:

```
$ se_cluster add <slave IP address>
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)
```

4. Verify the slaves have been added successfully:

```
$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |   16   |  128   | true   | false  | pcie{0xaaa}
      1     |    1   |    1   |    1   | true   | false  |
           isim{XC6SLX150-3FGG676/0x499602d2}
      2     |    1   |   16   |  128   | true   | false  | remote{192.168.0.12:7365/0xaba}
      3     |    1   |   16   |  128   | true   | false  | remote{192.168.0.13:7365/0xaca}
      4     |    1   |   16   |  128   | true   | false  | remote{192.168.0.14:7365/0xada}
      5     |    1   |   16   |  128   | true   | false  | remote{192.168.0.15:7365/0xaea}
      6     |    1   |   16   |  128   | true   | false  | remote{192.168.0.16:7365/0xafa}
...
```

You may now manage and use all machines from the master host. Notice that the remote machines do not need be set to `shared` on the master.

5. Optionally and for better clarity, you may choose to reorder the machine list and assign the real machines to consecutive machine indices to your preference. See swap for more details on reordering the machine list.

```
$ se_cluster swap 0,1
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    1   |    1   | true   | false  |
           isim{XC6SLX150-3FGG676/0x499602d2}
      1     |    1   |   16   |  128   | true   | false  | pcie{0xaaa}
      2     |    1   |   16   |  128   | true   | false  | remote{192.168.0.12:7365/0xaba}
      3     |    1   |   16   |  128   | true   | false  | remote{192.168.0.13:7365/0xaca}
      4     |    1   |   16   |  128   | true   | false  | remote{192.168.0.14:7365/0xada}
      5     |    1   |   16   |  128   | true   | false  | remote{192.168.0.15:7365/0xaea}
      6     |    1   |   16   |  128   | true   | false  | remote{192.168.0.16:7365/0xafa}
...
```

4 se_cluster commands and options

se_cluster is always used in the following form :

```
se_cluster [options] <command> [command_parameter]
```

In this document parameters surrounded by square brackets "[]", are optional, while parameters surrounded by angle brackets "<>" are mandatory.

options affect the general behavior of *se_cluster*, e.g. changing the port that is used. They may be applied or left out independently from the command used. In most common use cases options are not needed.

The **command** is always mandatory. It specifies the action to be performed. If it is not provided, the usage help is shown.

The **command_parameter** may be given to specify the behavior of **command** more precisely. Usually it is used to specify the machine(s). It might be mandatory or not supported, depending on the command.

Table 1 shows all possible options for the *se_cluster* command line interface. If an option expects a parameter value, it is indicated by <parameter name>.

Table 2 shows all possible commands for the *se_cluster* command line interface. If a command expects a parameter value, it is indicated by <parameter name>.

4.1 Overview of se_cluster options

CLI	CLI long	Description
-h	--help	Print built-in help and exit.
-v	--version	Print product version and exit.
	--nocolor	Do not print any colors.
-p <port>	--port <port>	Specifies the TCP port to be used. The default port is 7365.
	--compact	Print a more compact output ("info" command, only).

Table 1. command line options

4.2 Overview of se_cluster commands

A detailed description of each option and possible parameter values can be found in section 5 of this document.

Command	Command parameter	Description
list	[<machine_identifier>]	Print a list of all machines available in this cluster. Optionally, a machine identifier may be provided.
status	[<machine_identifier>]	Print the status of all machines. Optionally, a machine identifier may be provided.
info	<machine_identifier>	Print detailed information for given machine(s).
start	<machine_identifier>	Start given machine(s).
stop	<machine_identifier>	Stop given machine(s).
restart	<machine_identifier>	Restart given machine(s).
share	<machine_identifier>	Share given machine(s).
unshare	<machine_identifier>	Unshare given machine(s).
enabled	<machine_identifier>	Enable given machine(s).
disable	<machine_identifier>	Disable given machine(s).
serial	[<machine_identifier>]	Print serial numbers for all or given machine(s).
temperature	[<machine_identifier>]	Print current and max temperatures for all or given machine(s).
force-free	<machine_identifier>	Force to free given machine(s).
swap	<IDX1>,<IDX2>	Swap given machine(s) with indices IDX1 and IDX2.
reload	[<machine_identifier>]	Reload the machine list by allocating all devices. Optionally, a machine identifier may be provided.
add	<ip[:port]>	Add all shared machines given by host address and port to the cluster.
add	<DRIVER{PARAMS}>	Add a device using driver DRIVER and parameters DRIVER_PARAMS.
remove	<machine_identifier>	Remove the given machine(s) from the cluster.

Table 2. se_cluster commands and command parameters

4.3 machine_identifier

Most of the commands supported by *se_cluster* may affect one or more machines of the cluster. Therefore these commands expect a so called *machine identifier*, which explicitly specifies which machines of the cluster should be effected by the command. The machine identifier may specify one or more machines and follows an intuitive syntax which is described below.

The machine identifier can be one of the following values:

A single machine index	e.g. "0" / "1" / "2"
The word "all" to address all machines	e.g. "all"
The word "local" to address all local machines	e.g. "local"
The word "pcie" to address all local pcie machines	e.g. "pcie"
The word "remote" to address all remote machines	e.g. "remote"
The word "simulation" to address all simulation machines	e.g. "simulation"
A comma separated list of the values named above	e.g. "0,1,2"

4.4 DRIVER

Instead of specifying machines using the `machine_identifier`, commands may also allow to specify machines based on the device driver used. Currently this is only supported by the add command. To specify machines by the driver, a driver name and a set of driver parameters must be given in the form <DRIVER{DRIVER_PARAMS}>, where `DRIVER` may be one of `pcie`, `isim` or `remote`. In case no driver parameters should be given, the driver name must be followed by empty curly brackets "{}" to make sure it is not interpreted as a host name.

The *DRIVER* string is build up as follows:

```
<DRIVER>    := <drv>"{"<param>"}"
<drv>       := "pcie"|"remote"|"isim"
```

The driver parameters are driver dependent:

`pcie:`

```
   <param>     := [<serial>]
   <serial>    := <serial in hexadecimal representation>
```

`remote:`

```
   <param>     := <host>[:<port>]["/"[<serial>]]
   <serial>    := <serial in hexadecimal representation>
   <host>      := <hostname>|<host ip address>
   <port>      := 0..65535
```

`isim:`

```
   <param>     := [[[<fpga_type>]"/"]<serial>]
   <serial>    := <serial in hexadecimal representation>
   <fpga_type> := "XC6SLX150-3FGG676"|"10AX115H4F34E3SG"
```

Local machines (`pcie` driver) are physically connected to the host on that `se_cluster list` was executed (local host).

Simulation machines are virtual FPGA clusters that are available on the local host.

Remote machines are virtual FPGA clusters, that are locally available, but forward all machine communication via network to a physical FPGA cluster that is connected to a remote host.

5 Detailed description of all available se_cluster commands

5.1 list

Shows a table of all available machines. The printed columns are *Machine idx*, *#Contr*, *#Slots*, *#FPGAs*, *Enabled*, *Shared*, and *Location*.

One line within the table represents one machine which is currently available or was available in the past. Generally, entries for machines are not deleted unless the user manually deletes them. This behavior is intended. It ensures always to identify the same machine with the same machine index, even if a preceding machine has been removed. The columns present the following information:

Machine idx:	A machine's index starting at 0.
#Contr:	The number of controllers belonging to that machine.
#Slots:	The number of slots/cards within that machine.
#FPGAs:	The total summed number of user FPGAs within that machine.
Enabled:	A boolean value indicating whether the machine is enabled.
Shared:	A boolean value indicating whether the machine is locally shared.
Location:	A location string used to identify the machine's controllers. (See details below)

5.1.1 Enabled / disabled machines

Machines that are disabled are blocked for allocation by user processes. Also a `disabled` machine cannot be used remotely. In contrast to a stopped machine, a disabled local machine is still able to communicate via the PCIe interface. It actually behaves like an idle machine, but cannot be allocated. Therefore it will show up as < `idle, disabled` > in the machine list.

See section enable and disable for more details on how to enable/disables machines.

5.1.2 Shared machines

A `shared` machine may be used remotely from any other RIVYERA computer (client) within the cluster. See initial configuration for more details on how to configure remote machines using *se_cluster*

See section share and unshare for more details on how to share/unshare machines.

5.1.3 Machine location

The *Location* string basically specifies the driver and its parameters used to identify a machine's controller. See driver for a detailed description of the machine driver syntax. As *se_cluster* is mainly intended to manage remote machines, the table also shows the remote connection information (IP address and port) of each machine, if available.

5.2 status

The status command shows the status of all machines available on this host.

The status field my be one of:

< idle >	Indicates that the machine is working and free for use
< idle, disabled >	Indicates that the machine is working, but disabled for usage
< not present / not shared >	Indicates that a local machine is stopped, or not connected physically, or in case of a remote machine, stopped, disabled, or unshared on the remote host
< error >	Indicates that a communication error between se_cluster and the remote server has happened
-	Indicates that there is no machine with this machine index, e.g. after removing a machine which is not the last machine of the list
usage information	Shows detailed information about the user and/or process that is using the machine, in case the machine is in use.

Machines in state `idle`, which are not `disabled` may be allocated by user applications. The other fields are the same as in the list command.

5.3 info <machine_identifier>

Show detailed information about the machine given by `machine_identifier`. Details are listed for each FPGA card separately. This includes serial number, firmware version and -build, hardware revision, number and type of FPGAs as well as uptime and lifetime information of the FPGA card. Also additional information for the controller(s) (FPGA card(s) connected to PCIe) are displayed.

See Examples for a full example output of the `info` command.

5.4 start <machine_identifier>

The `start` command starts one or more machines given by `machine_identifier`. Starting a machine enables it to communicate with the host using the PCIe interface, making it available for allocating by user processes or sharing by the *se_clusterServer* service. If a machine has already been started before, it just gets re-enumerated. This is equivalent to the reload command.

Starting machines is usually done automatically during boot of the RIVYERA computer. Manually starting a machine may be useful in case of errors or after a machine has been stopped explicitly.

After starting any machine, the reload command is executed automatically.

5.5 stop <machine_identifier>

The stop command stops one or more machines given by `machine_identifier`. Stopping a machine will disable it to communicate with the host using the PCIe interface. A stopped machine is neither usable nor able to report its status until it is started again using start. Therefore a stopped machine will show up as `< not present / not shared >` in the machine list.

5.6 restart <machine_identifier>

The restart command first stops and then starts the machine(s) given by the machine_identifier. It is equivalent to first running stop and then start for the same machine.

5.7 share <machine_identifier>

The `share` command sets the machine(s) given by `machine_identifier` to `shared` state. This allows the machine to be managed and used remotely from other hosts (clients) within the cluster.

Notice that this command only affects local machines. If remote machine should be shared, this command must be called on the remote host. See section Initial configuration of RIVYERA clusters for more details.

5.8 unshare <machine_identifier>

The `unshare` command removes the `shared` state from the machine(s) given by `machine_identifier`.

Notice that this command only affects local machines. If remote machine should be unshared, this command must be called on the remote host. See section Initial configuration of RIVYERA clusters for more details.

5.9 enable <machine_identifier>

The `enable` command enables the machine(s) given by `machine_identifier`, allowing it to be used locally or remotely, as long as it is present and idle.

Notice that this command only affects local machines. If remote machine should be enabled, this command must be called on the remote host. See section Initial configuration of RIVYERA clusters for more details.

5.10 disable <machine_identifier>

The `disable` command disables the machine(s) given by `machine_identifier`, blocking it for allocation by local and remote users, even if it is present and idle.

Notice that this command only affects local machines. If remote machine should be disabled, this command must be called on the remote host. See section Initial configuration of RIVYERA clusters for more details.

5.11 serials <machine_identifier>

The `serials` command prints a table for each machine given by `machine_identifier`. The table contains the columns: *SERIAL*, *LIFETIME*, *UPTIME*, *BUILD#* and *SIGNATURE*. Each card makes up an individual entry within that table. The *SERIAL* column contains the card's serial number in hexadecimal format. In the *LIFETIME* column, the card's lifetime in seconds is printed. *BUILD#* refers to the card's firmware build number. The *SIGNATURE* feature is used to sign the just said values. Note that the tables entries -especially the lifetime-changes upon `se_cluster reload`, or `se_cluster/se_machine [re]start`, only.

5.12 force-free <machine_identifier>

The `force-free` command allows a user to kill a process that uses a specific machine given by `machine_identifier`.

There are no other processes killed than those spawned locally. In case, a remote client computer allocates a local machine then the local server process would be killed rather than the client process. Nevertheless, the client process might notice such a situation and also terminate itself.

If the machine is not in use, or it is a remote machine not being used from this computer, then this error message would be printed:

```
$ se_cluster force-free 0
There is no local process using machine 0.
```

A user has the permission to use the force-free command for freeing a machine if:

- the machine is used locally and ...
  - ... the process is owned by the user
  - ... or the user is *root*
- ... or the machine is used from a remote client computer and ...
  - ... the user belongs to group *rivyera_admin*
  - ... or the user is *root*

An error message indicates such insufficient permission:

```
$ se_cluster force-free 0
Insufficient permission to force-free machine 0.
```

5.13 swap <IDX1,IDX2>

Swaps the machine indices of two machines given by `IDX1` and `IDX2`. This command may be used to reassign indices, e.g. after deleting a machine from the cluster, or for changing the order of the machines. Reordering the machines might be useful in some situations, e.g. when user programs try to allocate machine 0 by default, or just to increase clarity.

See the list command for displaying indices of all available machines.

5.14 reload

The reload command updates the machine list based on all configuration changes made. To do so *se_cluster* re-enumerates all machines that are physically connected to the local host, simulation machines that are installed and remote machines shown by list. While enumerating the machines, *se_cluster* tries to connect to each remote machine. If the connection to any of the machines fails the `reload` command also fails. After starting or restarting a machine, the reload command is executed automatically.

The `reload` command is usually run automatically after a machine is started, but might be useful in specific situations, e.g. if a remote machine has been physically changed (added / removed FPGA cards) and has been restarted locally on the remote host only. In that case on a client machine the machine list might be inconsistent and `reload` is necessary on the client side.

5.15 add <HOST[:PORT]>

The add command adds a new machine to the cluster. If a parameter in the form <HOST[:PORT]> is specified, a remote machine will be added. To be able to add a remote machine, the `se_cluster` service must be running on the remote machine. See General information and Initial configuration of RIVYERA clusters for a more detailed description of the communication within RIVYERA clusters.

If no port is specified *se_cluster* will try to add a remote machine using the default port *7365*. Also notice that the remote machine must be `enabled`, `shared`, and started on the remote host and that no firewall is blocking port used (7365 by default). Otherwise the *add* command will fail. See example of se_cluster add failing for trouble shooting.

In case of success the host will immediately show up in the machine list shown by the list command and may be used by local user processes.

5.16 add <DRIVER{DRIVER_PARAMS}>

The add command may also be invoked by directly specifying a driver string. This also allows to add local or simulation machines, or to add specific controllers of a remote machine. See se_cluster commands and options for a full description of the machine driver syntax.

5.17 remove <machine_identifier>

The remove command removes the machine(s) given by `machine_identifier` from the cluster.

The machine will immediately disappear from the machine list shown by the list command. In case the removed machine was not the last machine (the one with the highest machine index), the machine index will be left in the list, showing up with "-" signs in all fields of the machine list. This ensures that machine indices of following machines stay unchanged. Free machine indices are used again when adding new machines. Machines may be reordered manually. See swap for more details on reordering of machine indices.

6 Examples

6.1 Listing all available Machines/Hosts

```
$ se_cluster list
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Location
------------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    8   |   64   | true   | false  | pcie{0x2c7}
      1     |    1   |    1   |    1   | true   | false  | isim{XC6SLX150-3FGG676/0x499602d2}
      2     |    1   |    1   |    1   | true   | false  | isim{XC6SLX150-3FGG676/0x499602d3}
      3     |    1   |    5   |   40   | true   | false  | remote{192.168.0.12:7365/0x2db}
```

Figure 1. se_cluster list

Figure 1 shows an example output for `se_cluster list` with one physically attached machine, two simulation machines and one remote machine. The *location* column indicates one controller for each machine. Using the status command the machine statuses are printed out (see figure 2).

6.2 Showing status of all machines: All machines ok and idle

```
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Driver | Status
------------+--------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    8   |   64   | true   | false  | pcie   | < idle >
      1     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      2     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      3     |    1   |    5   |   40   | true   | false  | remote | < idle >
```

Figure 2. se_cluster status (all idle and okay)

6.3 Showing status of all machines: Some machines locked

```
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Driver | Status
------------+--------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    8   |   64   | true   | false  | pcie   | user: user1, command:
           se_mon, pid: 9949
      1     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      2     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      3     |    1   |    5   |   40   | true   | false  | remote | server: 192.168.0.12:7365,
           user: user2, command: se_decrypt, pid: 4935
      4     |    1   |    8   |  128   | true   | false  | remote | server: 192.168.0.13:7365,
           client: 192.168.0.11:49711, user: user3, server’s pid: 5061
```

Figure 3. se_cluster status (some locked)

In this example there is a total number of five machines configured on this host.

Machine 0 is the physically connected FPGAs on the local host.

Machines 1 and 2 are simulation machines.

Machine 3 and 4 are other RIVYERA computers connected remotely.

Machines 0,3 and 4 are currently in use.

Notice the slight difference in the output of the status command. For the local machine, the user called `user1` is locking the machine with the command `se_mon`. Also the process ID of the `se_mon` command is shown: 9949.

For the remote machines (machine 3 and 4) the status command outputs the server IP address and port, to make clear that this information is reported by the remote host.

In case of machine 3, the machine is locked directly by the user `user2`, running the command `se_decrypt` locally on the remote host (i.e. on 192.168.0.12).

In case of machine 4, the machine is not locked locally on the remote host (192.168.0.13), but by a client from another host (192.168.0.11). Therefore, no details about the client program are listed in the output. Also the displayed process ID is the process ID of the server running on the remote host (192.168.0.13), because this is actually the process that locks the machine, while all communication between the FPGAs and the client program are tunneled via the server process.

While any non simulation machine is locked (i.e. in use) that machine cannot be used by any other process. Also no machine on the same host can be started / stopped while any local machine is locked.

6.4 Showing status of all machines: One machine unavailable

```
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Driver | Status
------------+--------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    2   |   16   | true   | false  | pcie   | < idle >
      1     |    1   |    8   |   64   | true   | false  | remote | < not present/not shared >
      2     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      3     |    1   |    1   |    1   | true   | false  | isim   | < idle >

INFO: Machines with status "< not present/not shared >" could not be located. In case of a
     remote machine, you need to make sure, the machine is present and set to enabled and shared
     on the remote computer.
```

Figure 4. se_cluster status (some unavailable)

This example shows the output of the status command where one machine (machine 1) is currently not available. That means the device is either not present on the remote host (e.g. not connected physically via PCIe), not `enabled` on the remote host, or not shared on the remote host. See enable and share for enabling and sharing the machine on the remote host.

6.5 Adding a remote machine: Server unavailable

```
$ se_cluster add 192.168.0.19
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Unable to connect to host 192.168.0.19:7365.
Unable to add: "192.168.0.19".
If this is a remote device, make sure, it is present, shared and enabled.
```

Figure 5. se_cluster add (failed)

This example shows a failed `add` command. If you see this output this might indicate one of the following problems:

- A general network problem
- A mistake in firewall configuration - the se_cluster port (7365 by default) must be reachable on the remote host
- The machine on the remote host is not present, not `enabled`, or not `shared` on the remote host (check with se_cluster status)
- The `se_cluster` service is not running on the remote host (check with `systemctl status se_cluster.service` on the remote host)
- IP or port of the remote machine is incorrect

6.6 Starting all machines of the cluster at once

```
$ se_cluster start all
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Starting may take a few minutes. Please be patient.
***************** SN: 0x00000aaa ******************.
Success.
***************** SN: 0x00000aba ******************.
Success.
***************** Reloading all Devices *******.
```

Figure 6. se_cluster start

The output displays the status of the `start` command ("Success" in this example) for each local and remote machine individually. Notice the automatic reload of all devices after starting any machine. This automatically updates the machine information displayed by the list command.

6.7 Stopping specific machines only

This example shows how to stop 2 machines of the cluster, given the corresponding machine indices (0 and 1). The output of the status command before and after stopping of the machines is displayed in this example as well:

```
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Driver | Status
------------+--------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    2   |   16   | true   | false  | pcie   | < idle >
      1     |    1   |    8   |   64   | true   | false  | remote | < idle >
      2     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      3     |    1   |    1   |    1   | true   | false  | isim   | < idle >
```

Figure 7. se_cluster status (before stop)

```
$ se_cluster stop 0,1
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

***************** SN: 0x00000aaa ******************.
Success.
***************** SN: 0x00000aba ******************.
Success.
```

Figure 8. se_cluster stop

```
$ se_cluster status
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)


Machines available in this cluster:

Machine idx | #Contr | #Slots | #FPGAs | Enabled| Shared | Driver | Status
------------+--------+--------+--------+--------+--------+--------+------------------------------
      0     |    1   |    2   |   16   | true   | false  | pcie   | < not present/not shared >
      1     |    1   |    8   |   64   | true   | false  | remote | < not present/not shared >
      2     |    1   |    1   |    1   | true   | false  | isim   | < idle >
      3     |    1   |    1   |    1   | true   | false  | isim   | < idle >

INFO: Machines with status "< not present/not shared >" could not be located. In case of a
     remote machine, you need to make sure, the machine is present and set to enabled and shared
     on the remote computer.
```

Figure 9. se_cluster status (after stop)

Notice that the status of machine 0 and 1 shows `< not present/not shared >` as the machines are physically stopped and thus no longer available until started again.

6.8 Displaying information about a RIVYERA computer

The following example shows the output of the info command for a RIVYERA computer with 4 FPGA cards and one controller:

```
$ se_cluster info 0
Running se_cluster version 1.95.03
Copyright (c) 2013-2023, SciEngines GmbH
All rights reserved.
SciEngines Rivyera Host-API version 1.95.06  (BUILD 1373)

Machine 0 Information:
  Controller 0:
    Timestamp:              Mon May 22 09:32:04 2023 (updated 1 day 3:30 ago)
    Serial:                 0x00000aba
    Rel. Serial DN:         0x00000aba
    Rel. Serial UP:         0x00000aba
    Location:               remote{192.168.2.216:7365/0x2c7}
    Slot Index:             0
    Enabled:                true
    Shared:                 false
    Status:                 not present/not shared
  Slot 0:
    Serial:                 0x00000aba
    Firmware Ver:           1.91.11
    Firmware Build:         1091
    Hardware Rev.:          3
    #FPGAs:                 8
    FPGA-Type:              XC6SLX150-3FGG676
    Lifetime:               339 days 15:05:59 (29343959s)
    Uptime:                 0 days 23:29:14 (84554s)
  Slot 1:
    Serial:                 0x00000aca
    Firmware Ver:           1.91.11
    Firmware Build:         1091
    Hardware Rev.:          3
    #FPGAs:                 8
    FPGA-Type:              XC6SLX150-3FGG676
    Lifetime:               339 days 15:05:47 (29343947s)
    Uptime:                 0 days 23:29:14 (84554s)
  Slot 2:
    Serial:                 0x00000ada
    Firmware Ver:           1.91.11
    Firmware Build:         1091
    Hardware Rev.:          3
    #FPGAs:                 8
    FPGA-Type:              XC6SLX150-3FGG676
    Lifetime:               339 days 15:05:29 (29343929s)
    Uptime:                 0 days 23:29:14 (84554s)
  Slot 3:
    Serial:                 0x00000aea
    Firmware Ver:           1.91.11
    Firmware Build:         1091
    Hardware Rev.:          3
    #FPGAs:                 8
    FPGA-Type:              XC6SLX150-3FGG676
    Lifetime:               339 days 15:06:58 (29344018s)
    Uptime:                 0 days 23:29:14 (84554s)
```

Figure 10. se_cluster info 0

Imprint:

SciEngines GmbH
Am Kiel-Kanal 2
D-24106 Kiel Germany

Phone:	+49(0)431-9086-2000
Fax:	+49(0)431-9086-2009
E-Mail:	info@SciEngines.com
Internet:	www.SciEngines.com
CEO:	Gerd Pfeiffer
Commercial Register:	Amtsgericht Kiel
Commercial Register No.:	HR B 9565 KI
VAT-Identification Number:	DE 814955925