Table of Contents

1. Introduction 2

2. delilah console client 2

2.1 Getting help 3

2.2 management commands 4

2.2.1 cluster 4

2.2.2 ls\_connections 6

2.3 data commands 6

2.3.1 ls 7

2.3.2 rm 7

2.3.4 set\_queue\_property 7

2.3.5 unset\_queue\_property 7

2.3.6 show\_stream\_block 7

2.3.7 ls\_blocks 8

2.3.8 ls\_buffers 8

2.4 delilah commands 8

2.4.1 help 8

2.4.2 ps 8

2.4.3 set\_mode 8

2.4.4 set 8

2.4.5 unset 9

2.4.6 connect 9

2.4.7 disconnect 9

2.4.8 alerts 9

2.4.9 show\_alerts 9

2.4.10 open\_alerts\_file 9

2.4.11 close\_alerts\_file 9

2.4.12 send\_alert 10

2.4.13 log 10

2.4.14 trace 10

2.4.15 verbose 10

2.4.16 wverbose 10

2.4.17 wdebug 10

2.4.18 wwrites 10

2.4.19 wreads 11

2.4.20 wtrace 11

2.5 local commands 11

2.5.1 ls\_local 11

2.5.2 rm\_local 11

2.5.3 history 11

2.6 module commands 12

2.6.1 ls\_modules 12

2.6.2 ls\_operations 12

2.6.3 ls\_datas 13

2.6.4 push\_module 13

2.6.5 reload\_modules 13

2.7 push&pop commands 14

2.7.1 push 14

2.7.2 pop 14

2.7.3 connect\_to\_queue 14

2.7.4 disconnect\_from\_queue 14

2.7.5 ls\_local\_queues 14

2.7.6 show\_local\_queue 15

2.7.7 push\_queue 15

2.8 stream commands 15

2.8.1 ls\_stream\_operations 15

2.8.2 add\_stream\_operation 15

2.8.3 rm\_stream\_operation 16

2.8.4 run 16

2.8.5 cancel\_stream\_operation 16

2.8.6 set\_stream\_operation\_property 16

2.8.7 unset\_stream\_operation\_property 16

2.8.8 add\_queue\_connection 17

2.8.9 rm\_queue\_connection 17

2.8.10 ls\_queues\_connections 17

2.8.11 ps\_stream 17

2.8.12 ls\_workers 17

2.8.13 init\_stream 17

2.8.14 ps\_workers 18

2.8.15 defrag 18

# 1. Introduction

Samson is a MapReduce platform, developed for processing large volumes of information (>1GB daily). delilah is its console client, used to connect to and to interact with the Samson platform.

# 2. delilah console client

'delilah' is the samson shell, used to interact with the samson platform.

Synopsis:

delilah <IP address> <port> [options]

delilah supports a number of options, the most important being:

-u print usage on the screen and then exit

-v verbose mode - upto five v:s accepted

--version print version on the screen and then exit

-user User to connect to SAMSON cluster

-password Password to connect to SAMSON cluster

-memory memory in GBytes

-load\_buffer\_size load buffer size in MBytes

-f File with commands to execute

-command Single command to be executed

delilah is included in the samson package and installed with the rest of the platform; it is an executable and the supported platforms are:

o Ubuntu,

o RedHat,

o CentOS and

o MacOS ?

delilah doesn't have to run in any of the nodes of the samson cluster it connects to, but it does need to be able to connect to all nodes in the cluster.

In order to tell delilah what cluster to connect to, its two command line parameters are used (IP Address and port).

The port parameter defaults to 1324, which is the default port for the samson worker, while the IP-address parameter defaults to 'localhost', i.e. the same node where delilah has been started.

As a shell, delilah offers a number of convenient features, such as:

o history Using the cursor keys, the delilah user can easily browse the history of

previous commands, to invoke a command previously invoked.

The history is kept on file, thus survives between sessions.

o auto-completion Using the tab key, the delilah user can auto-complete

command names, its options, operation names,

dataset names etc.

## 2.1 Getting help

The command to use to get help in delilah is called 'help'.

Once invoked, the following listing is presented on the screen:

================================================================================

SAMSON v 0.6.1

================================================================================

SAMSON is a distributed platform for efficient processing of unbounded streams of big data

Authors: Andreu Urruela, Grant Croker, J.Gregorio Escalada & Ken Zangelin

Telefonica I+D 2010-2012

--------------------------------------------------------------------------------

help all .................. get a list of all available commands

help categories ........... get a list of command categories

help <command> ............ get detailed information for a command

help <category> ........... get a list of commands for a particular category

--------------------------------------------------------------------------------

The help categories are:

o data

o delilah

o local

o management

o modules

o push&pop

o stream

To see the listing of all commands for a category, use the following command:

delilah> help <category name>

To see the help of an individual command, use 'help <command name>', for example:

delilah> help ls

-----------------------------------------------------------------------

COMMAND ls ( data )

-----------------------------------------------------------------------

DESCRIPTION: Show a list of all data queues in the system

USAGE:

ls [-rates] [-blocks] [-properties] [-group group\_field]

-rates Information about total size and current rate

-blocks Detailes information about blocks

-properties Get properties assigned to queues

-group X Group results by a particular field

-----------------------------------------------------------------------

## 2.2 management commands

'management' is the delilah command category to maintain clusters.

The commands in this category are:

o cluster - to add/remove/view cluster members

o ls\_connections - to see the status of all connections in the cluster

A SAMSON cluster is a set of SAMSON nodes that work together.

Nodes can easily be added or removed from a SAMSON cluster using delilah.

### 2.2.1 cluster

The cluster command has a number of subcommands:

o info Show current cluster definition and connection status.

o connections Show more information about current connections.

o pending Show pending packets for unconnected workers.

o connect Connect to another SAMSON cluster instance.

o add Add a node to this cluster.

A samsond clear instance should be running there.

o remove Remove one of the involved workers. The worker id should

be provided (see cluster info).

o get\_my\_id Get this delilah identifier (see ls\_connections).

Exercise: create a new samson cluster with three nodes

1. Make sure a 'free' samson worker is running in each of the nodes. With 'free' is meant that the worker does not belong to any cluster. If a samson worker already belongs to a cluster and you wish for it to belong to another cluster, just connect to it using Delilah and issue the 'cluster info' command. Now you have the 'id' of the node and you can issue the 'cluster remove <id>' command to free the node.

2. Connect to one of the three nodes (doesn't matter which one you start with), using the 'cluster connect' command. As this node is (was) a free node, a cluster containing only this node is created automatically.

3. Now add the other two nodes, using the 'cluster add' command.

4. To check that everything worked, use the 'cluster info' command and you should see the three nodes:

#### 2.2.1.1 cluster info

The cluster subcommand 'info' shows the list of nodes in the connected cluster.

Synopsis:

delilah> cluster info

Example output:

/----------------------------------------------------------------\

| Cluster 204695284815 ( version 2 ) |

|----------------------------------------------------------------|

| Worker | Host | Status |

|--------+-------------+-----------------------------------------|

| 0 | torkel:1324 | Connected In: 0 B/s Out: 0 B/s |

| 1 | torkel:2000 | Connected In: 0 B/s Out: 0 B/s |

| 2 | torkel:2001 | Connected In: 86.0 B/s Out: 0 B/s |

\----------------------------------------------------------------/

#### 2.2.1.2 cluster connections

The cluster subcommand 'connections' shows a listing with detailed information of all currently open connections.

Synopsis:

delilah> cluster connections

Example output:

/------------------------------------------------\

| Connections |

|------------------------------------------------|

| Name | Host | In | Out |

|----------+-------------+-----------+-----------|

| worker\_0 | torkel:1324 | 86.0 B/s | 3.00 B/s |

| worker\_1 | torkel:2000 | 86.0 B/s | 3.00 B/s |

| worker\_2 | torkel:2001 | 86.0 B/s | 3.00 B/s |

\------------------------------------------------/

#### 2.2.1.3 cluster pending

The cluster subcommand 'pending' shows all pending packets for unconnected workers:

Synopsis:

delilah> cluster pending

Example output:

/------------------------------\

| Pending packets |

|------------------------------|

| Connection | #Packets | Size |

|------------+----------+------|

...

\------------------------------/

#### 2.2.1.4 cluster connect

The cluster subcommand 'connect' lets you connect to a samson cluster. If you are already connected to a cluster, delilah will disconnect from that cluster and connect to the requested cluster, unless delilah detects outstanding operations. If so, to override this, the user would have to disconnect manually, using the command 'cluster disconnect' before connecting to the new cluster. To connect to a cluster, it is enoung to connect to any of the samson workers in the cluster.

Synopsis:

delilah> cluster connect host:port

#### 2.2.1.5 cluster add

To add a new node to an existing cluster (the cluster you're currently connected to), the subcommand 'add' is used. cluster add has one parameter, the name of the node where the samson worker instance is running. In case this samson worker has been started with a port that is not the default samson worker port (1324), this port must be added to the node using colon (:) as separator.

Synopsis:

delilah> cluster add host:port

#### 2.2.1.6 cluster remove

To remove a worker instance from a cluster, the subcommand 'remove' is used. A removed worker instance is immediately informed and this worker instance is now cluster-free, meaning that it can be added to any cluster. One option is required for this command - the host of the worker instance that is to be removed from the cluster. In case the worker is not using the default worker port, the port number is added to the hostname, using a colon as separator.

Synopsis:

delilah> cluster remove host:port

#### 2.2.1.7 cluster get\_my\_id

To retrieve the cluster id of the currently connected cluster, the subcommand 'get\_my\_id' is used. This identification number of the cluster can later be used in various listings to detect that connections are with this instance of delilah, and not another one. It is normal to have many instances of delilah running simultaneously.

Synopsis:

delilah> cluster get\_my\_id

### 2.2.2 ls\_connections

The management command ls\_connections shows the status of all connections in a cluster. E.g.:

Synopsis:

delilah> ls\_connections

## 2.3 data commands

There are seven different delilah commands to view and modify data queues and blocks:

o ls Show a list of all data queues in the system

o rm Remove a queue. Usage: rm queue

o set\_queue\_property Specify the value of property <property> for queue <queue>

o unset\_queue\_property Remove a previously defined property property for a queue

o show\_stream\_block Show data activity in a particular stream block

o ls\_blocks Show the list of data blocks managed in a SAMSON cluster. This is a debug tool

o ls\_buffers Show the list of data buffers managed in a SAMSON cluster. This is a debug tool

### 2.3.1 ls

'ls' is perhaps the most used command in delilah. It is used to see a listing of all data queues in the system and has a number of options:

-rates Information about total size and current rate

-blocks Detailes information about blocks

-properties Get properties assigned to queues

-group X Group results by a particular field

Synopsis:

delilah> ls [<queue name>]

### 2.3.2 rm

The 'rm' command removes a data queue from the system.

Synopsis:

delilah> rm <queue name>

### 2.3.4 set\_queue\_property

Properties of a Samson data queue is a set of variables associated to the queue.

<An example or two of what properties can accomplish>

Synopsis:

delilah> set\_queue\_property <queue> <property> <value>

### 2.3.5 unset\_queue\_property

The 'unset\_queue\_property' is used to remove a previously set property for a queue.

Synopsis:

delilah> unset\_queue\_property <queue> <property>

### 2.3.6 show\_stream\_block

The 'show\_stream\_block' command shows data activity in a particular stream block. It shows input and output queues, state queues, stream operations and internal states included in inner stream blocks.

Synopsis:

delilah> show\_stream\_block <path> [-rates]

### 2.3.7 ls\_blocks

The 'ls\_blocks' command shows the list of data blocks managed in the SAMSON cluster. This command is a tool used for debugging.

Synopsis:

delilah> ls\_blocks

### 2.3.8 ls\_buffers

The 'ls\_buffers' command shows the list of data buffers managed in the SAMSON cluster. This command is a tool used for debugging.

Synopsis:

delilah> ls\_buffers

## 2.4 delilah commands

The 'delilah' category of commands are typically the commands that do not invoke anything outside of the delilah executable, i.e. no message is sent to any worker in order to execute this type of commands. The exception to this rule are connect/disconnect and wverbose/wdebug

### 2.4.1 help

The 'help' command has been extensively described in chapter 2.1

### 2.4.2 ps

The 'ps' command is used to view information about delilah processes

Synopsis:

delilah> ps [-clear] [<id>]

The option '-clear', if set, removes all finished or erroneous processes. If a parameter (id) is sent to the 'ps' command, then extensive information about that process is shown.

### 2.4.3 set\_mode

The 'set\_mode' command is used to select the delilah working mode, that can take one of three values: 'normal', 'database', and 'logs'.

Synopsis:

delilah> set\_mode <normal|database|logs>

Here perhaps a short explanation of the three modes ... and the keyboard shortcuts to change modes ...

### 2.4.4 set

The 'set' command is used for setting an environment variable in delilah.

Synopsis:

delilah> set [<name>] [<value>]

When using 'set' without parameters, a listing of all set environment variables is shown.

### 2.4.5 unset

The 'unset' command is used for removing an environment variable in delilah.

Synopsis:

delilah> unset <name>

### 2.4.6 connect

The 'connect' command is used to connect the delilah to a SAMSON cluster or node.

Synopsis:

delilah> connect host [-port X] [-user X] [-password X]

### 2.4.7 disconnect

The 'disconnect' command is used to disconnect the delilah from its current SAMSON cluster.

Synopsis:

delilah> disconnect

### 2.4.8 alerts

The 'alerts' command is to activate or deactivate the delilah functionality to show incoming alerts on screen.

Alerts in the Samson platform is a mechanism that the samson workers use, to send messages between themselves and all connected delilahs.

Synopsis:

delilah> alerts <on>|<off>

### 2.4.9 show\_alerts

The 'show\_alerts' command is used to view a list of the last 10 incoming alerts. The number of alerts to view can be modified using the built-in option '-lines'

Synopsis:

delilah> show\_alerts [-lines (number)]

### 2.4.10 open\_alerts\_file

The 'open\_alerts\_file' command is used to open a file in the local file system to store all incoming alerts. This file must be closed using the 'close\_alerts\_file' command.

Synopsis:

delilah> open\_alerts\_file <file name>

### 2.4.11 close\_alerts\_file

The 'close\_alerts\_file' command is used to close the alerts file, previously opened with the 'open\_alerts\_file' command.

Synopsis:

delilah> close\_alerts\_file

### 2.4.12 send\_alert

The 'send\_alert' command is used to send an alert to all connected delilahs using a random worker as sender

Synopsis:

delilah> send\_alert [-worker X] [-error] [-warning] message

-worker X Use only this worker as broadcaster

-error Mark this trace as an error for correct visualization

-warning Mark this trace as a warning for correct visualization

### 2.4.13 log

This command is obsolete and should not be used.

To view the log files, the Samson logClient executable should be used

### 2.4.14 trace

Activate or deactivate trace levels in the Samson cluster.

Synopsis:

delilah> trace <on>|<off>

### 2.4.15 verbose

The 'verbose' command activates or deactivates verbose mode for the delilah.

When 'on', extra information for commands executed in delilah is shown on screen.

Synopsis:

delilah> verbose <on>|<off>

### 2.4.16 wverbose

The 'wverbose' command sets the verbose level for the samson workers in the cluster. The higher the number, the more verbose the workers get.

Synopsis:

delilah> wverbose <0-5>|<off>

### 2.4.17 wdebug

The 'wdebug' command sets the debug level for the samson workers in the cluster.

Synopsis:

delilah> wdebug <on>|<off>

### 2.4.18 wwrites

The 'wwrites' command sets the 'writes' debug level for the samson workers in the cluster.

Synopsis:

delilah> wwrites <on>|<off>

### 2.4.19 wreads

The 'wreads' command sets the 'reads' debug level for the samson workers in the cluster.

Synopsis:

delilah> wreads <on>|<off>

### 2.4.20 wtrace

The 'wtrace' command sets the 'trace' level for the samson workers in the cluster.

Synopsis:

delilah> wtrace <trace level comma-separated ranges>

The samson workers have 256 trace levels, from 0 to 255. Using this command, delilah changes the trace levels of all workers in the cluster. The option for 'wtrace' is a comma separated list of ranges, E.g. 0-4,67-99

## 2.5 local commands

The category 'local' groups the delilah commands that have no interaction whatsoever with any of the

workers in the cluster and that interacts with the local filesystem.

### 2.5.1 ls\_local

The 'ls\_local' command shows a listing of the current directory with relevant information about local data-sets.

Synopsis:

delilah> ls\_local

Example output:

/-----------------------------------------\

| Local files ( \* ) |

|-----------------------------------------|

| Name | Type | Size | Format |

|-----------+------+-------------+--------|

| dirr | DIR | | |

| test\_file | FILE | 2.00 bytes | - |

\-----------------------------------------/

### 2.5.2 rm\_local

The 'rm\_local' command removes a local file or directory (and all its contents).

Synopsis:

delilah> rm\_local <local filename>

### 2.5.3 history

As explained in 1.2.X, delilah keeps the command history on file to be retrieved for reusal in the same session or for future sessions of delilah. To see the command history, the command 'history' is used.

Synopsis:

delilah> history

## 2.6 module commands

A module in the samson platform is a shared library that implements functionality for the platform and thus extends it. The samson platform allows for modules to be loaded in run-time. Detailed information on how to create a new module is found in the document XXX.

### 2.6.1 ls\_modules

To view a list of all installed modules in the platform, the command 'ls\_modules' is used.

Synopsis:

delilah> ls\_modules

/----------------------------------------------------------------------\

| ls\_modules |

|----------------------------------------------------------------------|

| worker\_id | name | version | #operations | #datas | author |

|-----------+--------+---------+-------------+--------+----------------|

| 0 | system | 0.2 | 8.00 | 22.0 | Andreu Urruela |

| 2 | system | 0.2 | 8.00 | 22.0 | Andreu Urruela |

| 1 | system | 0.2 | 8.00 | 22.0 | Andreu Urruela |

\----------------------------------------------------------------------/

### 2.6.2 ls\_operations

An operation in delilah is a ...

To see all available operations in delilah, the command 'ls\_operations' is used.

Synopsis:

delilah> ls\_operations

Example output:

/---------------------------------------\

| ls\_operations |

|---------------------------------------|

| worker\_id | name | type |

|-----------+---------------+-----------|

| 2 | system.lines | splitter |

| 2 | system.map | map |

| 2 | system.map2 | map |

| 2 | system.map3 | map |

| 2 | system.parse | parser |

| 2 | system.reduce | reduce |

| 2 | system.str | parserOut |

| 2 | system.update | reduce |

| 0 | system.lines | splitter |

| 0 | system.map | map |

| 0 | system.map2 | map |

| 0 | system.map3 | map |

| 0 | system.parse | parser |

| 0 | system.reduce | reduce |

| 0 | system.str | parserOut |

| 0 | system.update | reduce |

| 1 | system.lines | splitter |

| 1 | system.map | map |

| 1 | system.map2 | map |

| 1 | system.map3 | map |

| 1 | system.parse | parser |

| 1 | system.reduce | reduce |

| 1 | system.str | parserOut |

| 1 | system.update | reduce |

\---------------------------------------/

### 2.6.3 ls\_datas

A 'data' in the samson platform is a data-type that can be used in creating Samson modules. To view a list of all the datas for the platform, the 'ls\_datas' command is used.

Synopsis:

delilah> ls\_datas

Example output:

/-------------------------------------------------------\

| ls\_datas |

|-------------------------------------------------------|

| worker\_id | name | help |

|-----------+------------------------+------------------|

| 2 | system.ComplexTimeSlot | Help coming soon |

| 2 | system.Date | Help coming soon |

| 2 | system.DateComplete | Help coming soon |

| 2 | system.Double | Help coming soon |

| 2 | system.Float | Help coming soon |

| 2 | system.Int16 | Help coming soon |

| 2 | system.Int32 | Help coming soon |

| 2 | system.Int8 | Help coming soon |

| 2 | system.String | Help coming soon |

| 2 | system.StringVector | Help coming soon |

| 2 | system.Time | Help coming soon |

...

\-------------------------------------------------------/

### 2.6.4 push\_module

The 'push\_module' command is used to add a new module to the platform.

Synopsis:

delilah> push\_module <file> <module\_name>

### 2.6.5 reload\_modules

In case a module is changed, it must be reloaded and the delilah cvommand to accomplish this is 'reload\_modules'.

Synopsis:

delilah> reload\_modules

## 2.7 push&pop commands

The category 'push&pop' includes the commands to push queues into the samson platform from local files in the file system, to pop them, i.e. create a local file with a copy of the information in the queue, as well as commands to manipulate and view queues.

### 2.7.1 push

The command 'push' pushes the content of a list of local file/directories to a queue in the samson platform.

Synopsis:

delilah> push file1/dir1 file2/dir2 file3/dir3 ... fileN/dirN <queue name>

### 2.7.2 pop

The command 'pop' copies the content of a samson queue to a local directory. This command also works for binary queues. To later view the content, the executable samsonCat is used. samsonCat is distributed with the samson platform.

<reference to samsonCat doc?>

Synopsis:

delilah> pop <queue> <local\_file\_or\_dir> [-force] [-show]

### 2.7.3 connect\_to\_queue

Using the command 'connect\_to\_queue', you can connect to a queue to receive live data from the samson platform. Received data will be stored in a local directory called stream\_out\_<queue>

Synopsis:

delilah> connect\_to\_queue <queue>

### 2.7.4 disconnect\_from\_queue

To stop the flow of 'connect\_to\_queue', the command 'disconnect\_from\_queue' is used.

Synopsis:

delilah> disconnect\_from\_queue <queue>

### 2.7.5 ls\_local\_queues

To view a list of local queues (in the current directory), the command 'ls\_local\_queues' is used. These queues have been typically being downloaded with command pop.

Synopsis:

delilah> ls\_local\_queues

Example output:

/---------------------------------------------------------------------\

| Local queues ( \* ) |

|---------------------------------------------------------------------|

| Name| Type | Size | Format |

|-----+------------+------------------------+-------------------------|

|words|SAMSON queue|36.9Kkvs in 2.54Mbytes|system.Value-system.Value |

\---------------------------------------------------------------------/

### 2.7.6 show\_local\_queue

The command 'show\_local\_queue' shows contents of a queue downloaded using pop. Modules should be installed locally

Synopsis:

delilah> show\_local\_queue <local\_dir> [-header] [-limit X]

### 2.7.7 push\_queue

To push the content of a queue onto another queue (or even various queues), the command 'push\_queue' is used.

Synopsis:

delilah> push\_queue <from\_queue> <to\_queue> [ <to\_queue2> ... ]

## 2.8 stream commands

The 'stream' category of delilah commands contain the commands that ...

### 2.8.1 ls\_stream\_operations

Show a list of defined stream operations (added with add\_stream\_operation)

Synopsis:

delilah> ls\_stream\_operations [-in] [-out] [-running] [-properties]

-in Information about data accepted in the operations

-out Information about data emitted

-running Currently running operations and status

-properties Show properties associated to each stream operation

### 2.8.2 add\_stream\_operation

To add a stream operation, the command 'add\_stream\_operation' is used. A stream operation is used to automatically process data from input queues to output queues. The stream operation will 'silently' work to process all data that comes in on the input-queues and push the results to the output queues. It will keep working until it is removed, using either of the 'rm\_stream\_operation' or 'cancel\_stream\_operation' commands.

The types of the queues (input or output) given as parameters to this command are defined by the operation itself. The operation knows how many input queues it is supposed to use and the queues will be read in order and assigned accordingly.

Synopsis:

delilah> add\_stream\_operation <name> <operation> <input-queues> <output-queues> [-forward]

-forward Option that allows to schedule reduce operations without state.

Joint against a constant queue

### 2.8.3 rm\_stream\_operation

To remove a stream operation, the command 'rm\_stream\_operation' is used.

Synopsis:

delilah> rm\_stream\_operation <name> [-f (silent)]

### 2.8.4 run

To manually start an operation, the command 'run' is used. This operation will keep running until it is either cancelled or removed, using the commands 'rm\_stream\_operation' or 'cancel\_stream\_operation'.

Synopsis:

delilah> run <op\_name> [queues...] [-clear\_inputs]

<op\_name> Name of the operation. See 'help ls\_operations' for more info

[queues] Name of the queues involved in this operation (inputs and outputs)

[-clear\_inputs] clear content from input queues before running this operation

### 2.8.5 cancel\_stream\_operation

To cancel a particular operation, the command 'cancel\_stream\_operation' is used.

Synopsis:

delilah> cancel\_stream\_operation <op\_id>

<op\_name> : Identifier of the operation. Usually something like XXXXX\_XXX

### 2.8.6 set\_stream\_operation\_property

The command 'set\_stream\_operation\_property' sets the value of an enviroment property associated to a stream operation (see add\_stream\_operation).

Synopsis:

delilah> set\_stream\_operation\_property <stream\_operation\_name> <variable\_name> <value>

### 2.8.7 unset\_stream\_operation\_property

The command 'unset\_stream\_operation\_property' removes an enviroment property associated to a stream operation (see add\_stream\_operation).

Synopsis:

delilah> unset\_stream\_operation\_property <stream\_operation\_name> <variable\_name>

### 2.8.8 add\_queue\_connection

To connect a queue to a set of queues, the command 'add\_queue\_connection' is used. All data from the source queue is automatically redirected to target queues.

Synopsis:

delilah> add\_queue\_connection <source\_queue> <target\_queue\_1> <target\_queue\_2> ... <target\_queue\_N>

### 2.8.9 rm\_queue\_connection

To remove a queue connection, previously established with 'add\_queue\_connection', the command 'rm\_queue\_connection' is used.

Synopsis:

delilah> rm\_queue\_connection <source\_queue> <target\_queue>

### 2.8.10 ls\_queues\_connections

To view a list of the queue connections defined with 'add\_queue\_connection', the command 'ls\_queues\_connections' is used.

Synopsis:

delilah> ls\_queues\_connections

### 2.8.11 ps\_stream

To obtain a list of the stream tasks currently running in all workers, the command 'ps\_stream' is used.

Synopsis:

delilah> ps\_stream

### 2.8.12 ls\_workers

The command 'ls\_workers' produces a list of current workers with current memory/ disk / process status.

Synopsis:

delilah> ls\_workers [-disk] [-engine]

[-disk] Show more info about disk activity in each worker

[-engine] Show more info about engine activity in each worker

### 2.8.13 init\_stream

The command 'init\_stream' executes an initialization script to setup some automatic stream operations.

Synopsis:

delilah> init\_stream [prefix] <script\_name>

[prefix] used to name operations and queues

<script\_name> name of the script (e.g. module.script)

### 2.8.14 ps\_workers

The command 'ps\_workers' is used to view a list of the commands under execution in each node of the cluster.

Synopsis:

delilah> ps\_workers <command pattern>

### 2.8.15 defrag

The command 'defrag' defragments the content of a particular queue

Synopsis:

delilah> defrag <queue> <destination\_queue>