

# Table of Contents

<b>GLite plugin description.....</b>	<b>1</b>
Cream checks.....	1
check_tom_fd.....	1
check_cream_cmd.....	1
check_cream_jobs.....	1
Batch System checks.....	2
check_lsf_jobs.....	2
check_pbs_jobs.....	2
Worker Node checks.....	3
check_wn_jobs.....	3
Generic checks.....	3
check_glite_host.....	3
Cream CLI checks.....	3
CreamCLI_Submit.....	3
WMS checks.....	4
check_wms_services.....	4
check_wms_queues.....	4
check_wms_jobs.....	5

# GLite plugin description

## Cream checks

The following checks are based on the output of a monitoring script installed by default on a CREAM CE. All these checks required root permission so instead of add user nagios to sudoers file we prefer to add a cron job which run the monitoring script and save the output on the file /tmp/cream.mon. This is the cron job which must be installed on the client before activating these plugins:

```
*/1 * * * * root /opt/glite/bin/glite_cream_load_monitor --show > /tmp/cream.mon
```

### check\_tom\_fd

This plugin counts the number of file descriptors used by tomcat5 process and generates an error if the number exceeds the thresholds specified.

Usage: ./check\_tom\_fd -w <fdnum> -c <fdnum>

Options:

```
-h                Print detailed help screen
-V                Print version information
-w <fdnum>       Set WARNING status if more than <fdnum> file descriptors (default 500)
-c <fdnum>       Set CRITICAL status if more than <fdnum> file descriptors (default 1000)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: tom_fd
#  output: TOM_FD OK: 107 tomcat5 file descriptors
#  perfddata: tom_fd=107;500;200;0
/perfddata:tom_fd=(\d+);(\d+);(\d+)/
and push @s, [ 'tom_fd',
               [ 'value', GAUGE, $1 ] ];
```

### check\_cream\_cmd

This plugin counts the number of pending commands inside CREAM database and generates an error if the number exceeds the thresholds specified.

Usage: ./check\_cream\_cmd -w <cmd> -c <cmd>

Options:

```
-h                Print detailed help screen
-V                Print version information
-w <cmd>          Set WARNING status if more than <cmd> pending commands (default 500)
-c <cmd>          Set CRITICAL status if more than <cmd> pending commands (default 1000)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: cream_cmd
#  output: CREAM_CMD OK: 0 pending commands in cream db
#  perfddata: cream_cmd= 0;1000;500;0
/perfddata:cream_cmd=(\d+);(\d+);(\d+)/
and push @s, [ 'cream_cmd',
               [ 'pending', GAUGE, $1 ] ];
```

### check\_cream\_jobs

This plugin counts the number of active jobs inside CREAM queue and generates an error if the number exceeds the thresholds specified.

Usage: ./check\_cream\_jobs -w <jobs> -c <jobs>

Options:

```
-h                Print detailed help screen
-V                Print version information
-w <jobs>         Set WARNING status if more than <jobs> active jobs commands (default 100)
-c <jobs>         Set CRITICAL status if more than <jobs> active jobs (default 5000)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: cream_jobs
#  output: CREAM_JOBS OK: 0 active jobs in cream queue
#  perfddata: cream_jobs= 0;1000;3000;0
/perfddata:cream_jobs=(\d+);(\d+);(\d+)/
and push @s, [ 'cream_jobs',
              [ 'active', GAUGE, $1 ]];
```

## Batch System checks

### check\_lsf\_jobs

This plugin counts the number of running and total jobs in lsf queue(s) and generates an error if the total number exceeds the thresholds specified.

Usage: ./check\_lsf\_jobs -w <jobs> -c <jobs> -q <queue>

Options:

```
-h                Print detailed help screen
-V                Print version information
-w <jobs>         Set WARNING status if more than <jobs> jobs are in lsf queue(s) (default 500)
-c <jobs>         Set CRITICAL status if more than <jobs> jobs are in lsf queue(s) (default 1000)
-q <queue>       Look for jobs in queue <queue>. (default sum on all queues)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: lsf_jobs
#  output: LSF_JOBS OK: 38 jobs in queue(s), 33 running
#  perfddata: lsf_jobs=38;33;500;1000;0
/perfddata:lsf_jobs=(\d+);(\d+);(\d+);(\d+)/
and push @s, [ 'lsf_jobs',
              [ 'tot', GAUGE, $1 ],
              [ 'run', GAUGE, $2 ]];
```

### check\_pbs\_jobs

This plugin counts the number of running and total jobs in pbs queue(s) and generates an error if the total number exceeds the thresholds specified.

Usage: ./check\_pbs\_jobs -w <jobs> -c <jobs> -q <queue>

Options:

```
-h                Print detailed help screen
-V                Print version information
-w <jobs>         Set WARNING status if more than <jobs> jobs are in pbs queue(s) (default 500)
-c <jobs>         Set CRITICAL status if more than <jobs> jobs are in pbs queue(s) (default 1000)
-q <queue>       Look for jobs in queue <queue>. (default sum on all queues)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: pbs_jobs
#  output: PBS_JOBS OK: 38 jobs in queue(s), 33 running
```

```
# perfdata: pbs_jobs=38;33;500;1000;0
/perfdata:pbs_jobs=(\d+);(\d+);(\d+);(\d+)/
and push @s, [ 'pbs_jobs',
               [ 'tot', GAUGE, $1 ],
               [ 'run', GAUGE, $2 ]];
```

## Worker Node checks

### check\_wn\_jobs

This plugin counts the number of running and total jobs in the worker node

Usage: ./check\_wn\_jobs -b <batchsystem>

Options:

```
-h                Print detailed help screen
-V                Print version information
-b <batchsystem> lsf || pbs [required]
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: wn_jobs
# output: WN_JOBS OK: 38 jobs in the wn, 33 running
# perfdata: wn_jobs=38;33;0
/perfdata:wn_jobs=(\d+);(\d+);(\d+)/
and push @s, [ 'wn_jobs',
               [ 'tot', GAUGE, $1 ],
               [ 'run', GAUGE, $2 ]];
```

## Generic checks

### check\_glite\_host

This plugin collects some gLite host informations

Usage: ./check\_glite\_host [-p] [-r]

Options:

```
-h                Print this help screen
-p                Print human readable informations
```

The informations collect with this plugin are used to create an introduction page for the host. This is an example of the command output:

```
CPU model: Intel(R) Xeon(R) CPU E5520 @ 2.27GHz
CPU number: 3
Mem Total: 2097152 kB
Kernel version: 2.6.18-194.17.1.el5xen x86_64
OS version: Scientific Linux SL release 5.5 (Boron)
gLite metapackages installed: glite-UI-version-3.2.8-0.s15
lcg metapackages installed: lcg-ManageVOTag-2.2.1-4 lcg-CA-1.37-1
```

## Cream CLI checks

### CreamCLI\_Submit

With this plugin we test direct submission to a CREAM CE. It can be used in two ways, you can give as input the Cream CE Endpoint using option "-c", or you can test all the registered queue of the given host. To do this

the plugin queries the given BDII to obtain all the endpoints host in the machine where the configured VO (default is *dteam*) is abilitate to submit. After submission the plugin checks periodically the job state until it finishes or a timeout is reached. It requires that user *nagios* can create voms proxy without password. It can use a configuration file (test.conf) which must be installed in the nagios home directory. An exampe of configuration file is given:

```
BDII="cert-bdii-04.cnaf.infn.it"
VO="dteam"
TIMEOUT=300
```

This is the help command:

This plugin uses cream CLI commands to test direct submission to CREAM CE. Success means that job's final state is DONE-OK. Failure means that the job doesn't finish correctly or timeout is reached.

This test can use a local configuration file (test.conf) and requires the possibility to create voms proxy without a password.

```
Usage: ./CreamCLI_Submit [-H <host> -b <bdii>] || [-c <cream>] -v <vo> -t <timeout>
```

Options:

```
-h          Print this help screen
-V          Print version information
-d          Print debug messages
-H <host>   Query the BDII to retrieve cream's endpoints associates to the <host>
-b <bdii>   BDII to query. If not specified it is token from configuration file.
-c <cream>  Endpoint of the CREAM ce.
-v <vo>     User vo
-t <timeout> Maximum waiting time (in seconds)
```

## WMS checks

### check\_wms\_services

This plugin check if all the services listed in /opt/glite/etc/gLiteservices are running

```
Usage: /usr/lib/nagios/plugins/gLite/check_wms_services [-h] [-V] [-d]
```

Options:

```
-h  Print detailed help screen
-V  Print version information
-d  Print debug messages
```

### check\_wms\_queues

This plugin counts the number of requests in all the WMS queues (wm, jc and ice) reading the output of the script /opt/glite/sbin/glite\_wms\_wmproxy\_load\_monitor. You can set a warning and or a critical threshold level.

```
Usage: /usr/lib/nagios/plugins/gLite/check_wms_queues [-h] [-V] [-d] [-w <reqs>] [-c <reqs>]
```

Options:

```
-h          Print detailed help screen
-V          Print version information
-d          Print debug messages
-w <reqs>   Set WARNING status if there are more <reqs> requests on a single queue (default 300)
-c <reqs>   Set CRITICAL status if there are more <reqs> requests on a single queue (default 500)
```

This is the rule used to create graph with nagiosgraph:

```
# Service type: wms_queues
# output: WMS_QUEUES OK: There are 0 requests in wm queue, 0 in jc queue and 0 in ice queue
# perfdata: wms_queues=0;0;0;300;500;0
/perfdata:wms_queues=(\d+);(\d+);(\d+);(\d+);(\d+);(\d+)/
and push @s, [ 'wms_queues',
               [ 'wm', GAUGE, $1 ],
               [ 'jc', GAUGE, $2 ],
               [ 'ice', GAUGE, $3]];
```

## check\_wms\_jobs

This plugin counts the number of the active jobs in the WMS, first the ones in condor queue and then the ones in ice queues. You can set a warning and or a critical total threshold level.

Usage: /usr/lib/nagios/plugins/gLite/check\_wms\_jobs [-h] [-V] [-d] [-w <reqs>] [-c <reqs>]

Options:

- h Print detailed help screen
- V Print version information
- d Print debug messages
- w <reqs> Set WARNING status if there are more <reqs> requests on a single queue (default 800)
- c <reqs> Set CRITICAL status if there are more <reqs> requests on a single queue (default 1200)

This is the rule used to create graph with nagiosgraph:

```
# Service type: wms_jobs
# output: WMS_JOBS OK: There are 76 jobs in condor queue and 0 jobs in ice queue
# perfdata: wms_jobs=76;0;800;1200;0
/perfdata:wms_jobs=(\d+);(\d+);(\d+);(\d+);(\d+)/
and push @s, [ 'wms_jobs',
               [ 'condor', GAUGE, $1 ],
               [ 'ice', GAUGE, $2 ]];
```

-- AlessioGianelle - 2011-01-12

---

This topic: EgeeJra1It > GlitePlugins

Topic revision: r7 - 2011-01-25 - AlessioGianelle



Copyright © 2008-2022 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback