

Table of Contents

LHCb Operational Procedures and Maintenance.....	1
Summary:.....	1
List of metrics and exceptions:.....	1
Metrics table.....	1
Exceptions table.....	2
Metric descriptions.....	3
DIRAC's Lemon Agent.....	3
Criticality definition:.....	3
Procedures:.....	3
Requirements:.....	3
Samclient monitoring.....	3
Requirements:.....	4
DIRAC certificate validity.....	4
Requirements:.....	4
DIRAC certificate mode and owner check.....	4
Requirements:.....	4
Procedures:.....	5
Maintaining of existing metrics:.....	5
Migrating to a different host.....	5
New metrics and exceptions.....	5
Removing.....	5
Disabling.....	6
Workarounding bugs/missing functionality.....	6
Convenient commands:.....	6

LHCb Operational Procedures and Maintenance

Summary:

We use Lemon to monitor critical services and machine states which proved to be point of failure in the past. There is only one metric for all DIRAC services and agents. Detailed description of the metric could be find in its own section.

The list of metrics and corresponding exceptions follows.

List of metrics and exceptions:

Metrics table

Metric ID	Metric name	Metric description	Metric class	VO	Services	Hosts
34	gridftp	Reused and modified metric that checks if globus-gridftp-server runs and if it runs under root with ppid 1	system.numberOfProcesses	LHCb	DIRAC	volhcb15
815	samclient_chklockfile	Reused and modified metric that checks if SAM tests submission is not stucked	samclient.checklockfile	LHCb	SamClient	volhcb05
4060	DIRAC_Cert_valid	Checks whether DIRAC certificate will remain valid next 14days	FIO::CertOK	LHCb	DIRAC	volhcb16
4061	DIRAC_Cert_key_perm	Checks whether DIRAC host.key has correct mode,uid,gid	file.info	LHCb	DIRAC	volhcb16
4090	opt_own_partition	Checks if /opt is on its own partition - parsing of /proc/mounts	log.Parse	LHCb	DIRAC	volhcb16
9104	partitionInfo	For this existing metric we added exception that checks if /opt is full (>XX%)	system.partitionInfo	LHCb	DIRAC	volhcb16
4076	DIRAC_Lemon_Agent_check	Checks whether log of DIRAC's Lemon Agent contains information about failure of critical services/agents	log.Parse	LHCb	DIRAC	volhcb12

This table is also part of the list of experiment-specific metrics for all experiments.

Exceptions table

Exception ID	Exception name	Corresponding metric(s)	Description	Hosts	
30063	gridftp_wrong	34	Raises an exception if gridftp is not properly running	volhcb15	http://service-cc-opm.web.ce
30668	samclient_lockfile_exception_lhcb	815	Raises an exception if there is logfile older than 2 hours in any samclient directory (check template for any changes of the threshold)	volhcb05	http://service-cc-opm.web.ce
30660	DIRAC_Cert_valid_err	4060	Raises an exception if the server certificate in DIRAC directory is going to expire in < 14days	all LHCb voboxes	http://service-cc-opm.web.ce
30661	DIRAC_Cert_key_perm_err	4061	Raises an exception if the DIRAC host.key don't have correct mode,uid,gid	all LHCb voboxes	http://service-cc-opm.web.ce
30690	opt_own_partition_err	4090	Raises an exception if the /opt is not on its own partition	all LHCb voboxes	http://service-cc-opm.web.ce
30689	opt_partition_full_err	9104	Raises an exception if /opt is more than 90% full (check template for any changes of the threshold)	all LHCb voboxes	http://service-cc-opm.web.ce
30676	DIRAC_Lemon_Agent_check_err	4076	Raises an exception if log of DIRAC's Lemon Agent contains information about failure of	all LHCb voboxes	NoNe ATM

Metric descriptions

Some of the used metrics deserve more comment to fully understand what is happening behind the scene.

DIRAC's Lemon Agent

DIRAC's Lemon Agent is the part of the DIRAC Framework systems that was written to enable Lemon monitoring of all DIRACs' services and agents. In this paragraph the *Lemon Agent* means part of the DIRAC framework, not the part of the Lemon monitoring system itself. This agent is written in a generic way --> there is only one Lemon Agent running on each host, which is same for all machines. The agent monitors all installed and setup services/agents on the machine. It gets the list from the local configuration (walking through the DIRAC's directory system). According to the criticality of the agent/service (see below), it outputs the status of the service/agent to a local log file (`/opt/dirac/runit/Framework/LemonAgent/log/current`) (which is automatically rotated by DIRAC). Then an independt TODO NAME Lemon metric regularly parses the log file and looks for failing critical services/agents.

This approach has many benefits, namely:

- only one generic agent for every host
- no need to define list of agents/services on hosts
- it gets installed automatically with DIRAC
- the log file is rotated for free
- no need to hassle with certificates

Criticality definition:

Every agent/service has defined its criticality. This criticality is defined per system (Production/Development) in Configuration System component. They can be redefined using local configuration file of the Lemon Agent (`/opt/dirac/etc/FrameworkSystem_LemonAgent.cfg`). There are only two levels of criticality right now: *Critical* and *NonCritical*. Lemon metric only checks failing *Critical* services/agents.

Procedures:

Given the fact that there is only one metric on one machines where may be running several services/agents, one has to login to the machine to see which service/agent is actually failing. This is true for operators in CCC, which are instructed to do so before contacting responsible persons. When the exception is first risen, the email containing this information is automatically sent from the machine before any alarm appears (the alarm is only risen after given number of consequentially failures).

Requirements:

Our own version of lemon-parse-log sensor which can handle UTC strings in it:

`/afs/cern.ch/user/j/jhorky/public/sl5/lemon-sensor-parse-log-1.1.2r1-1.x86_64.rpm`

Samclient monitoring

Monitoring of sam test submission system is done in the same way as on samXXX machines managed by SAM team. While the SAM is going to be retired, they do not accept any new VO on their machines --> we deployed the same sensor on our machine.

Sam test submission is monitored using samclient sensor, which must be installed by hand. There is only version for 32bit SL4, but the sensor is written in perl, so it should not matter which architecture it is deployed on (in fact, right now it is running on 64bit volhcb05 without problem). The lack of rpm in the sl5/64bit repository is the reason why it can't be installed using Quattor's directive.

Used samclient.lockfile metric class monitors age of all lockfiles inside .same directories under given parent directory (/home/santinel for volhcb05). It outputs its age and corresponding exception then checks whether it is older than set threshold. See templates linked from tables above.

Requirements:

The sensor must be installed be hand. It can be found here:

http://swrep.cern.ch/swrep/i386_slc4/lemon-sensor-samclient-1.0.0-7.noarch.rpm

DIRAC certificate validity

For some reason, more than one instance of FIO::CertOK on one host causes problems - if fetching information from both instance simultaneously (at the exact same time) one of the instance don't get updated. As workaround, we set the frequency of the two sensors to two distinct prime numbers around 5hours each, which means that the collision only occurs once per 10years.

Strange, but it works.

There is, however, still one unresolved thing. Running `lemon-host-check` will try to get fresh information from all metrics to see whether there is an exception or not. As this will cause the "collision" of the metrics and thus will result in output like this:

```
root@volhcb20 ~/ >/usr/sbin/lemon-host-check
[INFO] lemon-host-check version 1.3.3 started by root at Tue May 18 15:46:36 2010 on volhcb20.cern.ch
[VERB] 30660
[VERB]      Name:      exception.VOBOX_LHCb_DIRAC_Cert_valid_err (dirac-host-certificate)
[VERB]      Reason:    (null)
[VERB]      Notes:     possible false exception (cacheAll: enabled)
[VERB] Exceptions: 1 - Running actuators: 0 - Disabled exceptions: 0 - State: Production
```

The possible false exception here means that it wasn't able to retrieve status of corresponding metric. However, when **server** asks for data, he uses the same source of information as `lemon-cli` application and thus **does not suffer from this error**.

Requirements:

`="/system/monitoring/metric/_810/period" = 17761;`

DIRAC certificate mode and owner check

There is a bug in mainstream version of the lemon-sensor-file (it returns non-sense values for mode and also for group id of given file). Our own version must be installed. See requirements.

Requirements:

One must install our own version of the lemon-sensor-parse-log:

[/afs/cern.ch/user/j/jhorky/public/sl5/lemon-sensor-parse-log-1.1.2r1-1.x86_64.rpm](http://afs.cern.ch/user/j/jhorky/public/sl5/lemon-sensor-parse-log-1.1.2r1-1.x86_64.rpm)

Procedures:

Procedures for operators in CCC are maintained in the Operational Procedures Management Portal [↗](#). Existing procedures for the exceptions are listed in the table above, but it can be always found by searching the portal for the exception name (e.g. try to search for `gridftp_wrong`, you will get three pages - but only one of them is dedicated to VOBOXes). Everybody can try to change existing procedure, but it is a validator of the page who decides whether the changes will take effect or not.

It is always possible to create a new procedure page. In that case, it gets checked by a chosen validator and also by somebody from CC operational team.

Maintaining of existing metrics:

It is highly recommended to consult the dedicated tutorial page. There is also general official documentation [↗](#) for Lemon system and for some sensors [↗](#) too. Please note that the official documentation may not be fully up to date.

Here we list specific information concerning our own metrics.

Migrating to a different host

First of all check if it is not already enabled on the host, using right command. If it is not, find the template from which is given monitoring included. Right now, it is done in the root template of the machine (e.g. http://tpl-viewer.cern.ch/cdb-tpl-view/tpl_view.php?profile=profiles/profile_volhcb15 [↗](#)), but it may change (dirac common monitoring will probably be moved to one of the dirac templates). Be sure to read every comments concerning the metric, some of them require proper handling. Then, using a `cdbop` command (see tutorial), edit a target machine's template and include the required lines. Also comment out or delete corresponding lines in the previous machine template. Update and commit.

Do not forget to update the list of metrics/exceptions on this page and also on page for all experiments.

You should also connect to the machine and make sure, that the changes you have done are properly propagated (see this and this).

New metrics and exceptions

Right now, we have dedicated range of metric IDs (4060-4090) and expction IDs (30660-30690). Half of them are already used, mainly by non-active `VOBOX_LHCB_DIRAC_[service]_log_[time,content]_check` metrics, which we decided to abandon. These could be reused in future after contacting lemon support (lemon.support@cernNOSPAMPLEASE.ch) and deleting the templates (`/prod/customization/lhcb/vobox/pro_monitoring_metrics_dirac_[service]`).

Every time a new metric/exception is created, one has to contact lemon support, so they can update metric list (otherwise, it will not be listed on the web).

Removing

Is very similar to migrating. Just don't add the lines to the another machine's profile. Once again, do not forget to update the lists (see above).

Disabling

It can be done in two different ways. First one is to comment out corresponding include directives in a given machine's profile template (in case it is included from within the profile template and you want to disable it only for a given host). In case it is included from elsewhere (e.g. in some common DIRAC template, which is used by every vobox), you should NOT disable it in the template itself (unless you want to disable it globally). The correct way to do this is to add similar lines to machine's profile file:

```
"/system/monitoring/metric/_27/active" = false;
"/system/monitoring/exception/_30054/active" = false;
```

Make sure you specify these options AFTER the metric is included (otherwise these options will get overwritten).

Once again, do not forget to update the lists (see above).

Workarounding bugs/missing functionality

Some of the metrics require installing of our own version of sensors. These versions fixes some bugfixes and also provides some additional functionality that is listed below. When installing RPMs by hand, it is also needed to instruct quattor to not replace them, using following directives in the templates:

```
"/software/components/spma/userpkgs" = "yes";
"/software/components/spma/userprio" = "yes";
```

File sensor:

- Uid instead of gid returned: <https://savannah.cern.ch/bugs/?57289>
- Incorrect file mode: <https://savannah.cern.ch/bugs/?56806>

Log.Parse sensor

- Missing UTC in dformat string: https://savannah.cern.ch/bugs/?func=detailitem&item_id=55080

FIO::CertOK

- Race condition in the metric class * <http://savannah.cern.ch/bugs/?67629>
- no fix available

The RPMs are available at `/afs/cern.ch/user/j/jhorky/public/sl{4,5}/`

Convenient commands:

Checking settings of all metrics and exceptions on the host:

```
ncm-query --pan --dump /system/monitoring
```

```
-- JiriHorky - 17-May-2010
```

This topic: LCG > LhcbOperationalProceduresAndMaintainance

Topic revision: r10 - 2010-05-20 - JiriHorky



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

LhcbOperationalProceduresAndMaintainance < LCG < TWiki

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback