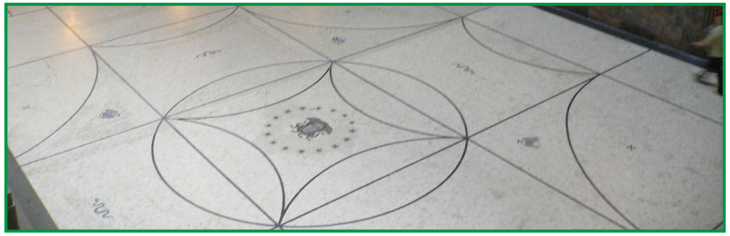
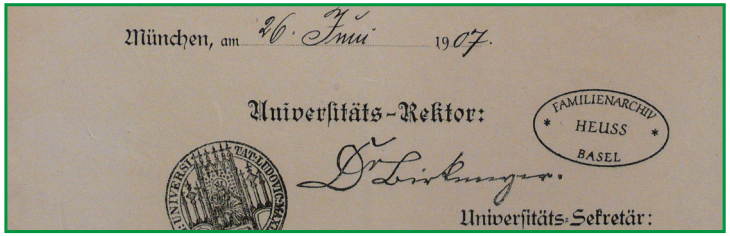
PROMETHEUS-DCACHE-EXPORTER

Christoph Anton Mitterer
mitterer@lmu.de

I. Introduction

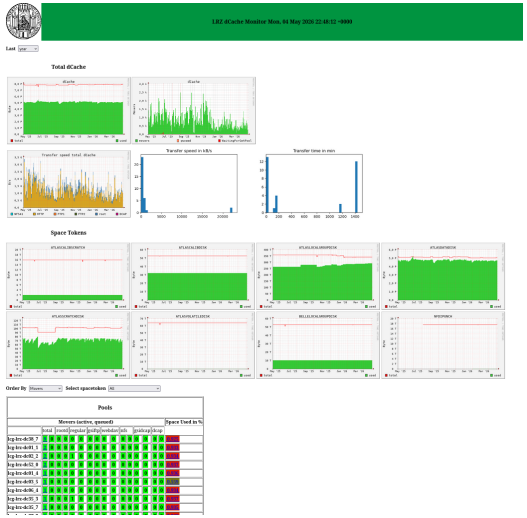




A Brief History of Monitoring/Alerting at LMU

- Since 2008 we've had used Lemon and later Ganglia ("between the devil and the deep blue sea") as well as Nagios followed by Icinga.
  Cons: quite old-fashioned, in particular the former two basically not extensible

  Special monitoring tool ("`dcache-mon`") by a bachelor student.
  Cons: "Not programmed as I'd do it.", a mix of PHP, Python2, Shell plus some 3rd party PHP templating library of questionable licence which was extremely painful to upgrade to recent PHP versions, also, stateful where it shouldn't be (even adding new tokens or queues, required changes to some undocumented SQLite database, no means of querying/combining metrics or alerting.



- Maintenance and keeping these security-supported became too much of an effort.
  Wanted only one system for both monitoring and alerting.
- Several systems were examined (in particular Prometheus, Checkmk and Netdata).
- Prometheus seemed to be the most powerful one.

Prometheus

- Apache Licence 2.0, written in Go, originally (≈2012) from SoundCloud now CNCF.
- Provides the back-end:
  Collection of metrics (usually via push), TimeSeriesDB, powerful query language ("PromQL") and a powerful alerting system (`alertmanager`).
- Front-end:
  Done by others (often Grafana).
- On top:
  Thanos (HA, merging, down-sampling, long term storage, etc.), Mirmir, et cetera.
- *Quite a beast, even more so with Thanos.*
- Exporters: collect the metrics, many available (see [0] and [1])
- Metrics (counter, gauge, histogram, summary), time series, labels

```
dcache_space_reservation_info{description="ATLASDATADISK",id="110000", instance="lcg-lrz-dcache0.grid.lrz.de"} 1
dcache_space_reservation_info{description="ATLASLOCALGROUPDISK", id="380987", instance="lcg-lrz-dcache0.grid.lrz.de"} 1
dcache_space_reservation_space_used_bytes{id="110000", instance="lcg-lrz-dcache0.grid.lrz.de"} 4748854393230490
dcache_space_reservation_space_used_bytes{id="380987", instance="lcg-lrz-dcache0.grid.lrz.de"} 292705998850930
```

- If a label "changes" ⇒ new time series
- PromQL: `dcache_space_reservation_space_used_bytes * on(id) group_left(description) dcache_space_reservation_info`

```
{description="ATLASDATADISK", id="110000", instance="lcg-lrz-dcache0.grid.lrz.de"} 4748869032235006
{description="ATLASLOCALGROUPDISK", id="380987", instance="lcg-lrz-dcache0.grid.lrz.de"} 292706065309706
```

II. Overview

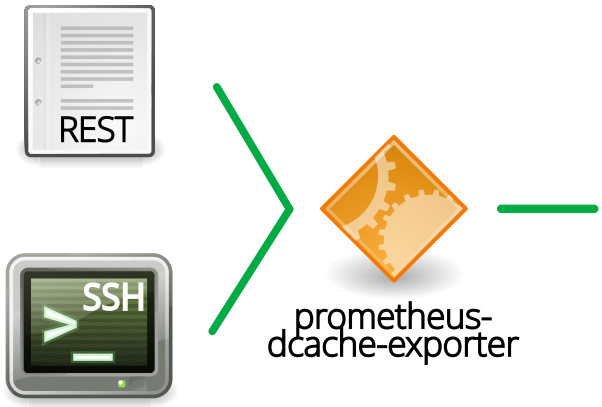
Basics

- Written in Python (≧ 3.11 or maybe even 3.13).
- Depends on `prometheus_client`, `httpx`, `paramiko`, `jsonpath`, `certifi`, `humanfriendly`[0] and `module_utils`.
- dCache (≧ 11.1) already includes a Prometheus exporter?!
  - At least currently, it exports "only" Java VM related data.
  - Last known upstream statement: no general monitoring planned
  - dCache integrated exporter is per (dCache-)domain, thus n domains require n open ports, each to be "scraped" by Prometheus.
    Doable, of course, but not too admin-friendly.
- Design goals:
  - generic *(it's not named "`prometheus-lmu-dcache-exporter`")* and thus useful to other sites (includes for example that different authentication methods for REST/SSH are supported)
  - fast *(scraping at LMU done every 10s)*
  - resilient (to errors from the input data, at least to some extent)
  - configurable (which metrics to output)

[0] Until dCache issue #8104 is resolved.

Data Processing Model

- Gets data from dCache's REST interface (`frontend`-service), which is JSON and easy to parse, and SSH interface (`admin`-service), which is less easy to parse.
  The latter, because not all data is *(yet?)* available in REST.
- Produces Prometheus' text metric exposition format out (which is closely related to the "OpenMetrics"-standard.




```
# HELP dcache_cell_info Information about a cell.
# TYPE dcache_cell_info gauge
dcache_cell_info{cell="PoolManager",domain="storagesystem0",version="11.2.4"} 1.0
dcache_cell_info{cell="SpaceManager",domain="storagesystem0",version="11.2.4"} 1.0
…
# HELP dcache_space_reservation_info Information about a space reservation.
# TYPE dcache_space_reservation_info gauge
dcache_space_reservation_info{description="ATLASDATADISK",id="110000"} 1.0
dcache_space_reservation_info{description="ATLASLOCALGROUPDISK",id="380987"} 1.0
…
# HELP dcache_space_reservation_space_total_bytes The total space of a space reservation in bytes.
# TYPE dcache_space_reservation_space_total_bytes gauge
dcache_space_reservation_space_total_bytes{id="110000"} 5.047987449515715e+015
dcache_space_reservation_space_used_bytes{id="110000"} 4.601484103193072e+015
…
# HELP dcache_pool_mover_active The number of active movers on the pool.
# TYPE dcache_pool_mover_active gauge
dcache_pool_mover_active{name="lcg-lrz-dc01_1",queue="nfs"} 21.0
dcache_pool_mover_active{name="lcg-lrz-dc01_1",queue="xrootd"} 0.0
…
```

Invocation

- Either as “true” exporter (`http`-export-mode) or dumping to `stdout` or (atomically) to `file`. (used with `systemd` timer and node textfile collector).
- For example: `prometheus-dcache-exporter --dcache-frontend-rest.base-url http://localhost:3880/api --dcache-admin-ssh.user admin`
- Documentation: 🤣 → `--help`-option

```
$ prometheus_dcache_exporter --help
Usage: __main__.py [-h] [--dcache-frontend-rest.base-url URL] [--dcache-frontend-rest.http-auth-type {none,basic_auth}] [--dcache-frontend-rest.http-auth-user USERNAME]
                   [--dcache-frontend-rest.http-auth-passphrase STRING] [--dcache-frontend-rest.allow-http-auth-without-tls] [--dcache-frontend-rest.tls-ca-certificates PATHNAME]
                   [--dcache-frontend-rest.tls-client-auth-certificate PATHNAME] [--dcache-frontend-rest.tls-client-auth-private-key PATHNAME]
                   [--dcache-frontend-rest.tls-client-auth-private-key-passphrase STRING] [--dcache-frontend-rest.connect-timeout DECIMAL] [--dcache-frontend-rest.read-timeout DECIMAL]
                   [--dcache-frontend-rest.pool-timeout DECIMAL] [--dcache-frontend-rest.connection-pool-max-size INTEGER] [--dcache-frontend-rest.connection-pool-max-idle-size INTEGER]
                   [--dcache-frontend-rest.connection-pool-max-idle-time DECIMAL] [--dcache-frontend-rest.max-concurrent-requests POSITIVE-INTEGER]
                   [--dcache-admin-ssh.host HOSTNAME-OR-ADDRESS] [--dcache-admin-ssh.port PORTNUMBER] [--dcache-admin-ssh.user USERNAME] [--dcache-admin-ssh.known-hosts PATHNAME]
                   [--dcache-admin-ssh.private-key PATHNAME] [--dcache-admin-ssh.private-key-passphrase STRING] [--dcache-admin-ssh.passphrase STRING]
                   [--dcache-admin-ssh.no-system-known-hosts] [--dcache-admin-ssh.no-default-private-key-locations] [--dcache-admin-ssh.no-agent] [--dcache-admin-ssh.tcp-timeout DECIMAL]
                   [--dcache-admin-ssh.banner-timeout DECIMAL] [--dcache-admin-ssh.authentication-timeout DECIMAL] [--dcache-admin-ssh.channel-open-timeout DECIMAL]
                   [--dcache-admin-ssh.command-channel-timeout DECIMAL] [--dcache-admin-ssh.keepalive-interval INTEGER] [--dcache-admin-ssh.max-concurrent-commands POSITIVE-INTEGER]
                   [--export-mode {http,stdout,file}] [--http-port PORTNUMBER] [--output-file PATHNAME] [--log.level {CRITICAL,50,FATAL,50,ERROR,40,WARN,30,WARNING,30,INFO,20,DEBUG,10}]

Options:
  -h, --help            show this help message and exit
  --dcache-frontend-rest.base-url URL
                        The base URL under which the REST interface of dCache's `frontend` service can be reached. Defaults to `http://localhost:3880/api/` if this option isn't given.
  --dcache-frontend-rest.http-auth-type {none,basic_auth}
                        The type of HTTP authentication to be performed. `none` (which is the default if this option isn't given) causes no HTTP authentication to be performed.
                        `basic_auth` causes basic authentication to be used and requires the `--dcache-frontend-rest.http-auth-user`- and
                        `--dcache-frontend-rest.http-auth-passphrase`-options to be given.
  --dcache-frontend-rest.http-auth-user USERNAME
                        The user used for HTTP authentication when connecting to the REST interface of dCache's `frontend` service.
  --dcache-frontend-rest.http-auth-passphrase STRING
                        The passphrase used for HTTP authentication when connecting to the REST interface of dCache's `frontend` service.
  --dcache-frontend-rest.allow-http-auth-without-tls
                        Allow HTTP authentication when connecting to the REST interface of dCache's `frontend` service without TLS.
  --dcache-frontend-rest.tls-ca-certificates PATHNAME
                        The CA certificates that are trusted when connecting to the REST interface of dCache's `frontend` service via TLS. If this option isn't given, the system's default
                        CA certificates are used. If a regular file (or symbolic link to such), it must consist of concatenated CA certificates in PEM format; if a directory (or symbolic
                        link to such), it must follow the well-known layout by OpenSSL. CRLs are ignored and OCSP isn't performed.
  --dcache-frontend-rest.tls-client-auth-certificate PATHNAME
                        The certificate and, unless the `--dcache-frontend-rest.tls-client-auth-private-key`-option is given, private key used for client authentication when connecting to
                        the REST interface of dCache's `frontend` service via TLS. It must be a regular file (or symbolic link to such) that consists of the certificate, depending on the
                        aforementioned, concatenated with the private key in PEM format.
  --dcache-frontend-rest.tls-client-auth-private-key PATHNAME
                        The private key used for client authentication when connecting to the REST interface of dCache's `frontend` service via TLS. It must be a regular file (or symbolic
                        link to such) that consists of the private key in PEM format.
  --dcache-frontend-rest.tls-client-auth-private-key-passphrase STRING
                        The passphrase used for decrypting the private key specified via the `--dcache-frontend-rest.tls-client-auth-certificate`- or
                        `--dcache-frontend-rest.tls-client-auth-private-key`-options.
  --dcache-frontend-rest.connect-timeout DECIMAL
                        The timeout in seconds for establishing the connection to the REST interface of dCache's `frontend` service, with non-positive decimals disabling it. Defaults to
                        `1.0` if this option isn't given.
  --dcache-frontend-rest.read-timeout DECIMAL
                        The timeout in seconds for reading a chunk of data from the REST interface of dCache's `frontend` service, with non-positive decimals disabling it. Defaults to
                        `1.0` if this option isn't given.
  --dcache-frontend-rest.pool-timeout DECIMAL
                        The timeout in seconds for getting a connection from the pool of connections to the REST interface of dCache's `frontend` service, with non-positive decimals
                        disabling it. Defaults to `1.0` if this option isn't given.
  --dcache-frontend-rest.connection-pool-max-size INTEGER
                        The maximum number of connections to the REST interface of dCache's `frontend` service in the connection pool, with non-positive decimals meaning unlimited.
                        Defaults to the value of the `--dcache-frontend-rest.max-concurrent-requests`-option if this option isn't given. The maximum number of requests that are actually
                        made concurrently is specified via the `--dcache-frontend-rest.max-concurrent-requests`-option.
  --dcache-frontend-rest.connection-pool-max-idle-size INTEGER
                        The maximum number of idle connections to the REST interface of dCache's `frontend` service in the connection pool, with negative decimals meaning unlimited and
                        zero meaning none. Defaults to the value of the `--dcache-frontend-rest.connection-pool-max-size`-option if this option isn't given.
  --dcache-frontend-rest.connection-pool-max-idle-time DECIMAL
                        The maximum time in seconds for which an idle connection to the REST interface of dCache's `frontend` service may be kept open in the connection pool, with negative
                        decimals meaning unlimited and zero meaning instant closure. Defaults to `305.0` if this option isn't given.
  --dcache-frontend-rest.max-concurrent-requests POSITIVE-INTEGER
                        The maximum number of requests that are concurrently made to the REST interface of dCache's `frontend` service. Defaults to `256` if this option isn't given.
  --dcache-admin-ssh.host HOSTNAME-OR-ADDRESS
                        The host under which the SSH interface of dCache's `admin` service can be reached. Defaults to `localhost` if this option isn't given.
  --dcache-admin-ssh.port PORTNUMBER
                        The port under which the SSH interface of dCache's `admin` service can be reached. Defaults to `22224` if this option isn't given.
  --dcache-admin-ssh.user USERNAME
                        The user used for connecting to the SSH interface of dCache's `admin` service.
  --dcache-admin-ssh.known-hosts PATHNAME
                        Adds the known hosts file (using OpenSSH's `ssh_known_hosts` format) to the list of those from which trusted SSH host keys are read. The list is processed after
                        that of any existing system known hosts files (`/etc/ssh/ssh_known_hosts`, `/etc/ssh/ssh_known_hosts2`, `~/.ssh/known_hosts` and `~/.ssh/known_hosts2`) and in the
                        order as given, with all but the first entry for a given being ignored.
  --dcache-admin-ssh.private-key PATHNAME
                        The file with the private key used for connecting to the SSH interface of dCache's `admin` service via the authentication method `publickey`.
  --dcache-admin-ssh.private-key-passphrase STRING
                        The passphrase used for decrypting the private key specified via the `--dcache-admin-ssh.private-key`-option.
  --dcache-admin-ssh.passphrase STRING
                        The passphrase used for connecting to the SSH interface of dCache's `admin` service via the authentication methods `password` and `keyboard-interactive`.
  --dcache-admin-ssh.no-system-known-hosts
                        Don't add any system known hosts files (`/etc/ssh/ssh_known_hosts`, `/etc/ssh/ssh_known_hosts2`, `~/.ssh/known_hosts` and `~/.ssh/known_hosts2`) to the list of
                        those from which trusted SSH host keys are read.
  --dcache-admin-ssh.no-default-private-key-locations
                        Don't use any private keys from default locations (which are the default of OpenSSH's `IdentityFile`-option).
  --dcache-admin-ssh.no-agent
                        Don't use any SSH agent.
  --dcache-admin-ssh.tcp-timeout DECIMAL
                        The timeout in seconds for establishing the TCP connection to the SSH interface of dCache's `admin` service, with non-positive decimals disabling it. Defaults to
                        `1.0` if this option isn't given.
  --dcache-admin-ssh.banner-timeout DECIMAL
                        The timeout in seconds for receiving any banner from the SSH interface of dCache's `admin` service, with non-positive decimals disabling it. Defaults to `1.0` if
                        this option isn't given.
  --dcache-admin-ssh.authentication-timeout DECIMAL
                        The timeout in seconds for authenticating to the SSH interface of dCache's `admin` service, with non-positive decimals disabling it. Defaults to `1.0` if this
                        option isn't given.
  --dcache-admin-ssh.channel-open-timeout DECIMAL
                        The timeout in seconds for opening a (SSH) channel to the SSH interface of dCache's `admin` service, with non-positive decimals disabling it. Defaults to `1.0` if
                        this option isn't given.
  --dcache-admin-ssh.command-channel-timeout DECIMAL
                        The timeout in seconds for read/write operations on the (SSH) channel of an executed command in the SSH interface of dCache's `admin` service, with non-positive
                        decimals disabling it. Defaults to `1.0` if this option isn't given.
  --dcache-admin-ssh.keepalive-interval INTEGER
                        The interval in seconds at which (client-initiated) SSH keepalive packets are sent to the SSH interface of dCache's `admin` service, with non-positive integers
                        disabling it. Defaults to `0` if this option isn't given.
  --dcache-admin-ssh.max-concurrent-commands POSITIVE-INTEGER
                        The maximum number of commands that are concurrently executed via the SSH interface of dCache's `admin` service. Defaults to `256` if this option isn't given.
  --export-mode {http,stdout,file}
                        The mode in which the metrics are exported. `http` (which is the default if this option isn't given) provides them via a HTTP server (that runs until the program is
                        terminated by a signal or an error), with metrics being (freshly) collected on every scrape. `stdout` writes them to standard output (once, after which the program
                        terminates). `file` writes them atomically (via moving a newly created temporary file) to the file specified via the `--output-file`-option (once, after which the
                        program terminates), with the output file being overwritten if it exists already.
  --http-port PORTNUMBER
                        The port on which the HTTP server listens when the export mode is `http` (ignored otherwise). Defaults to `9875` if this option isn't given.
  --output-file PATHNAME
                        The file to which the metrics are written when the export mode is `file` (ignored otherwise).
  --log.level {CRITICAL,50,FATAL,50,ERROR,40,WARN,30,WARNING,30,INFO,20,DEBUG,10}
                        The minimum level (name or number) of log messages to be logged. Defaults to `WARNING` (`30`) if this option isn't given.
```

Example Dashboard

Active movers: overall (all queues), per pool (all queues):

III. Issues, Wish list, Questions and Status

Issues I

- Multi-threading (done for REST and SSH):
  - Why
    - Ideally all pieces of raw data would be retrieved atomically (and thus in sync).
    - Getting (dCache) domain ↦ host mapping.
      Doing `get hostname` synchronously in every `System@`*`domain`* is painfully slow.
      `\s System@`*`domain1`*`,System@`*`domain2`*`,...` `get hostname` has its own issues.
      See dCache issue #8046.
  - "But CPython can't do multi-threading" – This is not CPU bound, so good. Plus the GIL is about to go away (see PEP 703).
  - Problems: Logging too noisy. Also, things break with too many threads (≈ 100).
    Unclear whether its the client (`paramiko` or `httpx`) or server (dCache).
    For SSH easy to fix (limit concurrency).
    For REST, strange effects. Getting `HTTP 429 (Too Many Requests)` despite insanely high thresholds set in `frontend`. In general connections are tried to be re-used. Disabling ironically seems to help (server issue?).
- Naming of metrics. Sounds simple, but names should be "good" and never change or people's time series would "break".

Issues II

- Resilience
  That is: How to handle errors during retrieval of the raw data (from REST and SSH)?
  - Simply abort the current metric collection / scrape?
  - In case of partial failure (for example if only SSH failed, but REST worked), try to deliver as much as possible?
    If so, how to indicate in Prometheus which data is valid (see Prometheus documentation issue #2977)?
  - Assuming that REST output (like JSON structure/names and types are always proper? But see for example dCache issue #8104
- How far to go with configurability?
  For example there's the `psu ls pool -l` data, but strictly speaking per-`PoolManager`. Default will be export for each, but in practise probably mostly useless, so there will be options to specifically select the desired `PoolManager`-instances.
- (Prometheus) `instance`-label (where the exporter runs) vs. hostnames (of domains): Some metrics are not related to a specific host, others are. Extra `host`-label? Swappable with `instance`?

Wish List

- Getting `/pool/*/usage/` as one resource. Having to request it for every pool is unnecessarily problematic.
- A new `/domains`, which contains at least the domain-host-mappings and perhaps other useful data (Java version and options?)
- What the `admin`-service's `psu ls pool -l` command gives but is not yet given by REST.
- More data in REST, in particular transfer totals (per pool, queue and/or protocol+flavour, maybe also totals per door (control data), maybe also split between data that was proxied via a door or not).
  (Prometheus uses time series of totals and calculates rates from these.)
  See dCache issue #8009.
- It might be nice to record the type of cell (for example to select `PoolManager`s).
  While there is `/cells[cellClass]`, only some of have useful values like `Pool`, `PoolManager` or `WebDAVDoor`.
  Most have however `org.dcache.cells.UniversalSpringCell`.

Questions

- REST data largely undocumented and often exact meanings are unclear.
  Examples:
  `/pool/{name}/usage/poolData/detailsData/{errorCode,arrorMessage}`
  Having something that indicates "pool is healthy & connected" would be nice.

  `/pool/{name}/usage/poolData/detailsData/costData/...`
  There's the static queues (like `store`, and `p2p`) but also the custom ones beneath `extendedMoverHash`. Can custom queue names collide with static ones?
- Which Prometheus metric "namespace" shall be used?
  Currently `dcache_*`, which is probably also used by dCache's own exporter.
  Not a big problem though, as one would typically run both exporters in different jobs, resulting in different `job`-labels.
- Assuming people would find this exporter useful, once it's released and people can see which metrics are already included, which others would they want to see?

Status

- Development took a bit longer than expected, because of …
  … oh hey look, there's an ~~Ikran~~ bird outside the window.

- Somewhere between alpha and beta, code not yet published until some remaining points are implemented/solved and more metrics exported.
- Developed/tested with dCache 11.2.
- Live test version running against LMU's dCache *(mostly to show at least some graphs here)*.
- Proper release likely in a few weeks, including Debian packaging and `systemd` units.

Finis coronat opus.