Introducing the dCache info service

Paul Millar <paul.millar@desy.de>
Who are you, listening to this?

• You are one of:
  – a member of the dCache collaboration,
  – a Tier-A integration collaborator (hello Tanja!),
  – a Tier-1 site-admin,
  – a valued college from CERN (hello Flavia!).

• What you will understand after listening:
  – An understanding of how the info service works.
  – Some ideas about how the data may be accessed and used.
Overview of talk

- The goals and non-goals of the info service.
- A medium-to-high level view of how it works.
- An overview of the provided information.
- How to obtain “live” information.
- Summary.
What is the info service?

- A robust, best-effort, “one-stop shop” overview of the current status of a dCache instance for external consumption.
  - **Robust**: the info service will continue to work independently of the rest of dCache.
  - **Best-effort**: there *may* be delays in information being updated (1 minute order-of-magnitude).
  - **One-stop shop**: you should be able to get all the information you require.

- It decouples updates from queries:
  - Querying is fast and robust
What the info service is not...

- Not used by other dCache components [*]
  - As this would:
    - break dCache's distributed architecture and introduce a (new) single point-of-failure,
    - increase latency in information propagation,
- Doesn't provide historic data:
  - Duplicating what sites (almost certainly) already have:
    - Ganglia, Nagios, LEMON, Mondin, Zabbix, ZenOSS, ...
- It is not an info-provider:
  - although the new WLCG GLUE info-provider is a customer of the info service.

[*] this is a small white lie: more details in the next slide
How it works.

Introducing the dCache info service
How updates are scheduled

- Data-Gathering Activities (DGAs) schedule fresh data requests.
  - DGAs give strong control over how often messages are sent.
  - We're careful not to over-burden components.
- Reply messages are processed independently.
  - Info cell is robust against messages being lost or the replies delivered out-of-order.
  - Allows for asynchronous updates: much faster updates, low bandwidth overhead (W.I.P.)
SIPs

- Derived data is the set of metrics calculated from other data within the state tree.
  - For example, `space.total` metric of a poolgroup is the sum of all `space.total` metrics of member pools.
- A Secondary Information Provider (SIP) calculates derived data.
  - Only triggered when *important* metrics change value.
- Uses:
  - Calculate aggregated space statistics (current)
  - High-level internal health checks (under investigation)
Storage in a tree

• Most metrics are held for finite time.
  – When time expires, they are automatically flushed.
  – Robust against stale information.
• A serialiser generates representations of state.
  – Adding new output formats is easy.
• Tree storage is abstracted.
  – Currently only uses memory, so serialisation is fast.
  – Structure is not hard-coded in the storage.
    • Storing new branches and/or metrics is easy.
What information is provided?

- Shallow structure with cross-links.
  - Normalised data: metrics appear only once.
- The following top-level branches exist:
  - summary: various aggregated information,
  - All other top-level branches are lists of items (as branches) with item-specific information below.
  - These lists include: pools, poolgroups, links, linkgroups, domains, doors, reservations.
- There's too much information to detail here.
Tiny fragments of state tree
How do I start the info service?

- Configure dCache so it's started on a node.
- ... or start it manually:
  - /opt/d-cache/bin/dcache info start
  - (optionally) have a cup of coffee (2—3 minutes) whilst initial set of data is populated.
- Start querying the information.
- No configuration needed for the info service [*].

[*] The GLUE info-provider does require careful configuration; but the info-provide is a separate, distinct component from the info service.
Accessing the information

• Via the admin interface
  – Commands for navigating state, like: cd, ls, pwd.
  – Choice of output format
• Via XML Conduit
  – A TCP connection get complete state as XML.
• Web front-end
  – Preferred method, but requires the httpd cell.
  – For example:
    • http://dcache.example.org:2288/info
    • http://dcache.example.org:2288/info/pools
Summary

• Info service provides a best-effort overview of a dCache instance.

• Maintaining state and divulging information are decoupled:
  – fast, robust.

• Supports some advanced features:
  – Derived data (re-)calculated as state changes.
  – Multiple output formats and transports.

• If additional metrics, data formats or transports are needed, they can be added.