dCache.org

Quality of Service and Data-Lifecycle for Big Data

Paul Millar
on behalf of the dCache team, INDIGO-DataCloud, ...

CluStor Workshop, DKRZ Hamburg, 2015-07-31

Fermilab NDGF Nordic DataGrid Facility DESY HELMHOLTZ ASSOCIATION LSDMA

Cheat sheet

dCache – open-source software for combining heterogeneous storage into a POSIX file-system for scientific data.

INDIGO-DataCloud – project to update cloud software for a European cloud infrastructure

RDA **(Research Data Alliance)** – organisation to help enable data sharing.

Big Data → Big Headaches

- Lots of data
- Limited resources → Data needs managing
- Which data is the most important?

  All of it!
- OK, which of the data …

  needs fast access,

  needs to be kept for references,

  can be deleted?

Introducing two concepts: QoS and DL

Quality-of-Service (**QoS**) is how the data should be handled right now; may be subject of an MoU or SLA.

Data-Lifecycle (**DL**) is how the data changes over time

Examples of Quality-of-Service

- **Availability** (of data):

  Store multiple copies / erasure encoding; store copy on tape; store on multiple tapes with different tape-libraries in different buildings without common power-supply

- **Reliability** (of service)

  Guarantee ability to accept certain amount of data over a certain time.

- **Integrity**:

  Periodically verify data against checksum, apply checksum as data is read

- **Latency**:

  Store on SSDs, spinning disks, tape

- **Bandwidth**:

  Guarantee ability to accept data at given rate; guarantee ability to deliver data at given rate.

Examples of Data-Lifecycle

- Changes to **QoS**:

  "Store data on SSDs, but move to magnetic disk one week after data ingest"

- Changes to **authz**:

  "Data is private for an embargo period, after which it becomes public."

- Changes to **"availability"**:

  "Delete old data if not cited in any paper";
  "Move old data into an Archive"

Why interest in QoS/DL?

- When dealing with Big Data, there's going to be a lot of data to manage.
- EU, through H2020, forces projects to think about it up-front:

  The "**Data Management Plan**".
- Broader in scope that QoS and DL, but encompasses those ideas.

Make DMP easier (or possible)

- Automate, automate, automate...

  Tell infrastructure what QoS and DL is needed:

- How to do this?
  - Need a **language** (vocabulary + grammar) to describe QoS and DL
  - Need a network **protocol** to allow interaction
  - Need **software** (client and server) that implements the protocol
  - Need **instances** (endpoints) that work with that new interface

The language of quality

- Need to know the **vocabulary** (what words mean) and **grammar** (how to describe what you want)
- In part, this already exists (e.g., SRM, CDMI)
  - Bespoke, limited, non-extensible, not always useful.
- Investigate via an **RDA Working-Group**
  - BoF meeting at RDA Plenary 6 in Paris 2015-09-25,
  - Start work on building this language,
  - Needs to be realistic: what can technology provide?

Network protocol

- Current plan: **CDMI**
  - ISO standard for cloud storage management,
  - Already has some QoS support,
  - Already supports arbitrary metadata,
- Likely involve coming up with a CDMI **extension**
  - SNIA seem keen, should learn more in SNIA Developers Conference (September)
- Working in **parallel** with RDA group.

Software to support interface

- Work already **underway** in providing CDMI support – both in dCache and more generally in INDIGO-DataCloud.

  Plan to support GPFS, HPSS and dCache.

- See the interface as **translating** QoS statements into policies in the underlying storage system:

  GPFS policies, HSM data-migration policies, dCache policies, etc.

Building a testbed

- INDIGO-DataCloud is building a testbed of **CDMI endpoints**

  Will cover several major institutes

- **Regular testing** to check what features are supported by the endpoints

  Setting QoS and DL policies can be one of these tests.

- **Progress** is measured by endpoints (and therefore, software) supporting different policies.

Next steps

- Starting building up the QoS/DL language.
- Start defining the CDMI extension.
- Adding CDMI support to dCache.
- Adding the QoS engine to dCache.

Thanks for listening, any questions?