File-event notification

Paul Millar
<paul.millar@desy.de>
dCache / DESY

This session ...?

- Collecting use-cases
- Discovering whether we (dCache) can do something quick that the VOs will find useful.
  - Useful for different VOs.
- Whether we can do something that others (DPM, Castor, Storm) can also implement.
- We're not aiming for perfect architecture, but it must be something we can live with.
  - Some future-proofing needed.

The basic ideas:

- Dealing with changes to file metadata
- Metadata from the SE's file catalogue.
  - but might include other metadata; e.g., whether a file's *actual* latency is online *right now*.
- Like to help external catalogues keep up-to-date with “reality”.
- Better if soln is SE neutral solution.
- Changes propagate to the VO, with QoS:
  - Latency
  - Reliability should VO's agent be unavailable for a period

How good is good enough?

- “the consumer doesn't need to be alive for the notification to be done, and it can "catch up" [...] Even if notifications are 90% reliable, then it's 90% more reliable than what we have now.” -- Brian Bockelman (CMS)

Security

- Is security an issue?
  - Events ~= reading the catalogue.
- Who should be allowed to access events?
  - Anyone,
  - Registered DN only,
  - VOMS (any member of a VO),
  - VOMS+specific role,

Scenario 1

- Files in the namespace are removed.
  - Site-admin is motivated to remove entries:
    - A disk, part of D1T0, fails at a site.
    - Site admin deletes files to make more space available (is this a site autonomous decision?)
    - Other possibilities?
  - A transfer fails, but the VO thinks it succeeded.
  - A file is deleted by user, but VO is unaware.
- VO is somehow notified that the file is deleted.

Scenario 2: job steering

- Files in D0T1 are brought online.
- The VO is notified that files are now “online”.
- The VO's job meta-schedular chooses the site preferentially for jobs that need the data.
- When the files are flushed from the disk cache, the preference for the site is removed.
- Questions:
  - There are potential issues with latency.
  - Are staging requests likely to be ad-hoc?
  - Is this covered already by SRM?

Scenario 3 SE/PhedEx-DB sync.

- A site admin happens to delete a local file
  - to free up space,
  - due to a lost disk server,
  - or because they found out the system's methods for automatic deletion stopped working 6 months ago
- Currently, they must file a ticket asking central operations (Slow, error-prone).
- With notifications, a PhEDEx agent could automatically invalidate deleted files from PhEDEx or DBS.

Scenario 4: CMS analysis

- Now, jobs are sent to data. A physicist must:
  - Request the dataset
  - Get it approved
  - Wait for it to be downloaded
  - *Then* create and send jobs.
- If we had file-arrival notifications:
  - Send the job description to the local server
  - The local server requests the data
  - Data still must be approved
  - As each file downloads, local server launches job.
  - Physicist gets an email saying results are ready.
- The person need wait only for jobs to finish.

What's useful to record?

- All files (including those on tape)
- Definitely
  - Full name
    - SE internal name, maybe not same as the SURL
- Optionally...
  - Last accessed,
  - When written,
  - Owner (user-ID for Castor),
  - Size,
  - Checksum (SE-specific),
  - Space (Token) they're in.