

Storage-Events and dCache: new frontiers in storage

Wednesday, 3 April 2019 16:00 (30 minutes)

dCache has introduced a new paradigm for scientific storage: storage events. In traditional interactions there is always the same pattern: the client requests some operation and the server replies with the result of that request. If the client wishes to learn the current status of some resource, it makes a request for this information. To discover when the state of that resource has changed, a client polls the service by making periodic requests that query the current status of the resource. Note that the time between successive queries must be carefully chosen: querying too often may place considerable strain on the storage system. Therefore, polling inevitably introduces latency.

It may happen that it is impossible to avoid placing unacceptable load on the storage system while providing a sufficiently quick response. In such situations, a typical solution is for the agent that modifies the storage system to inform interested parties through some communication side-channel. This coupling between the client that interacts with the storage and the agent reacting to the changes brings several disadvantages: the client must be custom (potentially a maintenance and platform availability problem), and it must know which agents it must inform (potentially a discovery problem).

Moreover, some events are apparently spontaneous and are not a result of client interactions. As an example, consider a client that wishes to learn when a file currently stored on a high-latency media (such as tape) is available for reading with low latency. This event is not triggered as the result of external client interaction, but rather from the activity of the tape system.

In the storage-event model, clients are alerted to activity within the storage system that they are interested in. The key distinction is that this happens without polling: when an interesting event happens, dCache informs the client with minimal delay. This allows the client to learn of changes in near realtime, without placing load on the server.

Storage events are widely applicable. They allow more robust and scalable solutions to several existing infrastructure challenges. For example, file catalogues may use storage events to learn of changes autonomously, without requiring custom clients; clients can request a large number of files be staged back from tape, and start processing as soon as files become available.

Storage events also allow for complex and innovative solutions. By breaking down the connection between the agent modifying storage and the agent reacting to those changes, new services are possible. For example, it is possible to couple computational workflows with data ingest events; domain-specific portals can react to data ingest quickly, so are always up-to-date with the available data.

dCache provides two mechanisms that support storage events: Kafka and Server-Sent Events (SSE). These provide complementary solutions that target different use-cases. Kafka storage-events are more suited for site-level integration, where the institute running dCache wishes to augment its behaviour by integrating it with other services. SSE, a standard web protocol for notification, gives ordinary dCache users access to storage events, allowing ad-hoc innovation.

In this paper, we will present a review of the storage events concept, an update on the current support for storage event support in dCache, and an overview of how various projects are using or plan to use storage events.

Primary authors: Dr ROSSI, Albert (FNAL); Dr LITVINTSEV, Dmitry (FNAL); Mr STAREK, Jürgen (DESY); Ms SAHAKYAN, Marina (DESY); Dr ADEYEMI, Olufemi (DESY); Dr FUHRMANN, Patrick (DESY/dCache.org); Dr MILLAR, Paul (DESY); Mr MKRTCHYAN, Tigran (DESY)

Presenter: Dr FUHRMANN, Patrick (DESY/dCache.org)

Session Classification: Data Management & Big Data

Track Classification: Data Management & Big Data