CASTOR end-to-end monitoring system

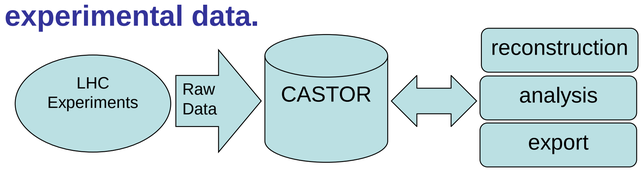
Witold Pokorski, Theodoros Rekatsinas[1], Dennis Waldron, Dirk Duellmann, Sébastien Ponce, Bartolomeu Rabaçal, Jacek Wojcieszuk

[1]Department of Electrical and Computer Engineering, National Technical University of Athens, Greece

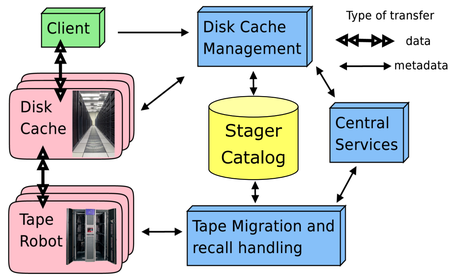
DM Data Management

CERN IT Department

The CERN Advanced STORage system (CASTOR) is to be used by the LHC experiments for storing experimental data.



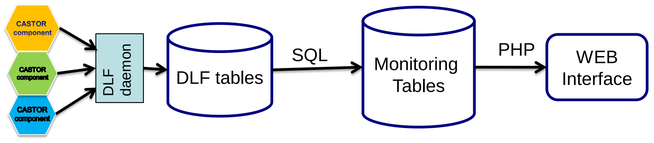
CASTOR architecture is based on the disk cache which serves to transfer data from and to the tapes. The system is able to serve thousands of requests per second, while managing millions of files.



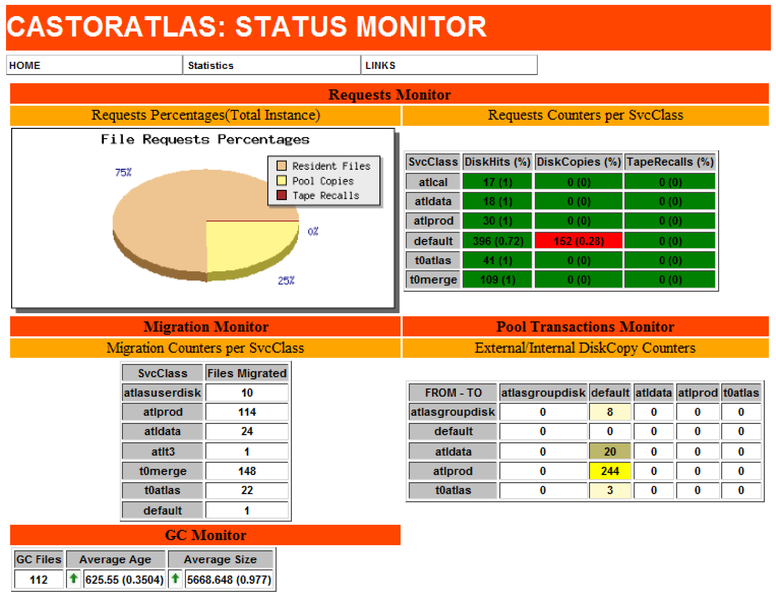
Due to the complexity of the system, an end-to-end monitoring system is needed
- **for users to check the current status**
- **for operators, to diagnose any problem**
- **for developers to observe the effectiveness of different algorithms and policies**

General Architecture

Our new high-level CASTOR-dedicated monitoring system consists of a Database back-end for event processing and storage and a Web Interface for visualization. The stream of log messages, generated by the different CASTOR components, is used as input. Those raw messages are initially collected by the Distributed Logging Facility (DLF), and then combined into a high-level, compact monitoring data.



- PL/SQL monitoring procedures pre-process incoming DLF messages accumulating and storing information for every type of event
- Monitoring tables correspond to specific event oriented metrics
- Web interface provides aggregated visualization (histograms, pie-charts, etc) of the monitoring data



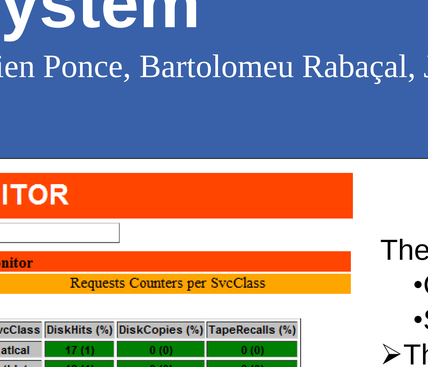
Monitoring DB schema

The Database Schema is composed of:

- Monitoring Tables, where each entry contains the full descriptions of the respective CASTOR event
- Materialized views, which implement CASTOR status monitors

Monitoring tables (DB) isolate the front-end (web interface) from the back-end (transport for log messages) offering design flexibility and the possibility to:

- switch system input from DLF transport to SYSLOG
- feed the data into other existing general monitoring systems like LEMON (LHC Era Monitoring) and similar



- implement several web interfaces simultaneously
  - summary dashboard for experiments
  - detailed monitoring for developers and operators

PL/SQL Procedures

PL/SQL procedures run periodically every 5 minutes to extract the information about significant CASTOR events from the incoming data stream. Some of the monitoring information comes directly from individual log messages (e.g. facility errors) while for the monitoring data like

- file read requests
- file migration to tape
- file garbage collection

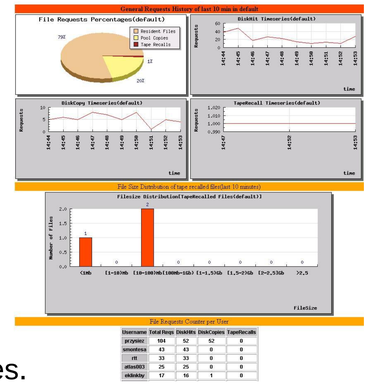
one needs to combine multiple daemon logs and store them as a single entry.

CASTOR monitoring web interface

The Web interface consists of two different levels
- Current Status Dashboard
- Statistical Information

➢The Dashboard feature uses tables with number indicators to present system's current status. Some plots (histograms, pie charts) are used to display short-term historical information.
Source: DB Materialized Views

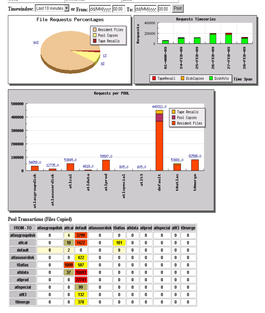
➢More detailed statistical and historical Information are in place to monitor CASTOR activity in time. Simple queries containing group, aggregate or analytic functions implement the desired metrics, which are displayed using JpGraph libraries.



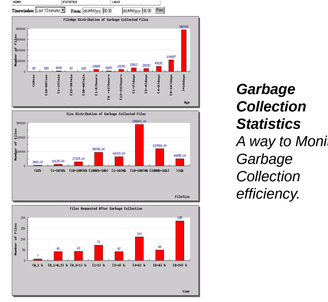
Castor Status Dashboard *Activity in Default Service Class, ATLAS CASTOR Instance (10 Minutes)*

In order to serve multiple user requests, retaining DB load to a minimum, JpGraph's built-in caching mechanism is used to cache images for repeated requests.

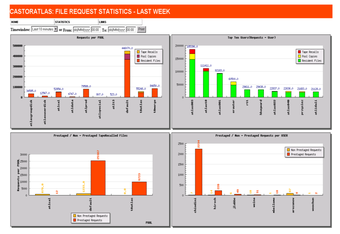
Examples of different metrics collected and displayed



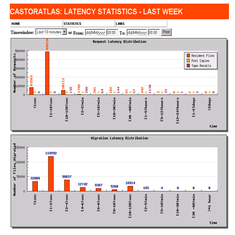
General Activity Statistics *File read request time series and disk cache efficiency metrics are displayed. Visualization of Disk-to-Disk file copies.*



Garbage Collection Statistics *A way to Monitor Garbage Collection efficiency.*



User Specific Metrics *Monitor user activity in time.*



Latency Statistics *Monitor CASTOR services efficiency.*

Performance and Stress Tests

To ensure that the performance requirements are met, the new monitoring system has been extensively tested using simulations of real life scenarios.
✓ PL/SQL procedures were tested by processing real CASTOR metadata. Their scalability was tested by replaying past data in higher speed (factor of 10). A C++ program was implemented to reconstruct daemon logs, compress and inject them into DLF.
✓The web interface was tested by simulating load scenarios with WAPT, a load, stress and performance testing tool for web sites.

CERN IT Department
CH-1211 Genève 23
Switzerland
http://cern.ch/it-dm

CHEP'09, 21 - 27 March 2009 Prague, Czech Republic