

CMS usage of NoSQL databases

Mattia Cinquilli

IT-ES-DNG Section Meeting
31-05-2011

- NoSQL: CouchDB & MongoDB
- DAS
 - Architecture
 - NoSQL motivations
- WMAgent
 - Architecture
 - NoSQL motivations
- Conclusions

Features:



- Document Store.
- Written in Erlang.
- Manages a collection of JSON documents.
- Exposes a RESTful HTTP API.
- Implements Multi-Version Concurrency Control.
- Distributed Architecture with Replication.





Features:

- Document Store.
- Written in C++.
- Manages a collection of JSON documents.
- Exposes a RESTful HTTP API.
- Provides atomic operations on fields.
- Supports dynamic queries with automatic use of indices.



DAS

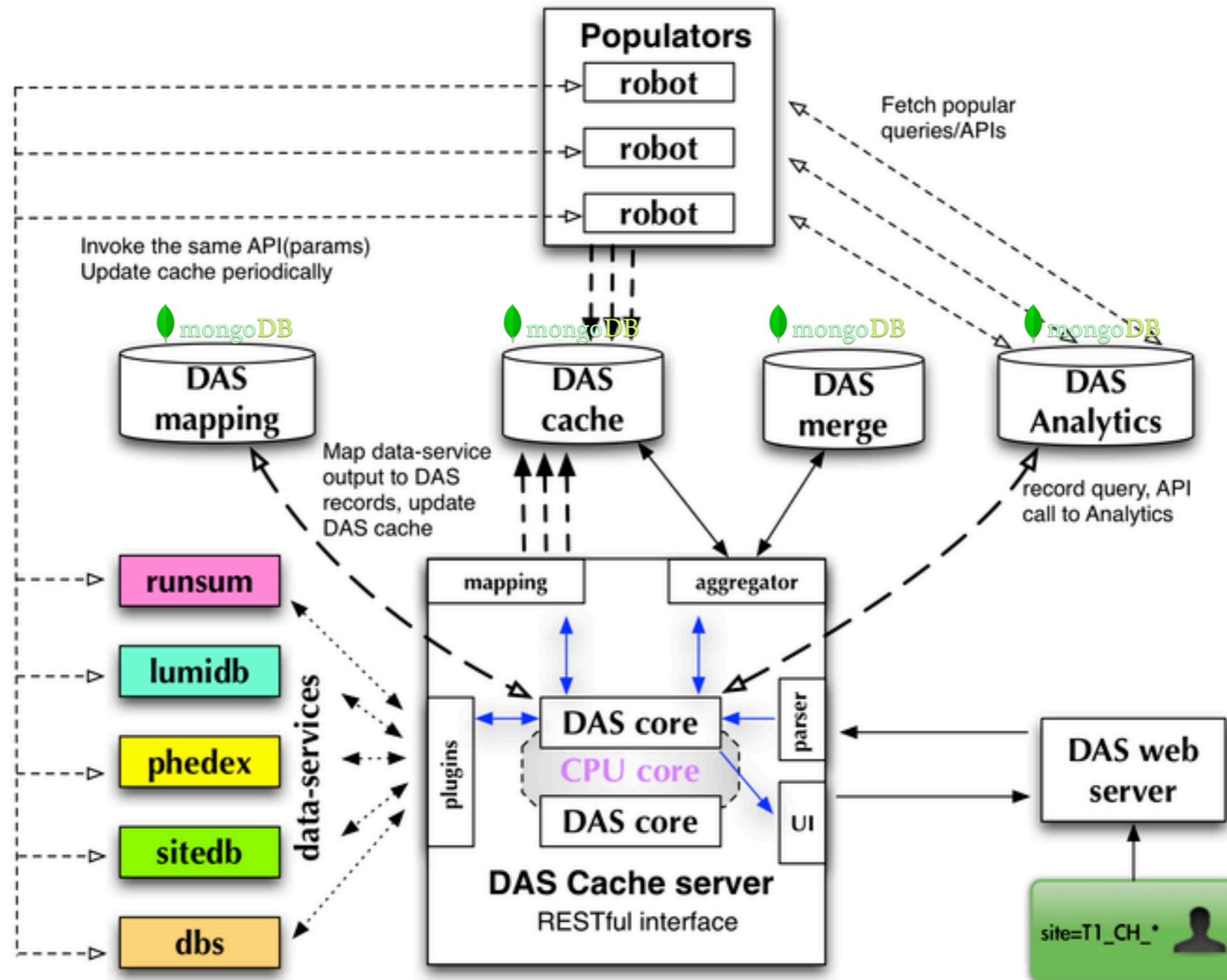
Data Aggregation System

- The Data Aggregation System provides to the users the ability to search and aggregate information across different data-services without knowing the specific data sources.
- (On demand) Fetches and aggregates meta-data information from existing CMS data-services under one single point by using services API and preserving specific security policies.
- Main key points and features:
 - transparent central meta-data repository
 - acts as proxy and caching layer
 - keyword search based system, with conditional operators
 - provides a common data representation

“Data Aggregation System - a system for information retrieval on demand over relational and non-relational distributed data sources”

Presented at CHEP 2010

Ball, G. (Imperial Coll., London); Kuznetsov, V. (Cornell U.); Metson, S. and Evans D.



- No needs of relational databases:
 - DAS does not require data preservation and transaction capabilities
 - Dynamic type of stored meta-data objects (no predefined schema)
- Document oriented databases (MongoDB, CouchDB)
 - Schema-less: arbitrary document structure storage
 - Replication and failover features
- MongoDB advantages:
 - Support of dynamic queries (like relational databases)
 - Full-text indexes (index defined on every single word of a field)
 - Auto-sharding



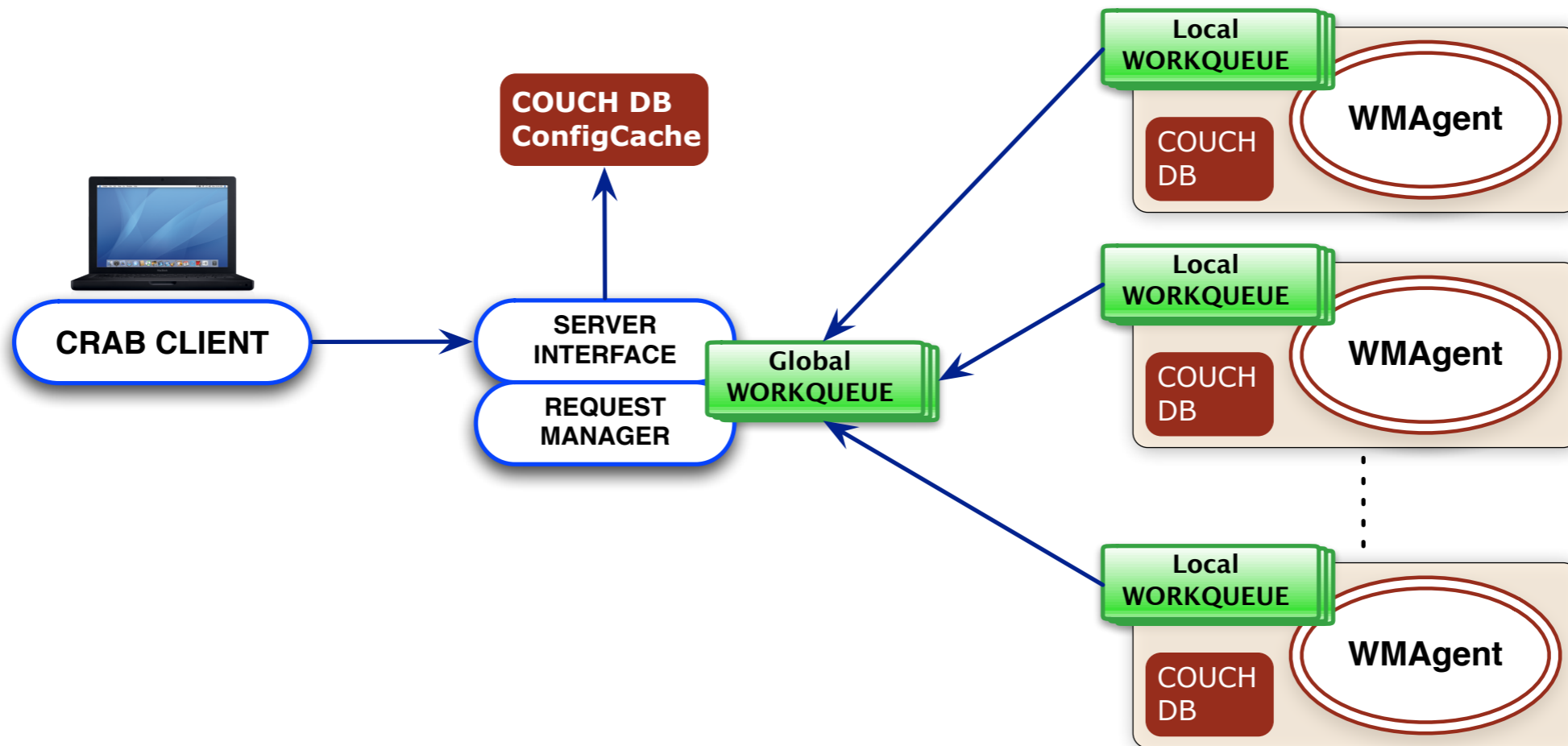
WMAgent (WM = Workload Management)

- WMAgent will replace all the three CMS workload management tools (Tier-0, ProdAgent, CRAB) with a completely new tool that will improve known bottlenecks and issues.
- It aims to provide a common infrastructure where the different workflows (analysis, (re)processing, ...) will run by using the very same code (a part of configurations and plug-ins), saving development man power, improving support with a common operational effort.
- CMS has just started real reprocessing workflow with WMAgent, while real user analysis workflows will start to run by the end of the year.

“Job life cycle management libraries for CMS workflow management projects”

Journal of Physics: Conference Series Volume 219 Part 4

Frank van Lingen et al 2010 J. Phys.: Conf. Ser. 219 042024

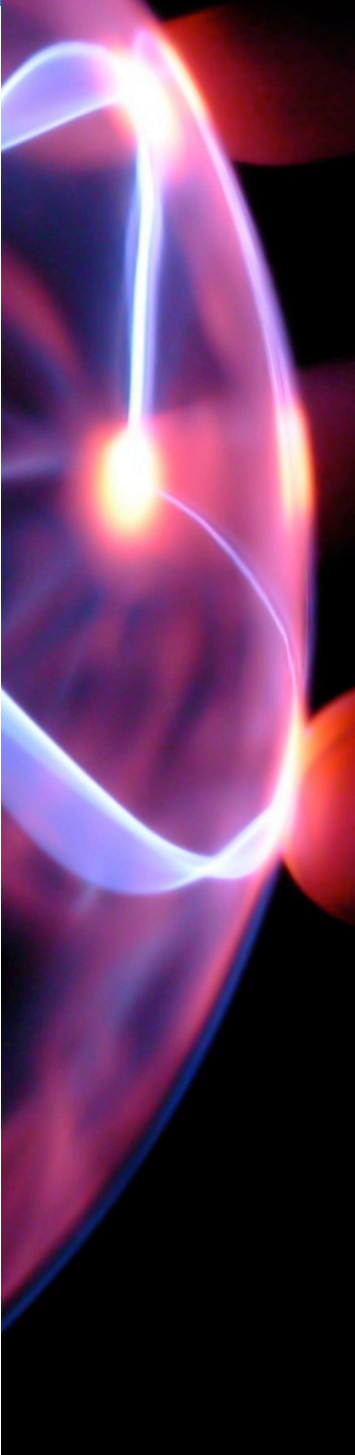


- Some reasons to adopt CouchDB:
 - data format changes very often and in many cases data does not have meta-data (thinking about the job reports and different formats of errors)
 - remote access of information and data replication between different endpoints, allowing an easy way to show monitoring information without impact on the system
- WMAgent intensively uses a local relational database (MySQL/Oracle) to handle transaction and status changes between different components inside the agent itself.
- CouchDB has been coupled with each WMAgent instance to:
 - reduce and decouple from the core database the load generated by monitoring needs
 - providing an easy point where to show/propagate/replicate information (job history, workflow reports, ...) and to provide API to build other applications
- Example: the agent itself pushes both job status information and job framework job reports into CouchDB, which becomes a storage where task/job information can be easily browsed through web pages

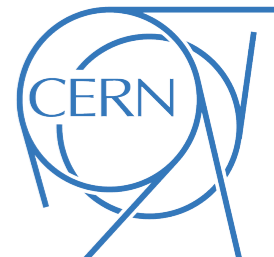
ES

WMAgent & CouchDB

CERN IT
Department



CERN IT Department
CH-1211 Geneva 23
Switzerland
www.cern.ch/it



Overview

+ Create Database ...

Name	Size	Number of Documents	Update Seq
reqmgrdb	24.1 KB	2	4
_users	4.1 KB	1	1
wmagent_acdc	72.1 KB	2	2
wmagent_configcache	184.1 KB	17	43
wmagent_jobdump/fwjrs	48.1 KB	1	1
wmagent_jobdump/jobs	44.1 KB	1	1

Showing 1-6 of 6 databases

← Previous Page | Rows per page: 10 | Next Page →

CouchDB
relax

Tools

- Overview
- Configuration
- Replicator
- Status
- Test Suite

Recent Databases

- _users
- wmagent_acdc
- wmagent_configcache
- wmagent_jobdump%2Ffwj
- wmagent_jobdump%2Fjob

Signup or Login

Futon on Apache CouchDB 1.0.2

*Example of a CouchDB overview web page
(instance used for testing and development)*

Apache CouchDB - Futon: Overview

Global Monitor x Summary for job 15000 x Apache CouchDB - Fut... x Apache CouchDB - Fut... x Apache CouchDB - Fut... x Apache CouchDB - Fut... x

Summary for job 4576

Overview Global Monitor x Summary for job 4576 x

http:// :5984/wmagent_jobdump%2Fjobs/_design/JobDump/_show/jobSummary/4576

Create Database

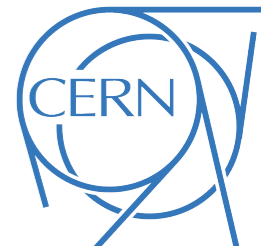
Name: 40d0f0e4-8084-11e0-b8f2-00221959e72f-843
 Owner: cmsdataops
 Workflow: cmsdataops_EWK-Summer11_R1-00011_110513_130745
 Task: /cmsdataops_EWK-Summer11_R1-00011_110513_130745/ReDigi
 Mask:
 1: [24128, 24128]
 Input Files:
 /store/mc/Summer11/QCD_Pt-15_TauBiased_TuneZ2_7TeV-pythia6/GEN-SIM/START311_V2-v1/0033/5A609527-516E-E011-9A03-0023AEFDEDA0.root
 State Transitions:
 Tue May 17 2011 14:50:38 new -> created
 Tue May 17 2011 14:53:34 created -> executing
 Wed May 18 2011 00:03:46 executing -> complete
 Wed May 18 2011 00:04:28 complete -> success
 Wed May 18 2011 00:05:42 success -> cleanout
 Output Files:
 /store/unmerged/Summer11/QCD_Pt-15_TauBiased_TuneZ2_7TeV-pythia6/DQM/PU_S3_START42_V11-v2/0000/72F3B1B9-CB80-E011-91F9-002590200B38.root
 Used by: [167455](#), [171689](#)
 /store/unmerged/Summer11/QCD_Pt-15_TauBiased_TuneZ2_7TeV-pythia6/GEN-SIM-RECO/PU_S3_START42_V11-v2/0000/C25A1BB7-CB80-E011-91F9-002590200B38.root
 Used by: [27615](#), [27925](#)
 Errors:
 (none)
 Log Archives:
 Retry 0 -> /store/unmerged/logs/prod/2011/5/17/cmsdataops_EWK-Summer11_R1-00011_110513_130745/ReDigi/0000/0/40d0f0e4-8084-11e0-b8f2-00221959e72f-843-0-logArchive.tar.gz
 Used by: [167458](#)

*Example of a couchapp that shows the jobsummary
(production CouchDB form reprocessing activities)*

The screenshot displays two web browser windows. The top window is Apache CouchDB Futon, showing the 'Overview' page for a job with ID 4576. The job details include: Name: 40d0f0e4-8084-11e0-b8f2-00221959e72f-843, Owner: cmsdataops, Workflow: cmsdataops_EWK-Summer11_R1-00011_110513_130745. The bottom window is the Asynctransfer Monitor System, featuring a CMS logo and a navigation menu with links for HOME, General Statistics, Files count by User, Files count by WorkFlow, Status by Dest-Source, and User Statistics. The main content area displays a stacked bar chart titled 'Job Status Gruped by Link [Destination, Source]'. The chart shows the number of jobs in 'done' (green), 'failed' (red), and 'other' (yellow) states across various links. The y-axis is labeled 'Number of Jobs' and ranges from 0 to 100. The x-axis is labeled 'T2_IT_Legnaro,T2_PT_LIP_Lisbon'. A legend indicates: done (green), failed (red), other (yellow). A tooltip instruction reads: '- Move the mouse over a section of the stacked bar chart to get more information. - The click on a section of the stacked bar chart will link to the statistics page of the selected link.'

More sophisticated couchapp example (integration CouchDB, CRAB asynchronous transfer system)

Thanks to *Hassen Riahi* for providing this image



- NoSQL has certainly many advantages coming from a newer technology with many features (map/reduce queries, REST interfaces, sharding, ...) and CMS has already tested and developed tools that take advantages of these...
 - ...but issues disclosure on production services has just started in CMS and first experience with real running services is being currently done.
- Very well known issue: an SQL expert developer takes some time to start efficiently with not relational database.
(Often this time is recovered due to the shorter development time taken by NoSQL database)
- When developing new things or need to rewrite applications, it is worth that dashboard team evaluates the usage of not relational database solutions (even on top of already existing systems)
- Next week there will be a 2 days workshop @CERN about “Database Futures Workshop”, with some contribution related to NoSQL usage and evaluation:
<https://indico.cern.ch/contributionListDisplay.py?confId=130874>