Modernizing the monitoring of Mass-storage systems

Alexandre Terrien

August 11th 2016

CERN IT-ST-FDO

Why do we need storage ?

No data, no analysis. No analysis, no papers. No papers, no CERN.

Experiments produced 10PB of data in July
Need to develop storage solutions

- Tape and disk, started late 90's
- Pure disk, started in 2010

Both still used in every experiment
Need to work correctly
Focus on CASTOR and its monitoring : log messages.

Log messages

What is a log message ?

Produced during relevant steps of the program execution

- Startup/shutdown
- Request received (from an user)
- End of processing
- Error

10 million messages per day

Line of formatted text. Example :

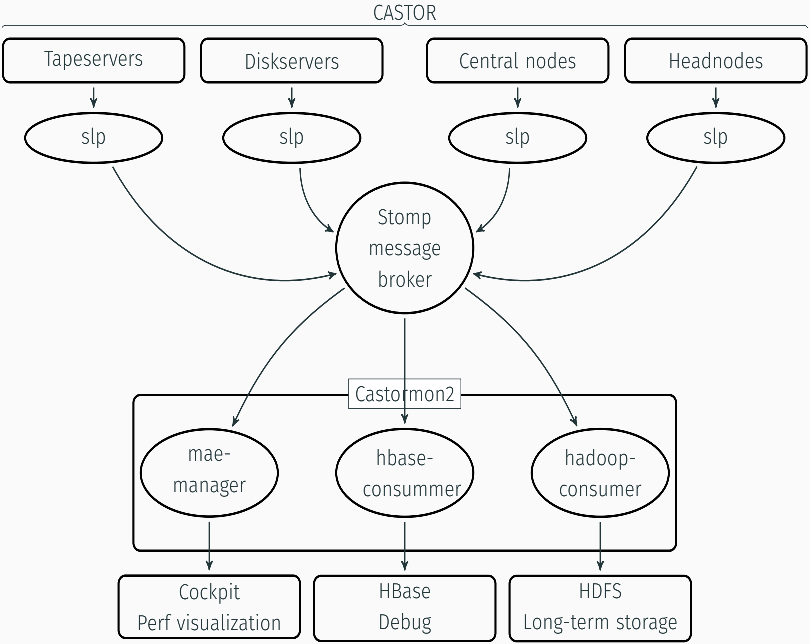
```
2016-07-21T16:30:43.172316+02:00 lxc2dev7.cern.ch nsd[18479]:
    LVL=Info TID=18530 MSG="Processing complete" REQID=54
    d41ca0-8f13-4dba-a966-1ef771d742f4 Function="closedir"
    Username="aterrien" Uid=93615 Gid=1028 Secure="No"
    ClientHost="lxc2dev7.cern.ch" Cwd="" Path="/" Guid=""
    RtnCode=0 ProcessingTime=0.049
```

Gives info on what, when, where, who

What do we do with log messages ?

- Know when something goes wrong
- Debug
  - History of a file
  - History of a request
- Statistics : performance of system
- Data mining : analysis of trends
- Backups : long-term storage

Project presentation


CASTOR
Tapeservers
Diskservers
Central nodes
Headnodes
slp
slp
slp
slp
Stomp message broker
Castormon2
mae-manager
hbase-consummer
hadoop-consumer
Cockpit
Perf visualization
HBase
Debug
HDFS
Long-term storage

My project at CERN

Replace most of our log management code.

- Use IT central services
- More adequate tools
- Better performances
- No single point of failure
- Less maintenance for us

Log messages...

- production
- routing
- aggregation

Results

Format log messages to a homogeneous format.
Regroup them on a single machine

Before

Production :
simple-log-producer

- Plugin-based
- Self-maintained

Routing : Message Broker

- Better solutions now

After

Flume for everything

- Apache project
- Supported by IT-MM team (not us !)
- Only have to maintain config files

Debug : storing into HBase

Know what happened to a file, a request

Filter messages, then write them into HBase

Independent messages : parallelization

Before

HBase-consumer

- Self-maintained
- Runs on one machine
- Everything done “By hand”

After

Apache Spark script

- Apache project
- Uses Hadoop cluster
- Describe high-level operations

Statistics : Performance, metrics

- Get info on the performances of the system
- Count, do statistics on some messages
- Filter, then regroup, then compute (MapReduce model)
- Apache Spark once again, for the same reasons
- Replaced tool named Metrics Analysis Engine
  - $\approx$ 3k lines of code
  - Lots of unused, complex code
  - Very CPU intensive on one machine

Store log messages to HDFS to recover in case of failure

Before

Hadoop-consumer

- Self-maintained
- Was actually working fairly well

After

IT-MM team's flume agent

- They do it "for free"
- Nothing to do on our side

- Replaced a lot of aging code
- IT central services
- Use of purpose-built tools

Thank you !

Backup slides

CERN Advanced STORage manager

Hierarchical storage system

- Disk and magnetic tape
- Different access protocols
- No replication

CASTOR

CERN Advanced STORage manager

Apache Spark

- MapReduce framework
- Python, Java, Scala, R
- High-level applications

Example code :

```
text_file = spark.textFile("hdfs://...")
text_file.flatMap(lambda line: line.split())
    .map(lambda word: (word, 1))
    .reduceByKey(lambda a, b: a+b)
```

Apache HBase

Database built on top of Hadoop

- Distributed
- “Horizontally” scalable
- Non-relational

Picture a big Excel spreadsheet
Use for CASTOR logs :

- One row : one file/request
- One column : one timestamp
- Associated cell : one log message