

CluMan

a cluster management tool

Sebastian Lopienski
Miguel Coelho dos Santos
Filipe Manana

HEPiX meeting – CERN, spring 2008

Large cluster management

- Tools already used
- Configuration inconsistency
- Re-configurations

the need

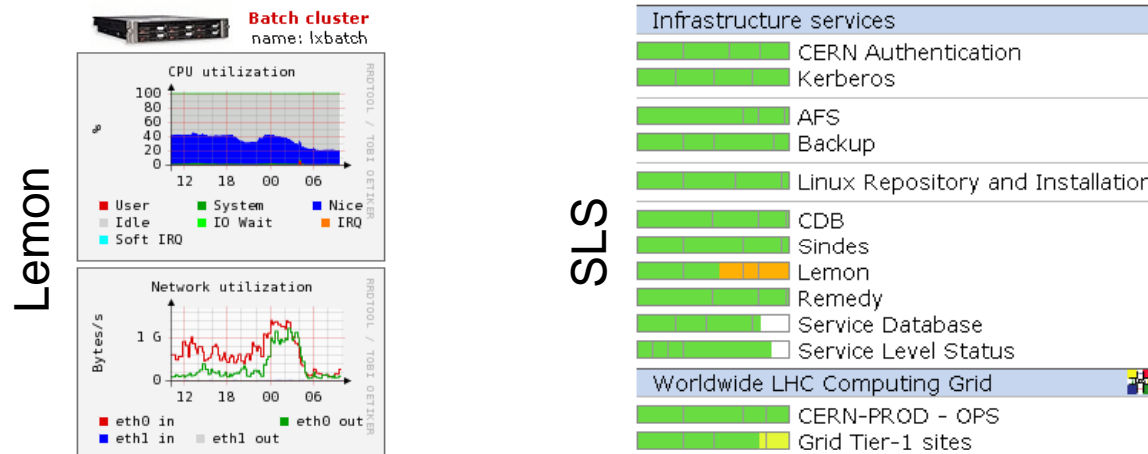
CluMan

- Main concepts
- Architecture
- Technologies

the solution

Large cluster management (the need)

- **CDB** – Quattor Configuration DB
- **SMS** – State Management System
- **SINDES** – Secure Information Delivery System
- **Lemon** – hardware and OS monitoring system
- **SLS** – Service Level Status display
- central logging (i.e. DLF, syslog-ng, etc)



We need all of these if we are to rise to the challenge.

- On large and complex clusters (services), where different applications/subservices are used, keeping **consistent configuration** across all components (i.e. CDB, SMS, LSF, CASTOR) **is a challenge**
- CDB should rule... but
 - not all configurations or status come from CDB – big systems with complex software usually have their own internal configuration database, sometimes with very dynamic changes
 - CDB is good for configuring hosts, but less so for configuring complex distributed services
 - some configurations need to be kept in synch across different pieces of software... **Tricky!**

- in large and complex clusters (re)configurations are often a **multi-step task**
- it usually involves different Quattor/ELFms components: CDB, SMS, SINDES, etc...
- large clusters are usually configured via standard config files but sometimes by custom tools that interact with application DBs
- even simple CDB changes also have their overhead
 - manual `cdbop:get:vi|perl|sed:update:commit`
 - should be a click of a button (or running of a script)

Example: adding a disk server to CASTOR

1. Put the server in **SMS** maintenance
2. Change the server's **CDB** profile to have the correct castor service templates and belong to the right service class
3. Reinstall (PrepareInstall: **AIMS, SINDES**)
4. Move the server inside **CASTOR** DB to the appropriate place (moveDiskServer)
5. Regenerate **LSF** configuration files from **CDB**
6. Restart **LSF** on master node
7. Release the server in **CASTOR**
8. Put the server in **SMS** production

Chances are, by time you reach the last step, the server has a hardware problem 😞

Checking configuration **consistency** is *critical*.

Automation is the only way to *scale*.

- Hardware interventions are constant in big clusters
- Like in reconfigurations, it is essential to have automated actions to remove and return single servers to/from the cluster:
 - on a batch node **draining** means waiting for all job to finish
 - on a disk server **draining** means copying all the data out
- These can be long tasks (i.e. several days)
 - we need to know for which node the tasks were launched
 - we need **to see progress**

- Automating configuration requires:
 - scripting of actions
 - ability to trigger several actions in parallel
 - ability to chain actions
 - track action progress / completion
 - monitor action errors

Accounting and Management driven reconfigurations

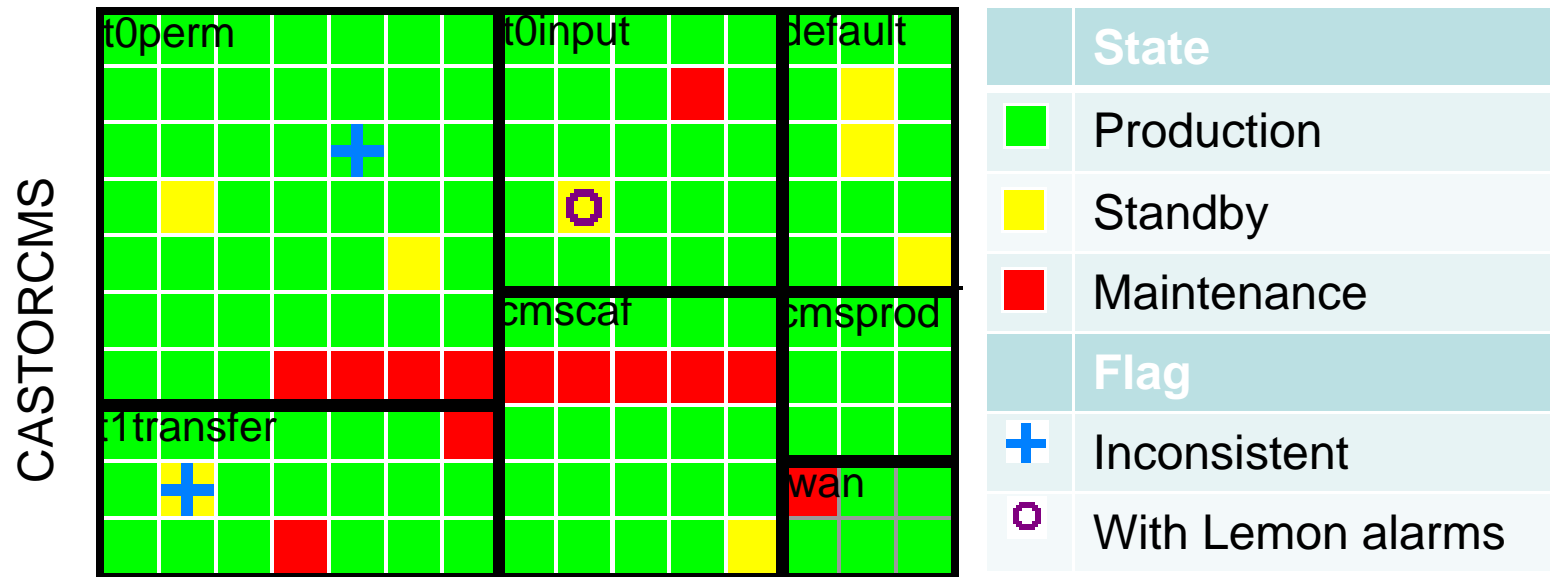
- Track resource allocation over time
- Track target capacities set by management
- When target capacity and allocation do not match, must trigger action on Service Provider

We need an interface where management can set targets

We need to display it

We need to make this work easy(-ier)

- Requirement: to see **a service overview**
- ... for example, as a HeatMap:



CluMan

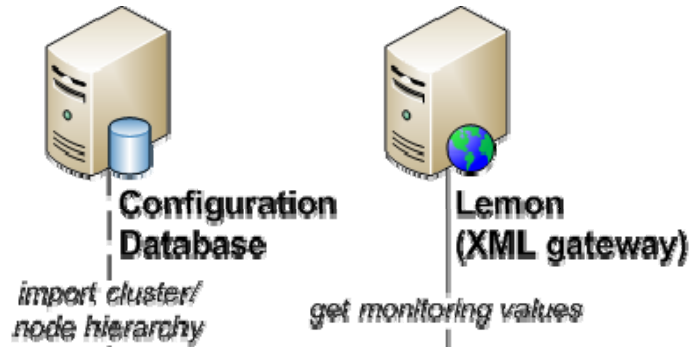
(the solution)*

* that is currently being developed by CERN IT/FIO

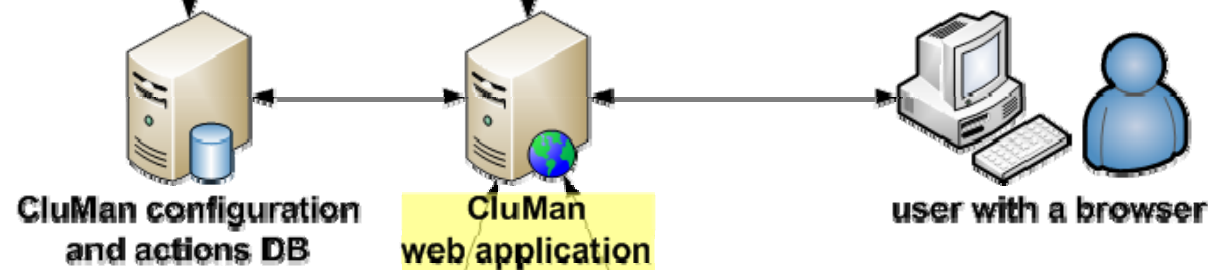
- Node **state**
 - a predefined set of possible values (states)
 - e.g. *production, stand-by, maintenance*
 - represented with different colors
- Node **flags**
 - e.g. *inconsistent configuration, warranty expires soon, in maintenance mode since long*
 - up or down; independent of each other
 - represented with icons
- Node and cluster **properties**
 - e.g. *average cpu load, free disk space*
- All are taken from relevant **Lemon metrics**

- Actions
 - **reconfiguration scripts** that run on nodes
 - launched from the web application
 - for single or multiple nodes, clusters etc.
 - they report back their progress and completion
- Views in the web application
 - actions launched (in progress) for a node, cluster
 - actions completed, stuck
 - *Retry* and *Mark as completed* buttons
- Action chains

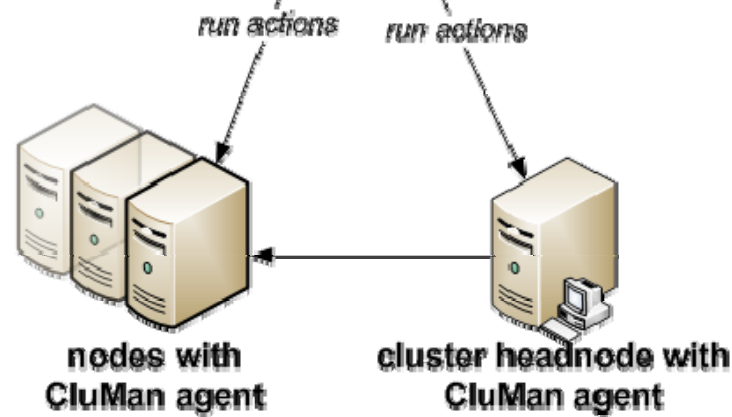
data sources



CluMan




managed machines



- **Google Web Toolkit** – a Java web application framework
 - for AJAX applications; converts Java code into JavaScript
- **Apache iBATIS** – an object/relational mapping framework
 - maps SQL queries (kept in external XML files) to Java objects
- **YAML** – a text format for data serialization
 - human-readable (unlike XML); data structures: records, arrays...

```
# a tree of entity records
- entity: &c2atlas # &c2atlas is an anchor for the record
  id: 123
  type: cluster
  name: castor

- entity:
  id: 307
  type: node
  name: lxfsrd4506
  parent: *c2atlas # reference to the previous record
```



- Quattor CDB: <http://www.quattor.org>
- SMS: <http://cern.ch/leaf>
- SINDES: <https://twiki.cern.ch/twiki/bin/view/FIOgroup/SinDes>
- SLS: <https://twiki.cern.ch/twiki/bin/view/FIOgroup/ServiceLevelStatusProject>
- Lemon: <http://cern.ch/lemon>

- Google Web Toolkit: <http://code.google.com/webtoolkit/>
- Apache iBATIS: <http://ibatis.apache.org/>
- YAML: <http://www.yaml.org/>



E-mail:

Sebastian.Lopienski@cern.ch

Miguel.Coelho.Santos@cern.ch

Filipe.David.Borba.Manana@cern.ch

Website:

<https://twiki.cern.ch/twiki/bin/view/FIOgroup/CluManProject>

Acknowledgements:

Olof Barring, Ignacio Reguero, Ricardo da Silva,
Jan van Eldik, Veronique Lefebure