DAQ LHC Workshop
Configuration management systems

Aymeric Dupont

On behalf of Alice, Atlas, CMS and LHCb

Summary

- Generalities
- Dropping Quattor ! Why ?
- Expectations
- Which candidate to take over ?
- Why puppet won the vote ?
- General status
- Experiments state of the art
- Experiments specific software deployment
- Evolution for LS1
- Conclusion

Generalities

- **Situation**
  - Why configuration management ?
    - Deploy and maintain
    - Ensure consistency of configuration
    - Guarantee service performance and persistency of a computing infrastructure
  - Which challenges are we facing ?
    - Growing complexity and heterogeneity
    - Increasing mission importance
    - Maintaining this with the same manpower
- **Current tool**
  - IT & experiments relying on Quattor (till past 2-3 years at least)

Why dropping Quattor ?

- **IT**
  - « home-made » solution (no longer supported)
  - Open-source solutions becoming more and more serious alternatives
- **Atlas**
  - Various configuration aspects not handled
  - Modules lacking of consistencies (no support)
  - Complex packages management (dependencies, workload)
  - Lack of flexibility in configuration instructions (execution order)
- **Cms**
  - To follow industry standards
  - To support much varied configurations aspects.
- **LHCb**
  - No dropping scheduled yet

Expectations

- **Which expectations for a new tool ?**
  - Wider features :
    - Installing/Configuring software
    - Configuring hardware (bonding, routes)
    - Handling every aspects of the administration of a heterogeneous infrastructure (virtual hosts declaration...)
  - Software support :
    - Continuous development
    - Bug tracking/fixes

Potential candidate evaluation

- **Cms**
  - CFEngine (2 months)
    - Grand-daddy solution for configuration management (Used by major computing company (Facebook, Cisco, AMD...))
- **Alice & Atlas**
  - CFEngine
  - Chef
    - Actively contributing community (recipes repositories)
- **LHCb**
  - First Puppet tests last year
  - New evaluation campaign to be launched
    - Migration to another CMT still being discussed (Experiment no longer mentionned in the next slides)

Why Puppet won the vote ?

- **IT**
  - Migration to Puppet as core of Agile Infrastructure (triggering factor)
- **Modular configuration**
  - Modules much easier to create
- **Finer grain management**
  - Additional facilities offered (files tidy-up, ssh keys...)
- **Large community support**
  - Lots of users feedbacks
  - Plenty of contributors (puppet forge)
- **Syntax**
  - Clear and easy to understand
  - Declarative approach similar to quattor
- **Behaviour**
  - Quattor-like (unlike CFEngine requiring successive runs)
- **First experience**
  - Small cluster administration from 2008 @ Univ. Johannesburg

Experiments
State of the art

General status

- **Production experience**
  - Atlas
    - ~3 years
    - TDAQ testbed (~300 systems)
    - All Point 1 machines administered.
  - Alice
    - ~1 year (production farm)
- **Cms**
  - Spare SLC6 templates (1 machine only)
  - Storage manager (1 machine only)

State of the art

Aspects	Alice	Atlas	CMS
Supported systems	SLC5+6		SLC6
Changes versioning	SVN		Local git
Changes propagation	On master then pushed to SVN	SVN itself	Modified Puppet-sync
Repositories management and setup	Flat repository Home-built RPMs versioned with SVN	Snapshotted by home-made scripts	
Experiment specific packages deployment	See next slide		

Specific packages deployment

- **Alice**
  - Explicit declaration for each subsystems
- **Atlas**
  - Explicit segregation based on environment vars for DAQ and Offline
  - Installation handled by Trigger and DAQ librarians
  - Yum repos pointing to specific snapshots for OS packages
- **Cms**
  - Dedicated repositories snapshotted by versions for DAQ
  - Timestamped snapshots for OS repositories (like IT)

State of the art (2)

Aspects	Alice	Atlas	CMS
Certificates for new machines	Removed during install Autosigned at first puppet run		Created on the fly during install
Monitoring	Logs	Puppet dashboard	None
Scalability	Single machine	Serverless setup	Single machine
Manifests browsing GUI	None	Auto-generated inclusion tree	None
Modules layout	One module/function (auth...)		One module/service (sssd/krb5/autofs/...)

LS1 Evolution

- **Alice**
  - Full scale migration
- **Atlas**
  - Migration to puppet 3 + puppetDB
  - Dashboard deprecated => Foreman
  - Pulp (Katello UI) for repo management ?
- **Cms**
  - Full scale migration (SLC6 only)
  - Improve scalability (still running on Webrick !)
  - Monitoring => Foreman
- **LHCb**
  - SLC6 migration (Quattor server)

Conclusion

- **Expectations fulfilled**
  - Puppet does the job (valid production experience)
  - Having IT migrating is a great asset
- **Collaborating and sharing informations**
  - Good contact with IT
  - Nice collaboration between experiments (Atlas scripts&docs to mirror&snapshot repositories)
  - Xxperiment meetings (IT attends them)
- **New challenges**
  - Setting up pre-production systems
  - Writing migration policies and validation procedures

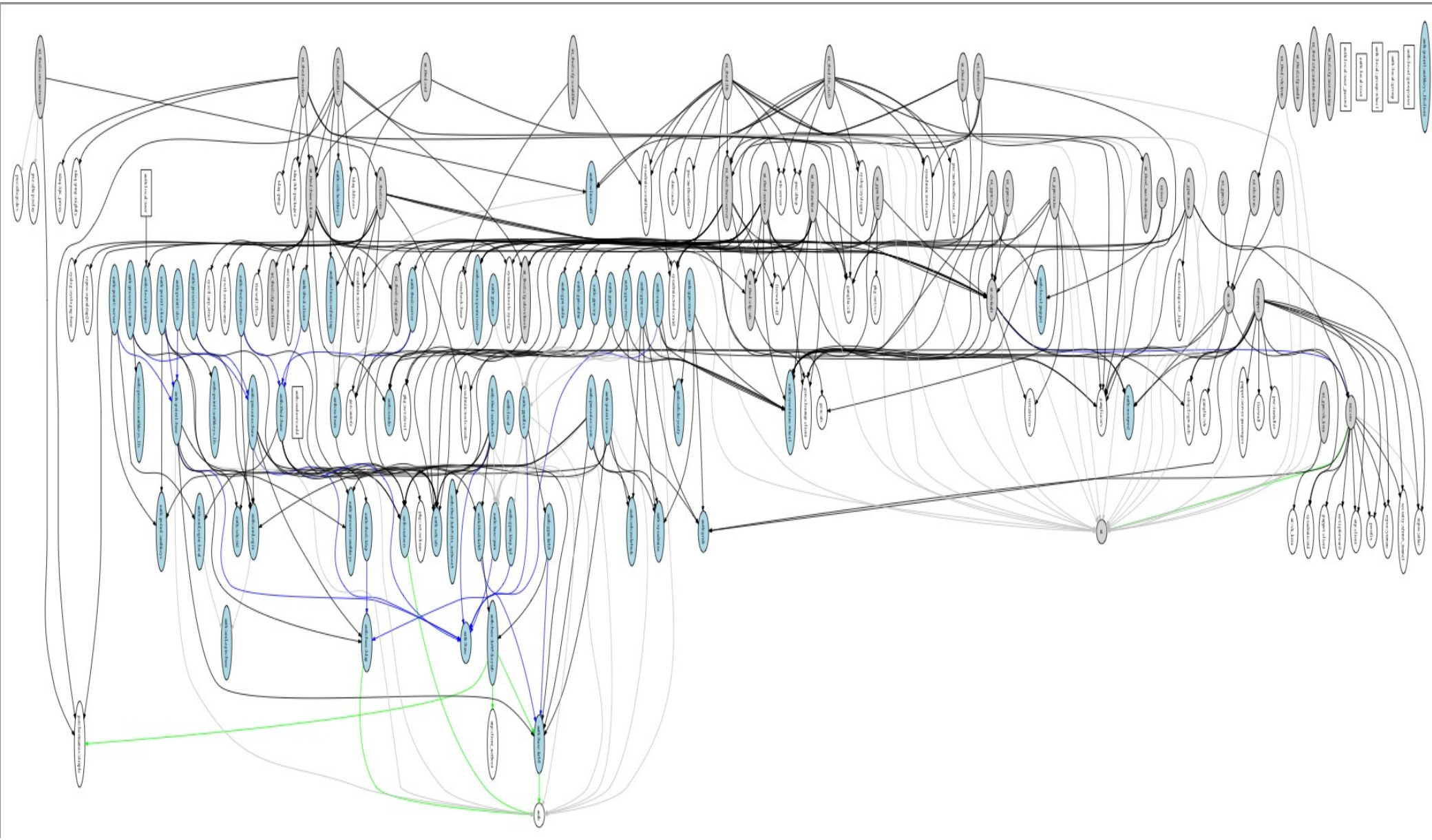
Thanks to...

- **Alice**
  - Ulrich Fuchs
  - Adriana Telesca
- **Atlas**
  - Sergio Ballestrero
- **Cms**
  - Marc Dobson
- **LHCb**
  - Niko Neufeld
  - Loïc Brarda

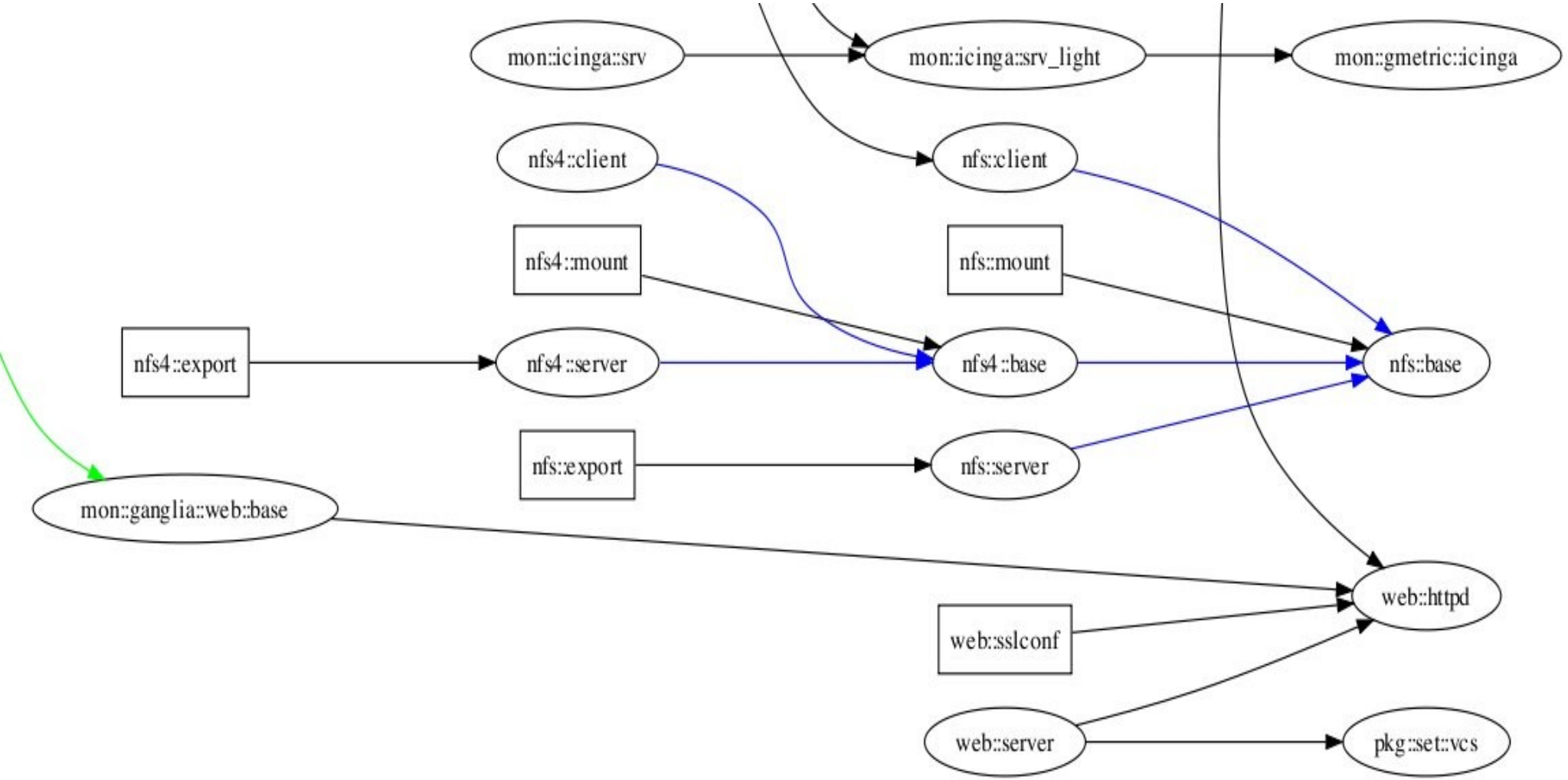
Questions ?

Extra slides

Atlas inclusion tree graph



Atlas inclusion tree graph (detail)



Puppet code example

Node declaration

```
## Public/build nodes
## with development tools
class nt_tbed::public {
    class { "nt": type=>"tbed::public" }
    include gen::hostnames::simple
    include auth::selinux::enforcing
    include nt_tbed::base::client
    include ganglia::cli
    ganglia::gmond::plugin {"users":}

    ## shared, single ssh-host-key
    include auth::ssh::hostkeys

    ## Applications
    include nt_tbed::cfg::develtools
    ## HLT
    include tdaq::hlt::packages
    include tdaq::hlt::eos

    package {
        ## needed by wish, ticket 1630
        ["tk"]: ensure=>present;
        ## PDF viewer, ticket 1664
        ["gv"]: ensure=>present;
    }
}
```

Class declaration

```
## Configure the smartd daemon
## from the smartctltools
class sysadmin::smartd ($type="") {
    if ($is_virtual=='true') {
        service {"smartd":ensure=>stopped,enable=>false}
    } else {
        pkg{"smartmontools":ensure=>present}
        # the new smartmontool rpm from sources has a db file
        # which is blocked by selinux - fix its context type
        file {
            "/usr/share/smartmontools/drivedb.h":
            selrole=>object_r,seltype=>etc_runtime_t,
            require=>Package["smartmontools"],notify=>Service["smartd"];
        }
        service {
            "smartd":ensure=>running,enable=>true,
            require=>Package["smartmontools"]
        }
        # check for HW raid, create config ?
        #file {"/dev/twa0":setype=>"fixed_disk_device_t"}
        $smarttype = $productname ? {
            "PowerEdge R410" => "^R410.$type",
            "PowerEdge R610" => "^R610.$type",
            "PowerEdge 2950" => "^PE2950.$type",
            default => "",
        }
        file {
            "/etc/smartd.conf":
            source=>[
            "puppet:///modules/site_$SITE/smartd/smartd.conf^$hostname",
            "puppet:///modules/site_$SITE/smartd/smartd.conf${smarttype}",
            "puppet:///modules/sysadmin/smartd.conf${smarttype}",
            "puppet:///modules/sysadmin/smartd.conf"
            ],
            notify=>Service["smartd"];
        }
    }
}
```