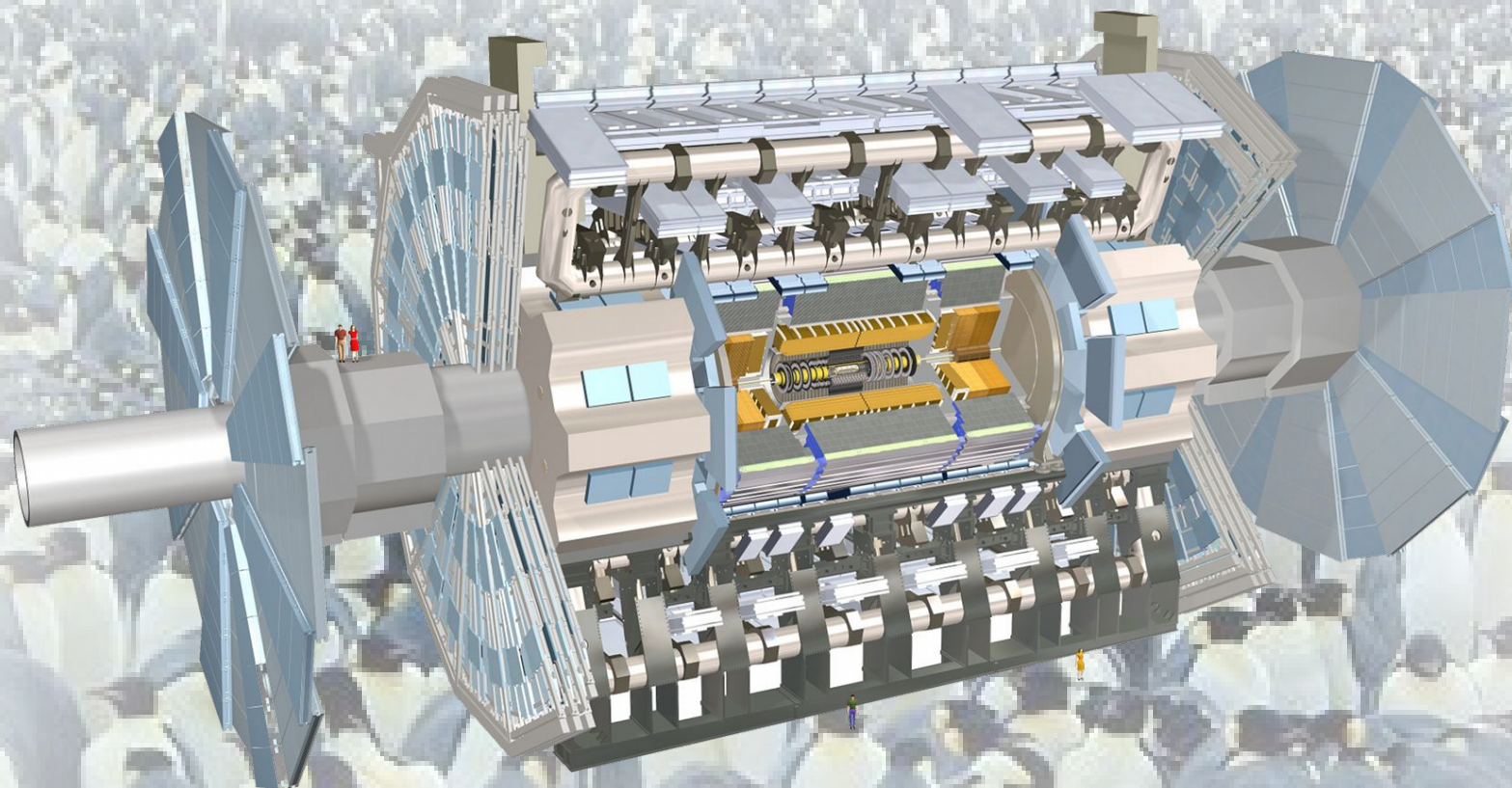
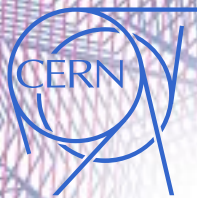


System administration work and opportunities in large Trigger and Data Acquisition systems *the ATLAS example*



Sergio Ballestrero - *University of Johannesburg, South Africa*
for the
ATLAS TDAQ SysAdmins team.



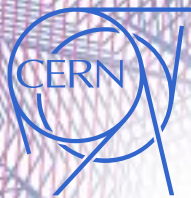
System Administrators' work



- Installing OS and software
- Configuring services
- Fixing hardware
- Creating user accounts
- Recovering files from backup...



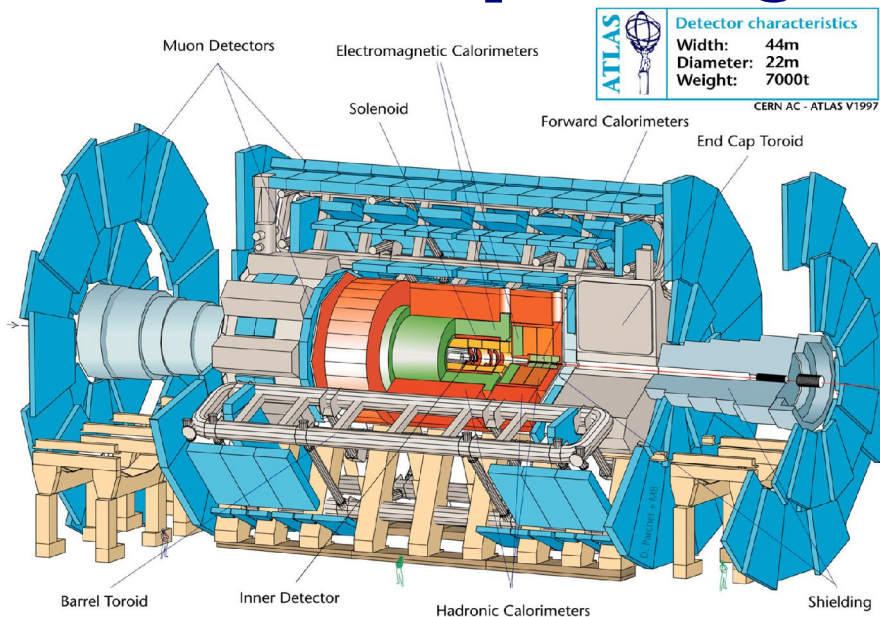
Do you really think it's just that?



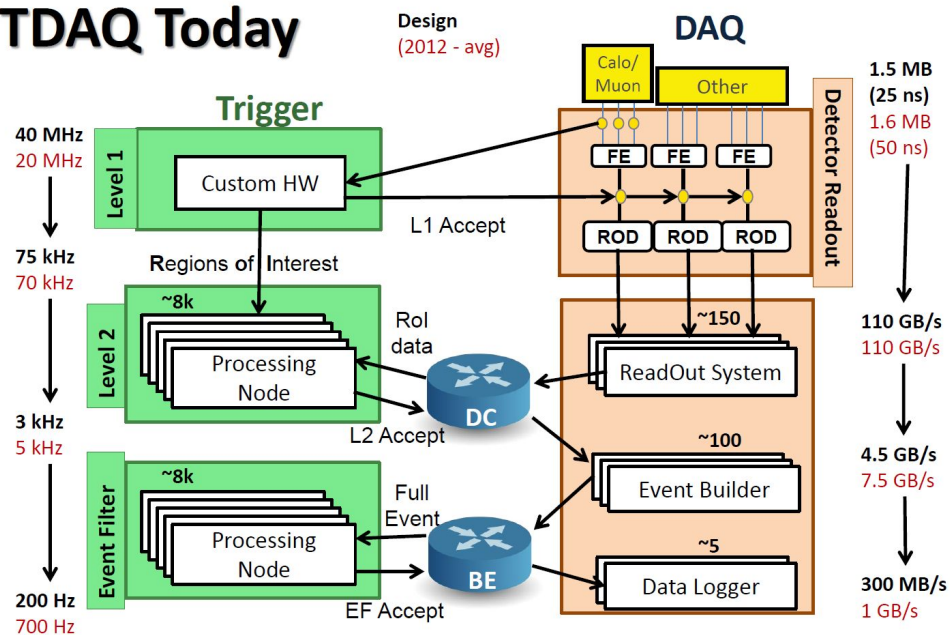
ATLAS Online Computing

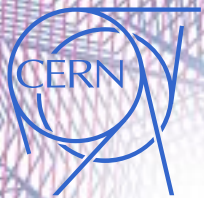


- Front-ends
- Readout
- Event filters
- Buffer storage
- Detector controls
- Infrastructure
- Control Rooms
- Status publishing
- Access control



TDAQ Today

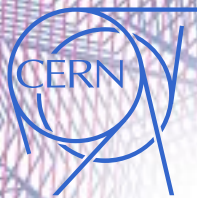




Many things - or not.



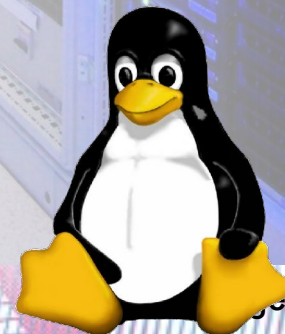
- Plenty of computers? Sure. But forget that.
- In the end, it's one big application, that happens to use one very custom piece of HW.
- That application is ATLAS Online. It consists of many pieces of custom code, different frameworks and goals, but it must work as one.
- That one custom HW is the whole of networks and computers near the ATLAS detector.
- If we just thought about one piece at a time, we could never make it work.



My computer is bigger...



- You probably know what it means to take care of one PC: installing OS, applications, configuring them, see it all works, remember to update, see it all works and the disk is not full...
- Now what if you have 10 PCs with 3 different configurations? "Remember" all the steps to make them the same..
- When you have 3000 with tens of different configs, that is not going to work.

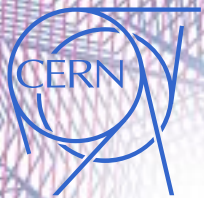


Control Freaks



- You need to make sure. Very sure. And then sure again.
- You need to do it every day, on thousands of (Linux) PCs.
- You need to do it perfectly, and the next person must be able to do exactly the same.

Doesn't that sound like the job for a computer?



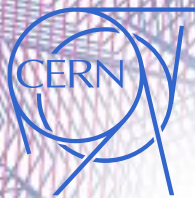
Puppet masters



- Turns out, softwares to do this exist. They are called **Central Configuration Management Systems**.
- They let you describe in detail the configuration of a PC, and take care of enforcing it systematically.

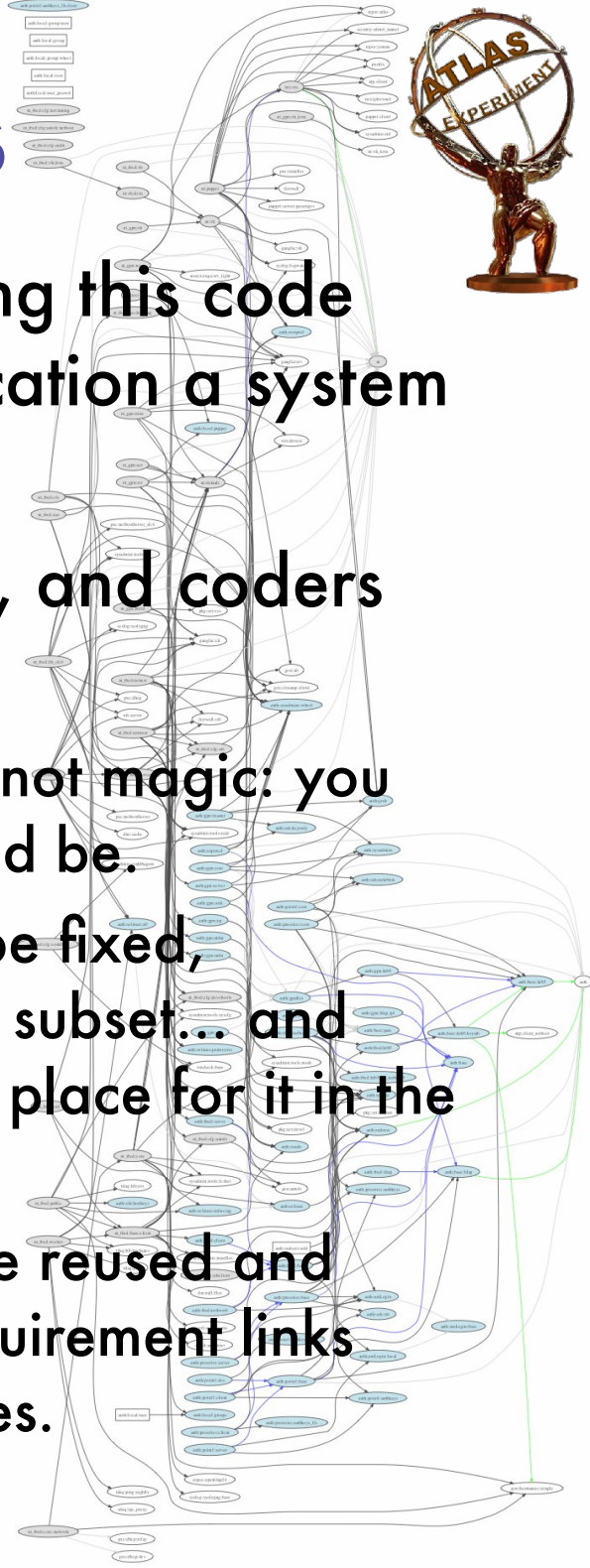


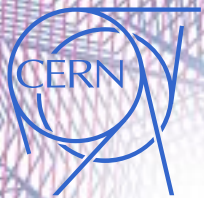
- These “descriptions” are themselves **code**.
 - Puppet uses a Domain Specific Language, that is (mostly) declarative and (kind of) object-oriented
 - ATLAS already has >15000 LOCs of Puppet code



Pulling the strings

- Our main job now is writing and maintaining this code that gives the one big ATLAS Online application a system it can run onto.
- We need admins who know (some) coding, and coders who know (some) system administration:
 - Puppet will put the configuration in place, but it's not magic: you still need to know what that configuration should be.
 - You also need to decide if the configuration will be fixed, parametric, applied to one host or all or which subset... and write your code accordingly, and find the right place for it in the class hierarchy.
 - And you need to design it so that the code can be reused and maintained, take care of making the actual requirement links explicit while avoiding a tangle of dependencies.



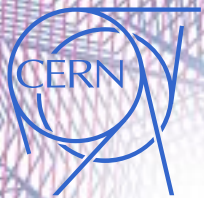


Look mom, no disk!



- In ATLAS we make extensive use of PCs with no operating system on disk, "netbooted" via PXE
 - Fastest way to provide single-image systems
 - Easy to switch between OS releases
 - Completely diskless systems have a few less points of failure (disk, controller, cables)
- It's not commonly done
 - so it requires ad-hoc development and support
 - The boot configuration is provided by our ConfDB
 - After boot, the single image is specialised using Puppet.

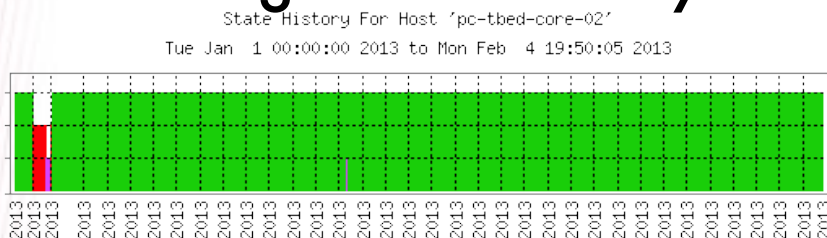




Obsessive Stalkers



- Say finally, everything works fine, now. And tomorrow?
 - Given a sufficient large time and population, everything that can possibly go wrong, will.
 - HW, systems and services need to be checked constantly, making sure that they're functional and healthy.



State Breakdowns:

Up : (97.493%)
Down : (1.647%)
Unreachable : (0.860%)
Indeterminate: (0.000%)



- Large systems need scalable monitoring and alerting tools, like Nagios, Icinga, Ganglia
 - These are good base tools but involve plenty of work on customization, integration, validation.

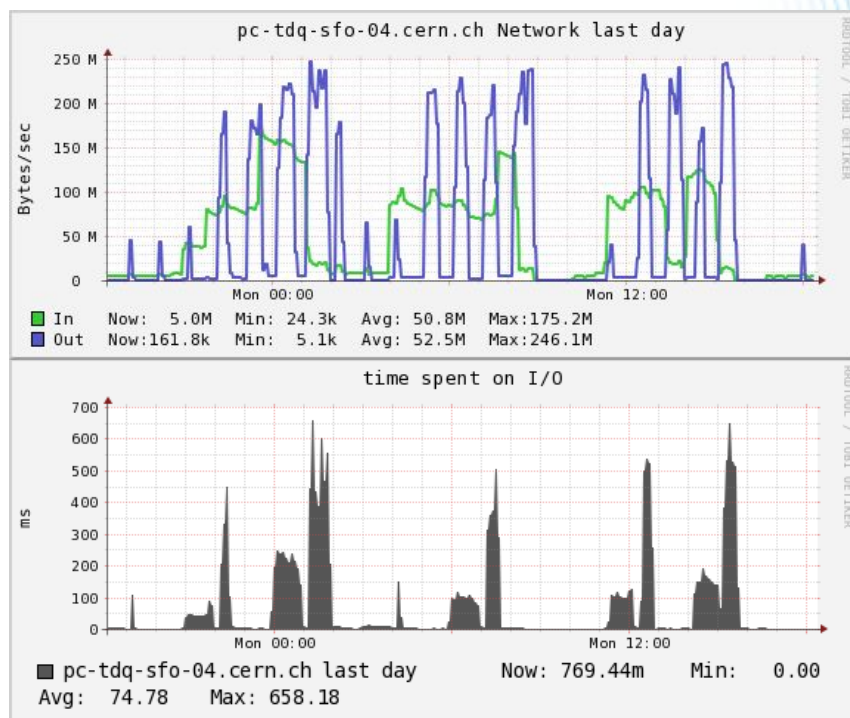
Performance maniacs

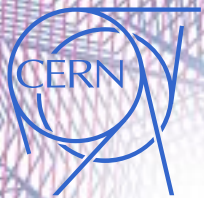


- Beyond making sure that a system works, we often need to know that it works well, and identify possible performance issues.



- Look for bottlenecks, scalability, design errors, guide HW choice
- Help in complex debug





Paranoid worriers

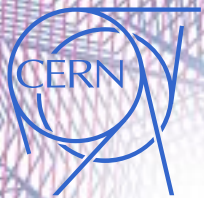


- When a pretty harmless hack touched CERN, it was all over the news.

We can't just think we're below the radar.

- Isolate the system, but keep remote access and usability; control incoming and outgoing, especially web sites
- Webservers, bastion/gateway hosts, role-based access control, short-lived authorizations, intrusion detection systems, security scanners, Single Sign-On authentication... plenty of fun!!

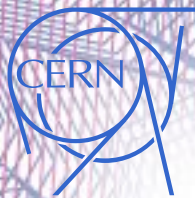




Looking ahead



- Technology changes, more variety - GPUs, many-core, ARM, which will give best performance/watt/space for which tasks?
- Distributed file systems, (maintainable) high-availability tools could allow drastic changes in the architecture
- Virtualization, cloud
 - from test and niche servers to full scale?
 - (ab)use the P1 computing power for other ATLAS tasks

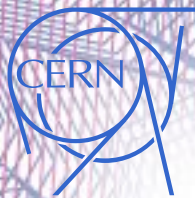


Opportunities @TDAQ SysAdm



- **Visiting students?**
possibly no ATLAS funding now, but we would like to start this kind of collaborations, especially for R&D
- **Short collaboration visits, up to 6 months**
support from ATLAS is possible
- **3-year Project Associates**
for Institute staff on leave of absence, with ATLAS support
- **other forms may be possible, what is most important for us is to find the right people**

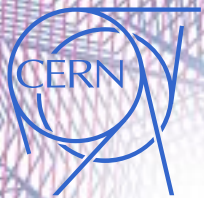
Contact: sergio.ballestrero@cern.ch



So why should you care?

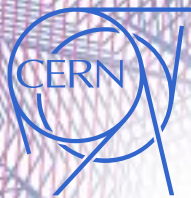


- you'll do some cool stuff
- you'll learn to play with big toys
- you'll get to pamper our beloved servers
- because ATLAS people actually care about the job we do - and that's rare.
- and because there's plenty of other things to learn and be excited about at CERN



Spares





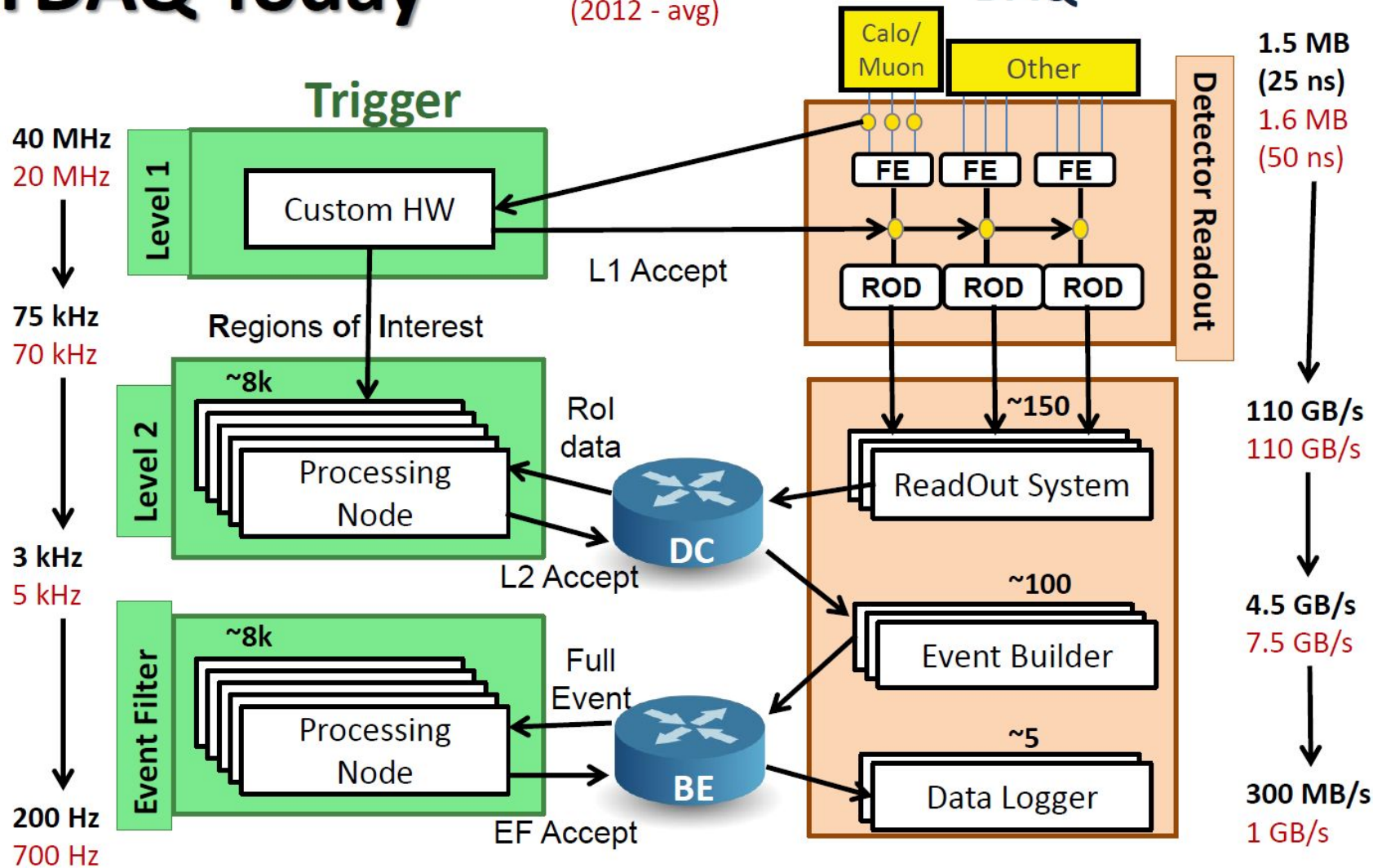
TDAQ Dataflow



TDAQ Today

Design
(2012 - avg)

DAQ



Drawing from N.Garelli

LocalBoot SLC5



Local Boot nodes

Provisioning by PXE + KickStart

- DHCP+PXE provided by an LFS, from ConfDB info
- Kickstart files generated by template-based system

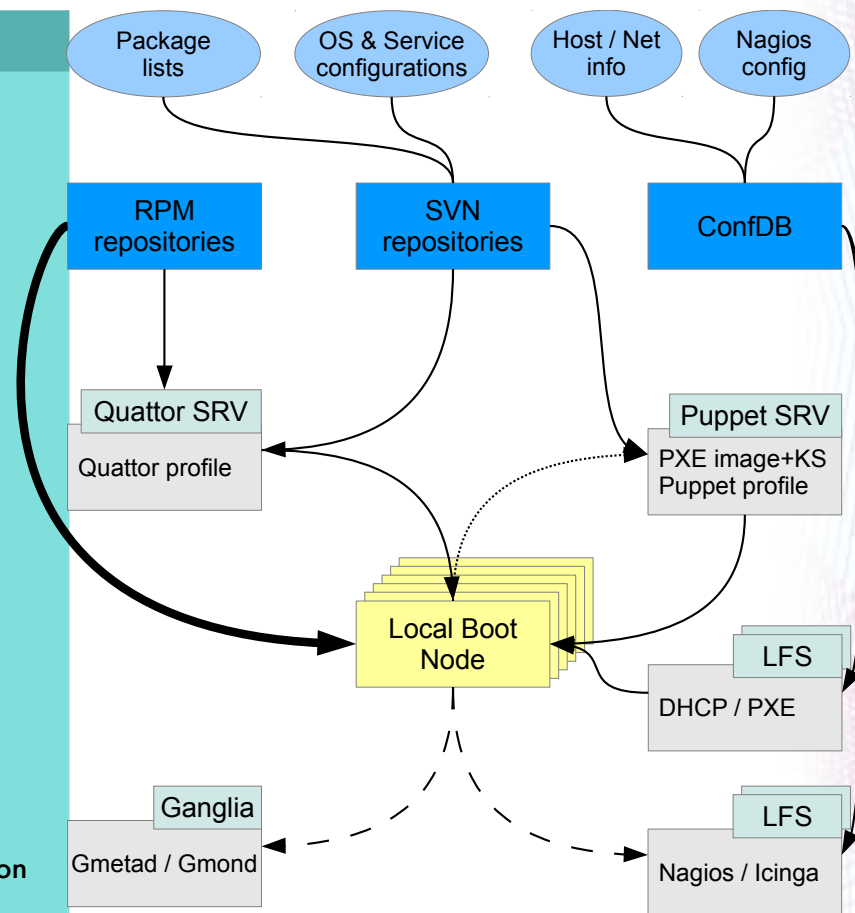
Quattor

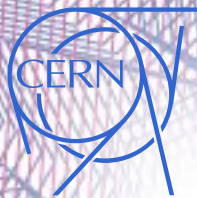
- CERN standard Configuration Management Tool
- Production system, managing 237 hosts in the Online Farm
- Tight control on installed packages
- Lack of flexibility for complex configuration/service dependencies
- Multiple languages for implementing modules

Puppet

- Widespread industry adoption, active development
- Full features, high flexibility
- Gentler learning curve
- Focus on consistency and idempotence
- In production, manages exclusively 25 complex servers and complements Quattor on the remaining 237
- Planned to completely replace Quattor on SLC6

For both Quattor and Puppet the configuration code is maintained in a Revision Control System (Subversion).





NetBoot SLC5



NetBooted Nodes

- ~2350 nodes boot the Scientific Linux CERN 5 OS via PXE
- ~80 Local File Server (LFS) hosts provide DHCP, PXE, TFTP for booting, /usr read-only directory via NFS.
- Configuration of DHCP, PXE and boot parameters provided by ConfDB, our CMT for NetBoot nodes which is described in a separate poster (Centralized configuration system for a large scale farm of network booted computers)

Boot With Me tool

- Generates PXE boot images (kernel + RAMdisk root) and /usr
- Uses a reference SLC5 VM image as source

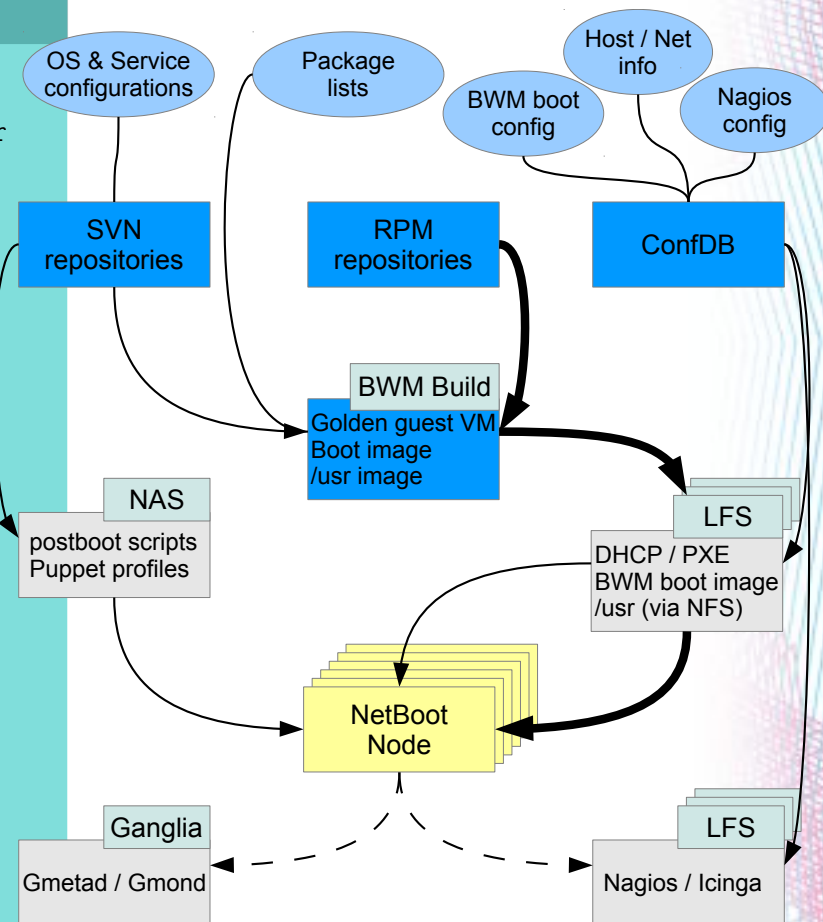
BWM post-boot script system

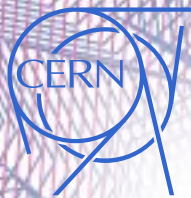
- Hierarchy of shell scripts, configures services, disk and NFS mounts etc
- Uses the standardised hostname to decide which sequence of scripts
- Stored on central Network-Attached Storage, executed by the client

BWM puppet

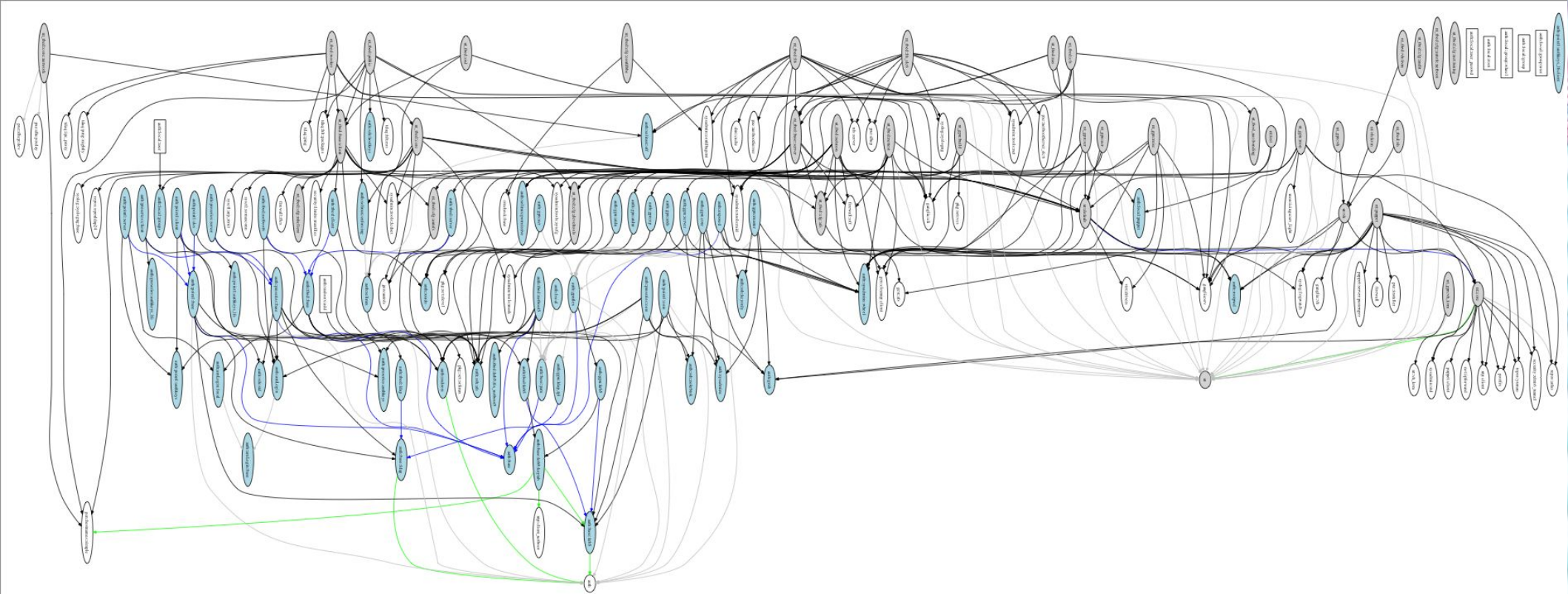
- Start to introduce Puppet profiles to replace BWM scripts
- Improve consistency and maintainability
- Serverless configuration, for scalability, using the NAS as storage

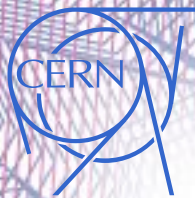
A Subversion repository is used to track changes of the BWM image creation configuration, of the post-boot scripts and Puppet profiles.





Puppet dependency-graph





Puppet code samples



Node "type" definition

```
## 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":}

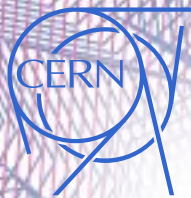
  ## shared, single ssh-host-key
  include auth::ssh::hostkeys

  ## 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;
  }
}
```

Service definition and configuration

```
## 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":settype=>"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"];
    }
  }
}
```

ConfDB GUI



ConfDB GUI ADMINISTRATIVE INTERFACE Cluster: All

Hostname: xpu-66 Host type: Clients Search + Advanced search Commit changes

Search results:

- pc-tdq-xpu-66001
- pc-tdq-xpu-66002
- pc-tdq-xpu-66003
- pc-tdq-xpu-66004
- pc-tdq-xpu-66005
- pc-tdq-xpu-66006
- pc-tdq-xpu-66007
- pc-tdq-xpu-66008
- pc-tdq-xpu-66009
- pc-tdq-xpu-66010
- pc-tdq-xpu-66011
- pc-tdq-xpu-66012
- pc-tdq-xpu-66013
- pc-tdq-xpu-66014
- pc-tdq-xpu-66015
- pc-tdq-xpu-66016
- pc-tdq-xpu-66017
- pc-tdq-xpu-66018
- pc-tdq-xpu-66019
- pc-tdq-xpu-66020
- pc-tdq-xpu-66021
- pc-tdq-xpu-66022
- pc-tdq-xpu-66023
- pc-tdq-xpu-66024
- pc-tdq-xpu-66025
- pc-tdq-xpu-66026
- pc-tdq-xpu-66027
- pc-tdq-xpu-66028
- pc-tdq-xpu-66029
- pc-tdq-xpu-66030
- pc-tdq-xpu-66031
- pc-tdq-xpu-66032
- pc-tdq-xpu-66033
- pc-tdq-xpu-66034
- pc-tdq-xpu-66035
- pc-tdq-xpu-66036
- pc-tdq-xpu-66037
- pc-tdq-xpu-66038
- pc-tdq-xpu-66039
- pc-tdq-xpu-66040

Hostname: pc-tdq-xpu-66015 LanDB Nagios Hardware DB

MACs: 00-26-6C-FA-C6-F0, 00-26-6C-FA-C6-F1, 00-26-6C-FA-C6-F3 Manufacturer: DELL Model: POWEREDGE C6100

Rack: Y.08-04.D1 [66] Position in Rack: U18 Building: 3178 Floor: 1W Room: 0804 Host Group: Point 1

Service Tag: OS Version: Net_SLC5_64 Description:

Ipmi Type: ipmi20 Nagios Server: pc-tdq-lfs-066 Config server: Choose...

BMC Specification: 37_1_1.26_2.0_20569_55 Include in DHCP relay list Don't include in DHCP Sync Nagios home directory Net Booted

PC Type: pc Netboot Server: pc-tdq-lfs-066

Boot parameters: root=/dev/ram0 ramdisk=131072 ip=dhcp selinux=0

NICs

- Type: control, Name: pc-tdq-xpu-66015, IP: 10.146.95.45, MAC: 00-26-6C-FA-C6-F0, Netmask: 255.255.255.0, Gateway: 10.146.95.1, Network domain: ATLAS
- Type: dc2, Name: pc-tdq-xpu-66015-dc2, Alias: pc-tdq-xpu-66015-ef2-vlan12, IP: 10.150.60.29, Netmask: 255.255.0.0, Gateway: 10.150.1.1, Network domain: ATLAS, Vlan ID: 12
- Type: ef2, Name: pc-tdq-xpu-66015-ef2, IP: 10.151.43.49, MAC: 00-26-6C-FA-C6-F1, Netmask: 255.255.255.0, Gateway: 10.151.43.1, Network domain: ATLAS
- Type: mgmt, Name: pc-tdq-xpu-66015-mgmt, IP: 10.146.95.44, MAC: 00-26-6C-FA-C6-F3, Netmask: 255.255.255.0, Gateway: 10.146.95.1, Network domain: ATLAS

Templates:

- BASIC-XPU
- INTERFACE_UP!*lo,ctrl0,ef2,vlan12*

Delete host(s) Update host info from LanDB Commit changes