The Architecture and Administration of the ATLAS Online Computing System

Presented by Marc Dobson, CERN
on behalf of
ATLAS TDAQ SysAdmin Team

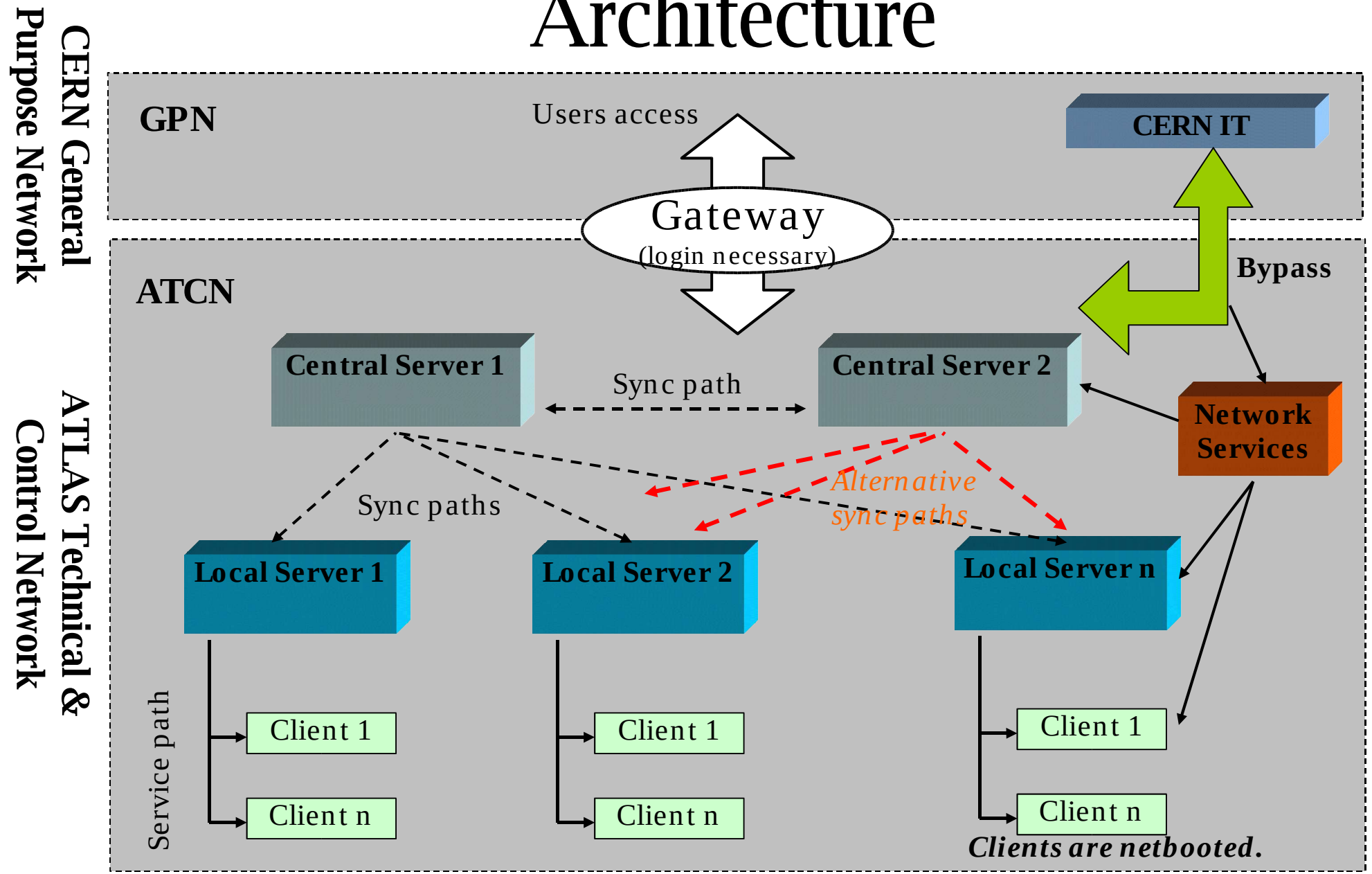
Content

- Introduction
- Architecture of system
- TDAQ pre-series system
- High availability/low maintenance
- Boot With Me, net booting
- Synchronization: requirements and results
- Authentication: Users/Roles and LDAP server
- Distributed File Systems
- Monitoring: Hosts and Network
- Summary

Introduction

- ATLAS requires a large on-site cluster of computers (> 2500)
- Needs to be administrated in a coherent and optimal way
- Reliability, robustness, and fast recovery/replacement are essential
- Control and Data taking system isolated from CERN network and only access is though an application gateway
- Data taking must run for up to 24 hours without outside connectivity (to Tier0 and IT)
- SW must be available to whole cluster: needs synchronization if located in different disks
- Separate control & data network, both require high bandwidth

ATLAS Experimental Area Architecture



Architecture: Details

- Bypass: used to connected to trusted services in IT, NTP, DNS, CASTOR (mass storage), Databases
- Central File Servers (CFS): two redundant and synchronized in real time (heartbeat) to hold master copies of SW and other items (e.g. OS for net boot)
- One Local File Server (LFS) per small number of clients which netboot
- Sync in hierarchy from CFS to LFS, then to local disk if needed
- Network services needed to mirror IT provided ones, due to requirement of 24 hour operation in isolation

Network design: see talk by S. Stancu, ID 289, CFN-6, Wed PM

Pre-series of final system
8 racks at Point-1 (10% of final dataflow)








One ROS rack
TC rack + horiz. Cooling
-
12 ROS
48 ROBINs

RoIB rack
TC rack + horiz. cooling
-
50% of RoIB

One Full L2 rack
TDAQ rack
-
30 HLT PCs

Partial Superv'r rack
TDAQ rack
-
3 HE PCs

One Switch rack
TDAQ rack
-
128-port GEth for L2+EB

Partial EFIO rack
TDAQ rack
-
10 HE PC (6 SFI - 2 SFO - 2 DFM)

Partial EF rack
TDAQ rack
-
12 HLT PCs

Partial ONLINE rack
TDAQ rack
-
4 HLT PC (monitoring)
2 LE PC (control)
2 Central File Servers

underground **: USA15**

surface**: SDX1**

- **ROS, L2, EFIO and EF racks:** one Local File Server, one or more Local Switches
- **Machine Park**: Dual Opteron and Xeon nodes, ROS nodes uniprocessor
- **OS issues:** Net booted and diskless nodes (local disks as scratch), running Scientific Linux, Cern version3.
- **Trigger** : Free trigger from L2SV or frequency manually set using LTP

Requirements: Low Maintenance, High Availability

- High availability very dependent on HW especially disk lifetime => get rid of disk and network boot the nodes

=> Boot With Me (BWM) project, corner stone of netboot

- Probably will need disk for TDAQ/detector analysis SW (nodes who require much disk access): local disk which can be replaced by network disk if it fails, i.e. not critical
- Requires servers for boot and disk serving:
  - Internal redundancy by redundant and hot-swap HW
  - External redundancy by allowing a client to interact with two or more servers and use network redundancy (link & PSU)
  - All servers must have synchronized copies of SW

Boot With Me (BWM): why?

- Need a system to build/configure netboot image, and configure the boot and post-boot stages
- Thousands of devices, organized according to function, need to be remotely operated. They cover a variety of HW with diversity increasing over the years (replacements, new machines), standard PCs and thin clients (single board computers, SBC)
- System must be flexible, easy to use and administer, low maintenance, minimal down time on an update, have a single point of definition, produce small image sizes (~30MB compared to full install of >=1GB)
- Most clients use PXE (Preboot eXecution Environment) boot ROM. Older ones use BOOTP.

BWM: Concepts

- Template: minimum specification for boot image (no services, only base OS. Can inherit from template.
- Project: main config specification which provides working image (services are configured properly, FS mounted, etc...) Project can be derived from single template or a project.
- Plugin: defines single functionality to be included in a project. Projects can include many plugins.
- The concepts hold definition of directory structure and list of files to be added to boot image (either from default OS installation or specific config)
- BWM config held in hierarchy of XML files. It does not depend on the reference OS: the same config can build images for different kernel versions.

- BWM config parsed by application, and populates a minimum Linux distribution based on the reference one + the specific configs

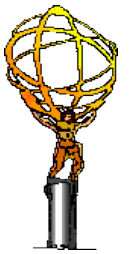
BWM: Boot Configuration

- To reduce number of boot images, some OS config is parametrized in image, and finalization is done in boot process with info provided by DHCP server (machine name & IP, kernel arguments, etc...)
- Enabling network is most critical part of system initialization, as there are usually several NICs (one for control, and one or more for data network), of potentially different physical types.
- Type of card identified by PCI ID, which is then matched to kernel module. Once loaded, more than one interface might be available.
- Decision algorithm implemented which can be controlled by kernel parameters
- System config files dependent on LFS, are configured at boot time (e.g. /etc/fstab for mount points)

BWM: Post-Boot Config

- After network config (& network drives are mounted) the client configuration finishes off with the post boot scripts in a shared repository
- Post boot scripts organized according to structured machine name:
  - Machine name convention: <device>-<det>-<function><sub-func/det-area>-<id>
    e.g. pc-tdaq-pub-01
- Post boot scripts executed from most generic to the most specific:
  e.g. pc, pc-tdaq, pc-tdaq-pub, pc-tdaq-pub-01
- Operations on the scope of TDAQ can be done in pc-tdaq files,
  e.g. mounting of TDAQ specific SW)
- Hardware specific things would be done in most specific file,
  e.g. pc-tdaq-pub-01
- Client (most) specific config files configurable by owner/user of machine

Synchronization: Requirements and Solutions



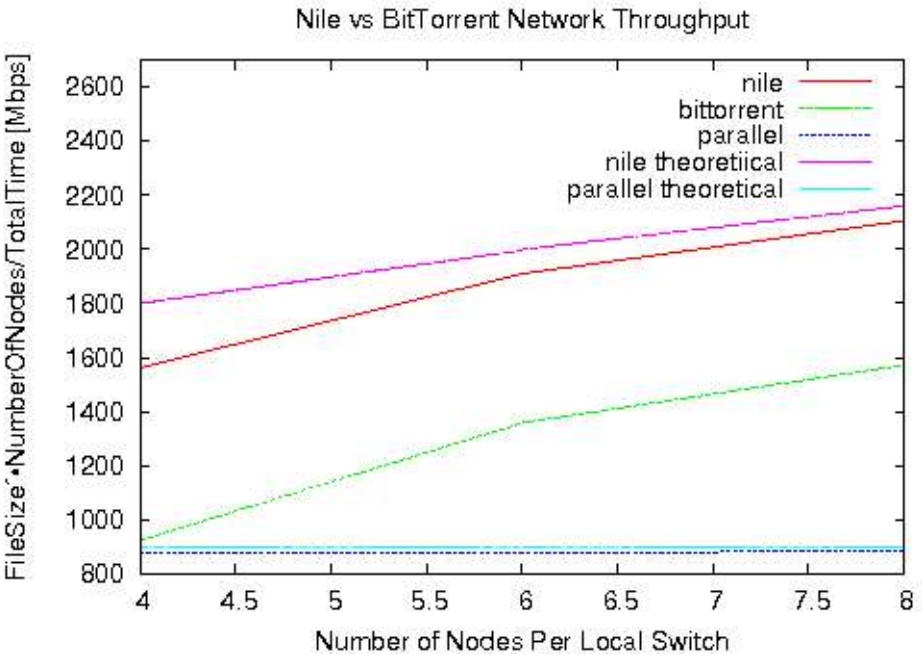
- Synchronization issue:
  - Multiple servers (150-200) are needed for the many TDAQ/detector clients (> 2500)
  - All servers hold same SW to export to clients
  - Clients with performance needs will have local disks for SW

=> So all servers and some client local disks need to be synchronized for the SW

- Possible sync mechanisms: standard rsync, configurable hierarchical sync (fixed routing), fully adaptive file sharing (adaptive routing)
- Sync studies started for deployment of TDAQ software on large cluster used for scalability tests (~600 nodes)

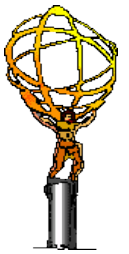
Synchronization: Results

- Two tools tried: P2P file sharing (adaptive routing), and worm like tool named Nile, calling SSH and rsync (which uses fixed routing)
- P2P is better for unknown networks (adapts to situation)
- Nile performs better for the hierarchical nature of the network topology in our experimental area system (symmetrical nature)
- Nile also has added functionality to do incremental synchronization due to underlying rsync



See talk by H. Garitaonandia, ID 200, STIS-3, Tue PM

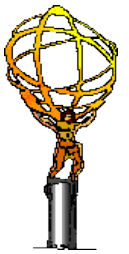
Authentication: User/Roles



- User based authentication wanted (not group based)

  => accountability and traceability
- Setting up a Role Based Access Control system:
  - user is able to belong to one or more roles
  - user can only be in one particular role at any time
  - that role allows the user to do certain tasks
- System is being setup for the control software with OS level support
- At the OS level, kerberos is being investigated for possible use

LDAP Server



- Independence from IT required for up to 24 hours:

  => own user/password DB
- System is standalone but in sync with IT AFS system for user names and IDs
- User authentication/authorization uses an LDAP server
- In addition to authentication info it holds:
  - Different login shells depending on computer
  - Groups
  - Later roles and their authorized tasks
  - User to role mapping

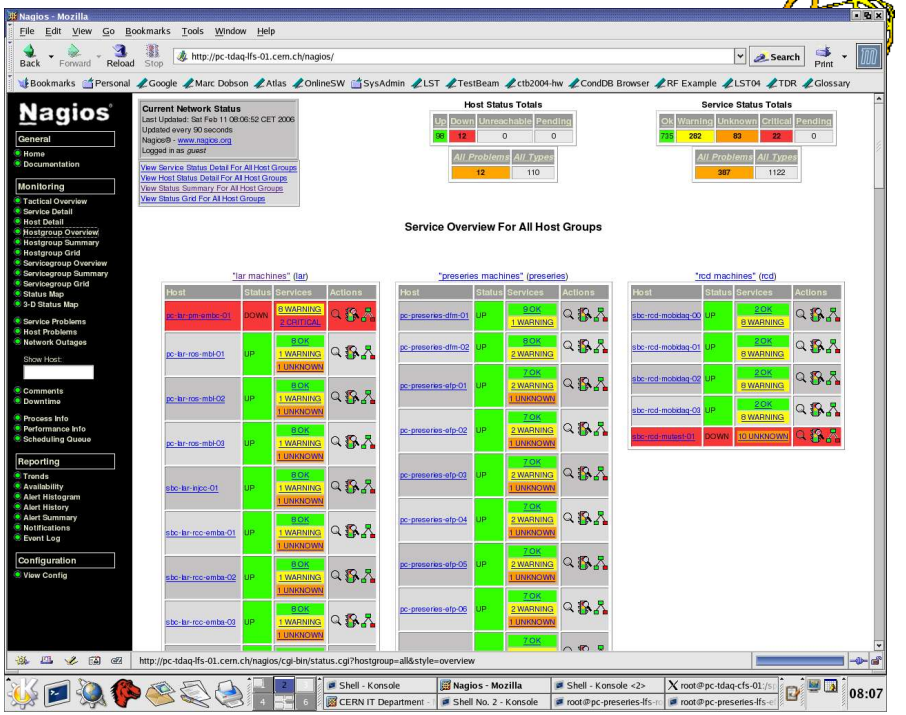
Distributed File Systems

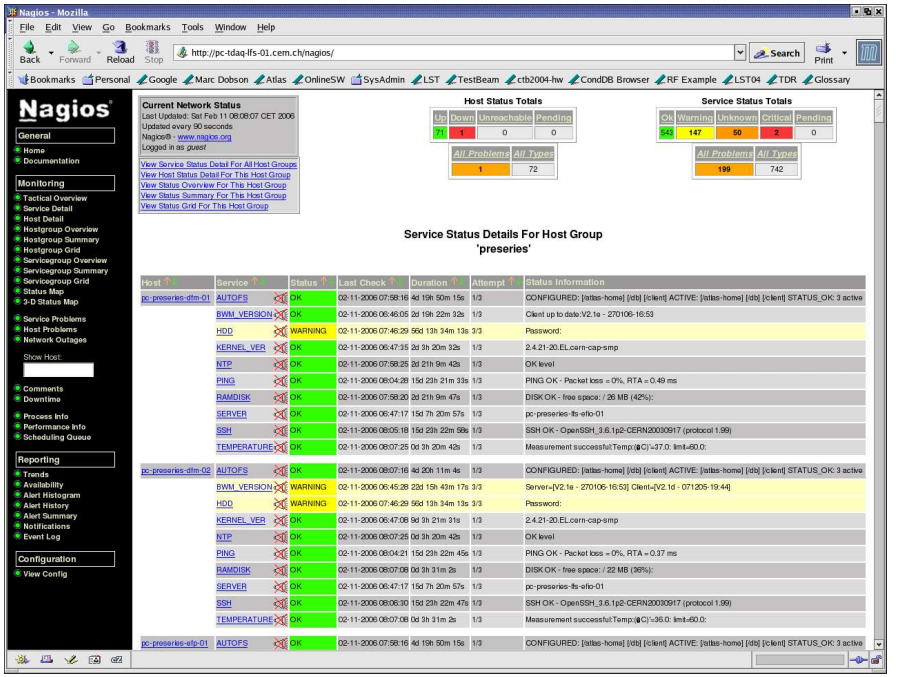
- Home directories required for users in the system:
  - Distributed FS, in Read/Write mode, over >2500 nodes
  - Few FS can do this: AFS, GFS, etc...
  - Using NFS to get started with: limitations appear with ~100 nodes
  - Investigations ongoing into alternate distributed file systems

- Need to distribute SW for the diskless nodes:
  - Distribution to limited set of nodes (up to 32) in Read Only mode
  - NFS chosen: standard, relatively performant (v3+), free
  - No performance problems seen with this system

Host Monitoring

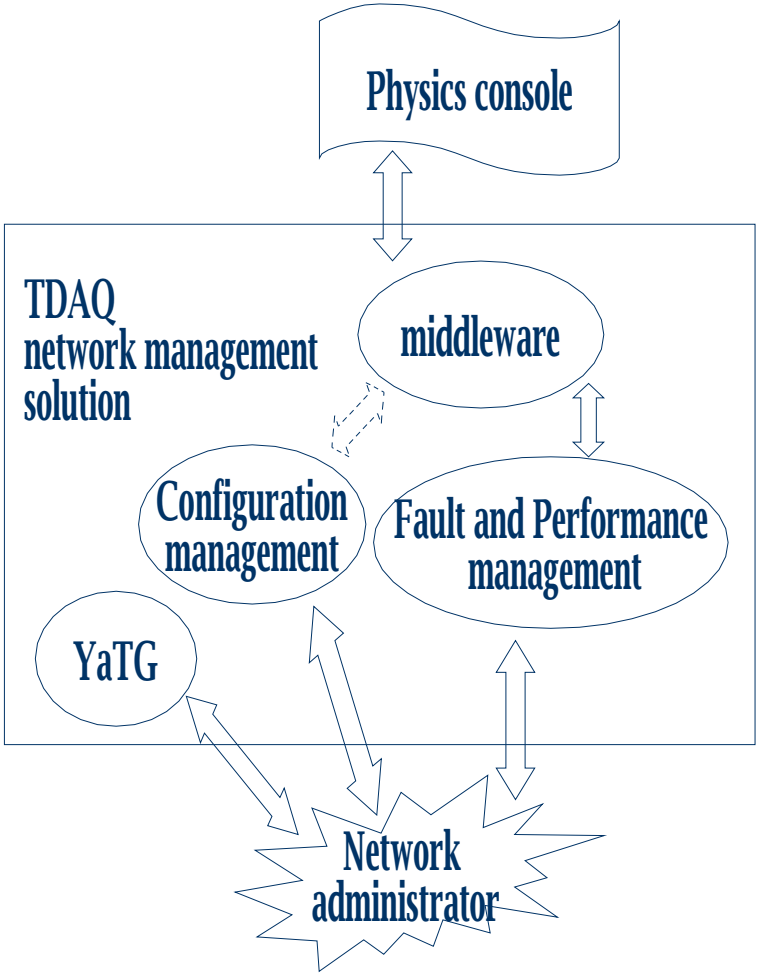
- Uses Nagios software
  - Basic services on all clients:
    - ping, SSH, ramdisk usage, NTP, kernel version, BWM version, LFS name, temperature, HDD state (if available), automount status
  - Advanced features: for special hosts (e.g. Gateways, LFS, network service machines)
    - NTPD, Number of users, state of exported FS, DHCPD
  - Email/SMS warnings sent for certain events (e.g. connection lost)





Network Config & Monitoring

- Monitoring uses commercial tool called Spectrum:
  - Used for Fault and Performance management (monitors at low rate, ~ 5 minutes)
  - Able to do Root Cause Analysis: finds the cause of the problem (very powerful feature)
- High Speed Monitoring: Yet another Traffic Grapher (YaTG) which interrogates switches using SNMP
- Configuration management: will use Rancid for the basic functionality



See talk by C. Meirosu, ID 41, OC-7, Thu PM

Summary

- Netbooted scheme, with client/server architecture has been demonstrated to work on small scale
- Inherent hierarchy of system ensures scalability to final ATLAS size (> 2500 nodes)
- BWM project has fulfilled its requirements and has proved very flexible
- Single area of known scalability problems: distributed RW File System, but solutions are available and need to be studied

System is being prepared for intensive use during the Cosmic Ray and Noise runs in March to July