Remote Management of nodes in the ATLAS Online Processing Farms

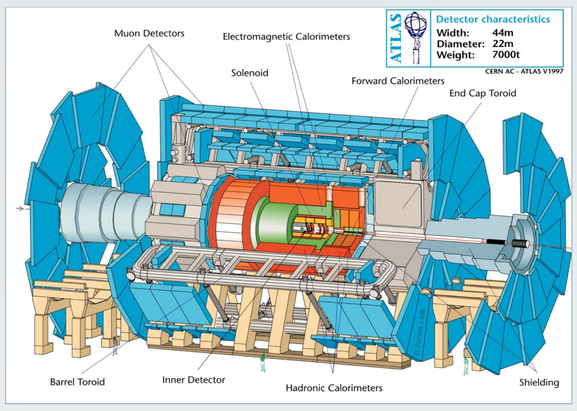
Marc Dobson*, Usman Ahmad Malik*#, Hegoi Garitaonandia Elejabarrieta+

CHEP'07, 2 – 7 Sept 2007, Victoria BC, Canada

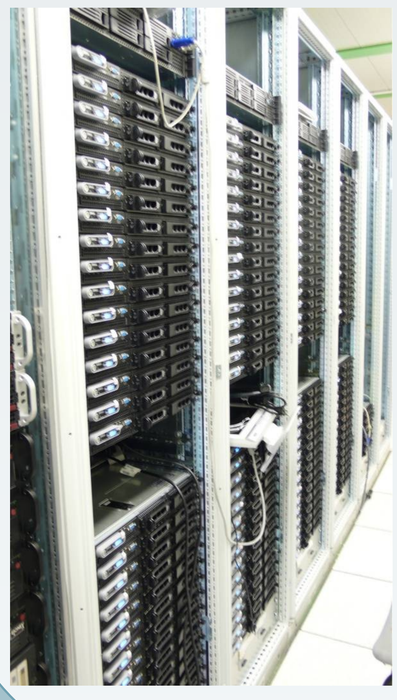
ATLAS Online Farms

Introduction

The ATLAS experiment consists of 3 levels of trigger, the first level being hardware based, and the second and third level (the High Level Trigger) which are software running on PCs. The HLT farms will be about 2500 nodes fed by 150 ReadOut System nodes. The online processing farms are made up of the HLT nodes, ROS nodes, and a number of other nodes for TDAQ processes and system administration services.



The administration of such a large cluster is a challenge especially due to high impact of any down time. The ability to quickly and remotely turn on/off machines, especially following a power cut, and the ability to monitor the hardware health whether the machine be on or off are some of the major issues which the ATLAS SysAdmin Team faced. To solve these problems ATLAS has decided wherever possible to use Intelligent Platform Management Interfaces (IPMI) for its nodes.



This poster will present the mechanisms which were developed to allow the distribution of management and monitoring commands to the cluster machines in parallel. These commands were run simultaneously on the prototype farm and on the small scale final farm already purchased. The commands and their distribution take into account the specificities of the different IPMI versions and implementations, and the network topology of the ATLAS Online system.
Results from timing measurements for the distribution of commands to many nodes will be shown. These measurements will cover the times for booting and for shutting down of the nodes and will be extrapolated to the final cluster size.

Online Processing Farms

All the nodes are network booted from their rack server (Local File Server) with Scientific Linux CERN 3 or 4 as Operating System.
There is a mixed environment for all the nodes in ATLAS: Uni-Processor, SMP, Rack Mounted, Desktop, Single Board Computers in VME crates.
All rack mounted nodes and some desktop nodes are equipped with IPMI.

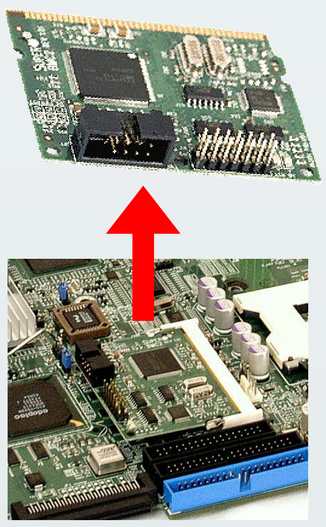
Hardware Management

Problem

The hardware management and monitoring of PCs is a general problem but with no generic tool set or standardization. Most monitoring tools offer the possibility to monitor (only monitor) the motherboard sensors, and their interface is now standardized. However this is limited to the motherboard, and not the upstream components such as power supplies, chassis fans.
The ability to control the hardware is not standardized or widely available in monitoring tools. A pre-curser of hardware control is Wake-on-Lan developed and supported by IBM and Intel. Other technologies are available for large clusters often requiring extra cards and cables. However they still cannot interact with the BIOS or when the machine is powered off.

Intelligent Platform Management Interface – IPMI

The IPMI specification was designed to meet these shortcomings and was produced by a consortium (DELL, HP, Intel, and NEC), with IPMI v1.0 appearing in 1998, v1.5 in 2001 and v2.0 in 2004.
The IPMI specification defines a set of common interfaces to computer hardware and firmware for monitoring system health and managing the system. IPMI operates independently from the Operating System (OS) and allows administrators to manage a system remotely even in the absence of the OS or if the system is not powered on. IPMI also works when the OS is running and offers all the monitoring and management facilities.
IPMI consists on a main controller called the Baseboard Management Controller (BMC, see picture) and other satellite controllers which communicate via a special BUS.
IPMI only gives the structure and format of the interfaces as a standard, and the implementation may vary from manufacturer to manufacturer.
IPMI v1.5 allows basic power on/off functionality, v2.0 allows much more (such as console redirection).
Some IPMI implementations share the onboard NIC with the OS and some have a dedicated management connection.

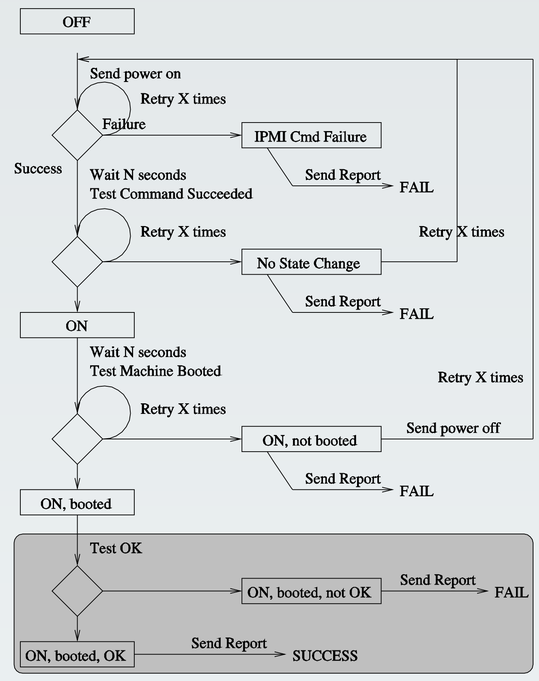


IPMI Software Tools

Thanks to an open specification, the IPMI open software tools have developed rapidly. Since IPMI v2.0 the manufacturer implementations have converged, and the tools interact with most them. ATLAS decided to use ipmitool which gave the most uniformity across implementations and versions.
The tools allow interaction with individual nodes. One important requirement is to be able to address many nodes at the same time. For this task Nile [] was chosen. It is developed in ATLAS for the distribution of commands in parallel to many nodes, and is based on worm technology. It can take into account the network topology and it collects all output from the clients.

FSM

Our first experience with the prototype system, has revealed that the basic commands offered by IPMI need some consolidation in order to cope with unforeseen circumstances in booting or shutting down nodes. Here are two examples to illustrate this:
All nodes are network booted by design and it is possible that the first DHCP request or TFTP request fails and consequently the boot also (eg congestion or server overload)
IPMI commands were indicated as executed but have been unsuccessful.
To achieve resilience in booting and shutting down a node, a simple state machine was designed to allow the appropriate checking, waiting and retry facilities. The state diagram for booting is illustrated in the figure. The part in grey has not been implemented yet.



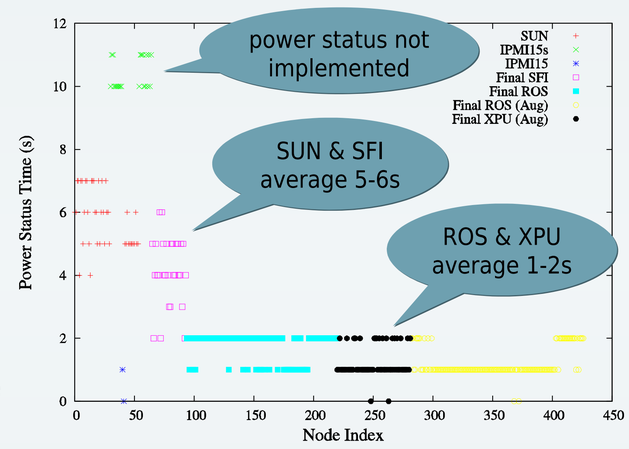
Results

All Online Farm PCs are network booted and served from a Local File Server usually located in the same or a nearby rack. This server is also used by Nile to distribute the commands to the local clients. The load and time of executing the commands are therefore distributed.

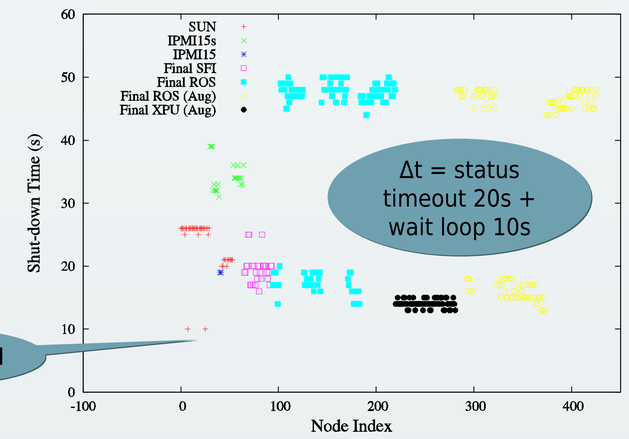
The measurements were done with a variety of hardware, from the pre-series system and the final system machines. The hardware was 42 SUN (IPMI over SSH), 11 IPMI1.5s implementation (supermicro implementation of 1.5), 2 IPMI 1.5 implementation, 126 IPMI 2.0 final ROS nodes, 29 IPMI 2.0 final SFI nodes (Event Builder), and 60 IPMI 2.0 final XPU (HLT processors).

Note the second ROS measurements were repeated at a later date to validate the XPU measurements (only done later).
Three types of measurement were done: getting the status of power on the system (on/off), shutting down (cut power), and booting the machines. The measurement precision is 1s.

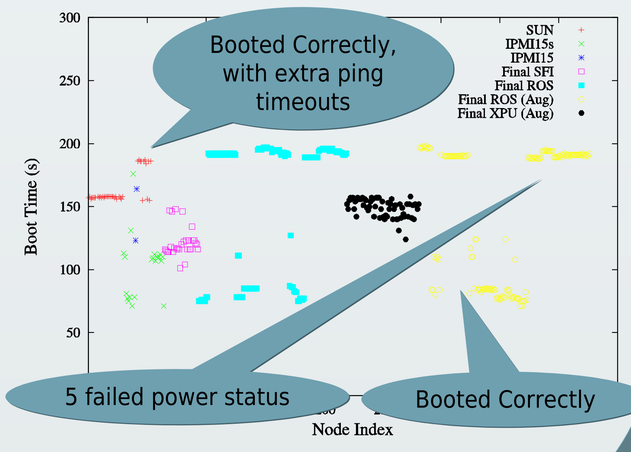
The first figure shows the timing for the power status. All PCs of the same type are grouped. IPMI 1.5s does not have status command. It is inferred from sensor information, thus longer time to get the status. For SUN longer time due to IPMI implementation over SSH.



The second figure shows the timing for turning the power off (execute power off and then check with status until it is off). Depending on implementation average is 20-35s consistent with time to send command, wait, and get the status. The two groups in the ROS section are due to a retry to get the status for some of them (additional time for wait – 10s – and timeout on status – 20s). Extra loop due to booting & IPMI from LFS in different rack.



The third figure shows the timing for booting the nodes (using the FSM defined above).
The FSM powers on a node, checks its status is on, then checks the OS is up with a network ping (the ping is in a loop of 10 tries with 30s wait between tries).
For all measurements except ROS nodes, there are sometimes different groups spaced 30-40s apart). This correspond to a normal boot, with more or less ping loops before the OS is ready (30s wait + 0 or 10s for failed ping).
For the ROS nodes, there are 2 groups, one around 80s and another around 195s. The first corresponds to a standard boot with one failed ping (40s). The second group corresponds to five power status failures before a successful first time ping (80 – 40 + (5 x 30) = 190s).



Outlook & Conclusions

In the light of the above results, the various loops and timeouts in the state machine need to be optimized. Furthermore the implementation of the state machine needs to be completed, with appropriate tests to check a machine is booted and usable. This level of check may also be taken care of by the Nagios monitoring system. It will also be necessary to extend this scheme also to the LFS machines which need to be brought up before the clients in the case of a power cut. Lastly the aim is to integrate this into some graphical interface for ease of use.

The poster has shown the different tools used for the remote hardware management of the nodes in the ATLAS Online Processing Farms and their robustness to different problems occurring during the booting of those nodes.

Author Information

*CERN (European Organization for Nuclear Research), Geneva, Switzerland
#National Center for Physics (NCP), Islamabad, Pakistan
+Instituto de Fisica de Altas Energias (IFAE), Barcelona, Spain