Remote power and console management in large datacenters

A Horváth
IT department, CERN, CH-1211 Genève 23, Switzerland
E-mail: Andras.Horvath@cern.ch

Abstract. Today's datacenters are often built of a large number of commodity hardware components. A number of common tasks including operating system installation, hardware and OS–level troubleshooting require low-level control of the machines' console and power buttons, yet local physical access is not only impractical and error–prone but often impossible. We report on the approach used at CERN for low—level remote management of more than 9000 machines in multiple datacenters.

CERN-IT-2010-006
11/11/2010

1. Background

As of 2010, CERN IT operates four machine rooms, one of which is off–site. Commodity PC hardware purchased from several vendors is used, among other tasks, to provide the necessary local disk and CPU capacity as well as to run services for the LHC Computing Grid. With a large number of machines installed and a limited number of system administrators available we have to rely on remote access and automation as much as possible both in everyday activities and in emergency situations. While local machine access is still possible, it is often cumbersome, error–prone and slow.

New hardware is purchased via competitive tendering and special effort is made to avoid any "vendor lock–in", so the solution to the remote management problem needed to stay vendor–independent and open as well. Other constraints were total cost of ownership and the manpower investment needed to deploy the system.

2. Goals

We set the following goals for the infrastructure:

- ability to turn individual machines on and off, including inducing an orderly OS shutdown if possible
- ability to query individual machines' power status
- ability to connect to individual machines' console, including OS console and BIOS
- log all console traffic
- scale up to multiple tens of thousands of nodes
- strict access control
- the above functionality must be accessible both by humans and by software (e.g. provide means for automation)

- encrypted authentication and console traffic
- robustness; graceful degradation and easy recovery
- minimal TCO
- full integration into the ELFMS[1] toolsuite
- efficient, minimal user interface

All of these goals were achieved (see also the FAQ at section 6.)

3. Use cases

3.1. Use cases for remote console

- Recovery from configuration or driver errors (i.e. no networking available in the OS)
- Manual boot (for instance to correct filesystem errors)
- Console log contains kernel backtraces, useful for debugging
- Changing BIOS settings where no other tool is available
- Connecting to disk arrays and other 'embedded' devices that are on a private LAN; remote mass reconfiguration via scripts

3.2. Use cases for remote power control

- Mass–reboot machines (mass reinstallation etc)
- Orderly emergency shutdown via ACPI[2] signal to OS
- Mass restart, in the proper order after a power outage or emergency shutdown
- Reset unresponsive machines
- (not yet used at CERN) on–demand hardware provisioning — to be explored, especially together with virtualization

4. Solution

4.1. Underlying technology

4.1.1. Server hardware Our choice of base technology, for both remote console and power control, fell to IPMI[3] because of its general availability, vendor independence and good tool support.

An IPMI Baseboard Management Controller (BMC) is usually incorporated in, or can be purchased for very little extra, for most current commodity PC mainboards. The BMC is a hardware device, independent from the main CPU, that can be used to monitor certain aspects of the host machine, control the main power supply as well as to access a 'virtual serial port' presented to the OS and BIOS. In IPMI version 2.0 with RMCP+ support, both authentication and network traffic to the BMC can be encrypted. IPMI is a vendor–independent de–facto standard implemented by most hardware manufacturers; therefore requiring a working remote IPMI 2.0 connection in all new procurements (including RMCP+ and Serial–Over–LAN functionality) does not impose significant restrictions on available hardware. On the software side, free open source tools as well as vendor–specific tools are available for manipulating IPMI BMC settings.

Most BMCs can "share" the network connection of the motherboard NIC. This feature combined with encrypted network traffic means that our general–purpose campus network can

[1] Extremely Large Fabric management system, http://elfms.web.cern.ch/elfms/

[2] The Advanced Configuration & Power Interface, see http://acpi.info

[3] Intelligent Platform Management Interface, see http://en.wikipedia.org/wiki/Intelligent_Platform_Management_Interface

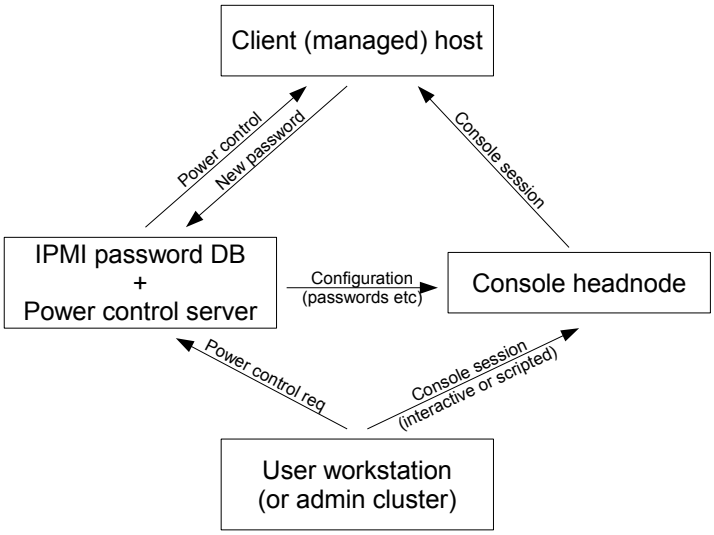

Figure 1. Communication diagram of the system components

also be used for remote management of the machines, thus removing the need for cabling a management LAN. (Caveat: see also section 5)

In our environment, all IPMI BMCs are configured locally from the running operating system at install time via the Quattor[4] framework. Individual, randomized passwords are generated for each BMC and uploaded in a secure manner to a central IPMI password database. The password database is never exposed to the users directly.

4.1.2. Virtual machines and embedded systems Virtual machines are being integrated into the same infrastructure via the API or CLI of the hypervisors.

Disk arrays, SAN switches and other equipment that generally does not possess IPMI capabilities can usually be managed over the network via a CLI, telnet or ssh. Since these are not considered highly secure devices, their access is limited to a physically separated private LAN. These private LANs are accessible from a console headnode (see below) used as gateway, and consoles can be accessed transparently via the console system. The low–level connection in these cases is usually telnet, but adding any character–based protocol to our implementation is trivial.

4.2. Headnodes

As of 2010, we install a special "headnode", an additional Linux PC for every 500 machines ("client hosts"). This machine is dedicated to the purpose of providing console access, storing console logs and monitoring the console connection. Another function fulfilled by the headnodes, though the details are beyond the scope of this paper, is to provide access and monitoring of embedded on private LAN.

Via the Quattor system, each headnode can have access to the IPMI passwords of its clients from the password database. These are stored locally and — via a wrapper — are used by

[4] http://www.quattor.org/, part of the ELFMS toolsuite

an open–source console application (`conserver`[5]) to establish a permanent connection to each BMC over the network. Addition or removal of a client, as well as a BMC password change or host ACL change, is not disruptive to the running system, i.e. it can be reconfigured online.

Console connections are retried and reestablished in case the connection is lost. Users connect to `conserver` via its CLI client (over SSL) and they are authenticated by their Kerberos5 token. Authorization is done by reusing an existing access control list in Quattor: only those who have root access to a machine are permitted to connect to its console. This eliminates the need of maintaining a separate ACL and leverages the existing system of user groups.

The remote console system gives interactive access to the "serial" console, logs all traffic (even if there's no one connected — very useful for capturing kernel backtraces!), and can play back these logs in an interactive session.

4.3. Remote power control

An IPMI BMC can control a server's power supply as long as the server is plugged into power, even if the server itself is turned off. The BMC (and therefore our system) can turn machines on or off, induce an ACPI event that most operating systems (including Linux and Windows) interpret as shutdown, or query the current power status.

At the moment, remote power control is run through a central server, although it would be possible to use each headnode for the power control of the respective clients. The power control server is stateless and can be restarted without affecting functionality.

Users connect to the server via HTTPS (using a command line interface, not a webpage) and are authenticated again by their Kerberos5 token. The same access control mechanism is used as for consoles: those with root access to the node in Quattor have remote control over its power supply.) The server will run the respective commands on the users' behalf; a fixed–size thread pool enables quick operation on many clients at once even if some of them are unaccessible without overwhelming the server.

The same basic infrastructure, but with a special application, is in place for emergency shutdown of a large number of machines in case of a cooling system issue in a machine room.

4.4. Client configuration

IPMI Serial–Over–LAN (SOL) presents a 'virtual serial port' to the operating system; this needs to be configured in Linux as the console device, with the same baud rate as is set up in the BMC. This is done via a Quattor NCM component automatically. Another Quattor component takes care of generating BMC passwords and IP/MAC address, enabling SOL, and applying hardware–specific settings. BIOSes can usually be set up to redirect (copy) all BIOS input/output to this serial port as well, thereby making remote BIOS access possible.

4.5. IPMI password database

A very simple SQLite[6] database keeps track of two passwords for each BMC, one with OPERATOR privileges (to access the console) and another with ADMINISTRATOR privileges (to control the power supply.)[7] In addition, the BMC IP address is also copied here so that all information needed to access a BMC can be queried with a single database lookup.

Passwords are uploaded (or changed) by the `ncm-ipmi` Quattor component when a new machine is installed, via HTTPS, authenticated both ways via X.509 certificates. Each machine is only allowed to write its own passwords into the database.

[5] http://www.conserver.com/

[6] A simple relational database without a server. http://www.sqlite.org/

[7] In practice, many BMCs allow OPERATORs to access power controls as well.

Console headnodes can download the passwords of their own clients only (again via two–way X.509–authenticated HTTPS) whereas the remote power control server has local access to the database and runs the power control commands on the users' behalf.

Should the database be lost, or passwords become compromised, it can be re–generated by re–running `ncm-ipmi` on the respective machines.

4.6. Software used

In addition to the ELFMS/Quattor toolsuite, the open source `conserver` and `ipmitool`[8] suites are used to provide a console access framework and, respectively, to configure and access BMCs; the remote power control service code was developed in–house. A wrapper script was written to incorporate `ipmitool` in `conserver` without exposing passwords in the latter's configuration file (which can be read by clients) and some packaging changes were also introduced to fit our environment. The same wrapper script takes care of retrying the IPMI SOL connection with some known workarounds should the connection fail. It can also access consoles via other protocols like telnet (see section 4.1.2), and more can be added easily.

4.7. Operation of a large–scale system

New machines are assigned a headnode before they're first installed, the appropriate fields in Quattor and other databases are filled in, and automatic installation takes care of client configuration. Headnodes are also automatically (and transparently) updated from the Quattor configuration. Similarly, new headnodes can be installed in an automated manner.

Management tools are available for moving IPMI clients from one headnode to another; while this severs the console connection for a short while it is almost transparent to the users. This is important for retiring headnodes or, for example, the migration of client machines to a new physical location with a separate network.

A separate background process takes the connection information between headnode and clients from Quattor and produces DNS entries similar to
`<hostname>.headnode.cern.ch` which are used by users to connect to consoles. This way, console access depends only on the DNS service, the Kerberos domain controller (KDC) and the given headnode. The first two are essential to any other service at CERN and thus are highly available, and the headnode can be changed easily if needed, as described above. Similarly, remote power management depends only on DNS, the Kerberos KDC and the central power management server which also hosts the IPMI password database. The fact that our system only depends on a minimal number of low-complexity services is important as both console and power control are likely to be accessed in an emergency recovery situation.

5. Issues, known limitations and considerations

While technically feasible, in the current Quattor setup it is not possible for headnodes to become clients of each other. Thus, remote console access to headnodes is limited (but remote power control works). A new Quattor schema is proposed to lift this limitation but its implementation will take time and effort.

Most modern BMCs offer several additional services such as remote Keyboard–Video–Mouse (KVM) access (usually over HTTP or VNC protocols), SSH access to the BMC, remote export of USB devices etc. If these BMCs are "exposed" on a public LAN — like at CERN — one needs to consider the security implications of allowing so many services, which sometimes have default credentials, to be accessible to the users on the LAN. Disabling such services or closing the network port on the BMC usually requires a special firmware version from the manufacturer.

[8] http://ipmitool.sourceforge.net/

While IPMI BMCs can almost always be accessed locally on any new hardware, sometimes it takes long debugging and even firmware upgrades to get remote access (and SOL) working. Some BMCs prefer the same MAC address as the integrated NIC, some have and need their own; ARP replies and SOL must be enabled; BMC users must be enabled to connect and to use SOL; etc. Firmware on BMC, motherboard and (where applicable) enclosures needs to be at the right revision level, and firmware revision strings do not always follow 'standard' version numbering rules. We have established good working relationships with our suppliers as well as with the motherboard/BMC manufacturers directly and we are prepared to work together in getting BMCs to work. Fortunately, this work is mostly successful; with the right firmware and settings most modern BMCs can be got to work.

Although `ipmitool` is available in most Linux distributions, "enterprise" Linux variants generally carry very old versions and need upgrading, otherwise BMC access may not work.

Machines that don't have a text-based console (i.e. those running Windows) can not benefit from a serial console system. Although functionality could be added to access the KVM built into most BMCs for interactive sessions the traditional concept of a 'console log' does not apply to these operating systems. Fortunately, the majority of our systems run Linux. (Remote power control, including ACPI OS shutdown, works fine on Windows.)

6. Frequently Asked Questions

Is the system described here usable without ELFMS/Quattor?

Certainly! The automated configuration and access control will of course be missing, but by manually configuring clients and headnodes the console system will work fine. Manually issuing power commands (running `ipmitool`) is also possible.

Why don't you use KVM-Over-IP switches?

KVM switches provide console output as an image, so logging and automated scripting is not possible. Moreover, the cost of IP-enabled KVM switch is not negligible.

Why don't you use remote controlled power distributors?

Mainly for cost reasons, but note also that these devices can't send an ACPI signal for an orderly OS shutdown. If need be, however, any device with an API or CLI can be integrated into our system with relative ease.

7. References

- Console Service documentation:
  https://twiki.cern.ch/twiki/bin/view/FIOgroup/TsiConsole
- Remote Power Control Service documentation:
  https://twiki.cern.ch/twiki/bin/view/ITProc/KbRemotePowerControl
- Conserver webpage: http://www.conserver.com/
- IPMItool webpage: http://ipmitool.sourceforge.net/