Example xCAT installation on an iDataplex configuration

08/24/10, 04:26:49 PM

Table of Contents

1. Introduction and description of example configuration

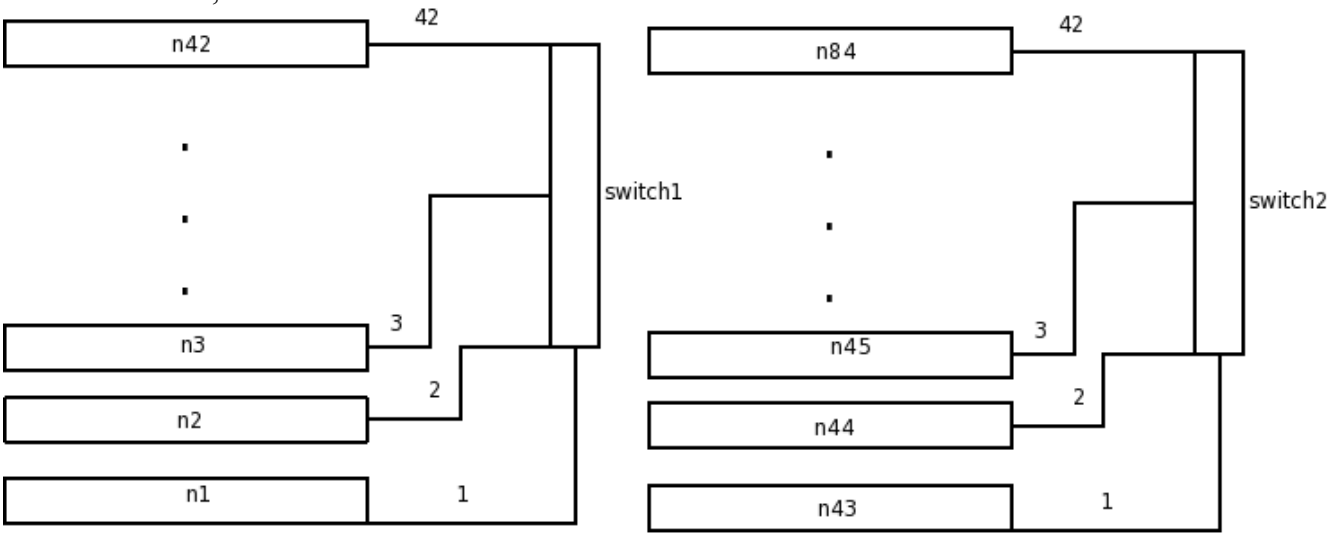
This example configuration is intended as an introduction to xCAT. It will assume the use of IBM e1350 defaults as documented at ftp://ftp.software.ibm.com/eserver/xseries/1350FS_0507.pdf.

This configuration will have a single dx340 management server with 167 other dx340 servers as nodes. The OS deployed will be RH Enterprise Linux 5.1, x86_64 edition. Here is a diagram of the racks:

Rack 1						Rack 2					
A	B		C	D		A	B		C	D	
n42			n84			n126			mgt		
n41			n83			n125			n167		
n40			n82			n124			n166		
n39			n81			n123			n165		
n38			n80			n122			n164		
n37			n79			n121			n163		
n36			n78			n120			n162		
n35			n77			n119			n161		
n34			n76			n118			n160		
n33			n75			n117			n159		
n32			n74			n116			n158		
n31			n73			n115			n157		
n30			n72			n114			n156		
n29			n71			n113			n155		
n28			n70			n112			n154		
n27			n69			n111			n153		
n26			n68			n110			n152		
n25		switch1	n67		switch2	n109		switch3	n151		switch4
n24			n66			n108			n150		
n23			n65			n107			n149		
n22			n64			n106			n148		
n21			n63			n105			n147		
n20			n62			n104			n146		
n19			n61			n103			n145		
n18			n60			n102			n144		
n17			n59			n101			n143		
n16			n58			n100			n142		
n15			n57			n99			n141		
n14			n56			n98			n140		
n13	PDU		n55	PDU		n97	PDU		n139	PDU	
n12			n54			n96			n138		
n11			n53			n95			n137		
n10			n52			n94			n136		
n9			n51			n93			n135		
n8			n50			n92			n134		
n7			n49			n91			n133		
n6			n48			n90			n132		
n5			n47			n89			n131		
n4			n46			n88			n130		
n3			n45			n87			n129		
n2			n44			n86			n128		
n1			n43			n85			n127		

The management node is known as 'mgt', the nodes are n1-n167, and the domain will be 'cluster'

The network is physically laid out such that port number on a switch is equal to the U position number within a column, like this:



2. Prepare for xCAT installation

xCAT install process will scan and populate certain settings from the running configuration. Having the networks configured ahead of time will aid in correct configuration.

2.1. Install the management server OS

Install RHEL5 Server 5.1 on the management server. It is recommended to ensure that dhcp, bind (not bind-chroot), expect, httpd, nfs-utils, vsftpd, and perl-XML-Parser are installed. If the management server will be on the network and RHN activated, these installs will happen automatically later if not done now.

2.2. Ensure that SELinux is disabled.

/etc/sysconfig/selinux should contain SELINUX=disabled
If this change had to be made, reboot the system.

2.3. Prevent DHCP client from overwriting DNS configuration

Find the /etc/sysconfig/network-scripts/ifcfg-* files relevant to any NICs that are DHCP configured. Put PEERDNS=no into them.

2.4. Configure NICS

Configure the cluster facing nics. An example /etc/sysconfig/network-scripts/ifcfg-eth1:

```
DEVICE=eth1
ONBOOT=yes
BOOTPROTO=static
IPADDR=172.20.0.1
NETMASK=255.240.0.0
```

2.5. Configure hostname

/etc/sysconfig/network should have HOSTNAME=(desired hostname)

2.6. Configure dns resolution

/etc/resolv.conf contents in this example:

```
search cluster
nameserver 172.20.0.1
```

2.7. Setup basic hosts file

Ensure a line like the following is in /etc/hosts:

```
172.20.0.1 mgt.cluster mgt
```

2.8. Restart management server

Though it is possible to restart the correct services for all settings except SELinux, the simplest step would be to reboot the management server at this point.

2.9. Configure ethernet switches

xCAT will use the ethernet switches for discovery. In general, this requires that the user in advance set up an ip address and basic snmp functionality. Allowing the snmp version 1 community string “public” read access will allow xCAT to communicate without further customization. It is also recommended that spanning tree be set to portfast or edge-port for faster boot performance. Please see the relevant switch documentation as to how to configure these items.

3. Install xCAT

There are two general scenarios for installation, 3.1a for disconnected operation, 3.1b for live operation. Pick either one, but not both.

3.1a. Prepare to install xCAT from disk or media

If not able to or not wishing to use the live internet repository, choose this option. Go to http://sourceforge.net/projects/xcat/, and click the green 'Download xCAT' link. Download core-repo and dep-repo tar.bz2 files
Proceed to extract to a directory:

```
# mkdir -p /install/xcat
# cd /install/xcat
# tar jxvf ~/core-repo-2.0*.tar.bz2
# tar jxvf ~/dep-repo-2.0*.tar.bz2
# xcat-core/mklocalrepo.sh
# xcat-dep/rh5/x86_64/mklocalrepo.sh
```

3.1b. Prepare to install xCAT from live internet hosted repository.

When using the live internet repostiory, simply make sure the correct repo files are in /etc/yum.repos.d:

```
# cd /etc/yum.repos.d
# wget http://xcat.sourceforge.net/yum/xcat-dep/rh5/x86_64/xCAT-dep.repo
# wget http://xcat.sourceforge.net/yum/xcat-core/xCAT-core.repo
```

3.2. Install xCAT packages

Use yum to install xCAT and chase all the dependencies for you:

```
# yum install xCAT.x86_64
# . /etc/profile.d/xcat.sh
```

4. Configure xCAT

4.1. Verify site table settings.

The process until now should have produced likely accurate defaults, however, run the following command to use vi to review the site table contents.

```
tabedit site
```

4.2. Load e1350 templates

This configuration will use the provided sample templates as is, to load them:

```
cd /opt/xcat/share/xcat/templates/e1350/
for i in *csv; do tabrestore $i; done
```

4.3. Declare a dynamic range for discovery.

In this case, we'll designate 172.20.255.1-172.20.255.254 as a dynamic range:

```
chtab net=172.16.0.0 networks.dynamicrange=172.20.255.1-172.20.255.254
```

4.4. Customize template settings

The templates that came with xCAT 2.0 should work as provided for this example. If the situation has a difference (for example U position or IP address scheme difference), a user can either configure it as they would in xCAT 1.x (each node has it's dedicated entry), or modify the group-level definitions to fit the scheme. If interested in an example of template style modification, see Appendix A.

4.5. Declare use of SOL

If not using a terminal server, SOL is recommended, but not required to be configured. To instruct xCAT to configure SOL in installed operating systems on dx340 systems:

```
chtab node=compute nodehm.serialport=1 nodehm.serialspeed=19200 nodehm.serialflow=hard
```

4.6. Add nodes to nodelist

Here use the power of the templates if used to define the nodes quickly:

```
# nodeadd n1-n167 groups=ipmi,idataplex,42perswitch,compute ,all
# nodeadd bmc1-bmc167 groups=84bmcperrack
# nodeadd switch1-switch4 groups=switch
```

At this point, xCAT should be ready to begin managing services.

5. Begin using xCAT to configure system and discover nodes

5.1. Setup hosts file

Ask xCAT to write out a hosts file per the hosts table (skip if writing /etc/hosts by hand):

```
# makehosts switch,idataplex-bmc,ipmi
```

5.2. Setup DNS

Ensure that /etc/sysconfig/named does not have ROOTDIR set, then:

```
# makedns && service named start
```

5.3. Setup DHCP

```
# makedhcp -n && service dhcpd restart
```

5.4. Configure conserver

```
# makeconservercf && service conserver start
```

5.5. Discover nodes

Walk over to systems, hit power buttons, watch tail -f /var/log/messages as nodes discover themselves

5.6. Verify state of nodes

After about 5-10 minutes, nodes should be configured and ready for hardware management:

```
# rpower all stat|xcoll
====================================
n1,n10,n100,n101,n102,n103,n104,n105,n106,n107,n108,n109,n11,n110,n111,n112,n113,n114,n115,n
116,n117,n118,n119,n12,n120,n121,n122,n123,n124,n125,n126,n127,n128,n129,n13,n130,n131,n132,
n133,n134,n135,n136,n137,n138,n139,n14,n140,n141,n142,n143,n144,n145,n146,n147,n148,n149,n1
5,n150,n151,n152,n153,n154,n155,n156,n157,n158,n159,n16,n160,n161,n162,n163,n164,n165,n166,
n167,n17,n18,n19,n2,n20,n21,n22,n23,n24,n25,n26,n27,n28,n29,n3,n30,n31,n32,n33,n34,n35,n36,n3
7,n38,n39,n4,n40,n41,n42,n43,n44,n45,n46,n47,n48,n49,n5,n50,n51,n52,n53,n54,n55,n56,n57,n58,n5
9,n6,n60,n61,n62,n63,n64,n65,n66,n67,n68,n69,n7,n70,n71,n72,n73,n74,n75,n76,n77,n78,n79,n8,n80
,n81,n82,n83,n84,n85,n86,n87,n88,n89,n9,n90,n91,n92,n93,n94,n95,n96,n97,n98,n99
====================================
on
```

6. Install nodes

6.1. Begin installation

The following command will commence installation to disk on all of the nodes

```
# rinstall -o rhels5.1 all
```

6.2. Monitor installation

It is possible to use the wcons command to monitor a sampling of the nodes:

```
# wcons n1,n20,n80,n100
```

Additionally, nodestat may be used to check the status of a node as it installs:

```
# nodestat n20,n21
n20: installing man-pages - 2.39-10.el5 (0%)
n21: installing prep
```

After some time, the nodes should be up and ready for general usage

7. Useful Applications of xCAT commands

For any given command, typing 'man command' should give an in depth document on the workings of that command. Here are some examples of using key commands and command combinations in useful ways.

7.1. Adding groups to a set of nodes

In this configuration, a handy convenience group would be the lower systems in the chassis, the ones able to read temperature and fanspeed. In this case, the odd systems would be on the bottom, so to do this:

```
# nodech '/n.*[13579]$' groups,=bottom
```

7.2. Listing attributes

We can list discovered and expanded versions of attributes (Actual vpd should appear instead of *) :

```
# nodels n97 nodepos.rack nodepos.u vpd.serial vpd.mtm
n97: nodepos.u: A-13
n97: nodepos.rack: 2
n97: vpd.serial: ********
n97: vpd.mtm: *******
```

7.3. Verifying consistency and version of firmware

Combining the use of in-band and out-of-band utilities with xcoll, it is possible to quickly analyze the level and consistency of firmware across the servers:

```
# rinv n1-n3 mprom|xcoll
====================================
n1,n2,n3
====================================
BMC Firmware: 1.18
```

The BMC does not have the BIOS version, so to do the same for that, use psh:

```
# psh n1-n3 dmidecode|grep "BIOS Information" -A4|grep Version|xcoll
====================================
n1,n2,n3
====================================
        Version: I1E123A
```

7.4. Reading and interpreting sensor readings

If the configuration is louder than expected (iDataplex chassis should nominally have a fairly modest noise impact), find the nodes with elevated fanspeed:

```
# rvitals bottom fanspeed|sort -k 4|tail -n 3
n3: PSU FAN3: 2160 RPM
n3: PSU FAN4: 2240 RPM
n3: PSU FAN1: 2320 RPM
```

In this example, the fanspeeds are pretty typical. If fan speeds are elevated, there may be a thermal issue. In a dx340 system, if near 10,000 RPM, there is probably either a defective sensor or misprogrammed power supply.

To find the warmest detected temperatures in a configuration:

```
# rvitals bottom temp|grep Domain|sort -t: -k 3|tail -n 3
n3: Domain B Therm 1: 46 C (115 F)
n7: Domain A Therm 1: 47 C (117 F)
n3: Domain A Therm 1: 49 C (120 F)
```

Change tail to head in the above examples to seek the slowest fans/lowest temperatures. Currently, an iDataplex chassis without a planar tray in the top position will report '0 C' for Domain B temperatures.

8. Advanced features

8.1 Use the driver update disk

Linux supplies the “driver update disk” mechanism to support the devices which cannot be driven by the released distribute during the installation process. “driver update disk” is a media which containing the drivers and related configuration files for certain devices. The “driver update disk” is always supplied by the vendor of device. One “driver update disk” can contain multiple drivers for different os release and different hardware architecture. The Redhat and Suse have different “driver update disk” format.
xCAT supports to load the “driver update disk” to drive the devices during the installation or netboot process.

Refer to following steps to use “driver update disk” :

1. Get the “driver update disk” from the vendor of device. The “driver update disk” should support the <os> and <arch> of the target node.
2. Copy the “driver update disk” into directory <installdir>/ driverdisk/<os>/<arch>.
<installdir> is the directory which xCAT stores the installation material. The name of this directory is stored in the table site.installdir. The default value is “/install”.
<os> is the operating system type of the target node
<arch> is the hardware architecture of the target node.
3. Run the “nodeset” command for the diskfull node; Run the “genimage” command for the diskless node.
4. Start the installation as common node.

It depends on the format of “driver update disk” that whether the drivers in the “driver update disk” will be installed to the target diskfull node persistently. If the “driver update disk” does not support to install the drivers to the installed system, get the kmod rpm packages and use the otherpkgs postscript to install it.
The steps to install the kmod by the otherpkgs mechanism:

1. Get the kmod rpm packages, includes the dependency packages. (The kmod rpm maybe can be get from the “driver update disk”)
2. Put rpms into the /install/post/otherpkgs/<os>/<arch> directory.
3. Put the name of the packages to the file /opt/xcat/share/xcat/netboot(install)/<platform>/<otherpkgs.pkglist>. The file name can be one of:
   profile.os.arch.otherpkgs.pkglist
   profile.os.otherpkgs.pkglist
   profile.arch.otherpkgs.pkglist
   profile.otherpkgs.pkglist

Note: If the nodes have already installed and up and running, after finished the preceding steps, you can run the following command to have the kmod rpms installed for an installed node:

```
updatenode noderange otherpkgs
```

Appendix A. Template modification example

Templates support powerful expressions for defining a scheme based configuration. This can help for more dynamic configurations or defining a site-standard set of defaults once and applying to multiple clusters. Here we will take two of the default schemes and modify them to support a configuration where n1 is in switch port 2, U position 2, and so on in the first rack. Keep in mind that this is merely an option, not a requirement, and per-node settings are always an option for those that would prefer it. First, extract the current templates for nodepos.rack, nodepos.u, and the ip addresses for nodes and bmcs:

```
# gettab node=idataplex nodepos.rack nodepos.u
nodepos.u: |\D+(\d+).*$|(sprintf("%c",(65+2*(((($1-1)/42)%2))))(($1-1)%42+1)|
nodepos.rack: |\D+(\d+).*$|(1+(($1-1)/84))|
# gettab node=42perswitch switch.port switch.switch
switch.switch: |\D+(\d+).*$|switch(($1-1)/42+1)|
switch.port: |\D+(\d+).*$|(($1-1)%42+1)|
# gettab node=84nodeperrack hosts.ip
    |\D+(\d+).*$|172.20.(101+(($1-1)/84)).(($1-1)%84+1)|
# gettab node=84bmcperrack hosts.ip
   |\D+(\d+).*$|172.29.(101+(($1-1)/84)).(($1-1)%84+1)|
```

The left hand side of the values represents how a number is extracted from a node's name, by enclosing it in parentheses. The right hand side can then perform some arithmetic to designate a value. In this case, we are changing the underlined offset to '-0' to reflect the fact that n1 should have the value n2 would have had in the default configuration.

```
# chtab node=idataplex \
  nodepos.u='|\D+(\d+).*$|(sprintf("%c",(65+2*(((($1-0)/42)%2))))(($1-0)%42+1)|'\
  nodepos.rack='|\D+(\d+).*$|(1+(($1-0)/84))| '
# chtab node=42perswitch \
  switch.switch='|\D+(\d+).*$|switch(($1-0)/42+1)|' \
  switch.port='|\D+(\d+).*$|(($1-0)%42+1)| '
# chtab node=84nodeperrack \
   hosts.ip='|\D+(\d+).*$|172.20.(101+(($1-0)/84)).(($1-0)%84+1)|'
# chtab node=84bmcperrack \
    hosts.ip='|\D+(\d+).*$|172.29.(101+(($1-0)/84)).(($1-0)%84+1)|'
```