Managed Platforms Unit

The MPU is one of the School of Informatics' units of computing staff. It's responsible for the Linux platform which forms the basis of DICE. It also maintains the tools needed by that platform, principally LCFG.

Category: SL7 server base (356)

LVM, SL7 and physical volume device names

The use of traditional /dev/sd[xx] disk device names has become increasingly unsafe in recent Linux installs. This is particularly so in an environment with SAN devices. For a while now, we have been mounting disk partitions by UUID, but until recently the LVM component has continued to rely on /dev/sd[xx] names.

The LVM component, on configure(), checks to see whether any additional physical volumes have been configured for a volume group. It does this by using the 'pvs' component to enumerate the physical volumes are associated with each volume group. The command vgextend (or vgcreate) are used to add a physical volume to a volume group. Unfortunately, these commands store away the resulting /dev/sd[xx] name of they physical volumes – and not the /dev/by-uuid name. This means that on subsequent reboots, there's a high chance that the /dev/sd[xx] name will be wrong for a physical volume.

The solution is to generate a UUID on each physical volume (based on a hash of the physical path name), label the physical volume with that UUID (with pvcreate) and look for that UUID using 'pvs -o pv_uuid,vg_name' instead of looking for the /dev/sd[xx] device.

June 21, 2016 Alastair Scobie SL7 server base (356) Leave a comment

Consistent network names – virtual

hardware

As described in an earlier post, we have recently enabled consistent network interface names under SL7 – for physical machines.

We have concluded that this is not practical to do for virtual hardware. The device names presented by the consistent network interface name scheme depend on the the underlying configuration of the virtual guests. As this configuration is not under our control, the device names will be unpredictable.

March 10, 2016 Alastair Scobie SL7 server base (356) Leave a comment

SL7 – multipath and LVM

We have been working on checking support for DM multipath and LVM under SL7.2. Our first attempts at doing this under SL7.1 failed miserably as a result of unpredictable FibreChannel problems – we've found, in the past, that support for FibreChannel in early dot releases of new RHEL/SL major releases is flaky.

First off, we discovered that our standard LCFG SL7 platform had *dmraid* (software RAID) enabled as standard. This was creating a lot of excessive "noise" from the kernel at boottime as the *dmraid* module attempts to scan every attached block device – this was particularly noticeable on a SAN attached host with multiple routes to SAN volumes. We have disabled *dmraid* by default, and created a header file to pull it back in where required.

We next looked at DM multipath. Confusingly, despite the version of DM-multipath not having changed between SL6 and SL7, various parameters have changed. A template for SL7 (actually, EL7) was created, and a couple of multipath component resources added, to support the new parameters that we need to tweak. With SL6, and earlier, we added manual configuration to support our IBM disk array. There is built-in support for this array in SL7, so our manual configuration can be removed. Note, however, that we have never added configuration for the DotHill arrays – it may be that we are using inappropriate values for these (eg no_path_retry=fail rather than queue) under SL6.

Then onto LVM. Some confusing behaviour was discovered to be caused by the daemon *lvmetad* which is now enabled by default on SL7. For some reason, on some system boots, pvscan was returning an unknown physical volume device for a volume group – this was making the LVM component re-adding the configured physical volume to the volume group (because it couldn't see it in the list). This in turn created a new PV UUID, and you ended up with a volume group with an increasing number of missing physical volumes. The volume group would work, however. It's possible that running LVM in initramfs would fix this, but disabling *lvmetad* also fixed it. It seems that the purpose of *lvmetad* is to reduce the time taken at system boot to scan block devices for LVM physical volumes. We don't have so many physical devices that an LVM scan at boottime takes ages so running *lvmetad* seems unnecessary. However, we may need to revisit this in the future?

In SL6 and earlier, multipath and LVM configuration was required to be loaded into the initrd. Both the multipath and lvm components would trigger a rebuild of the initrd (via the kernel component) whenever their configuration changed. It looks like this is not necessary for SL7, even where multipath provisioned filesystems are mounted in /etc/fstab. We have done lots of testing, but it's still possible that we've just been lucky with the timing. If we do need to load multipath and/or LVM configuration into the initrd, we will need to consider how best to do this. Under SL7, dracut will no longer automatically upload LVM and multipath configuration into the initrd : modifications to the kernel, lvm and multipath components may all be required.

📅 March 10, 2016 Alastair Scobie SL7 server base (356) Leave a comment

Consistent network interface names and LCFG

As explained in an earlier post, we are moving to the more "modern" consistent network interface naming scheme because the old-style method of hard-wiring interfaces to interface names of the form eth0 no longer works with RHEL7. This is a problem for machines with multiple interfaces – eg servers.

(Note that you can stick with the legacy naming scheme by defining LCFG_NET-

WORK_LEGACY_NAMING at the head of a machine's profile).

As a recap, under the consistent naming scheme, interfaces are known as :-

Device	Name
On-board (embedded) interface	em[1234…]
PCI card interface	p<slot>p<port>
Virtual	p<slot>p<port>_<virtualif>

For example, a minimally configured Dell R730 has four on-board interfaces : these would be called em1, em2, em3 and em4.

We could modify all our LCFG configuration to use the new names directly. However, there are many LCFG macros which assume that network interfaces are of form eth[n] and changing these would be somewhat disruptive. We have decided to stick with using the old form eth[n] in LCFG configuration, providing a means of associating these names with a real physical device. The macro call LCFG_NETWORK_SET_DEVICE(eth0,em1) will associate the LCFG name eth0 with the physical device em1. The hardware headers for each machine model will include calls to LCFG_NETWORK_SET_DEVICE for all the onboard network interfaces. This means that the process should be largely transparent on most machines.

Some machines have both embedded and PCI-E interfaces. For example, in Informatics, HP DL180s will commonly have two onboard interfaces and one PCI-E interface. We usually configure these machines such that a network bond is formed using one of the onboard interfaces (usually the second as the first is used for IPMI) and the PCI-E interface. On DICE, the following will be configured by default for these machines :-

```
LCFG_NETWORK_SET_DEVICE(eth0,p1p1)
LCFG_NETWORK_SET_DEVICE(eth1,em2)
```

Note that old form of eth[n] is only used in LCFG configuration – all operating system tools (eg netstat) and configuration will expect names of the form em[n] or pp.

We shall probably convert LCFG configuration to use the native network interface names throughout – possibly at the next major platform upgrade.

February 8, 2016 Alastair Scobie SL7 server base (356) Leave a comment

Hardware monitoring and RAID on SL7

Informatics uses a Nagios monitoring system to keep track of the health and current status of many of its services and servers. One of the components of the Nagios environment is `lcfg-hwmon`. This periodically performs some routine health checks on servers and services then sends the results to Nagios, which alerts administrators if necessary. `lcfg-hwmon` checks several things:

- It warns if any disks are mounted read-only. The SL6 version excluded device names starting `/media/` and `/dev/loop`. The SL7 version also ignores anything mounted on `/sys/fs/cgroup`. This check can be disabled by giving the `hwmon.readonlydisk` resource a false value.
- If it finds RAID controller software it uses this to get the current status of the machine's RAID arrays, then it reports any problems found. It knows about MegaRAID SAS, HP P410, Dell H200 and SAS 5i/R RAID types. Note that the software does not attempt to find out what sort of RAID controller the machine actually has, so the administrator has to be sure to use the correct RAID header when configuring the machine.
- It warns if any of the machine's power supply units has failed or is indicating a problem.

As well as the periodic checks from `cron` a manual status check can be done with

```
/usr/sbin/check_hwmon --stdout
```

If the `--stdout` option is omitted the result is sent to Nagios rather than displayed on the shell output.

Version 0.21.2-1 of `lcfg-hwmon` functions properly on SL7 servers. In Informatics, any server using `dice/options/server*.h` gets `lcfg-hwmon`. Other LCFG servers can get it like this:

```
#include <lcfg/options/hwmon.h>
```

In related news, the RAID controller software for the RAID types listed above is now installed on SL7 servers by the same headers as on SL6. The HP P410 RAID software has changed its name from `hpacucli` to `hpssacli` but seems otherwise identical. The Dell H200 software `sas2ircu` has gained a few extra commands (SETOFFLINE, SETONLINE, ALTBOOTIR, ALTBOOTENC) but the existing commands seem unchanged. The other varieties of RAID software are much as they were on SL6.

January 19, 2016 Chris Cooke SL7 server base (356) Leave a comment

Network device naming

During our original project to port LCFG to SL7 we were only really considering desktops which typically have a single network interface. To get things working quickly we decided to stick with the "legacy" network device naming scheme which gives us interfaces named like eth0, eth1, eth2, etc. This works just fine with a single interface since we will only ever need access to eth0 but as we've moved onto adding network interface bonding for servers we have some discovered some problems. Many of our servers have two controllers each of which has two devices, for maximum reliability we wish to bond over one device from each controller. Traditionally we have done this by naming eth0 as the first device on the first controller and eth1 as the first device on the second controller. We have found with the legacy support on SL7 that this is not possible as they always come out as eth0 and eth2 (eth1 being the second device on the first controller) it seems that the ability to rename interfaces based on MAC address is not working correctly. Due to the way we have configured bonding in LCFG, for simplicity we really would like the two interfaces to continue to be named eth0 and eth1. To resolve this problem we have decided that it is now time to convert to the "modern" naming scheme as described in the Redhat network guide. The interfaces can then be aliased as eth0 and eth1 after they have been configured with their "consistent" names. This appears to work as desired but requires some changes be made in the LCFG headers and we will be working through this transition over the next few weeks. It is likely that the complete change to the default approach will have to wait until the SL7.2 upgrade to ensure we don't break anything. The first step will be to move the "legacy" support out of the lcfg-level header (`lcfg/defaults/network.h`) into the ed-level header, this will not have any impact for most users but makes it possible to easily enable and disable the naming schemes for testing purposes. New headers have been provided – `lcfg/options/network-legacy-names.h` and `lcfg/options/network-modern-names.h` – to make it easy to swap between the two naming schemes. Once we are confident that this modern approach is reliable we will update the various hardware support headers in the lcfg-level so that it works for the various server models we have in Informatics.

📅 January 13, 2016 squinney SL7 server base (356) Leave a comment

Progress so far on SL7 server base

Every year or two we migrate all of DICE to a newer operating system version, so that we can keep up with technology advances and security fixes. Most recently we've been moving it from Scientific Linux 6 to Scientific Linux 7.

When migrating DICE to a new platform, we make the move in several stages. First we need our configuration environment LCFG fully working on the new OS (see for instance Work involved in porting DICE to SL7); then we work on the desktop computing environment, and the research and teaching software it needs; after that comes the tools and environment for servers. We're tackling the last of those stages now, the SL7 server platform project. We have several hundred servers, hosting both a variety of services and a range of behind-the-scenes support functions.

So far we've tested and passed these things:

- **Server networking features**. Setting NM_CONTROLLED=no in the network interface config files allows us to use the old networking scripts to setup *bonding*, *bridging* and *VLANs*. We'll take a look at doing this with Network Manager later on, since the old networking scripts will probably be removed at some point, but in the meantime we have access to the networking functionality which our servers need.
- **IPMI**. We use it for our monitoring needs and for Serial Over LAN (remote consoles and remote power control).
- Our standard SL7 **disk partition layout**.
- The basic active checks for our **Nagios** monitoring setup.
- We've installed the software needed by the Nagios passive check which monitors network bonding, and it's now working correctly.

- The **hwmon** passive check does a variety of hardware health tests. These ones have been tested and work on SL7: read-only disk mounts; MegaSAS RAID; dual power supply redundancy; LSI SAS 5i/R RAID.
- **RAID controller** software and LCFG configuration headers for MegaSAS RAID and for LSI SAS 5i/R RAID.
- The **toohot** overheating emergency shutdown tool.
- Fibre Channel **Multipath**. The ability to use multiple paths through the FC fabric increases the dependability of our storage area network facilities.
- **LVM**. This storage abstraction layer is used for storage space for the VMs on our virtualisation servers.
- We have rethought the **DNS** configuration for SL7. Instead of using only *localhost* for DNS lookups, SL7 servers will be configured to query the full set of DNS servers.

We're currently working on support for other RAID types, on LCFG **apacheconf** and on other aspects of Fibre Channel functionality.

📅 November 25, 2015 Chris Cooke SL7 server base (356) Leave a comment

Proudly powered by WordPress