Recording Local Storage Configuration

FOSDEM — Lightning Talks

1st February 2020

Alasdair Kergon

agk@redhat.com

Red Hat

Common Problems

- My system has a problem with its storage devices.
  - It's triggered thousands of log messages from different layers of the storage stack and applications on top.
    - How do I correlate them and quickly identify which are the important ones relating directly to the cause rather than the consequences?
  - How do I extract all the system log messages relating to a specific device?
- Log messages are rather inconsistent, with different types of messages using different types of identifiers.
  - Some identifiers (like device number – major/minor numbers) may change each time I boot.
- Information in /dev and /sys is transient. No records are kept of some information I might want to refer to again later.

Towards Solutions

- Need a persistent identifier for each device.
  - Already have well-established /dev/disk/by-* created by udev rules.
    - Tracks ids, labels, paths, uuids etc.
- So let's record this information somewhere persistent where we can query it later.
  - System journal can store structured data as key-value pairs.
  - journalctl can filter this by identifiers that don't change.
- Using this information, starting from the time of the last boot, we can 'play back' recorded changes and reproduce part of the configuration as it was at earlier points in time.

Enter storage-logger

- When a block device is added or removed an event is generated – a uevent – which triggers udev rule processing that performs actions such as finding out what type of device it is and what is on it and creating nodes and symbolic links in /dev.
- Record the results of all this in the system journal.
- Include additional useful information from /sys.
- It consists of a simple bash script run at the end of uevent rule processing.
- In future, part of it could be folded into existing udev rules and the rest could be built directly into udev itself.

Enter lsblkj

- A new wrapper around lsblk, currently implemented in perl.
- Takes new time arguments --since and --until.
- Plays back the uevents recorded in the journal between those times.
- Creates temporary /dev and /sys directories that look similar to the original ones at the specified time.
- Invokes lsblk using these temporary directories instead of the real ones.

Enter skydive

- Transfer the newly-recorded data into a graph database highlighting the relationships between the storage layer components.
- Store additional transient data such as performance metrics there too.
- Layer a graphical interface on top.

Example parameters

- journalctl
  -t UDEVLOG
  --output verbose
  --output-fields=PERSISTENT_STORAGE_ID,MAJOR,MINOR
  --since "2020-02-01 18:00:00"
  --until "2020-02-01 18:10:00"
  PERSISTENT_STORAGE_ID=dm-name-vg1-lvol0
- lsblkj --until "2020-02-01 18:10:00"

Example output – journalctl

```
# journalctl -t UDEVLOG --output verbose
--output-fields=PERSISTENT_STORAGE_ID,MAJOR,MINOR
PERSISTENT_STORAGE_ID=dm-name-fedora_kvm--01--guest10-root

-- Logs begin at Mon 2020-01-27 11:23:30 CET, end at Sat 2020-02-01
12:40:56 CET. --
Mon 2020-01-27 11:14:44.053704 CET
[s=eb2c7dddabab423180e32632882802bb;i=3c5;b=bcd345b7d087493a99468522653479
0a;m=c11e17;t=59d1c604ffedb;x=8b09a96eadde58e6]
    MINOR=0
    MAJOR=253
    PERSISTENT_STORAGE_ID=dm-uuid-LVM-J6yzG5EAvddvNFPS6rGSbdb21qluti3
MxP1D4AjMbn45qqhWfAJIftL6oXfgovdB
    PERSISTENT_STORAGE_ID=dm-name-fedora_kvm--01--guest10-root
```

Example output – lsblkj (1)

```
# lsblkj
NAME                                    MAJ:MIN RM  SIZE RO TYPE
MOUNTPOINT
vda                                     252:0    0   50G  0 disk
├─vda1                                  252:1    0    1G  0 part
└─vda2                                  252:2    0   49G  0 part
  ├─fedora_kvm--01--guest10-root        253:0    0   15G  0 lvm
  └─fedora_kvm--01--guest10-swap        253:1    0    4G  0 lvm
```

Example output – lsblkj (2)

Create a device called 'test1'

```
# date
Sat 01 Feb 2020 12:42:23 PM CET

# dmsetup create test1
0 50 error
```

Example output – lsblkj (3)

Make it bigger

```
# date
Sat 01 Feb 2020 12:42:35 PM CET

# dmsetup load test1
0 50000 error

# dmsetup resume test1

# date
Sat 01 Feb 2020 12:42:56 PM CET
```

Example output – lsblkj (4)

```
# lsblkj --until "2020-02-01 12:42:30"
NAME                                    MAJ:MIN RM  SIZE RO TYPE
MOUNTPOINT
vda                                     252:0    0   50G  0 disk
├─vda1                                  252:1    0    1G  0 part
└─vda2                                  252:2    0   49G  0 part
  ├─fedora_kvm--01--guest10-root        253:0    0   15G  0 lvm
  └─fedora_kvm--01--guest10-swap        253:1    0    4G  0 lvm
```

Example output – lsblkj (5)

```
# lsblkj --until "2020-02-01 12:42:40"
NAME                                  MAJ:MIN RM  SIZE RO TYPE
MOUNTPOINT
vda                                   252:0    0   50G  0 disk
├─vda1                                252:1    0    1G  0 part
└─vda2                                252:2    0   49G  0 part
  ├─fedora_kvm--01--guest10-root      253:0    0   15G  0 lvm
  └─fedora_kvm--01--guest10-swap      253:1    0    4G  0 lvm
test1                                 253:2    0   25K  0 dm
```

Example output – lsblkj (6)

```
# lsblkj
NAME                                    MAJ:MIN RM   SIZE RO TYPE
MOUNTPOINT
vda                                     252:0    0    50G  0 disk
├─vda1                                  252:1    0     1G  0 part
└─vda2                                  252:2    0    49G  0 part
  ├─fedora_kvm--01--guest10-root        253:0    0    15G  0 lvm
  └─fedora_kvm--01--guest10-swap        253:1    0     4G  0 lvm
test1                                   253:2    0 24.4M  0 dm
```

Demo

- skydive video prepared by Todd Gill <tgill@redhat.com>

Links

- storage-logger and lsblkj

  https://github.com/lvmteam/storage-logger/

  https://copr.fedorainfracloud.org/coprs/agk/storage-logger/

- Skydive

  http://skydive.io/

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.


linkedin.com/company/red-hat


youtube.com/user/RedHatVideos


facebook.com/redhatinc


twitter.com/RedHat

Red Hat