
facebook

Lessons learned from a large OpenBMC deployment

Christopher Covington and Samantha Downs

Production Engineers

What is Production Engineering?

- Security
- Scalability
- Efficiency
- Configuration management
- Upgrades
- Deployment
- Performance tuning
- Disaster readiness
- Capacity planning
- Reliability
- Contingency planning

What is OpenBMC?

- Open-source Board Management Controller (BMC)
- Yocto/OpenEmbedded Linux distribution
- Linux Foundation

Our OpenBMC deployment

What does the fleet look like?

- 100,000’s of OpenBMC devices in production
- 9 different platforms
- ARM v5/v6 single core CPUs
- 256+ MB RAM
- 32 MB SPI NOR Flash
- U-Boot

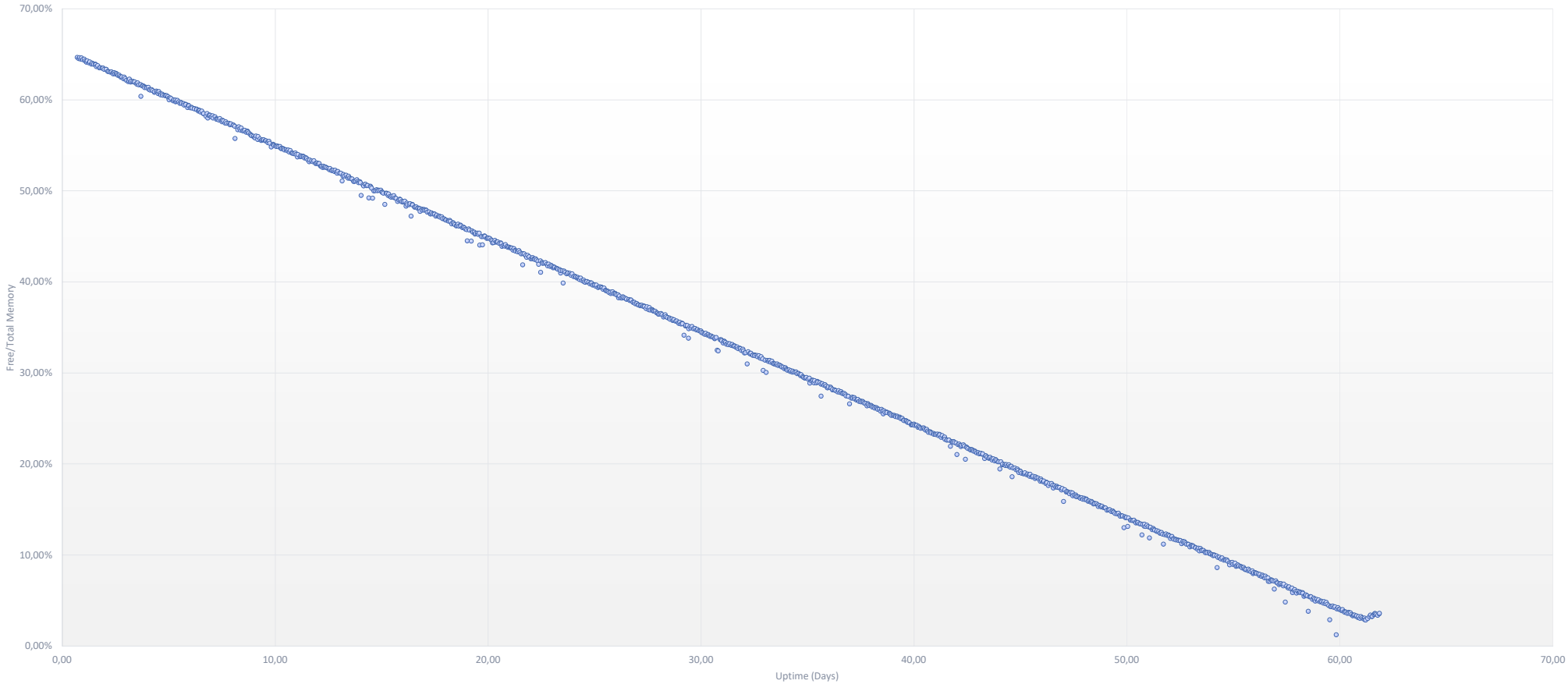
Three Stories

(Three Lessons learned)

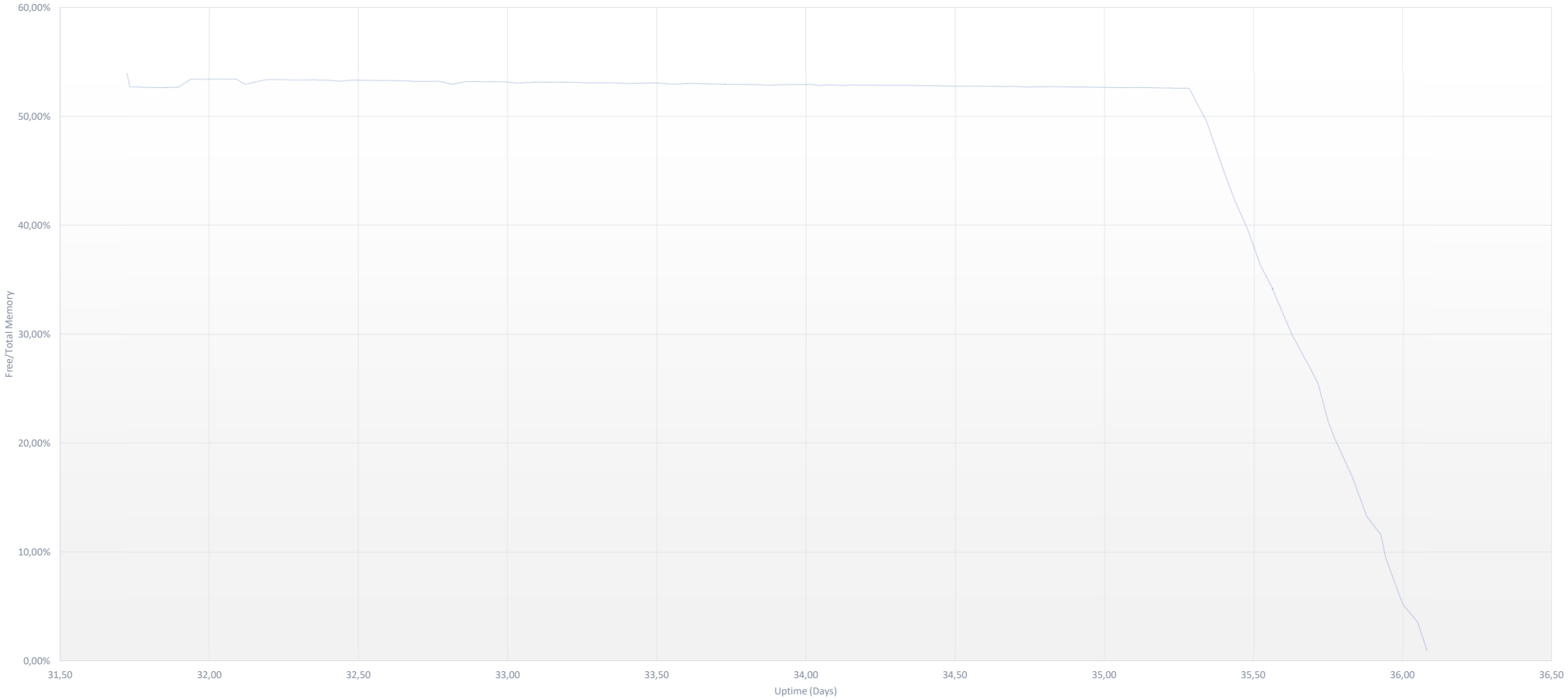
- Out of Memory in 1 to 60 Days
- The Pain of Passwords
- (Re)Designing for Resilience

Out of Memory in 1 to 60 Days

Out of Memory in 60 Days



Out of Memory in 1 Day



Out of Memory in 1 to 60 Days

Why to Engage Upstream and Rebase Often

- We’ve seen three problems:
  - Linux kernel inode cache memory leak in 2.6
  - rsyslog memory leak
  - Lack of log rotation on tmpfs (unbounded max size)

The Pain of Passwords

The Pain of Passwords

Why to Invest In Security

- Setting the BMC’s hostname broke pexpect based tools
- Turning off the echo flag fixed it

The Pain of Passwords

Why to Invest In Security

- Trusted CA and Authorized Principals are great
  - SSH server configuration is easy
  - CA only contacted when creating/renewing client certificates
  - Group based (not user based)
  - Certificates expire and can be revoked

Trusted CA and Authorized Principals Example

```
$ ssh-keygen -C CA -f ca
TrustedUserCAKeys /etc/ssh/ca.pub
$ ssh-keygen -t ecdsa  # or -t rsa, up to you
$ ssh-keygen -s ca -I cov -n root -V +1w -z 1 id_ecdsa.pub
$ ssh-keygen -Lf id_ecdsa-cert.pubid_ecdsa-cert.pub:
  Type: ecdsa-sha2-nistp256-cert-v01@openssh.com user certificate
  Public key: ECDSA-CERT ...
  Signing CA: ECDSA ...
  Key ID: "cov"
  Serial: 1
  Valid: from 2016-01-13T15:26:00 to 2016-01-20T15:27:00
  Principals:
    root
  Critical Options: (none)
  Extensions:
    permit-X11-forwarding
    permit-agent-forwarding
    permit-port-forwarding
    permit-pty
    permit-user-rc
$ ssh root@bmc-oob
Accepted publickey for root from 1.2.3.4 port 123 ssh2: ECDSA-CERT ID cov (serial 1) CA ECDSA fingerprint...
```

(Re)Designing for Resilience

(Re)Designing for Resilience

- The problem:
  - OpenBMC rsyslog retention began dropping dramatically
  - Coincided with rolling out a new OpenBMC image to one of the platforms

Upon further investigation

root@openbmc:~# cat /var/log/messages | wc -l
9999
root@openbmc:~# grep -c 'fscd: Exception with board' \
/var/log/messages
5184

(Re)Designing for Resilience

- What happened?
  - New image reformatted persistent storage
  - This deleted cache files that are **only** populated by IPMI messages sent by the host system's BIOS
  - Without the cached files, the error occurs every few seconds for each DIMM of each host (4 * 4 * <lots of devices>)

(Re)Designing for Resilience

- Rebooting all of the hosts isn’t an option, so:
  - Construct the DIMM info IPMI message using dmidecode output on the Linux host
  - Send it using ipmitool ☺

Report DIMM info from the host

```
root@some-host:~# ipmitool raw 0x30 0x1c 0x01 \
    0xc6 0x55 0x08 0x00 0x20 0x00 0x00
```


OpenBMC
UEFI (bootloader)
Host (Linux)
rsyslog
IPMI DIMM Info
Write cache files
Boot OS
Read cached DIMM info
Read cached DIMM info
Cache files deleted on OpenBMC
Read cached DIMM info
FSCD Exception
Read cached DIMM info
FSCD Exception
Parse dmidecode
Send the IPMI DIMM info using ipmitool
Read cached DIMM info
Read cached DIMM info

(Re)Designing for Resilience

- Can we do better?
  - Manage dependencies (don’t depend on one off events)
  - Clear interfaces
  - How do other industries do this?

Contact

- Christopher Covington: cov@fb.com
- Samantha Downs: sdowns@fb.com
- Links:
  - Linux Foundation repo: https://github.com/openbmc/openbmc
  - FB repo: https://github.com/facebook/openbmc
  - https://www.openbmc.org/
  - https://code.fb.com/security/scalable-and-secure-access-with-ssh/

Questions?


facebook