Unveiling OpenBMC: Exploring features and preparing the build environment

Yocto DevDay 2023

Maciej Pijanowski
Maciej Pijanowski
Engineering Manager

- @macpijan
- maciej.pijanowski@3mdeb.com
- linkedin.com/in/maciej-pijanowski-9868ab120
- over 7 years in 3mdeb
- Open-source contributor
- Interested in:
  - build systems (e.g., Yocto)
  - embedded, OSS, OSF
  - firmware/OS security
coreboot licensed service providers since 2016 and leadership participants
UEFI Adopters since 2018
Yocto Participants and Embedded Linux experts since 2019
Official consultants for Linux Foundation fwupd/LVFS project since 2020
IBM OpenPOWER Foundation members since 2020
• What is BMC?
• What is OpenBMC?
• OpenBMC features
• Repository overview
• Build environment
• Adding Supermicro X11SSH-TF
• Results
• Next steps
What is BMC?

https://www.aspeedtech.com/server_ast2500/
What is OpenBMC?

- Linux Foundation project since 2018
- Linux distribution for BMCs
- A collection of open-source tools implementing BMC features
- Built with Yocto
- Code: https://github.com/openbmc/openbmc
- Documentation: https://github.com/openbmc/docs
Project in dynamic development

Some significant changes in the last years
  - switching from OpenBMC REST API to Redfish API
  - switching from phosphor-webui to webui-vue
  - switching mechanisms for inventory, power control, ...

Documentation not always keeps up with the development
  - you may come across some outdated/deprecated documents
  - not particularly clear for a newcomer what is up to date
  - it may be confusing at first
OpenBMC features

- Web-based user interface
- Event logging and alerting
- Sensors monitoring
- Host management (power, fans, LEDs, ...)
- Management protocols support (IPMI, Redfish)
- Serial over LAN console
- KVM
- Virtual media
- Firmware update (BMC and host)
- More
Web-based user interface

- [https://github.com/openbmc/webui-vue](https://github.com/openbmc/webui-vue)
Event logging and alerting

- [https://github.com/openbmc/phosphor-webui](https://github.com/openbmc/phosphor-webui)
## Sensors

### All sensors present in the system

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Low critical</th>
<th>Low warning</th>
<th>Current</th>
<th>High warning</th>
<th>High critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature Ambient</td>
<td>-40 °C</td>
<td>0 °C</td>
<td>21.375 °C</td>
<td>35 °C</td>
<td>55 °C</td>
</tr>
<tr>
<td>Temperature Cpu 1 Ambient</td>
<td>-40 °C</td>
<td>0 °C</td>
<td>25.5 °C</td>
<td>65 °C</td>
<td>85 °C</td>
</tr>
<tr>
<td>Temperature Pcie</td>
<td>-40 °C</td>
<td>0 °C</td>
<td>26 °C</td>
<td>45 °C</td>
<td>65 °C</td>
</tr>
<tr>
<td>Fan Tach Fan0</td>
<td>300 RPM</td>
<td>500 RPM</td>
<td>8823 RPM</td>
<td>20000 RPM</td>
<td>30000 RPM</td>
</tr>
</tbody>
</table>
Server power operations

**Current status**

Last power operation at Jun 12, 2023 11:06:03 AM GMT+2

- **talos - [redacted]**
  - Running

**Select a power operation**

- **Warm reboot**
  - Attempts to perform an orderly shutdown before restarting the server
- **Cold reboot**
  - Shuts down the server immediately, then restarts it
- **Orderly shutdown**
  - Attempts to stop all software on the server before removing power
- **Immediate shutdown**
  - Removes power from the server without waiting for software to stop
Serial over LAN console

Access the Serial over LAN console

The Serial over LAN (SoL) console redirects the output of the server’s serial port to a browser window on your workstation.

- "Boot from USB"
- "Select a new boot device"
- "Boot from list"
- "Continue to the main menu"
- "Exit to recovery shell"

Open in new tab
Virtual media

Specify image file location to start session.

Virtual media device

Choose file  No file selected

Start
Exposes mechanisms for updating both host and BMC firmware

- Multiple ways of providing/trIGGERING update
  - scp / TFTP and local DBus command (via cmdline)
  - OpenBMC REST API
  - Redfish API
  - end-user can use web UI as well

- Documentation
  - https://github.com/openbmc/docs/tree/master/architecture/code-update
- IPMI replacement standardized by the DMTF in 2015 (v1.0)
- Aims to be interoperable (cross-vendors)
- Provides REST API for platform management
  - status, power control, boot order, Secure Boot, sensors, users, ...
- Implemented by bmcweb in OpenBMC
  - [https://github.com/openbmc/bmcweb/blob/master/Redfish.md](https://github.com/openbmc/bmcweb/blob/master/Redfish.md)
- Example API call for gracefull shutdown

```
curl -k -H "X-Auth-Token: $token" -H "Content-Type: application/json" \
   -X POST https://${bmc}/redfish/v1/Systems/system/Actions/ComputerSystem.Reset \
   -d '{"ResetType": "GracefulShutdown"}ʹ
```
- Platform Level Data Model
  - protocol and data model for BMC <-> hardware communication
  - provides a common language and messages structure
  - interoperability in system management operations
- OpenBMC implementation
  - [https://github.com/openbmc/pldm](https://github.com/openbmc/pldm)
- OpenBMC presentation on PLDM stack
  - [https://www.youtube.com/watch?v=AvW0ENBbkPg](https://www.youtube.com/watch?v=AvW0ENBbkPg)
- PLDM Base Specification:
  - [https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.1.zip](https://www.dmtf.org/sites/default/files/standards/documents/DSP0240_1.1.zip)
- PLDM stack on OpenBMC
  - [https://github.com/openbmc/docs/blob/master/designs/pldm-stack.md](https://github.com/openbmc/docs/blob/master/designs/pldm-stack.md)
  - explains reasoning for using it, and its advantages over IPMI
• Security Protocol and Data Model
  ◦ efficient access to low-level security capabilities and operations
  ◦ establish trust between onboard components
  ◦ establish encrypted/authenticated communication channel
  ◦ can be used by other DMTF-defined mechanisms
• Implemented in OpenBMC via libspdm
  ◦ https://github.com/DMTF/libspdm
• OpenBMC presentation on SPDM
  ◦ https://www.youtube.com/watch?v=PmgXkLJYI-E
• Specification
  ◦ https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_1.2
- A single huge repository with multiple meta-layers
  - poky / OE layers committed directly (not as submodules)
- Layers with common OpenBMC recipes (such as meta-phosphor)
- Common BSP layers (such as meta-aspeed, meta-openpower)
- OEM layers for specific boards (such as meta-ibm/meta-romulus)
• Developer documentation:
  – [https://github.com/openbmc/docs/blob/master/development/README.md](https://github.com/openbmc/docs/blob/master/development/README.md)
  – it recommends VM running Ubuntu 18.04
• We have proceeded with Docker container based on Ubuntu 20.04
  – [https://github.com/3mdeb/yocto-docker](https://github.com/3mdeb/yocto-docker)

• Quick start commands

```bash
source setup romulus
bitbake obmc-phosphor-image
runqemu slirp nographic
```
Adding Supermicro X11SSH-TF

- Old patchset from ML
  - https://lists.ozlabs.org/pipermail/openbmc/2017-October/009561.html
- Existing code for X11SPI
  - https://github.com/openbmc/openbmc/tree/master/meta-supermicro/meta-x11spi

Some typical porting changes
  - syntax, paths, licenses changes
• Deprecated workbook
  - https://github.com/openbmc/entity-manager
  - https://github.com/openbmc/phosphor-inventory-manager
  - https://github.com/openbmc/skeleton/tree/master/libopenbmc_intf
• Deprecated power control scripts
  - https://github.com/openbmc/x86-power-control
  - https://github.com/openbmc/phosphor-state-manager
- Code
  - https://github.com/3mdeb/openbmc/commits/supermicro-x11ssh-f-2.14.0
- Running in QEMU via supermicrox11-bmc machine

```bash
qemu-system-arm -machine supermicrox11-bmc
  -drive file=./build/x11ssh/tmp/deploys/images/x11ssh/obmc-phosphor-image-x11ssh.static.mtd,format=raw,if=mtd
  -m 256
  -nographic
  -net nic
  -net user,hostfwd=:127.0.0.1:2222-:22,hostfwd=:127.0.0.1:2443-:443,hostfwd=udp:127.0.0.1:2623-:623,hostname=qemu
```

```
[ 5.006442] systemd[1]: Failed to set pretimeout_governor to 'panic': No such file or directory
[ 5.006925] systemd[1]: Failed to set watchdog pretimeout governor to 'panic', ignoring: No such file or directory
[ 6.985448] systemd[1]: Failed to put bus name to hashmap: File exists
[ 6.986164] systemd[1]: xyz.openbmc_project.State.Host@0.service: Two services allocated for the same bus name
  xyz.openbmc_project.State.Host0, refusing operation.
  sys-subsystem-net-devices-%i.device
[ 8.935897] systemd[1]: Failed to isolate default target: Unit xyz.openbmc_project.State.Host@0.service failed to load
  properly, please adjust/correct and reload service manager: File exists
[ 9.071687] watchdog: watchdog0: watchdog did not stop!
[!!!!!!!] Failed to isolate default target.
```
Next steps

- Successful QEMU start
- Flashing and testing on hardware
- BMC RX for debugging
  - [https://github.com/Keno/bmcnonsense/blob/master/blog/03-serial2.md](https://github.com/Keno/bmcnonsense/blob/master/blog/03-serial2.md)
- External flashing procedure
  - [https://github.com/Keno/bmcnonsense/blob/master/blog/05-flashing3.md](https://github.com/Keno/bmcnonsense/blob/master/blog/05-flashing3.md)
- Credits to Keno for leaving valuable information
- OpenBMC intro and porting
- Talk on Intel Server firmware based on OpenBMC
  - [https://www.youtube.com/watch?v=i1FiOexyKTI](https://www.youtube.com/watch?v=i1FiOexyKTI)
- OpenBMC YouTube channel with interesting videos
  - [https://www.youtube.com/@openbmc9752/videos](https://www.youtube.com/@openbmc9752/videos)
- Other OpenBMC presentations
  - [https://github.com/openbmc/openbmc/wiki/Presentations](https://github.com/openbmc/openbmc/wiki/Presentations)
- Redfish talk from ELC
  - [https://www.youtube.com/watch?v=nBCjuuOjxRQ](https://www.youtube.com/watch?v=nBCjuuOjxRQ)
- Aspeed boards supported in QEMU
  - [https://www.qemu.org/docs/master/system/arm/aspeed.html](https://www.qemu.org/docs/master/system/arm/aspeed.html)
We are open to cooperate and discuss

- contact@3mdeb.com
- facebook.com/3mdeb
- @3mdeb_com
- linkedin.com/company/3mdeb
- https://3mdeb.com
- Book a call
- Sign up for the newsletter

Feel free to contact us if you believe we can help you in any way. We are always open to cooperate and discuss.