

# MemVerge Memory Viewer

## 1.2.1 Installation Guide

### MemVerge Memory Viewer 1.2.1 Installation Guide

- 1.0) Install hwloc from the source code
  - 1.1) Install the prerequisites
  - 1.2) Download and build hwloc
- 2.0) Install the Platform Telemetry Tool
  - 2.1) Install the Intel PCM Utility
  - 2.2) Install the AMD uProf Utility
    - 2.2.1) Install BCC
    - 2.2.2) Install AMD uProf
- 3.0) Install Memory Viewer
  - 3.1) Download Memory Viewer
  - 3.2) Install or Upgrade Memory Viewer
- 5.0) Start Memory Viewer
- 6.0) Activate the Memory Viewer License
  - 6.1) Server has internet connectivity
  - 6.2) Server has no internet connection.
- 7.0) CXL Topology JSON File
- 8.0) Custom Logo for White Labelling
- 9.0) Create a Systemd Service for Memory Viewer
- 10.0) Build Intel PCM from Source Code
- 11.0) Changelog

## 1.0) Install hwloc from the source code

## 1.1) Install the prerequisites

Ubuntu 22.04:

```
$ sudo apt install build-essential numactl libnuma1 libnuma-dev  
libpciaccess-dev libpciaccess0 libxml2 libxml2-dev cpuid  
libcpuid-dev libpci-dev libpci3
```

Fedora 36:

```
$ sudo dnf group install "C Development Tools and Libraries"  
$ sudo dnf install numactl numactl-libs numactl-devel  
libpciaccess libpciaccess-devel libxml2 libxml2-devel cpuid  
libcpuid-devel
```

CentOS/RHEL:

```
$ sudo yum groupinstall 'Development Tools'  
$ sudo yum install numactl numactl-libs numactl-devel  
libpciaccess libpciaccess-devel libxml2 libxml2-devel cpuid  
libcpuid-devel
```

## 1.2 Download and build hwloc

The instructions configure `hwloc` to install to `/opt/hwloc`, which can be changed.

```
$ git clone https://github.com/open-mpi/hwloc
$ cd hwloc
$ ./autogen.sh
$ ./configure --prefix=/opt/hwloc
$ make -j all
```

Install hwloc to `/opt/hwloc`

```
$ sudo make install
```

Verify the `lstopo` command version is 3.0 or later

```
$ sudo /opt/hwloc/bin/lstopo-no-graphics --version
lstopo-no-graphics 3.0.0a1-git
```

Check the Kernel config file to include DDR Modules in the output.

```
$ grep CONFIG_DMI_SYSFS /boot/config-$(uname -r)
CONFIG_DMI_SYSFS=y
```

If the value is `=y`, then no further action is needed. If the value is `=m` (module), load the `dmi-sysfs` driver if it's not already loaded:

```
$ sudo modprobe dmi-sysfs
```

Update the root users PATH environment variable to source the lstopo installed in `/opt` first

```
# vim ~/.bashrc
export PATH=/opt/hwloc/bin:$PATH
export LD_LIBRARY_PATH=/opt/hwloc/lib:$LD_LIBRARY_PATH
# source ~/.bashrc
```

Run `lstopo-no-graphics` to display the system topology, for example:

```
$ which lstopo-no-graphics
/opt/hwloc/bin/lstopo-no-graphics
$ sudo lstopo-no-graphics --no-caches --no-icaches --no-smt --
no-useless-caches --verbose
```

## 2.0) Install the Platform Telemetry Tool

For Intel, use Step 2.1.

For AMD, use Step 2.2

## 2.1) Install the Intel PCM Utility

**Note:** These public 'pcm' packages may not support CXL. See Section 10 to compile PCM from the source code.

On Intel platforms, install the Intel PCM (<https://github.com/intel/pcm>) utility:

**Ubuntu/Debian:**

```
$ sudo apt install pcm
```

**openSUSE:**

```
$ sudo zypper install pcm
```

**RHEL8.5 or later:**

```
sudo dnf install pcm
```

**Fedora:**

```
sudo yum install pcm
```

## 2.2) Install the AMD uProf Utility

### 2.2.1) Install BCC

On AMD platforms, the AMD uProf package requires **BCC**. To install the BCC tools before installing the AMD uProf tool, run:

**Fedora/RHEL/CentOS:**

```
$ sudo dnf install bcc bcc-devel bcc-tools
```

**Ubuntu:**

```
$ sudo apt install bpfcc-tools libbpfcc-dev
```

### 2.2.2) Install AMD uProf

Download the [AMD uProf](https://www.amd.com/en/developer/uprof.html) (<https://www.amd.com/en/developer/uprof.html>) RPM or DEB package file from the AMD website after agreeing to the EULA.

Install the package

**Fedora/RHEL/CentOS:**

```
$ sudo dnf install ./amduprof-4.0-341.x86_64.rpm
```

**Ubuntu:**

```
$ sudo apt install ./amduprof_4.0-341_amd64.deb
```

## 3.0) Install Memory Viewer

### 3.1) Download Memory Viewer

Download the Memory Viewer package from the MemVerge download site:

- **RHEL/CentOS/Fedora:** [mvmv-1.2.1-dragonfruit.x86\\_64.rpm](#)
- **Ubuntu:** [mvmv-1.2.1-dragonfruit.x86\\_64.deb](#)

If the host has an Internet connection, use `curl` or `wget` to download the file directly on the target server.

For example:

```
$ wget https://memory-viewer-  
bucket.s3.amazonaws.com/releases/1.2.1/mvmv-1.2.1-  
dragonfruit.x86_64.rpm
```

If the host does not have Internet access, download it and the dependency packages to a host that does, then transfer the packages to the target server. Install the dependency packages before installing MemVerge Memory Machine using the package manager command – `yum`, `apt`, `dnf`, etc.

MemVerge Memory Viewer package depends on the following:

- cxi-cli
- ipmctl
- libipmctl
- ndctl
- daxctl
- hwloc (lstopo)\*

<https://pkgs.org/> is a good place to find package names and download links for your Linux Distro Version.

\* Memory Viewer works best when using `lstopo` version 3.x, which is currently only available from the GitHub Project (<https://github.com/open-mpi/hwloc>).

## 3.2) Install or Upgrade Memory Viewer

Install or Upgrade Memory Viewer using the RPM or DEB package file appropriate for your Linux distribution. If the host has Internet access, it will automatically install and upgrade the dependency packages.

**RHEL/CentOS/Fedora:**

```
$ sudo dnf install ./mvmv-1.2.1-dragonfruit.x86_64.rpm
```

**Ubuntu:**



```
$ sudo apt install ./mvmv-1.2.1-dragonfruit.x86_64.deb
```

Example:

```
# sudo dnf install ./mvmv-1.2.1-dragonfruit.x86_64.rpm
Last metadata expiration check: 1:40:19 ago on Tue 25 Jul 2023
11:12:47 AM PDT.
```

```
Dependencies resolved.
```

```
=====
=====
Package      Architecture  Version           Repository
  Size
=====
=====
```

```
Installing:
```

```
  mvmv        x86_64        1.2.1-dragonfruit @commandline
8.2 M
```

```
Installing dependencies:
```

```
  cxl-libs    x86_64        76.1-1.fc36       updates
    55 k
  ipmctl      x86_64        03.00.00.0468-3.fc36 updates
    89 k
  libipmctl   x86_64        03.00.00.0468-3.fc36 updates
    400 k
  ndctl       x86_64        76.1-1.fc36       updates
191 k
  pcm         x86_64        202212-0.fc36     updates
1.6 M
```

## Transaction Summary

```
=====
=====
```

```
Install 6 Packages
```

```
Total size: 11 M
```

```
Total download size: 2.3 M
```

```
Installed size: 34 M
```

```
Is this ok [y/N]: y
```

## 5.0) Start Memory Viewer

Memory Viewer runs a web server on port 8080. To start the server, run:

```
# mvmv server
Collecting system status...
Log is available at: /root/.mvmv/log/mvmv.log

Server started successfully.
To access the web application, open this URL in a browser:
  http://0.0.0.0:8080/
```

Use Ctrl-C to stop the server

To change the default IP address and port, run

```
mvmv server -a <IP>:<port>
```

To make the change permanent, edit `~/ .mvmv/mvmv.yaml` and change the `address` line to the required configuration, then restart the server:

```
address: 0.0.0.0:8080
```

If your Linux distro enables a firewall by default, most do, open the port for remote access, for example:

```
$ sudo firewall-cmd --add-port 8080/tcp --permanent  
$ sudo firewall-cmd --reload
```

Navigate to the server hostname or ip address and port in your browser, eg:

```
http://cx1server1.example.com:8080
```

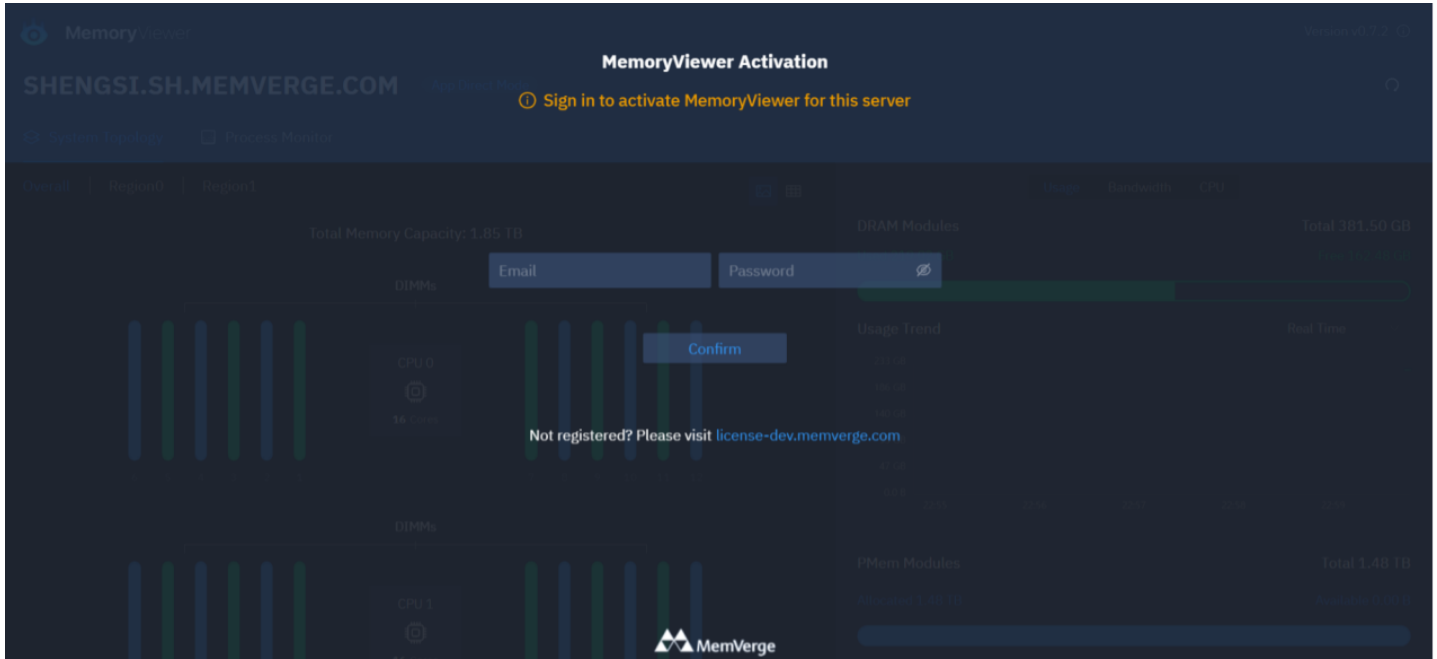
You will be presented with the License Wizard on the first visit.

## 6.0) Activate the Memory Viewer License

Memory Viewer is a licensed product from MemVerge. If the server has Internet connectivity, follow Step 6.1, if not, Step 6.2.

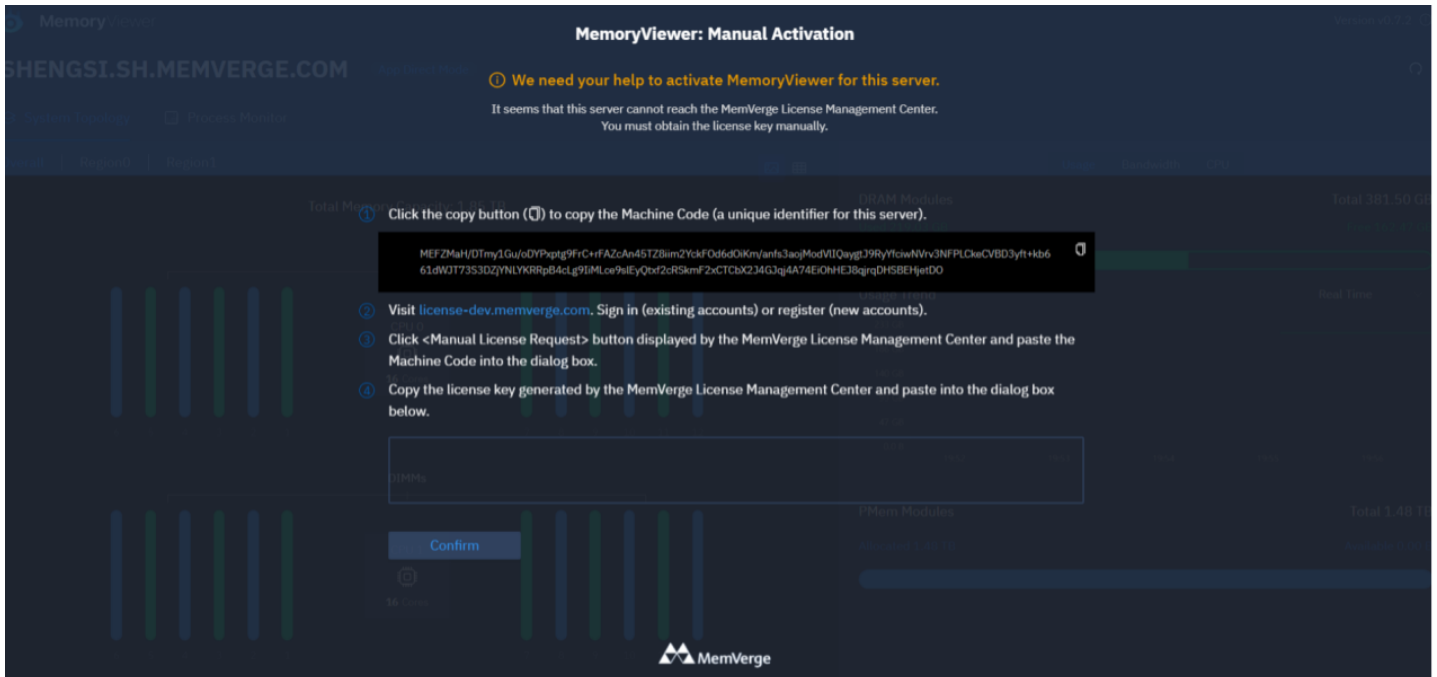
### 6.1) Server has internet connectivity

Enter an existing account username and password or register a new account for <https://license.memverge.com/>. The Memory Viewer server will automatically activate the license.



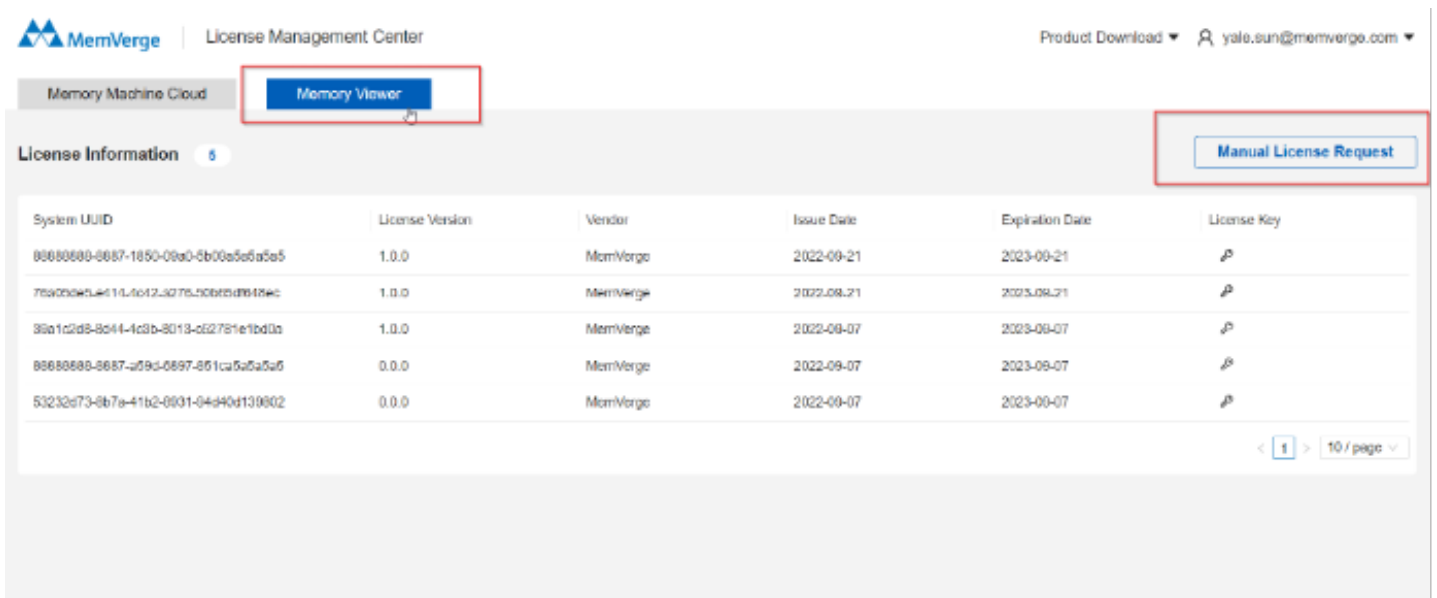
## 6.2) Sever has no internet connection.

6.2.1) Follow the on-screen instructions to manually get the license key and enter it into the box.



6.2.2) Visit <https://license.memverge.com/>, sign in with an existing account or register a new account, and manually get the license key following the steps.

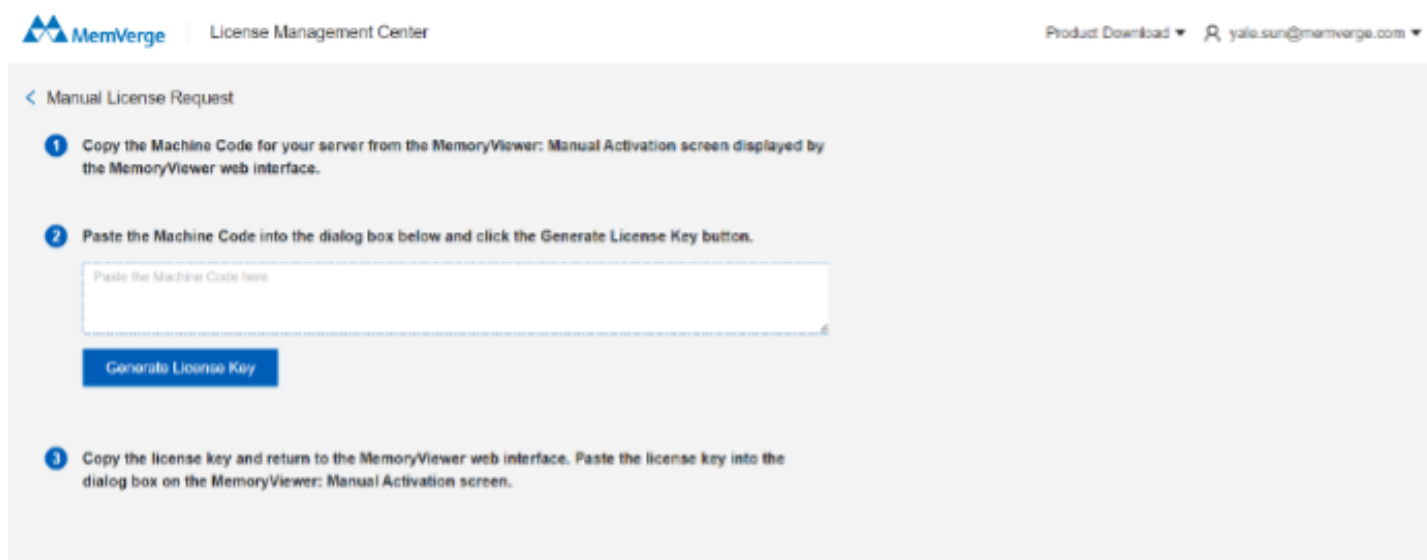
Click the 'Memory Viewer' tab, and click "Manual License Request"



Copy the Machine Code for your server from the 'Memory Viewer: Manual Activation' screen displayed in the browser interface.

Paste the Machine Code into the dialogue box and click "Generate License Key"

Copy the license key and return to the Memory Viewer browser interface. Paste the license key into the dialogue box in the 'Memory Viewer: Manual Activation' screen



MemVerge | License Management Center

Product Download | yafe.sun@memverge.com

### < Manual License Request

- 1 Copy the Machine Code for your server from the MemoryViewer: Manual Activation screen displayed by the MemoryViewer web interface.
- 2 Paste the Machine Code into the dialog box below and click the Generate License Key button.  
  
Generate License Key
- 3 Copy the license key and return to the MemoryViewer web interface. Paste the license key into the dialog box on the MemoryViewer: Manual Activation screen.

## 7.0) CXL Topology JSON File

When Memory Viewer cannot automatically determine the server's topology, a JSON file must be created to represent the correct server topology.

7.1) Create a CXL topology JSON file using the following format. If you use the system service described in section 9, the file should reside in `/root/.mvmv/cxlfile.json`. For example, two CXL devices connected to each CPU Socket would look similar to the following example:

```
$ sudo vim /root/.mvmv/cxlfile.json  
[
```

```
{
  "serialNumber": "",
  "firmwareVersion": "",
  "pmemSize": 0,
  "ramSize": 137438953472,
  "numaNode": 0,
  "address": "0000:df:00.0",
  "deviceName": "mem0",
  "pciDeviceName": "CXL: Micron Technology Inc Device
6400",
  "vendor": "Micron Technology Inc",
  "linkWidth": 8,
  "socket": 0
},
{
  "serialNumber": "",
  "firmwareVersion": "",
  "pmemSize": 0,
  "ramSize": 137438953472,
  "numaNode": 0,
  "address": "0000:2a:00.0",
  "deviceName": "mem1",
  "pciDeviceName": "CXL: Micron Technology Inc Device
6400",
  "vendor": "Micron Technology Inc",
  "linkWidth": 8,
  "socket": 1
}
]
```

7.2) Create and edit `/root/.mvmv/mvmv.yaml`, and add a **cxlFile** entry with the full path to 'cxlfile.json', for example:

```
$ vim ~/.mvmv/mvmv.yaml
address: 0.0.0.0:8080
.....
cxlFile: /root/.mvmv/cxlfile.json
```

## 8.0) Custom Logo for White Labelling

MemVerge Memory Viewer 1.2 allows company logos to be included in the interface. Replace the "custom-logo.png" file in the `/usr/bin/mvmvWeb/` directory with your company logo, keeping the same file name. The image height should be 30px. The width is less important.

```
# ls -l
asset-manifest.json
custom-logo.png <<<<< Replace this file with your company logo
index.html
logo.png
static
```

Your company logo will be shown on the header. For example, a customer logo for "CXL Memory Co" looks like this:



## 9.0) Create a Systemd Service for Memory Viewer

To manage the Memory Viewer web server, create a system service file to control it with 'systemctl'.

Create `/etc/systemd/system/mmv.service` with the following contents

```
$ sudo vim /etc/systemd/system/mmv.service
[Unit]
Description=MemVerge Memory Viewer
After=network.target

[Service]
Environment=HOME=/root
Environment=PATH=/opt/pcm/sbin:/usr/local/sbin:/usr/local/bin:/u
sr/sbin:/usr/bin:/sbin:/bin:/snap/bin
ExecStart=/usr/bin/mmv server

[Install]
WantedBy=default.target
```

Reload systemd to recognize the new service:

```
$ sudo systemctl daemon-reload
```

Start the service using the following command:

```
$ sudo systemctl start mvmv
```

Verify that the service is running without errors:

```
$ sudo systemctl status mvmv
```

Enable the Memory Viewer service to start automatically at system boot

```
$ sudo systemctl enable mvmv
```

## 10.0) Build Intel PCM from Source Code

The online 'pcm' package may not support CXL. It is best to compile CXL from the source code to get full functionality.

Install the prerequisite packages

**Ubuntu 22.04:**

```
$ sudo apt install build-essential cmake
```

### Fedora 36:

```
$ sudo dnf group install "C Development Tools and Libraries"  
$ sudo dnf install cmake
```

### CentOS/RHEL:

```
$ sudo yum groupinstall 'Development Tools'  
$ sudo yum install cmake
```

Clone the PCM GitHub repository and build

```
git clone --recursive https://github.com/intel/pcm  
cd pcm  
mkdir build  
cd build  
cmake -DCMAKE_INSTALL_PREFIX:PATH=/opt/pcm ..  
cmake --build . --parallel  
sudo cmake --install . --prefix /opt/pcm
```

If using the NDA version, download the [NDA patches](#) from Intel's website, then run:

```
tar zxvf PCM-NDA-20230512-191505-0a09b814.tgz
./NDA-PCM.sh
cd NDA-PCM/
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX:PATH=/opt/pcm ..
cmake --build . --parallel
sudo cmake --install . --prefix /opt/pcm
```

Remove the `./build` directory to clean up and rebuild, as there is no `cmake clean`.

Update the root users PATH in the `.bashrc` file and add or modify an entry for `PATH`. For example, add this to the end of the file:

```
$ sudo vim /root/.bashrc
[...snip...]
export PATH=/opt/pcm/sbin:$PATH
```

## 11.0) Changelog

07/25/2023

Memory Viewer v1.2.1 released:

- **RHEL/CentOS/Fedora:** [mvmv-1.2.1-dragonfruit.x86\\_64.rpm](#)
- **Ubuntu:** [mvmv-1.2.1-dragonfruit.x86\\_64.deb](#)

## Bugs Fixed in this release:

- MV-364: Support SK hynix Niagara CXL memory appliance
- MV-365: CXL Device Capacity is reported as `0Bytes`
- MV-366: CXL devices in `system-ram` node should report the NUMA `meminfo -> MemoryUsed` not 100% Used
- MV-367: When Memory Machine is running, the CXL device capacity is counted twice

**07/17/2023**

## Memory Viewer v1.2.0 released:

- **RHEL/CentOS/Fedora:** [mvmv-1.2.0-dragonfruit.x86\\_64.rpm](#)
- **Ubuntu:** [mvmv-1.2.0-dragonfruit.x86\\_64.deb](#)