RHEL 9 CXL 활용가이드

For Samsung SMRC

Table of Contents

1. 본 문서에 대하여

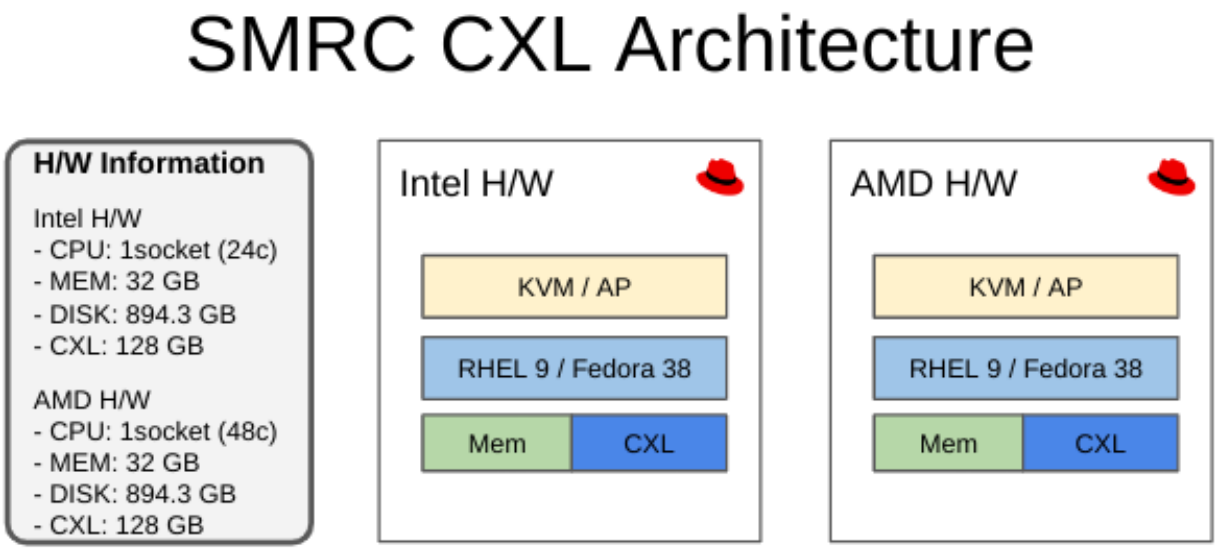
본 문서는 SMRC CXL 환경구성을 위해 Baremetal 환경의 AMD CPU 또는 INTEL CPU를 이용하여 CXL 정보, 구성, 기능, 사용법 등을 확인하는 목적으로 구성하는 방법을 가이드하기 위해 작성된 문서입니다.

2. 시스템 아키텍처

2.1 CXL 개요 및 환경구성 아키텍처

Machine Learning, Artificial Intelligence, IMDB 기술의 데이터 처리량 및 메모리 요구 사항이 기하급수적으로 증가하고 있기 때문에 더 큰메모리 대역폭과 용량을 갖춘 시스템에 대한 수요가 높아지고 있는 추세입니다. 그러나, CPU의 Core수가 증가하고 있음에도 불구하고 메모리 구성 컴퍼넌트는 확장성은 제한되고 있는 실정이어서 메모리의 확장성에 한계에 직면하고 있습니다.
이를 개선하기 위한 솔루션인 삼성전자의 최신 CXL Memory Expander는 최대 512GB의 DDR5 DRAM 메모리로 제공되어 서버 메모리 용량을 수십 테라바이트대로 확장하는 동시에 메모리 대역폭을 초당 몇 테라바이트대로 증가시킬 수 있으며, CXL 메모리 익스팬더는 x8 PCIe 5.0 인터페이스를 사용해서 레인당 최대 32GT/s의 전송 속도로 CPU에 연결할 수 있습니다.

이번 활용가이드에서 검증한 CXL 환경구성 아키텍처는 AMD CPU 또는 INTEL CPU가 장착된 SuperMicro Baremetal System 에 RHEL 9.3 Server OS를 구성하여 CXL 정보, 구성, 기능, 사용법 등을 확인하였으며, CXL Memory Expander 를 활용하여 가상머신, 어플리케이션에서 사용할 수 있는 기본적인 옵션을 설명합니다.



2.2 CXL 환경구성 정보

본 가이드의 기본환경은 Supermicro 시스템의 AMD, INTEL x86 CPU 기반으로 진행하였으며, CXL Expander Memory 의 Red Hat Enterprise Linux (RHEL) 환경에서 인식여부, 각종 Application 자원할당 등 다양한 시나리오로 진행을 하였습니다.

2.2.1 Baremetal Host information

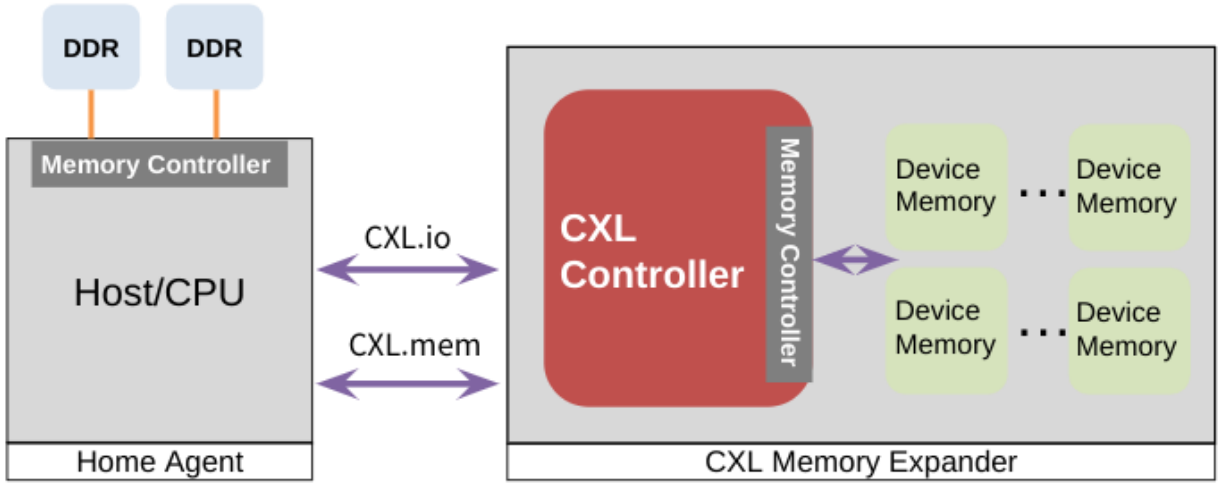
TYPE	OS Ver	Server Model	BIOS rev	CPU	Memory	Disk
AMD	RHEL 9.3	Supermicro Super Server (Base Board : H13SSF)	07/07/2023	48 core 1 Socket	32 GB (1 NUMA)	894.3 GB
INTEL	RHEL 9.3	Supermicro SSG-121E-NE316R (Base Board : X13DSF-A)	08/14/2023	24 core 1 Socket	32 GB (1 NUMA)	894.3 GB

2.2.2 CPU Model information

TYPE	CPU Model	OEM Strings
AMD	AMD EPYC 9454P 48-Core Processor	AMD EPYC Soc/Genoa Supermicro Motherboard-H13 Series
INTEL	Intel(R) Xeon(R) Gold 6442Y	Intel Sapphire Rapids/Emmitsburg/EagleStream Supermicro motherboard-X13 Series

2.2.3 Samsung CXL Expander information

TYPE	CXL Expander Model
AMD	CXL: Montage Technology Co., Ltd. Device c000
INTEL	CXL: Montage Technology Co., Ltd. Device c000



2.2.4 CXL Operating System (Linux) 호환성

Samsung CXL Expander (1.1) Memory 는 INTEL, AMD CPU 환경에서 인식이 가능하며, OS Kernel 의 CXL 지원여부를 사전에 파악해야 합니다. 아래는 검증을 위한 OS 지원표 입니다.

TYPE	BIOS Mode	Linux Version
AMD	SPM Disable (System RAM Type)	Red Hat Enterprise Linux 9.2 이상 (Kernel : 5.14.0-284)
		* **Fedora 37, 38** 등 대부분의 **O/S** 에서 메모리 인식
	SPM Enable (DAX HotPlug Type)	Red Hat Enterprise Linux 9.2 이상 (Kernel : 5.14.0-284)
		Fedora 37 이상 (Kernel : kernel-6.5.8-100)
INTEL	SPM Disable (System RAM Type)	**BIOS NOT Support**
	SPM Enable (DAX HotPlug Type)	Red Hat Enterprise Linux 9.2 이상 (Kernel : 5.14.0-284)
		Fedora 37 이상 (Kernel : kernel-6.5.8-100)

3. AMD, INTEL CPU 환경 CXL 구성

CXL 메모리 장치는 Flexbus 레인을 통해 CXL 루트 포트 또는 CXL 스위치 다운스트림 포트에 연결될 수 있으며, CXL 장치는 표준 PCIe 메커니즘을 사용하여 MMCFG 및 MMIO 영역에 매핑됩니다. 이를 위해 사전 BIOS 설정작업은 중요하며, 아키텍처 별 설정이 필요합니다.

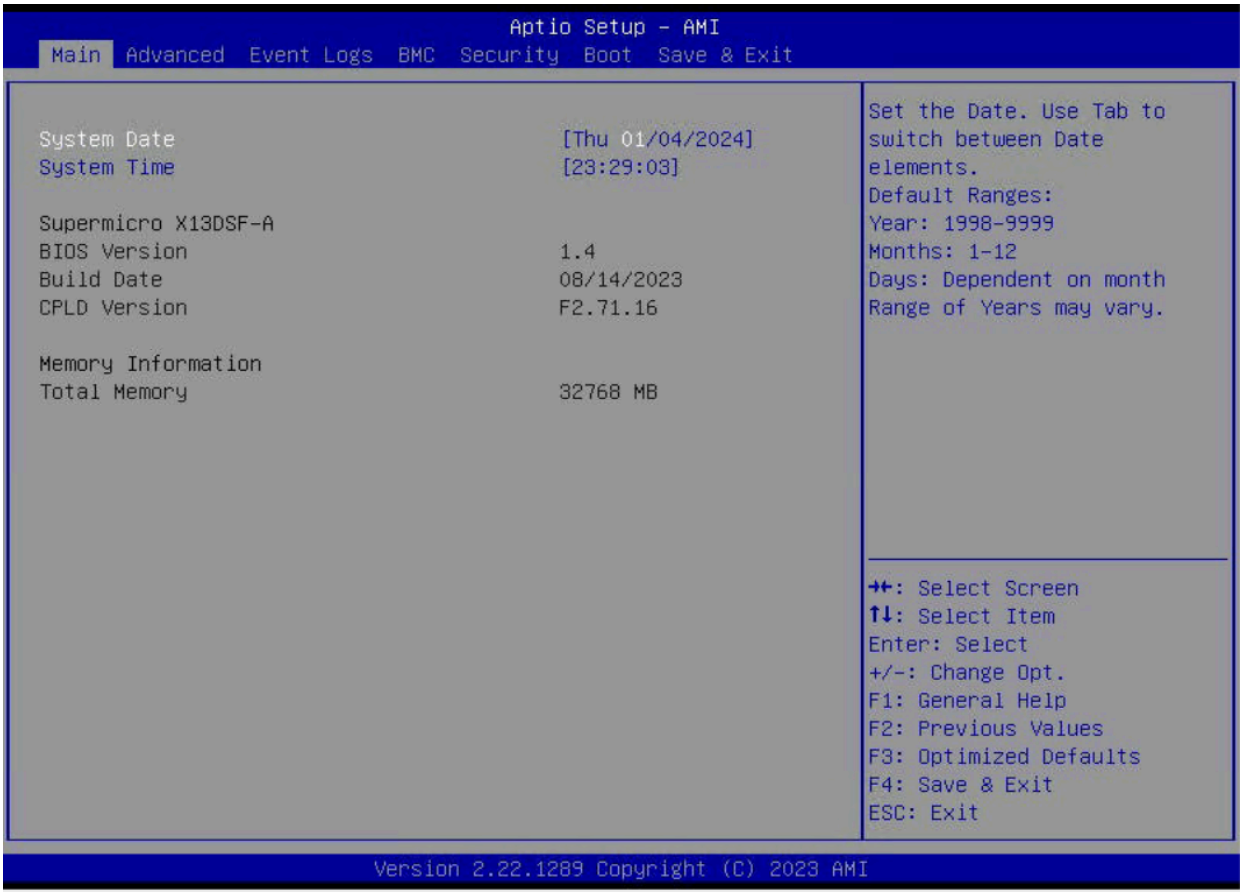
3.1 INTEL CPU 환경에 BIOS 구성

3.1.1 INTEL BIOS Version

CXL Memory Expander 를 지원하는 INTEL CPU 아키텍처의 경우 아래의 환경을 충족해야 합니다.

- ➔ Vendor : American Megatrends International, LLC.
- ➔ System Product Name : SSG-121E-NE316R
- ➔ Board Product Name : X13DSF-A (1.01)
- ➔ BIOS Version : 1.4 (Revision : 5.31)
- ➔ Build Date : 08/14/2023
- ➔ CPLD Version : F2.71.16

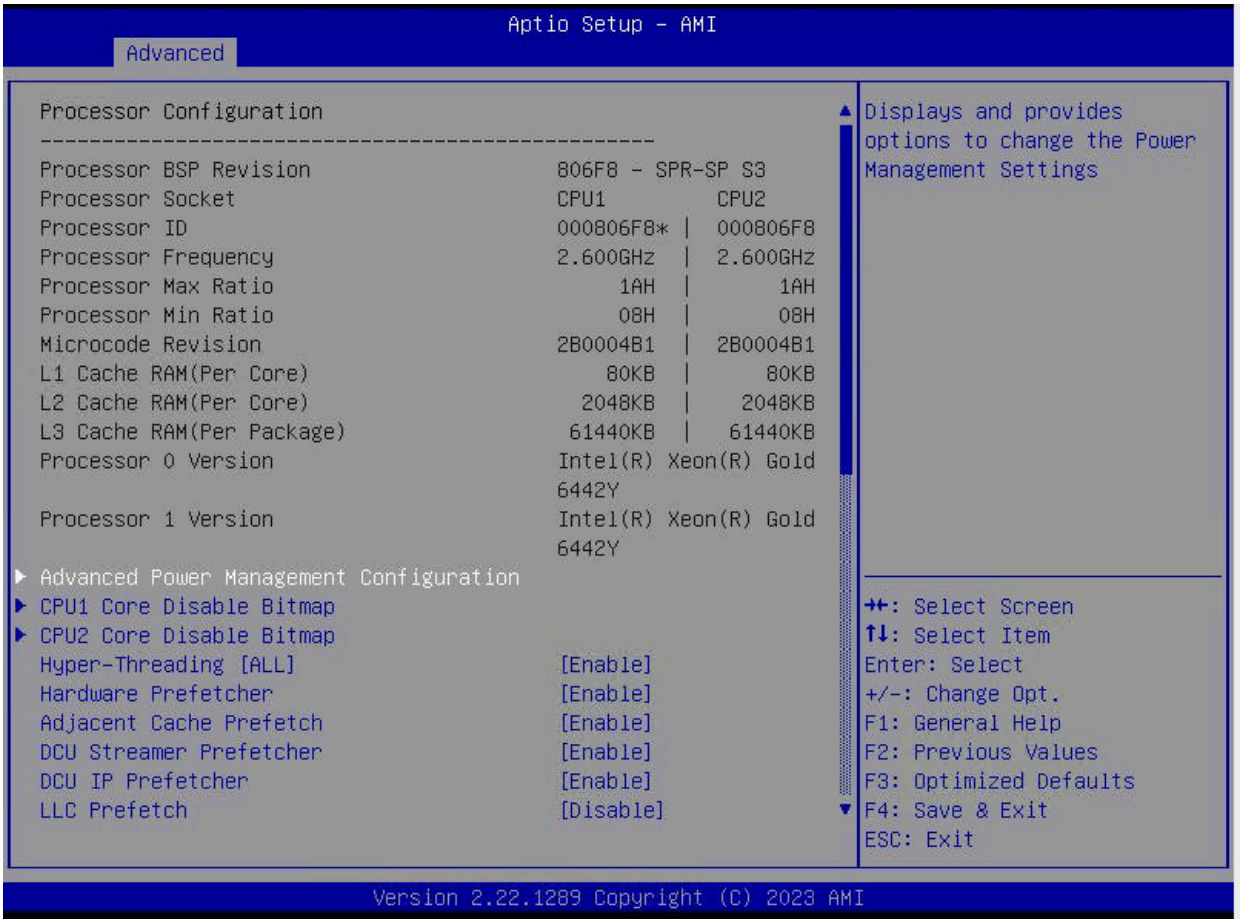
CXL Memory Expander를 지원하는 INTEL BIOS의 경우 Total Memory에 CXL Memory Expander 용량이 포함되는 경우가 있고 아닌 경우가 있습니다. 현재 활용가이드의 BIOS 환경에서는 CXL Memory Expander 용량을 포함하지 않은 Local Memory 용량만 보여 주고 있습니다.


Aptio Setup – AMI
Main Advanced Event Logs BMC Security Boot Save & Exit
System Date [Thu 01/04/2024]
System Time [23:29:03]
Supermicro X13DSF-A
BIOS Version 1.4
Build Date 08/14/2023
CPLD Version F2.71.16
Memory Information
Total Memory 32768 MB
Set the Date. Use Tab to switch between Date elements.
Default Ranges:
Year: 1998-9999
Months: 1-12
Days: Dependent on month
Range of Years may vary.
→←: Select Screen
↑↓: Select Item
Enter: Select
+/-: Change Opt.
F1: General Help
F2: Previous Values
F3: Optimized Defaults
F4: Save & Exit
ESC: Exit
Version 2.22.1289 Copyright (C) 2023 AMI

3.1.2 INTEL CPU 정보

INTEL Sapphire Rapids CPU 는 PCIe5.0 및 CXL(Compute Express Link) 1.1 상호 연결로 증가된 I/O 환경을 제공합니다.

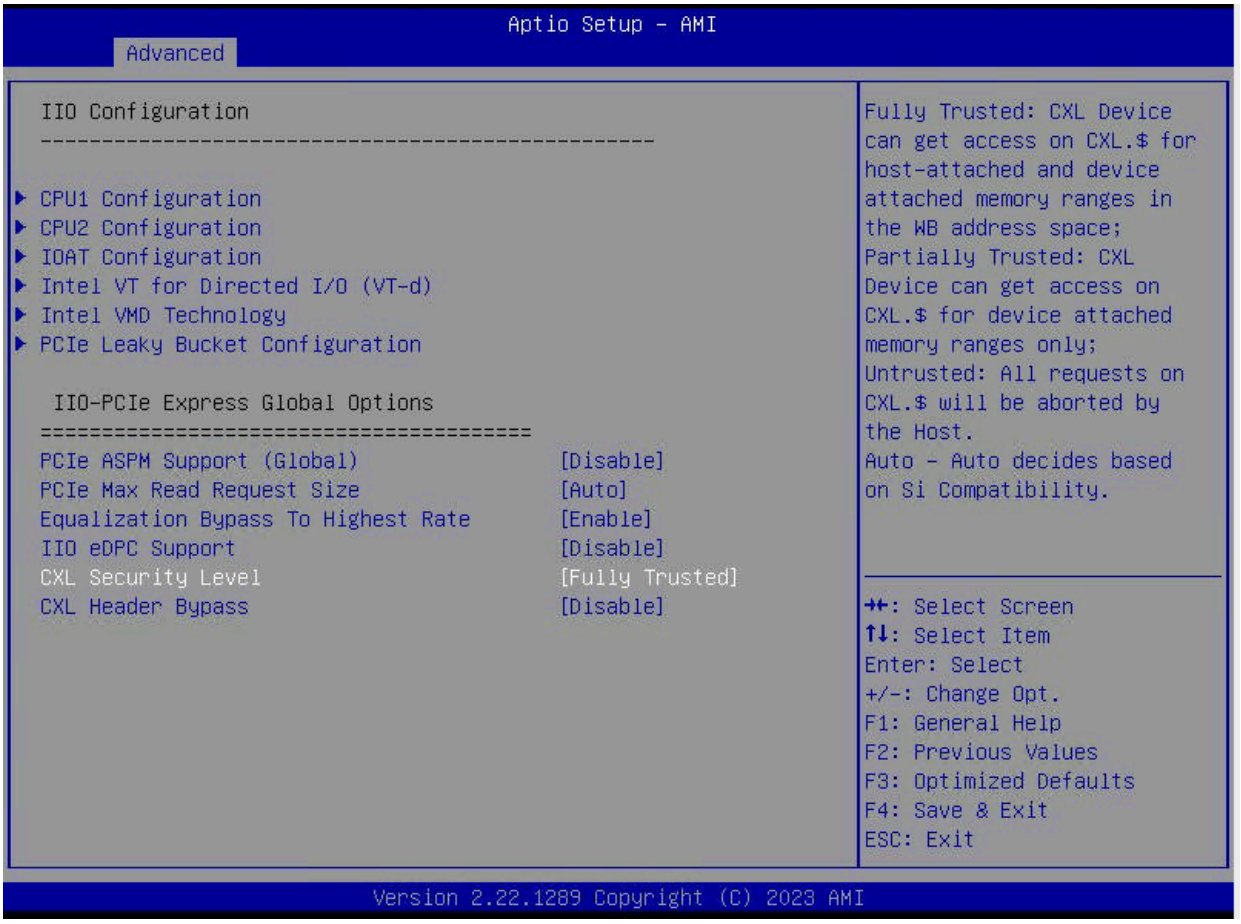
➔ CPU Model Name : Intel(R) Xeon(R) Gold 6442Y
➔ (INTEL Sapphire Rapids/Emmitsburg/EagleStream)
➔ Supermicro motherboard-X13 Series



3.1.3 INTEL CXL BIOS 설정

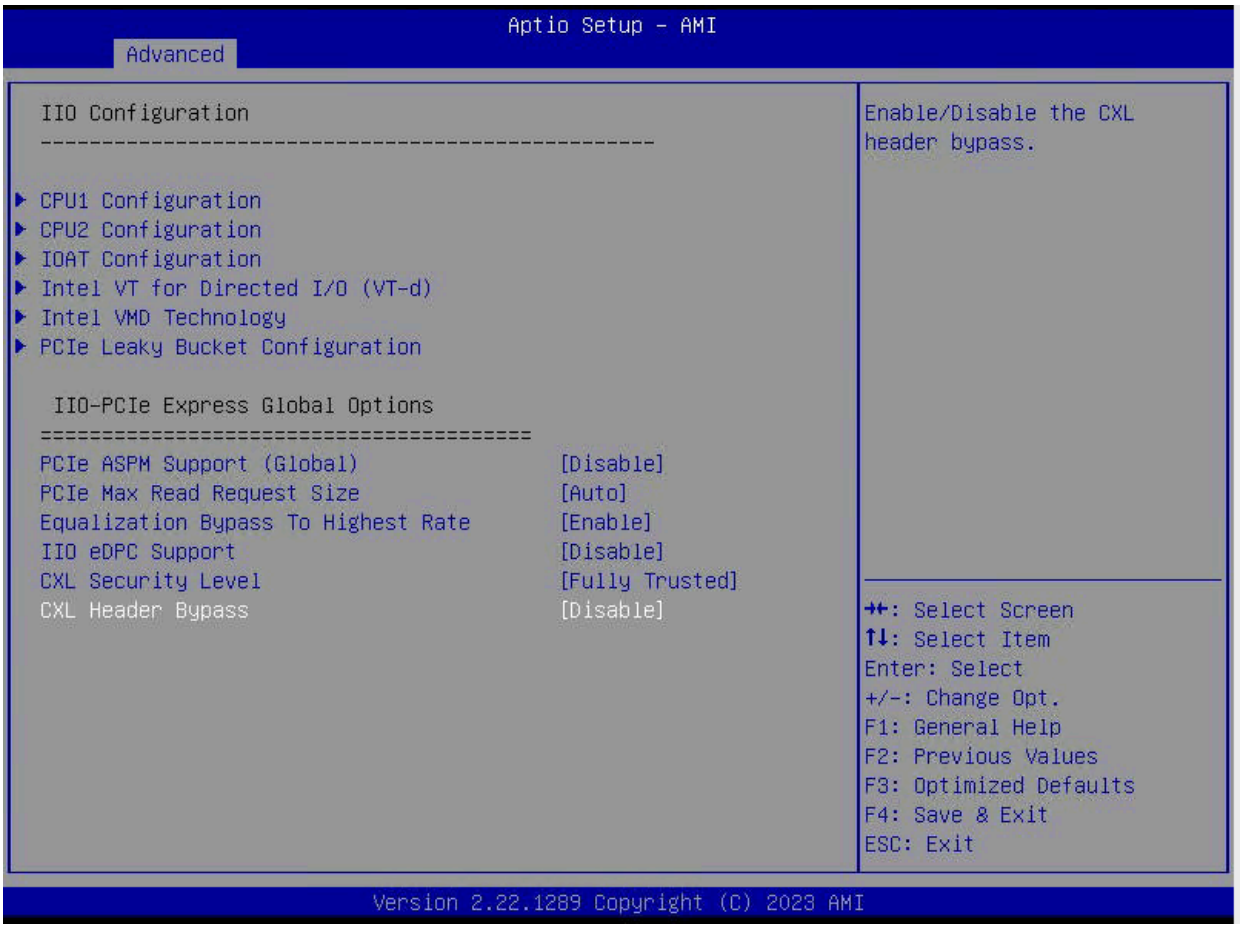
완전 신뢰모드**(Fully Trusted)** 로 설정하여 **CXL** 장치에 대한 보안수준을 설정합니다**.**

➔ **Advanced Menu ➔ Chipset Configuration ➔ North Bridge ➔ IIO Configuration ➔ CXL Security Level : [Fully Trusted]**



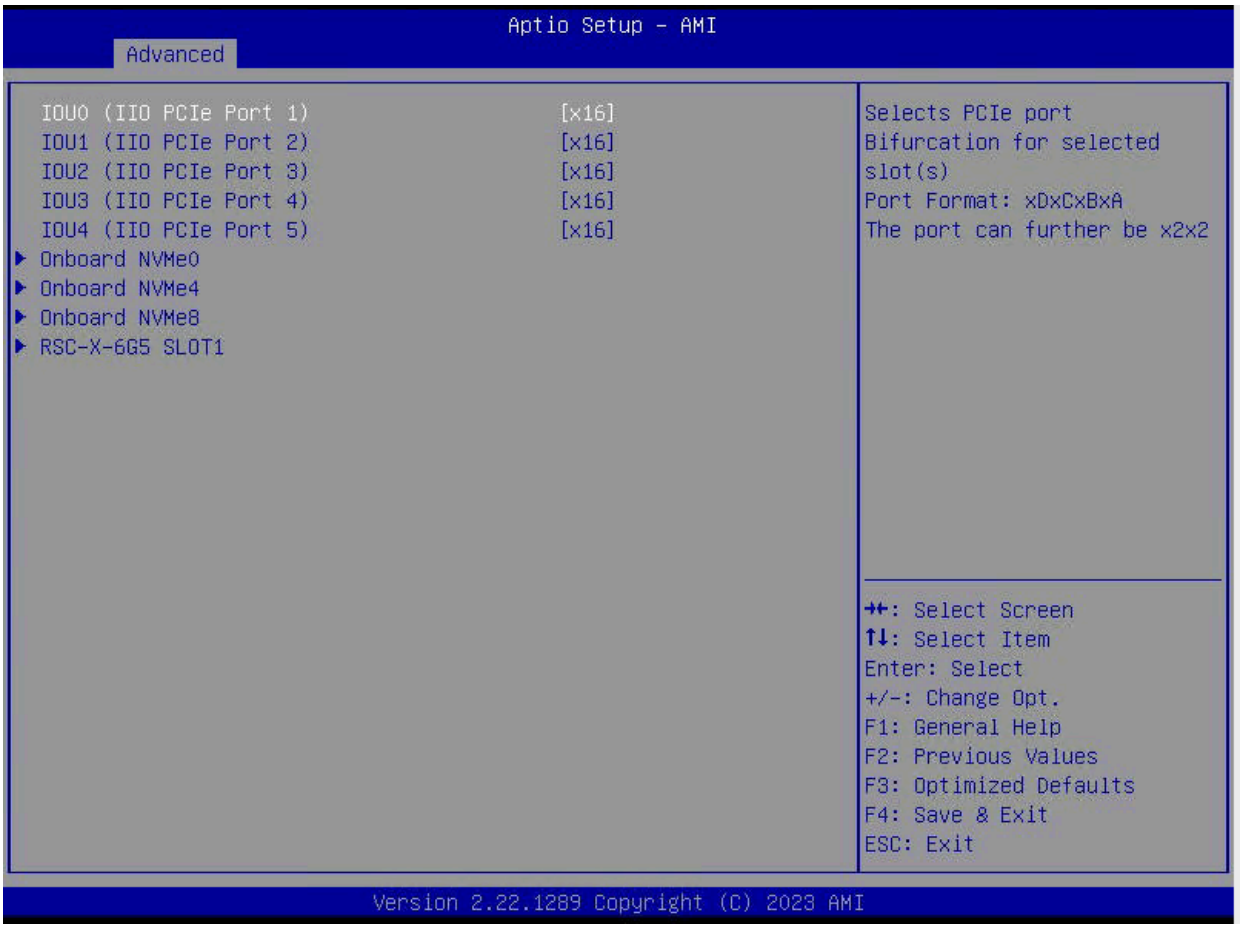
CXL 헤더우회를 비활성화 합니다.

➔ **Advanced Menu ➔ Chipset Configuration ➔ North Bridge ➔ IIO Configuration ➔ CXL Header bypass : [Disable]**



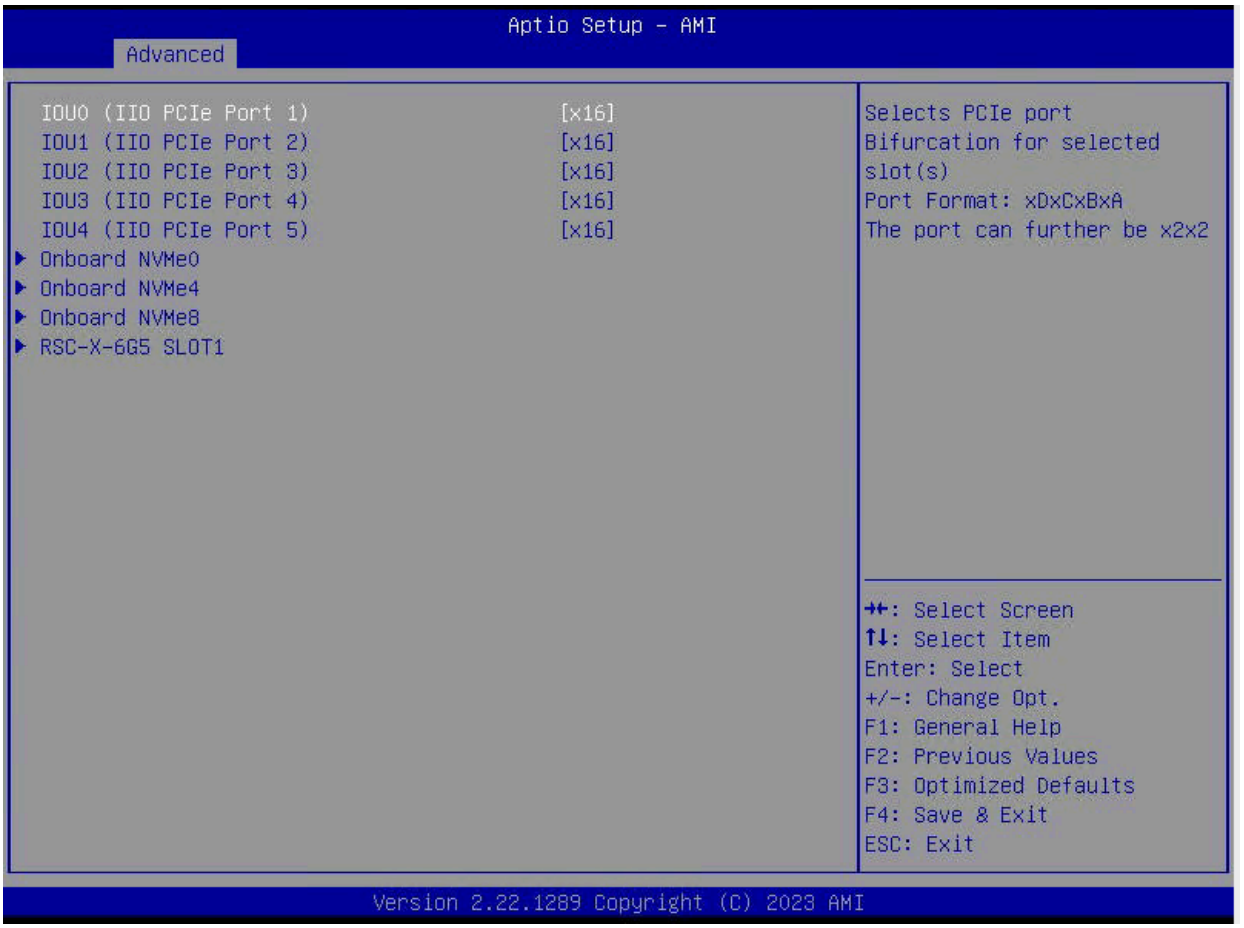
CPU0 에 대한 **PCIe** 포트의 분기 설정을 구성합니다.

➔ **Advanced Menu ➔ Chipset Configuration ➔ North Bridge ➔ IIO Configuration ➔ CPU0 Configuration ➔ IOU3 (IIO PCIe Port 3) : [x16]**



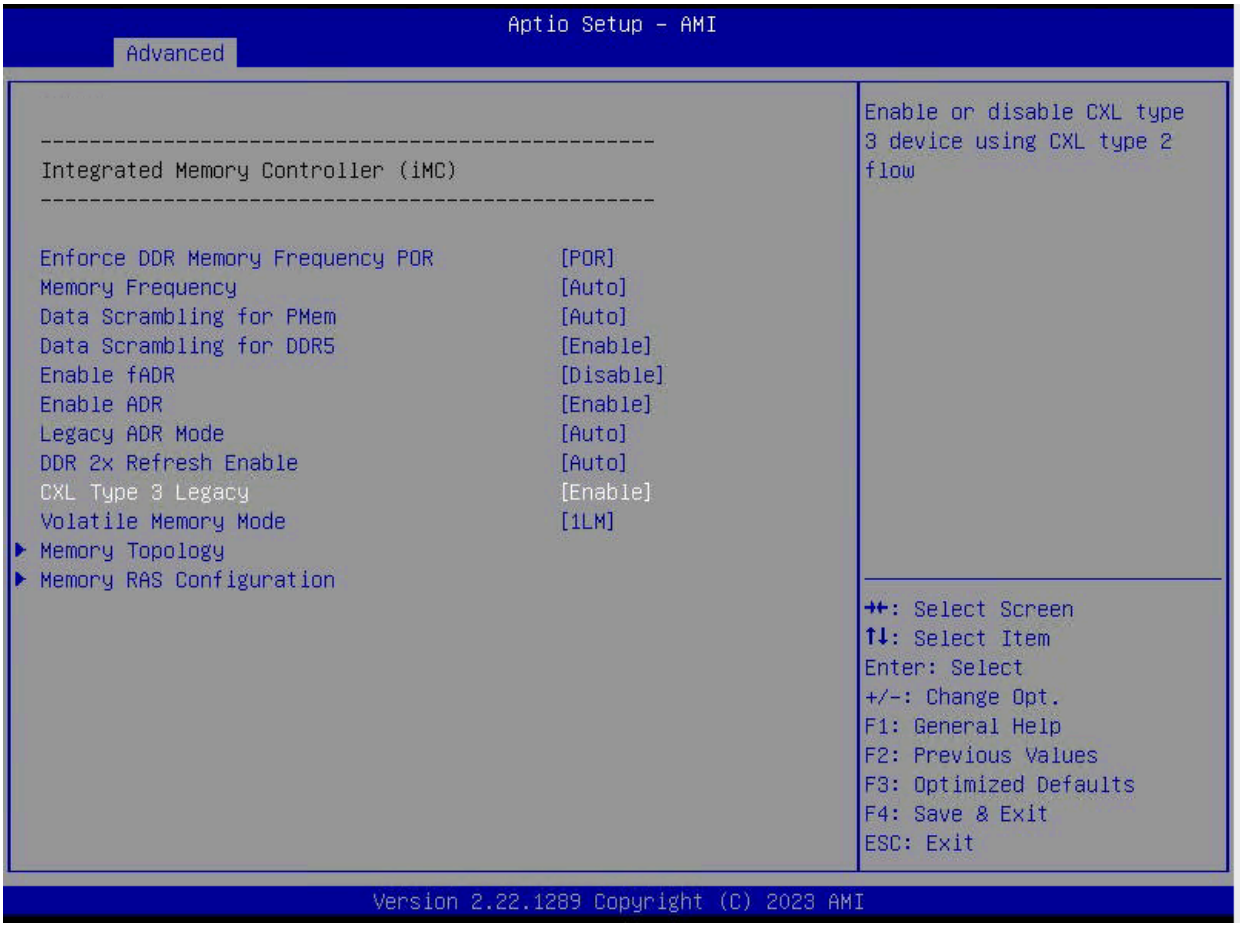
CPU1 에 대한 **PCIe** 포트의 분기 설정을 구성합니다.

➔ **Advanced Menu ➔ Chipset Configuration ➔ North Bridge ➔ IIO Configuration ➔ CPU1 Configuration ➔ IOU3 (IIO PCIe Port 3) : [x16]**



CXL Type 3 장치에 대한 레거시 지원을 활성화합니다.

➔ **Advanced Menu ➔ NB Configuration ➔ Memory Configuration ➔ CXL Type3 legacy : [Enable]**

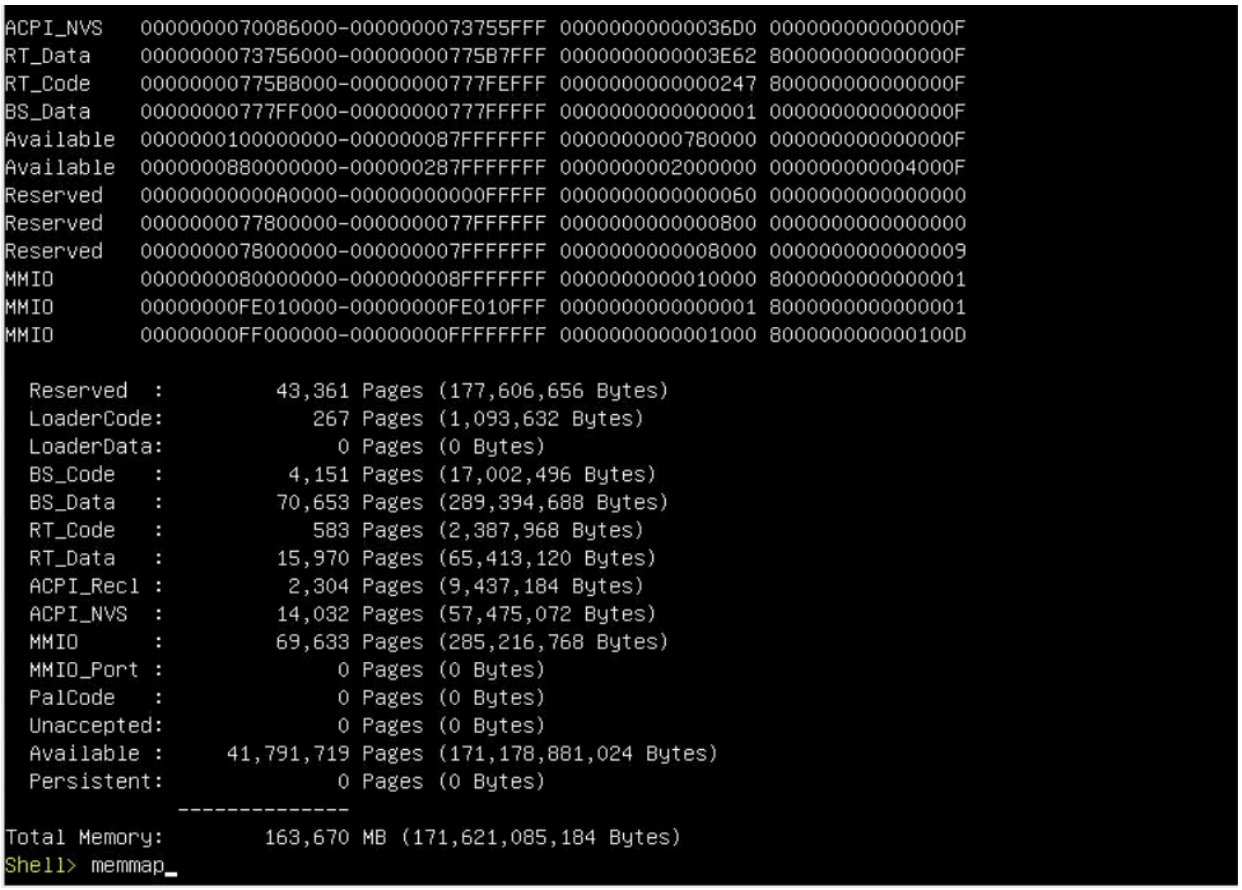


CXL EFI_MEMORY_SP 에 대한 설정을 진행합니다. **(Default : Enabled)**

➔ 이 설정은 CXL 메모리를 특수 목적 메모리로 설정합니다. INTEL 아키텍처의 경우 Enabled 설정만 가능합니다.
➔ Enabled : DAX Mode 사용
➔ Disabled : Direct System Memory 사용 (Intel 사용 불가-MB 기능 제공 불가)

3.1.4 INTEL BIOS UEFI Shell 환경 점검

모든 BIOS 설정 절차가 마무리 된 후 운영체제 설치 전 CXL Device 에 대한 인식여부를 BIOS UEFI Shell Mode 에서 확인할 수 있습니다.



이 설정은 BIOS의 UEFI Shell 진입 후 Local memory와 CXL memory가 정상적으로 인식 되었는지 확인 할 수 있습니다. UEFI Shell의 memmap 명령어를 이용하여 Local Memory 32GB 와 CXL Memory 128GB 가 인식되어 Total Memory 160GB가 인식된 것을 확인 할 수 있습니다.

- ➔ **Local memory : 32GB**
- ➔ **CXL memory : 128GB**

3.2 AMD CPU 환경에 BIOS 구성

3.2.1 AMD BIOS Version

Samsung CXL Memory Expander 를 지원하는 AMD 아키텍처의 경우 아래의 환경을 충족해야 합니다.

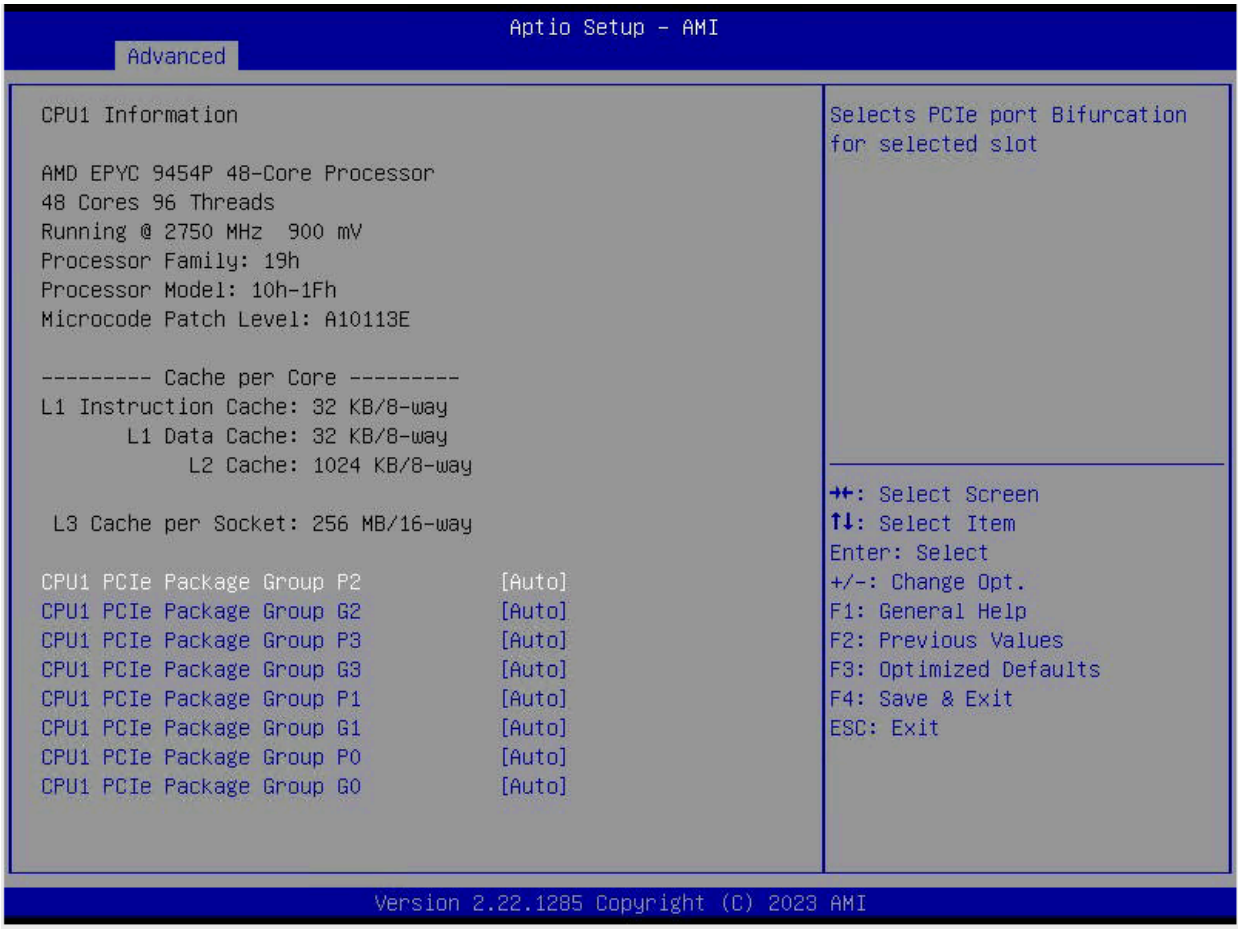
- ➔ Vendor : American Megatrends International, LLC.
- ➔ System Product Name : Super Server
- ➔ Board Product Name : H13SSF (1.01)
- ➔ BIOS Version : 1.0
- ➔ BIOS Revision : 5.27
- ➔ Build Date : 07/07/2023
- ➔ CPLD Version : F2.76.07

CXL Memory Expander를 지원하는 AMD BIOS의 경우 Total Memory에 CXL Memory Expander 용량이 포함되는 경우가 있고 아닌 경우가 있습니다. 현재 활용가이드의 BIOS 환경에서는 CXL Memory Expander 용량을 포함하지 않은 Local Memory 용량만 보여 주고 있습니다.

3.2.2 AMD CPU 정보

AMD EPYC Genoa CPU 는 PCIe5.0 및 CXL(Compute Express Link) 1.1 상호 연결로 증가된 I/O 환경을 제공합니다.

➔ CPU Model Name : AMD EPYC 9454P 48-Core Processor
(AMD EPYC Soc/Genoa)
➔ Supermicro Motherboard-H13 Series



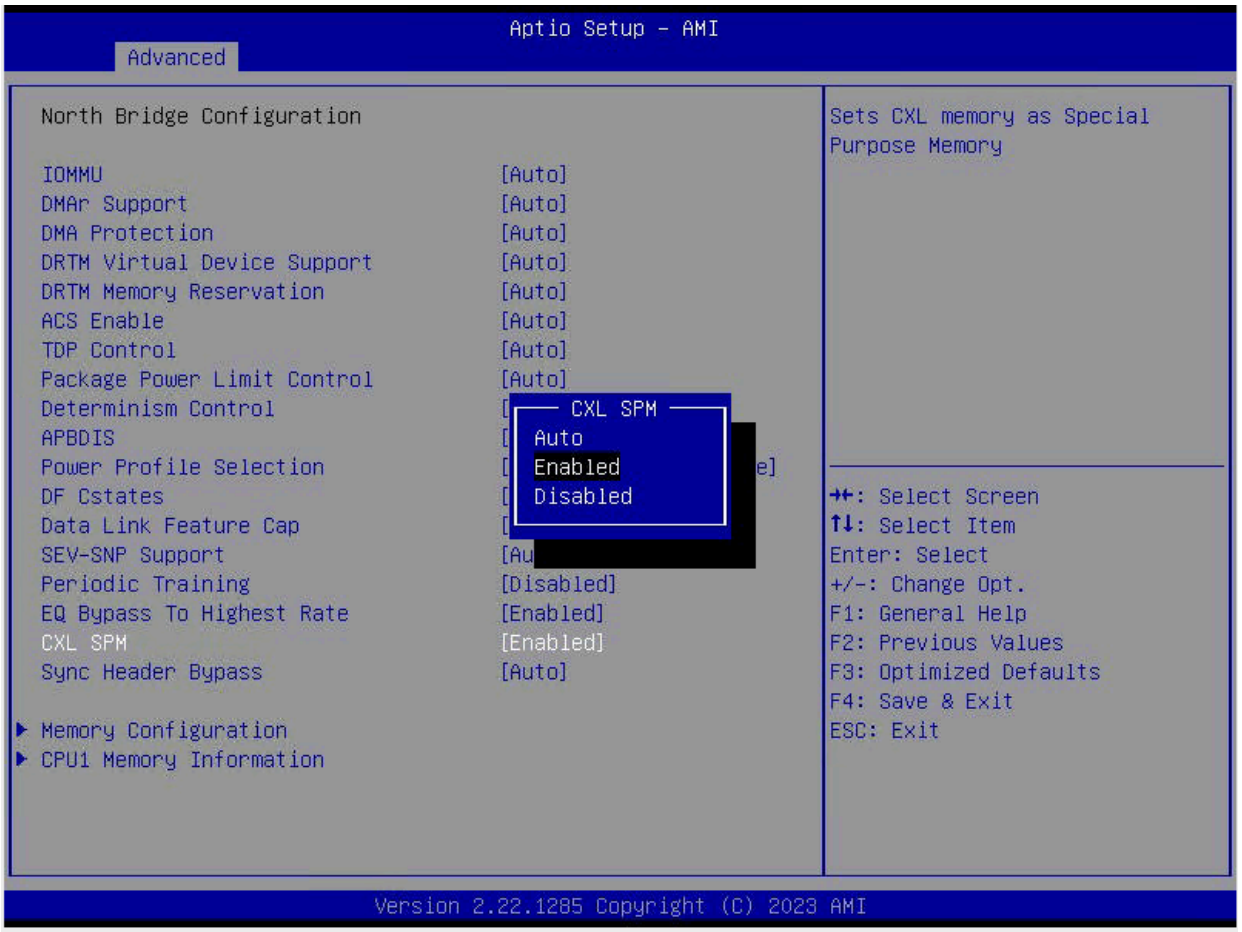
3.2.3 AMD CXL SPM Mode

이 설정은 CXL 메모리를 특수 목적 메모리로 설정합니다. 활성화/비활성에의 차이는 아래와 같습니다.

➔ **Advanced Menu ➔ NB Configuration ➔ CXL SPM : [Enabled]**

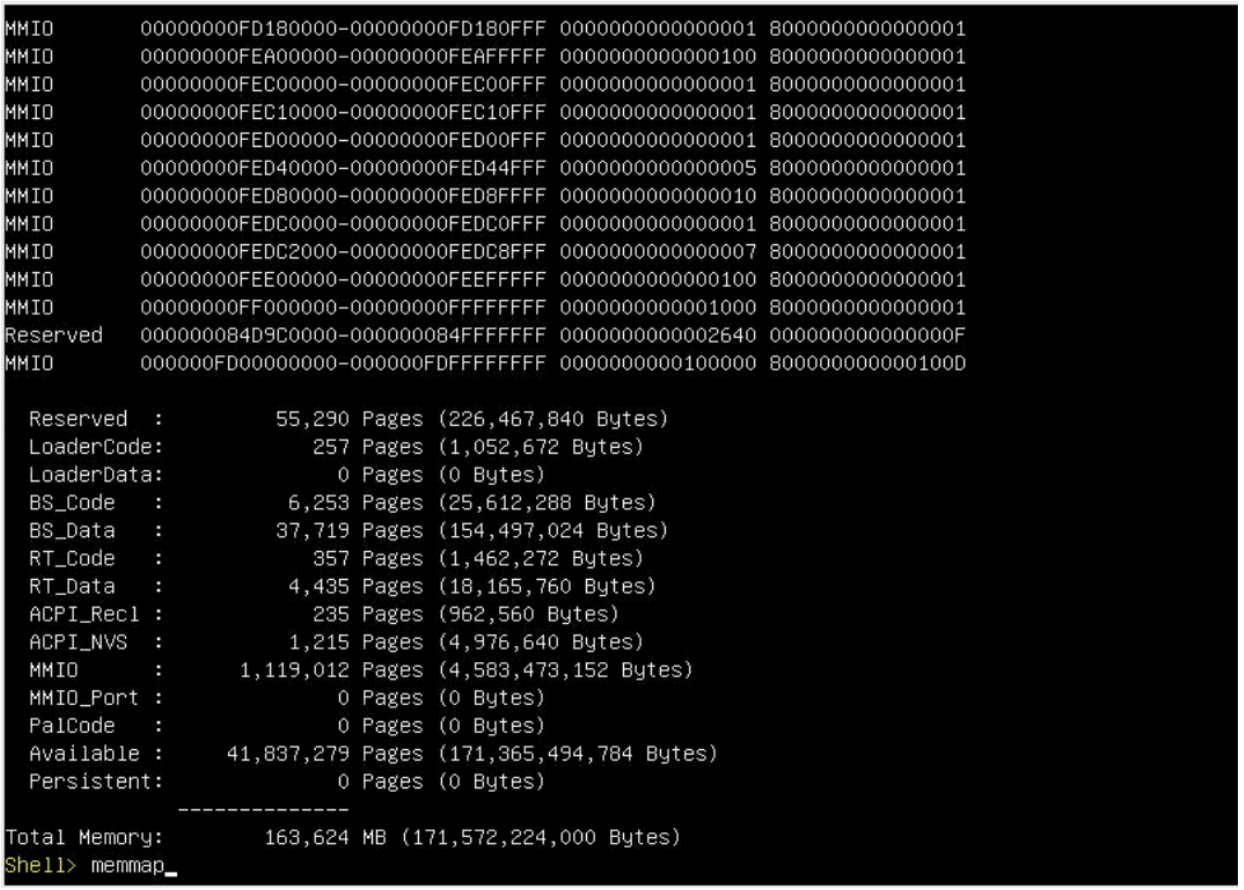
➔ **Enabled : DAX Mode** 사용
(OS 레벨에서 **CXL Expander**를 목적에 맞게 활성화 후 사용**)**

➔ **Disabled : Direct System Memory** 사용
(BIOS 레벨에서 **CXL Expander**를 사전 활성화 후 **OS** 에서 **Local memory** 인식 후 사용**)**



3.2.4 AMD BIOS UEFI Shell 환경 점검

모든 BIOS 설정 절차가 마무리 된 후 운영체제 설치 전 CXL Device 에 대한 인식여부를 BIOS UEFI Shell Mode 에서 확인할 수 있습니다.



이 설정은 BIOS의 UEFI Shell 진입 후 Local memory와 CXL memory가 정상적으로 인식 되었는지 확인 할 수 있습니다. UEFI Shell의 memmap 명령어를 이용하여 Local Memory 32GB 와 CXL Memory 128GB 가 인식되어 Total Memory 160GB가 인식된 것을 확인 할 수 있습니다.

- ➔ **Local memory : 32GB**
- ➔ **CXL memory : 128GB**

3.3 RHEL9 OS CXL Device 확인

BIOS 설정 및 RHEL9.3 운영체제 부팅 후 시스템 메모리를 확장합니다. 메모리 확장 후 기본적으로 확인해야 할 항목과 방법에 대해 알아봅니다.

3.3.1 RHEL9 CXL Device 확인

CXL volatile-memdev Type (Memory Expander) 구성 확인을 위해 OS의 커널 및 CXL H/W 정보를 확인합니다.

```
[root@rhel93-cxl ~]# uname -a

Linux rhel93-cxl 5.14.0-362.8.1.el9_3.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Oct 3
11:12:36 EDT 2023 x86_64 x86_64 x86_64 GNU/Linux
```

CXL 1.1 이상의 프로토콜을 사용하기 위해서는 CXL 지원이 가능한 CPU 확보가 중요합니다.

```
[root@rhel93-cxl ~]# lscpu
Architecture:            x86_64
  CPU op-mode(s):        32-bit, 64-bit
  Address sizes:         52 bits physical, 57 bits virtual
  Byte Order:            Little Endian
CPU(s):                  96
  On-line CPU(s) list:   0-95
Vendor ID:               AuthenticAMD
  BIOS Vendor ID:        Advanced Micro Devices, Inc.
  Model name:            AMD EPYC 9454P 48-Core Processor
    BIOS Model name:     AMD EPYC 9454P 48-Core Processor
    CPU family:          25
    Model:               17
    Thread(s) per core:  2
    Core(s) per socket:  48
    Socket(s):           1
    Stepping:            1
    Frequency boost:     enabled
    CPU max MHz:         3810.7910
    CPU min MHz:         1500.0000
    BogoMIPS:            5500.32
...
Virtualization features:
  Virtualization:        AMD-V
Caches (sum of all):
  L1d:                   1.5 MiB (48 instances)
  L1i:                   1.5 MiB (48 instances)
  L2:                    48 MiB (48 instances)
  L3:                    256 MiB (8 instances)
NUMA:
  NUMA node(s):          1
  NUMA node0 CPU(s):     0-95
Vulnerabilities:
  Gather data sampling:  Not affected
```

```
  Itlb multihit:         Not affected
  L1tf:                  Not affected
  Mds:                   Not affected
...
```

3.3.2 RHEL9 CXL Kernel Module 확인

CXL 관련 커널 모듈 cxl_mem, cxl_pci, dax_hmem, device_dax 등이 정상적으로 로딩이 되었는지 확인합니다.

```
[root@rhel93-cxl ~]# lsmod | grep -i -e cxl -e dax
cxl_mem                16384  0
device_dax             20480  0
cxl_port               16384  0
dax_hmem               16384  0
cxl_pci                36864  0
cxl_acpi               24576  0
cxl_core              167936  4 cxl_port,cxl_mem,cxl_pci,cxl_acpi
```

CXL 관련 커널 모듈 cxl_mem, cxl_pci, dax_hmem, device_dax 의 상세정보를 확인합니다. 기본적으로 RHEL9.3 이상 커널 환경에서는 CXL Expander Memory Device를 인식할 수 있는 드라이버가 포함되어 있습니다.

```
[root@rhel93-cxl ~]# modinfo cxl_mem cxl_pci device_dax dax_hmem

filename:
/lib/modules/5.14.0-362.8.1.el9_3.x86_64/kernel/drivers/cxl/cxl_mem.ko.xz
softdep:        pre: cxl_port
alias:          cxl:t5*
import_ns:      CXL
license:        GPL v2
rhelversion:    9.3
srcversion:     587493E932500A8B5B52605
depends:        cxl_core
retpoline:      Y
intree:         Y
name:           cxl_mem
vermagic:       5.14.0-362.8.1.el9_3.x86_64 SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Red Hat Enterprise Linux kernel signing key
sig_key:        1F:F0:D0:D7:72:A6:ED:12:E3:1B:CC:C9:E0:49:15:03:79:D5:1F:69
sig_hashalgo:   sha256
...
filename:
/lib/modules/5.14.0-362.8.1.el9_3.x86_64/kernel/drivers/cxl/cxl_pci.ko.xz
import_ns:      CXL
license:        GPL v2
rhelversion:    9.3
```

```
srcversion:     F295766BE87D4121C5232B8
alias:          pci:v*d*sv*sd*bc05sc02i10*
depends:        cxl_core
retpoline:      Y
intree:         Y
name:           cxl_pci
vermagic:       5.14.0-362.8.1.el9_3.x86_64 SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Red Hat Enterprise Linux kernel signing key
sig_key:        1F:F0:D0:D7:72:A6:ED:12:E3:1B:CC:C9:E0:49:15:03:79:D5:1F:69
sig_hashalgo:   sha256
...
filename:
/lib/modules/5.14.0-362.8.1.el9_3.x86_64/kernel/drivers/dax/device_dax.ko.xz
alias:          dax:t0*
license:        GPL v2
author:         Intel Corporation
rhelversion:    9.3
srcversion:     4372B92DBFE4ED5D9ACBC9F
depends:
retpoline:      Y
intree:         Y
name:           device_dax
vermagic:       5.14.0-362.8.1.el9_3.x86_64 SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Red Hat Enterprise Linux kernel signing key
sig_key:        1F:F0:D0:D7:72:A6:ED:12:E3:1B:CC:C9:E0:49:15:03:79:D5:1F:69
sig_hashalgo:   sha256
...
filename:
/lib/modules/5.14.0-362.8.1.el9_3.x86_64/kernel/drivers/dax/hmem/dax_hmem.ko.xz
author:         Intel Corporation
license:        GPL v2
alias:          platform:hmem*
rhelversion:    9.3
srcversion:     23AA81D251443E3A0BC0784
depends:
retpoline:      Y
intree:         Y
name:           dax_hmem
vermagic:       5.14.0-362.8.1.el9_3.x86_64 SMP preempt mod_unload modversions
sig_id:         PKCS#7
signer:         Red Hat Enterprise Linux kernel signing key
sig_key:        1F:F0:D0:D7:72:A6:ED:12:E3:1B:CC:C9:E0:49:15:03:79:D5:1F:69
sig_hashalgo:   sha256
...
```

PCI 정보를 확인하는 lspci 명령어를 수행하여 CXL 장치의 PCI BUS, PORT 및 Kernel Driver, Kernel Module 정보 등을 확인합니다.

```
[root@rhel93-cxl ~]# lspci -nnvv
7f:00.0 CXL [0502]: Montage Technology Co., Ltd. Device [1b00:c000] (rev 01)
(prog-if 10 [CXL Memory Device (CXL 2.x)])
```

```
        Control: I/O- Mem+ BusMaster- SpecCycle- MemWINV- VGASnoop- ParErr-
Stepping- SERR- FastB2B- DisINTx-
        Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort-
<MAbort- >SERR- <PERR- INTx-
        IOMMU group: 54
        Region 0: Memory at 3cc000000000 (64-bit, prefetchable) [size=16M]
        Region 2: Memory at 3ca000000000 (64-bit, prefetchable) [size=128G]
...
        Kernel driver in use: cxl_pci
        Kernel modules: cxl_pci
```

하드웨어 정보를 상세하게 확인하는 lshw 명령어를 수행하여 CXL 장치의 product, vendor, version과 같은 상세 정보와 Memory Slot의 정보를 확인 합니다.

```
[root@rhel93-cxl ~]# lshw
...
    *-memory:1
          description: CXL
          product: Montage Technology Co., Ltd.
          vendor: Montage Technology Co., Ltd.
          physical id: 11
          bus info: pci@0000:7f:00.0
          version: 01
          width: 64 bits
          clock: 33MHz (30.3ns)
...
```

3.3.3 RHEL9 CXL NUMA 확인

CXL Memory 인식 전 Memory 정보와 CXL Memory 인식 수행 후 Memory 및 NUMA의 변화와 커널의 buddyinfo 정보의 변화를 확인합니다.

CXL Memory 인식 전 Total Memory 용량과 NUMA 정보를 확인합니다.

```
[root@rhel93-cxl ~]# free -k
               total        used        free      shared  buff/cache   available
Mem:        32319884     3182724    18902080       37376    10736660    29137160
Swap:       16359420           0    16359420

[root@rhel93-cxl ~]# numactl -H
available: 1 nodes (0)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95
node 0 size: 31562 MB
node 0 free: 18251 MB
node distances:
node   0
```

```
  0:  10
```

CXL Memory 인식 전 dax device 정보를 확인 하고 daxctl 명령어를 수행하여 ONLINE 상에서 CXL 장치를 활성화 합니다. (Online Memory Hotplug 기능은 Kernel 에서 제공되야만 사용 가능한 기능입니다.)

```
[root@rhel93-cxl ~]# daxctl list
[
  {
    "chardev":"dax0.0",
    "size":137438953472,
    "target_node":1,
    "align":2097152,
    "mode":"devdax"
  }
]

[root@rhel93-cxl ~]# daxctl reconfigure-device --no-movable --mode=system-ram
dax0.0
[
  {
    "chardev":"dax0.0",
    "size":137438953472,
    "target_node":1,
    "align":2097152,
    "mode":"system-ram",
    "online_memblocks":1024,
    "total_memblocks":1024,
    "movable":false
  }
]
```

Normal 영역 **:**
Normal 영역은 일반적인 커널 메모리 할당에 사용됩니다. 이 영역은 주로 정적인 커널 데이터 및 코드를 저장하는 데 사용됩니다. 이러한 데이터 및 코드는 시스템 부팅 시에 할당되고 메모리 해제가 거의 발생하지 않는 특성을 가집니다. Normal 영역의 메모리는 커널 초기화 및 기본 동작을 지원하는 데 사용됩니다.

Movable 영역 **:**
Movable 영역은 커널에서 동적으로 할당 및 해제되는 메모리를 관리하는 데 사용됩니다. 이러한 동적 할당 및 해제는 커널의 특정 동작 및 기능에 필요합니다. Movable 영역은 주로 커널의 "slab allocator"와 같은 메모리 할당자에서 사용됩니다. 이 영역은 커널 내부에서 객체 또는 자료구조의 동적 할당 및 해제를 처리하는 데 사용됩니다. Movable 영역의 주요 특징은 메모리 단편화를 최소화하기 위해 메모리 블록을 이동시킬 수 있는 유연성을 제공한다는 것입니다.

daxctl 명령어를 이용하여 CXL Memory 인식 후 Memory 용량과 NUMA 정보를 확인 합니다. 이때 CXL 장치의 Memory 용량이 System Memory에 증가가 되었는지 확인하고 NUMA 정보를 확인 시 CPU가 없는 ZERO CPU NUMA NODE가 추가된 것이 확인이 되어야 합니다. (CXL 메모리 디바이스 daxctl online-memory 수행 결과 ZERO cpu numa node 1가 생성되는 것을 확인)

```
[root@rhel93-cxl ~]# free -k
               total        used        free      shared  buff/cache   available
Mem:       166537612     5273884   151224072       45572    11012336   161263728
Swap:       16359420           0    16359420

[root@rhel93-cxl ~]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95
node 0 size: 31562 MB
node 0 free: 16608 MB
node 1 cpus:              ← CXL numa node 1
node 1 size: 131072 MB    ← CXL numa node 1
node 1 free: 131071 MB    ← CXL numa node 1
node distances:
node   0   1
  0:  10  50
  1:  255  10
```

메모리 PAGE 할당 및 메모리 단편화 정보를 확인 할 수 있는 /proc/buddyinfo를 확인하여 CXL 영역의 메모리 NODE가 Normal 영역으로 추가가 되었는지 확인합니다.

```
[root@rhel93-cxl ~]# cat /proc/buddyinfo
Node 0, zone      DMA      2      2      0      0      0      0      0      0
1      1      2
Node 0, zone    DMA32     11     10     10      9      9      8      9      8
7      7    559
Node 0, zone   Normal   9598  10996   7298   5656   4628   2377   1094    491
162     35   3145
Node 1, zone   Normal      0      0      0     19     12      7     10      7
7      2  32763
```

CXL 관련 로그를 확인하여 CXL 장치 인식에 특별한 문제가 없는지 확인합니다.

```
[root@rhel93-cxl ~]# cat /var/log/messages | grep -i -e cxl -e montage | grep -v
"rhel93-cxl"
Jan  5 09:20:41 localhost kernel: acpi ACPI0016:00: _OSC: OS supports
[CXL11PortRegAccess CXL20PortDevRegAccess CXLProtocolErrorReporting CXLNativeHot]
Jan  5 09:20:41 localhost kernel: acpi ACPI0016:00: _OSC: OS now controls
[CXLMemErrorReporting]
Jan  5 09:20:53 localhost kernel: cxl root0: Failed to populate active decoder
targets
```

```
Jan  5 09:20:53 localhost kernel: cxl_acpi ACPI0017:00: Failed to add decode range
[0x850000000 - 0x284fffffff]
Jan  5 09:20:53 localhost kernel: pci0000:7f: host supports CXL
Jan  5 09:20:53 localhost kernel: cxl_pci 0000:7f:00.0: No component registers
(-19)
Jan  5 09:20:53 localhost kernel: cxl_pci 0000:7f:00.0: DOE: [d80] failed to cache
protocols : -5
Jan  5 09:20:53 localhost kernel: cxl_pci 0000:7f:00.0: Failed to create MB object
for MB @ d80
Jan  5 09:20:53 localhost kernel: cxl_pci 0000:7f:00.0: Failed to request region
0x0000000000001fff-0x000000000010201e
Jan  5 09:20:54 localhost kernel: cxl_mem mem0: CXL port topology root0 not enabled
```

3.4 RHEL9 OS Kernel 변경 사항

CXL의 Memory Expander 구성 후 OS의 Kernel 변경사항에 대하여 점검합니다.

3.4.1 RHEL9 NUMA Status

numactl 및 numastat 툴은 프로세스 및 운영 체제에 대한 NUMA 노드 메모리 통계를 표시하고, 관리자에게 프로세스 메모리가 시스템 전체에 분산되어 있는지 또는 특정 노드에 중앙 집중화되는지 여부를 보여줍니다. CXL Memory NUMA 확인을 numactl 및 numastat 명령어를 이용하여 kernel의 변경 사항을 확인합니다. 이때 CPU가 없는 ZERO CPU NUMA NODE 가 추가된 것이 확인이 되어야 하며, CXL 장치의 메모리 용량만큼의 정보가 추가된 Memory NODE 로 확인이 되어야 합니다.

```
[root@rhel93-cxl ~]# numactl -H
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95
node 0 size: 31562 MB
node 0 free: 16610 MB
node 1 cpus:
node 1 size: 131072 MB
node 1 free: 131071 MB
node distances:
node   0   1
  0:  10  50
  1:  255  10

## CXL 메모리 daxctl online-memory 수행 결과 ZERO cpu numa node 1가 생성되는 것을 확인

[root@rhel93-cxl ~]# numastat -m
                          Node 0          Node 1           Total
                 --------------- --------------- ---------------
MemTotal               31562.37       131072.00       162634.37
MemFree                16611.27       131071.22       147682.49
MemUsed                14951.10            0.78        14951.88
Active                   236.14            0.00          236.14
```

```
Inactive                10383.11            0.00        10383.11
Active(anon)                1.50            0.00            1.50
Inactive(anon)            132.35            0.00          132.35
Active(file)              234.63            0.00          234.63
Inactive(file)          10250.75            0.00        10250.75
Unevictable                 0.00            0.00            0.00
Mlocked                     0.00            0.00            0.00
Dirty                       0.00            0.00            0.00
Writeback                   0.00            0.00            0.00
FilePages               10529.89            0.00        10529.89
Mapped                     60.03            0.00           60.03
AnonPages                  88.59            0.00           88.59
Shmem                      44.50            0.00           44.50
KernelStack                17.08            0.00           17.08
PageTables                  1.94            0.00            1.94
NFS_Unstable                0.00            0.00            0.00
Bounce                      0.00            0.00            0.00
WritebackTmp                0.00            0.00            0.00
Slab                      609.23            0.78          610.01
SReclaimable              224.30            0.00          224.30
SUnreclaim                384.93            0.78          385.71
AnonHugePages              24.00            0.00           24.00
ShmemHugePages              0.00            0.00            0.00
ShmemPmdMapped              0.00            0.00            0.00
HugePages_Total             0.00            0.00            0.00
HugePages_Free              0.00            0.00            0.00
HugePages_Surp              0.00            0.00            0.00
KReclaimable              224.30            0.00          224.30
```

3.4.2 RHEL9 Memory Hotplug

/proc/meminfo는 시스템의 RAM 사용량에 대한 자세한 정보를 보고하므로 /proc/ 디렉토리에서 메모리 확인을 위한 가장 일반적으로 사용되는 파일 중 하나입니다. meminfo 정보를 확인하여 CXL Memory Expander의 용량이 정상적으로 추가가 되었는지 확인합니다.

```
[root@rhel93-cxl ~]# cat /proc/meminfo
MemTotal:       166537612 kB
MemFree:        151224744 kB
MemAvailable:   161264472 kB
Buffers:            4700 kB
Cached:         10778000 kB
SwapCached:            0 kB
Active:           241980 kB
Inactive:       10631944 kB
Active(anon):       1544 kB
Inactive(anon):   135252 kB
Active(file):     240436 kB
Inactive(file): 10496692 kB
Unevictable:           0 kB
Mlocked:               0 kB
SwapTotal:      16359420 kB
SwapFree:       16359420 kB
```

```
Zswap:                 0 kB
Zswapped:              0 kB
Dirty:                 4 kB
Writeback:             0 kB
...
```

3.4.3 RHEL9 Memory BuddyAllocator

/proc/buddyinfo는 주로 메모리 단편화 문제를 진단하는 데 사용됩니다. 버디 알고리즘을 사용하면 각 열은 특정 시간에 사용할 수 있는 특정 순서, 크기의 페이지 수를 나타냅니다. CXL 영역의 메모리 NODE가 Normal 영역으로 버디 알고리즘에 추가가 되었는지 확인합니다.

```
[root@rhel93-cxl ~]# cat /proc/buddyinfo
Node 0, zone      DMA      2      2      0      0      0      0      0      0
1      1      2
Node 0, zone    DMA32     11     10     10      9      9      8      9      8
7      7    559
Node 0, zone   Normal   8813  10763   7214   5588   4580   2404   1090    497
164     38   3145
Node 1, zone   Normal      0      0      0     19     12      7     10      7
7      2  32763     ← CXL numa node 1
```

3.4.4 RHEL9 Memory Maps

/proc/iomem는 각 물리적 장치에 대한 시스템 메모리의 현재 맵을 보여줍니다. CXL Memory Expander 물리적 정치가 메모리의 현재 물리맵에 추가가 되었는지 확인을 합니다.

```
[root@rhel93-cxl ~]# lsmem
RANGE                                  SIZE  STATE REMOVABLE   BLOCK
0x0000000000000000-0x00000000a7ffffff   2.6G online       yes    0-20
0x0000000100000000-0x000000284fffffff 157.3G online       yes 32-1289

Memory block size:       128M
Total online memory:    159.9G
Total offline memory:      0B

[root@rhel93-cxl ~]# cat /proc/iomem
00000000-00000fff : Reserved
00001000-0009ffff : System RAM
000a0000-000fffff : Reserved
  000a0000-000bffff : PCI Bus 0000:c0
  000c0000-000dffff : PCI Bus 0000:00
  000f0000-000fffff : System ROM
00100000-75bbffff : System RAM
75bc0000-75bfdfff : ACPI Non-volatile Storage
75bfe000-75cbffff : System RAM
```

```
...

100000000-84d9bffff : System RAM
  2ed600000-2ee5fffff : Kernel code
  2ee600000-2ef12bfff : Kernel rodata
  2ef200000-2ef776fff : Kernel data
  2f0002000-2f11fffff : Kernel bss
84d9c0000-84fffffff : Reserved
850000000-284fffffff : CXL Window 0
  850000000-284fffffff : hmem.0
    850000000-284fffffff : Soft Reserved
      850000000-284fffffff : dax0.0
        850000000-284fffffff : System RAM (kmem)
...
```

3.4.5 RHEL9 Memory Zone

/proc/zoneinfo는 proc 파일 시스템에서 zoneinfo 노드로 Zone 별 통계 데이터를 확인할 수 있습니다. 각 영역에는 영역이 얼마나 많은 압력을 받고 있는지 추적하는 데 도움이 되는 "pages_min", "pages_low" 및 "pages_high"라는 세 개의 워터마크가 있으며 CXL Memory Expander 추가 후 해당 워터마크가 CXL Memory Node에도 정상적으로 인식이 되었는지 확인합니다.

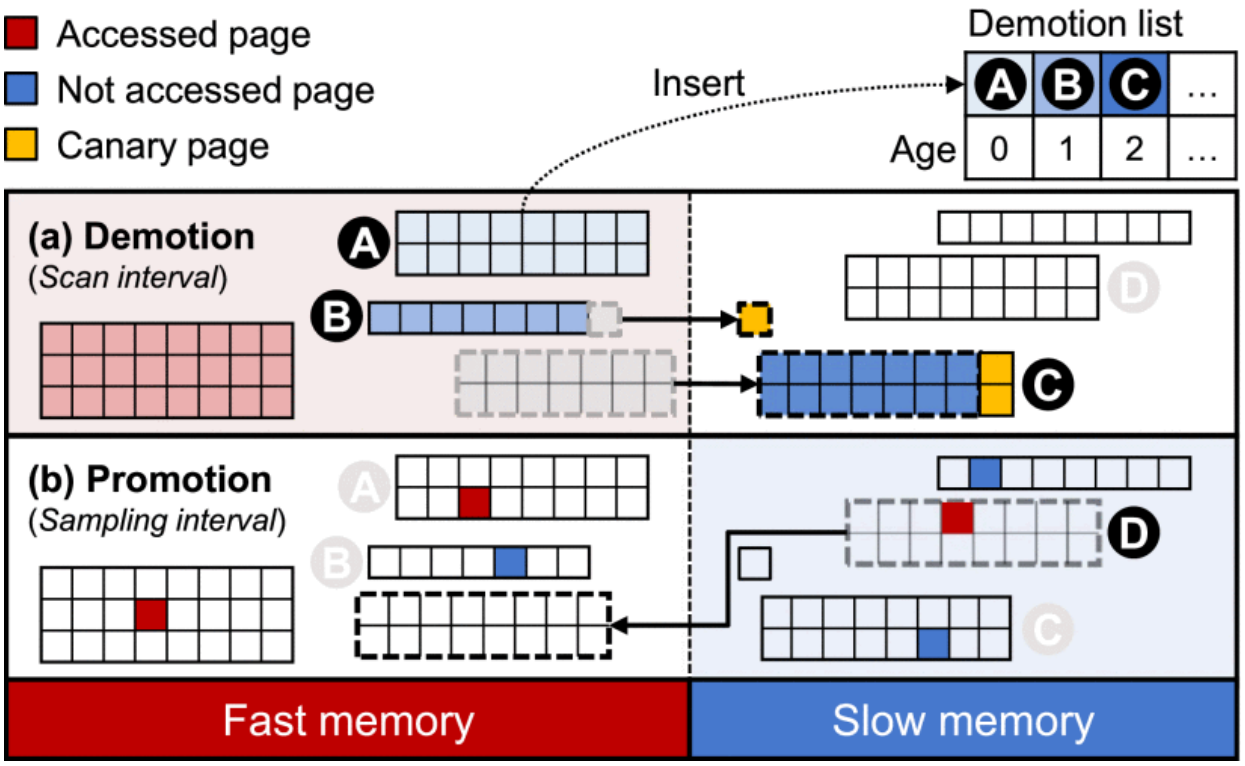
```
[root@rhel93-cxl ~]# cat /proc/zoneinfo | grep -A10 "Node 1"
...
Node 1, zone   Normal
  per-node stats
      nr_inactive_anon 0
      nr_active_anon 0
      nr_inactive_file 0
      nr_active_file 0
      nr_unevictable 0
      nr_slab_reclaimable 0
      nr_slab_unreclaimable 200
      nr_isolated_anon 0
      nr_isolated_file 0
...
 pages free     33554232
        boost    0
        min      18156
        low      51710
        high     85264
        spanned  33554432
        present  33554432
        managed  33554432
        cma      0
        protection: (0, 0, 0, 0, 0)
      nr_free_pages 33554232
      nr_zone_inactive_anon 0
     nr_zone_active_anon 0
      nr_zone_inactive_file 0
```

```
      nr_zone_active_file 0
      nr_zone_unevictable 0
      nr_zone_write_pending 0
      nr_mlock     0
      nr_bounce    0
      nr_zspages   0
      nr_free_cma  0
      numa_hit     0
      numa_miss    0
      numa_foreign 0
      numa_interleave 0
      numa_local   0
      numa_other   0
...
```

4. CXL 연동 활용 가이드

4.1 메모리 Tiering (Page Demotion) 활용

자주 액세스하지 않는 데이터를 로컬 메모리에 저장하게되면 자주 사용하는 메모리 (핫메모리)에 대한 성능상 손해가 발생합니다. 이러한 문제점을 CXL 메모리를 이용하여 시스템 성능을 최적화하고 개선할 수 있습니다. RHEL 에서 제공하는 Memory Demotion 기능을 이용하면 자주 사용하는 핫메모리를 로컬 메모리에 배치하고 자주 사용하지 않는 메모리는 CXL 계층에 배치하여 필요시 메모리간 데이터 마이그레이션을 구현할 수 있습니다.



(https://ieeexplore.ieee.org/document/10016720)

4.1.1 Memory Tiering 구성

dax device system-ram type 구성을 위해 cxl , dax, numa 관련 패키지를 다운로드 합니다. 그리고 dax 장치 리스트를 확인 하고 CXL Memory Expander 장치를 system-ram 형식의 Normal Type으로 인식합니다.

```
[root@rhel93-cxl ~]# dnf install cxl-cli daxctl pciutils numactl

[root@rhel93-cxl ~]# daxctl list

[root@rhel93-cxl ~]# daxctl reconfigure-device --mode=system-ram --no-movable dax0.0
```

NUMA Demotion(강등) 기능 활성화와, numa_balancing Memory Tiering 모드로 설정합니다. 이 옵션이 활성화되면 Local Memory 영역에 과도한 압력이 가해질 시 속도가 느린 디스크 영역으로 스왑 전 CXL Memory 계층으로 Page 마이그레이션이 가능합니다.

```
[root@rhel93-cxl ~]# echo 1 > /sys/kernel/mm/numa/demotion_enabled

[root@rhel93-cxl ~]# echo 2 > /proc/sys/kernel/numa_balancing
```

➔ **numa_balancing** 지원 옵션 리스트
0 : NUMA_BALANCING_DISABLED
1 : NUMA_BALANCING_NORMAL
2 : NUMA_BALANCING_MEMORY_TIERING

4.1.2 Memory Tiering Stress 검증

numa memory tiering 설정 검증을 위해 stress 벤치마킹 툴이 사용되었습니다.
stress 벤치마킹 툴은 EPEL repository에서 제공되므로 EPEL 패키지를 먼저 다운로드 후 stress 패키지 설치가 필요합니다. stress 벤치마킹 툴을 설치 후 swap 사용 조건을 위해 Memory 용량 70% 초과 부하를 수행합니다.

```
[root@rhel93-cxl ~]# dnf install
https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm -y

[root@rhel93-cxl ~]# dnf install stress

[root@rhel93-cxl ~]# stress --vm 2 --vm-bytes 100G --vm-hang 1
```

메모리 과부하시 numa memory tiering 동작이 잘 되었는지 확인합니다. CXL Memory Expander 장치가 추가 후 memory tiering 미 설정시 Total Memory 사용률이 약 84% 시 Swap Memory가 사용이 됩니다. 하지만 CXL Memory Expander 장치가 추가 후 memory tiering이 설정된 환경에서는 아래 검증 확인 내용과 같이 속도가 느린 swap 영역은 사용되지 않고 Cold Tier Memory CXL Memory Expander로 pgpromote_success, pgdemote_kswapd, pgmigrate_success 동작 등이 동작 됨을 검증 할 수 있습니다.

```
======================================
                Node 0 Node 1  Total     ← CXL NUMA NODE 1
                ------ ------ ------
MemTotal        128115 128993 257108
MemFree            806  39917  40723
MemUsed         127309  89076 216385
Active             128      1    129
Inactive        117285  88264 205549
Active(anon)         2      0      2
Inactive(anon)  117195  88080 205275
```

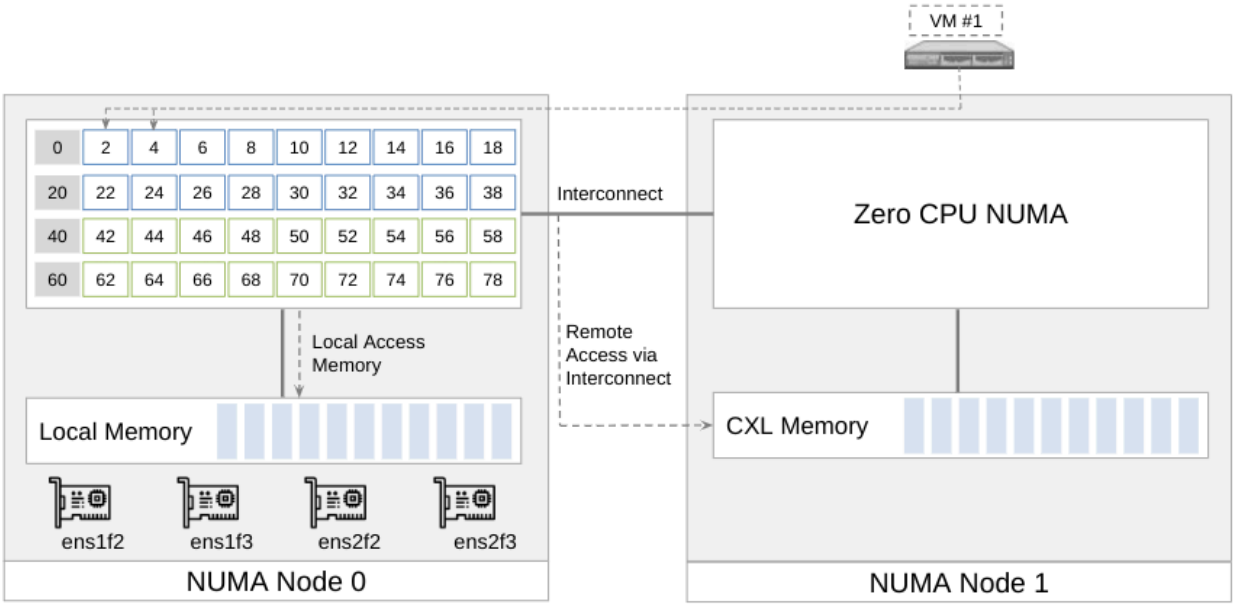
```
Active(file)         126       1     127
Inactive(file)        91     184     274
=====================================
               total        used        free      shared  buff/cache   available
Mem:          257107      217255       40724          49         523       39852
Swap:          16383           0       16383
=====================================
-523M

---Mem Use Rate---
84.50%

---Swap Use Rate---
0.00%
=====================================
pgpromote_success 15812802
pgdemote_kswapd 16029301
pgdemote_direct 0
pgmigrate_success 31882371
pgmigrate_fail 1
```

➔ **pgpromote_success:**
이 값은 성공적으로 페이지 프로모션이 발생한 횟수를 나타냅니다. 페이지 프로모션은 메모리의 페이지를 더 높은 수준의 캐시나 메모리로 이동시키는 프로세스입니다. 페이지 프로모션은 주로 더 높은 수준의 캐시에서 빠르게 액세스할 수 있도록 페이지를 캐시로 가져오는 것을 의미합니다.

➔ **pgdemote_kswapd:**
페이지 디모션은 메모리의 페이지를 더 낮은 수준의 캐시나 메모리로 이동시키는 프로세스를 나타냅니다. 이 값은 kswapd(커널 스왑 데몬)가 페이지 디모션을 수행한 횟수를 나타냅니다. kswapd는 메모리 부족 상황에서 페이지를 디모션하거나 스왑아웃하여 더 많은 여유 메모리를 확보합니다.

4.2 가상머신 활용 (Zero-CPU Numa)

대규모 클라우드 환경에서는 VM (Instance) 에 예약된 로컬 메모리로 인해 서비스 확장에 제약이 있습니다. 이를 극복하고자 CXL 메모리 영역에 VM 을 할당하고 향후 Resource Scale-up 이 유용하게 인프라 구성이 가능합니다. 아래는 KVM 환경에 CXL Numa Memory 를 할당하는 예제입니다.



4.2.1 KVM VM (Numatune 기반) 생성

KVM Guest 를 RHEL9.3 로 Install 을 하여 준비를 합니다. RHEL 9.3 qcow2 image로 생성 후 VM을 Start 합니다. (VM Local Memory 20GB)

```
[root@rhel93-host ~]# virsh start rhel93-vm
```

libvirt XML 수정을 위해 설치 된 KVM Guest RHEL9.3 를 Shutdown 합니다.

```
[root@rhel93-host ~]# virsh shutdown rhel93-vm
```

4.2.2 Libvirt XML 수정

virsh edit 를 이용하여 생성한 Guest RHEL9.3 의 속성을 편집합니다. 아래 예제는 VM 에 할당 된 Memory Nodeset 을 CXL Zero-cpu NUMA 1 로 지정하는 예입니다. 이와 같이 적용 시 VM 은 Local CPU 와 Memory 를 서로 다른 NUMA 영역에서 할당 받을 수 있습니다.

```
[root@rhel93-host ~]# virsh edit rhel93-vm
...
```

```
  <numatune>
    <memory mode='strict' nodeset='1'/> ← CXL NUMA Node 1
    <memnode cellid='0' mode='strict' nodeset='1'/> ← CXL NUMA Node 1
  </numatune>
...
```

4.2.3 KVM with CXL 검증

Guest RHEL9.3 VM 을 Start 하여 메모리 Write 검증을 진행합니다. 아래 예제는 기본적으로 Mount 되어 있는 KVM VM Guest 내 /dev/shm 영역에 임의 데이터를 사용하여 KVM Host 의 CXL 메모리 소비를 검증하는 방법입니다.
(VM 내 Inactive File Backed Cache 9GB 테스트 적재 테스트)

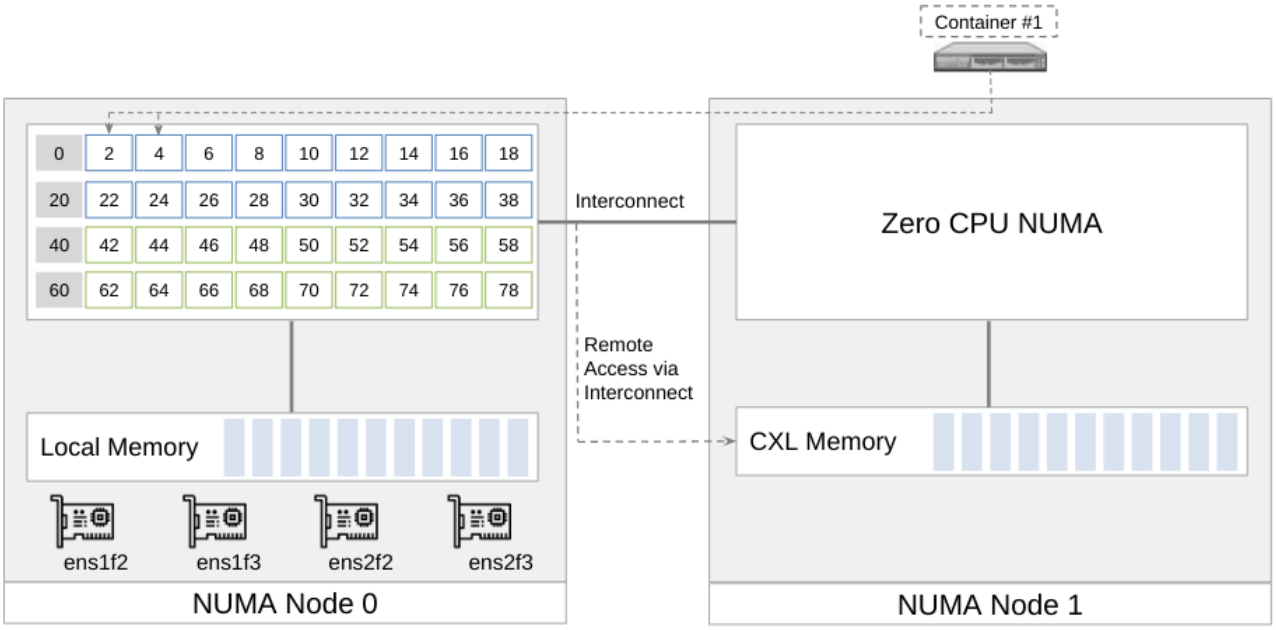
```
[root@rhel93-vm ~]# dd if=/dev/zero of=/dev/shm/test.img bs=1M count=9000
```

위와 같이 KVM VM Guest 내 /dev/shm 영역에 데이터를 사용하게되면, KVM Host 에서 CXL 영역의 사용추이를 확인 할 수 있습니다.
(VM 에 할당 된 CXL memory 20GB 내에서 test.img 파일 9GB 및 Kernel 영역 포함하여 약 11GB 정도의 CXL NUMA NODE 1의 메모리를 사용하는 것을 검증)

```
VM 내 적재 테스트 전 CXL 메모리 상태

[root@rhel93-host ~]# numastat -cm
============================================
                 Node 0 Node 1  Total      ← CXL NUMA NODE 1
                 ------ ------ ------
MemTotal         128167 128953 257120
MemFree          117472 128281 245753
MemUsed          10695  672    11367
Active           81     0      82
Inactive         825    64     889
Active(anon)     2      0      2
Inactive(anon)   753    0      754

VM 내 적재 테스트 후 CXL 메모리 상태

[root@rhel93-host ~]# numastat -cm
============================================
                 Node 0 Node 1  Total      ← CXL NUMA NODE 1
                 ------ ------ ------
MemTotal         128167 128953 257120
MemFree          117449 117826 235275
MemUsed           10718  11127  21845     ← CXL numa node 1 사용량 증가확인
Active               83    733    816
Inactive            841   9739  10579
Active(anon)          2    708    710
Inactive(anon)      750   9399  10149
```

4.3 컨테이너 활용 (Zero-CPU Numa)

기존 VM 내 Application 이 컨테이너 환경으로 옮겨감에 따라 로컬 메모리 한계로 인해 노드가 대규모로 확장되고 있습니다. 이를 극복하고자 CXL 메모리 영역에 Container Application 을 할당하고 향후 Scale-up 이 유용하게 인프라 구성이 가능합니다. 아래는 Podman 환경에 CXL Numa Memory 를 할당하는 예제입니다.



4.3.1 CGroup Memory Node 확인

Podman 에서 할당할 수 있는 CGroup Memory NUMA 노드를 확인합니다. 만약 CXL Memory Node 가 없을 시 Container Process 에 메모리 바인딩에 실패하므로 아래와 같이 설정이 필요합니다.

```
[root@rhel93-cxl machine.slice]# cd /sys/fs/cgroup/cpuset/machine.slice/
[root@rhel93-cxl machine.slice]# echo "0-1" > cpuset.mems
[root@rhel93-cxl machine.slice]# systemctl daemon-reload
```

4.3.2 Podman Container 생성

NUMA #0 의 CPU 와 NUMA #1 CXL Memory 를 사용하는 Podman 컨테이너를 생성합니다.

```
[root@rhel93-cxl machine.slice]# podman run -dt -p 8080:8080/tcp -e
HTTPD_VAR_RUN=/run/httpd \
-e HTTPD_MAIN_CONF_D_PATH=/etc/httpd/conf.d \
-e HTTPD_MAIN_CONF_PATH=/etc/httpd/conf \
-e HTTPD_CONTAINER_SCRIPTS_PATH=/usr/share/container-scripts/httpd/ \
```

```
--name httpd \
--cpuset-cpus=2,4 \
--cpuset-mems=1 \
registry.fedoraproject.org/f29/httpd /usr/bin/run-httpd
6e5c86b74262e6e2b70a34050081056bc2d32e4694688afbda55e4be3b5d06d8
```

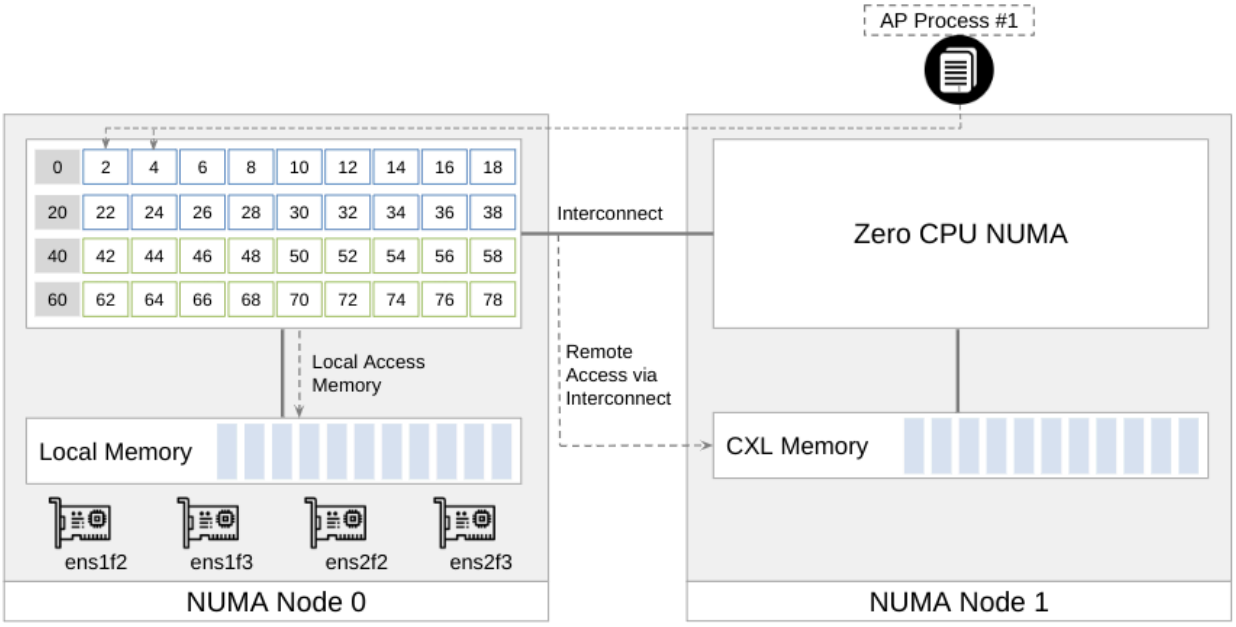
컨테이너 생성 후 해당 CXL Memory 영역에 정상적으로 메모리를 사용하는지 확인합니다.

```
[root@rhel93-cxl machine.slice]# podman ps
CONTAINER ID  IMAGE                                    COMMAND
CREATED         STATUS         PORTS                   NAMES
6e5c86b74262  registry.fedoraproject.org/f29/httpd:latest  /usr/bin/run-http...  31
seconds ago  Up 32 seconds  0.0.0.0:8080->8080/tcp  httpd
```

```
[root@rhel93-cxl /]# numastat -cm (정상적으로 CXL NUMA Node 에 증가 됨)

                Node 0 Node 1  Total
                ------ ------ ------
MemTotal        128500 522240 650740
MemFree          74842 522154 596996
MemUsed          53658     86  53744
Active            3149      0   3149
Inactive         27605     69  27673
Active(anon)        22      0     22
Inactive(anon)   13117     69  13186
Active(file)      3127      0   3127
Inactive(file)   14487      0  14487
Unevictable        441      0    441
Mlocked            438      0    438
Dirty                0      0      0
Writeback            0      0      0
FilePages        17768     60  17828
Mapped            3289      0   3289
AnonPages        11229      8  11237
Shmem              119     60    180
KernelStack        155      3    158
PageTables         127      2    128
NFS_Unstable         0      0      0
Bounce               0      0      0
WritebackTmp         0      0      0
Slab              3766      6   3772
...
```

4.4 응용프로그램 활용 (Zero-CPU Numa Binding)

기본적으로 CXL Memory 를 확장하게 되면 Zero CPU NUMA 형태의 Zone 이 생성이 됩니다. 이는 normal, movable 속성의 메모리로서 커널에 할당이 가능하며, Application 에서 필요시 NUMA Binding 기술을 이용하여 활용할 수 있습니다.



4.4.1 Application Memory Binding

아래 명령어는 NUMA 노드 0에 CPU를 바인딩하고, 1번 노드의 메모리에서 프로그램을 실행하도록 지정합니다. 이렇게 하면 프로그램이 NUMA 지정된 해당 노드의 메모리를 사용하게 됩니다.

```
[root@rhel93-cxl ~]# numactl --cpunodebind=0 --membind=1 ./swapmem_test_1s
```

4.4.2 Application 예제 검증소스

검증을 위한 swapmem_test 의 예제 소스코드는 아래와 같습니다.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

// gcc -o swapmem_test swapmem_test.c

int main(int argc, char** argv) {
    int max = -1;
    int mb = 0;
    char* buffer;

    if(argc > 1)
        max = atoi(argv[1]);
```

```
    while((buffer=malloc(20480*10240)) != NULL && mb != max) {
        memset(buffer, 0, 20480*10240);
        mb++;
        printf("Allocated %d MB\n", mb);
        sleep(1);
    }

return 0;

}
```

4.4.3 Application with CXL 검증

swapmem_test 을 수행하기 전 CXL Node 1 의 상태는 아래와 같이 사용량이 거의 없습니다.

```
=====================================
                Node 0 Node 1  Total  ← CXL NUMA NODE 1
                ------ ------ ------
MemTotal         31562 131072 162634
MemFree          16606 131071 147677
MemUsed          14956    563  14957
Active             120      0    120
Inactive           876      0    876
Active(anon)         3      0      3
Inactive(anon)     503      0    503
Active(file)       117      0    117
Inactive(file)     373      0    373
=====================================

---Mem Use Rate---
4.72%

---Swap Use Rate---
0.00%
=====================================
```

swapmem_test 을 수행 후 numactl membind 옵션을 통해 CXL Memory 영역에 데이터가 증가되는 것을 확인할 수 있습니다. 프로그램 실행 전과 실행 후에 numactl을 사용하여 CXL Numa Memory의 격리를 수행한 결과, 프로그램 실행 후에는 메모리의 Node 1이 사용되고 있음을 확인할 수 있습니다. 이로 인해 프로그램 실행 후에는 해당 프로세스가 Node 1의 메모리를 활용하고 있음을 나타냅니다.

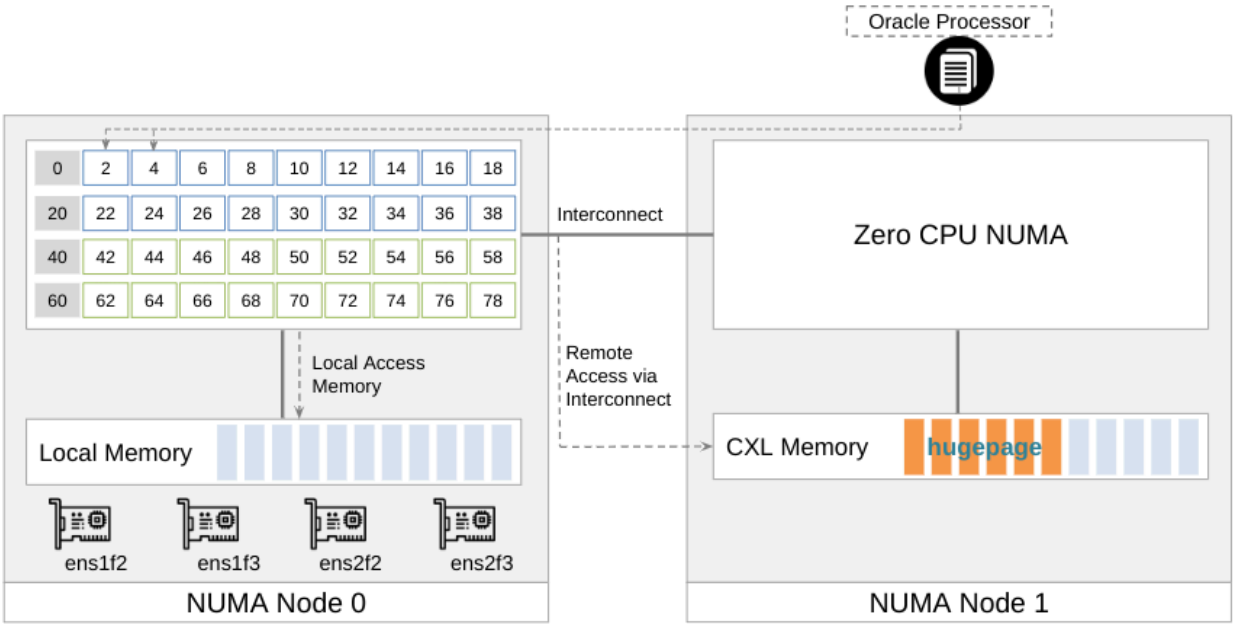
```
=====================================
                Node 0 Node 1  Total  ← CXL NUMA NODE 1
                ------ ------ ------
MemTotal         31562 131072 162634
MemFree          16606   2194  22962
MemUsed          14956 128878 139672  ← CXL numa node 1 사용량 증가
Active             120      0    120
```

```
Inactive             876 128033 128909
Active(anon)           3      0      3
Inactive(anon)       503 128033 128536
Active(file)         117      0    117
Inactive(file)       373      0    373
======================================
-667M

---Mem Use Rate---
54.64%

---Swap Use Rate---
49.71%
======================================
```

4.5 대용량페이지 (Hugepage) 설정

기본적으로 커널은 메모리에서 디스크로 페이지를 교환하거나 메모리 내에서 디스크를 캐싱하여 메모리를 가장 효율적으로 사용하려고 시도합니다. 이는 거의 모든 환경에서 잘 작동하지만, 데이터베이스 애플리케이션을 활용하는 경우 스토리지에서 대기 중인 모든 항목으로 인해 높은 iowait 또는 응답하지 않는 서버와 같은 문제가 발생하는 것을 볼 수 있습니다. 이는 애플리케이션이 데이터에 액세스하려고 시도하는 동안 메모리 페이지를 교체하거나 교체하려고 할 때 디스크 방문 횟수가 높기 때문에 발생합니다.

Hugepage를 사용하면 애플리케이션에서 사용할 수 있도록 미리 할당된 큰 공간을 확보할 수 있습니다. 이 메모리는 디스크에서 스왑 인/아웃되지 않으며 이전 환경에서 나타났던 압력을 일부 완화해 줍니다. 또한 메모리 관리에 대한 오버헤드 감소와 같은 다른 이점도 제공합니다.



4.5.1 OracleDB Hugepage 설정

기본적으로 CXL Memory 를 확장하게 되면 Zero CPU NUMA 형태의 Zone 이 생성되며, Oracle DB 에서 필요시 Hugepage로 설정하여 사용 할 수 있습니다. Oracle ASMM은 SGA 만 자동으로 메모리 관리를 하며 PGA는 자동으로 해주지 않으므로 상황에 맞게 normal 한 값 설정이 필요합니다.

```
[oracle@rhel93-cxl ~]$ dbca
```

oracle 계정으로 전환 후 oracle db instance를 구동합니다.

```
[root@rhel93-cxl ~]# su - oracle
[oracle@rhel93-cxl ~]$ sqlplus / as sysdba
...
```

```
SQL> startup;
ORACLE instance started.

Total System Global Area 8.1068E+10 bytes
Fixed Size                 30393664 bytes
Variable Size            1.0201E+10 bytes
Database Buffers         7.0599E+10 bytes
Redo Buffers              238039040 bytes
Database mounted.
Database opened.
```

4.5.2 Kernel Hugepage 설정

Oracle 에서 제공되는 스크립트를 이용하여 확인 시 아래와 같이 Hugepage 설정에 대한 권장값이 출력됩니다. 이 값을 이용하여 RHEL9.3 환경에서 CXL Memory 영역 (NUMA #1) 에 Hugepage 를 설정할 수 있습니다. Oracle 에서 제공하는 hugepage 스크립트를 수행하여 권장 값을 알 수 있으며, nr_hugepages 에 값을 지정할 수 있습니다.

```
[root@rhel93-cxl ~]# ./hugepages_settings.sh

This script is provided by Doc ID 401749.1 from My Oracle Support
...
* The shared memory segments can be listed by command:
     # ipcs -m
...

Recommended setting: vm.nr_hugepages = 38658

# CXL Memory Expander NODE인 node1에만 Hugepage 적용
[root@rhel93-cxl ~]# echo 38658 >
/sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages

# CXL Memory Expander NODE인 node1 Hugepage 적용 확인
[root@rhel93-cxl ~]# cat
/sys/devices/system/node/node1/hugepages/hugepages-2048kB/nr_hugepages
38658
```

Hugepage 설정 후 사용권한을 지정합니다. (group id 54322) 그리고 /proc/meminfo를 확인하여 HugePages_Total, HugePages_Free 값이 설정한 값으로 설정이 되었는지 확인합니다.

```
[root@rhel93-cxl ~]# id oracle
uid=54321(oracle) gid=54321(oinstall) groups=54321(oinstall),54322(dba),54323(oper)

[root@rhel93-cxl ~]# echo 54322 > /proc/sys/vm/hugetlb_shm_group
[root@rhel93-cxl ~]# cat /proc/sys/vm/hugetlb_shm_group
54322

[root@rhel93-cxl ~]# grep Huge /proc/meminfo
AnonHugePages:    260096 kB
```

```
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:   38658
HugePages_Free:    38658
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:        79171584 kB
```

4.5.3 OracleDB with CXL 검증

Oracle DB 를 기동 후 Hugepage 사용여부를 확인합니다. Hugepage를 Oracle DB에 적용 시 Oracle DB Instance를 재 구동을 해야지만 Hugepage 적용 값이 Oracle DB에 적용이 됩니다.

```
# 현재 설정된 oracle hugepage 설정 값 확인
[root@rhel93-cxl ~]# su - oracle
[oracle@rhel93-cxl ~]$ sqlplus / as sysdba

SQL> show parameter use_large_pages

NAME                                 TYPE        VALUE
------------------------------------ ----------- ------------------------------
use_large_pages                      string      TRUE

# oracle hugepage use_large_pages=only 설정
SQL> alter system set use_large_pages=only scope=spfile;

System altered.

# oracle hugepage 적용을 위한 oracle db instance shutdown 후 start 수행
SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.

SQL> startup;
ORACLE instance started.

Total System Global Area 8.1068E+10 bytes
Fixed Size                 30393664 bytes
Variable Size            1.0201E+10 bytes
Database Buffers         7.0599E+10 bytes
Redo Buffers              238039040 bytes
Database mounted.
Database opened.

# oracle hugepage 적용 후 HugePages_Free 값을 확인 하여 HugePage의 개수가 소진 되었는지 확인
[root@rhel93-cxl ~]# grep Huge /proc/meminfo
AnonHugePages:    360448 kB
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:   38658
```

```
HugePages_Free:      123 ← CXL 영역의 hugepage 소진 검증
HugePages_Rsvd:      123
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:        79171584 kB
```

```
# oracle startup script 설정 및 oracle DB 구동
[oracle@rhel93-cxl ~]$ vim oracle_startup.sh
#!/bin/sh

lsnrctl start
. ~oracle/.bash_profile
sqlplus / as sysdba << __END__
startup
exit
__END__

[oracle@rhel93-cxl ~]$ chmod 755 oracle_startup.sh

# oracle stop script 설정
[oracle@rhel93-cxl ~]$ vim oracle_shutdown.sh
#!/bin/sh

lsnrctl stop
. ~oracle/.bash_profile
sqlplus / as sysdba << __END__
shutdown abort
exit
__END__

[oracle@rhel93-cxl ~]$ chmod 755 oracle_shutdown.sh

# oracle startup script를 --membind 1 옵션으로 CXL Memory Expander NODE 1에 구동
[oracle@rhel93-cxl ~]$ numactl --cpunodebind 0 --membind 1 ./oracle_startup.sh
```

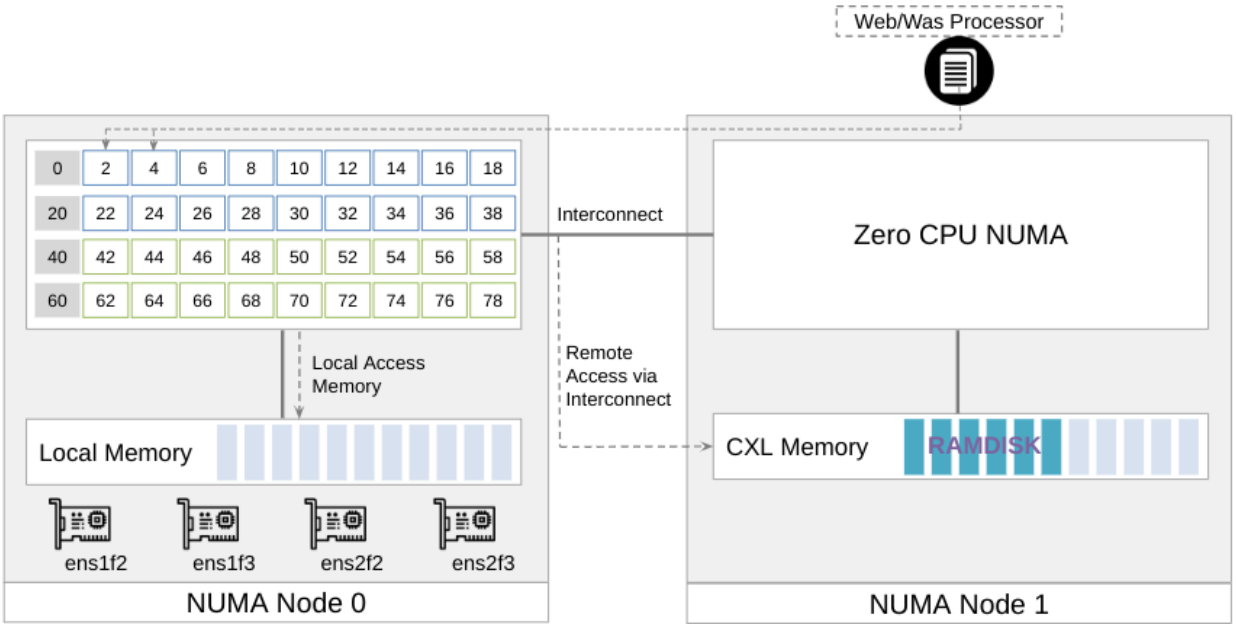
Oracle db Process Hugepage 사용 및 적용 최종확인을 합니다. 확인 시 CXL Memory Expander NODE인 Node 1의 사용률의 변화와 HugePages_Free: 값의 변화를 확인 하여 Oracle DB에 CXL Memory Expander를 이용한 HugePage 설정이 정상인지 확인합니다.

```
# Oracle 구동 전 Memory 사용률
======================================
                 Node 0 Node 1  Total
                 ------ ------ ------
MemTotal          31562 131072 162634
MemFree           16606 131071 147677
MemUsed           14956      1  14957
Active               47      0     47
Inactive            652      0    652
Active(anon)          1      0      1
Inactive(anon)      512      0    512
```

```
Active(file)          46      0     46
Inactive(file)       140      0    140
====================================

[root@rhel93-cxl ~]# cat /proc/meminfo |grep -i huge
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:   38658
HugePages_Free:    38658
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:        79171584 kB

# Oracle 구동 후 Memory 사용률
====================================
                 Node 0 Node 1  Total
                 ------ ------ ------
MemTotal          31562 131072 162634
MemFree           16606  51396  68002
MemUsed           14956  79676  94632
Active               49    130    179
Inactive            683   1444   2127
Active(anon)          2      1      2
Inactive(anon)      515   1239   1755
Active(file)         47    129    176
Inactive(file)      167    205    372
====================================

[root@rhel93-cxl ~]# cat /proc/meminfo |grep -i huge
ShmemHugePages:        0 kB
FileHugePages:         0 kB
HugePages_Total:   38658
HugePages_Free:      124 ← Oracle DB 프로세스가 CXL 영역에 생성된 hugepage 사용 확인
HugePages_Rsvd:      123
HugePages_Surp:        0
Hugepagesize:       2048 kB
Hugetlb:        79171584 kB

# Oracle DB 프로세스가 CXL Memory Expander NODE인 Node 1 에만 적용 되었는지 확인을 위해
numastat 명령어로 확인 할 수 있습니다.

[root@rhel93-cxl ~]# numastat `pidof oracle`

Per-node process memory usage (in MBs)
PID                              Node 0          Node 1           Total
-------------------  --------------- --------------- ---------------
335460 (ora_pmon_cdb1)             4.08         3137.34         3141.42
...
336856 (ora_q003_cdb1)             4.08         4933.88         4937.96
-------------------  --------------- --------------- ---------------
Total                            992.06      1117083.21      1118075.27
```

4.6 램디스크 설정

램디스크 파티션에 저장된 파일은 메모리가 디스크보다 빠르기 때문에 하드 드라이브에 비해 더 빠르게 액세스할 수 있습니다. 일부 데이터에 지속적으로 액세스하면 성능을 높이는 데 도움이 됩니다. (예: 램디스크 파티션을 사용하여 웹서버 데이터를 저장하는 경우 데이터가 메모리에 보관되므로 더 빠른 속도로 데이터에 액세스할 수 있습니다.)



4.6.1 tmpfs type 생성

- ➔ 응용 프로그램이 POSIX와 호환되거나 Red Hat Enterprise Linux 시스템에서 GLIBC(2.2 이상)를 사용하는 경우 일반적으로 공유 메모리(shm_open, shm_unlink)에 /dev/shm을 사용합니다.
- ➔ /dev/shm에서 마운트되는 임시 파일 시스템(tmpfs)입니다 /etc/fstab. 따라서 tmpfs에 지원되는 "size"와 같은 표준 옵션을 사용하여 /dev/shm에서 tmpfs의 크기를 늘리거나 줄일 수 있습니다(기본적으로 사용 가능한 시스템 RAM의 절반입니다)

daxctl 을 이용하여 hotplug 로 추가 된 CXL Memory 영역을 확인합니다. 여기서 numactl, numastat 명령어를 이용하여 CXL Memory Expander NODE가 Node 1로 추가 되었음을 확인 할 수 있습니다.

```
[root@rhel93-cxl ~]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95
node 0 size: 31562 MB
node 0 free: 16609 MB
node 1 cpus:                ← CXL NUMA NODE 1
node 1 size: 131072 MB      ← CXL NUMA NODE 1
node 1 free: 131071 MB      ← CXL NUMA NODE 1
```

```
node distances:
node   0   1
  0:  10  50
  1:  255  10

## CXL 메모리 디바이스 daxctl online-memory 수행 결과 ZERO cpu numa node 1가 생성되는 것을 확인

[root@rhel93-cxl ~]# numastat -cm

                Node 0 Node 1  Total     ← CXL NUMA NODE 1
                ------ ------ ------
MemTotal         31562 131072 162634
MemFree          16609 131071 147680
MemUsed          14954      1  14954
Active             236      0    236
Inactive         10383      0  10383
Active(anon)         2      0      2
Inactive(anon)     132      0    132
Active(file)       235      0    235
Inactive(file)   10251      0  10251
Unevictable          0      0      0
Mlocked              0      0      0
Dirty                0      0      0
Writeback            0      0      0
FilePages        10530      0  10530
Mapped              60      0     60
AnonPages           89      0     89
Shmem               45      0     45
KernelStack         18      0     18
PageTables           2      0      2
NFS_Unstable         0      0      0
Bounce               0      0      0
WritebackTmp         0      0      0
Slab               610      1    611
SReclaimable       224      0    224
SUnreclaim         386      1    387
AnonHugePages       24      0     24
ShmemHugePages       0      0      0
ShmemPmdMapped       0      0      0
HugePages_Total      0      0      0
HugePages_Free       0      0      0
HugePages_Surp       0      0      0
KReclaimable       224      0    224
```

tmpfs 를 생성할 때 mount mpol 옵션을 이용하여 CXL Memory 영역 (NUMA #1) 을 지정합니다. 그리고 CXL Memory Expander의 Memory 용량 60GB를 해당 tmpfs에 할당을 합니다.

```
# tmpfs 생성
[root@rhel93-cxl ~]# mkdir /mnt/tmpfs

[root@rhel93-cxl ~]# mount -t tmpfs -o size=60g,mpol=bind:1 tmpfs /mnt/tmpfs
```

생성 된 tmpfs 의 mount 속성과 사이즈를 확인합니다.

```
# CXL memory (NUMA #1) mount 옵션 검증
[root@rhel93-cxl ~]# df -h /mnt/tmpfs
Filesystem      Size  Used Avail Use% Mounted on
tmpfs            60G     0   60G   0% /mnt/tmpfs

[root@rhel93-cxl ~]# mount | grep /mnt/tmpfs
tmpfs on /mnt/tmpfs type tmpfs
(rw,relatime,seclabel,size=62914560k,inode64,mpol=bind:1)
```

4.6.2 tmpfs with CXL 검증

Kernel 의 Cache 등을 모두 초기화한 후 tmpfs 영역 (/mnt/tmpfs) 에 데이터를 기록하는 검증을 시작합니다.

```
# Memory 테스트 전 현재 메모리 사용량 확인
[root@rhel93-cxl ~]# free -m
              total        used        free      shared  buff/cache   available
Mem:          162634        5149      147680          44       10754      157484
Swap:          15975           0       15975

# Memory 테스트 전 사전 적재된 메모리 초기화
[root@rhel93-cxl ~]# echo 3 > /proc/sys/vm/drop_caches
   -    echo 3는 시스템 메모리 pagecache, dentries, inodes 항목 drop
```

```
# CXL memory NUMA NODE 1 영역으로 파일 생성 수행
[root@rhel93-cxl ~]# numactl --cpunodebind=0 --membind=1 dd if=/dev/zero
of=/mnt/tmpfs/test.img bs=1M count=50000
50000+0 records in
50000+0 records out
52428800000 bytes (52 GB, 49 GiB) copied, 17.08 s, 3.1 GB/s
```

tmpfs 를 사용하기 전후의 상태를 /proc/zoneinfo 이용하여 비교합니다. CXL Memory 는 normal type 으로 생성되어 nr_free page 가 감소되는 것을 확인할 수 있습니다.

```
# dd 수행 전 현재 페이지 사용량 점검
[root@rhel93-cxl ~]# watch -d -n1 "sed -n -e '/Normal/,/numa_other/ p'
/proc/zoneinfo |grep -E 'Normal|pages free|low|min|high'"
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
Node 1, zone   Normal
  pages free     33554232   ← ( 최초 CXL numa node1 memory free size 확인 )
```

```
        min      18156
        low      51710
        high     85264
```

```
# dd 수행 후 페이지 사용량 결과
[root@rhel93-cxl ~]# sed -n -e "/Normal/,/numa_other/ p" /proc/zoneinfo |grep -E
'Normal|pages free|low|min|high'
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
Node 1, zone   Normal
  pages free     20725009 ← ( CXL numa node 1 memory 소진 확인 )
        min      18156
        low      51710
        high     85264
```

tmpfs 메모리 적재 테스트 완료 후 tmpfs 영역에 기록된 데이터를 제거하여 nr_free page 가 증가되는 것을 확인할 수 있습니다.

```
# tmpfs 초기화 수행(1) : 검증 시 생성한 test.img 삭제
[root@rhel93-cxl ~]# rm -f /mnt/tmpfs/test.img

# tmpfs 초기화 수행(2) : 검증 시 생성한 cache Memory 초기화를 위해 drop_cache 수행
[root@rhel93-cxl ~]# echo 3 > /proc/sys/vm/drop_caches

# tmpfs 초기화 수행(3) : 검증 시 생성한 Memory Data로 인해 소진된 pages free 값이 복구
되었는지 확인
[root@rhel93-cxl ~]# sed -n -e "/Normal/,/numa_other/ p" /proc/zoneinfo |grep -E
'Normal|pages free|low|min|high'
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
Node 2, zone   Normal
  pages free     33554049 ← ( CXL numa node 1 memory 복구 확인 )
        min      18156
        low      51710
        high     85264
```

4.6.3 ramfs type 생성

- ➔ 램디스크는 메모리(RAM)의 일부로 할당되어 별도의 파티션으로 사용할 수 있습니다. (데이터를 저장하기 위해 디스크 파티션으로 사용되는 메모리)
- ➔ ramfs는 동적으로 공간을 할당 하므로 ramfs에 데이터를 쓰는 작업은 시스템에서 사용가능한 메모리 크기를 넘어가지 않도록 해야하며, ramfs는 swap을 사용하지 않습니다.

daxctl 을 이용하여 hotplug 로 추가 된 CXL Memory 영역을 확인합니다. 여기서 numactl, numastat 명령어를 이용하여 CXL Memory Expander NODE가 Node 1로 추가 되었음을 확인 할 수 있습니다.

```
# CXL Memory numa 확인
[root@rhel93-cxl ~]# numactl -H
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82
83 84 85 86 87 88 89 90 91 92 93 94 95
node 0 size: 31562 MB
node 0 free: 16609 MB
node 1 cpus:
node 1 size: 131072 MB
node 1 free: 131071 MB
node distances:
node   0   1
  0:  10  50
  1:  255  10


# CXL 디바이스 daxctl online-memory 수행 결과 ZERO cpu numa node 1가 생성되는 것을 확인
```

커널 메시지를 확인 할 수 있는 dmesg를 사용하여 RAMDISK 드라이버 초기화를 확인합니다.

```
# 램디스크 드라이버 module 지원 가능여부 점검
[root@rhel93-cxl ~]#  dmesg | grep RAMDISK
[    0.010821] RAMDISK: [mem 0x52959000-0x56244fff]
```

커널 모듈을 로딩하는 modprobe 명령을 사용하여 램디스크 드라이버 모듈을 로드합니다. 그리고 영구 적용이 필요 시 brd.conf 파일을 /etc/modprobe.d/ 에 생성하여 해당 모듈이 시스템 reboot 후 에도 적용이 되게 설정합니다.

```
# 램디스크 드라이버 모듈 로드
1) 램디스크 60GB 드라이버 생성 - 1회성
[root@rhel93-cxl ~]# modprobe brd rd_size=62914560

2) 램디스크 60GB 드라이버 생성 - 영구 설정
[root@rhel93-cxl ~]# cat /etc/modprobe.d/brd.conf
```

```
options brd rd_numa_node=1 rd_size=62914560

# 램디스크 생성 brd 명령어 옵션 가이드
   -  rd_nr : Maximum number of brd devices
   -  rd_size : Size of each RAM disk in kbytes.
   -  max_part : Maximum number of partitions per RAM disk
```

램디스크 모듈을 커널에 로딩을 수행 후 /dev 하위에 ramx와 같은 장치들의 추가를 확인하고 fdisk 명령어 등을 이용하여 해당 램디스크의 용량이 설정한 용량으로 잘 설정 되었는지 확인합니다.

```
# 램디스크 디바이스 생성 확인
[root@rhel93-cxl ~]# ls -l /dev/ram*
brw-rw----. 1 root disk 1,  0 Oct 26 11:03 /dev/ram0
brw-rw----. 1 root disk 1,  1 Oct 26 11:03 /dev/ram1
brw-rw----. 1 root disk 1, 10 Oct 26 11:03 /dev/ram10
brw-rw----. 1 root disk 1, 11 Oct 26 11:03 /dev/ram11
brw-rw----. 1 root disk 1, 12 Oct 26 11:03 /dev/ram12
brw-rw----. 1 root disk 1, 13 Oct 26 11:03 /dev/ram13
brw-rw----. 1 root disk 1, 14 Oct 26 11:03 /dev/ram14
brw-rw----. 1 root disk 1, 15 Oct 26 11:03 /dev/ram15
brw-rw----. 1 root disk 1,  2 Oct 26 11:03 /dev/ram2
brw-rw----. 1 root disk 1,  3 Oct 26 11:03 /dev/ram3
brw-rw----. 1 root disk 1,  4 Oct 26 11:03 /dev/ram4
brw-rw----. 1 root disk 1,  5 Oct 26 11:03 /dev/ram5
brw-rw----. 1 root disk 1,  6 Oct 26 11:03 /dev/ram6
brw-rw----. 1 root disk 1,  7 Oct 26 11:03 /dev/ram7
brw-rw----. 1 root disk 1,  8 Oct 26 11:03 /dev/ram8
brw-rw----. 1 root disk 1,  9 Oct 26 11:03 /dev/ram9

( default로 램디스크 16개 생성, 필요시 brd 명령어 rd_nr=X 옵션으로 램디스크 개수 지정 )

# 램디스크 디바이스 용량 확인
[root@rhel93-cxl ~]# fdisk -l /dev/ram0 | grep Disk
Disk /dev/ram0: 60 GiB, 64424509440 bytes, 125829120 sectors
```

4.6.4 ramfs with CXL 검증

➔ RAMDISKs Memory 검증 (dd with oflag=direct option)

daxctl 을 이용하여 hotplug 로 추가 된 CXL Memory 영역을 확인합니다. 여기서 numactl, numastat 명령어를 이용하여 CXL Memory Expander NODE가 Node 1로 추가 되었음을 확인 할 수 있습니다.

```
# Memory 테스트 전 현재 메모리 사용량 확인
[root@rhel93-cxl ~]# free -m
               total        used        free      shared  buff/cache   available
Mem:           162634        5152      147677          44       10754      157482
Swap:          15975            0       15975
```

```
[root@rhel93-cxl ~]# numastat -cm

                Node 0 Node 1  Total    ← CXL NUMA NODE 1
                ------ ------ ------
MemTotal         31562 131072 162634
MemFree          16609 131071 147680
MemUsed          14954      1  14954
Active             236      0    236
Inactive         10383      0  10383
Active(anon)         2      0      2
Inactive(anon)     132      0    132
Active(file)       235      0    235
Inactive(file)   10251      0  10251
Unevictable          0      0      0
Mlocked              0      0      0
Dirty                0      0      0
Writeback            0      0      0
FilePages        10530      0  10530
Mapped              60      0     60
AnonPages           89      0     89
Shmem               45      0     45
KernelStack         18      0     18
PageTables           2      0      2
NFS_Unstable         0      0      0
Bounce               0      0      0
WritebackTmp         0      0      0
Slab               610      1    611
SReclaimable       224      0    224
SUnreclaim         386      1    387
AnonHugePages       24      0     24
ShmemHugePages       0      0      0
ShmemPmdMapped       0      0      0
HugePages_Total      0      0      0
HugePages_Free       0      0      0
HugePages_Surp       0      0      0
KReclaimable       224      0    224
```

Kernel 의 Cache 등을 모두 초기화 한 후 ramfs 영역 (/dev/ram0) 에 데이터를 기록하는 검증을 시작합니다. 이때에 tmpfs 검증 시 와는 다르게 oflag=direct를 수행하여 Memory Cache Data를 사용하지 않고 메모리에 데이터를 기록하게 수행합니다.

```
# Memory 테스트 전 사전 적재된 메모리 초기화
[root@rhel93-cxl ~]# echo 3 > /proc/sys/vm/drop_caches
   -    echo 3는 시스템 메모리 pagecache, dentries, inodes 항목 drop

# CXL memory NUMA NODE 1 영역으로 파일 생성 수행
[root@rhel93-cxl ~]# numactl --membind=1 dd if=/dev/zero of=/dev/ram0 bs=1M
count=50000 oflag=direct
50000+0 records in
50000+0 records out
52428800000 bytes (52 GB, 49 GiB) copied, 14.7771 s, 3.5 GB/s
```

ramfs 를 사용하기 전후의 상태를 /proc/zoneinfo 이용하여 비교합니다. CXL Memory 는 normal type 으로 생성되어 nr_free page 가 감소되는 것을 확인할 수 있습니다.

```
# dd 수행 전 현재 페이지 사용량 점검
[root@rhel93-cxl ~]# watch -d -n1 "sed -n -e '/Normal/,/numa_other/ p'
/proc/zoneinfo |grep -E 'Normal|pages free|low|min|high'"
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
Node 1, zone   Normal
  pages free     33554232   ← ( 최초 CXL numa node 1 memory free size 확인 )
        min      18156
        low      51710
        high     85264
```

```
# dd 수행 후 페이지 사용량 결과
[root@rhel93-cxl ~]# sed -n -e "/Normal/,/numa_other/ p" /proc/zoneinfo |grep -E
'Normal|pages free|low|min|high'
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
Node 1, zone   Normal
  pages free     20725009 ← ( CXL numa node 1 memory 소진 확인 )
        min      18156
        low      51710
        high     85264
```

ramfs 메모리 적재 테스트 완료 후 ramfs 영역에 기록된 데이터를 제거하여 nr_free page 가 증가되는 것을 확인할 수 있습니다.

```
# ramfs 초기화 수행(1) : 검증 시 생성한 brd 모듈 삭제
[root@rhel93-cxl ~]# modprobe -r brd

# ramfs 초기화 수행(2) : 검증 시 생성한 cache Memory 초기화를 위해 drop_cache 수행
[root@rhel93-cxl ~]# echo 3 > /proc/sys/vm/drop_caches

# ramfs 초기화 수행(3) : 검증 시 생성한 Memory 로 인해 소진된 pages free 복구 여부 확인
[root@rhel93-cxl ~]# sed -n -e "/Normal/,/numa_other/ p" /proc/zoneinfo |grep -E
'Normal|pages free|low|min|high'
[root@rhel93-cxl ~]# sed -n -e "/Normal/,/numa_other/ p" /proc/zoneinfo |grep -E
'Normal|pages free|low|min|high'
Node 0, zone   Normal
  pages free     3669091
        min      4054
        low      11547
        high     19040
```

```
Node 2, zone   Normal
  pages free     33554049 ← ( CXL numa node 1 memory 복구 확인 )
        min      18156
        low      51710
        high     85264
```

```
[root@rhel93-cxl ~]# free  -m ( dd test 50G Memory 반환 완료 )
               total        used        free      shared  buff/cache   available
Mem:          162634        5152      147677          44       10754      157482
Swap:          15975           0       15975

[root@rhel93-cxl ~]# numastat -cm ( MemFree영역이 oflag=direct 옵션에 따른 정상을 확인 )
                 Node 0 Node 1  Total    ← CXL NUMA NODE 1
                 ------ ------ ------
MemTotal          31562 131072 162634
MemFree           16607 131071 147678
MemUsed           14956      1  14956
Active              236      0    236
Inactive          10384      0  10384
Active(anon)          2      0      2
Inactive(anon)      133      0    133
Active(file)        235      0    235
Inactive(file)    10251      0  10251
Unevictable           0      0      0
Mlocked               0      0      0
Dirty                 0      0      0
Writeback             0      0      0
FilePages         10530      0  10530
Mapped               60      0     60
AnonPages            89      0     89
Shmem                45      0     45
KernelStack          17      0     17
PageTables            2      0      2
NFS_Unstable          0      0      0
Bounce                0      0      0
WritebackTmp          0      0      0
Slab                611      1    612
SReclaimable        225      0    225
SUnreclaim          386      1    387
AnonHugePages        24      0     24
ShmemHugePages        0      0      0
ShmemPmdMapped        0      0      0
HugePages_Total       0      0      0
HugePages_Free        0      0      0
HugePages_Surp        0      0      0
KReclaimable        225      0    225
```

5. Revision History

Revision #	Date	Summary of changes	Author
1.0	Jan, 2024	챕터 3 AMD, INTEL CPU 환경 CXL 구성 - 테스트 환경 하드웨어 변경 - Intel (CPU : 2Socket → 1Socket, Local Mem : 128GB → 32GB, CXL Expander : 4EA → 1EA) - AMD (Local Mem : 128GB → 32GB, CXL Expander : 4EA → 1EA)	Red Hat
1.0	Jan, 2024	챕터 4 CXL 연동 활용 가이드 - 테스트 아키텍처 Intel/AMD 동일 구성 표준화 업데이트	Red Hat
1.0	Jan, 2024	챕터 2 시스템 업데이트 - 오타 수정	Red Hat