



# resctrl: multiple controls

Reinette Chatre

None of this is merged. All of it is up for critique.



LINUX  
PLUMBERS  
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

# Current status of multiple controls

- RFC v2: <https://lore.kernel.org/lkml/d258a32f-12d5-464d-abe9-4720fb3e44b3@intel.com/>
- Interface discussions settled
- Upstreaming underway: cleanup series and MPAM driver RFC

## 4 planned user interface changes:

1. Scope: resource vs. control
2. Per-domain "max" for bitmap controls
3. New resources: users should not expect control with the same name; "native"-only unless emulated
4. Control mode change may impact monitoring files



LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



# Separate resource scope from control scope

Today

- **MB, SMBA:** memory bandwidth at egress of L3, controlled at L3 scope

Needs to support

- **AMD GLBE:** memory bandwidth at egress of L3, controlled at either NUMA node or "system" scope
- **MPAM:** memory bandwidth at ingress of NUMA node, controlled at NUMA node scope

The interface

- Control name pattern: ``<resource>_<res_scope><ctrl_scope>_<control>``; drop `<ctrl_scope>` when the two scopes match
- Per-control file:  
`info/<resource>\_<res\_scope>/schemata/<resource>\_<res\_scope><ctrl\_scope>\_<control>/scope`
  - Tells userspace what a domain ID in the schemata means (L3 cache ID, NUMA node ID, ...)



LINUX

PLUMBERS CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026



# New interface: emulated controls

```
info/  
└─ MB/  
   └─ control_mode  
      └─ schemata/  
         └─ MB/  
            └─ max:100  
               └─ MB_L3_MAX/  
                  └─ max:200  
                     └─ min:1  
                        └─ resolution:200  
                           └─ scale:1  
                              └─ scope:L3  
                                 └─ tolerance:1  
                                    └─ type:scalar linear  
                                       └─ unit:all  
                                          └─ min:10  
                                             └─ resolution:100  
                                                └─ scale:1  
                                                   └─ scope:L3  
                                                      └─ tolerance:10  
                                                         └─ type:scalar linear  
                                                            └─ unit:all  
                                                               └─ MB_L3_MIN/  
                                                                  └─ max:200  
                                                                     └─ min:1
```

```
# cat info/MB/control_mode  
[legacy] native  
# cat schemata  
MB:0=100;1=100  
MB_L3_MIN:0=200;1=200  
# cat info/MB/min_bandwidth  
10  
# echo native > info/MB/control_mode  
# cat info/MB/control_mode  
legacy [native]  
# cat info/MB/min_bandwidth  
#  
# cat schemata  
MB_L3_MAX:0=200;1=200  
MB_L3_MIN:0=200;1=200
```

# New interface: planned changes

per-domain bitmap  
maximum

```
info/  
└─ L3/  
   └─ control_mode:[legacy]  
      schemata/  
      └─ L3/  
         └─ max:0=0xffff;1=0xff  
            └─ min_bits:1  
               └─ scope:L3  
                  type:bitmap
```

new resource without control of same  
name, control\_mode is native

```
info/  
└─ MB_NODE/  
   └─ control_mode:[native]  
      schemata/  
      └─ MB_NODE_MAX/  
         └─ scope:node
```

L3 bandwidth allocated at NUMA  
node scope

```
info/  
└─ MB/  
   └─ control_mode:[legacy]  
      schemata/  
      └─ MB/  
         └─ MB_L3NODE/  
            └─ scope:node
```



# Wrap-up

## Current limitations

- Cache features (CDP, IO alloc, pseudo-locking): one control only
- Memory bandwidth software controller (mba\_MBps): legacy MB control only

## Upstreaming

- Feedback appreciated
- Need support with upstreaming of API changes



# CPU Bandwidth regulation in MPAM and resctrl

- FEAT\_MPAM\_PE\_BW\_CTRL introduces per-CPU bandwidth regulation for MPAM.
- It's documented in the Arm ARM. D24.12.5 MPAMBW0\_EL1 has most of the details.
- It adds new per PE registers, MPAMBWx\_Ely. These specify a maximum bandwidth for requests exiting the CPU. These can be written at context switch at the same time as PARTID/PMG in MPAMx\_Ely.
- If hardware scaling is enabled then the maximum bandwidth is divided by the number of PE currently making requests with the same PARTID at the same exception level.

## How should resctrl expose CPU Bandwidth regulation?

- For now, assume there is no h/w scaling.
- What should the scope be? CPU? It can be called MB\_CPU\_MAX. Unwieldy when there are lots of CPUs but simplest.
- Or should we increase the scope?
- Do we want software scaling so that the bandwidth maximum stays the same even if the tasks are running on multiple CPUs?
- How do we distinguish it from other \_MAX controls?



# Updates on CPU-less Node and Memory Bandwidth Usage Max Hard Limit on ARM MPAM

Fenghua Yu, NVIDIA | Linux Plumbers Conference, Czechia Oct 05, 2026

# Introduction

- Proposed enabling CPU-less memory node in resctrl on LPC 2025 [1][2]
- New resource control interfaces are being discussed on LKML[3]
- Working on CPU-less memory node based on new resource control interfaces [4]
- Enabling max memory bandwidth usage hard limit on MPAM [5]

# CPU-less Memory Node

- A traditional NUMA node has CPUs, cache, and memory
  - A local CPU accesses MPAM registers to control or monitor L2/L3 cache and memory bandwidth on the node
- A CPU-less memory node only has memory
  - No CPU
  - No L2/L3 cache
- Why CPU-less memory node?
  - Resolve ratios and over-provisioning of CPU and memory
  - Resolve imbalanced scaling of CPU and memory
  - Disaggregate memory from host processors
    - Better utilization of CPU and memory
    - Reduced TCO
  - Example: CXL Type 3 memory expansion and pooling

# New Schemata Line for CPU-less MBA

## Proposed resctrl Changes for MBA

- Current MBA line in resctrl “schemata” file:
  - MB: <L3 cache id 0>=<bw>; <L3 cache id 1>=<bw>
- No L3 cache on CPU-less node
  - The current MBA line cannot be used
- Need to create a new line for MBA on CPU-less node
  - MB\_NODE: <numa node id 0>=<bw>; <numa node id 1>=<bw>
  - NUMA node id is stored in ris->nid for each resource node
- schemata example on two CPU nodes and two GPU (which has no CPU) nodes
  - MB\_NODE: 0=20;1=20;34=80;35=80      ← allocate 80% bw on CPU nodes (0 and 1) and 80% bw on GPU nodes (34 and 35)

# New Monitoring Event Paths for CPU-less MBM

## Proposed resctrl Changes for MBM [4]

- Current MBM events in resctrl are per L3
  - `<partition>/mon_data/mon_L3_<cache id>/mbm_total_bytes`: total memory rw bytes
- No L3 cache on CPU-less node
  - The current total bytes cannot be used
- Need to monitor the total bytes per NUMA node on CPU-less node:
  - `<partition>/mon_data/mon_MB_<numa node id>/mbm_total_bytes`: total memory r/w bytes on this CPU-less node

# Max Memory Bandwidth Hard Limit

- Soft limit of allocated max memory bandwidth
  - `MPAMCFG_MBW_MAX.hardlimit[31]=0`
  - Limits bandwidth to max during contention; permits extra bandwidth at lower priority when unallocated capacity exists
  - Maximizes overall memory bandwidth utilization
  - But less predictable isolation under light memory bandwidth load
- Hard limit of allocated max memory bandwidth
  - `MPAMCFG_MBW_MAX.hardlimit[31]=1`
  - Strictly caps bandwidth usage at max, blocking excess requests regardless of system availability
  - Guarantees strict bandwidth isolation between workloads
  - Potential waste of idle bandwidth capacity

# Memory Bandwidth Monitoring Counter Assignment on CPU-less nodes

- `mbm_L3_assignments` is L3 cache id based (legacy)
- `mbm_NODE_assignments` is extended to numa node scope
- Example:
  - `mbm_NODE_assignments: 0=e;1=e;34=e;35=e`

# Discussion on the Proposed resctrl API Changes

- Valid schemata change of new “MB\_NODE” line for CPU-less MBA?
- Valid total bytes per NUMA node id changes?
- Valid MB\_HLIM per domain/numa node?

# References

1. Resctrl/RDT/PQoS/MPAM/RISC-V micro conference on Linux Plumbers Conference 2025, Tokyo: <https://lpc.events/event/19/contributions/2093/>
2. NVIDIA 7.0 Base OS kernel: [https://github.com/NVIDIA/NV-Kernels/tree/26.04\\_linux-nvidia-bos](https://github.com/NVIDIA/NV-Kernels/tree/26.04_linux-nvidia-bos)
3. Reinette's resctrl control branch RFC v2: [git://git.kernel.org/pub/scm/linux/kernel/git/reinette/linux.git](https://git.kernel.org/pub/scm/linux/kernel/git/reinette/linux.git) branch resctrl/controls\_rfc\_v2.4
4. Enable CPU-less memory node on resctrl/mpam RFC v2: <https://lore.kernel.org/lkml/20260831172245.42253-1-fenghuay@nvidia.com/>
5. Enable max memory bandwidth usage hard limit: [https://github.com/NVIDIA/NV-Kernels/tree/26.04\\_linux-nvidia-bos](https://github.com/NVIDIA/NV-Kernels/tree/26.04_linux-nvidia-bos)
6. Resctrl kernel document: Documentation/filesystems/resctrl.rst
7. ARM MPAM specification: <https://support.arm.com/documentation/ih0099/latest/>
8. ARM MPAM ACPI specification: <https://support.arm.com/documentation/den0065/3-0/>

