arm

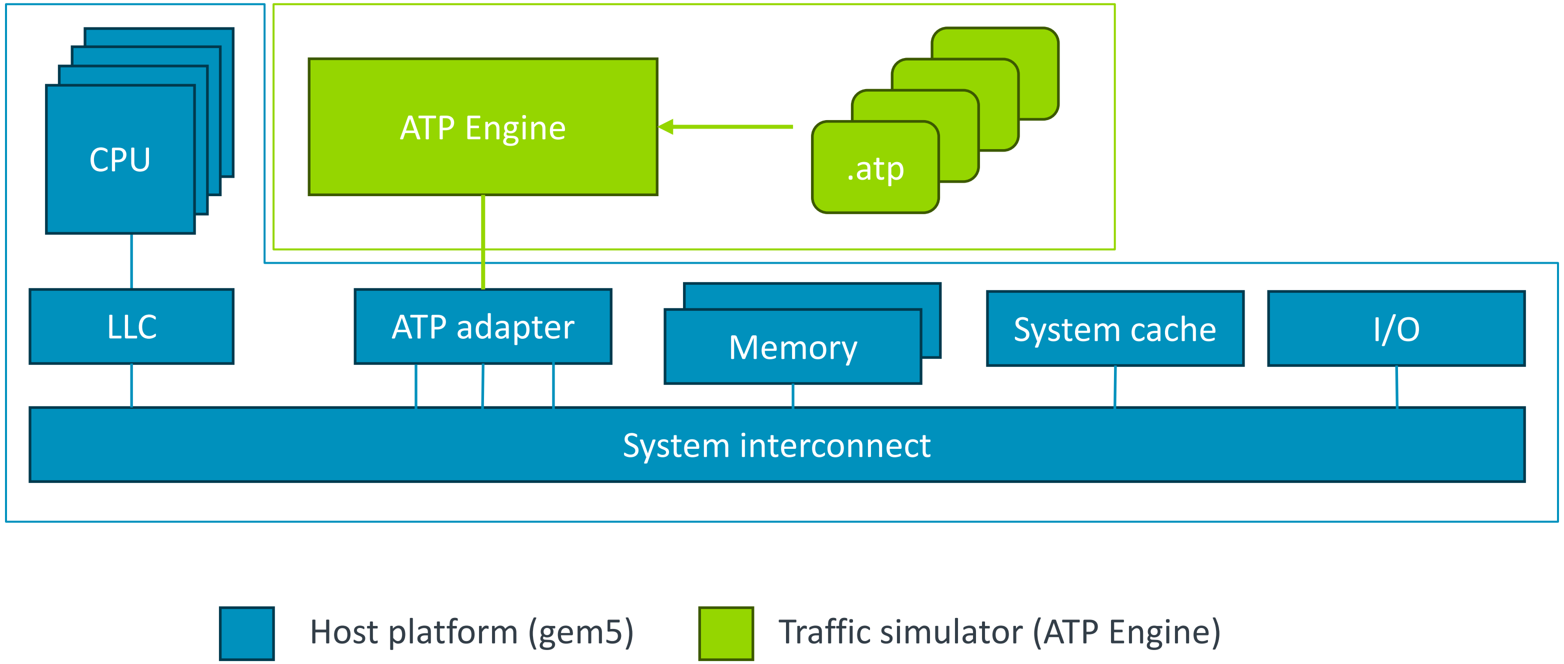
Introduction to MPAM cache partitioning in the virtual platform

Adrian Herrera (adrian.herrera@arm.com)

Senior Engineer, Architecture & Technology, Arm

Summary of the virtual platform

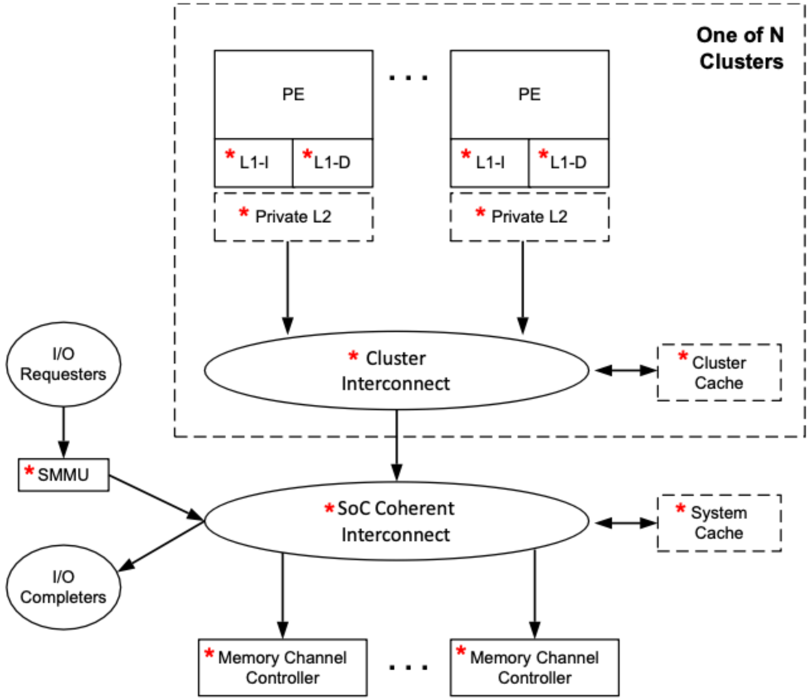
- Introduced originally in ECRTS 2022 as part of the industrial challenge
- Based on the gem5 simulator and the ATP Engine traffic simulator
- Goals: increase accessibility of the challenge, leverage the active gem5 community, facilitate sharing of features and experiments



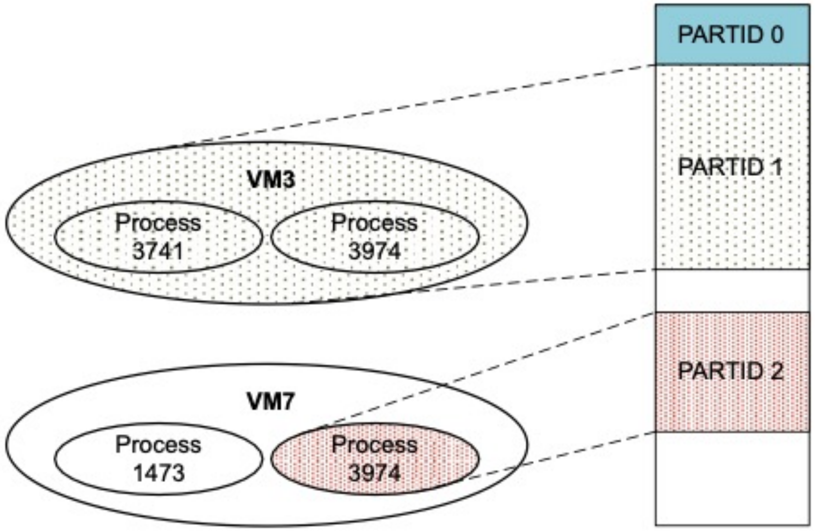
Available in: https://github.com/ecrtsorg/gem5/tree/stable/configs/example/arm/atp

Introduction to MPAM

- Arm’s Memory Partitioning and Monitoring (MPAM) architecture extension
- Original target problem: uncontrolled performance effects between software components, especially in networking and servers
- Enables software to label traffic, apportion resources, and monitor their usage

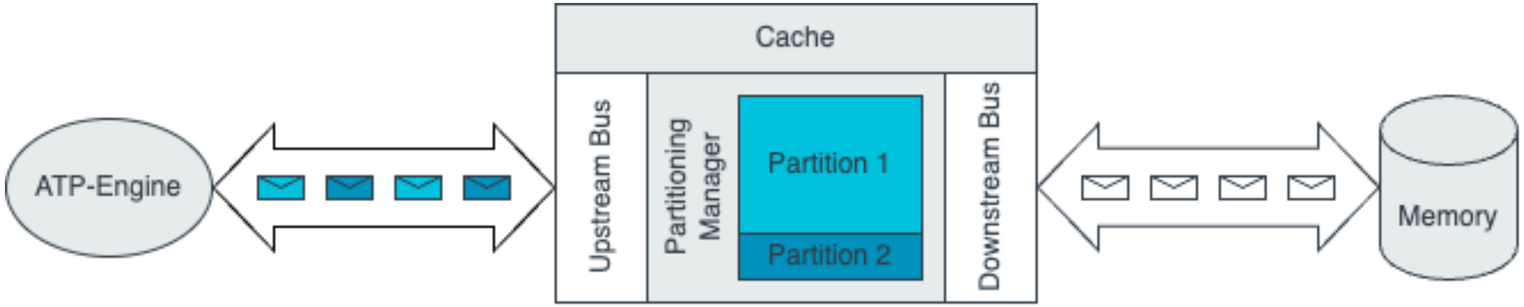


* Memory-System Component (MSC) that might contain MPAM resource controls



MPAM cache partitioning in gem5

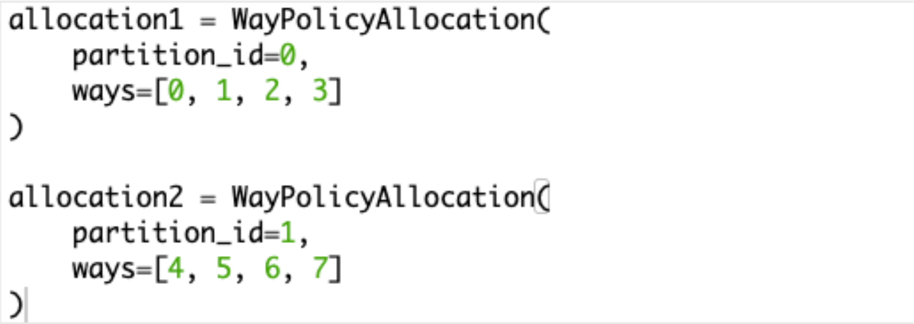
- Available from gem5 v24.0 and ATP Engine v3.2, released June 2024
- Packet is extended to store MPAM information (traffic label)
- ATP Engine and gem5 CPU models are extended to label outgoing traffic
- Cache-portion partitioning
  - Allocate portions, which are fixed-sized, uniquely identifiable and shareable
- Cache maximum-capacity partitioning with hard limit
  - Establish a maximum limit of usable capacity for a partition



Details: https://community.arm.com/arm-community-blogs/b/architectures-and-processors-blog/posts/gem5-cache-partitioning

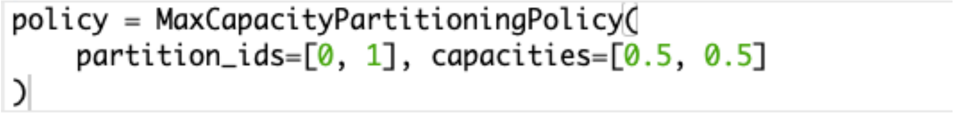
Using MPAM cache partitioning in gem5

- Configuring partitioning policies

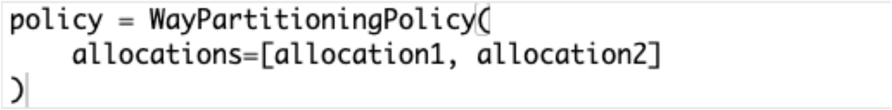


```
allocation1 = WayPolicyAllocation(
    partition_id=0,
    ways=[0, 1, 2, 3]
)

allocation2 = WayPolicyAllocation(
    partition_id=1,
    ways=[4, 5, 6, 7]
)
```

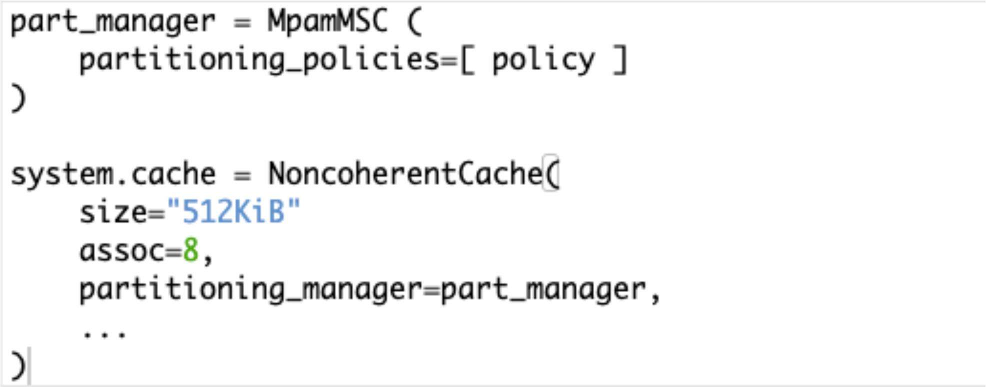


```
policy = MaxCapacityPartitioningPolicy(
    partition_ids=[0, 1], capacities=[0.5, 0.5]
)
```



```
policy = WayPartitioningPolicy(
    allocations=[allocation1, allocation2]
)
```

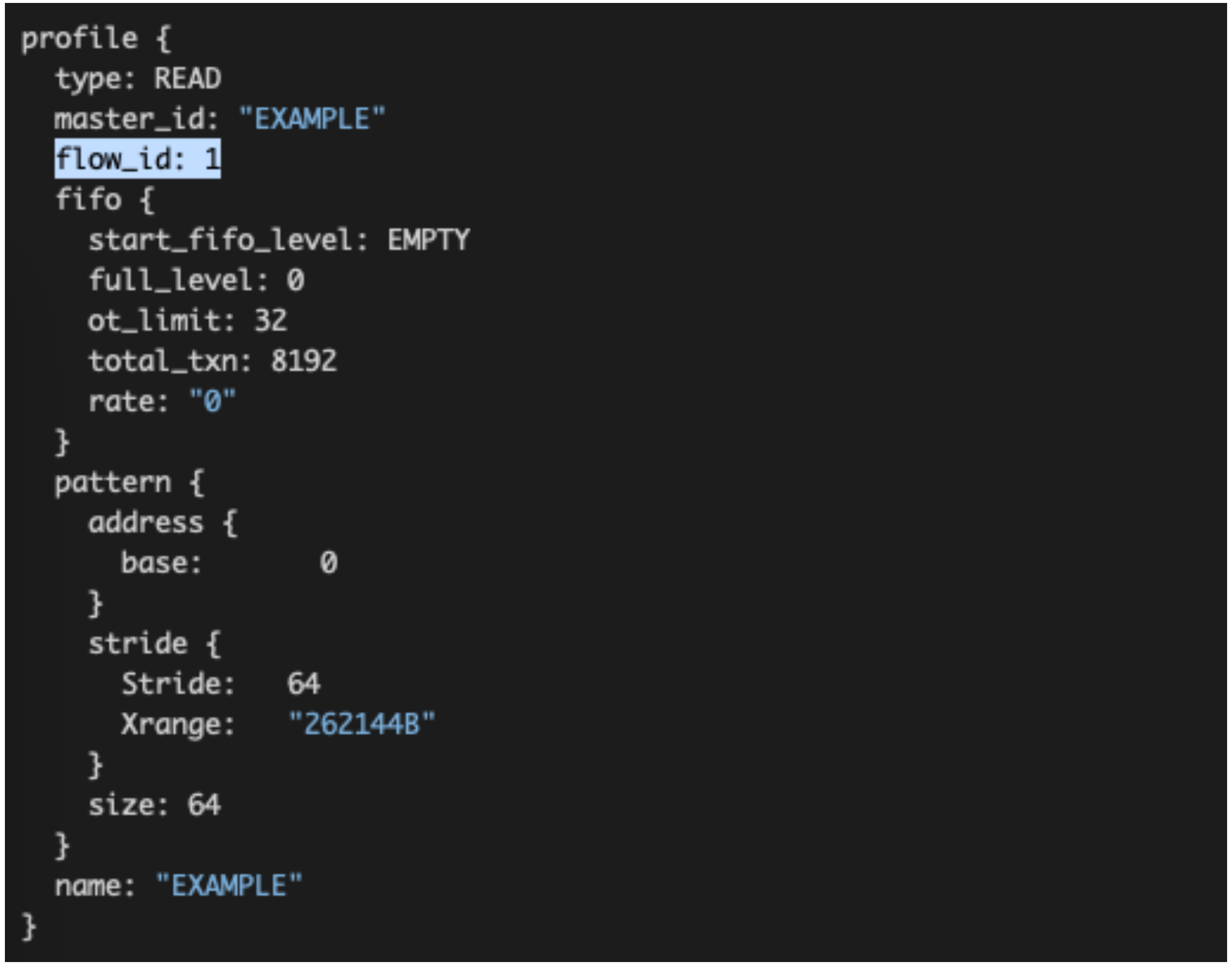
- Configuring the partition manager and the managed cache



```
part_manager = MpamMSC (
    partitioning_policies=[ policy ]
)

system.cache = NoncoherentCache(
    size="512KiB"
    assoc=8,
    partitioning_manager=part_manager,
    ...
)
```

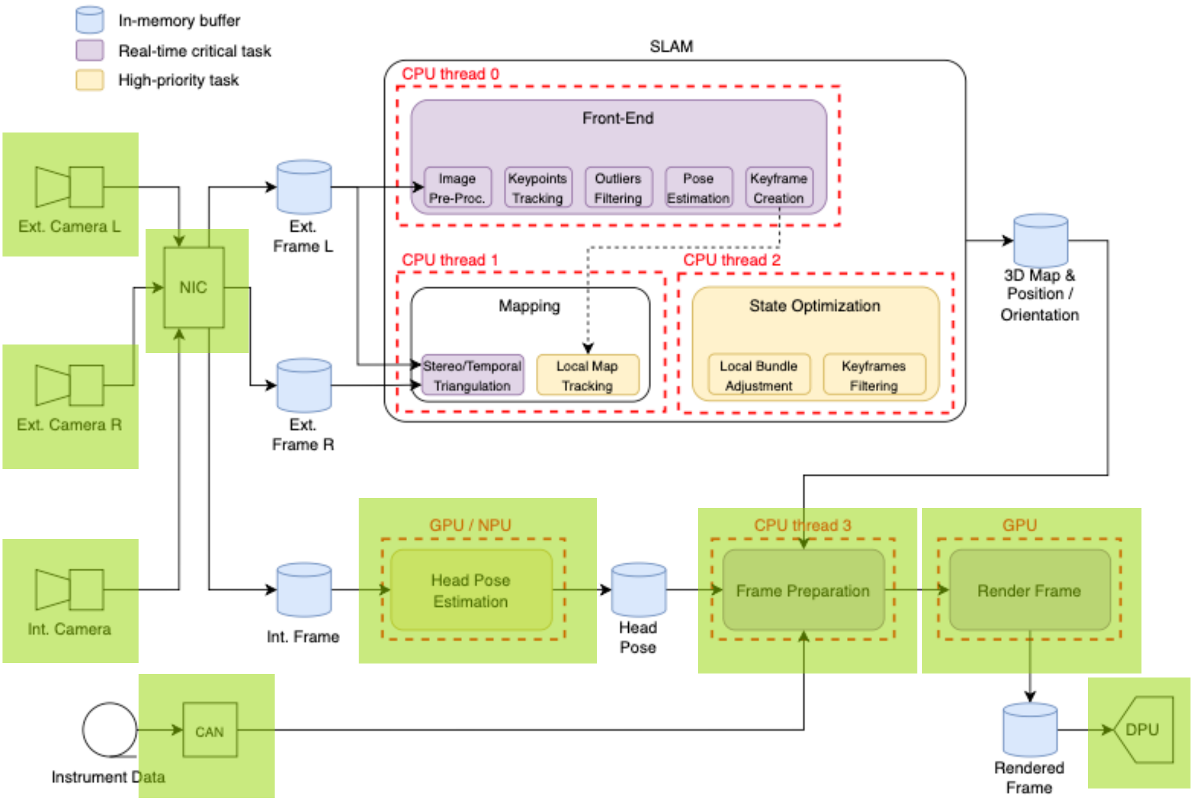
Example Adaptive Traffic Profile (ATP) with flow ID



```
profile {
  type: READ
  master_id: "EXAMPLE"
  flow_id: 1
  fifo {
    start_fifo_level: EMPTY
    full_level: 0
    ot_limit: 32
    total_txn: 8192
    rate: "0"
  }
  pattern {
    address {
      base:       0
    }
    stride {
      Stride:   64
      Xrange:   "262144B"
    }
    size: 64
  }
  name: "EXAMPLE"
}
```

Next steps and final remarks

- Arm will shortly release a set of traffic profiles (ATPs) for the elements highlighted in green in the image
  - Goals are to provide participants with reference ATPs, and enable them to focus on mechanisms to manage the traffic at the processor and system level
- Please, do reach out
  - Concerns and suggestions regarding the challenge, virtual platform, etc.
  - Over at the official mailing list https://groups.google.com/g/ecrts-industrial-challenge
  - Directly to me at adrian.herrera@arm.com, or any of the challenge authors in the original paper



Thank You

Danke

Gracias

Grazie

谢谢

ありがとう

Asante

Merci

감사합니다

धन्यवाद

Kiitos

شكرًا

ধন্যবাদ

תודה

ధన్యవాదములు