ISCA' 21

Don't Forget the I/O When Allocating Your LLC

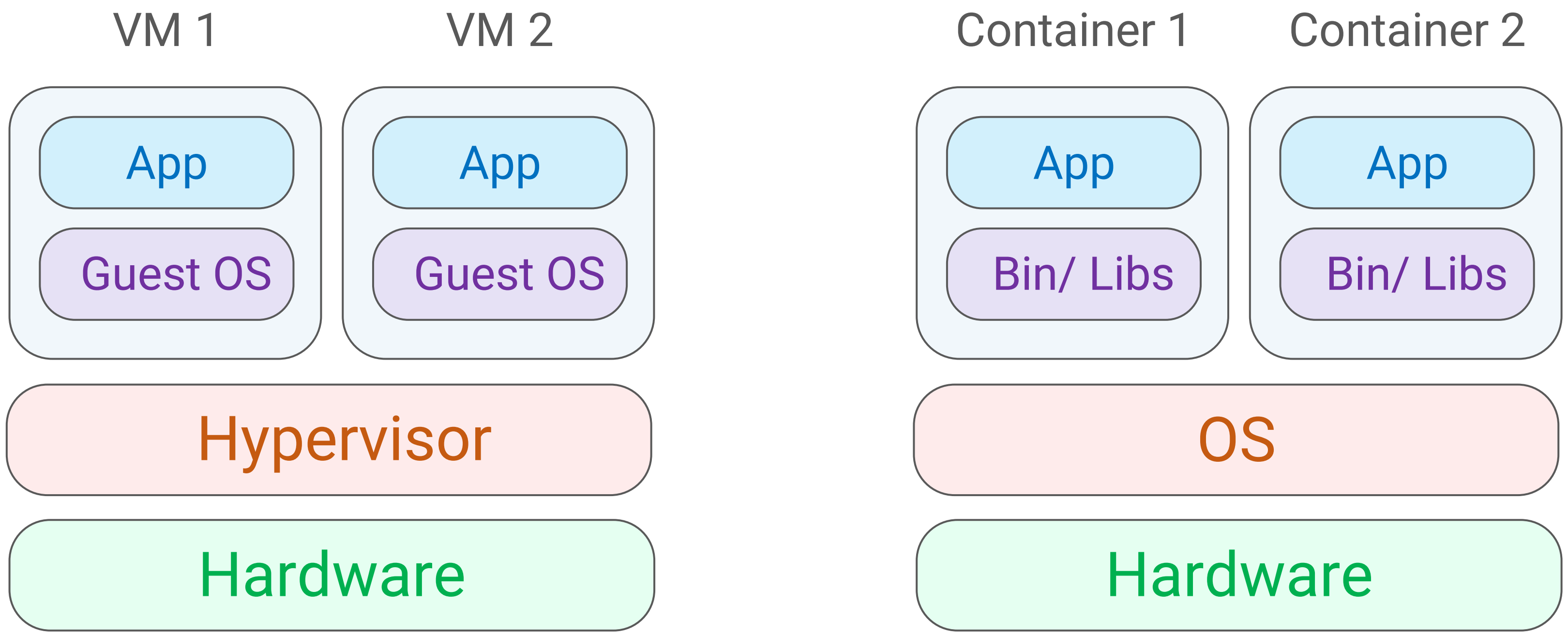
Yifan Yuan, Mohammad Alian, Yipeng Wang, Ren Wang, Ilia Kurakin, Charlie Tai, Nam Sung Kim

Shiv Bhushan Tripathi

CDOS Talk
04th July 2022

Resource Sharing in Cloud

Resource Sharing in Cloud



VM 1

VM 2

App

Guest OS

App

Guest OS

Hypervisor

Hardware

Container 1

Container 2

App

Bin/ Libs

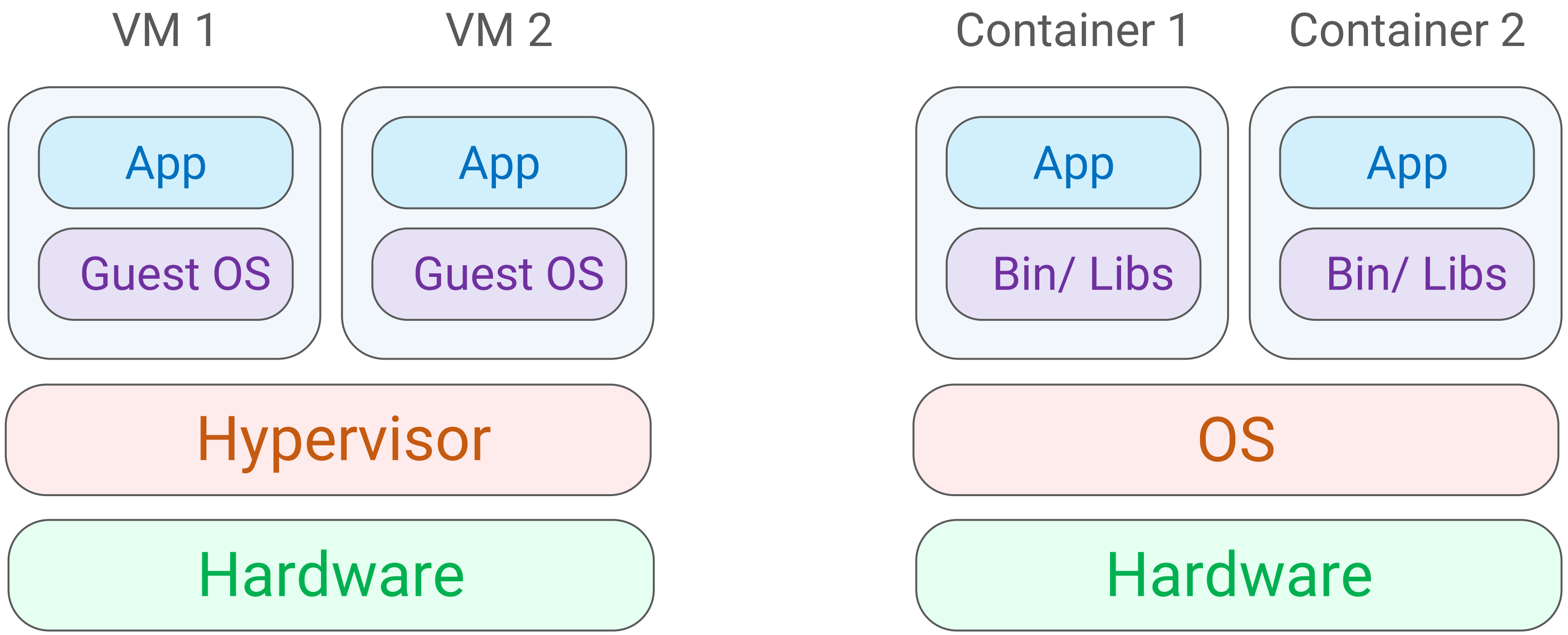
App

Bin/ Libs

OS

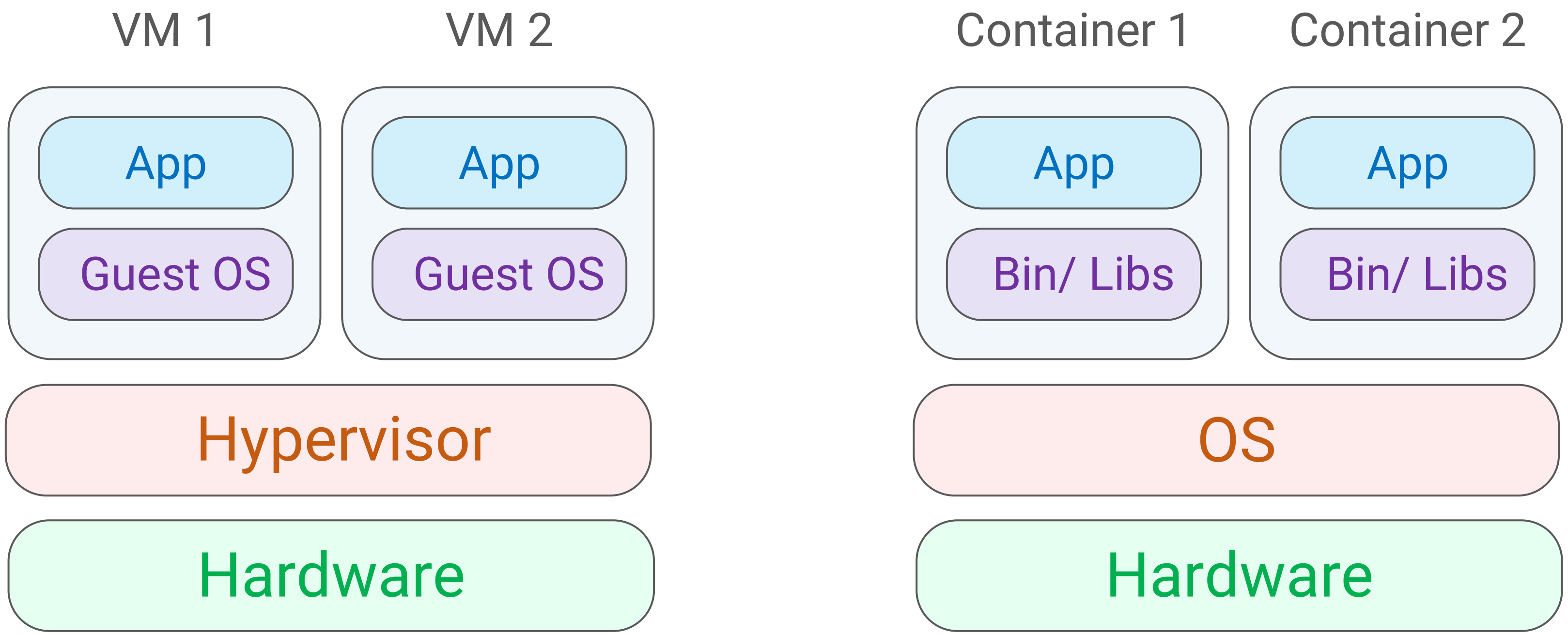
Hardware

Resource Sharing in Cloud



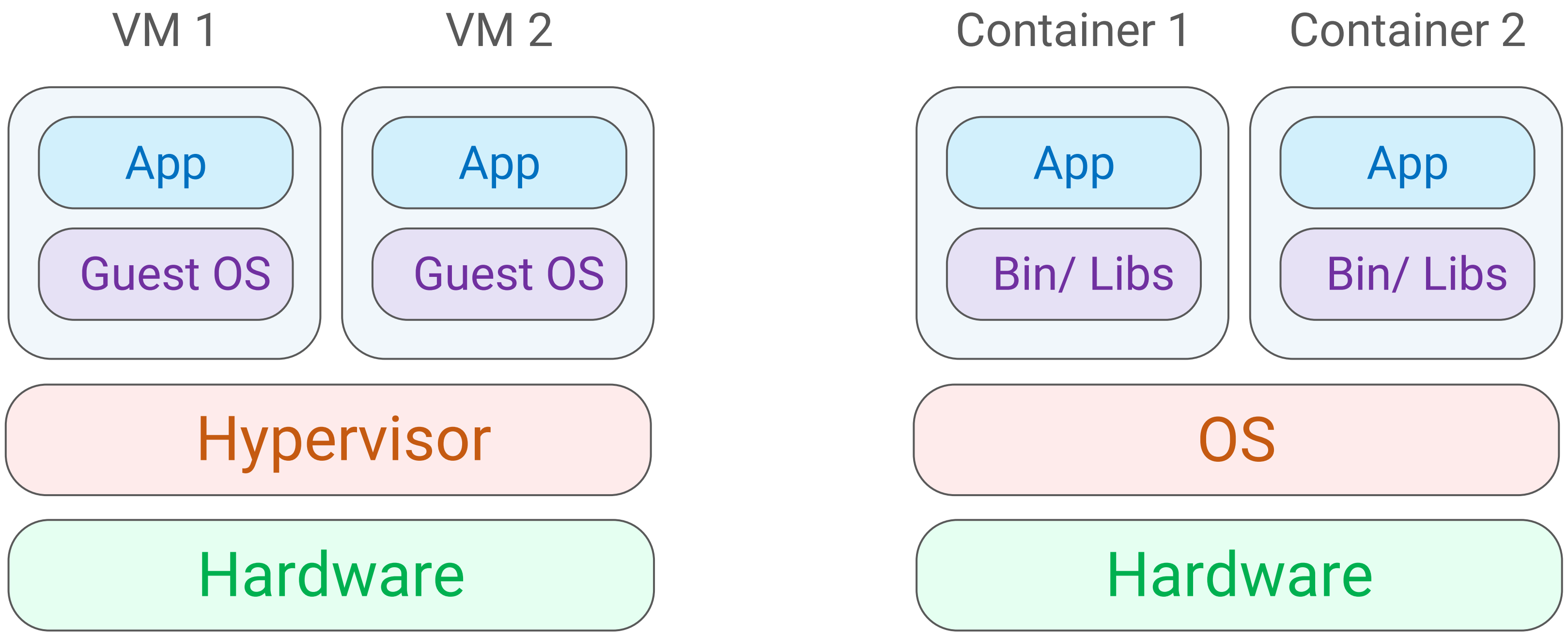
- Shared hardware resources for better utilization, e.g. Core, Memory, LLC, etc.

Resource Sharing in Cloud



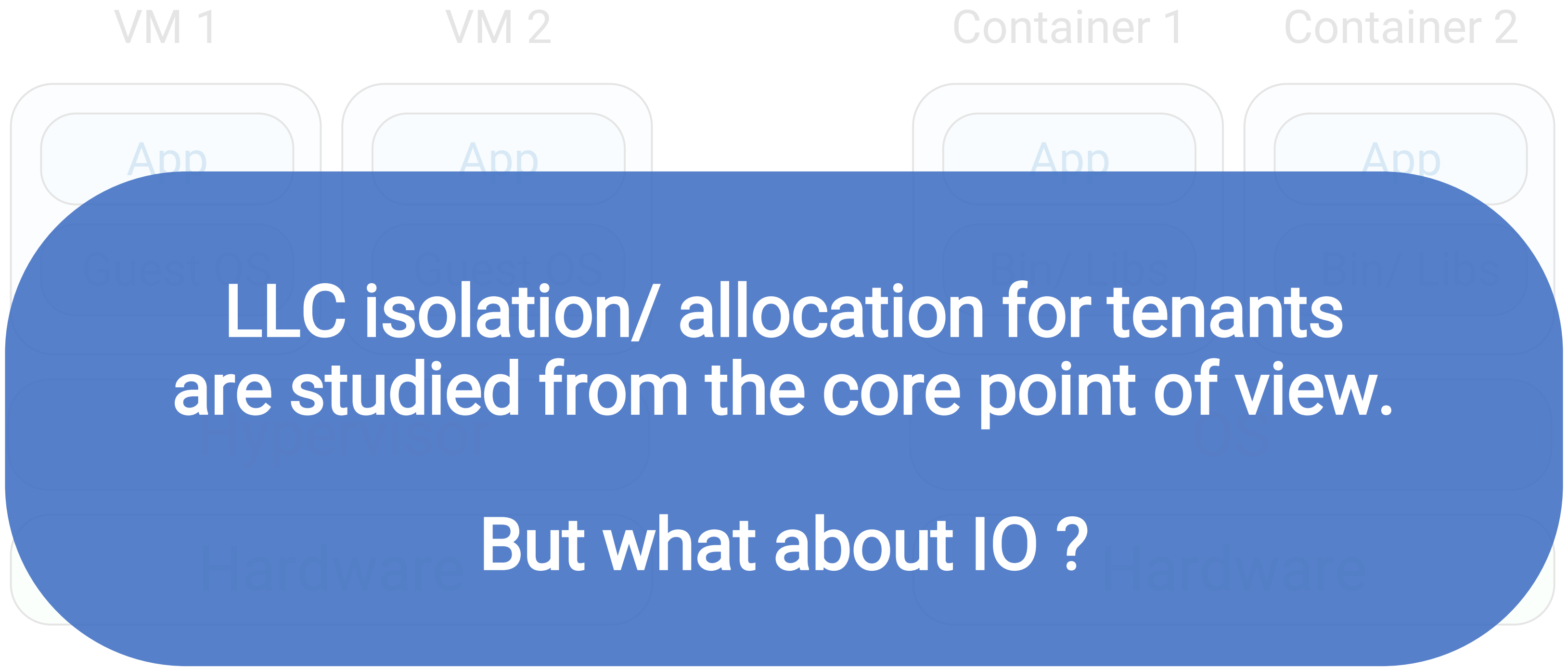
- Shared hardware resources for better utilization, e.g. Core, Memory, LLC, etc.
- Leads to interference: Causing performance degradations for tenants

Resource Sharing in Cloud



- Shared hardware resources for better utilization, e.g. Core, Memory, LLC, etc.
- Leads to interference: Causing performance degradations for tenants
- Need to allocate and isolate for different tenants: the focus is LLC

Resource Sharing in Cloud



LLC isolation/ allocation for tenants are studied from the core point of view.

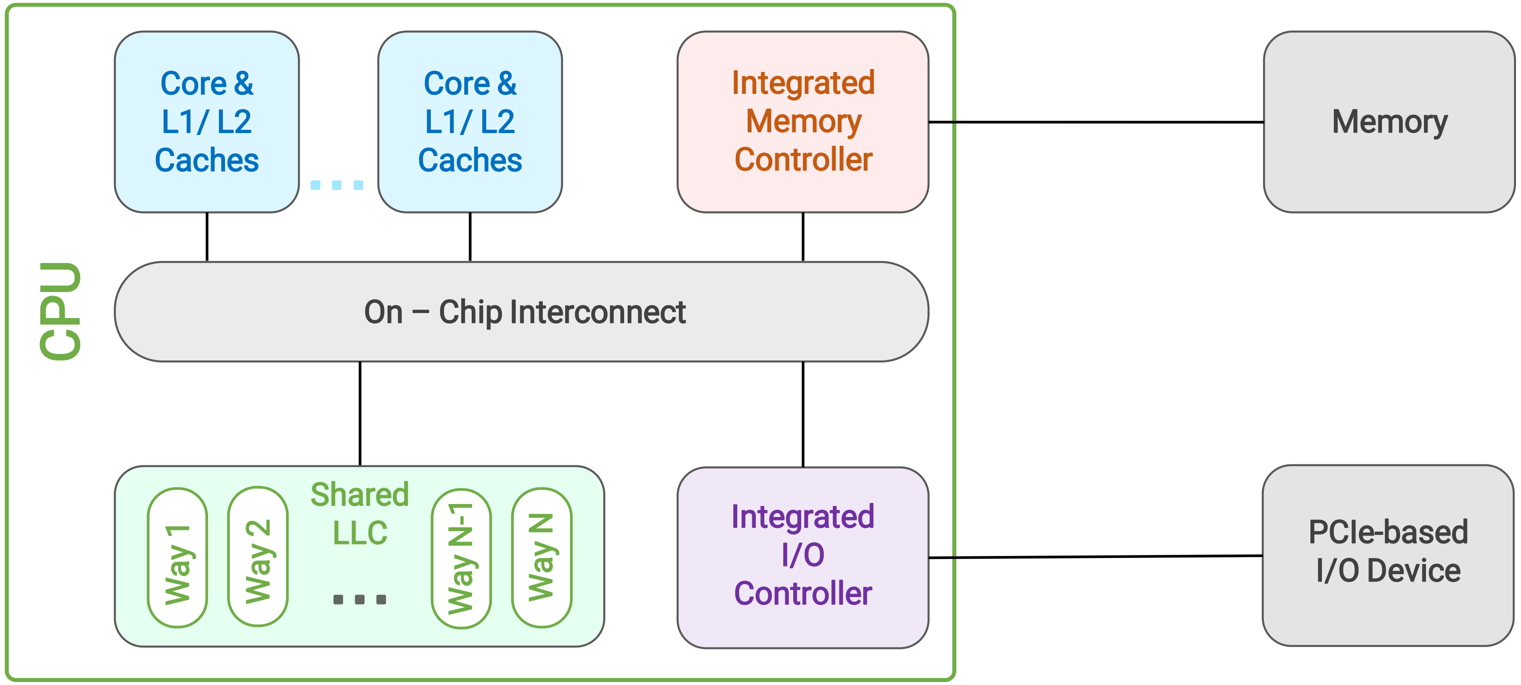
But what about IO ?

- Shared hardware resources for better utilization, e.g. Core, Memory, LLC, etc.
- Leads to interference: Causing performance degradations for tenants
- Need to allocate and isolate for different tenants: the focus is LLC

LLC Management and IO

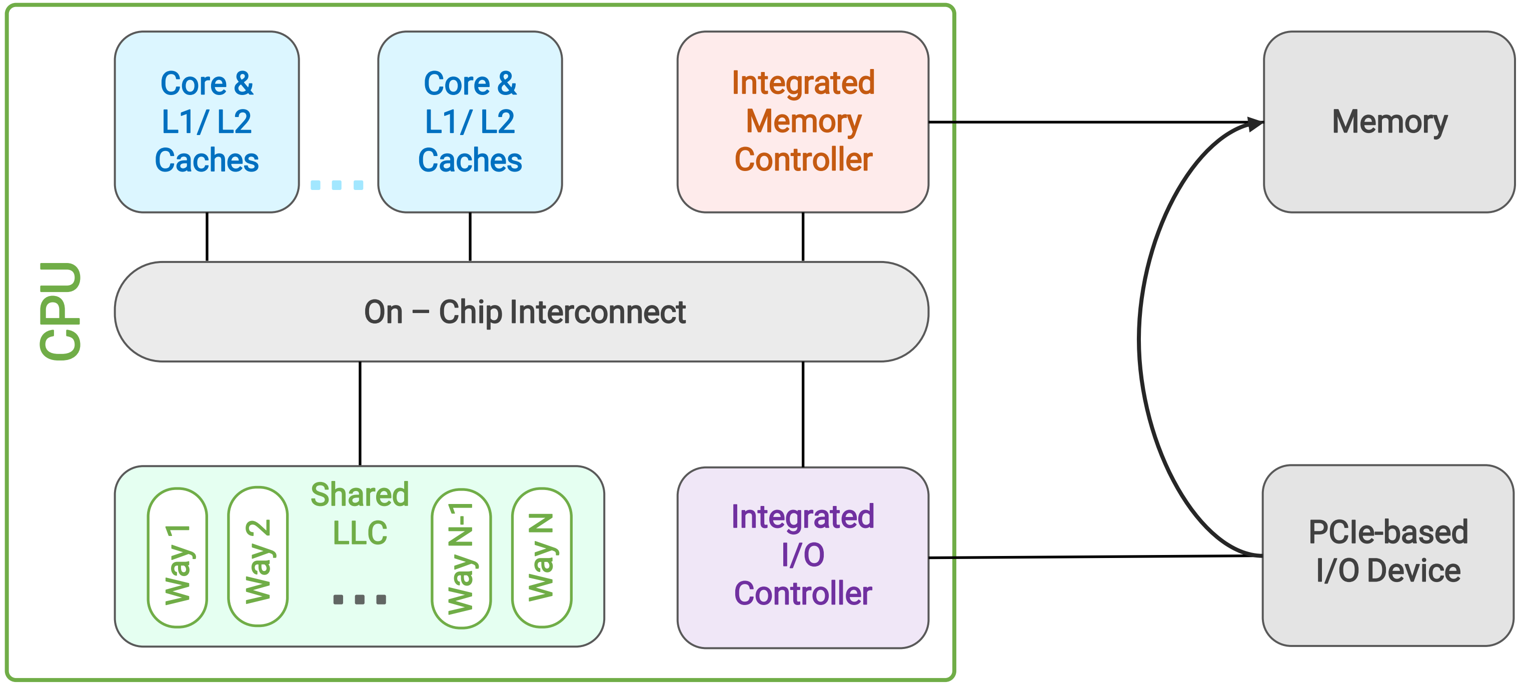
04-July-2022

LLC Management and IO



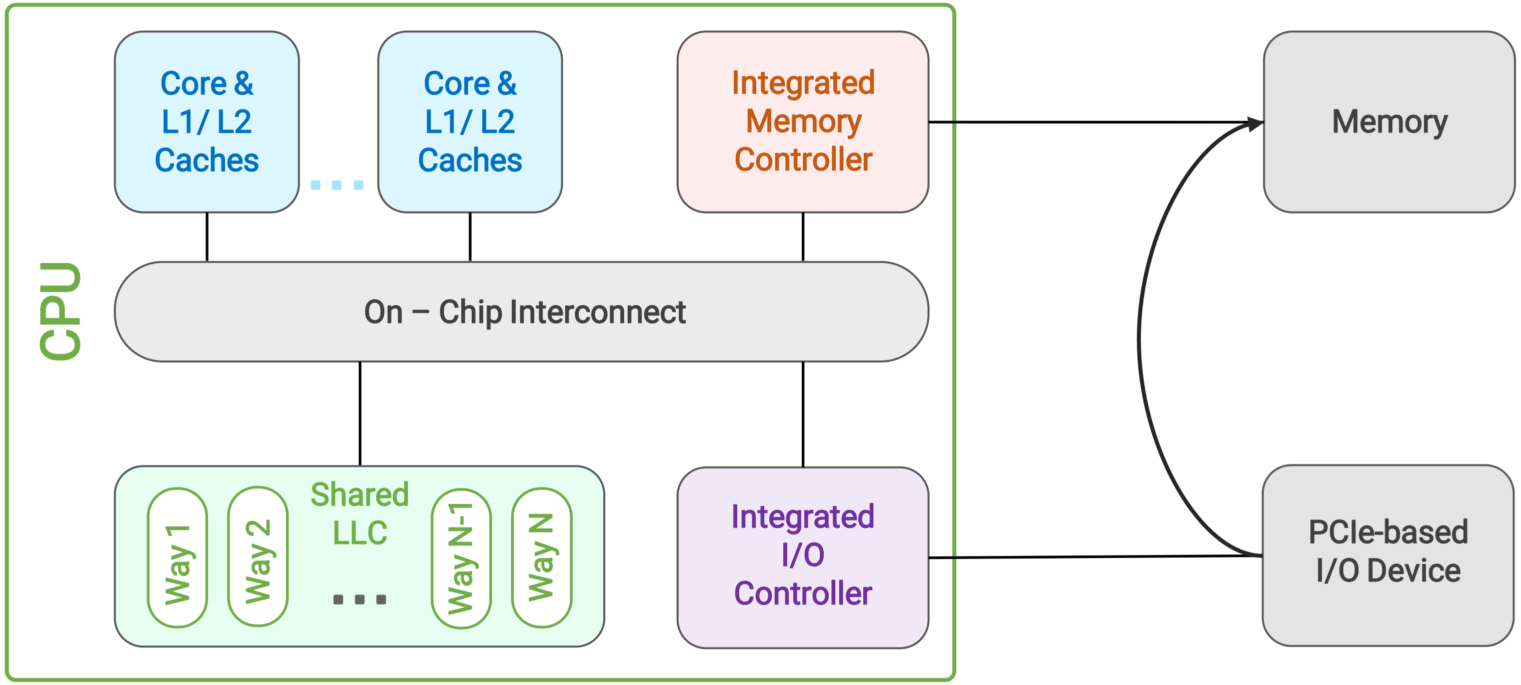
04-July-2022

LLC Management and IO



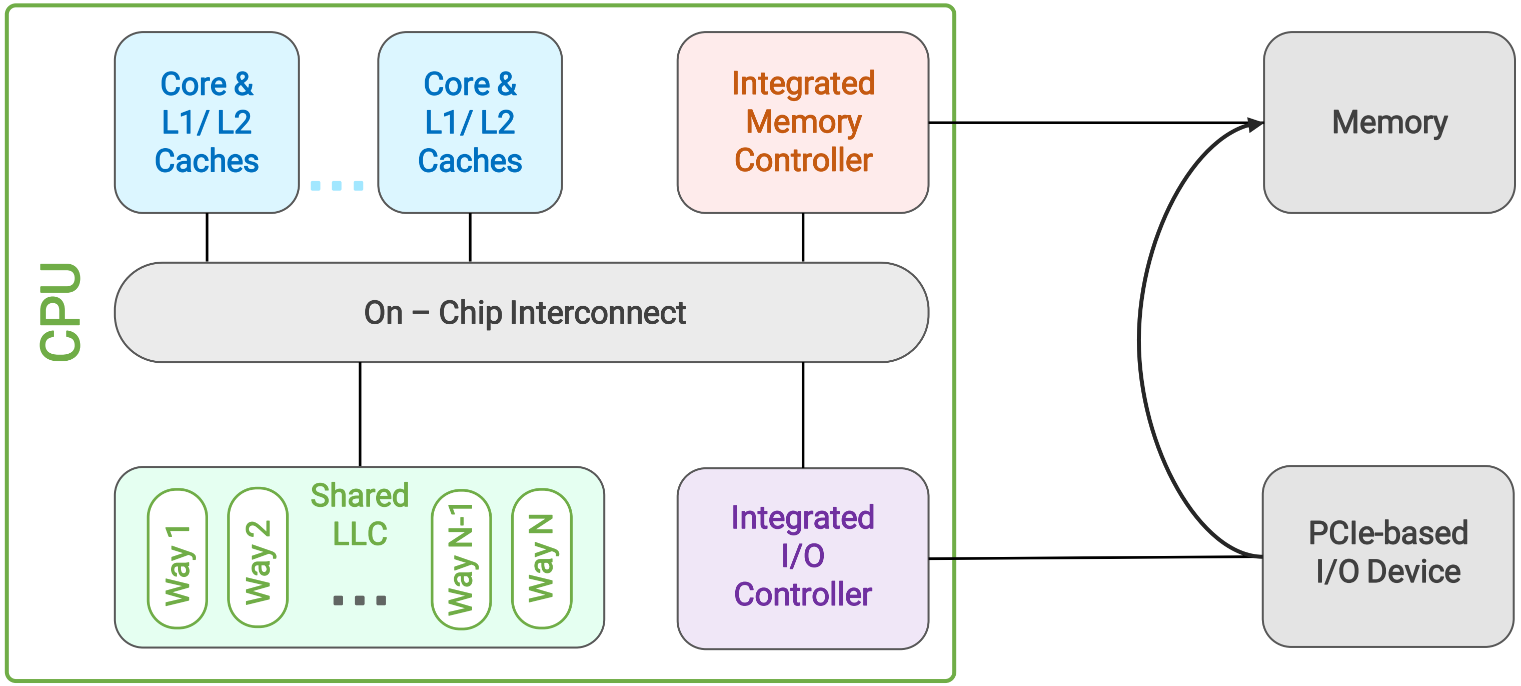
- Memory involvement for IO (DMA)

LLC Management and IO



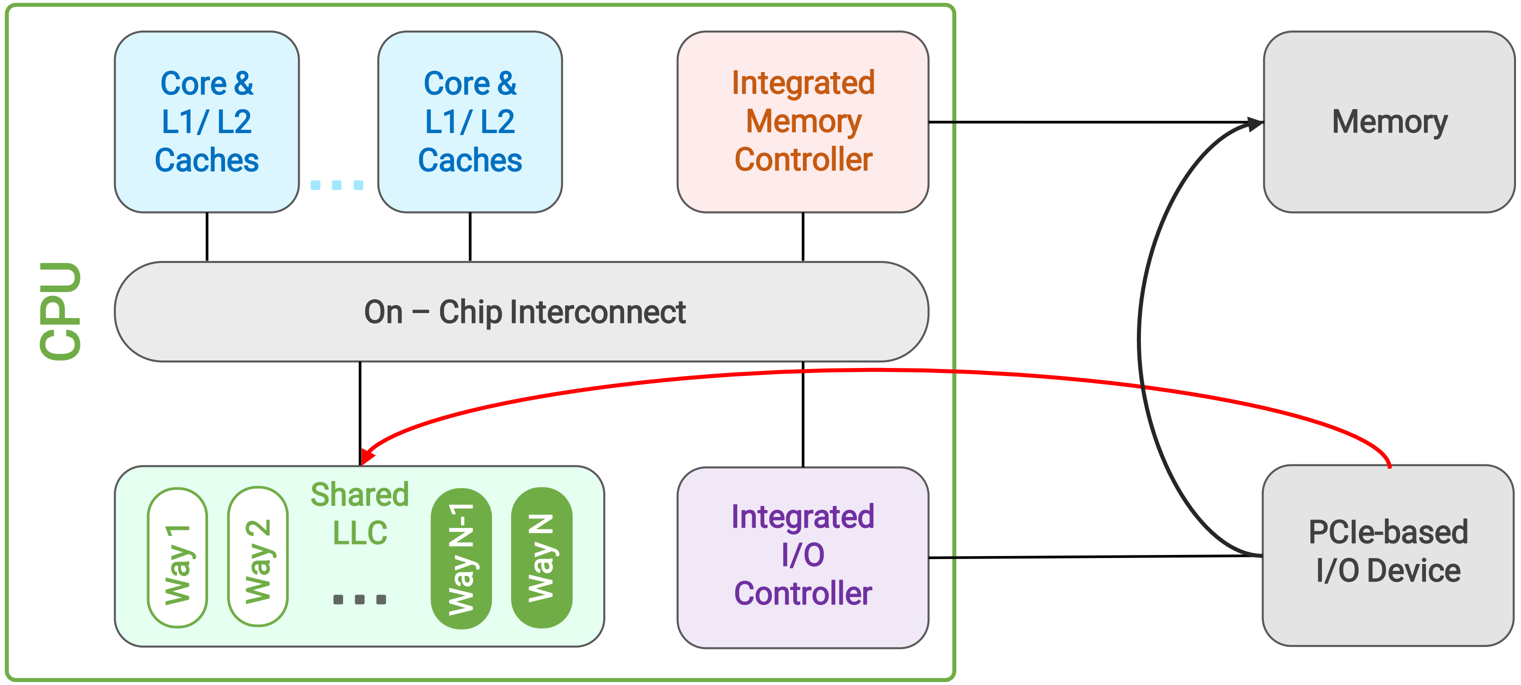
- Memory involvement for IO (DMA)
- Problems: High Speed Devices (NIC, SSDs)

LLC Management and IO



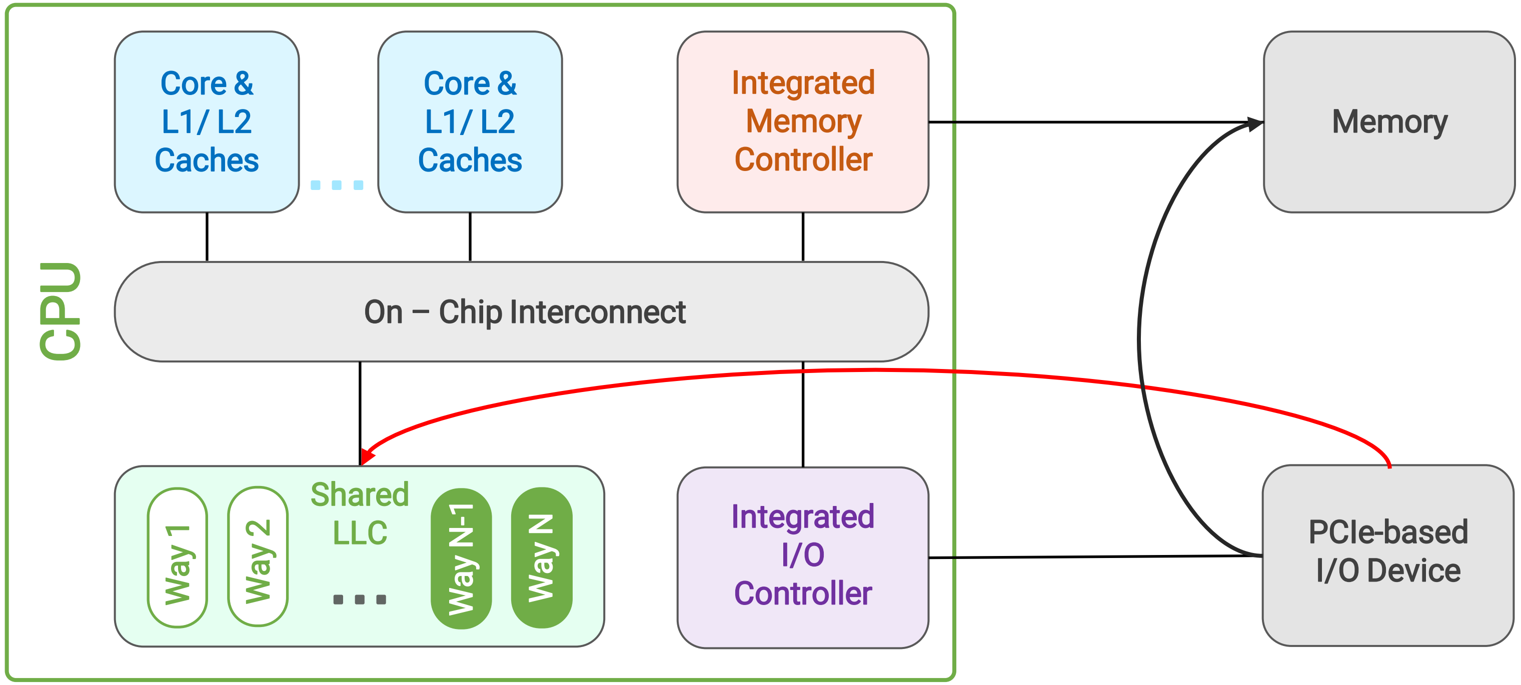
- Memory involvement for IO (DMA)
- Problems: High Speed Devices (NIC, SSDs), Poor memory bandwidth utilization

LLC Management and IO



- Memory involvement for IO (DMA)
- Problems: High Speed Devices (NIC, SSDs), Poor memory bandwidth utilization
- DDIO (2 LLCs ways)

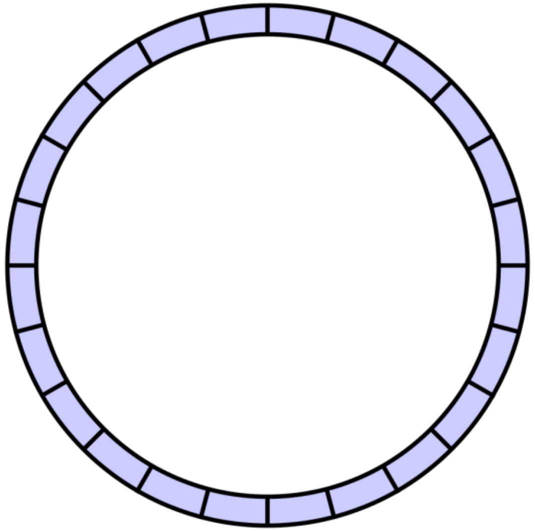
LLC Management and IO



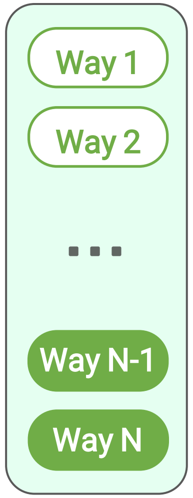
- Memory involvement for IO (DMA)
- Problems: High Speed Devices (NIC, SSDs), Poor memory bandwidth utilization
- DDIO (2 LLCs ways): introduces new challenges

Challenge 1: Leaky DMA

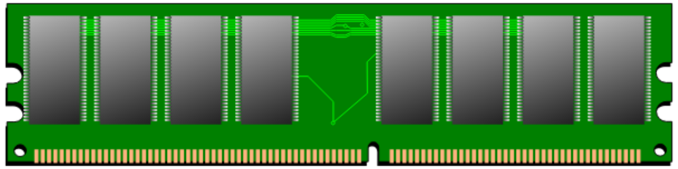
Challenge 1: Leaky DMA



Rx Ring Buffer

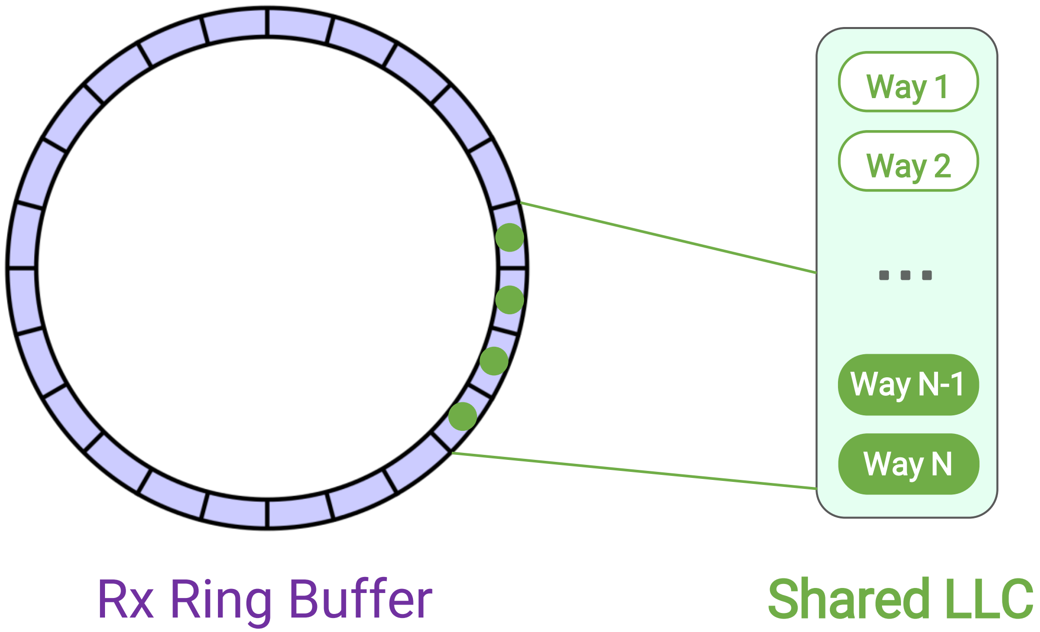


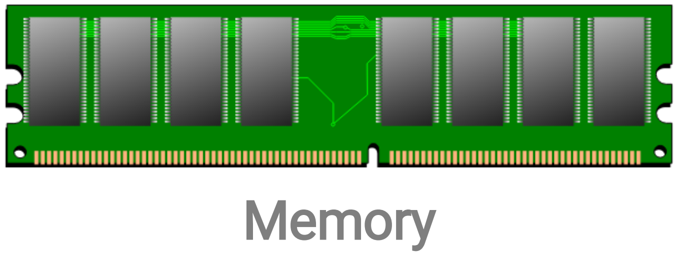
Shared LLC



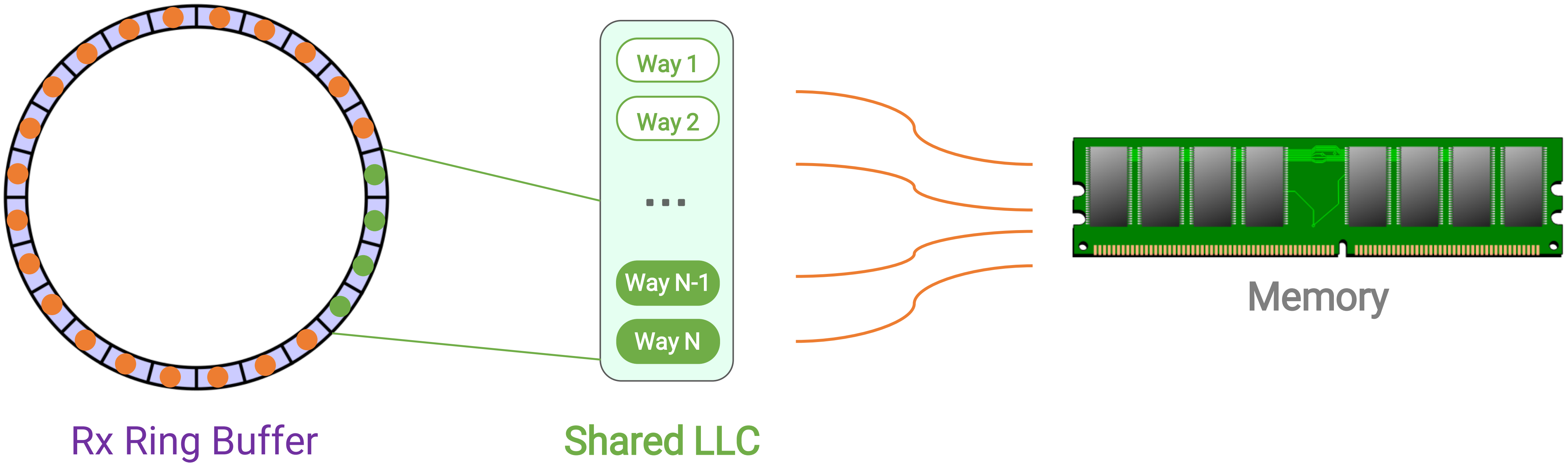
Memory

Challenge 1: Leaky DMA





Challenge 1: Leaky DMA



Challenge 1: Leaky DMA

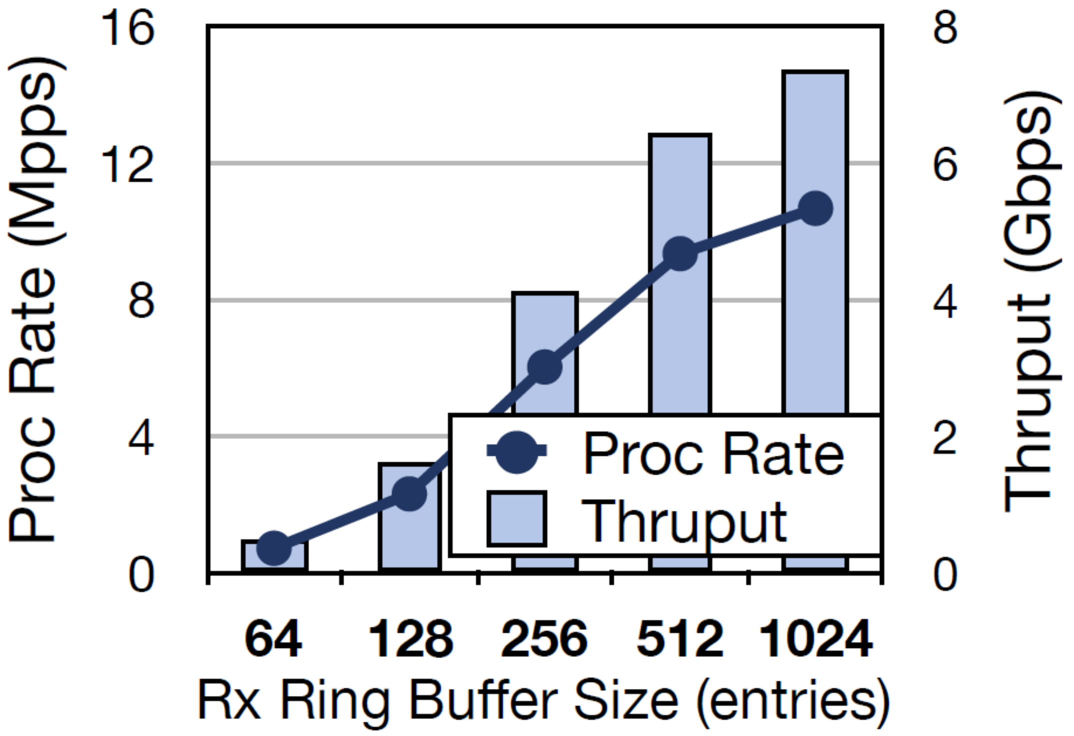
- ResQ proposes to reduce Rx buffer size

Challenge 1: Leaky DMA

- ResQ proposes to reduce Rx buffer size:
  - Very less buffer size for multiple tenants

Challenge 1: Leaky DMA

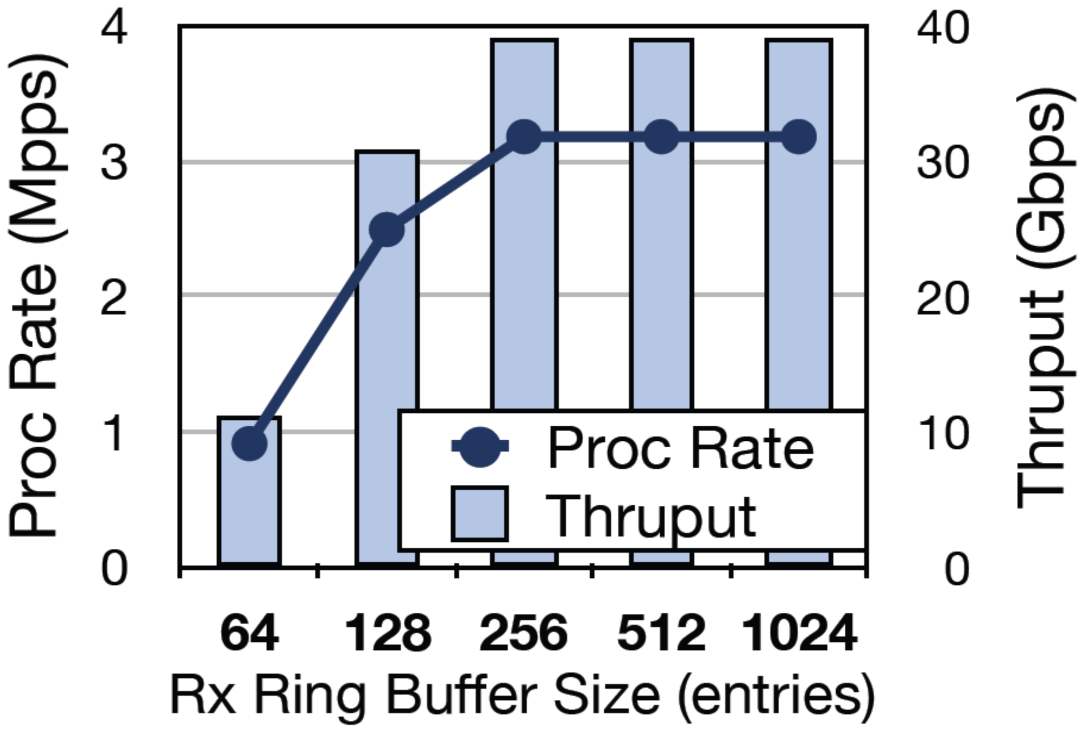
- ResQ proposes to reduce Rx buffer size:
  - Very less buffer size for multiple tenants
  - Throughput drop



(a) 64B small packet.

Challenge 1: Leaky DMA

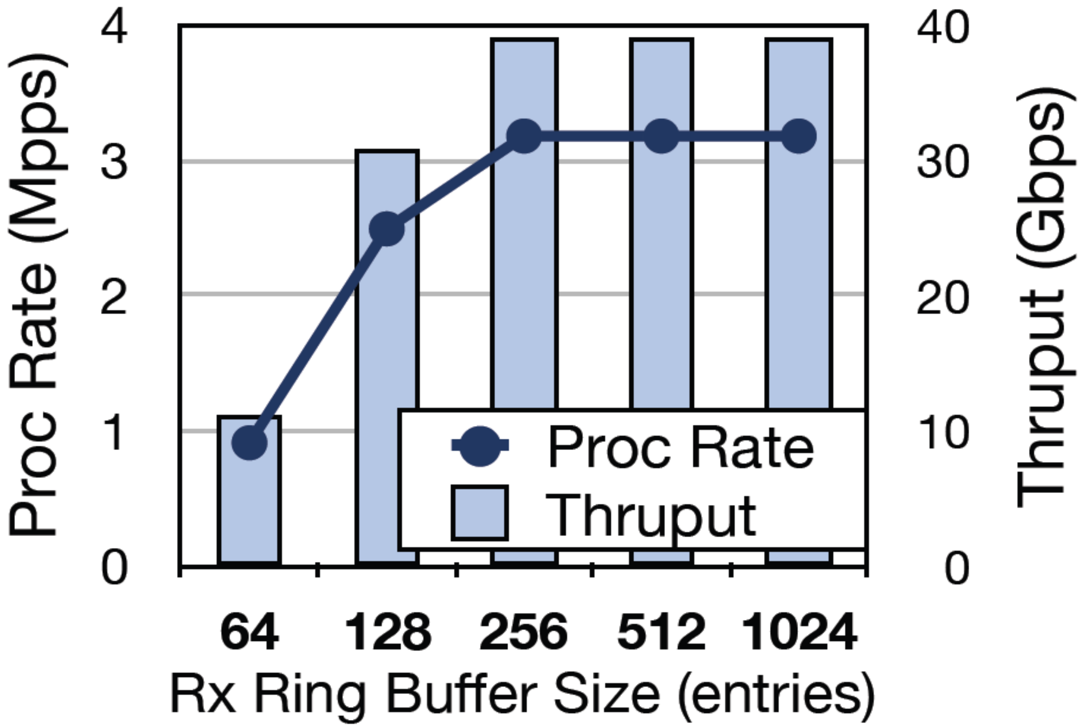
- ResQ proposes to reduce Rx buffer size:
  - Very less buffer size for multiple tenants
  - Throughput drop



(b) 1.5KB large packet.

Challenge 1: Leaky DMA

- ResQ proposes to reduce Rx buffer size:
    - Very less buffer size for multiple tenants
    - Throughput drop
- **How to adaptively change DDIO ways ?**

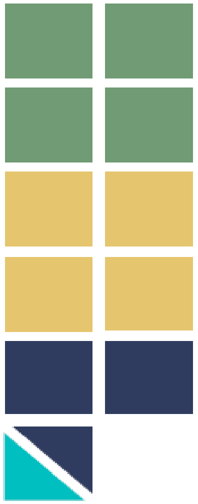


(b) 1.5KB large packet.

Challenge 2: Latent Contender

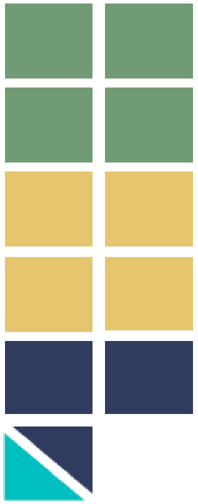
Challenge 2: Latent Contender

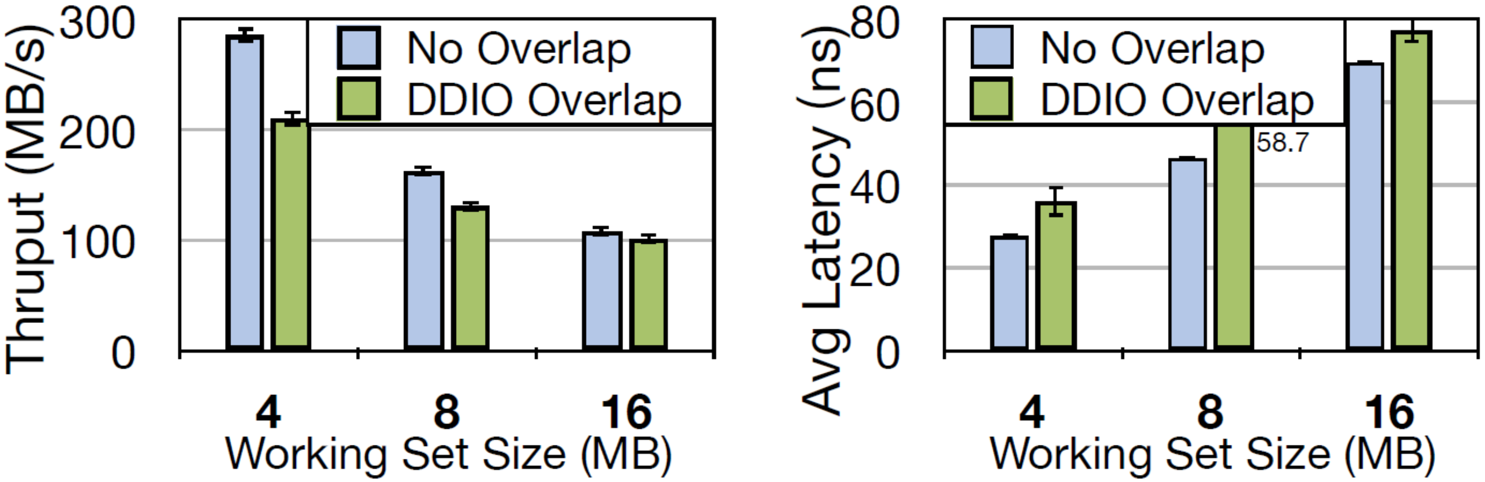
- Overlapping of DDIO ways with Core affect performance



Challenge 2: Latent Contender

- Overlapping of DDIO ways with Core affect performance





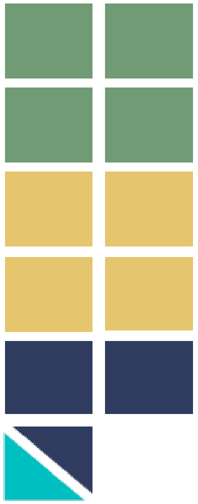
(a) Throughput.

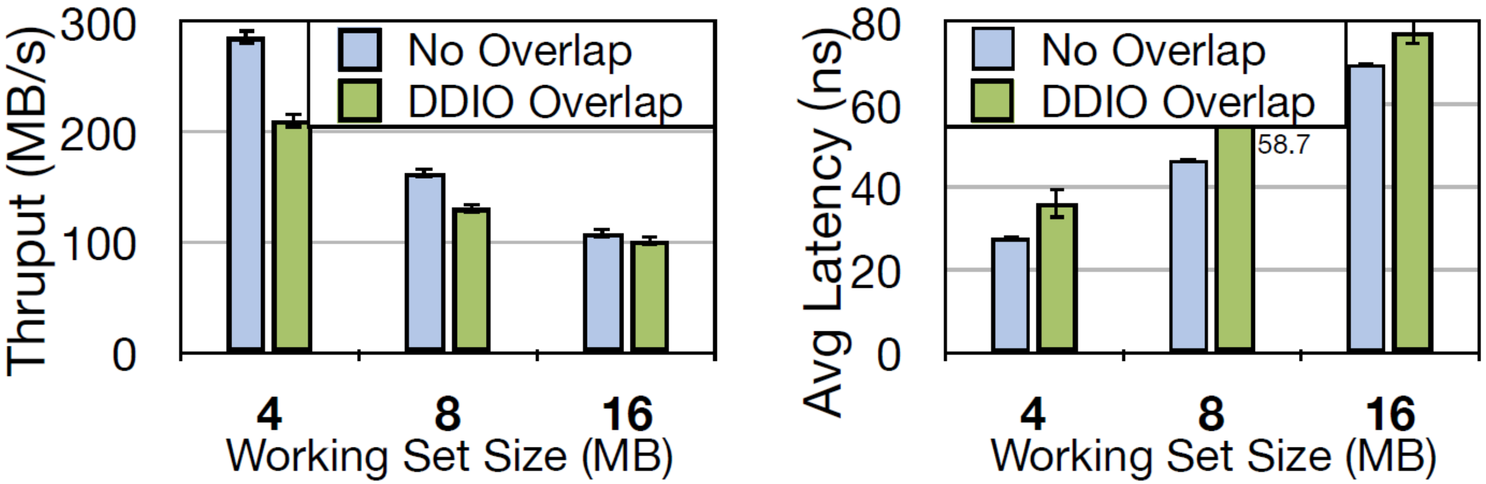
(b) Latency.

- DDIO overlap worsen the X-Mem's (random read pattern) throughput and avg latency

Challenge 2: Latent Contender

- Overlapping of DDIO ways with Core affect performance

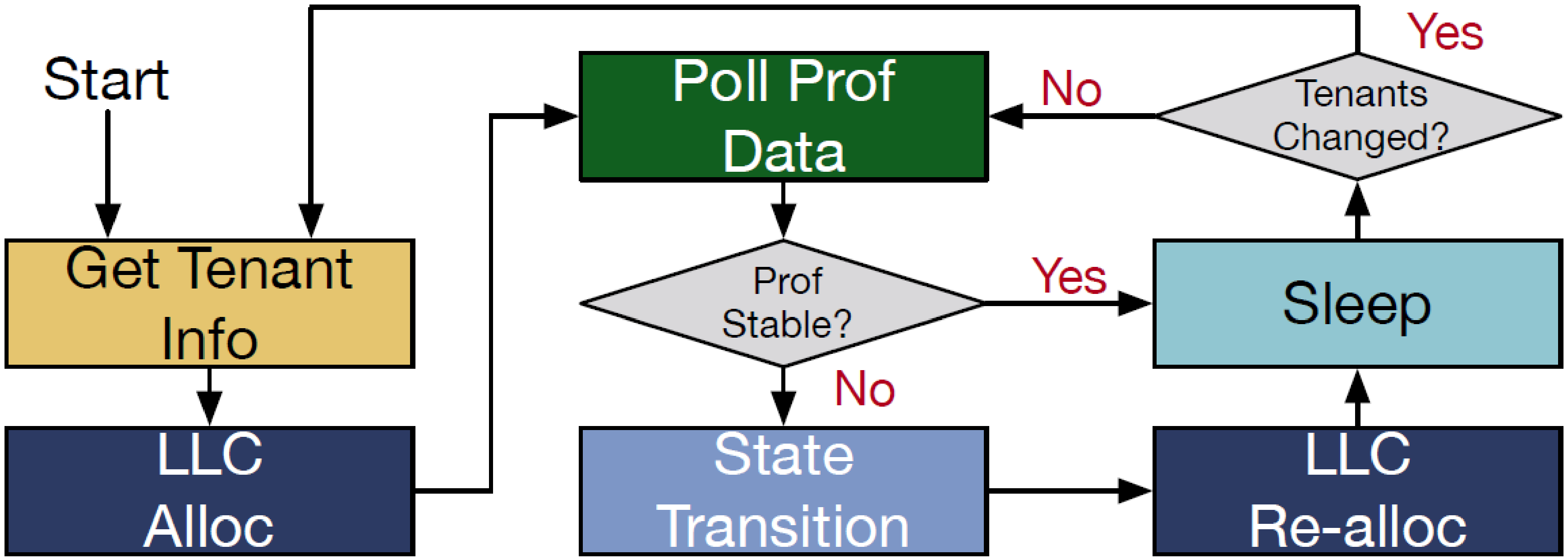




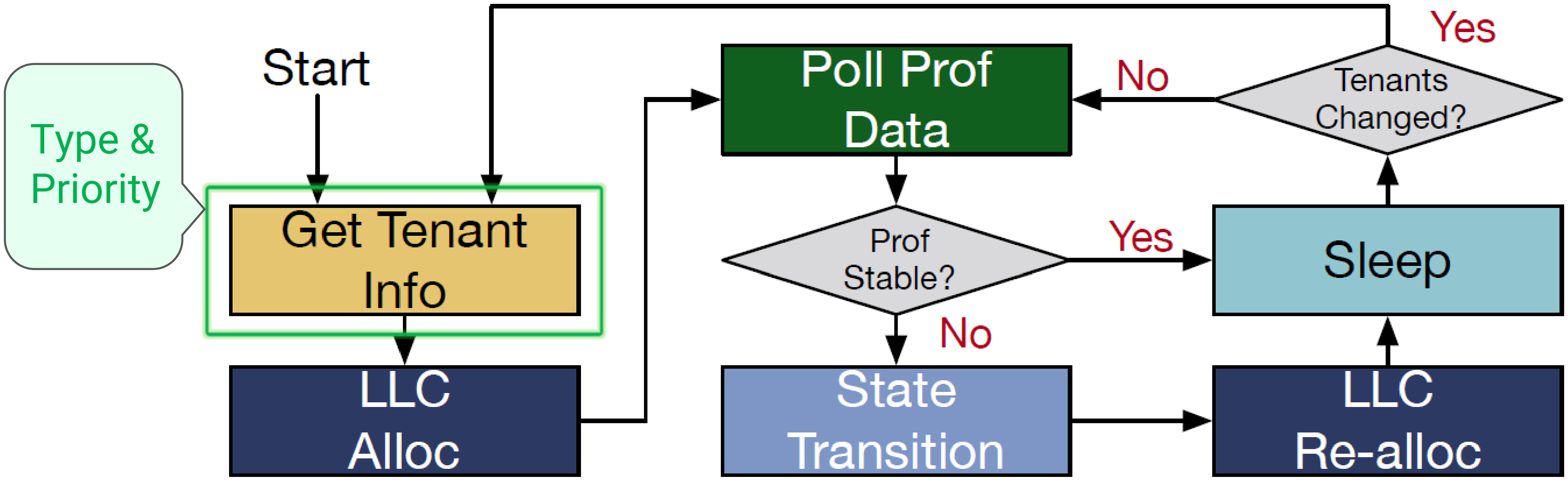
(a) Throughput. **(b)** Latency.

- DDIO overlap worsen the X-Mem’s (random read pattern) throughput and avg latency
- **How to share LLC ways with DDIO ?**

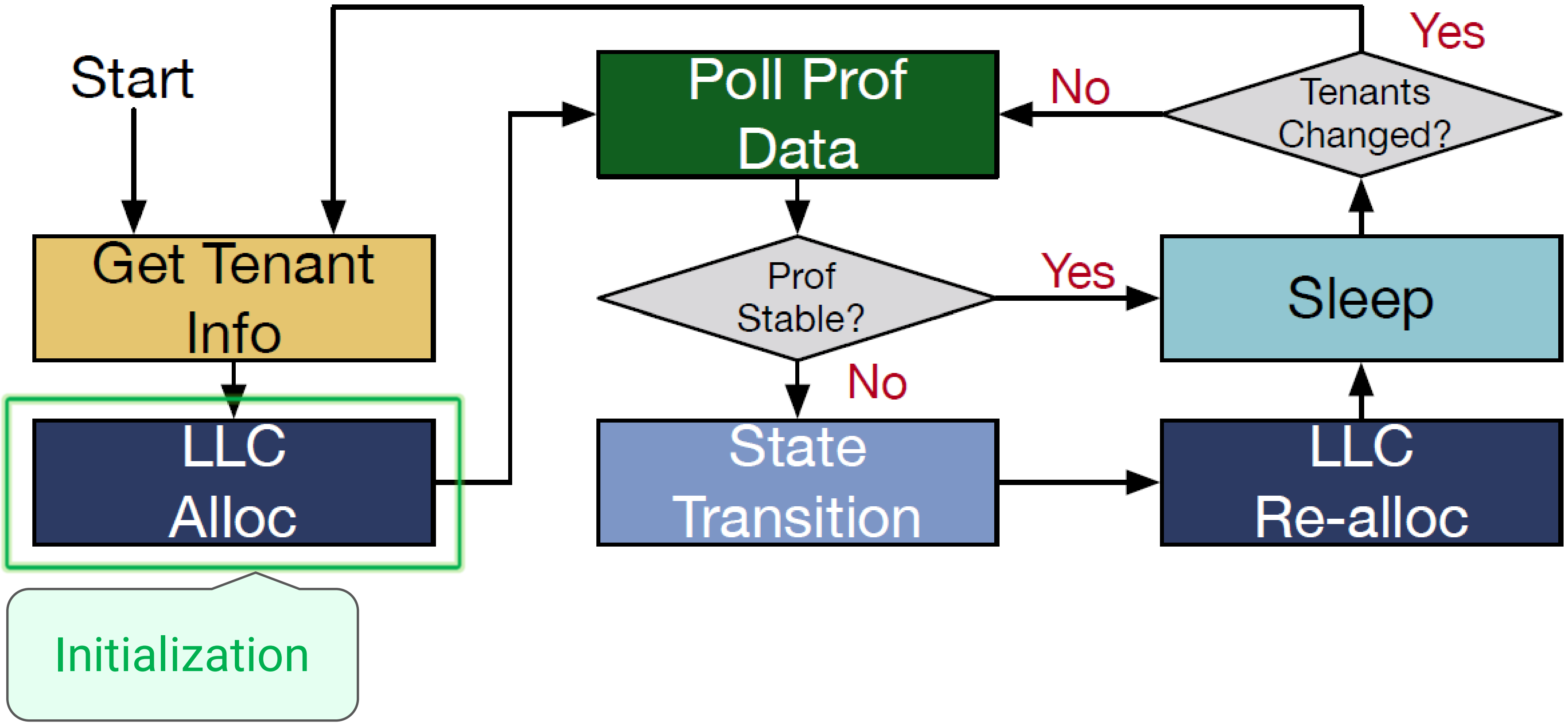
IAT: Design


Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
No
Yes

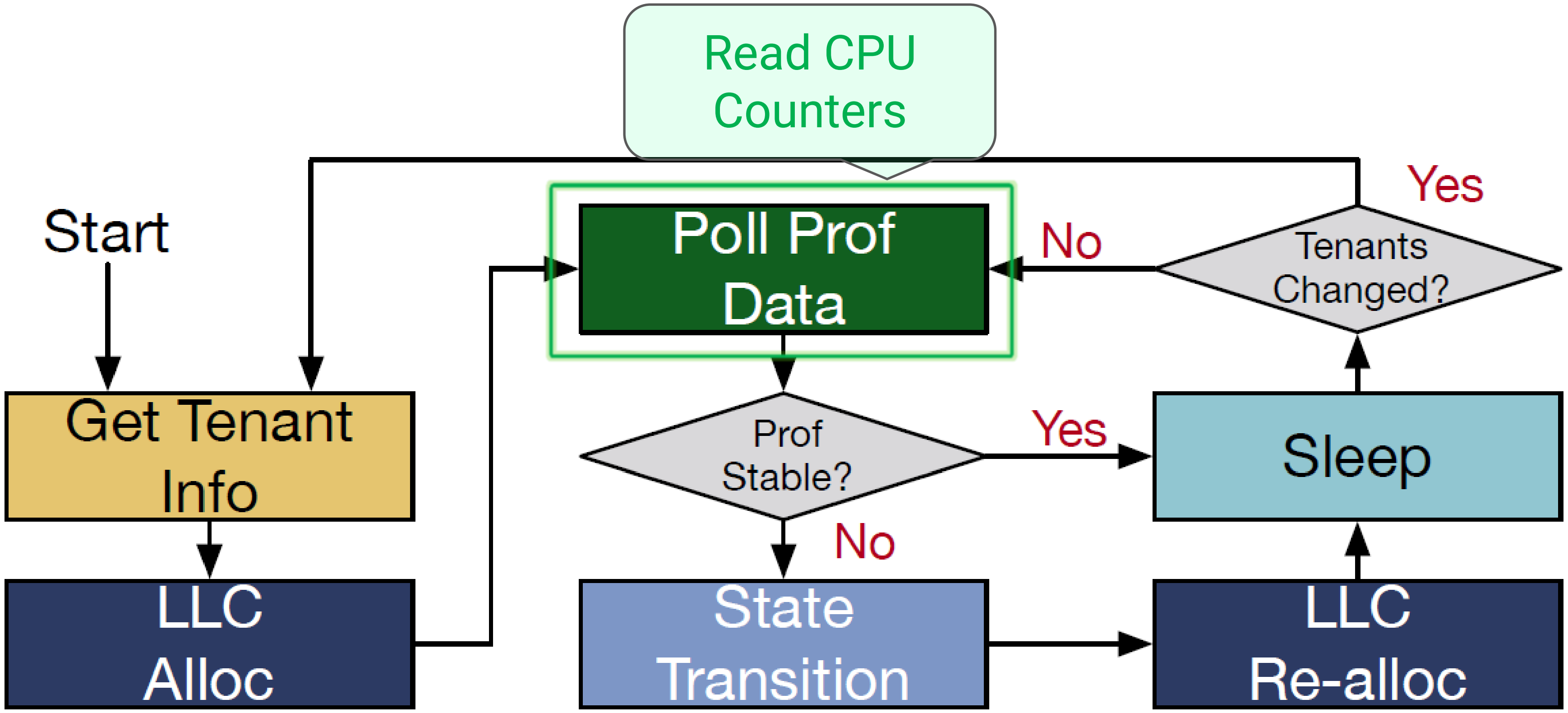
IAT: Design


Type & Priority
Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
No
Yes

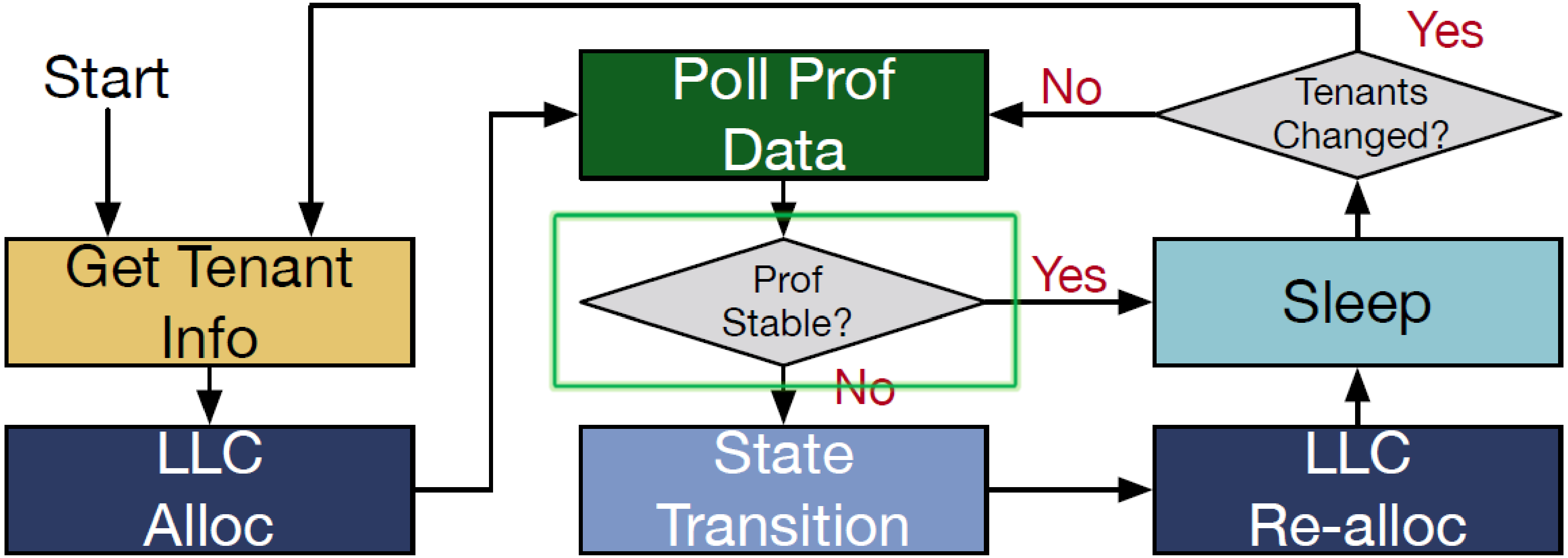
IAT: Design


Start
Get Tenant Info
LLC Alloc
Initialization
Poll Prof Data
Prof Stable?
Yes
No
State Transition
LLC Re-alloc
Sleep
Tenants Changed?
No
Yes

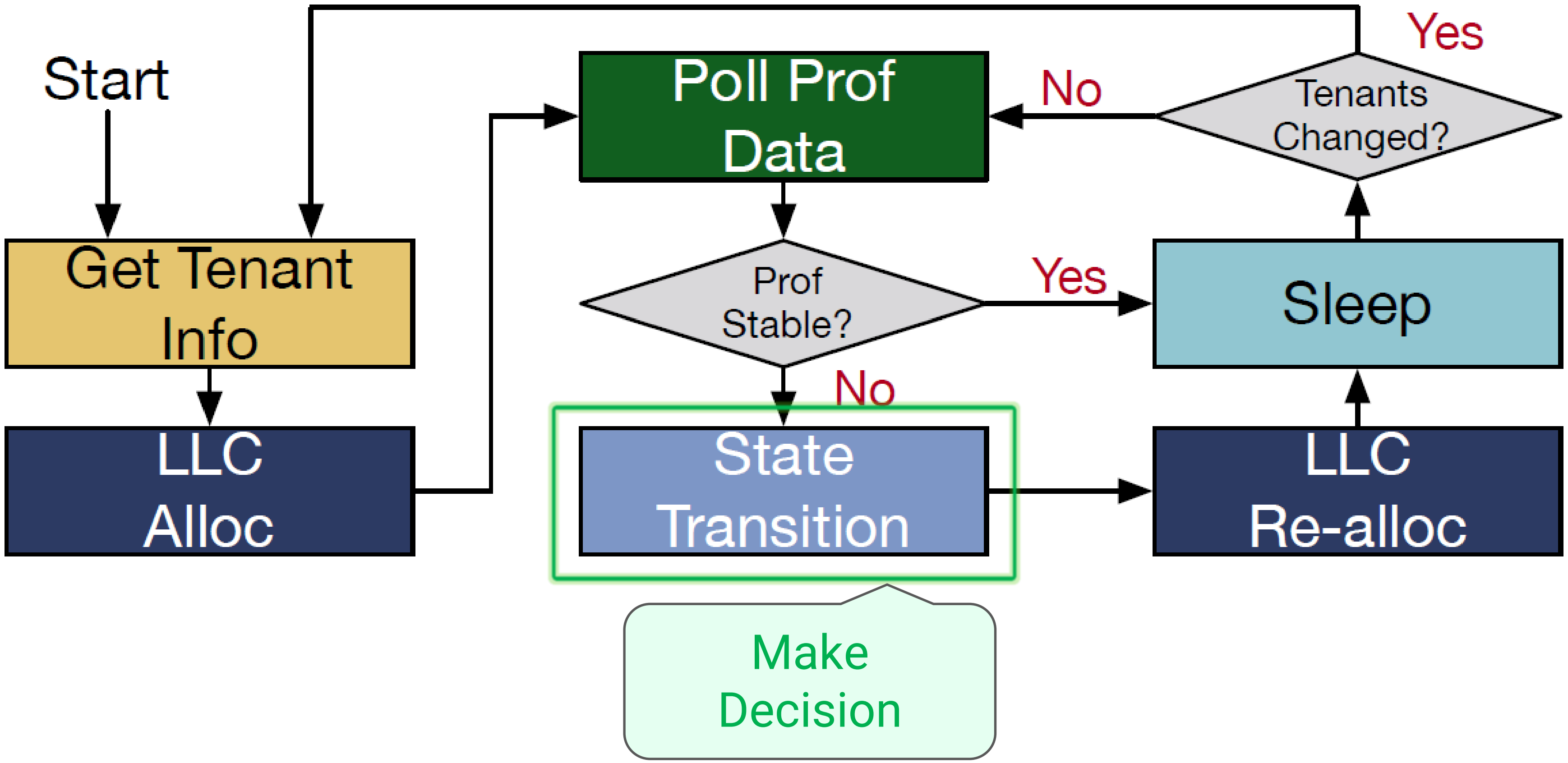
IAT: Design


Read CPU Counters
Start
Poll Prof Data
No
Tenants Changed?
Yes
Get Tenant Info
Prof Stable?
Yes
Sleep
No
LLC Alloc
State Transition
LLC Re-alloc

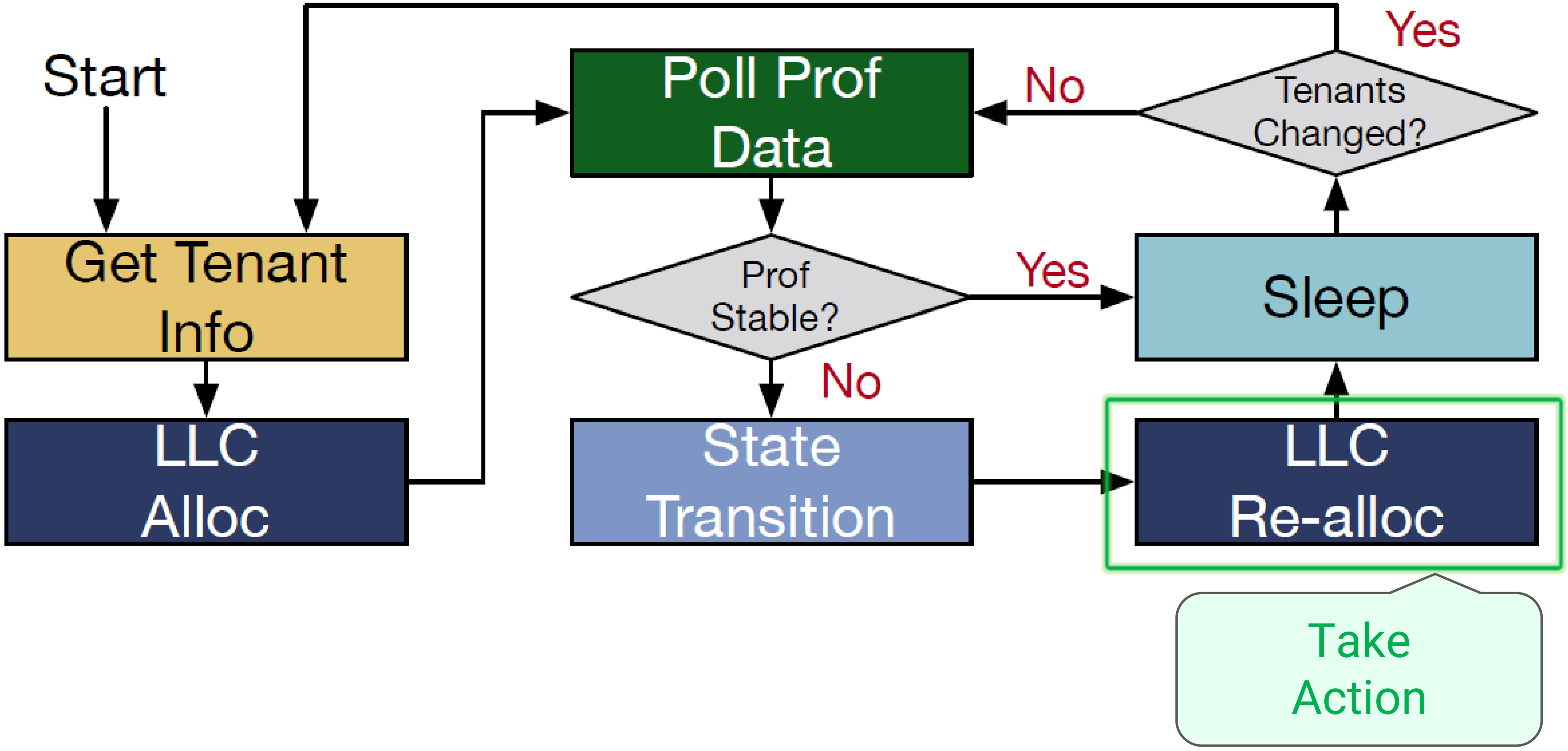
IAT: Design


Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
Yes
No

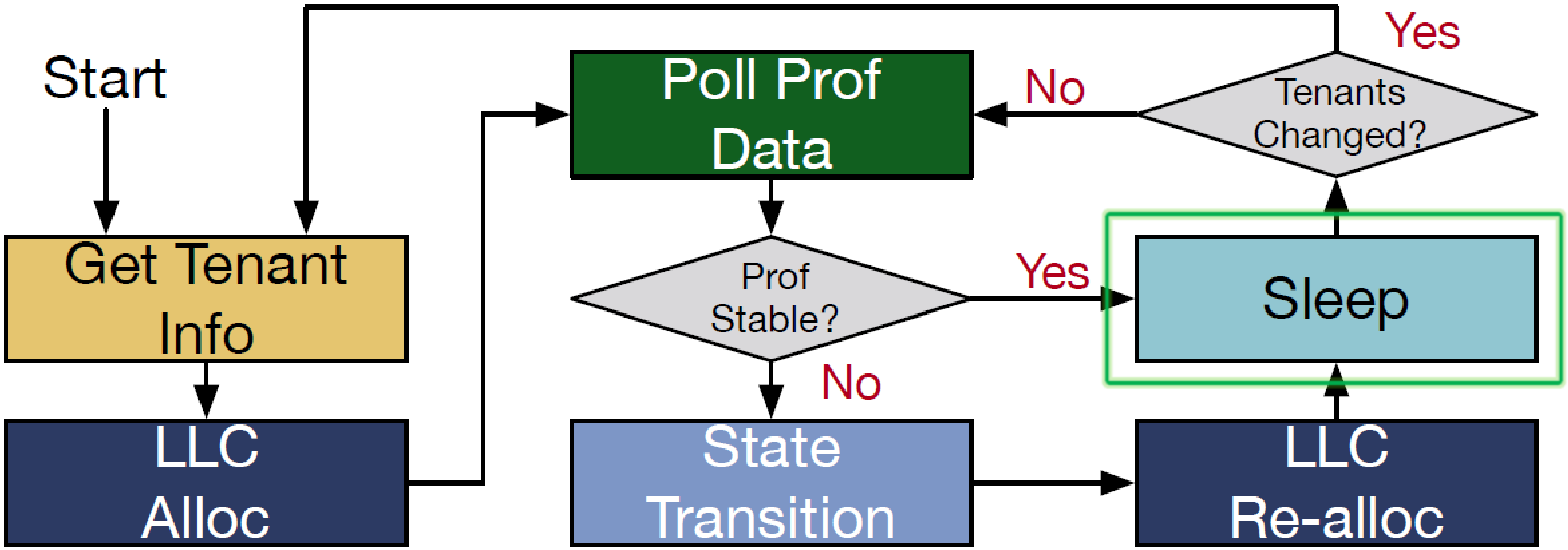
IAT: Design


Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
Yes
No
Make Decision

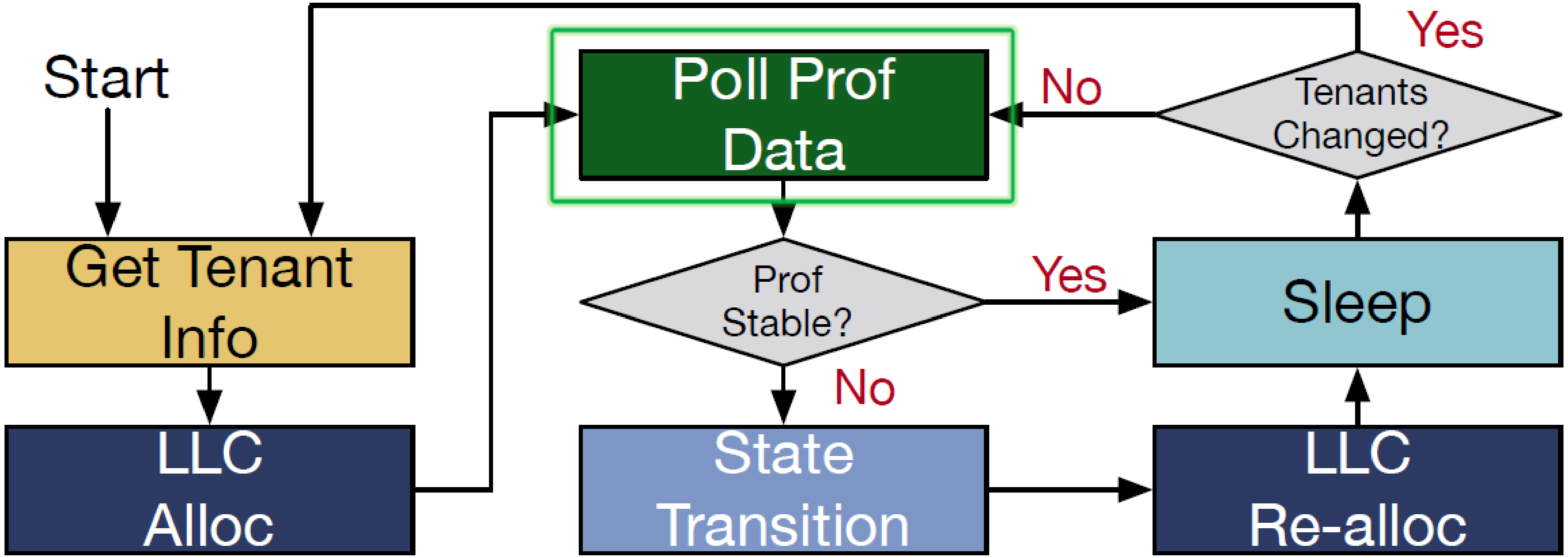
IAT: Design


Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
No
Yes
Take Action

IAT: Design


Start
Get Tenant Info
LLC Alloc
Poll Prof Data
Prof Stable?
Yes
No
State Transition
Sleep
LLC Re-alloc
Tenants Changed?
Yes
No

IAT: Poll Prof Data



IAT: Poll Prof Data

○ IPC

IAT: Poll Prof Data

- IPC: performance of an application

IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss

IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss: memory/ cache access characteristics of a workload

IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss: memory/ cache access characteristics of a workload
- DDIO hit and miss (chip wide metrics)

IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss: memory/ cache access characteristics of a workload
- DDIO hit and miss (chip wide metrics):
    - Hit when targeted cache-line in LLC

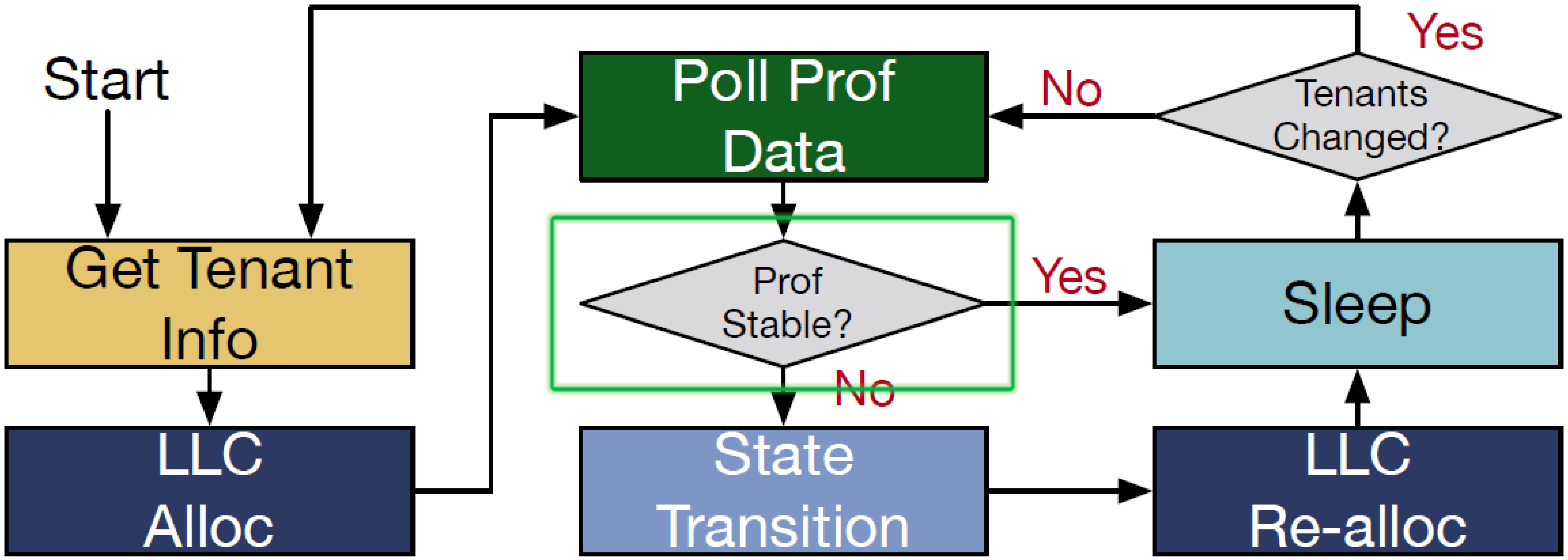
IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss: memory/ cache access characteristics of a workload
- DDIO hit and miss (chip wide metrics):
    - Hit when targeted cache-line in LLC
    - Miss when a victim has to be evicted for DDIO

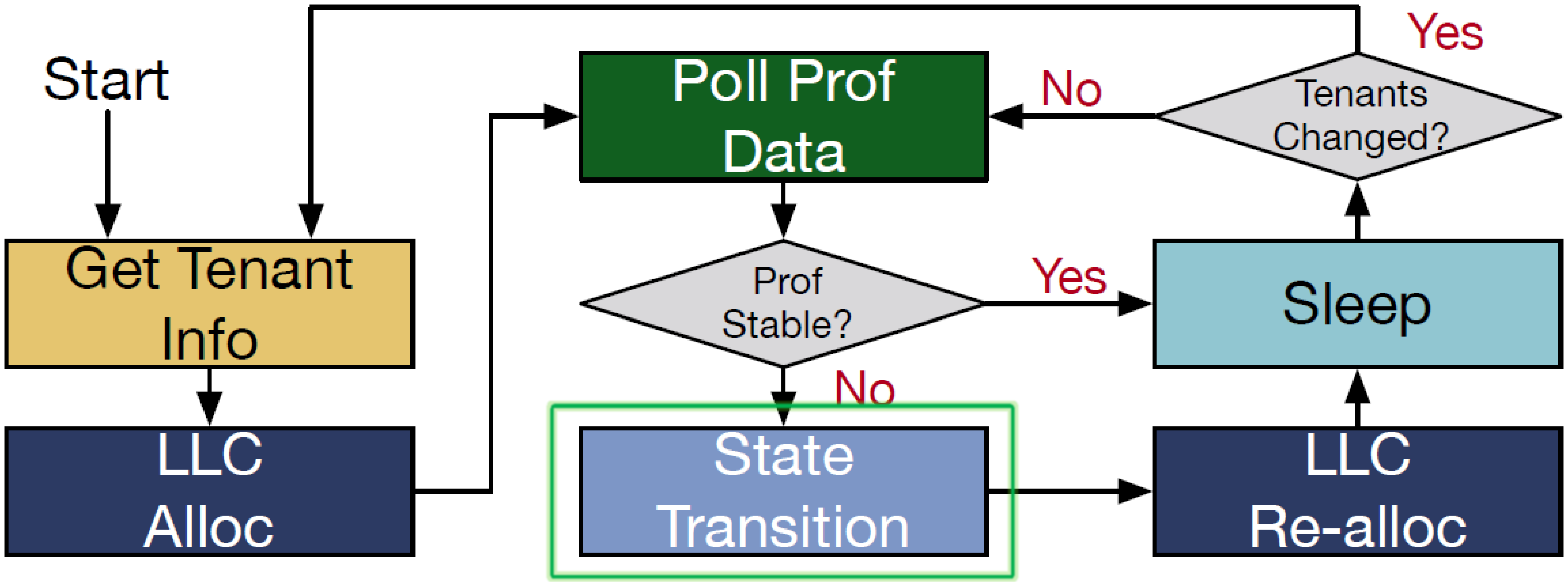
IAT: Poll Prof Data

- IPC: performance of an application
- LLC ref and miss: memory/ cache access characteristics of a workload
- DDIO hit and miss (chip wide metrics):
  - Hit when targeted cache-line in LLC
  - Miss when a victim has to be evicted for DDIO
  - Tells about IO pressure on LLC

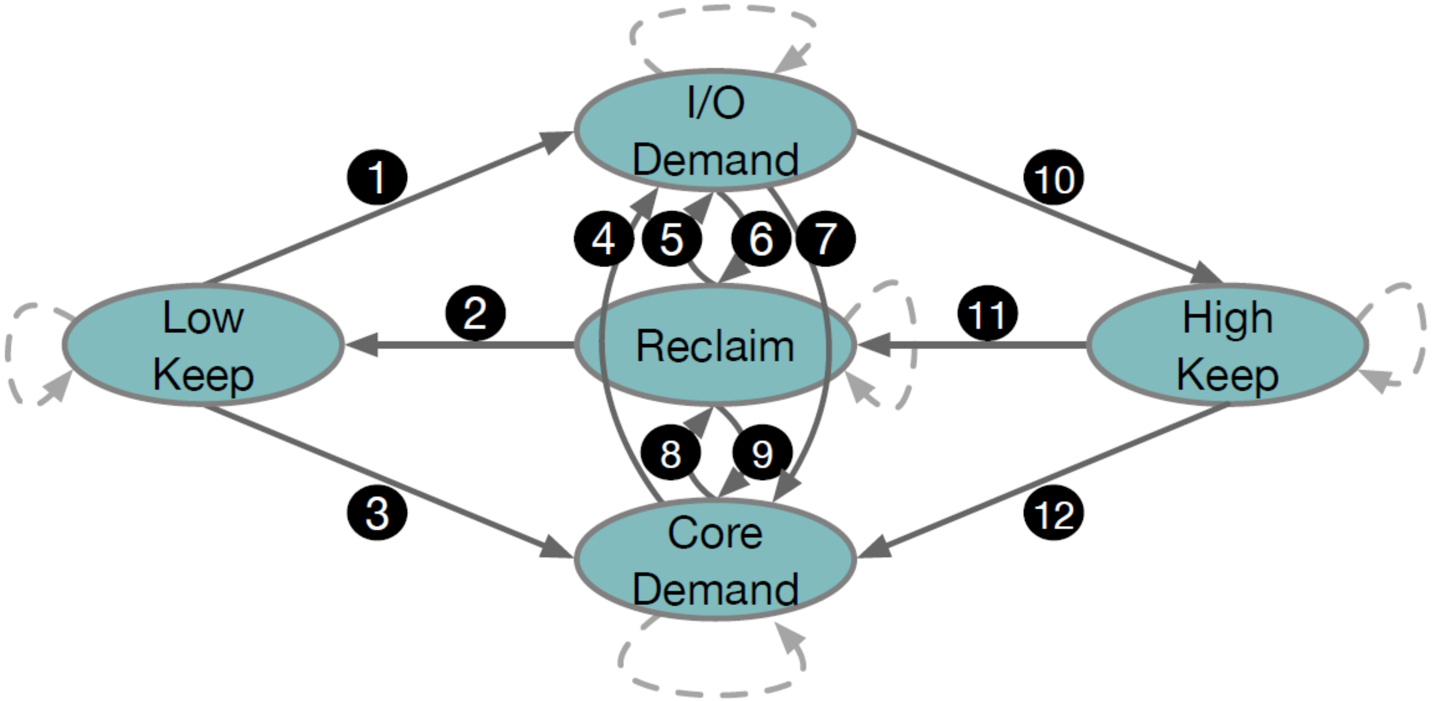
IAT: State Transition



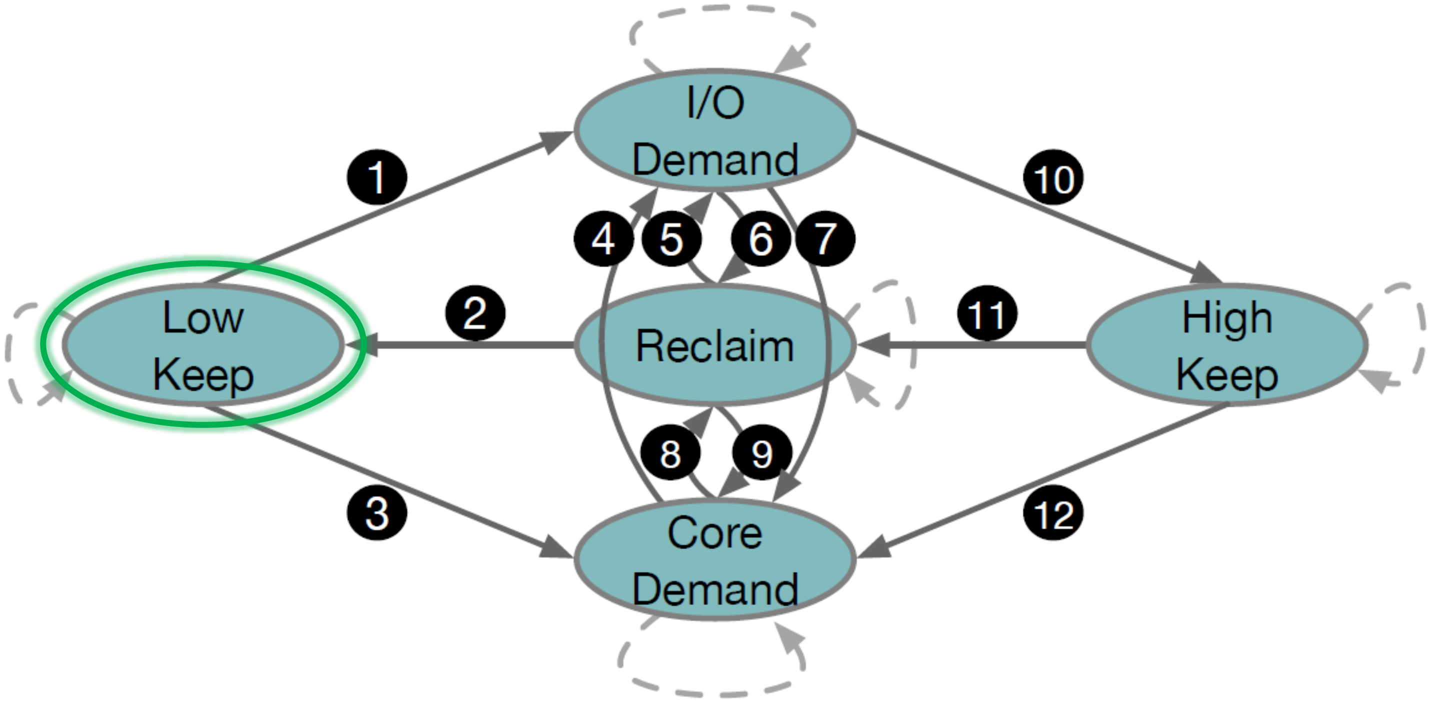
IAT: State Transition



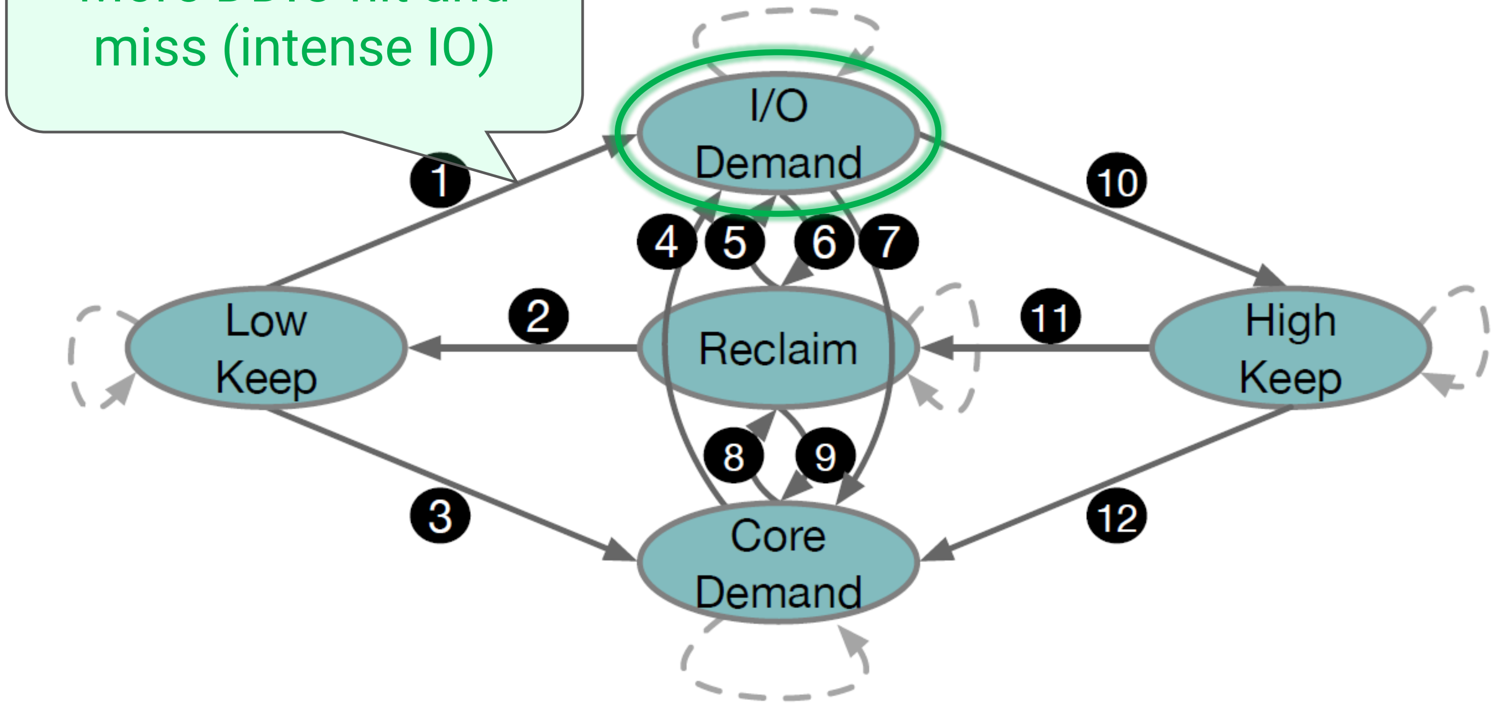
IAT: State Transition



IAT: State Transition

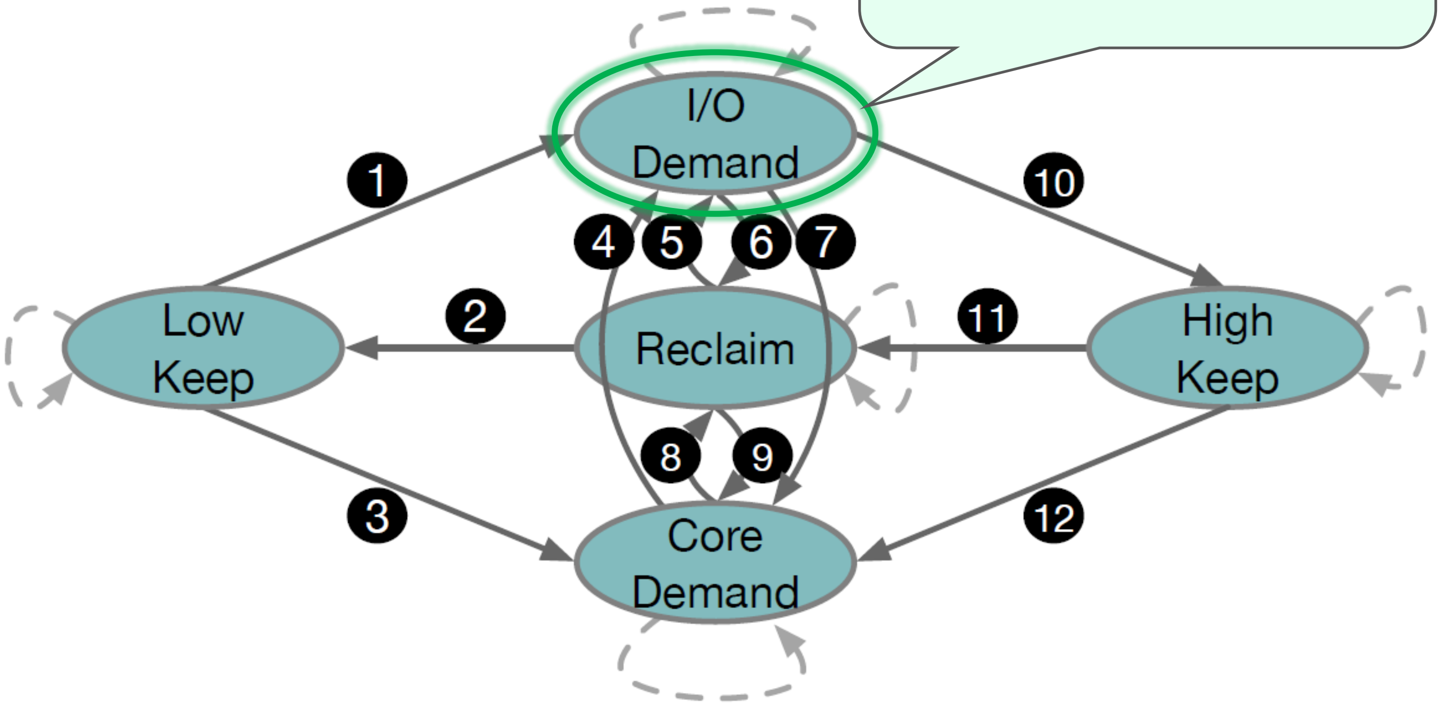


IAT: State Transition



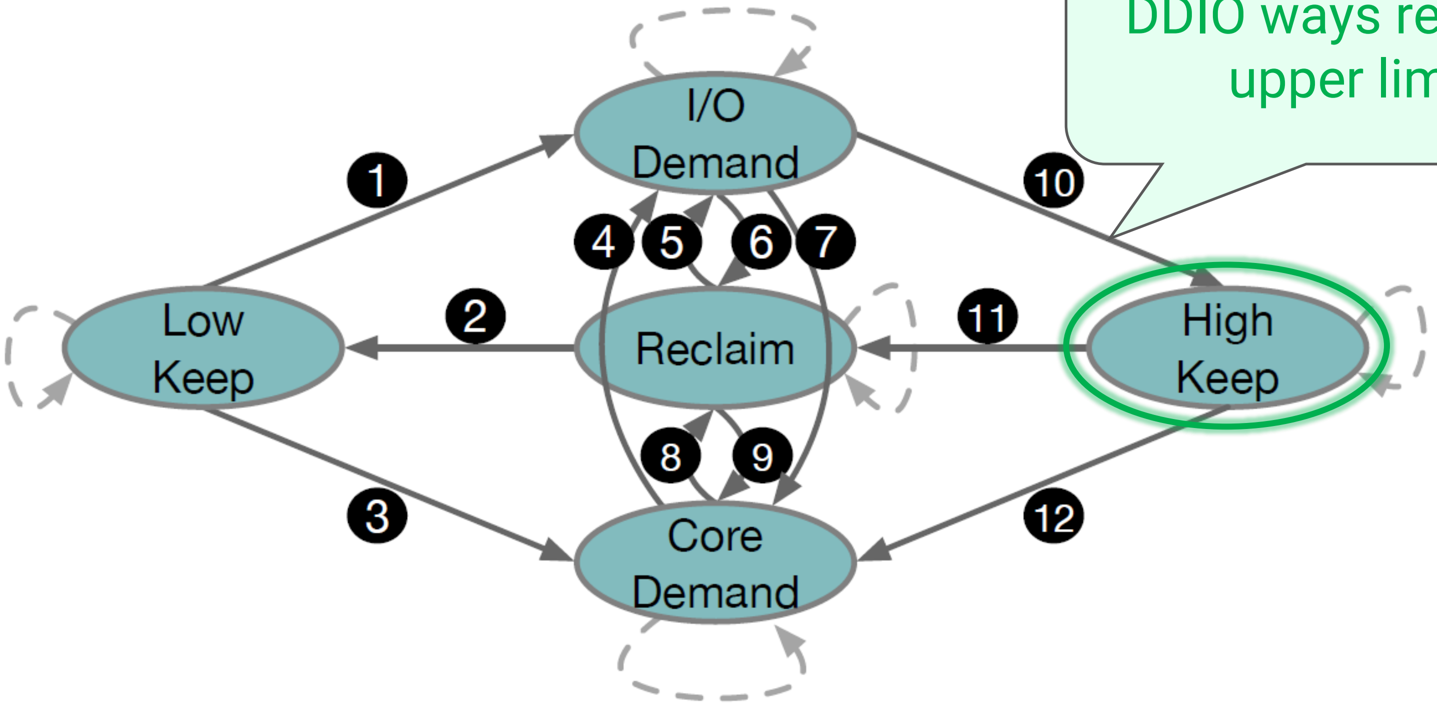
More DDIO hit and miss (intense IO)

IAT: State Transition



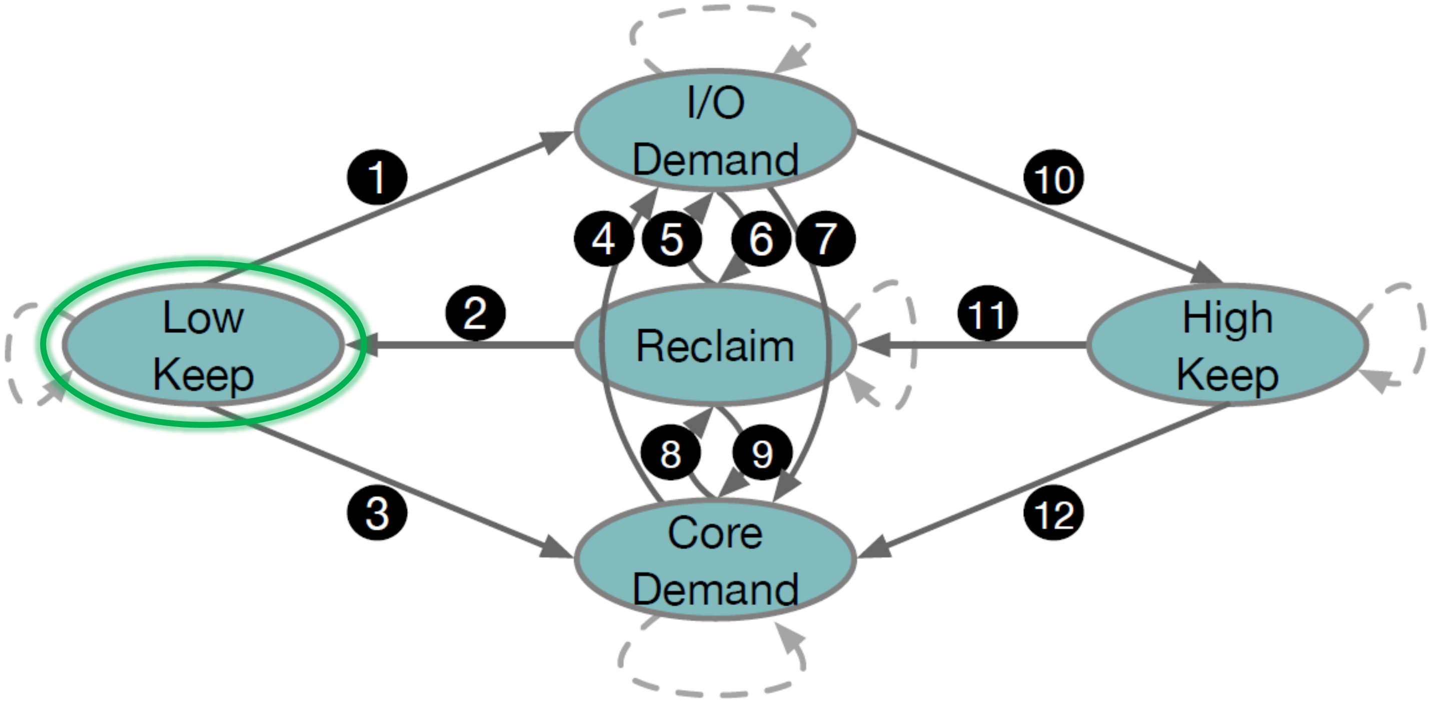
Keep on Increasing DDIO ways

IAT: State Transition

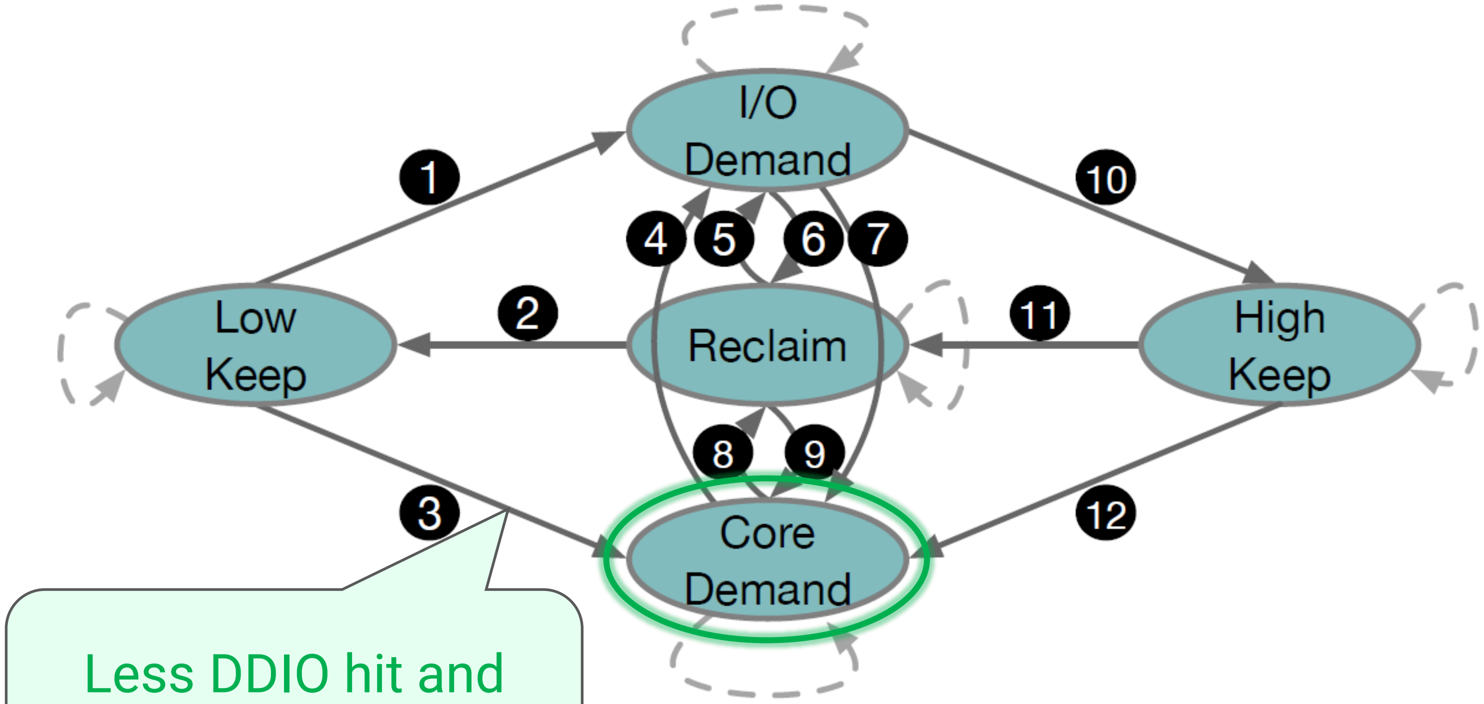


DDIO ways reaches upper limit

IAT: State Transition

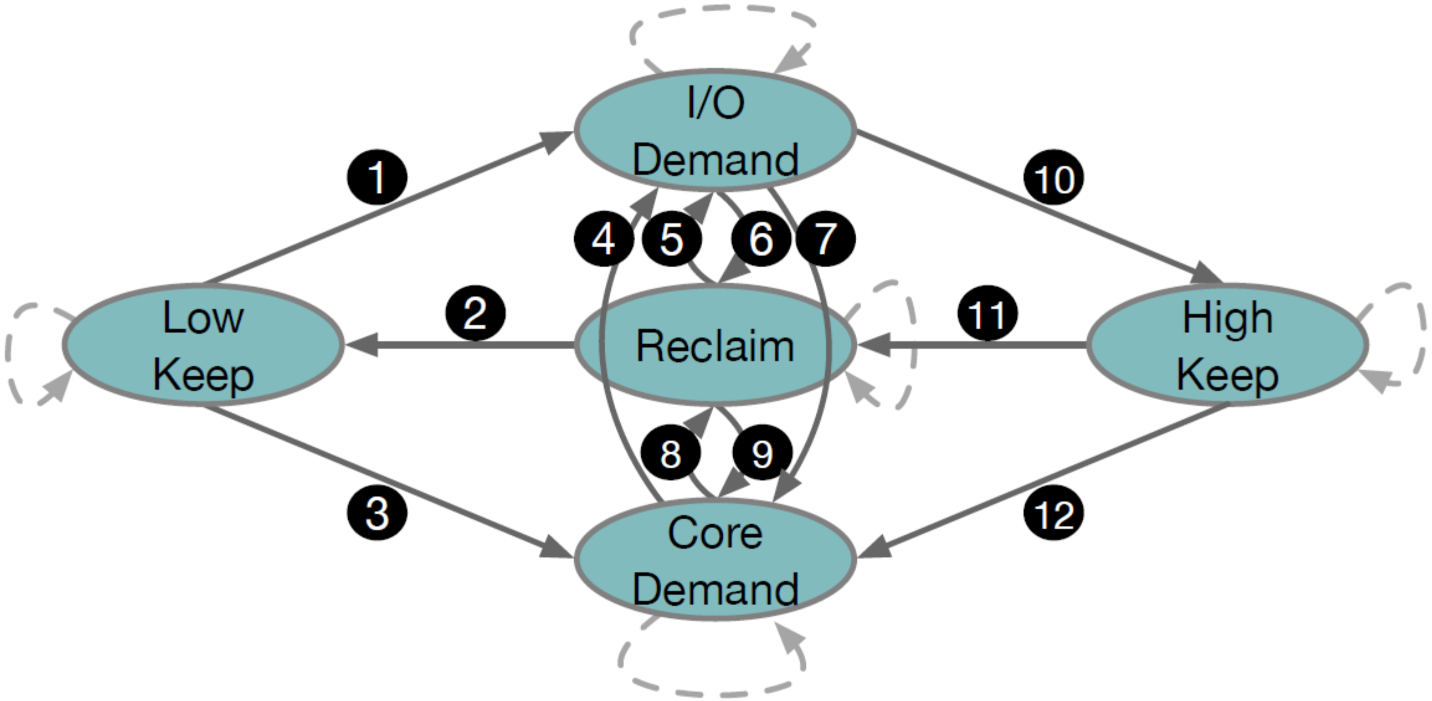


IAT: State Transition

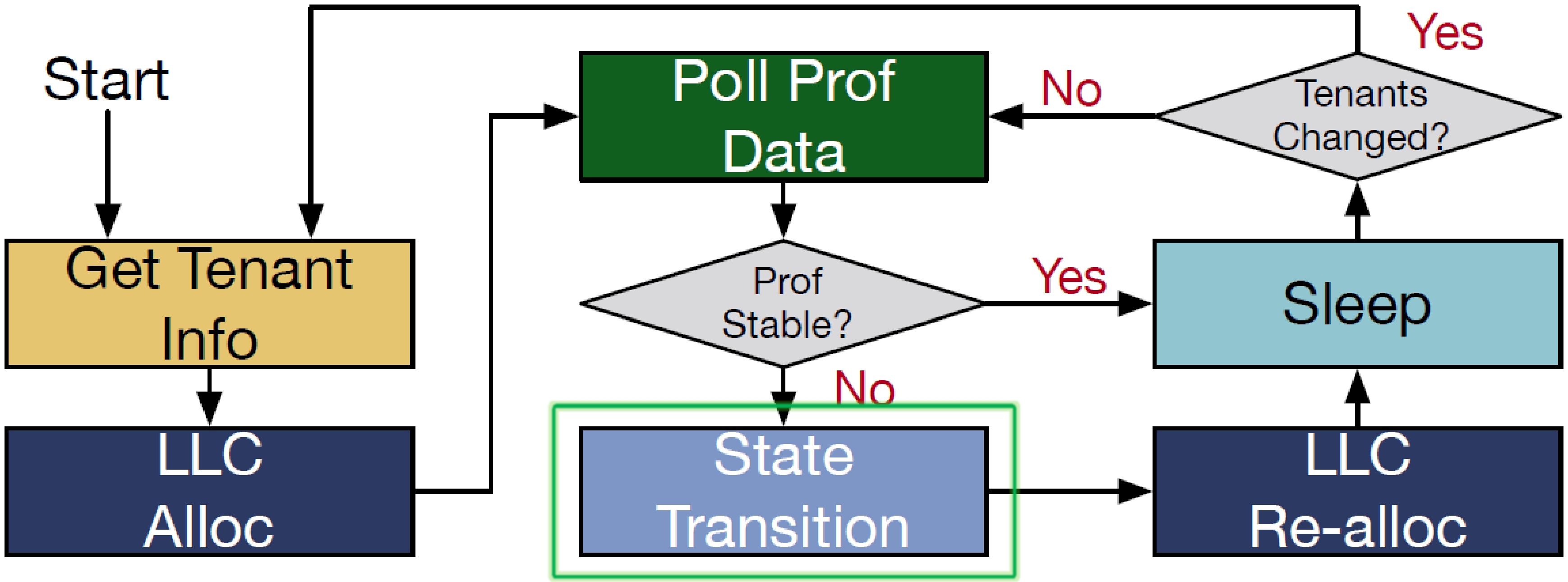


Less DDIO hit and more LLC ref

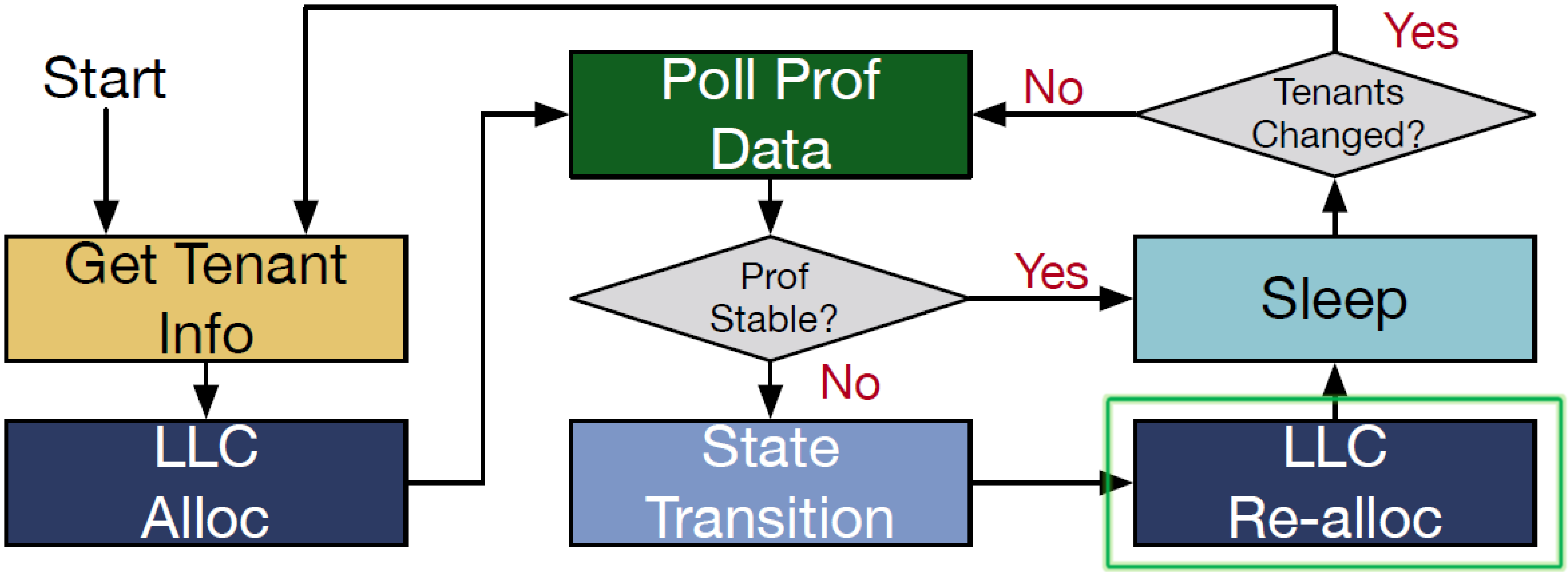
IAT: State Transition



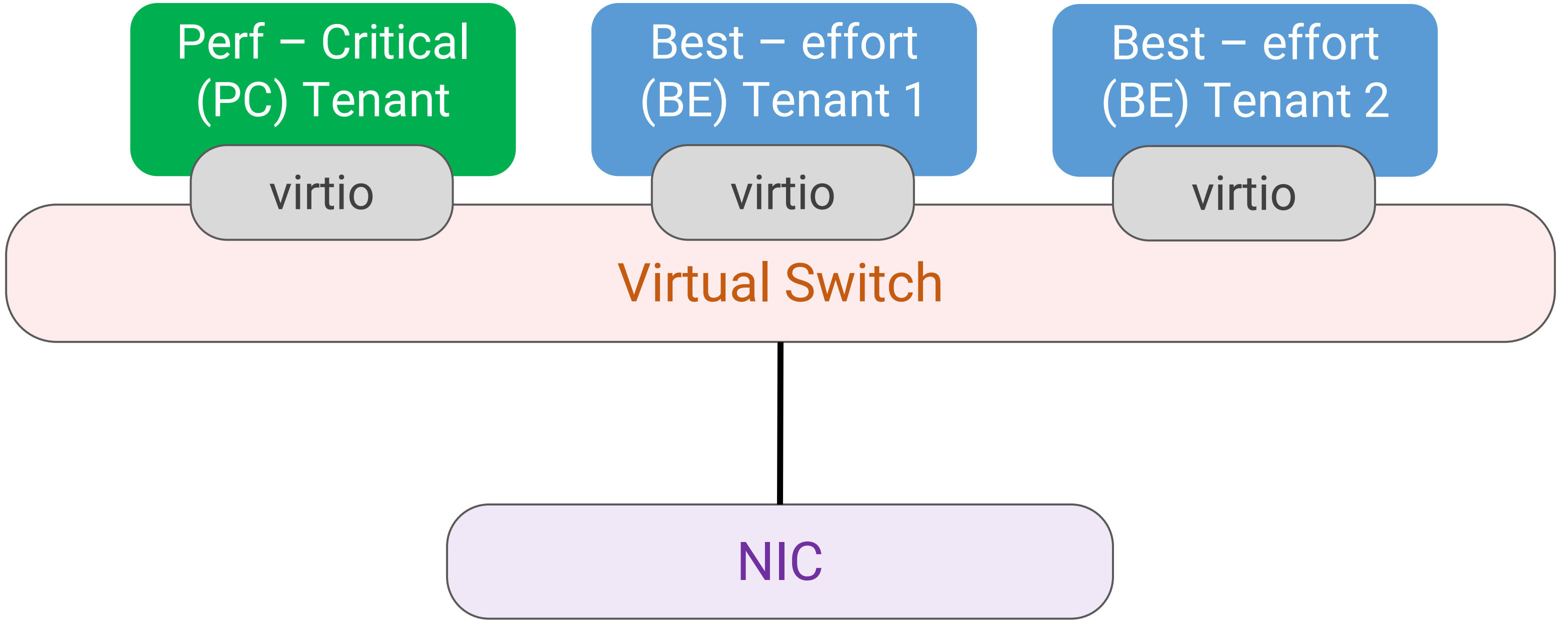
IAT: LLC Re – alloc



IAT: LLC Re – alloc

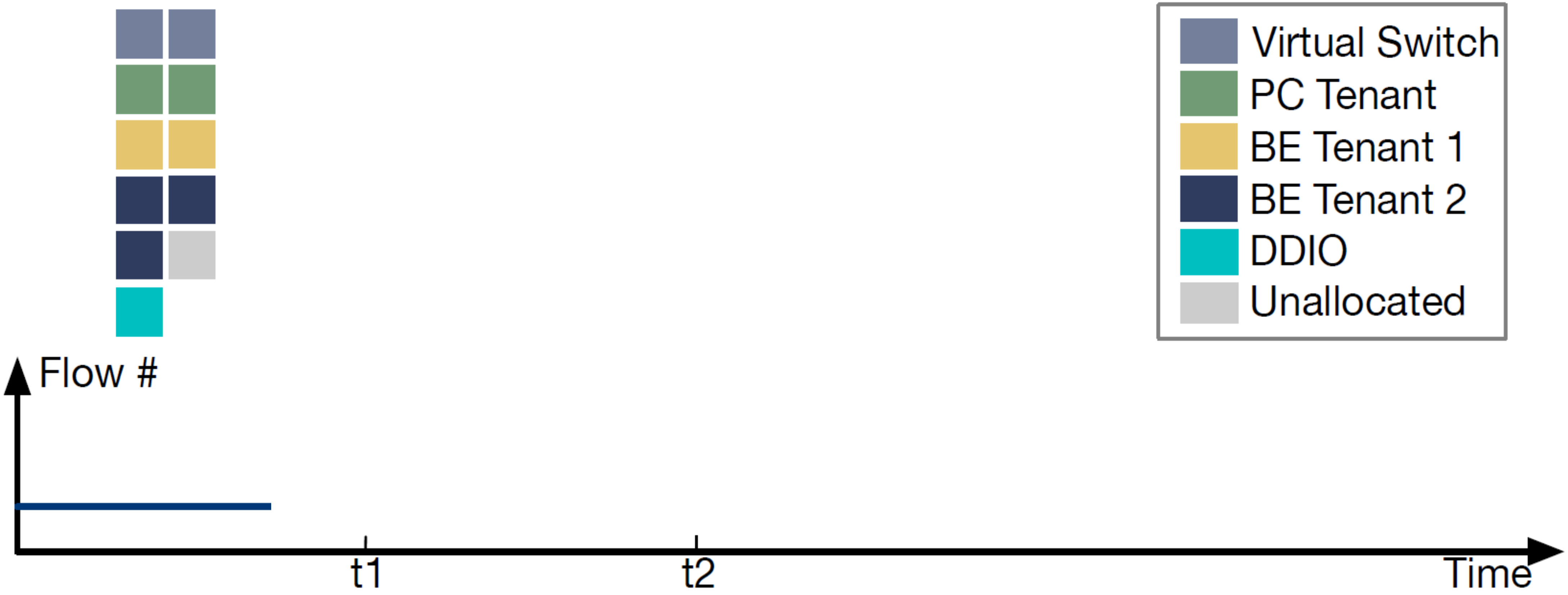


IAT: LLC Re – alloc (Example 1)

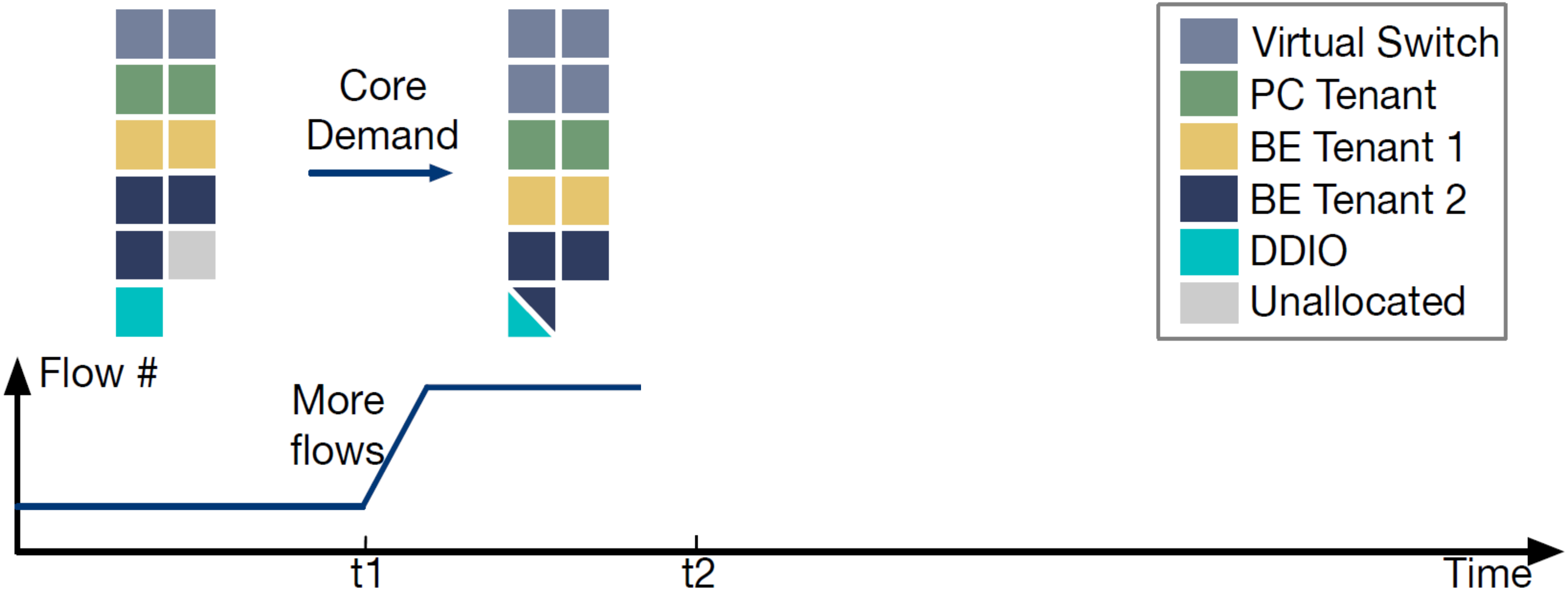

Perf – Critical (PC) Tenant
Best – effort (BE) Tenant 1
Best – effort (BE) Tenant 2
virtio
virtio
virtio
Virtual Switch
NIC

Example 1: Aggregation Model

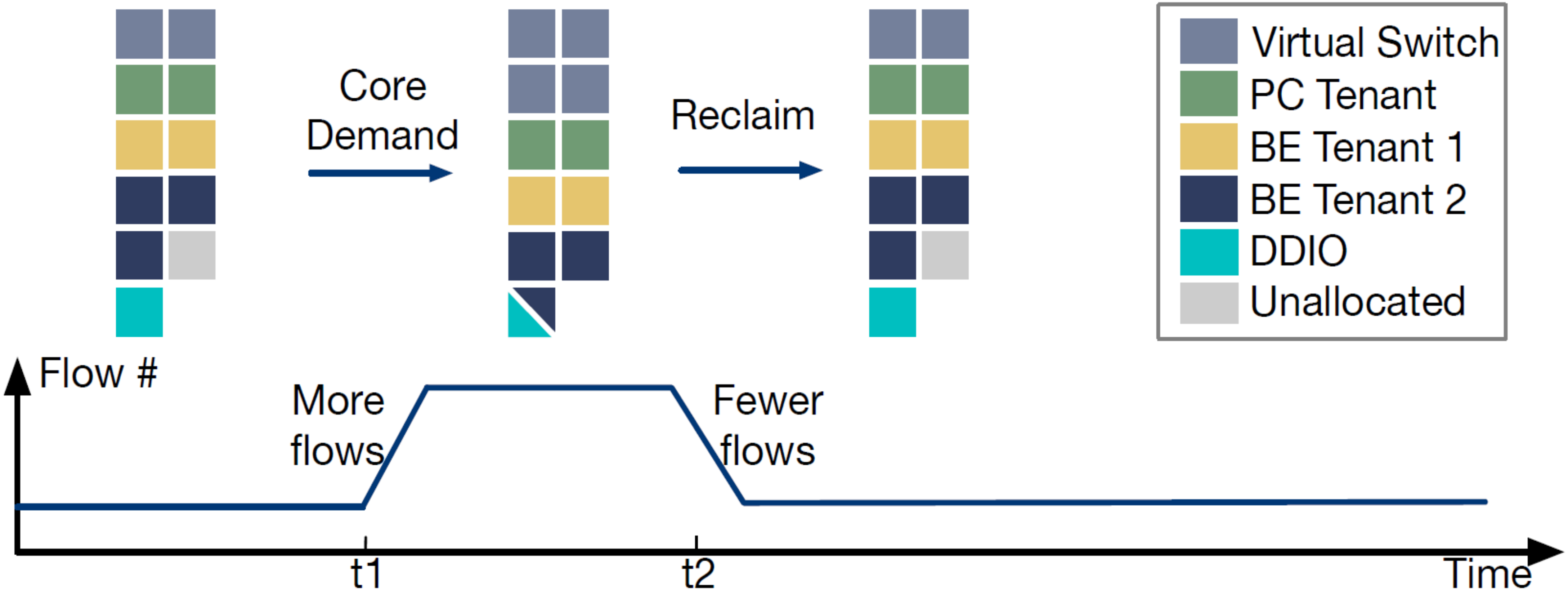
IAT: LLC Re – alloc (Example 1)



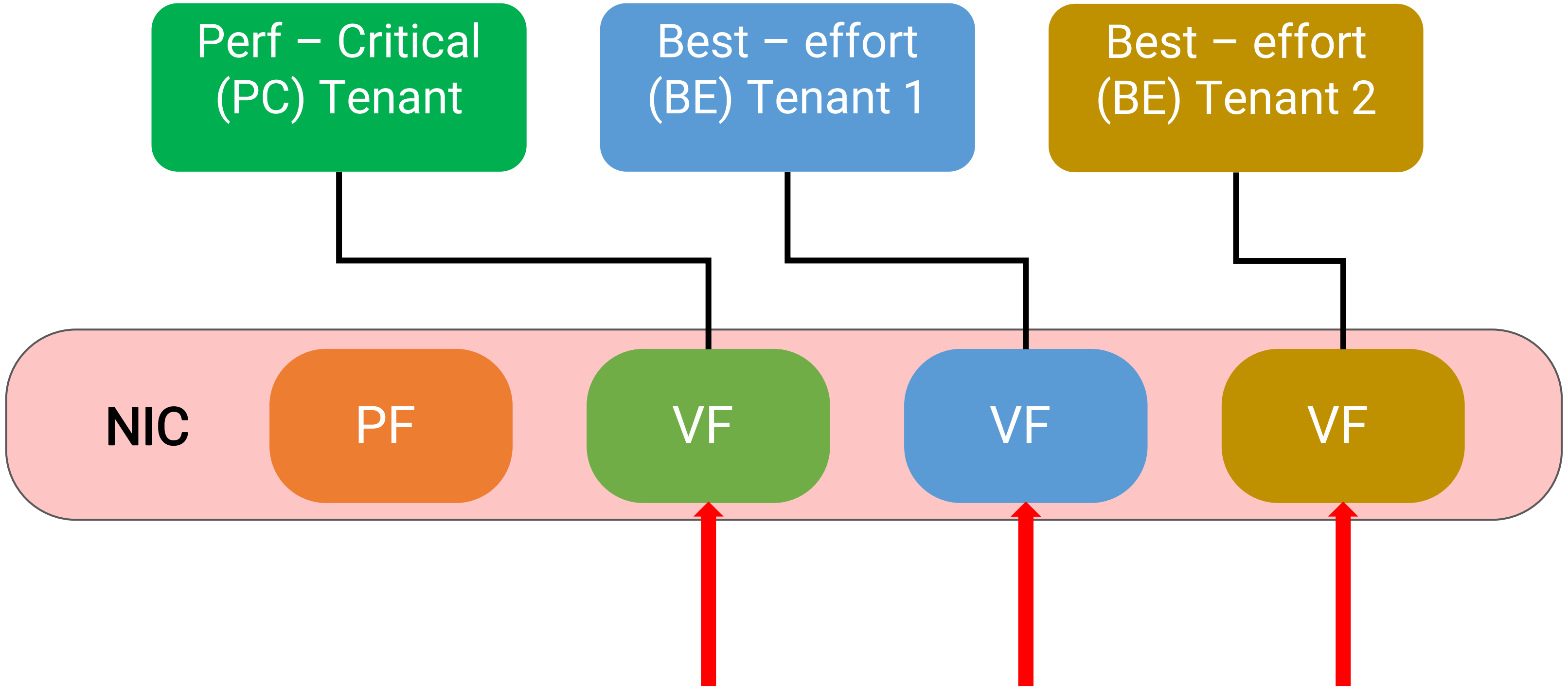
IAT: LLC Re – alloc (Example 1)



IAT: LLC Re – alloc (Example 1)

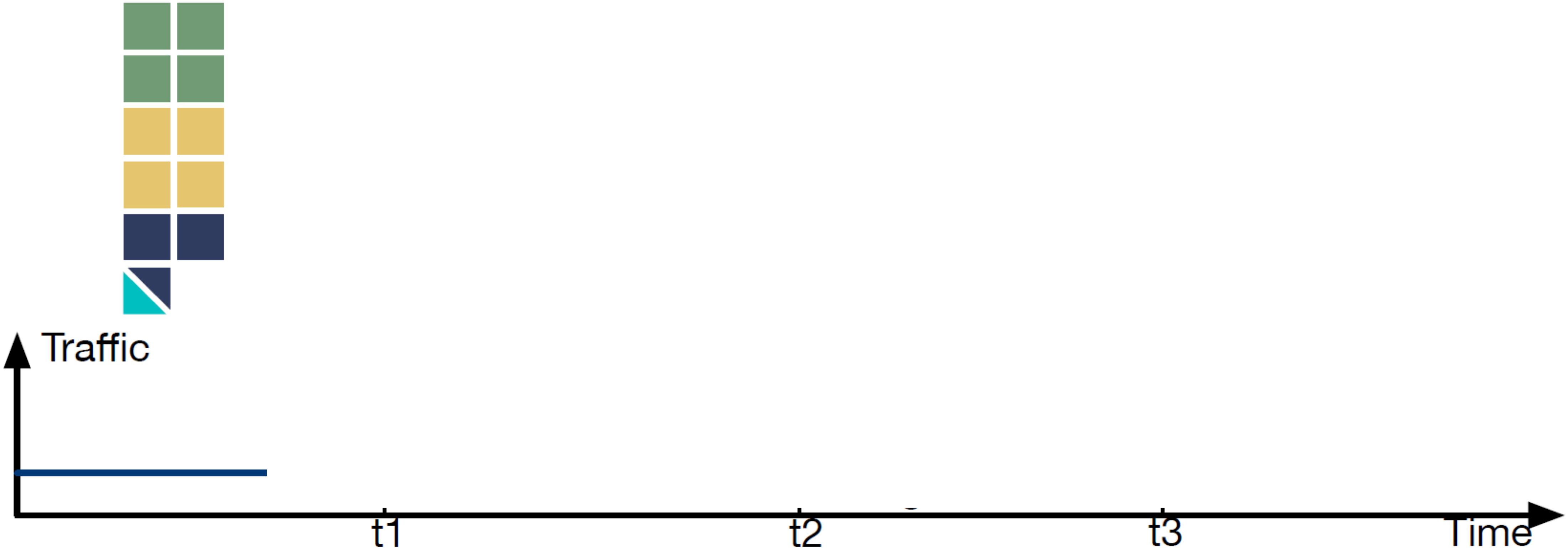


IAT: LLC Re – alloc (Example 2)

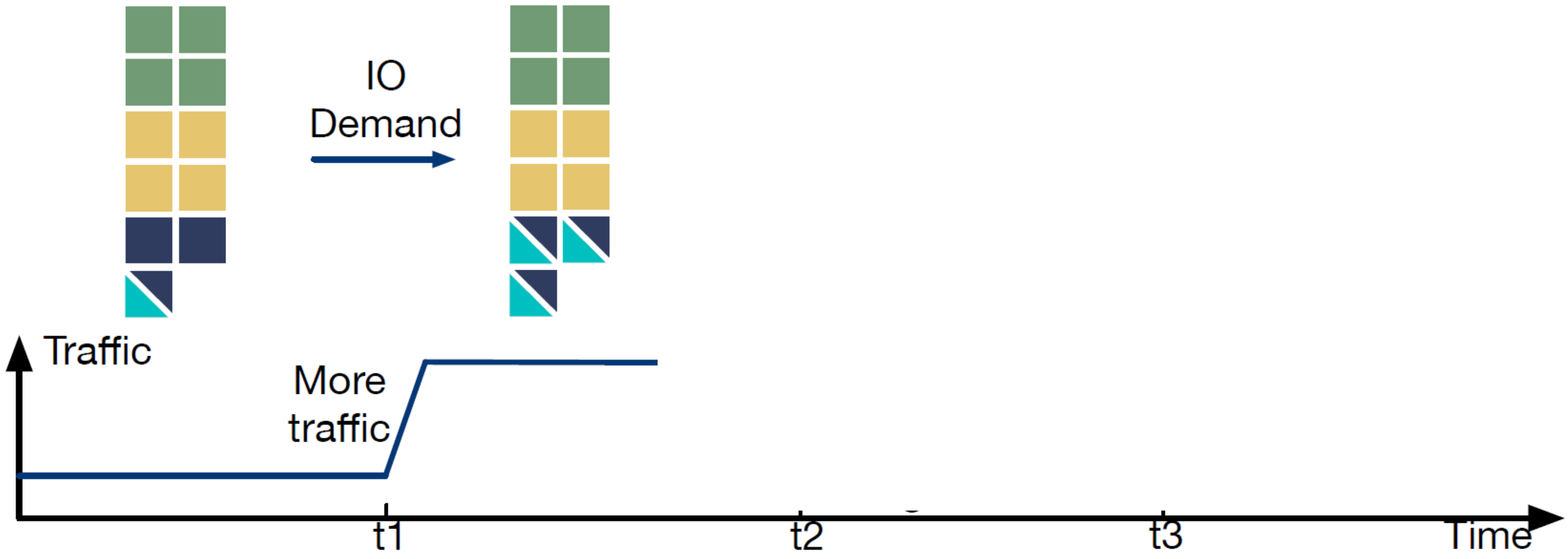

Perf – Critical (PC) Tenant
Best – effort (BE) Tenant 1
Best – effort (BE) Tenant 2
NIC
PF
VF
VF
VF

Example 2: Slicing Model

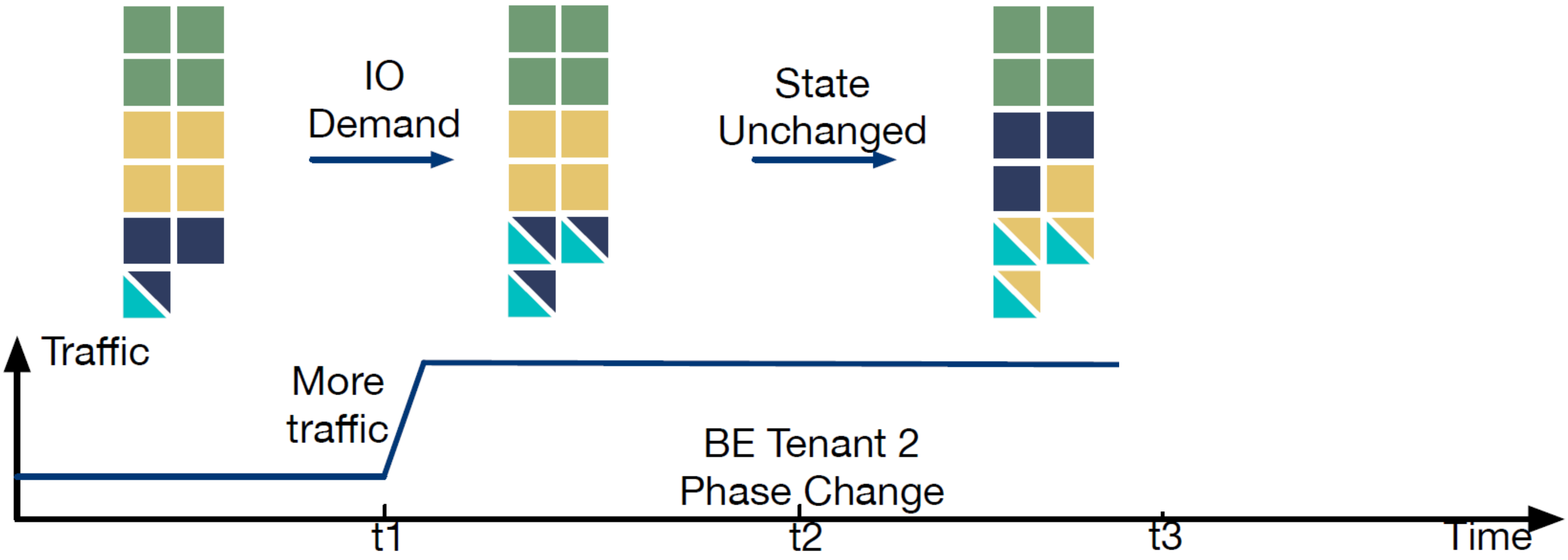
IAT: LLC Re – alloc (Example 2)



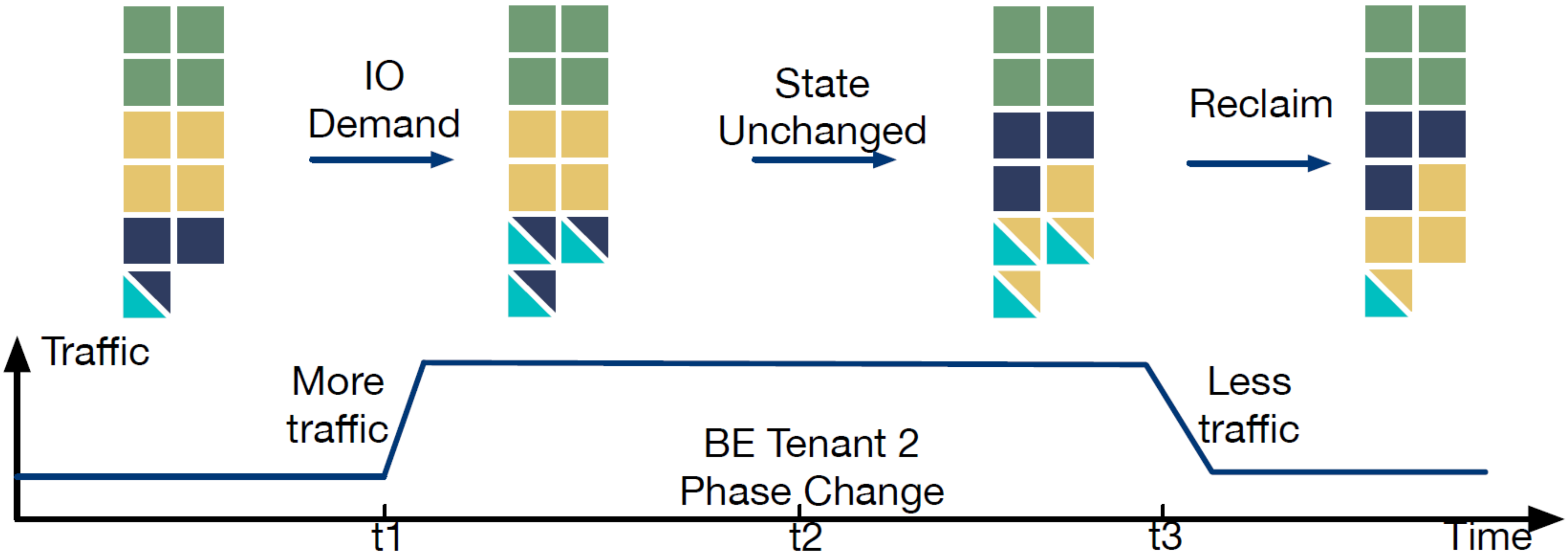
IAT: LLC Re – alloc (Example 2)



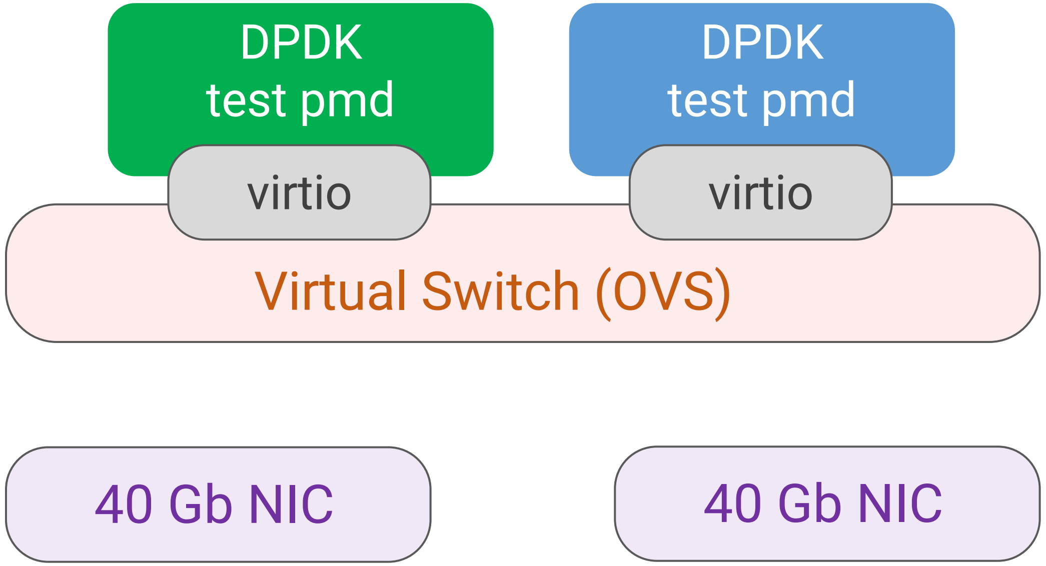
IAT: LLC Re – alloc (Example 2)



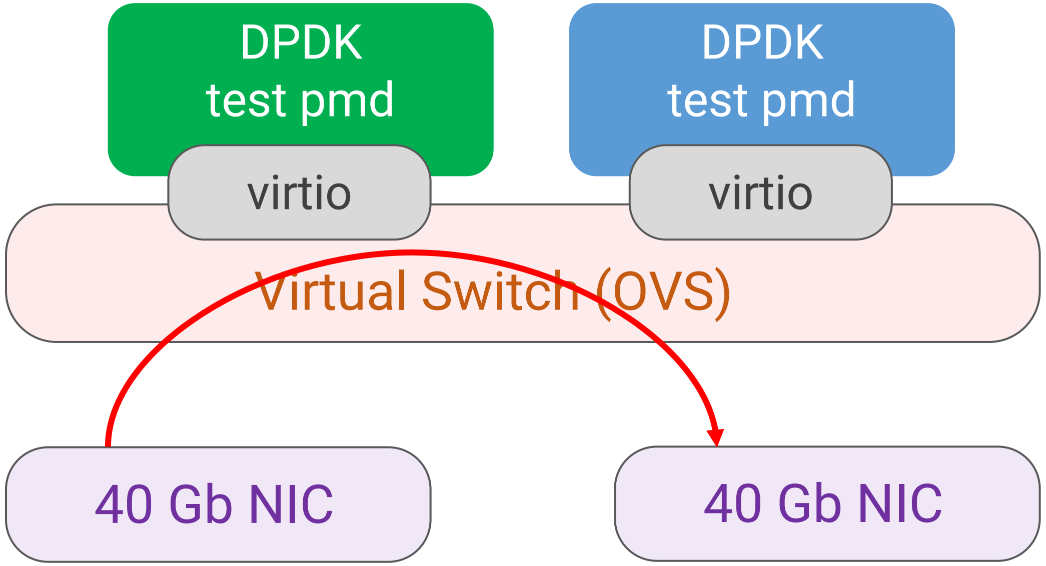
IAT: LLC Re – alloc (Example 2)



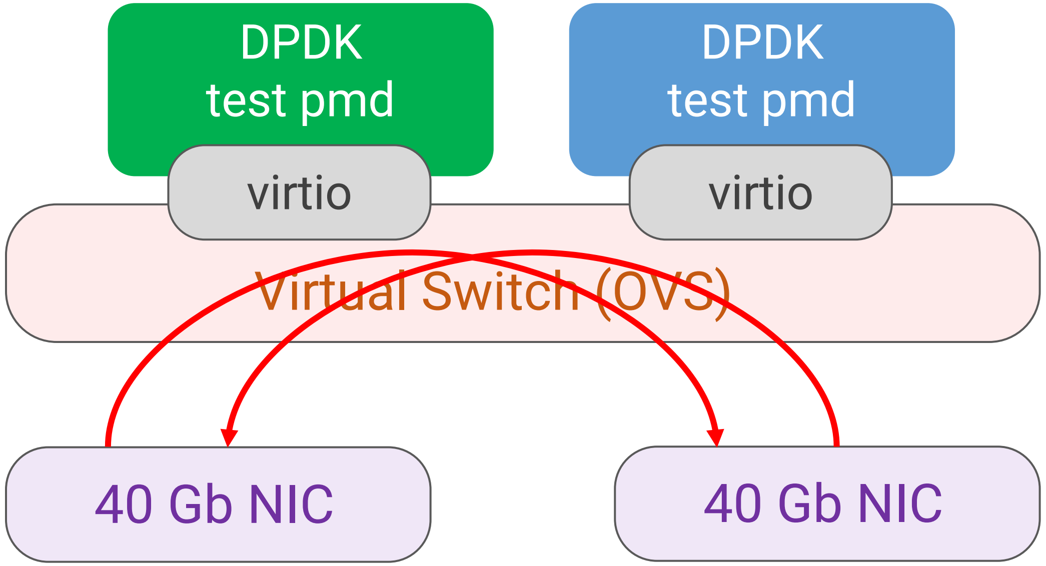
IAT: Solving Leaky DMA Problem



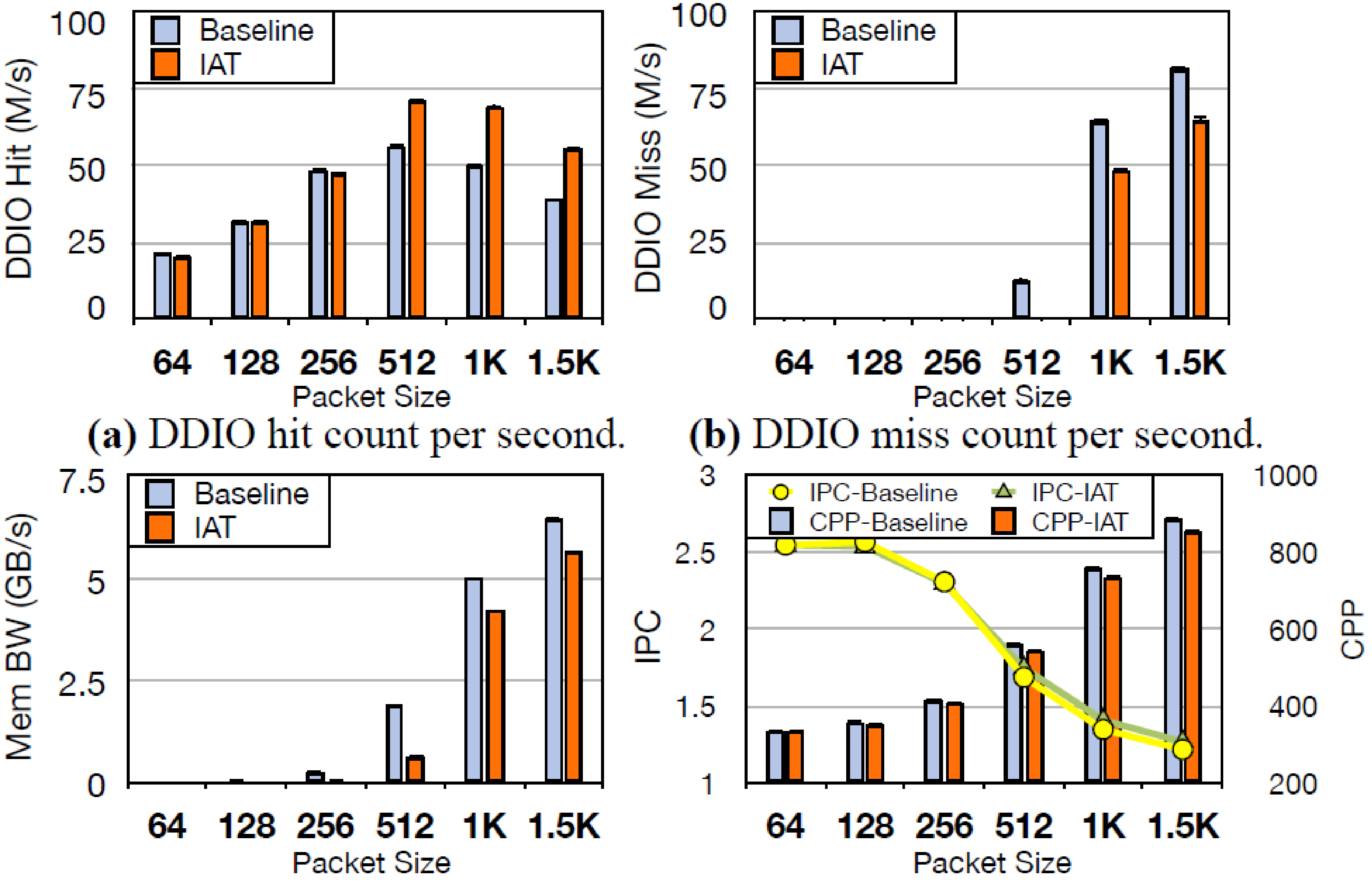
IAT: Solving Leaky DMA Problem



IAT: Solving Leaky DMA Problem



IAT: Solving Leaky DMA Problem



(a) DDIO hit count per second.

(b) DDIO miss count per second.

(c) Memory bandwidth consumption.

(d) OVS IPC and CPP.

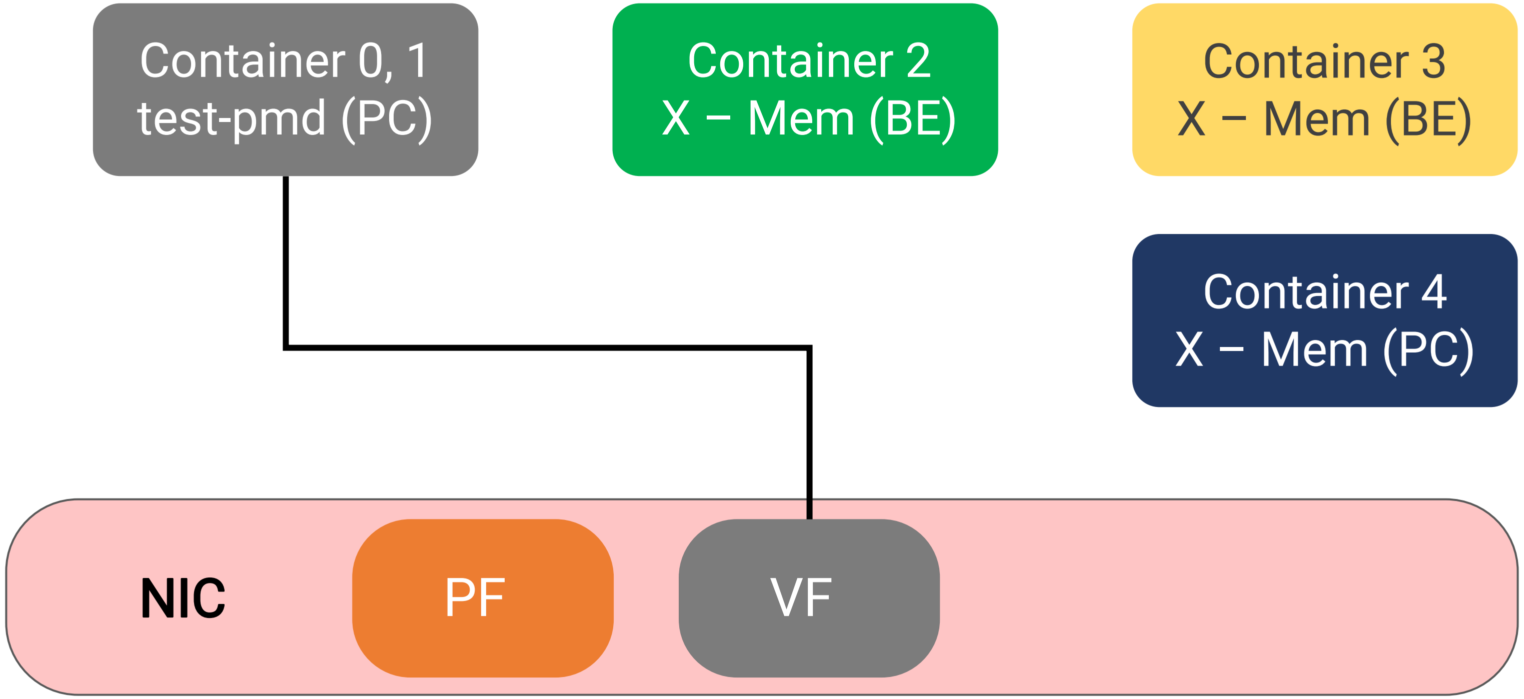
IAT: Solving Latent Contender Problem

IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval

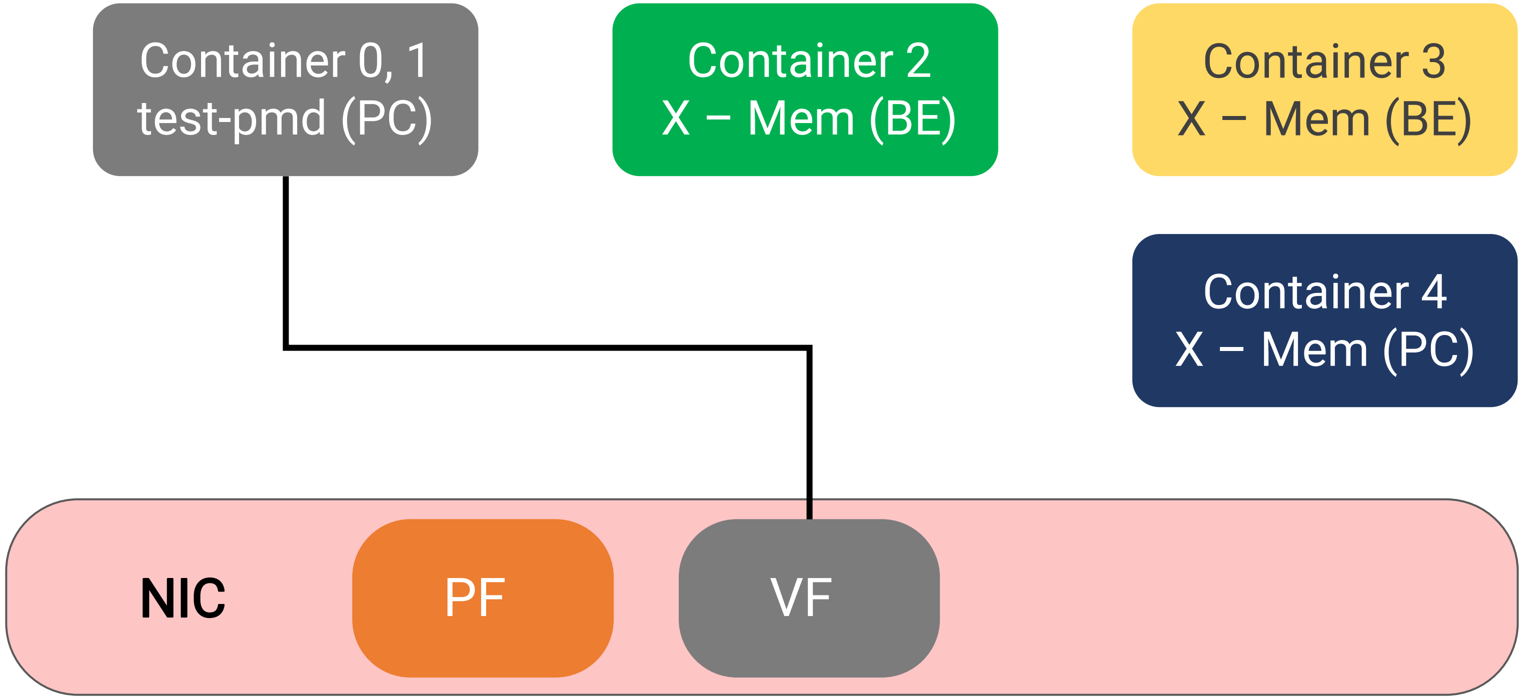
IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval



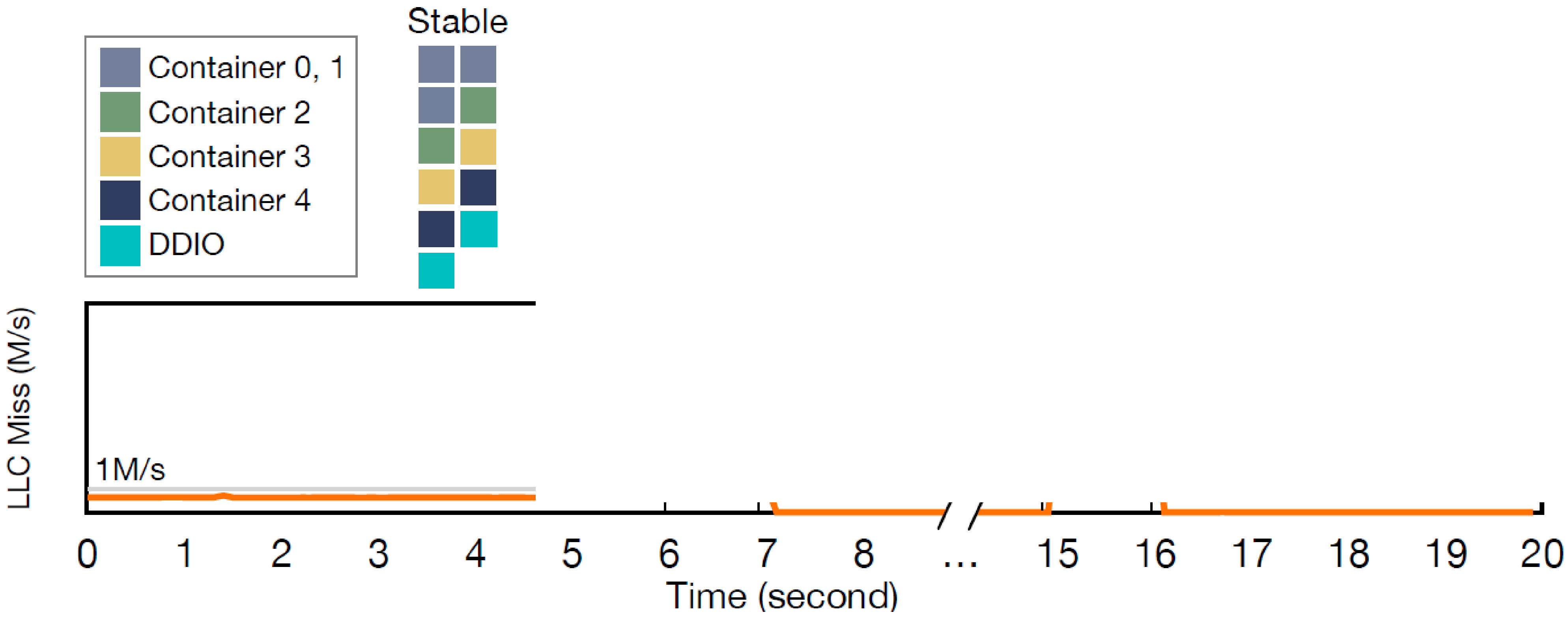
IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



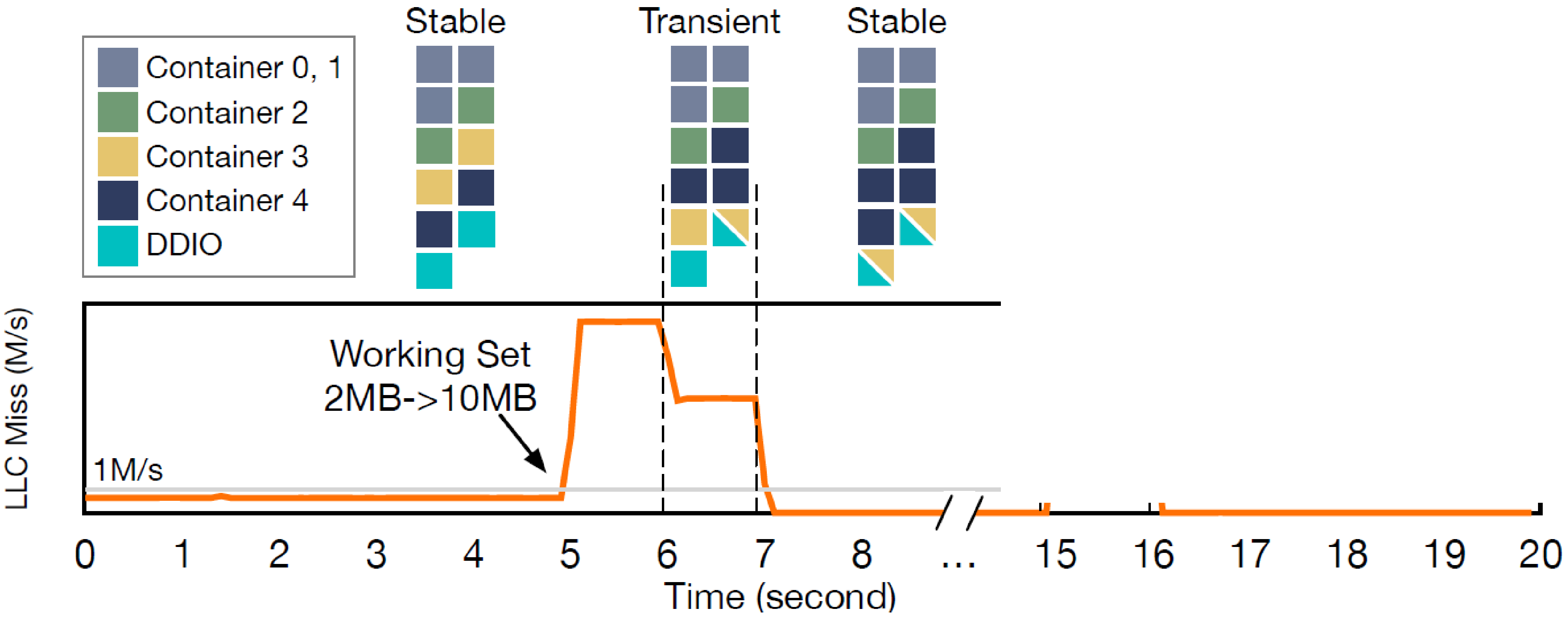
IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



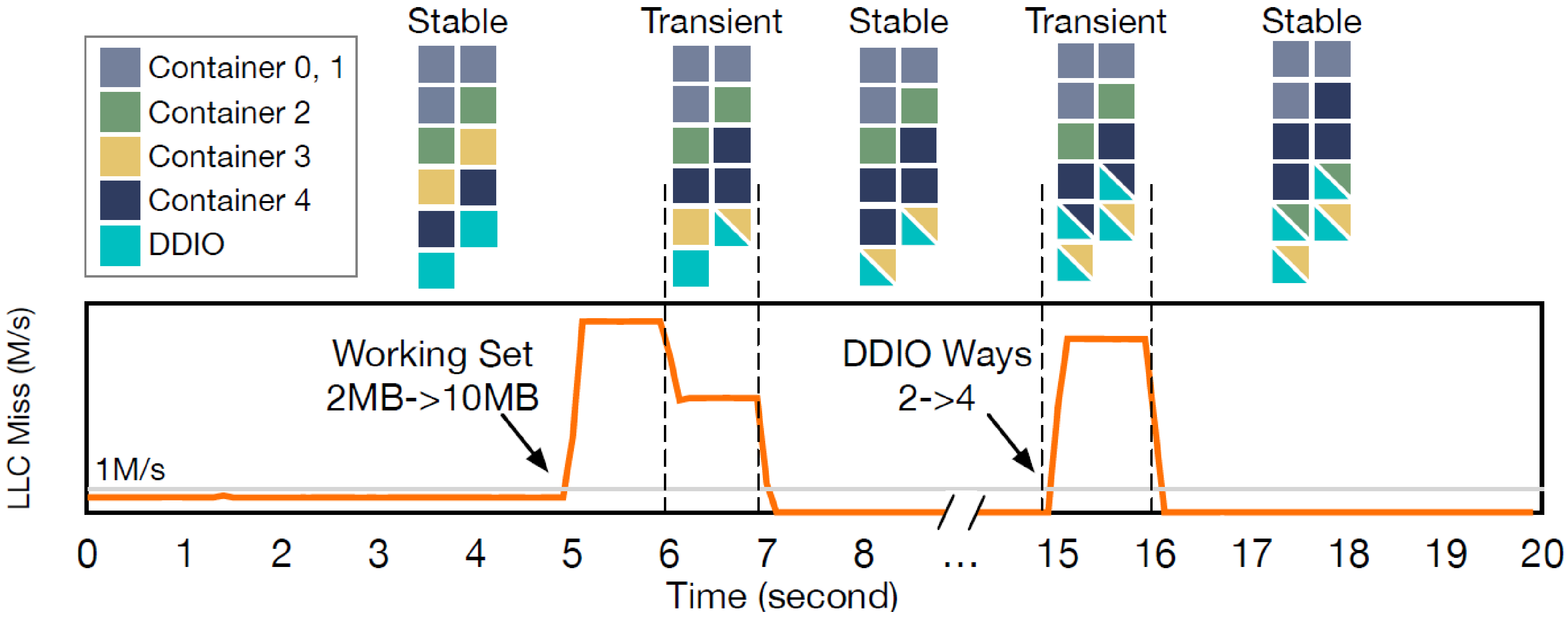
IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



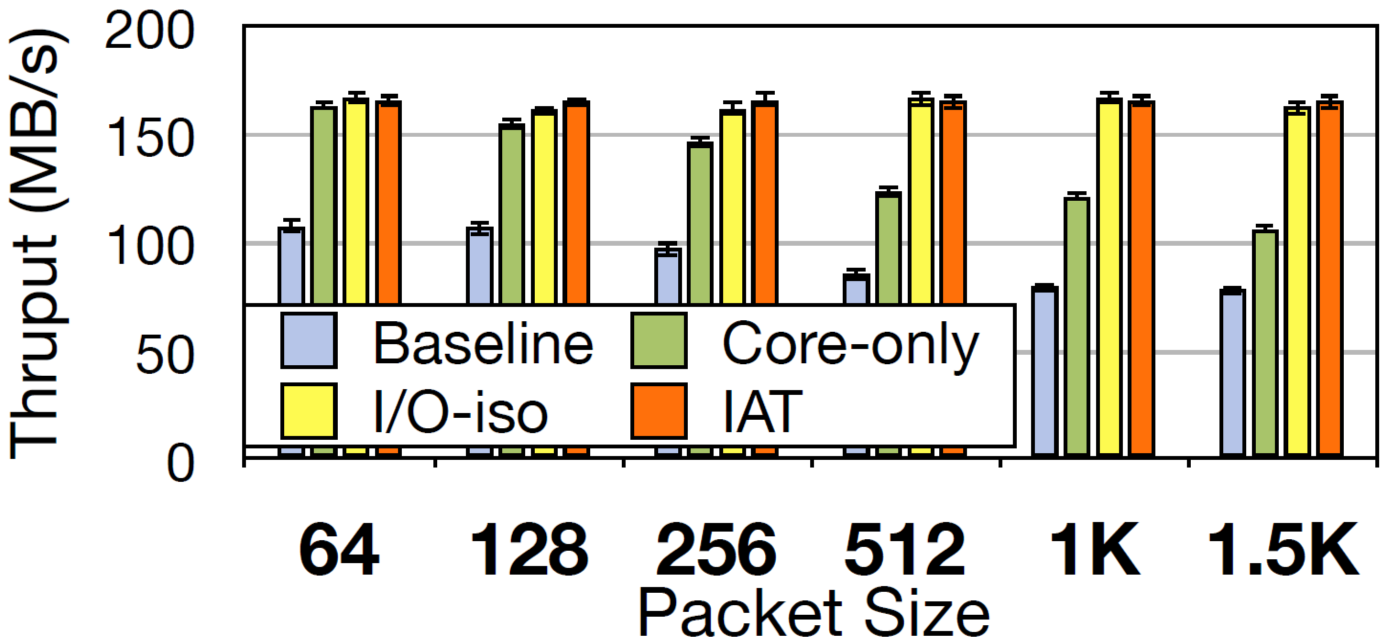
IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



IAT: Solving Latent Contender Problem

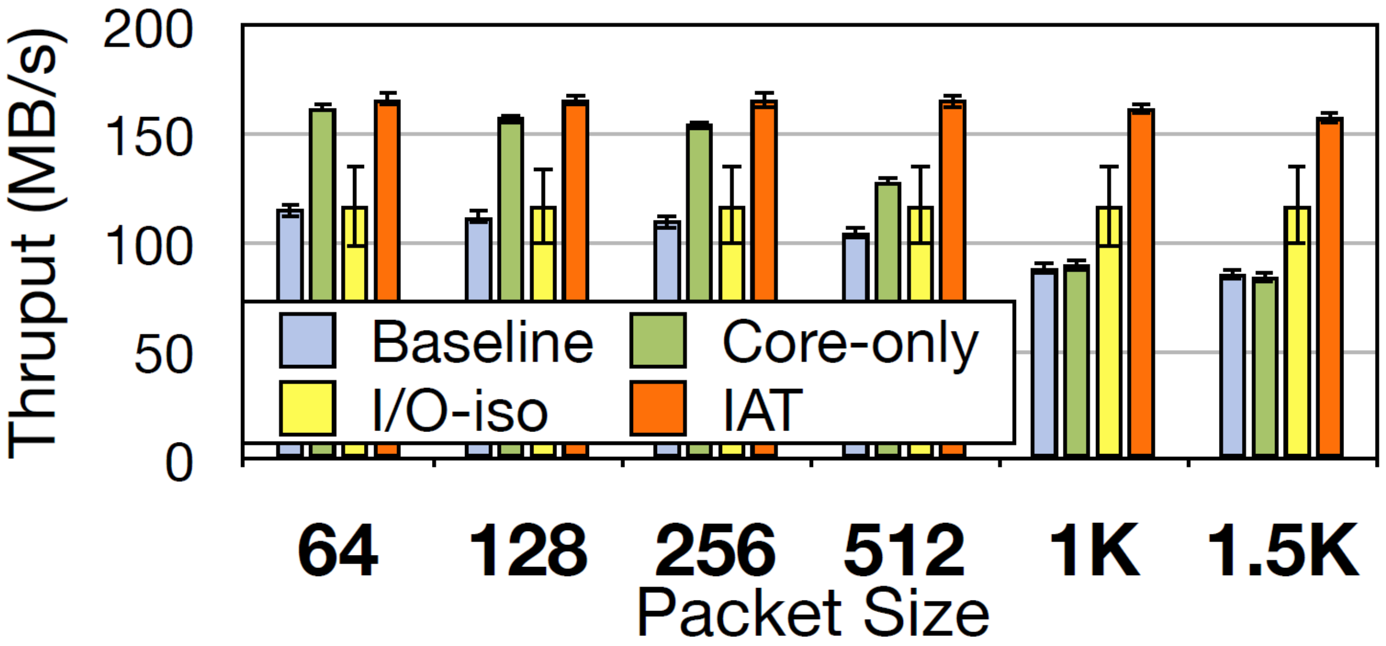
- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



(a) Throughput between 5s and 15s.

IAT: Solving Latent Contender Problem

- Change in X-Mem working set size and DDIO ways at different time interval: **Container 4**



(c) Throughput after 15s.

IAT: Overhead

IAT: Overhead

○ Implemented as user – space daemon in Ubuntu

IAT: Overhead

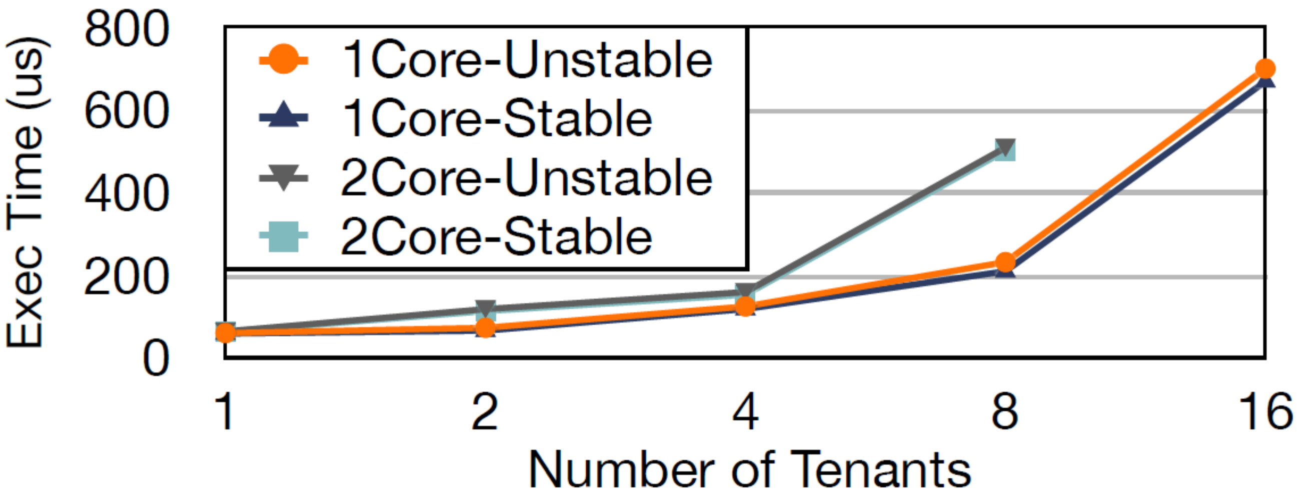
- Implemented as user – space daemon in Ubuntu

- IAT execution time is measured in following cases:
  - One vs two cores per tenant
  - Stable: no need to re – allocate LLC (only `Poll Prof Data` time)
  - Unstable: need to take action (`PPD + ST + LLC Re – alloc` time)

IAT: Overhead

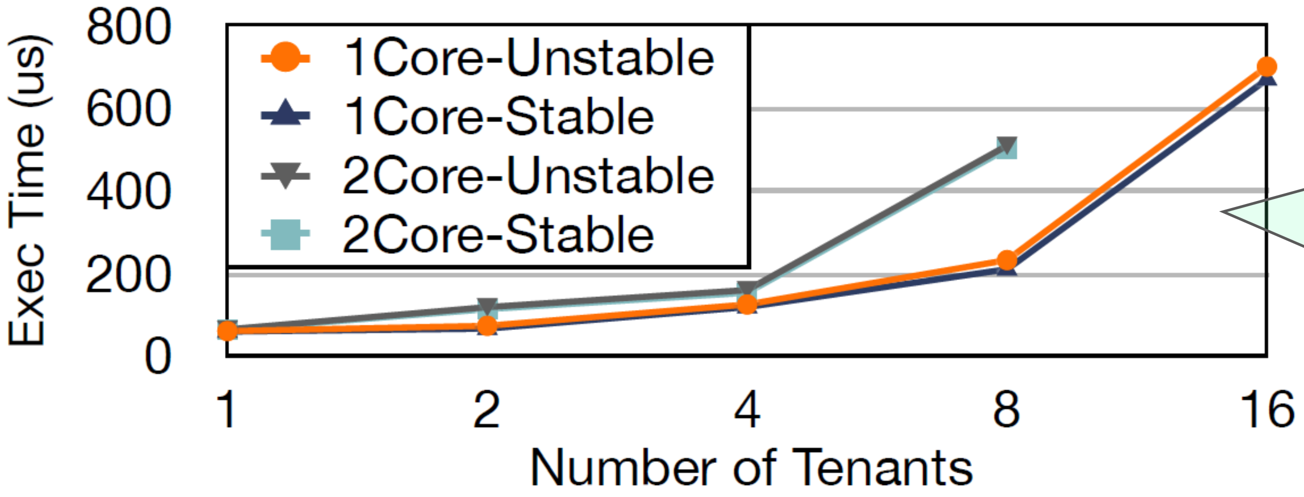
- Implemented as user – space daemon in Ubuntu

- IAT execution time is measured in following cases:
  - One vs two cores per tenant
  - Stable: no need to re – allocate LLC (only `Poll Prof Data` time)
  - Unstable: need to take action (`PPD + ST + LLC Re – alloc` time)



IAT: Overhead

- Implemented as user – space daemon in Ubuntu
- IAT execution time is measured in following cases:
  - One vs two cores per tenant
  - Stable: no need to re – allocate LLC (only `Poll Prof Data` time)
  - Unstable: need to take action (`PPD + ST + LLC Re – alloc` time)



Overhead is **Negligible**

Conclusion

- DDIO introduces new challenges for better LLC management (IO aware)

Conclusion

- DDIO introduces new challenges for better LLC management (IO aware)

- Both the problems, Leaky DMA and Latent Contender, are solved by IAT

Conclusion

- DDIO introduces new challenges for better LLC management (IO aware)
- Both the problems, Leaky DMA and Latent Contender, are solved by IAT
- IAT reduces interference caused by DDIO with negligible overhead

Questions ?

Thank You