

## 1 What is DDIO?

Data Direct I/O Technology (DDIO) transfers packets directly to Last Level Cache (LLC) rather than main memory. DDIO updates a cache line if it is already available in LLC; otherwise, it allocates the cache line in a limited portion of LLC (i.e., **2 ways** in a n-way set-associative cache).

DDIO was introduced to improve the performance of I/O applications by mitigating expensive DRAM accesses.

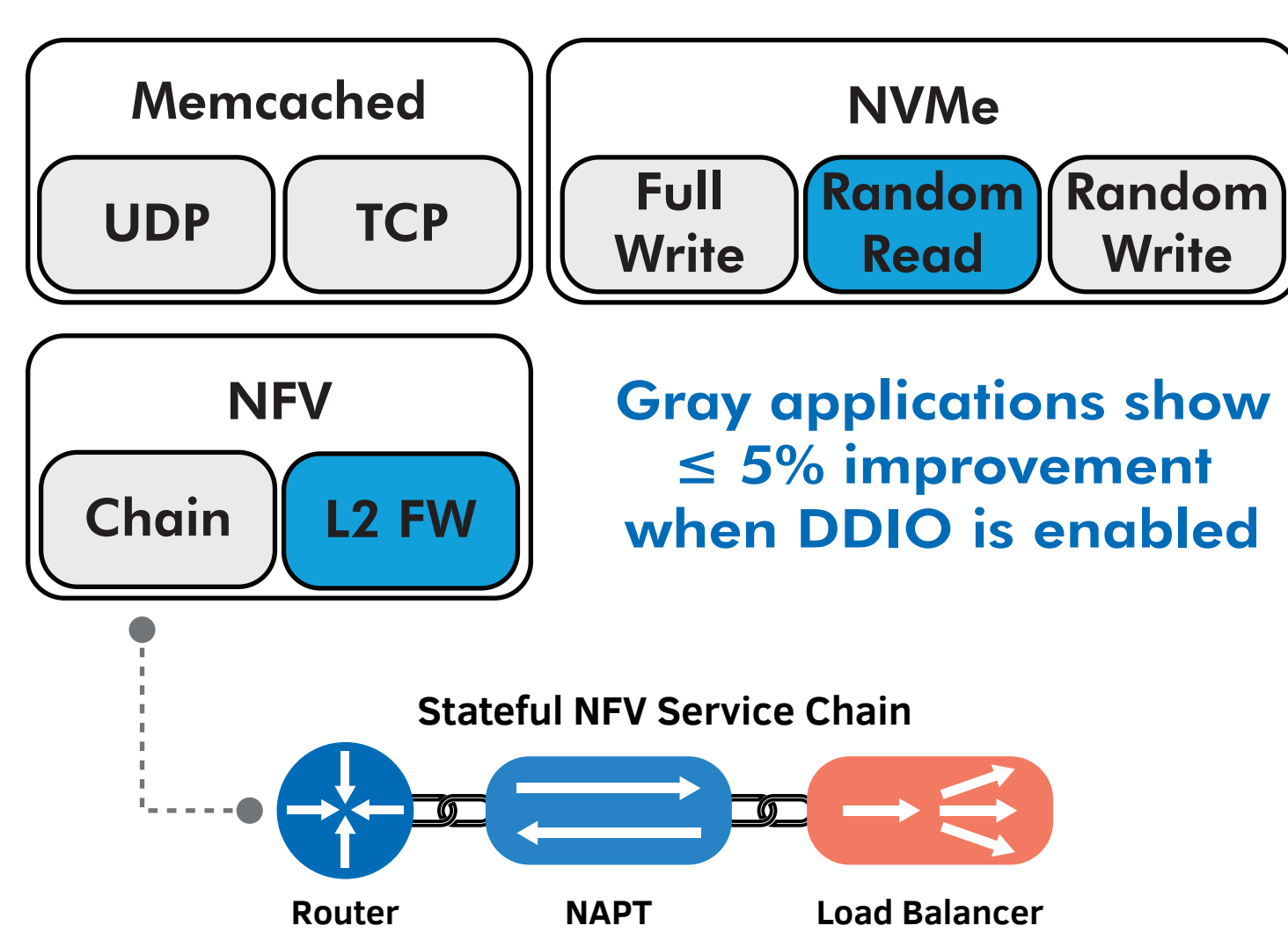
## 2 DDIO Can Become a Bottleneck

Faster link speeds causes DDIO *fail* to provide the expected benefits, as new incoming packets can repeatedly evict previously received packets (i.e., **not-yet-processed** and **already-processed** packets) from the LLC. The probability of eviction is high when:

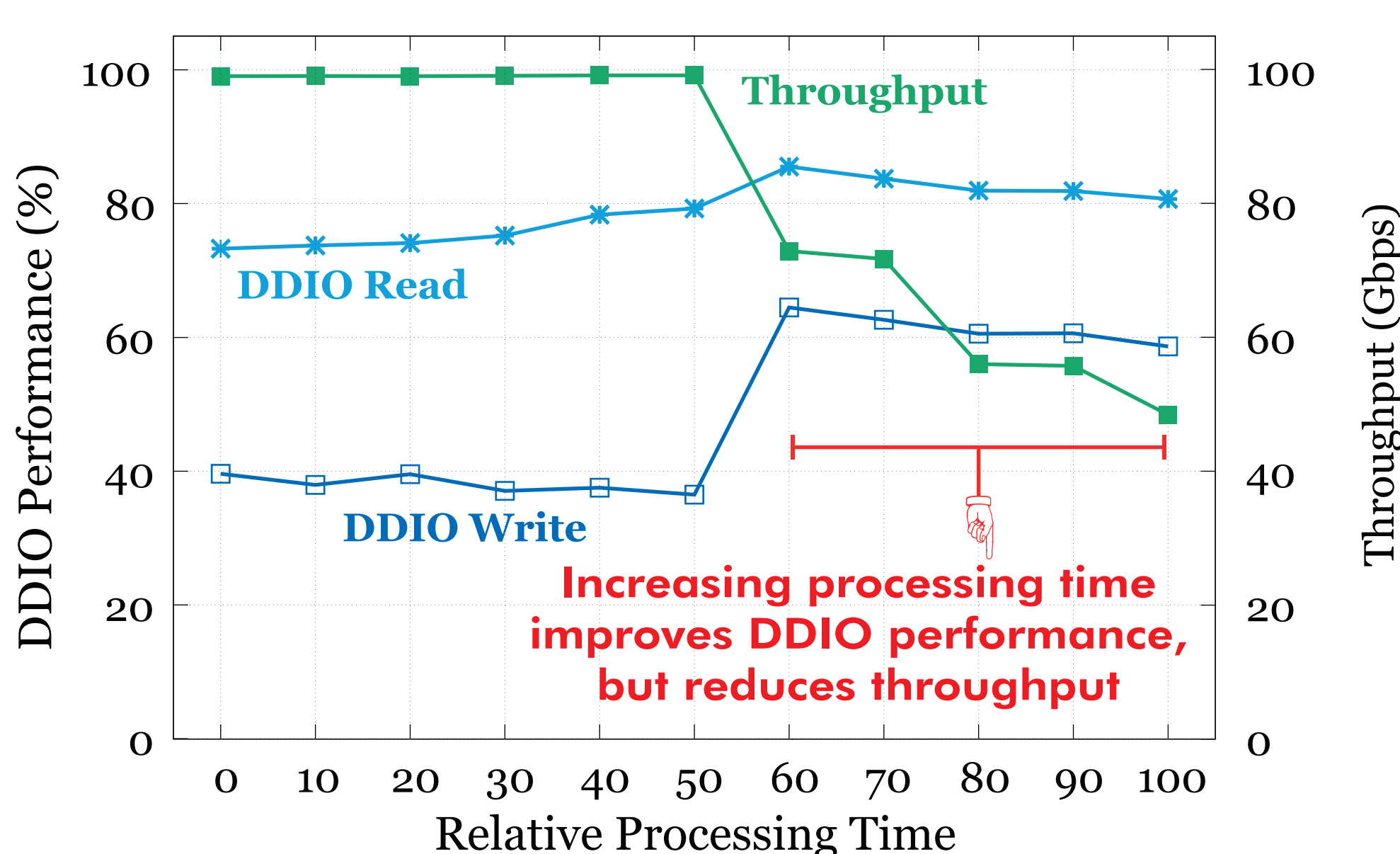
- High #Receive (RX) descriptors
- High load imbalance factor
- Receiving rate  $\approx 100$  Gbps
- I/O intensive application
- Packet size  $\geq 512$  Byte

## 3 Sensitivity to DDIO

Different applications have different levels of sensitivity to DDIO.

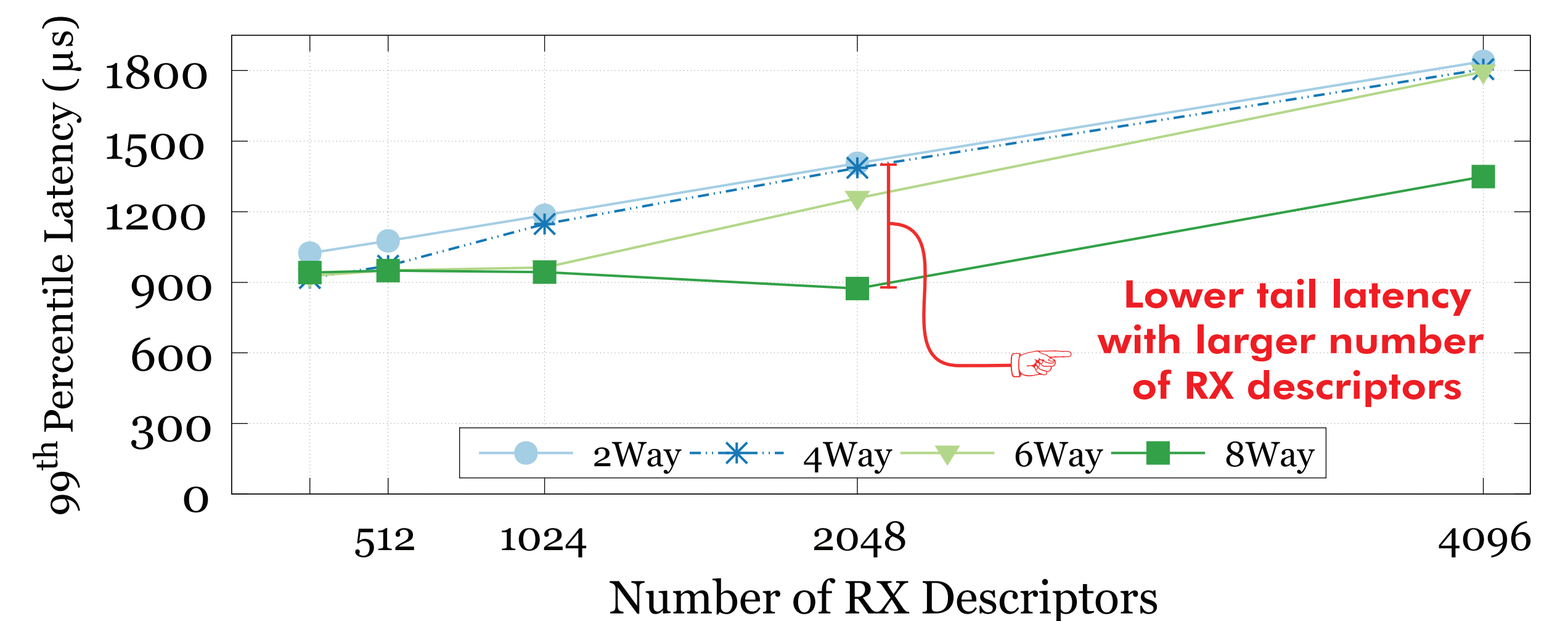


Moreover, performance of DDIO only matters when an application is **I/O bound**, rather than CPU/memory bound.

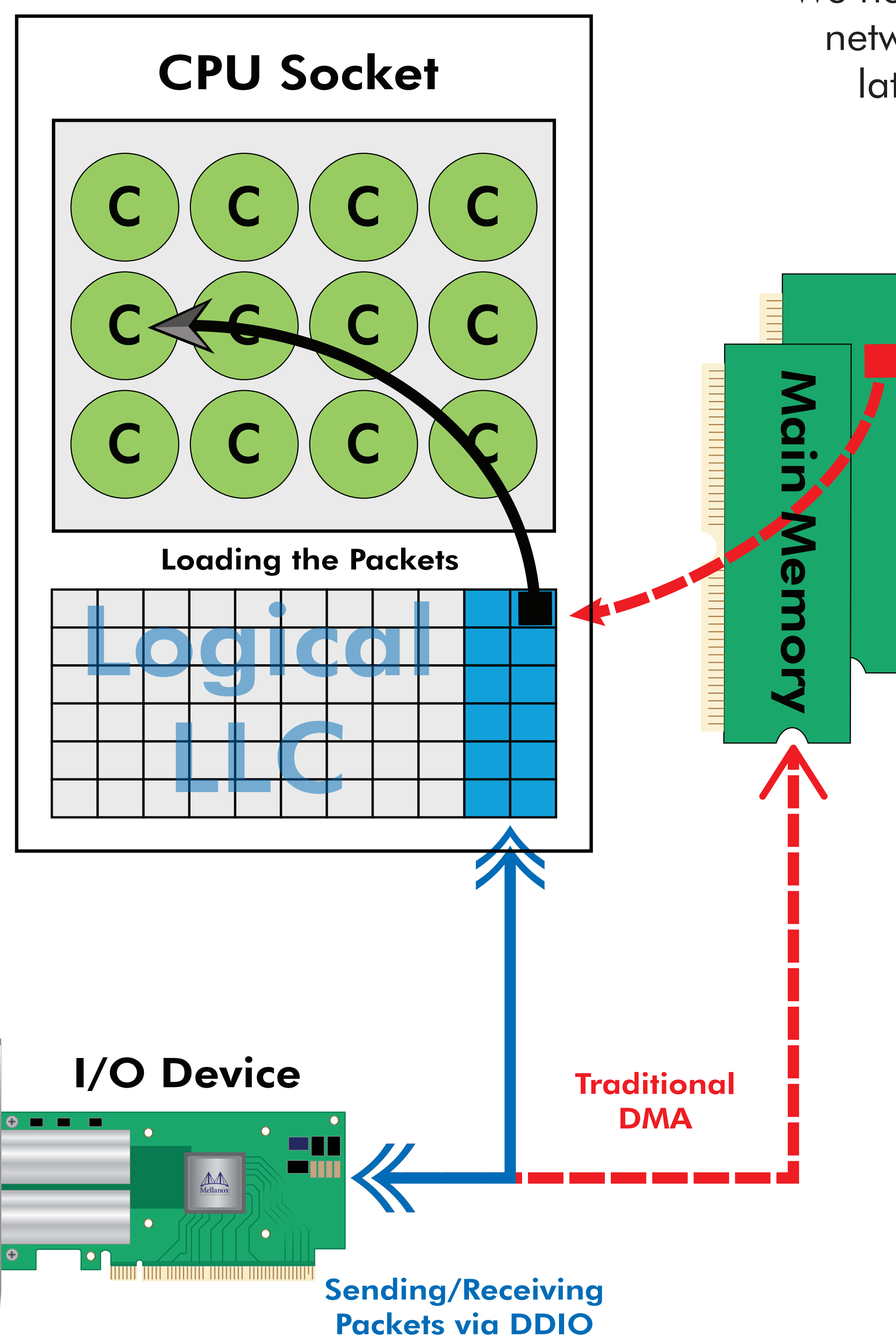


## 4 How to Fine-tune DDIO

A little-discussed register called "**IIO LLC WAYS**" can be used to tune the capacity of DDIO. Fine-tuning DDIO enables us to process packets with a larger number of RX descriptors while providing the *same or better* performance.



We need more RX descriptors for  $\geq 100$  Gbps networks, as additional descriptors reduces the latency incurred by **packet loss** and **PAUSE frames**.



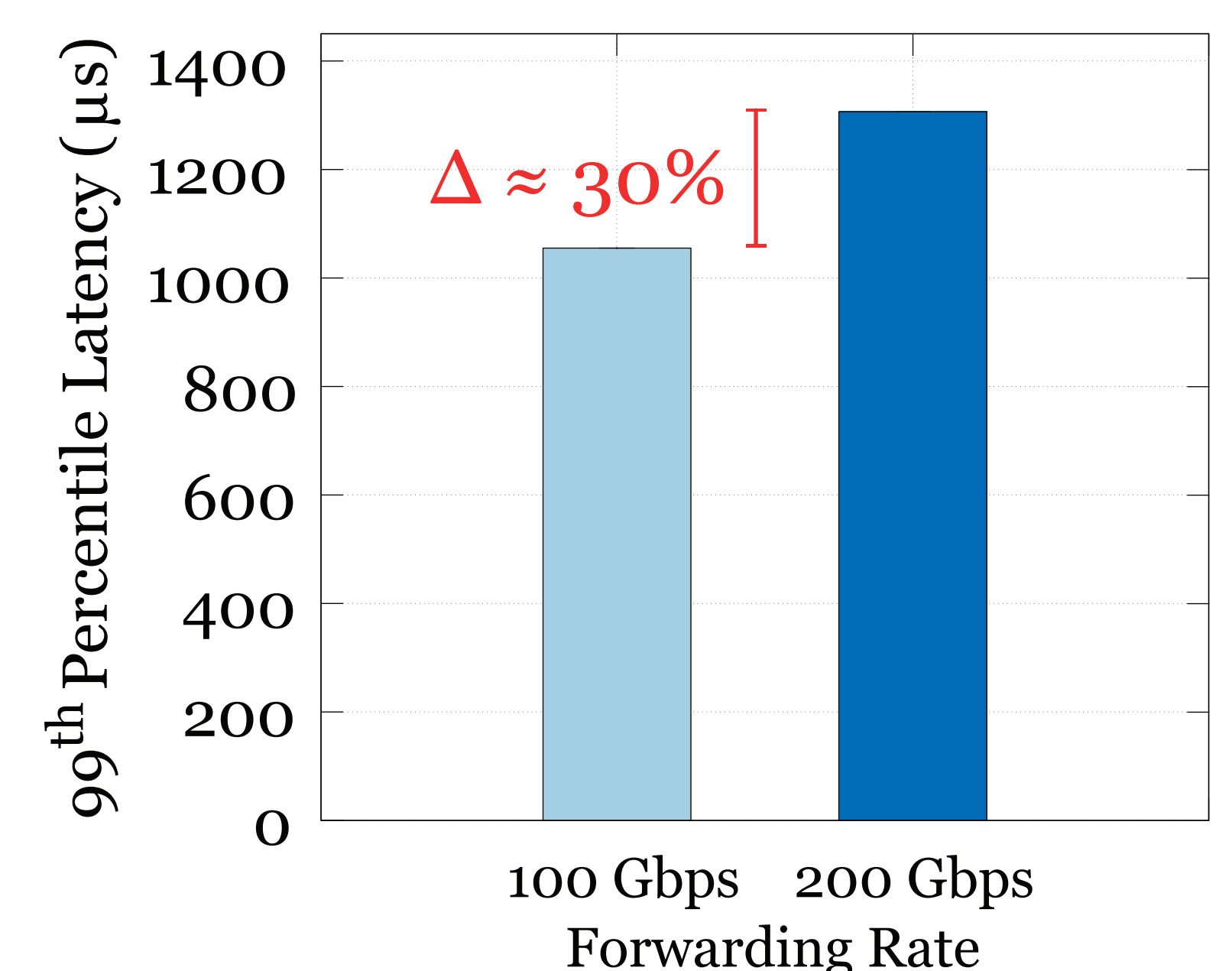
### IIO LLC WAYS

11000000000000

The default value has only 2 set bits

## 5 Toward 200 Gbps

**Problem:** DDIO can degrade performance with faster link speeds, due to the higher cache injection rate.



## 6 Conclusion

There is no **one-size-fits-all** approach to utilize DDIO. Therefore, it is important to optimize DDIO based on the characteristics of applications and their workload, especially for multi-hundred-gigabit networks.

**Approach:** LLC could be bypassed for low-priority or DDIO-insensitive application, thus making room for the high-priority or highly-DDIO-sensitive applications. **Bypassing** could be done via:

- Disabling DDIO for an specific I/O device or
- Exploiting a remote processor's socket to DMA data