Dragonfly : Is Data Migration Evil in the NVM File System?

Jungwook Han, Hongsu Byun, Hyungjoon Kwon,
Sungyong Park and Youngjae Kim

AMGCC`21

Sep 27, 2021

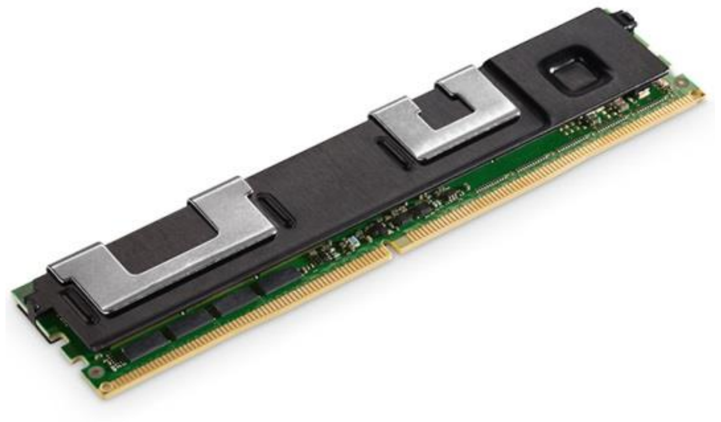
DISCOS LABORATORY

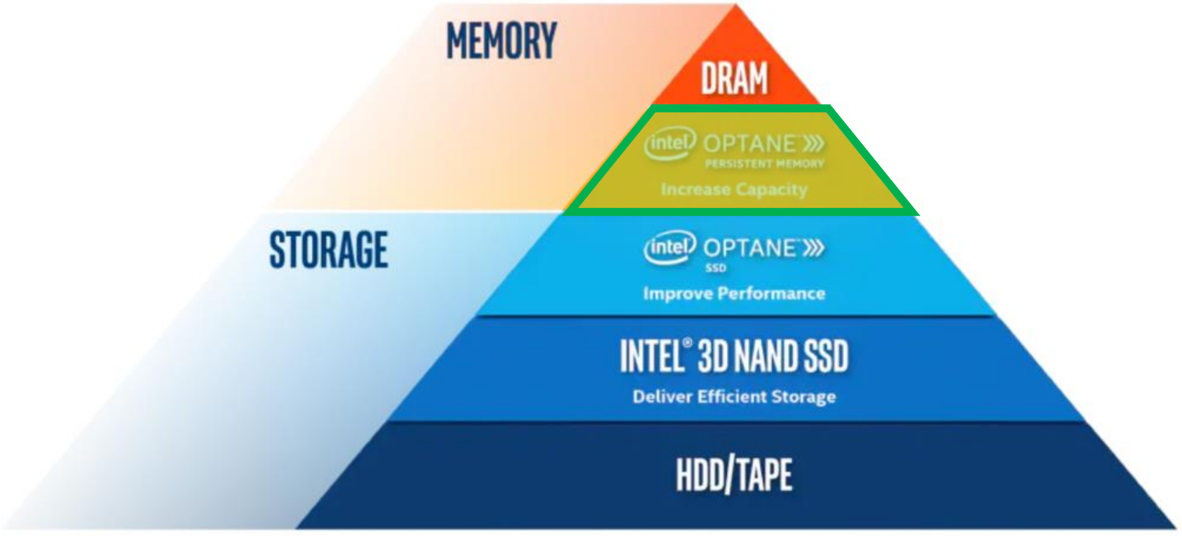
Department of Computer Science and Engineering
Sogang University, Seoul
South Korea

서강대학교 SOGANG UNIVERSITY

Contents

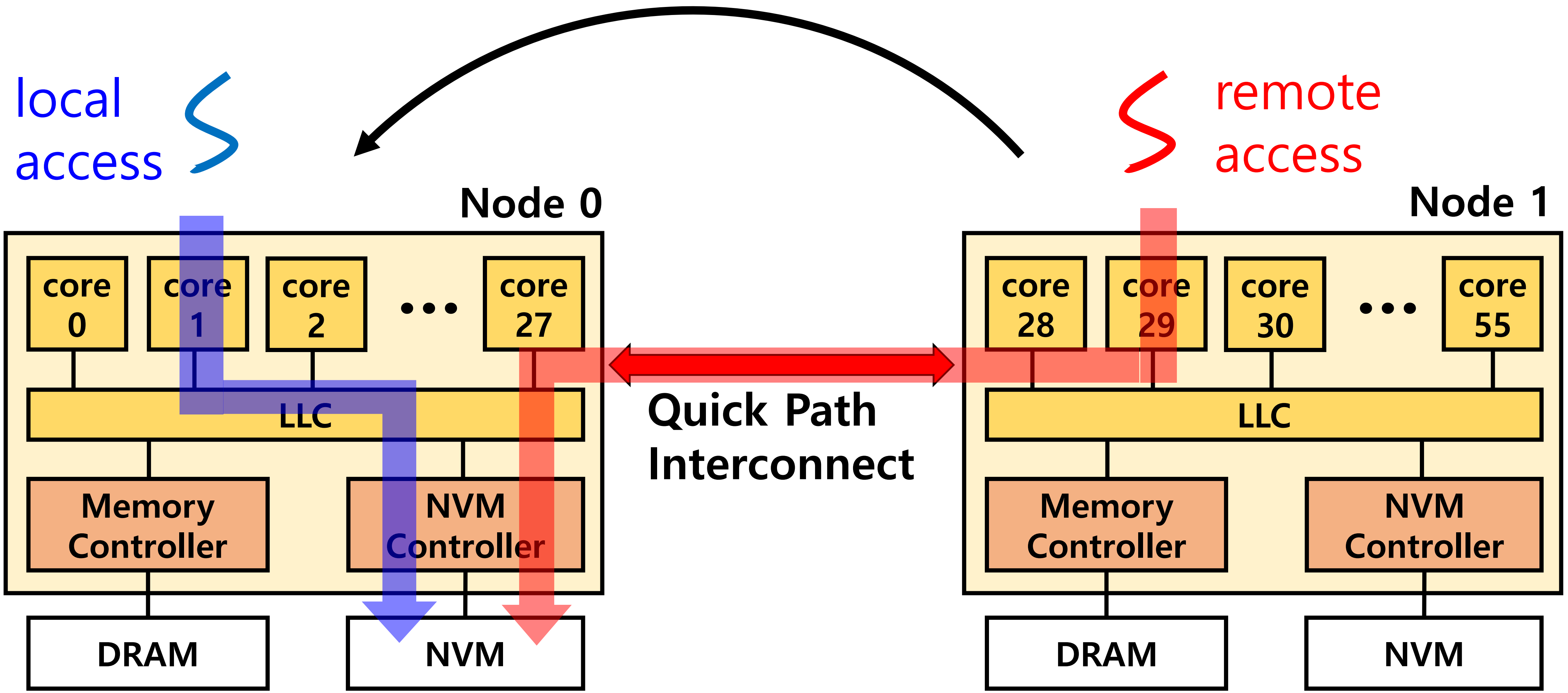
Introduction : Non-Volatile Memory





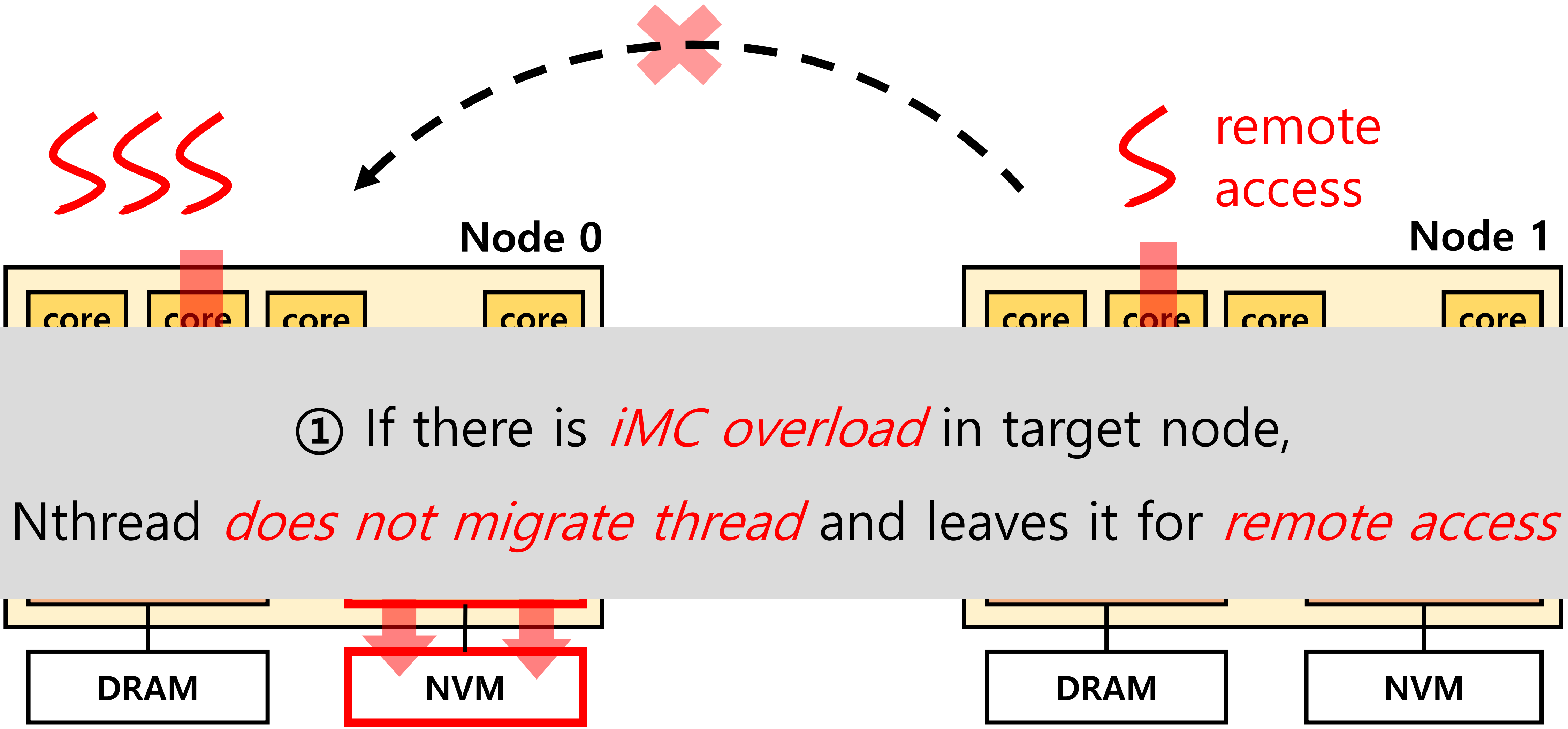
- **Non-Volatile Memory**
  - Low latency
  - High bandwidth
  - Persistent
  - Byte-addressable

Background : Thread Migration[MSST`20]



[MSST`20]J. Wang, D. Jiang, and J. Xiong, "NUMA-Aware Thread Migration for High Performance NVMM File Systems," in Proceedings of the 36th International Conference on Massive Storage Systems and Technology, MSST '20, 2020.

Background : Limitation of Thread Migration



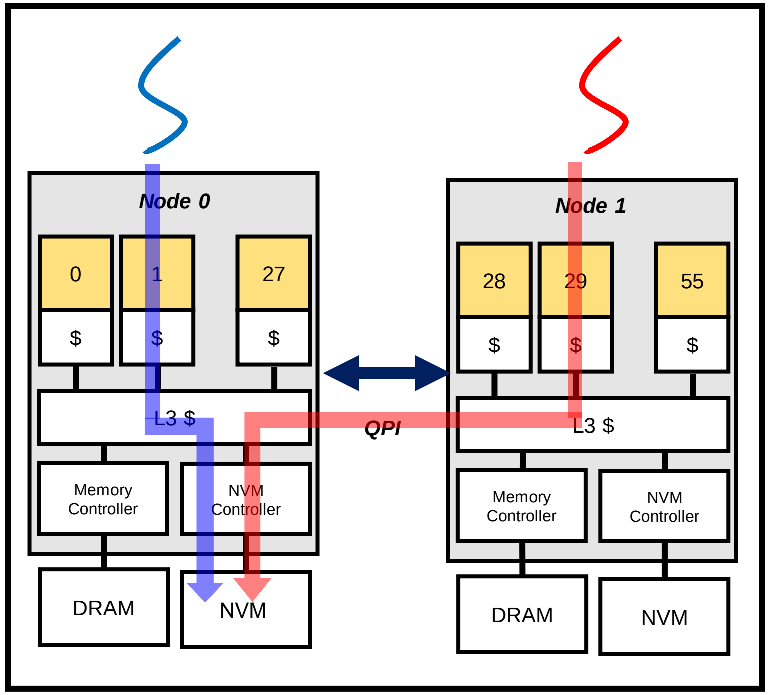
① If there is *iMC overload* in target node,

Nthread *does not migrate thread* and leaves it for *remote access*

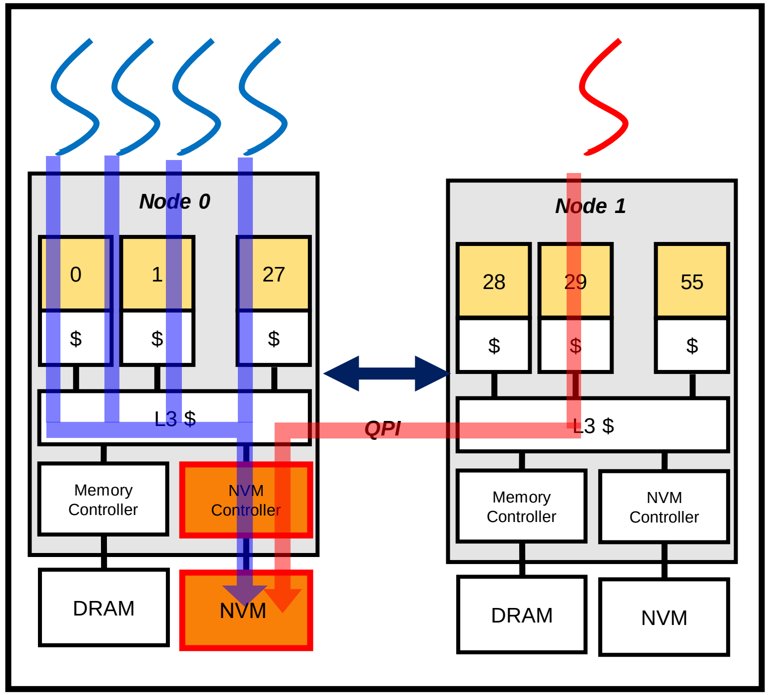
서강대학교 Sogang University

Motivation : Experiment

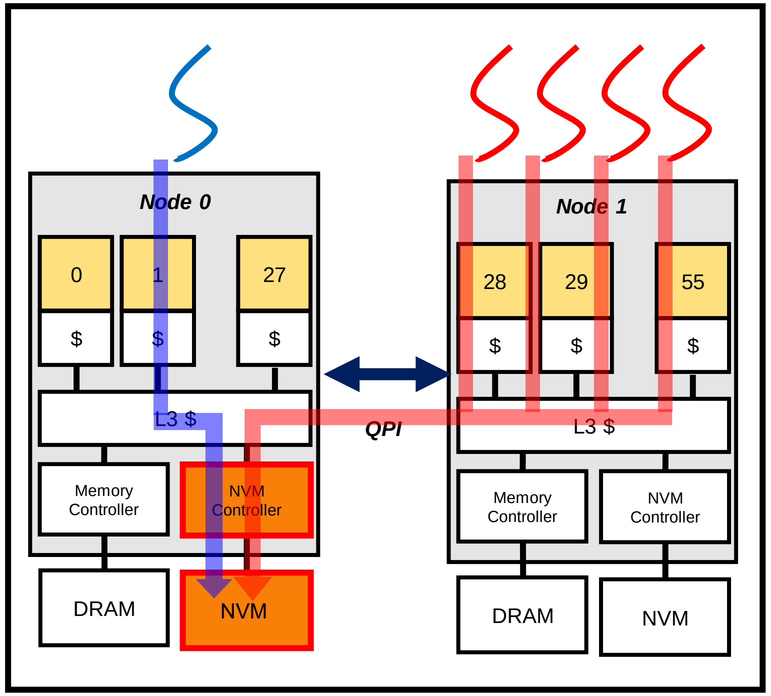
"To compare the performance ***with and without iMC overload***"



Normal



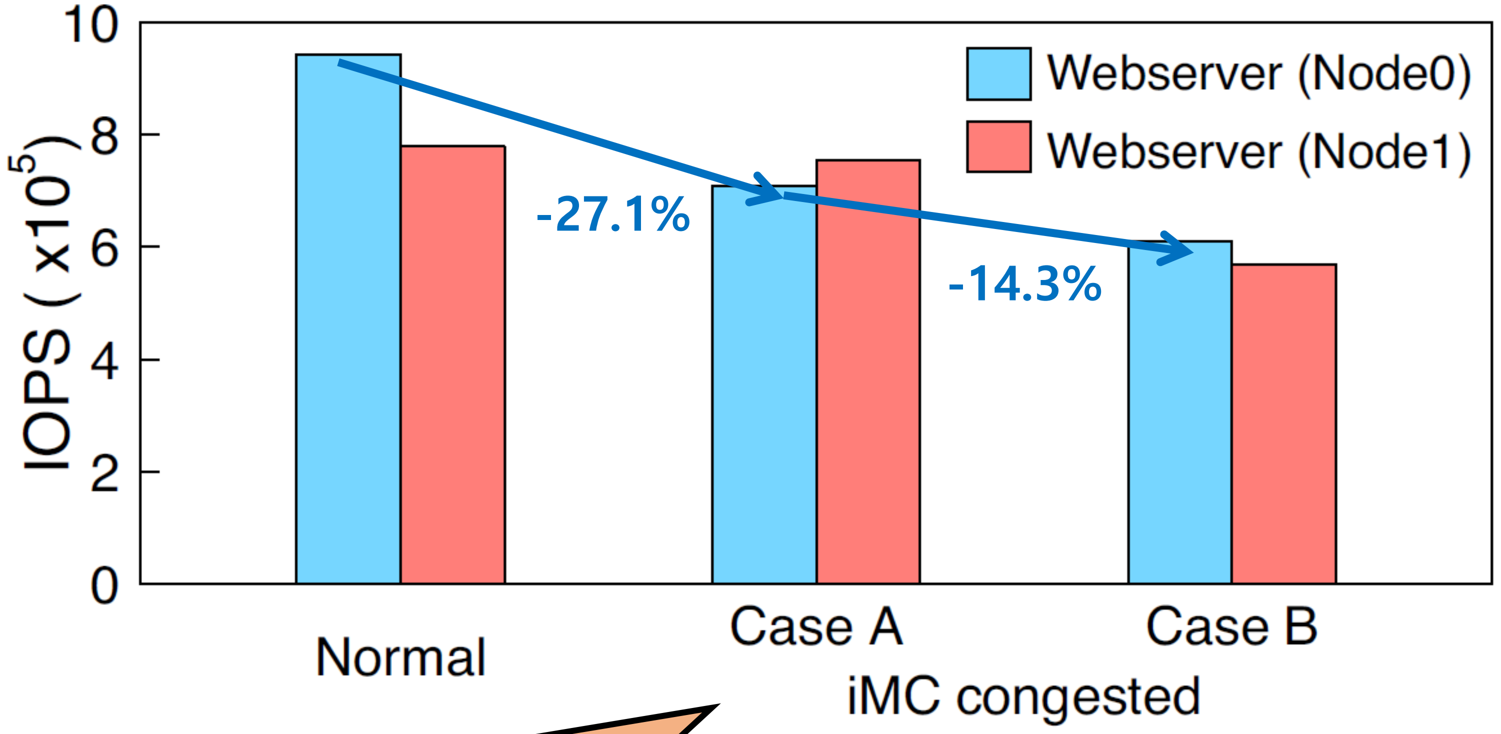
iMC congested (Case A)



iMC congested (Case B)

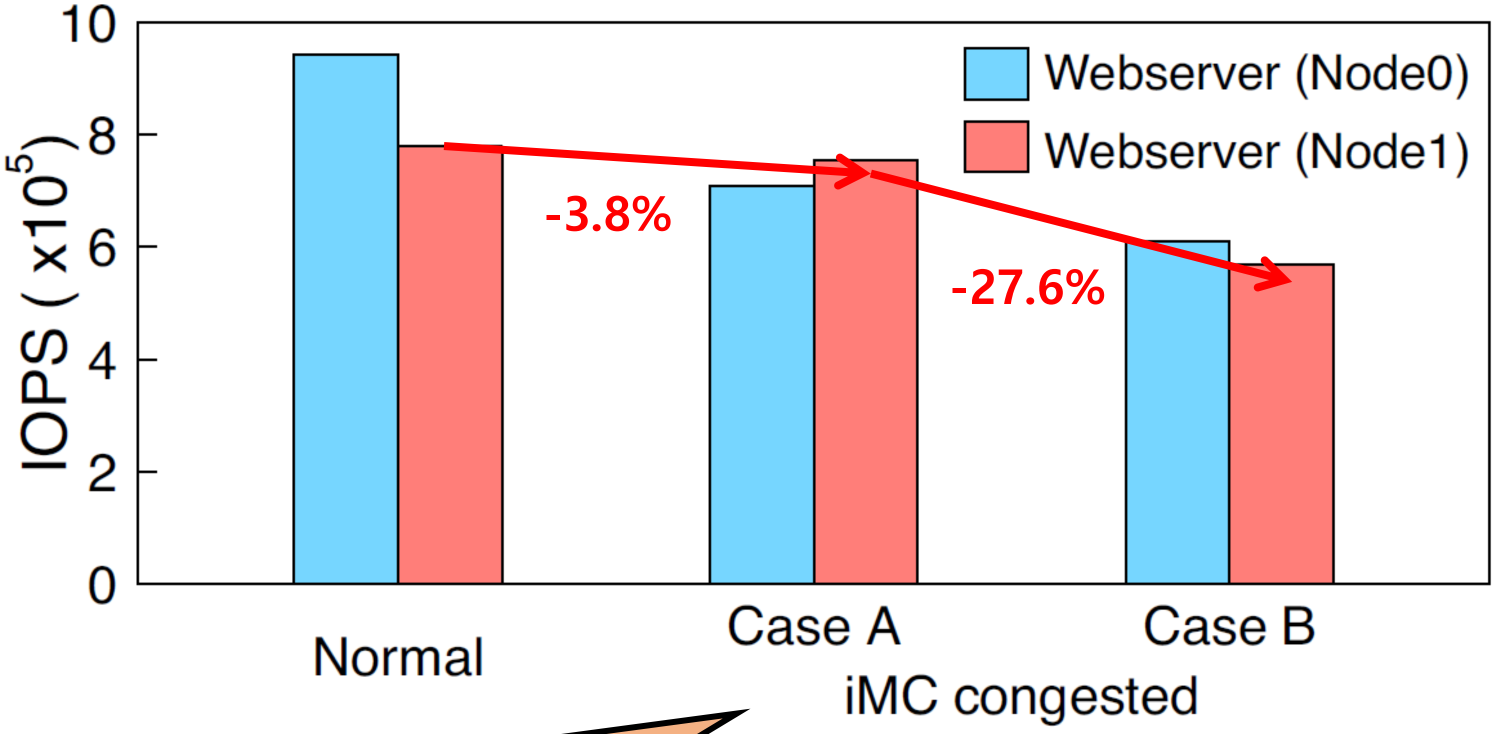
※ **Normal** : Case where the iMC is ***not overloaded***.

Motivation : Experiment result



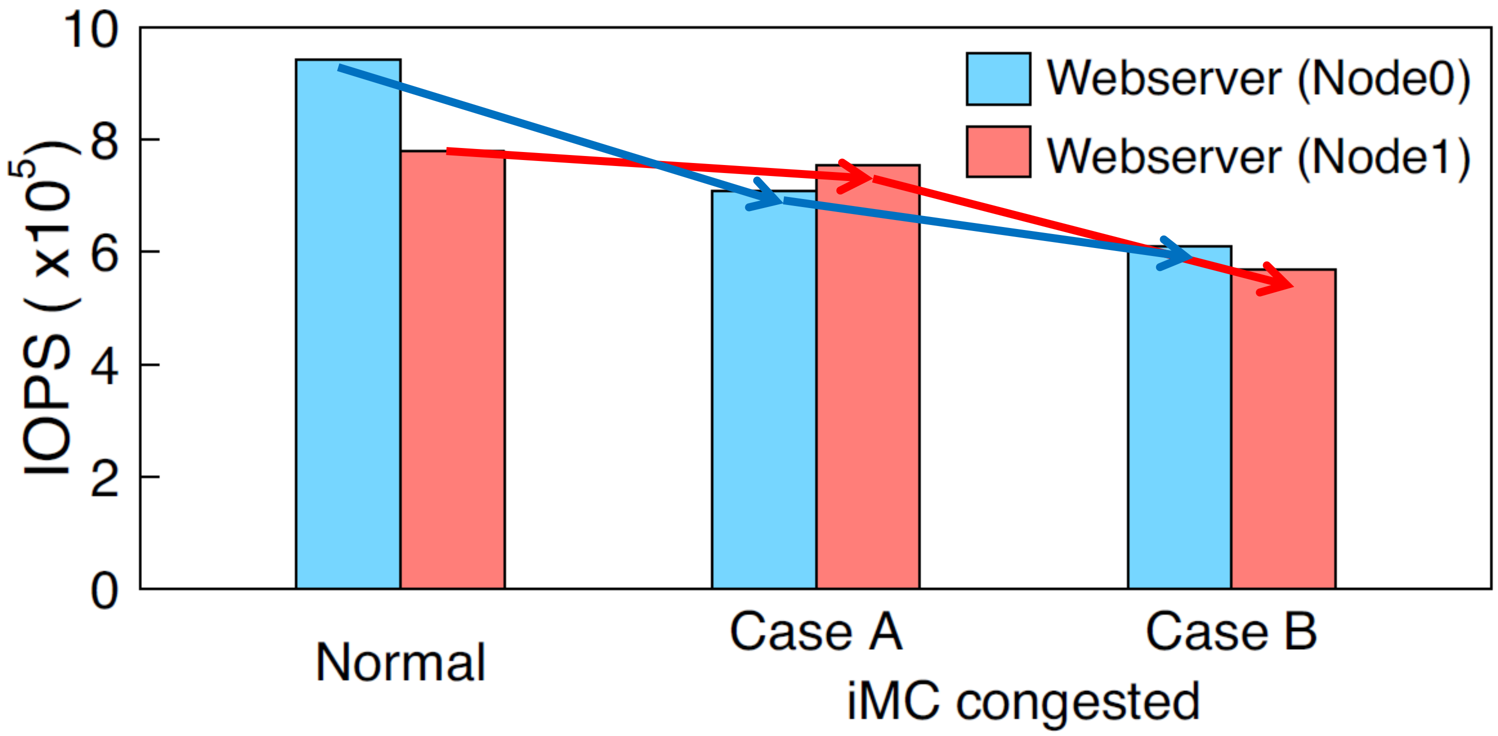
If there is iMC overload, **local access application** throughput **decreases**.

Motivation : Experiment result



If there is iMC overload, **remote access application** throughput **also decreases**.

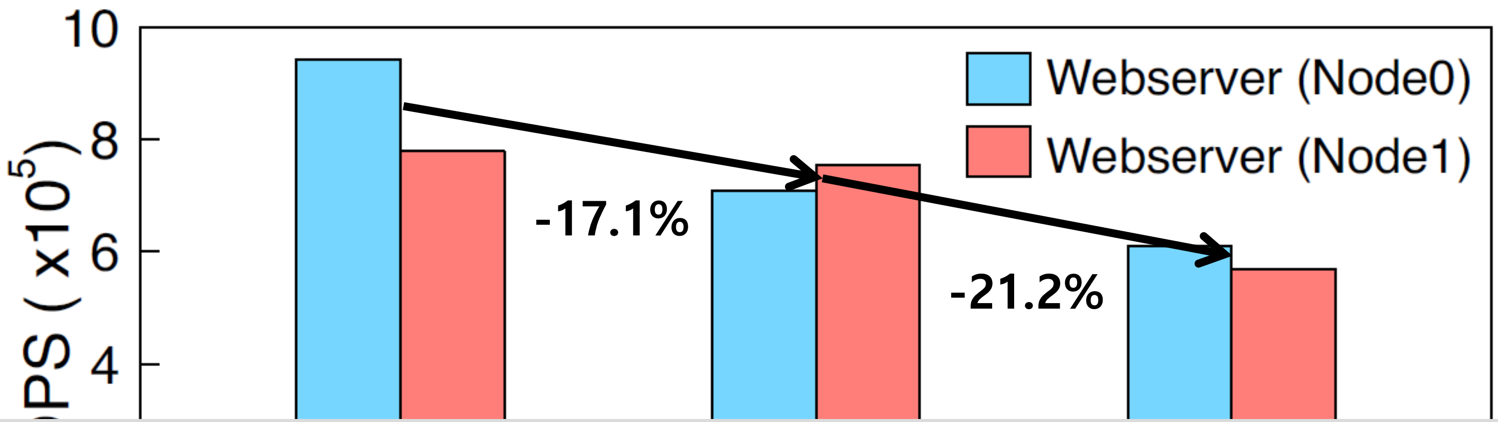
Motivation : Experiment result



If **iMC is overloaded**, both the throughput of remote access and local access **drops**.

서강대학교
SOGANG UNIVERSITY

Motivation : Experiment result



② As a result, if *iMC is overloaded* in optane server,

it has *a fatal effect* on performance.

If **iMC is overloaded**, both the throughput of remote access and local access **drops**.

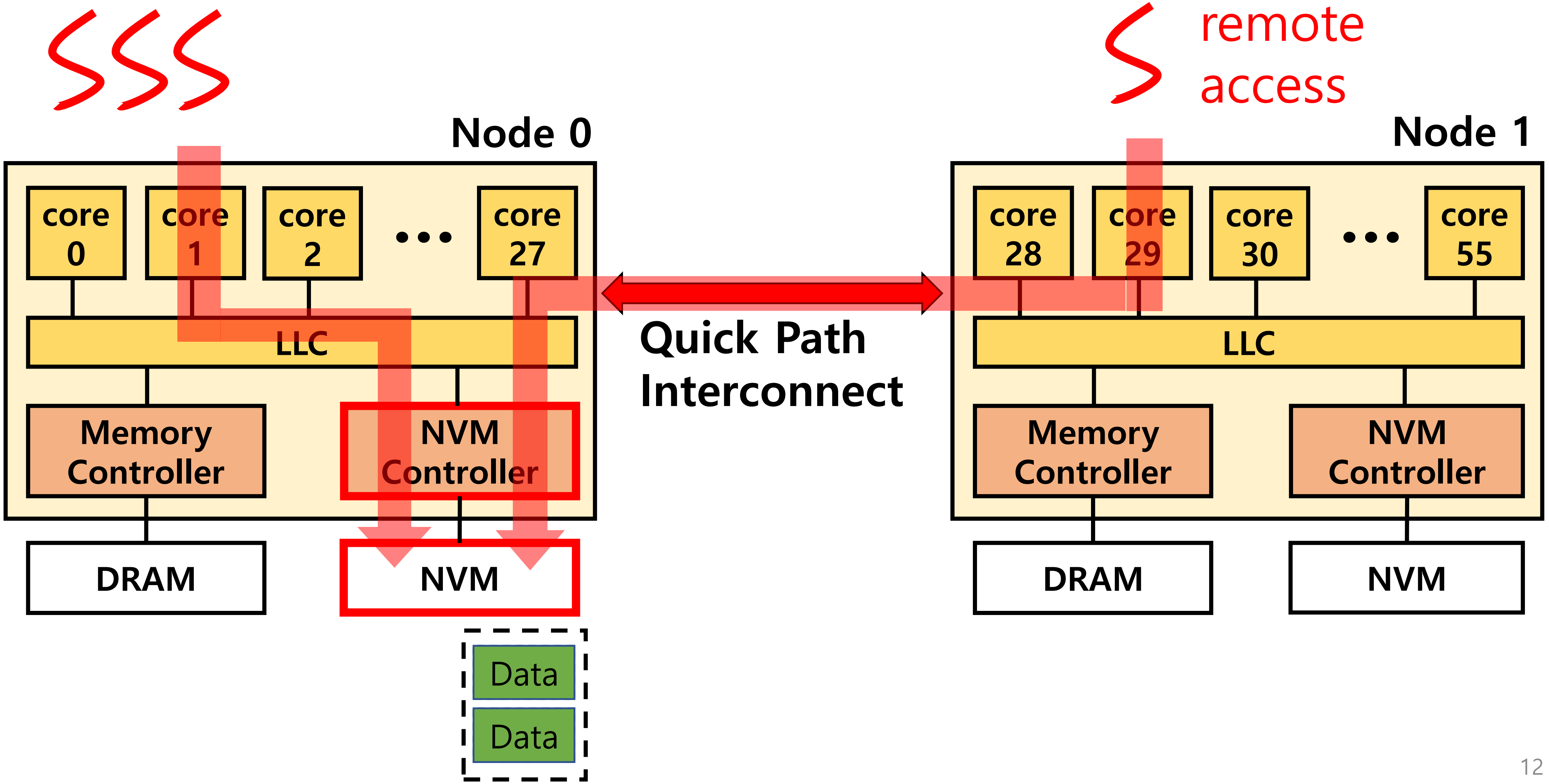
As **the ratio of remote access** increases, the overall throughput **severely** decreases.

Motivation : Problem Definition

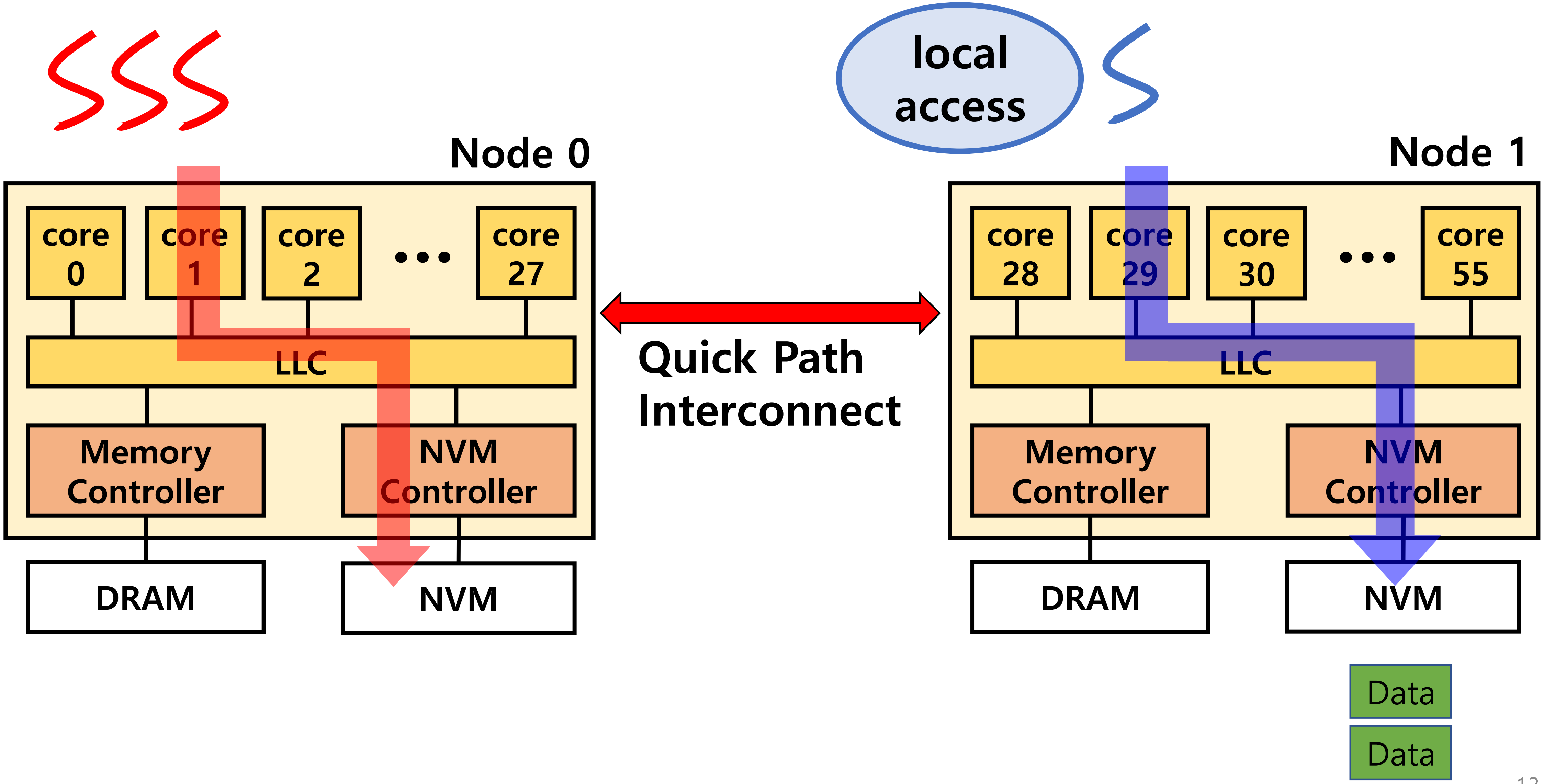
① If there is *iMC overload* in target node, Nthread *does not migrate thread* and leaves it for *remote access*

② As a result, if *iMC is overloaded* in optane server, it has *a fatal effect* on performance.

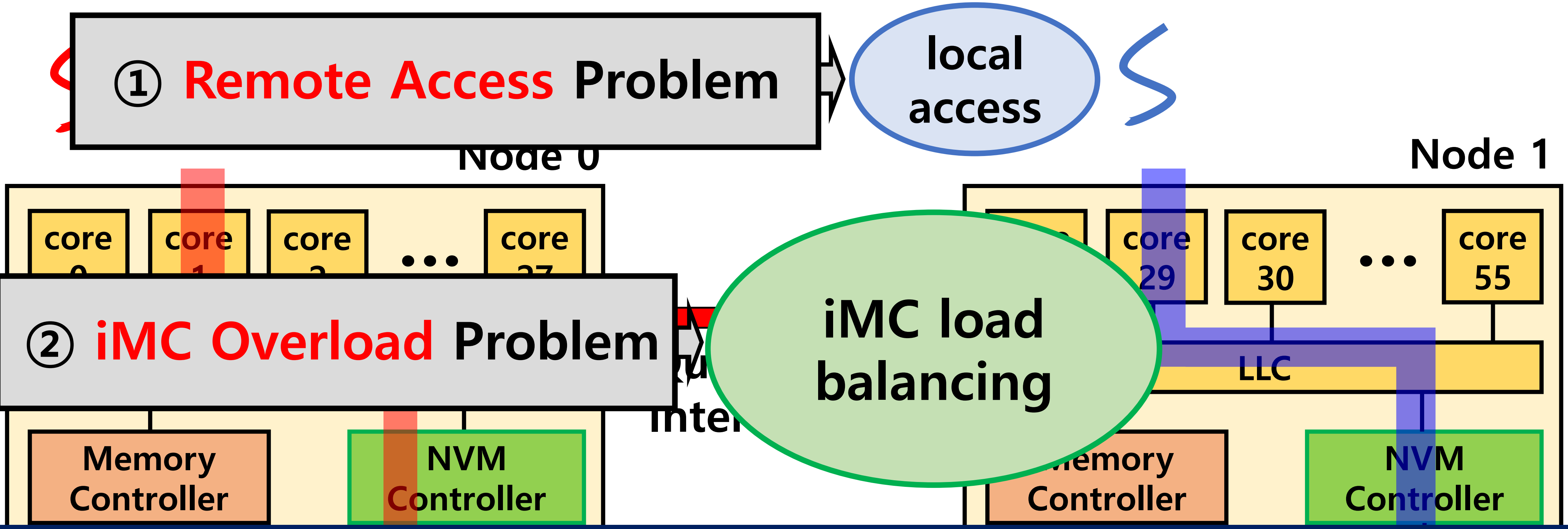
Motivation : Data Migration Solve The Problems



Motivation : Data Migration Solve The Problems



Motivation : Data Migration Solve The Problems



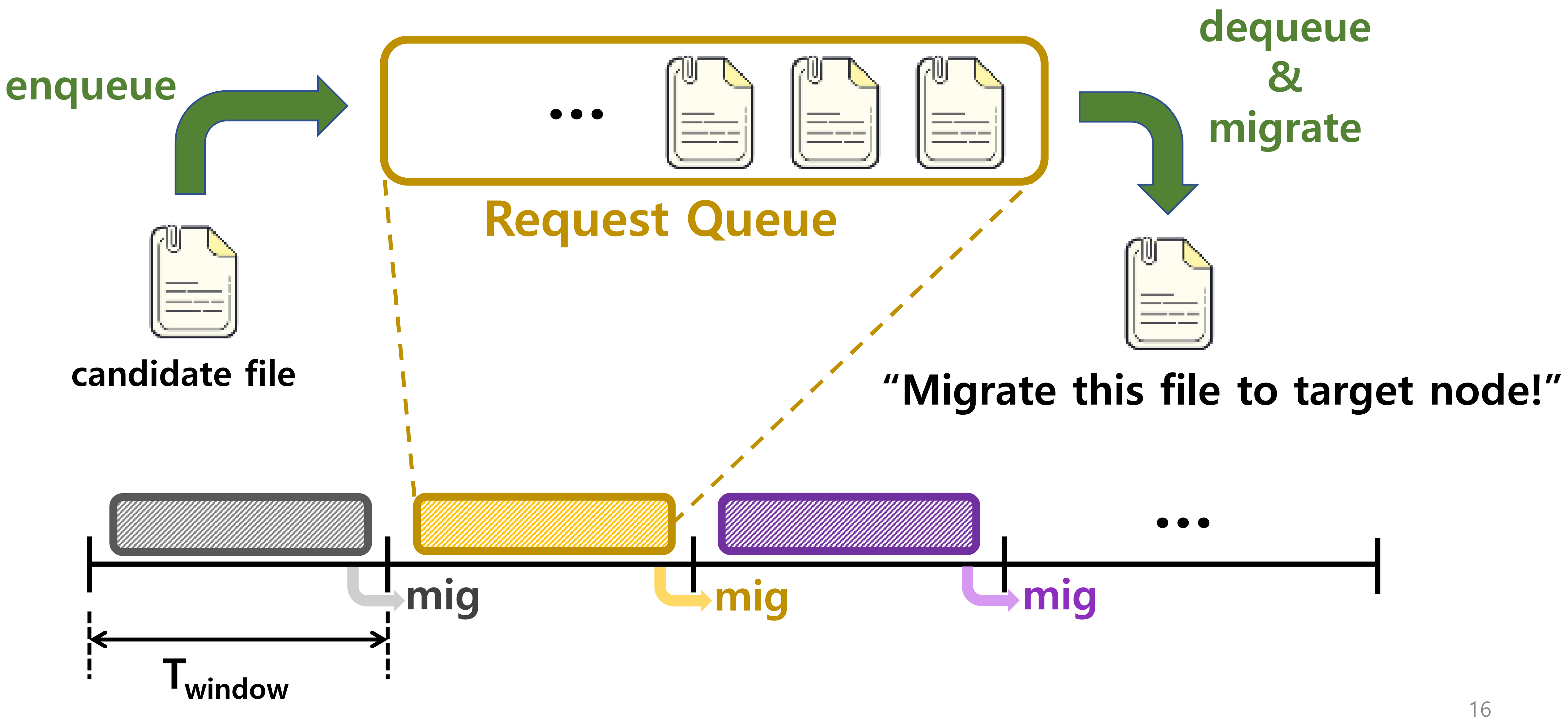
BUT! Data Migration Overhead Exists!

Design of Dragonfly : Overview

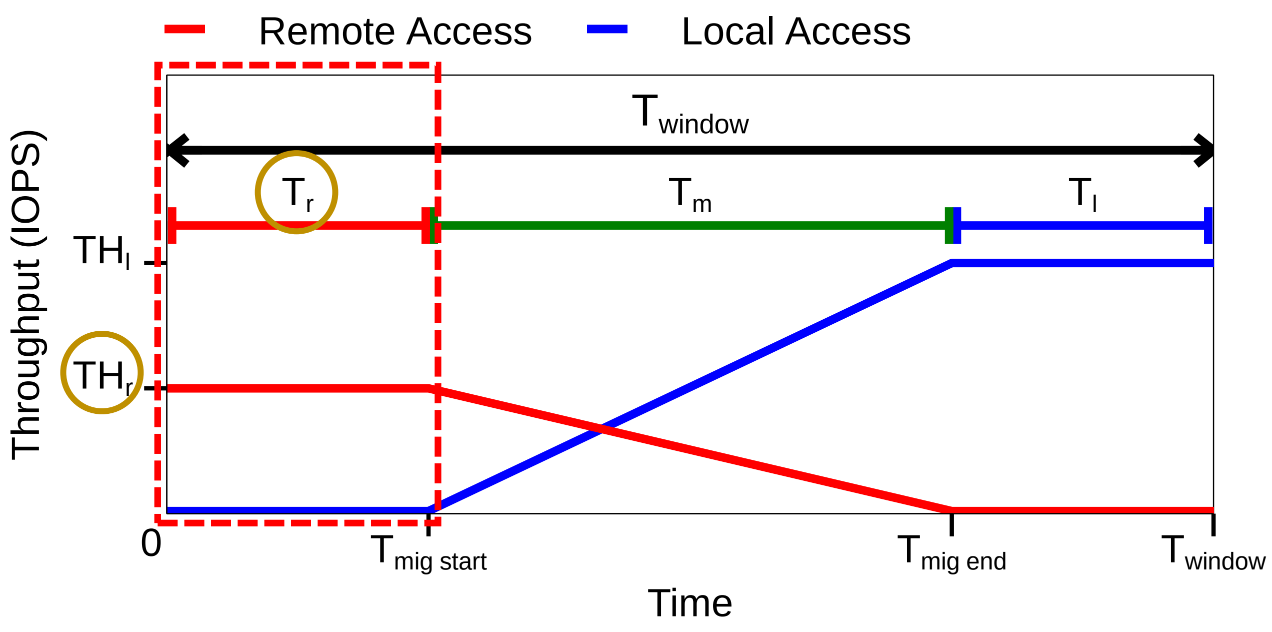
- Dragonfly is a ***Data Migration Module*** in NVM filesystem
- Implemented on ***NOVA,*** which is NVM filesystem
- ***Uses Request Queue*** as a core data structure
- Migrates Data through ***MTP, Migration Triggering Policy***

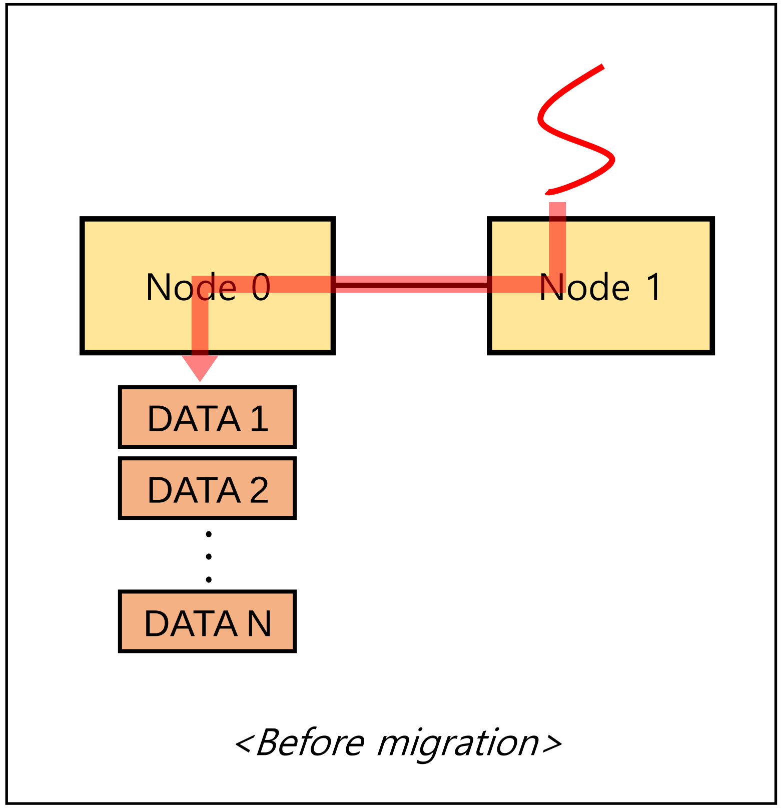
J. Xu and S. Swanson, "NOVA: A Log-structured File System for Hybrid Volatile/Non-volatile Main Memories," in Proceedings of the USENIX Conference on File and Storage Technologies, FAST '16, 2016.

Design of Dragonfly : Request Queue



Design of Dragonfly : MTP (Migration Trigger Policy)





T_r = time for remote access

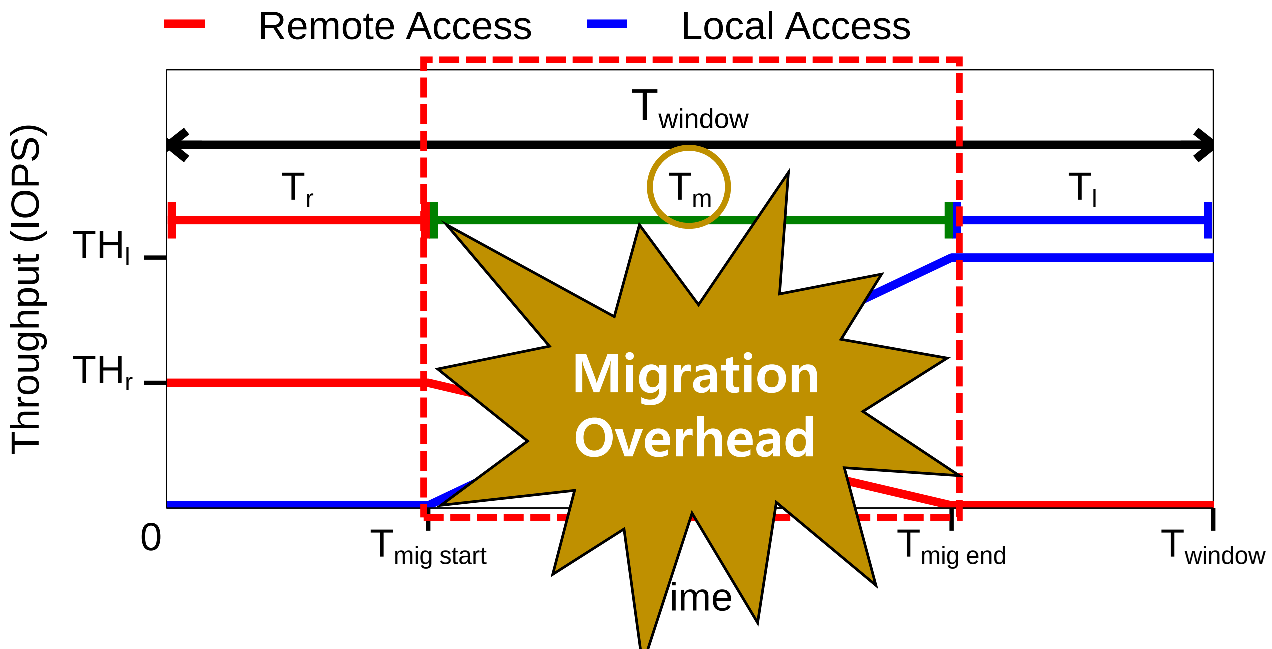
T_l = time for local access

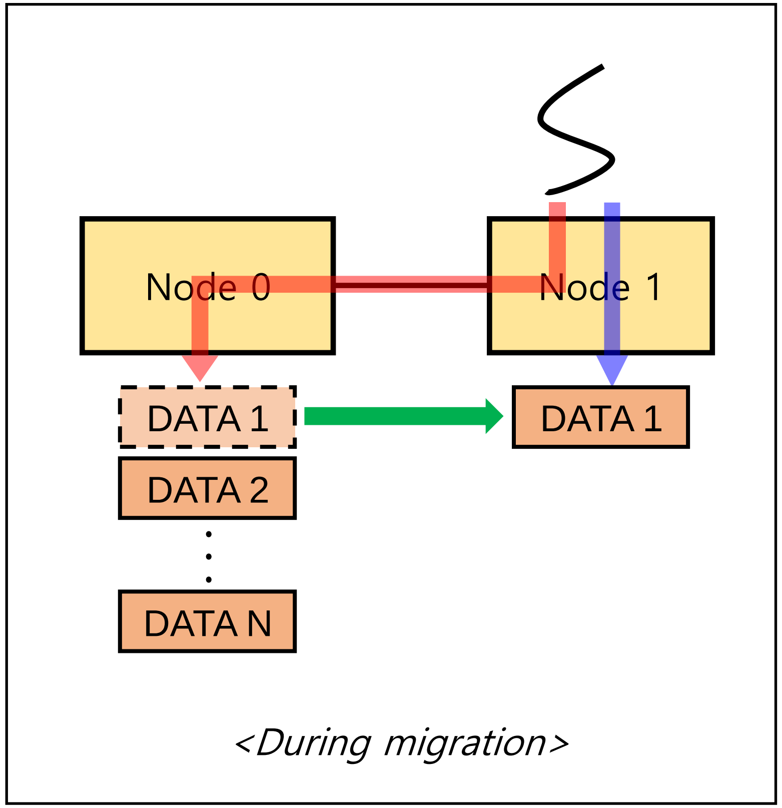
T_m = migration time

TH_r = throughput for remote access

TH_l = throughput for local access

Design of Dragonfly : MTP (Migration Trigger Policy)





T_r = time for remote access

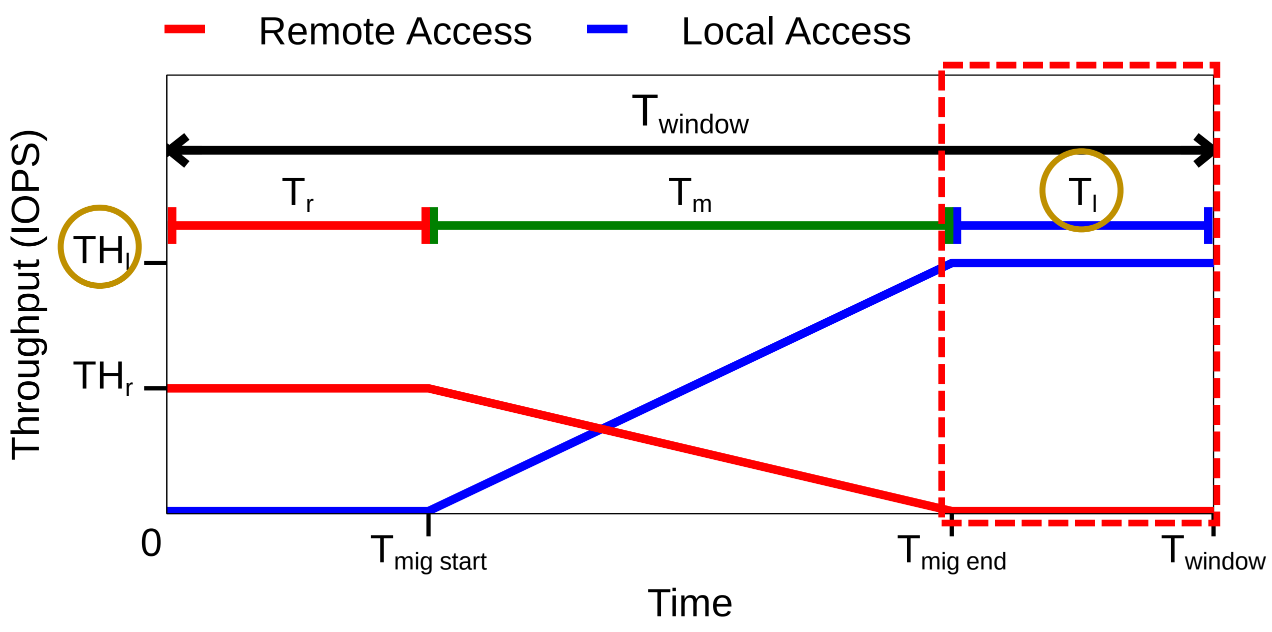
T_l = time for local access

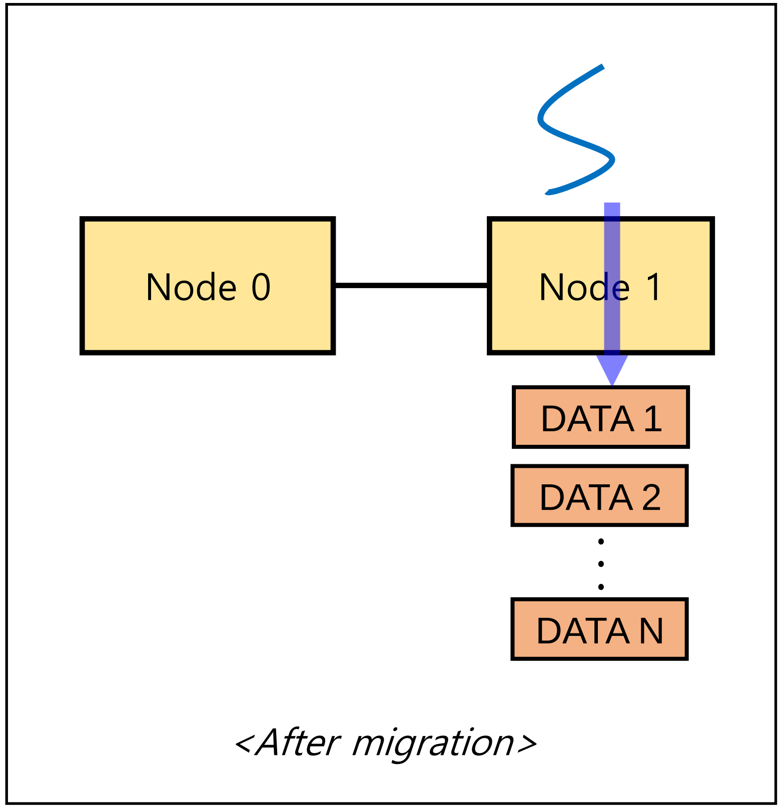
T_m = migration time

TH_r = throughput for remote access

TH_l = throughput for local access

Design of Dragonfly : MTP (Migration Trigger Policy)





T_r = time for remote access

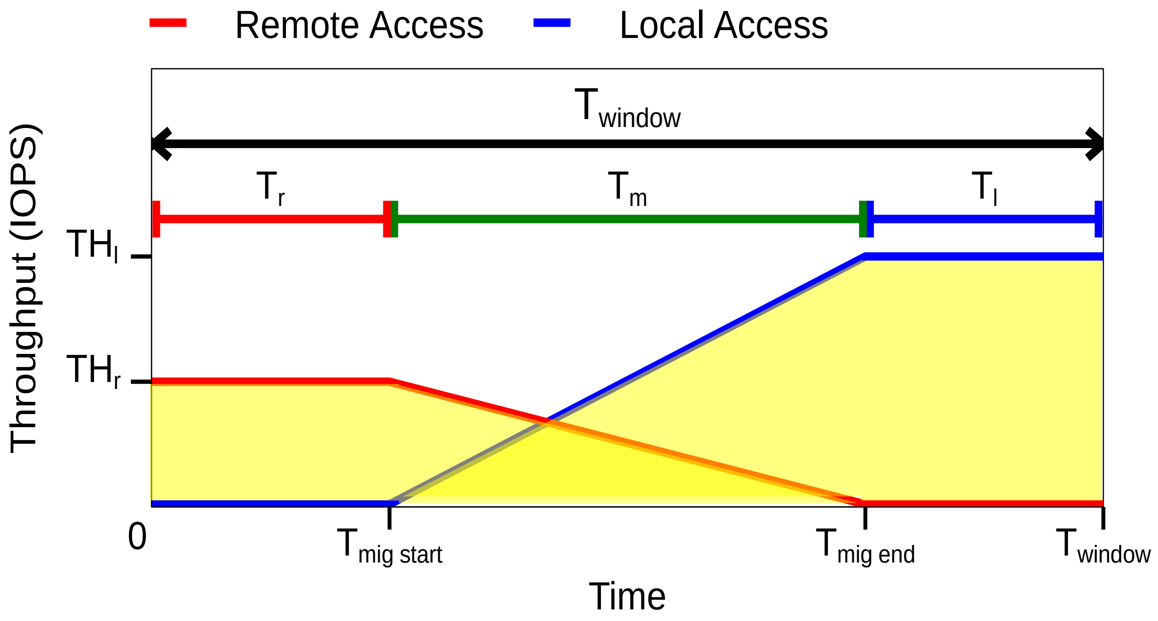
T_l = time for local access

T_m = migration time

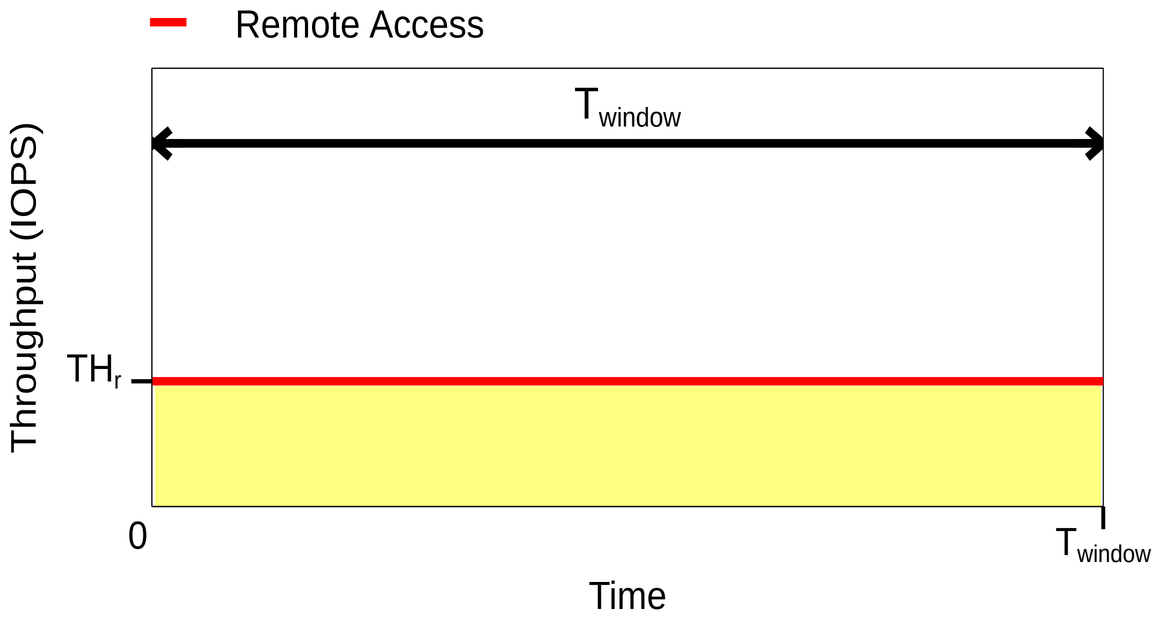
TH_r = throughput for remote access

TH_l = throughput for local access

Design of Dragonfly : MTP (Migration Trigger Policy)

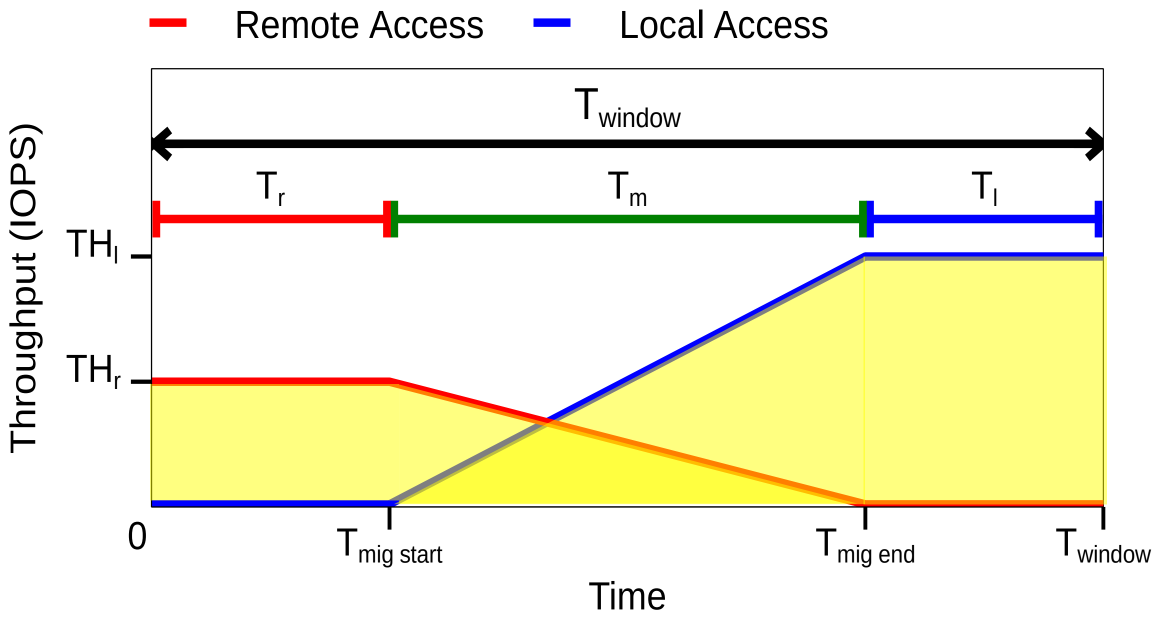


$$TH_r \times T_r + \int_0^{T_m} \{TH_r + \frac{TH_l - TH_r}{T_m} \times t\} dt - O_m + TH_l \times T_l$$

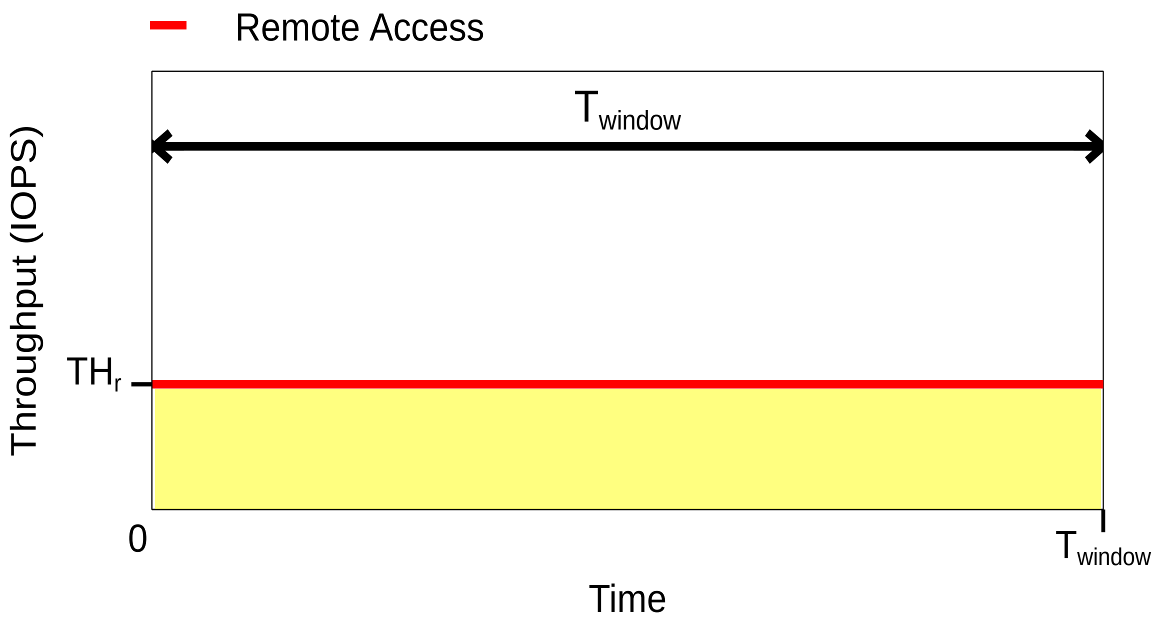


$$TH_r \times T_{window}$$

Design of Dragonfly : MTP (Migration Trigger Policy)

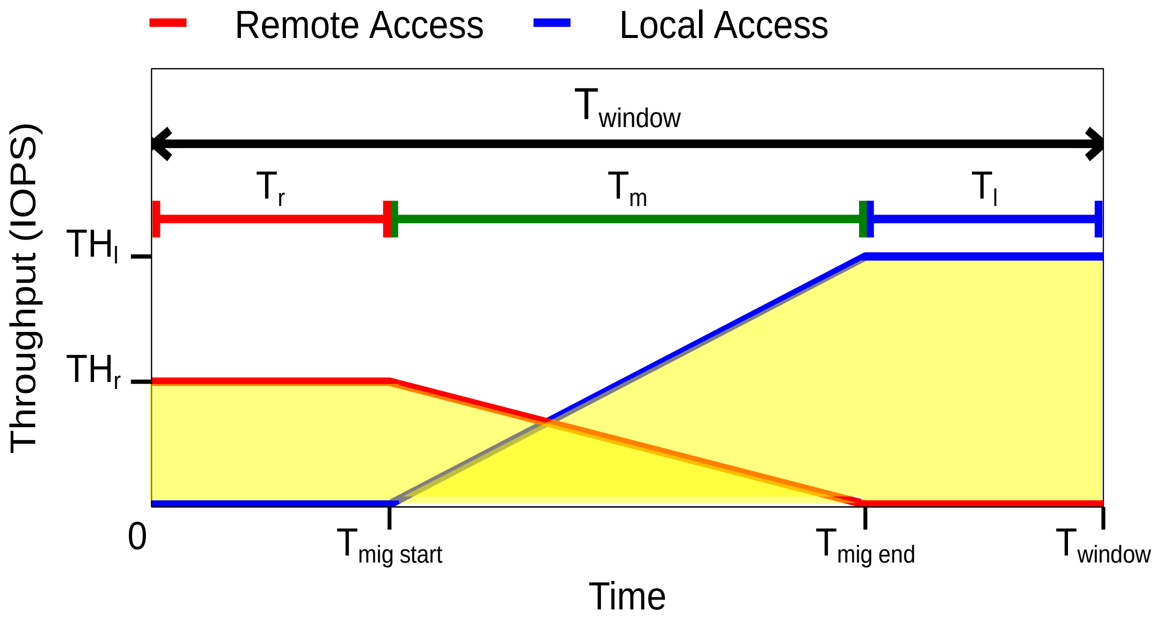


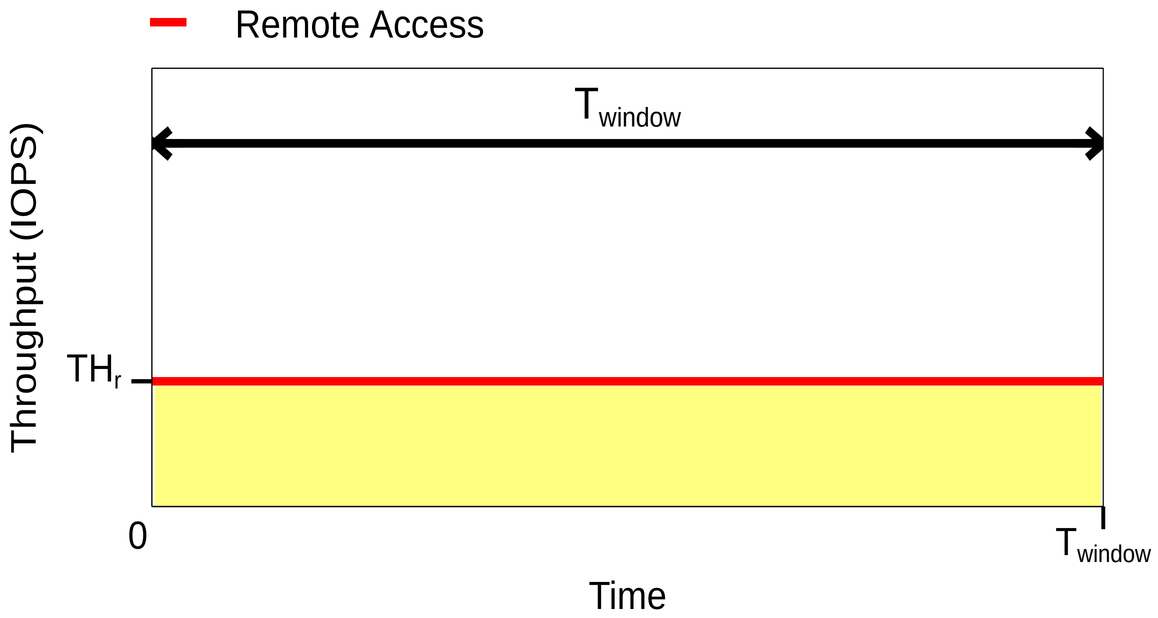
$$TH_r \times T_r + \int_0^{T_m} \{TH_r + \frac{TH_l - TH_r}{T_m} \times t\} dt - O_m + TH_l \times T_l$$



$$TH_r \times T_{window}$$

Design of Dragonfly : MTP (Migration Trigger Policy)





$$TH_r \times T_r + \int_0^{T_m} \{TH_r + \frac{TH_l - TH_r}{T_m} \times t\} dt - O_m + TH_l \times T_l \quad > \quad TH_r \times T_{window}$$

$$T_{window} = T_r + T_m + T_l$$

$$T_{window} \approx T_m + T_l, \ (\because (1), \ T_r \approx 0)$$

$$T_m < 2 \times (T_{window} - K) \quad (K = \frac{O_m}{TH_l - TH_r})$$

서강대학교 SOGANG UNIVERSITY

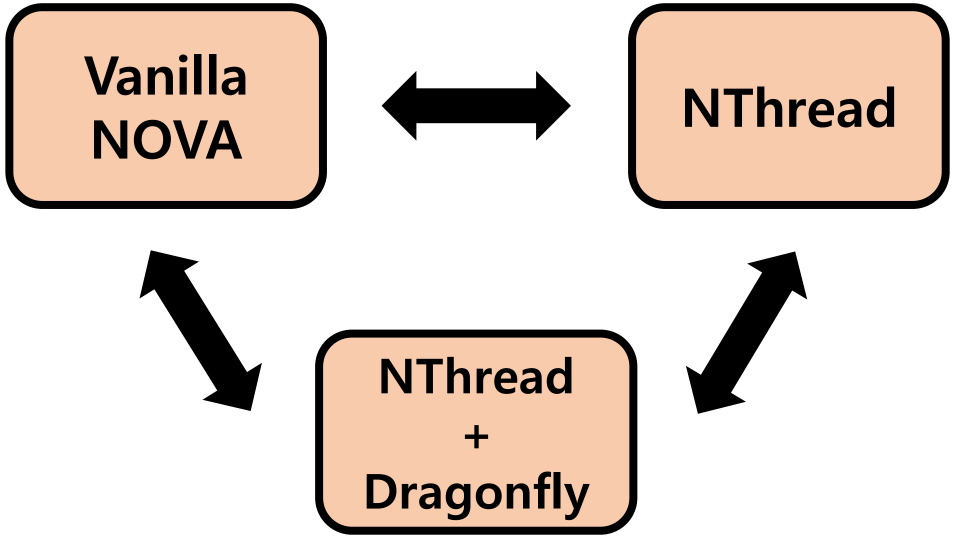
Evaluation

Testbed

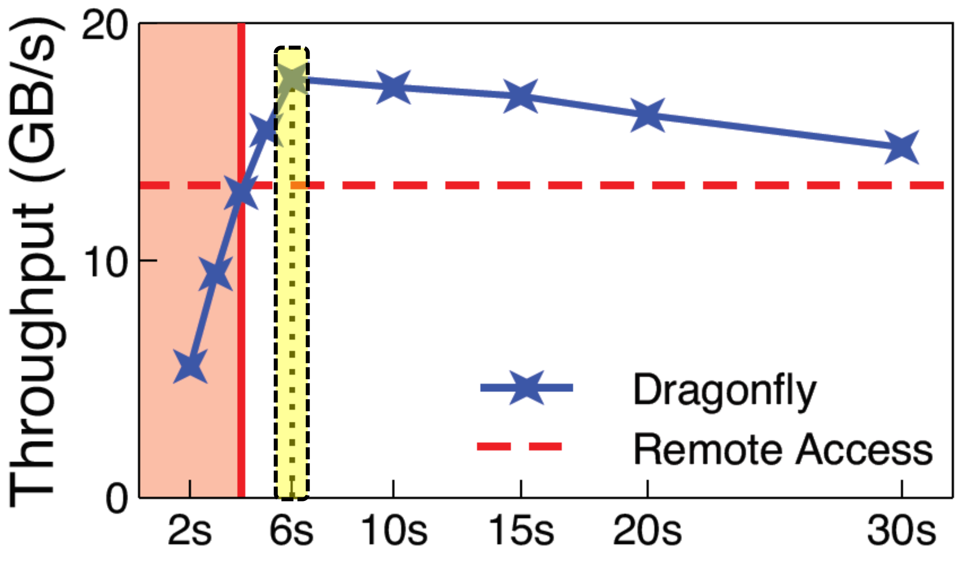
CPU	Intel(R) Xeon(R) Platinum 8280M v2 2.70GHz CPU Nodes (#): 2, Cores per Node (#): 28
Memory	DRAMs per Node (#): 6, DDR4, 64 GB * 12 (=768GB)
PM	Intel Optane DC Persistent Memory PMs per Node (#): 6, 128 GB * 12 (=1.5TB)
OS	Linux kernel 5.1.0

Configurations (Synthetic Workloads generated via **Filebench** Benchmark)

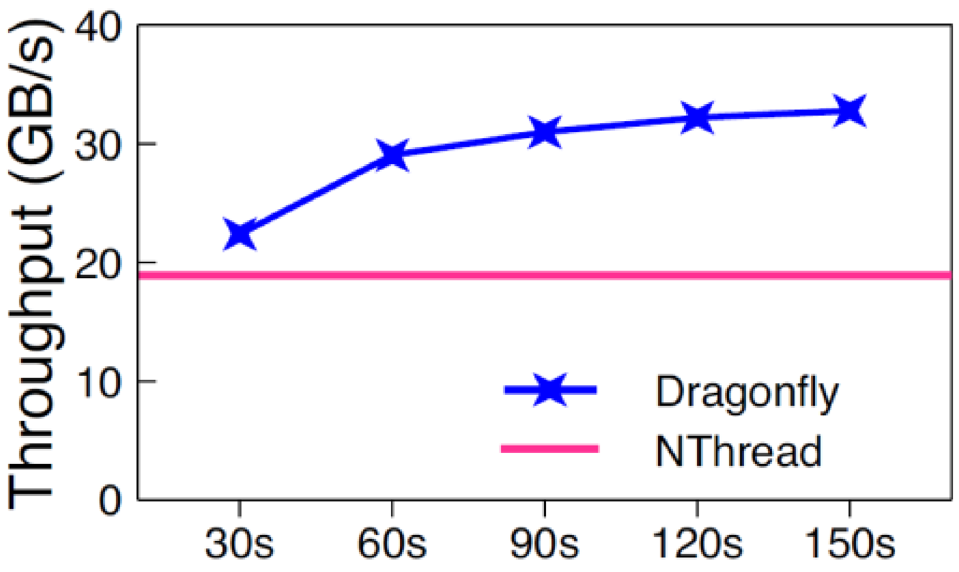
Application	Heavy Case			Light Case			Read:
	Size (KB)	File (#)	Thr (#)	Size (KB)	File (#)	Thr (#)	Write Ratio
Webserver	160	10K	14	16	10K	7	10:1
Webproxy	160	100K	14	64	10K	7	5:1
Videoserver	2GB	50	28	1GB	50	14	RO
Fileserver	128	10k	14	128	1K	7	1:2



Evaluation : Webserver (Heavy) app.



(c) Time Window (T_{window})

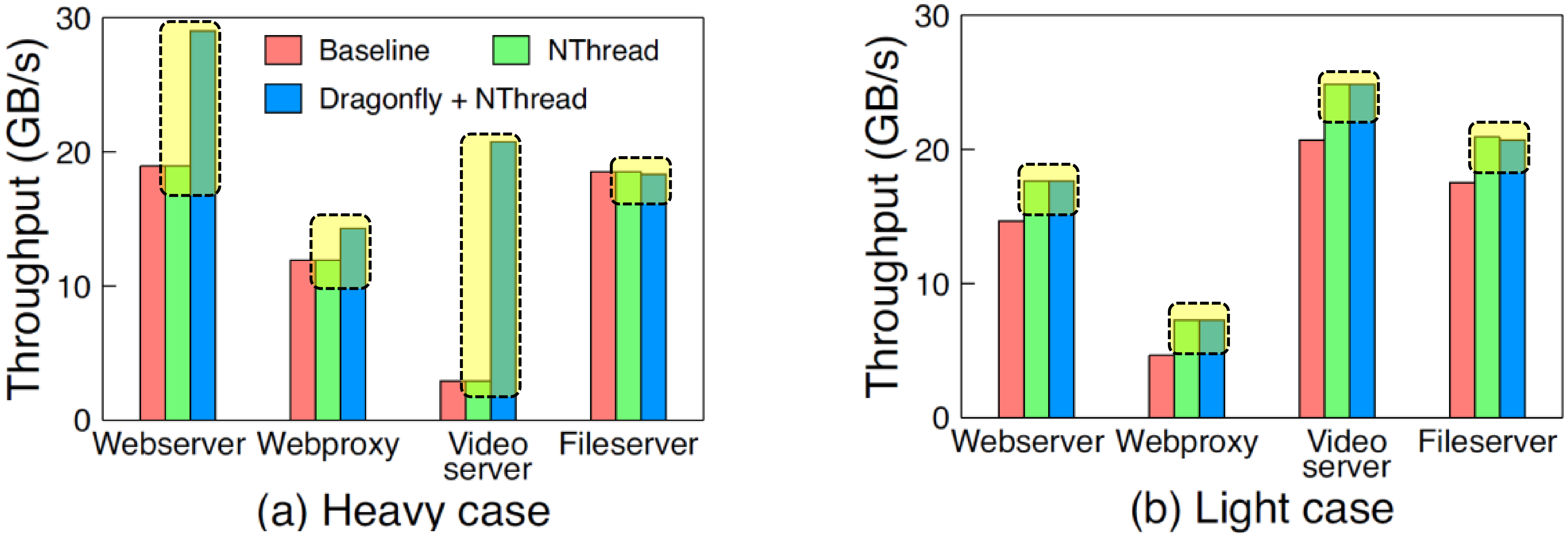


(d) Runtime

The **best efficiency** was shown at ***6*** seconds for **Webserver (Heavy)**.

The longer the runtime, the greater the **benefit from local access**.

Evaluation



Dragonfly works well

1) **For read-intensive workload**
2) **When there is iMC overload in target node**

Conclusion

We proposed Dragonfly, data migration module in NOVA filesystem

1. Introduce "***MTP***", a model-based migration policy to maximize the benefit of data migration
2. Dragonfly maximize the ***local access*** and ***distribute the load*** of iMC.
3. As a result, Dragonfly showed an average 3.26× and a maximum of 7.1× higher performance of the ***read-intensive workload*** than NThread in the situation where the ***iMC was overloaded***.

Data migration with ***a well-defined policy*** is effective in ***NVM filesystem.***

DISCOS LABORATORY

Thank you

Q & A

Jungwook Han

Sogang University Seoul, Republic of Korea

<immerhjw@gmail.com>