Practical Memory Disaggregation

A Case Study in Network-Informed Data Systems Design

Mosharaf Chowdhury

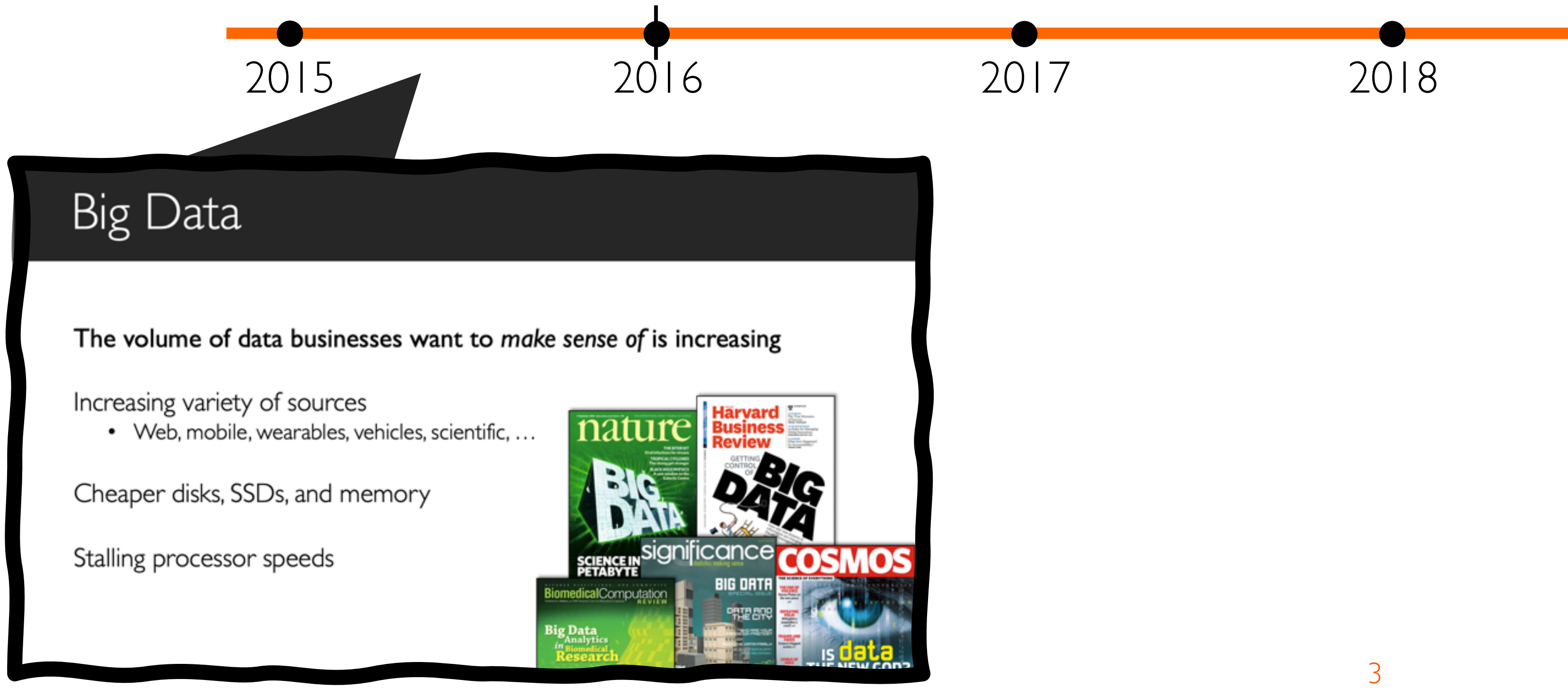
November 2020

SymbioticLab

UNIVERSITY OF MICHIGAN

Five Years Ago…

The volume of data businesses want to *make sense of* is increasing



1. Data Volume Will Keep Increasing

2015 2016 2017 2018

Big Data

The volume of data businesses want to *make sense of* is increasing

Increasing variety of sources
- Web, mobile, wearables, vehicles, scientific, …

Cheaper disks, SSDs, and memory

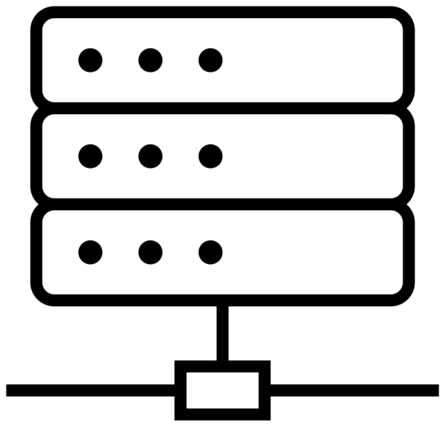
Stalling processor speeds

Data Systems

Big Data Analytics
AI/ML Tools

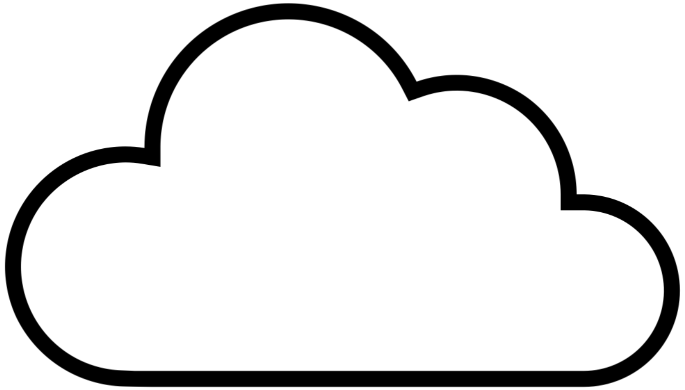
- Massive data
- High parallelism
- GPU clusters
- Distributed
- …

2. Deployed in Diverse Networks



Within a Rack

< 10 μs



Within a Datacenter

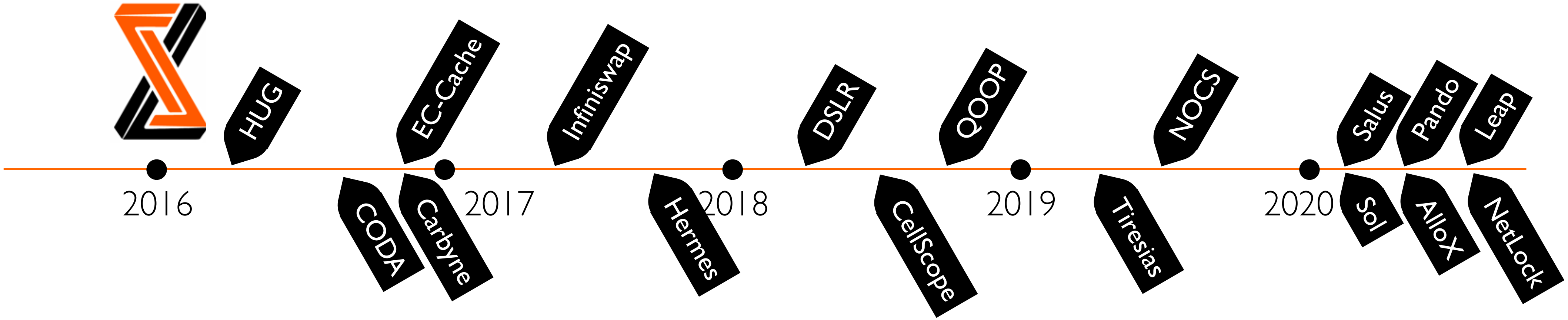
~ 1 ms

Over the World

> 100 ms

Network-Informed Data Systems Design

I. Network-adaptive Big Data and AI/ML systems

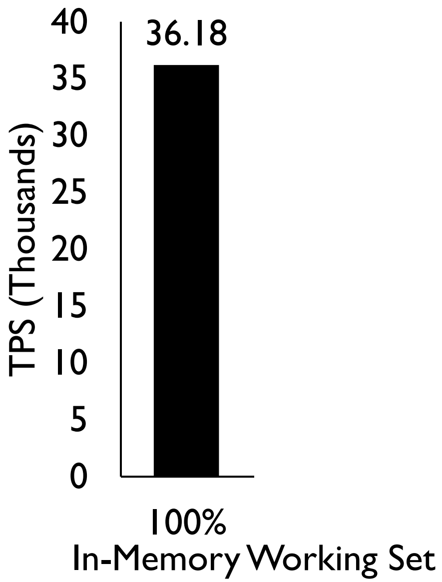


II. Tailoring data systems to extreme networks
   I. Computation over the Internet
   II. Leveraging high-speed networks

Practical Memory Disaggregation

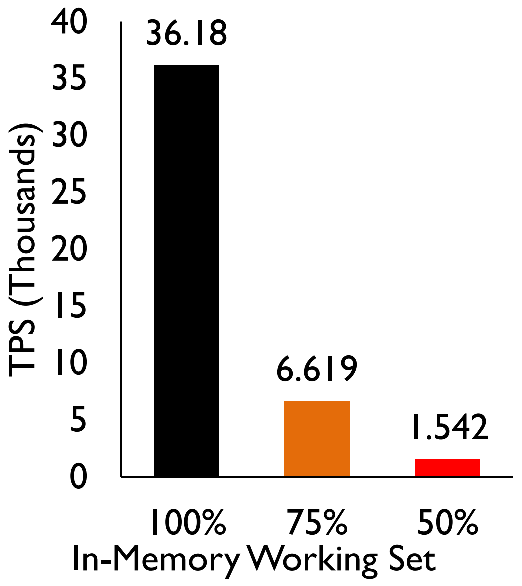
Memory is King!

Perform Great!



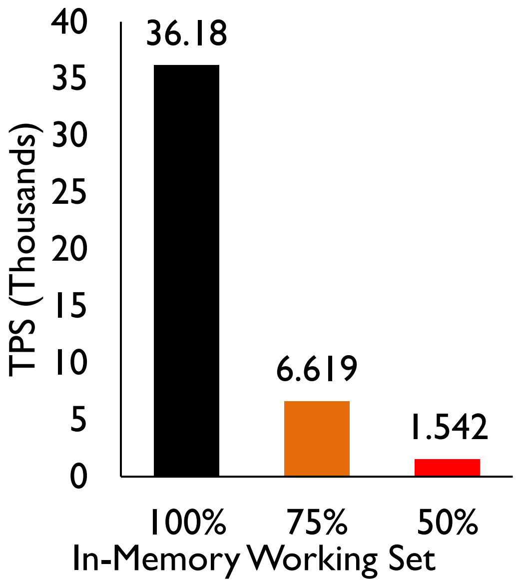
TPC-C on VoltDB

Perform Great Until Memory Runs Out

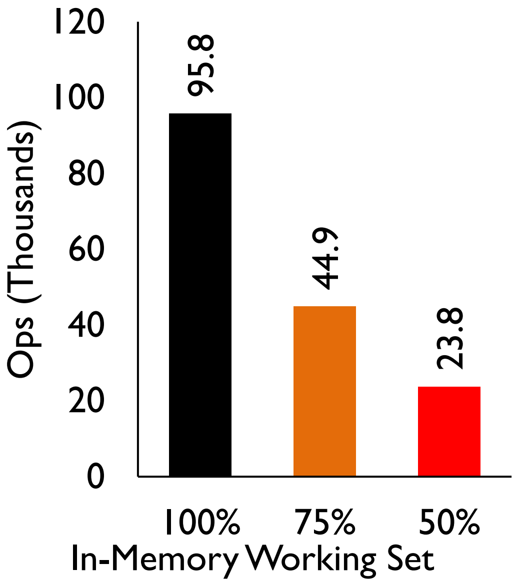


TPC-C on VoltDB

Perform Great Until Memory Runs Out

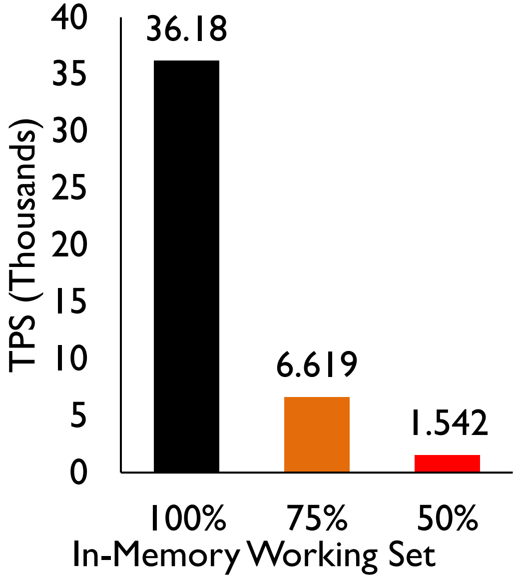


TPC-C on VoltDB

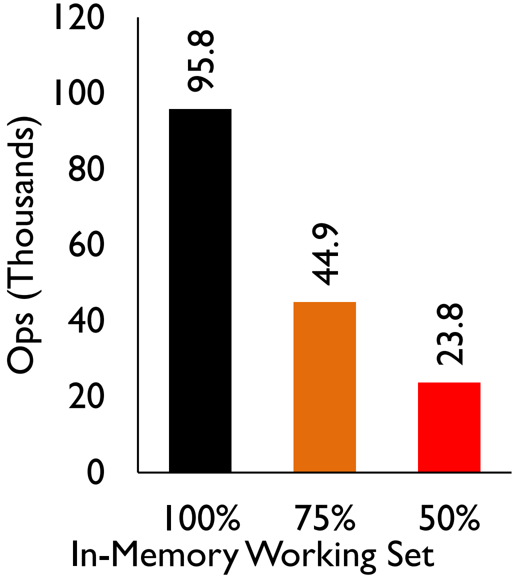


FB Workload on Memcached

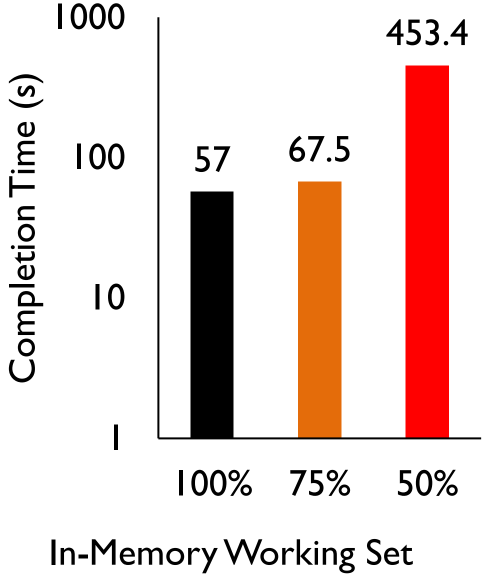
Perform Great Until Memory Runs Out



TPC-C on VoltDB

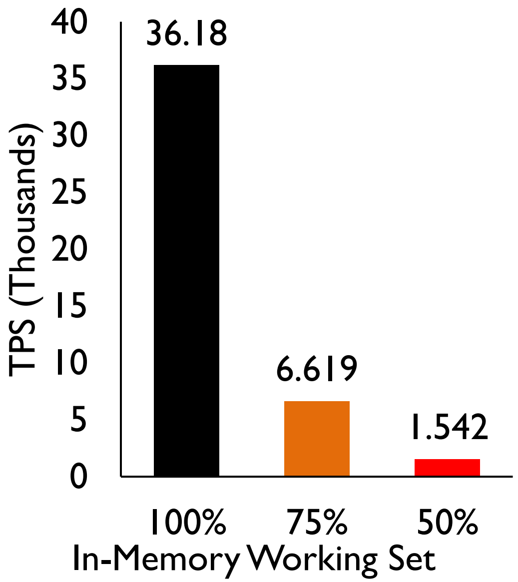


FB Workload on Memcached



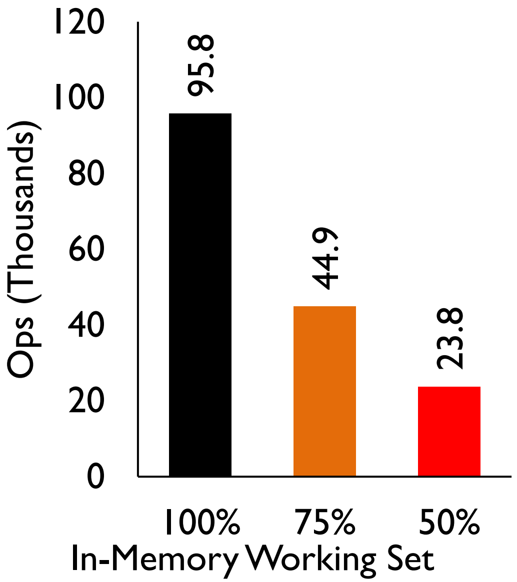
PageRank on PowerGraph

50% Less Memory Causes Slowdown of …



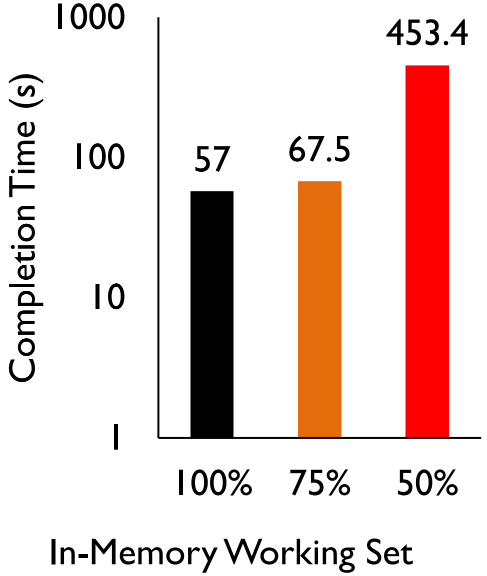
TPC-C on VoltDB

24X



FB Workload on Memcached

4X



PageRank on PowerGraph

8X

Between a Rock and a Hard Place

Underallocation

Leads to severe performance loss

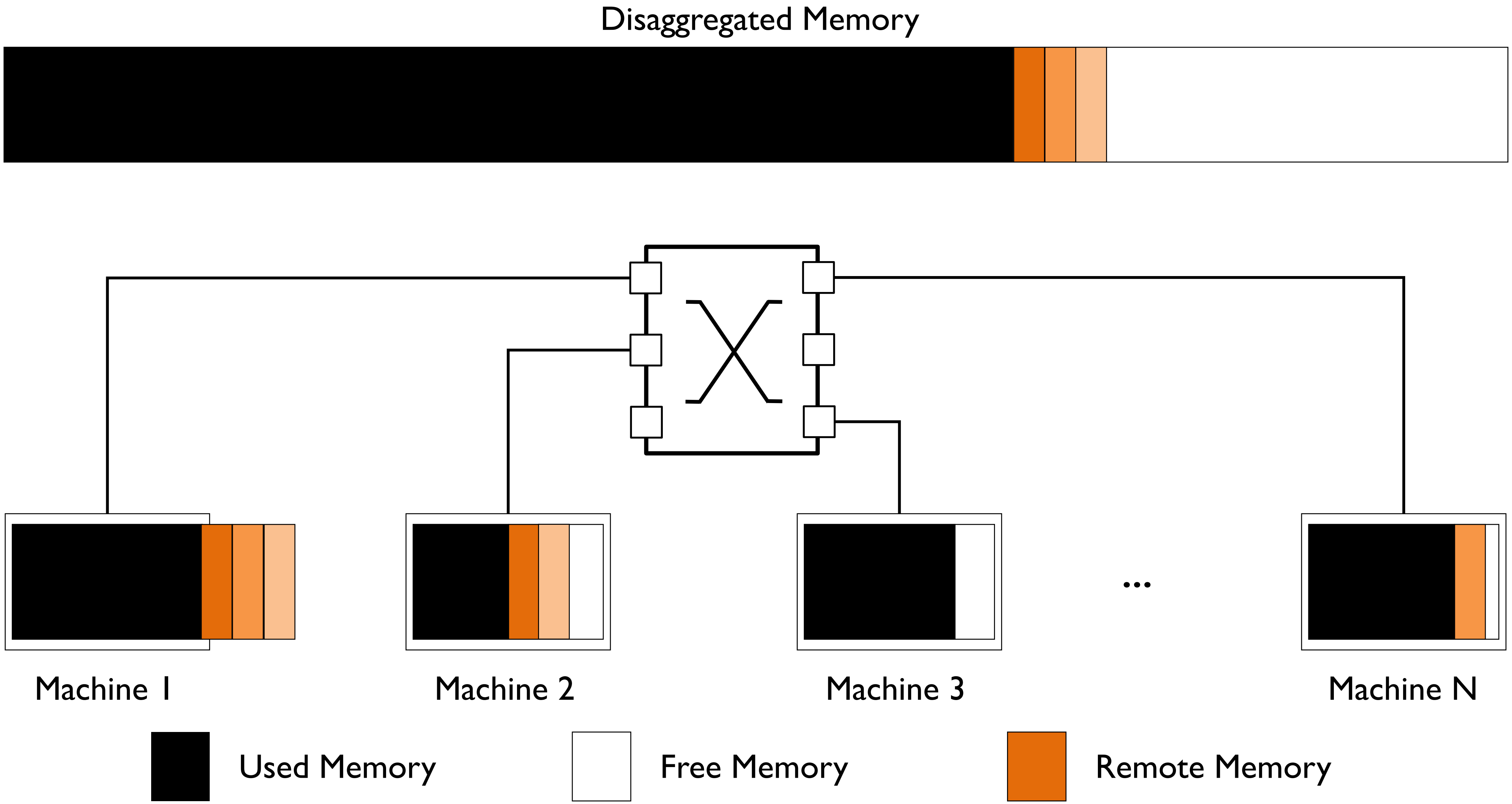
vs.

Overallocation

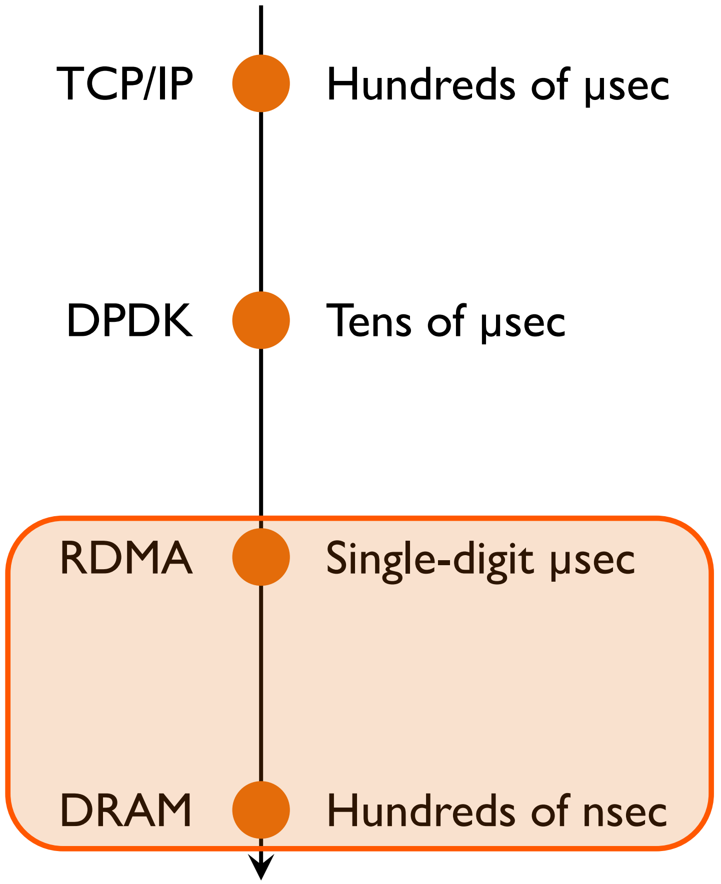
Leads to underutilization

30-50% in Google, Alibaba, and Facebook

Memory Disaggregation



Network is Getting Faster!



time to access a 4KB memory page

What is Practical Memory Disaggregation?

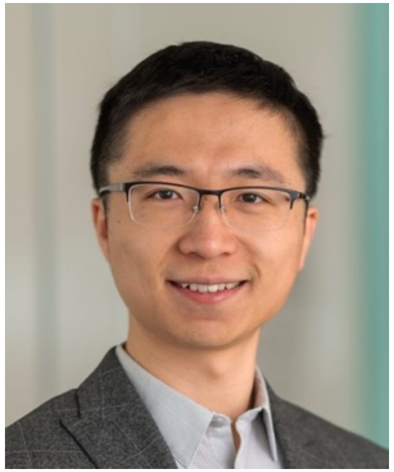
1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. Isolation
6. Resilience
7. Security
8. Generality
9. ...

What is Practical Memory Disaggregation?

1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. Isolation
6. Resilience
7. Security
8. Generality
9. …

Infiniswap

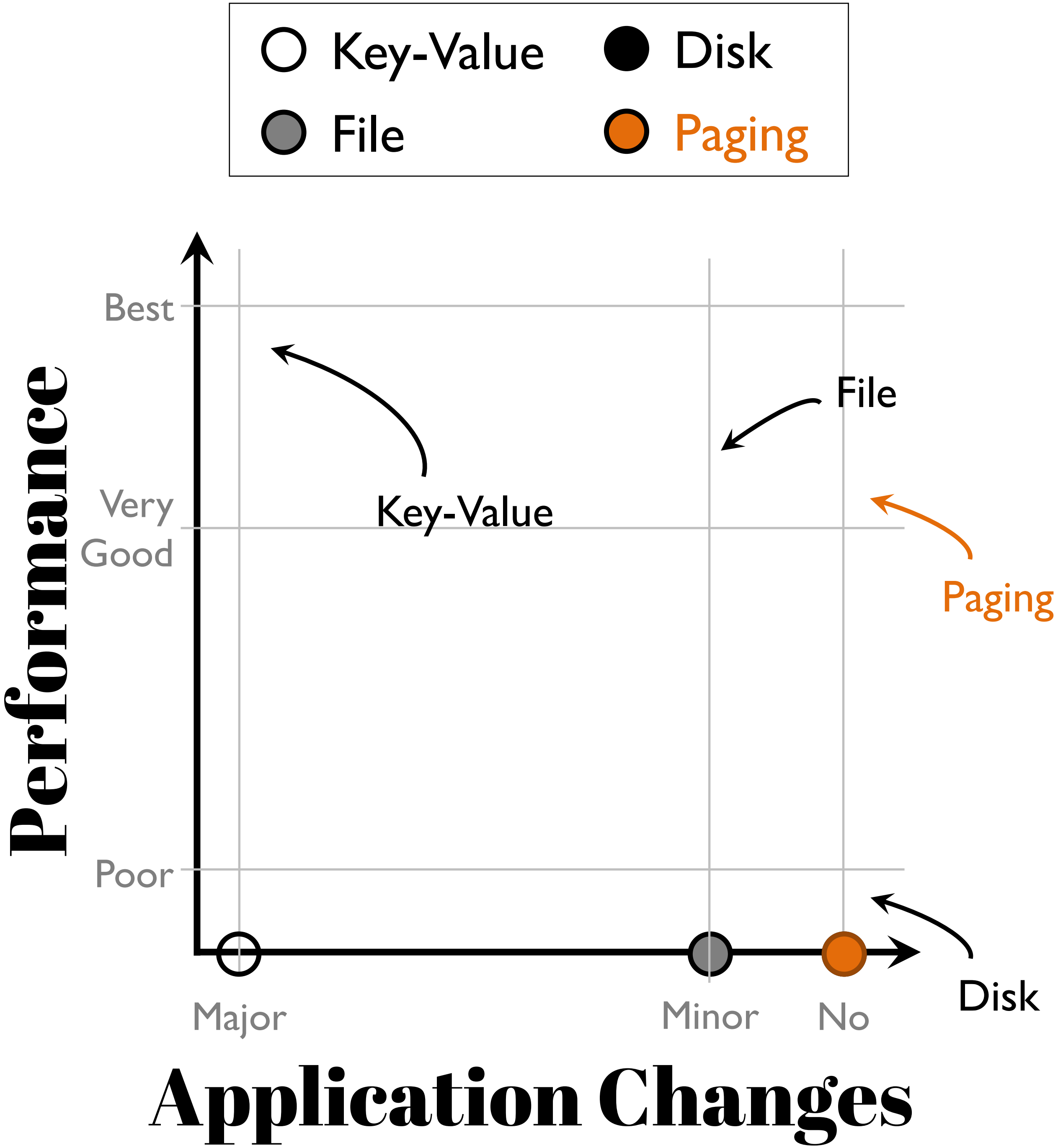
Efficient Memory Disaggregation



w/ Juncheng Gu and many others

NSDI'17

How can we enable any application to leverage disaggregated memory without sacrificing performance?



Remote Memory Paging

Exposes memory across server boundaries

- Scalable
- Efficient
- Fault-tolerant

No changes to

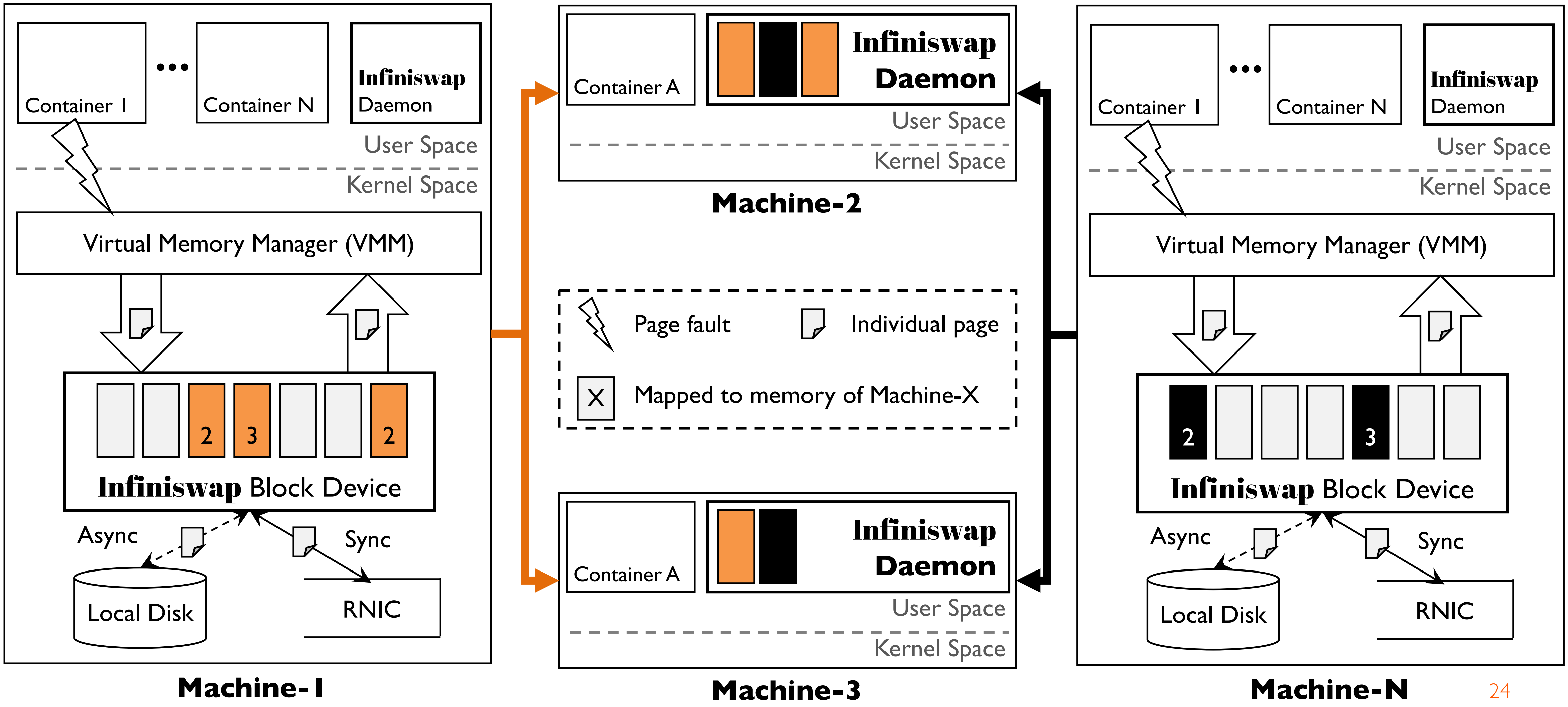
- applications,
- operating systems, or
- hardware

Core Idea

Exposes free remote memory as swap devices in a decentralized manner without affecting remote processes

1. **Infiniswap** Block Device	*Finds free remote memory, maps pages, and provides fault tolerance without any central coordination*
2. **Infiniswap** Daemon	*Proactively evicts remote pages to ensure transparent, best-effort service*

Infiniswap in One Slide



Scalability via Decentralization

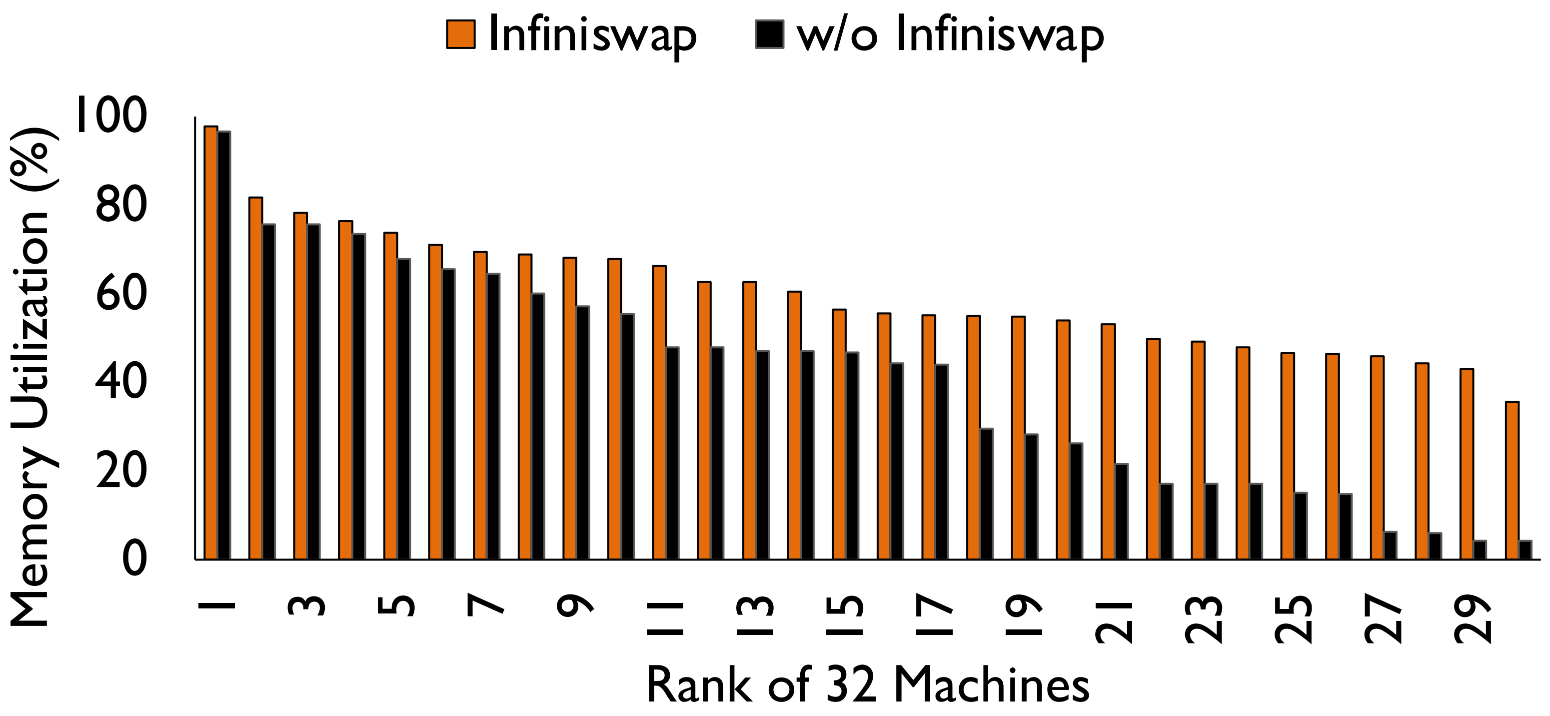
How to find free remote memory in a large cluster?

- **Problem:** Centralized solution can be slow and expensive
- Solution: Power of two choices

How to evict mapped memory?

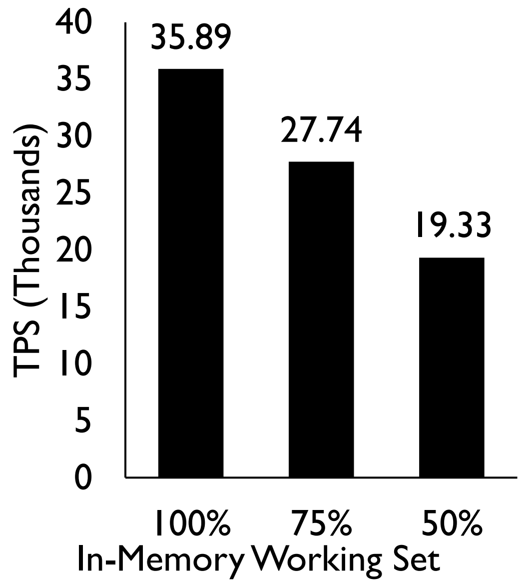
- **Problem:** LRU/LFU is hard because one-sided RDMA bypasses CPU
- Solution: Power of many choices

Higher Efficiency & Better Load Balancing

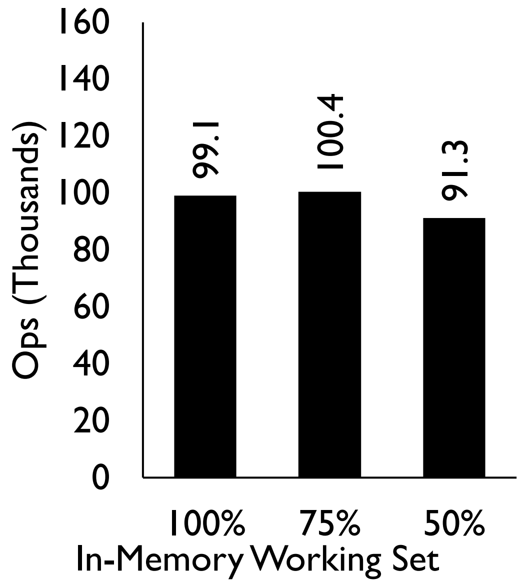


47% Higher Utilization

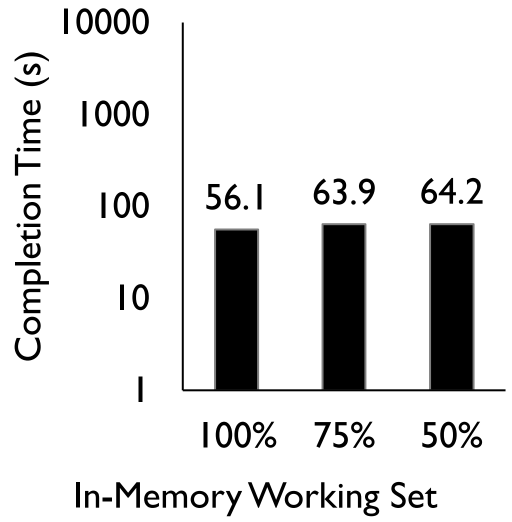
Even on 50% Memory, Slowdown is



TPC-C on VoltDB



FB Workload on Memcached



PageRank on PowerGraph

<2X ≈1X ≈1X

What is Practical Memory Disaggregation?

1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. Isolation
6. Resilience
7. Security
8. Generality
9. ...

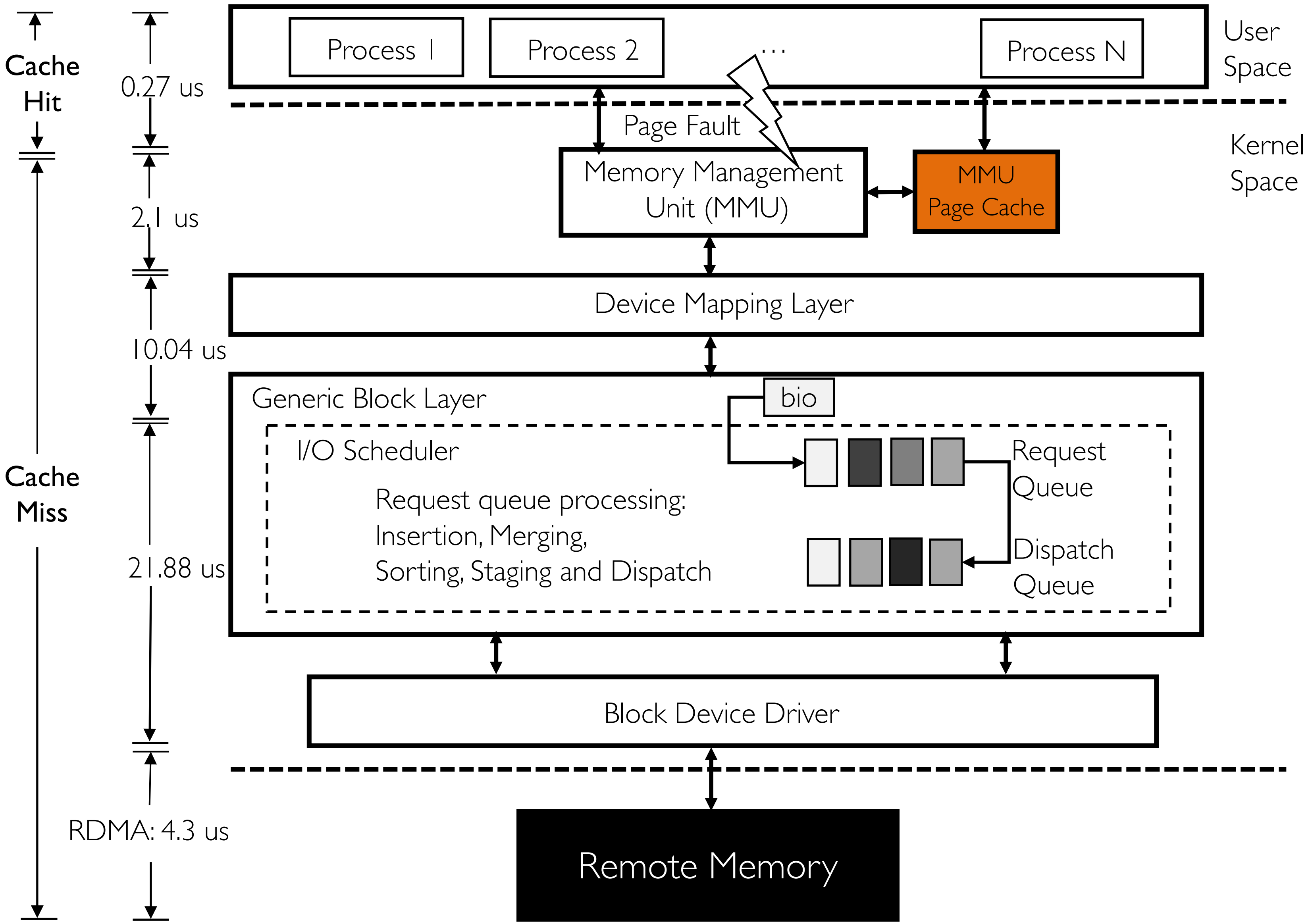
Leap

Effectively Prefetching Remote Memory

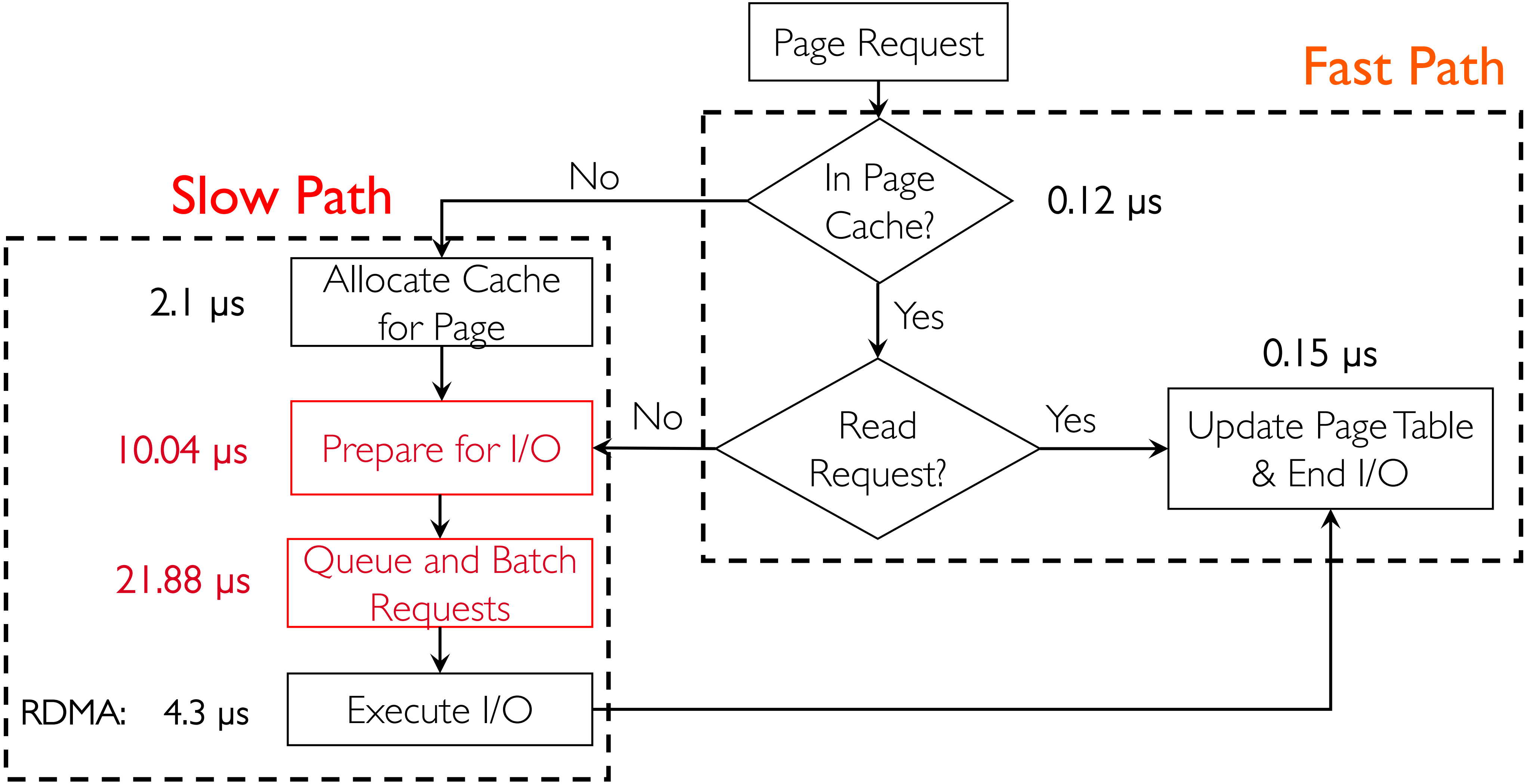
w/ Hasan Al Maruf

ATC'20 Best Paper

Life of a Page



Where Does the Time Go?

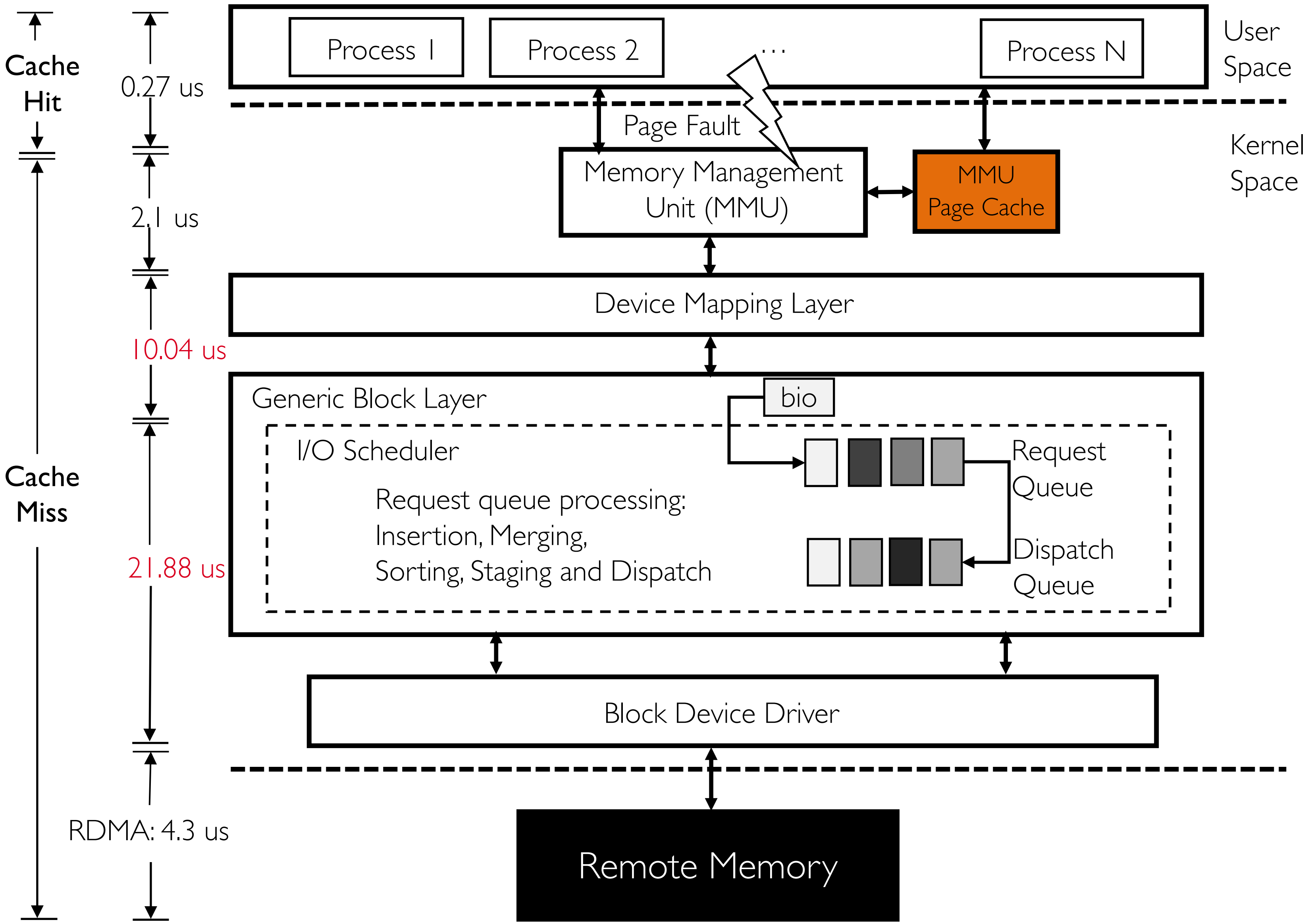


We Need to

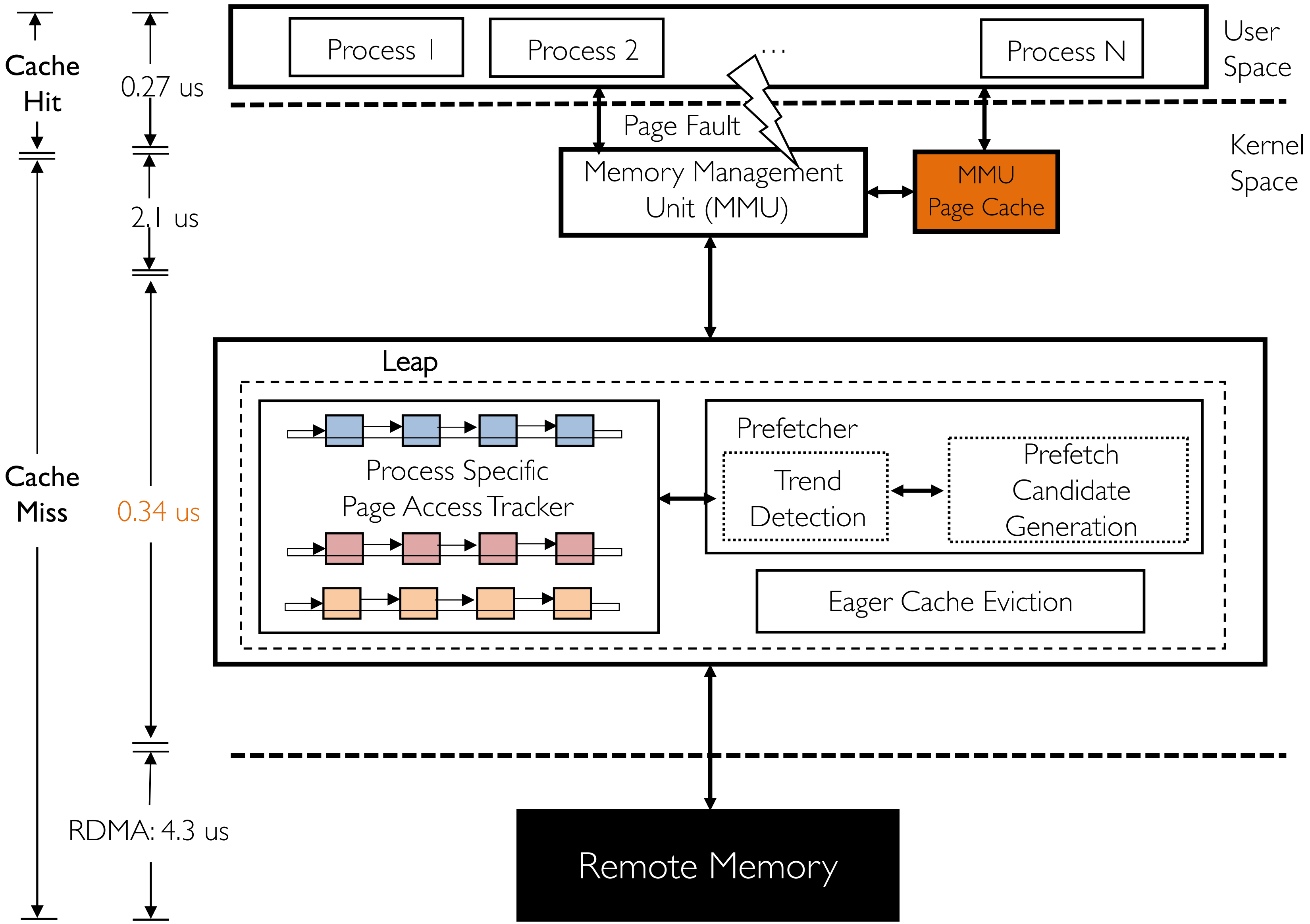
1. **Increase cache hit**
    - Faster path serves more page faults

2. **Reduce the latency of the slow path**
    - Remove block-layer operations unnecessary for RDMA

Life of a Page



Life of a Page w/ Leap



Prefetching in Linux

Reads ahead pages sequentially

Based only on the last page access
- *too aggressive on seq*: cache pollution
- *too conservative off seq*: brings nothing

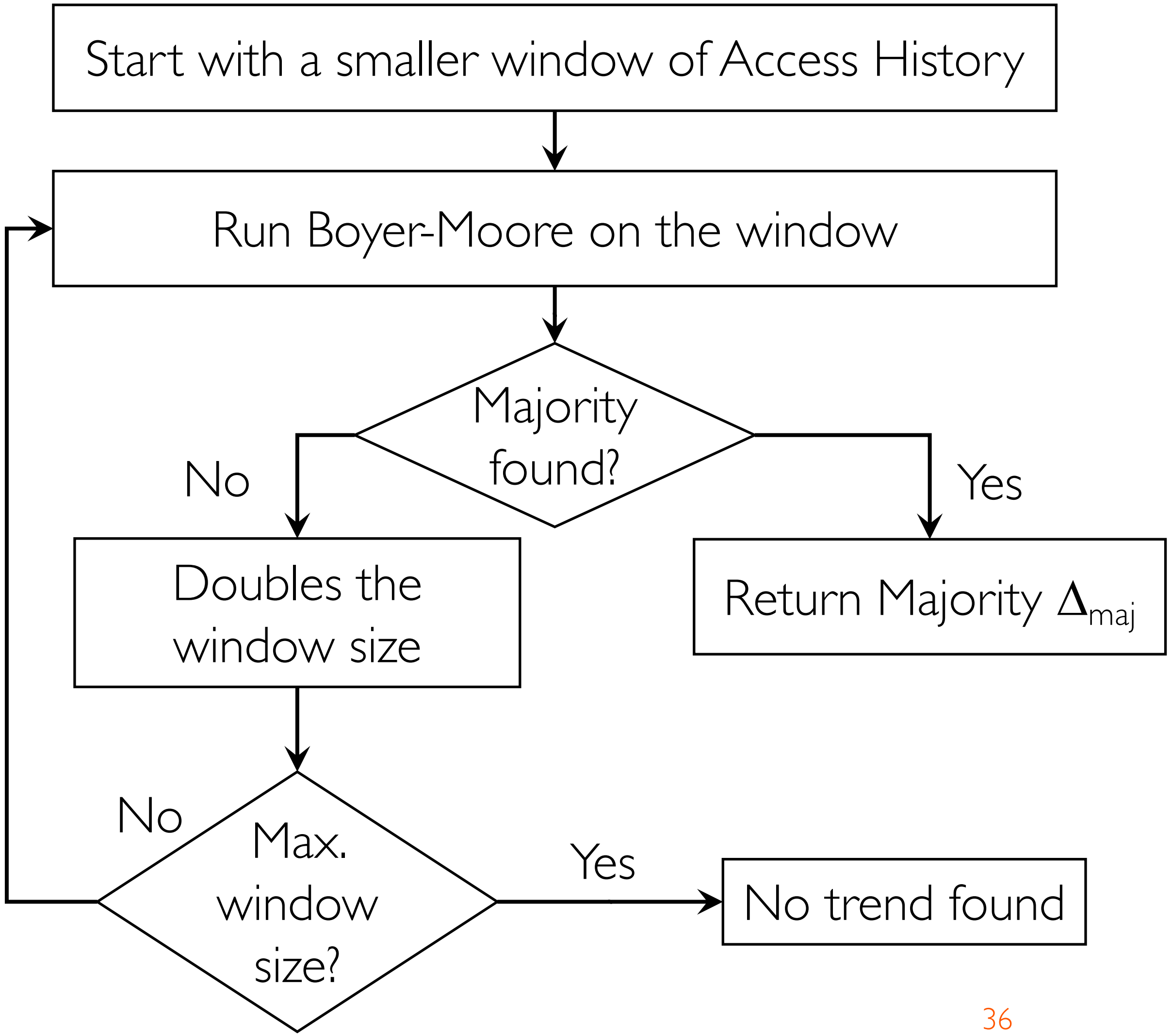
Does not distinguish between processes
Cannot detect thread-level access irregularities

Trend Detection in Leap

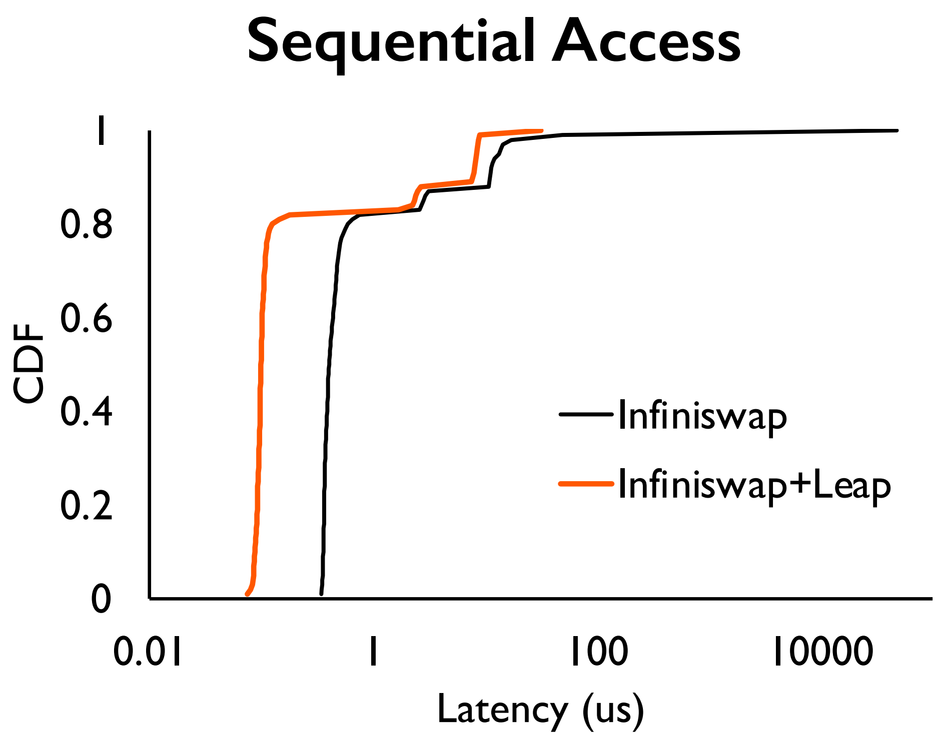
Resilient to short term irregularity

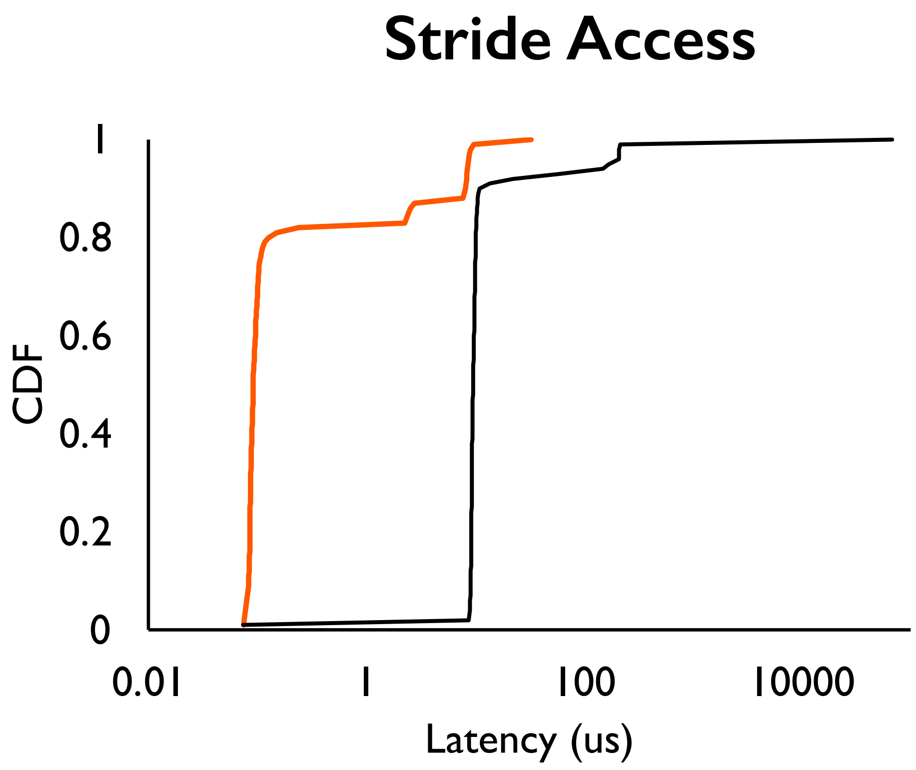
Identifies the majority element in access history

Regular trends can be found within recent accesses



Lowers Remote Page Access Latency by…

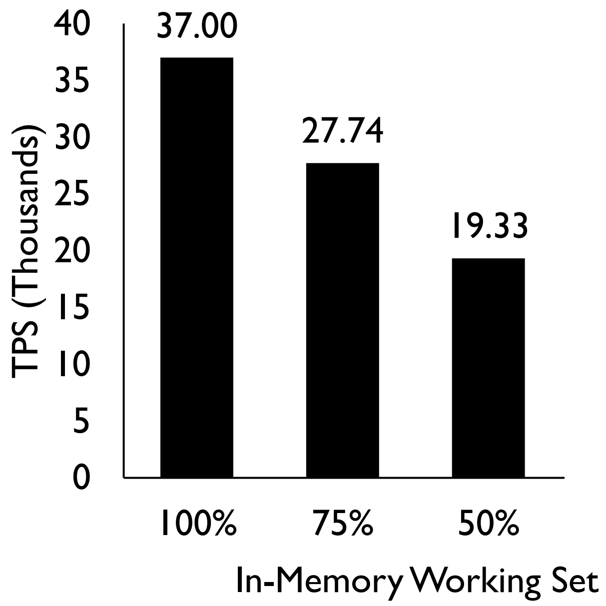




4X **104X**

Performs Great Even After Memory Runs Out

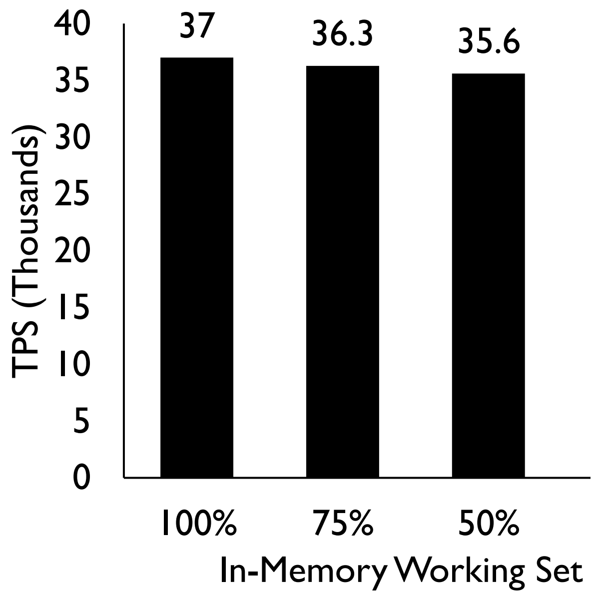
Infiniswap



TPC-C on VoltDB

<2X

Infiniswap + Leap

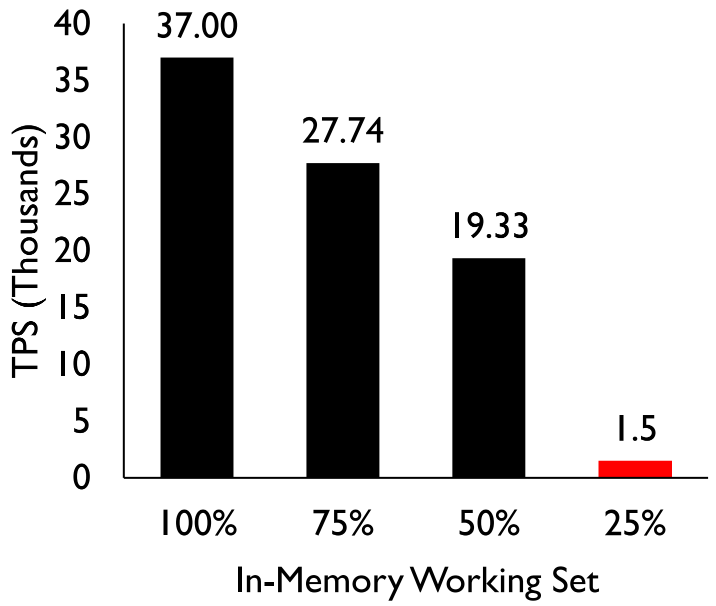


TPC-C on VoltDB

≈1X

Performs Great Even After Memory Runs Out

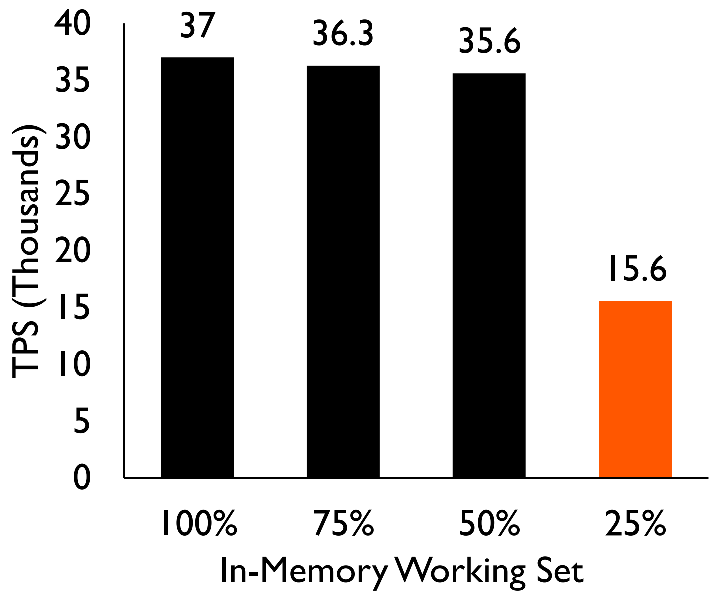
Infiniswap



TPC-C on VoltDB

24X

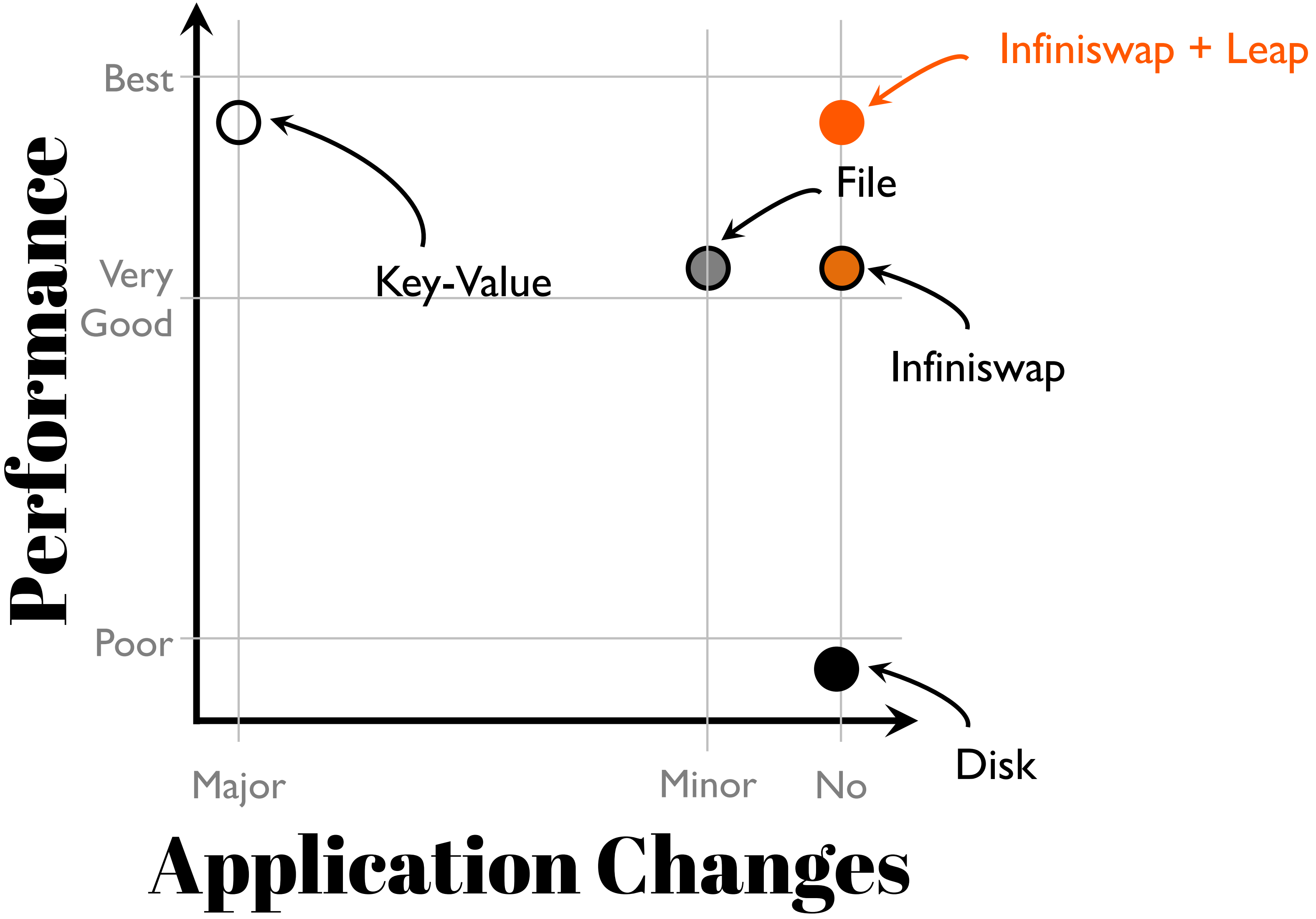
Infiniswap + Leap



TPC-C on VoltDB

2.4X

Applicability & Performance



What is **Practical** Memory Disaggregation?

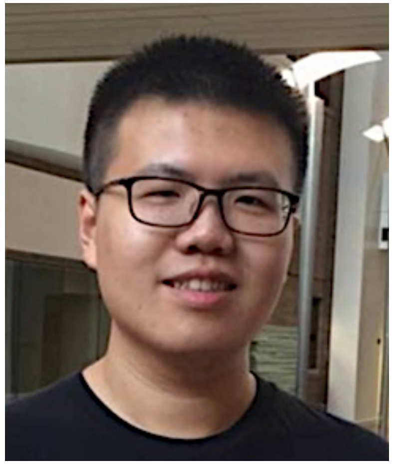
1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. **Isolation** — Justitia
6. **Resilience** — Hydra
7. **Security** — Memtrade
8. Generality
9. …

What is Practical Memory Disaggregation?

1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. Isolation
6. Resilience
7. Security
8. Generality
9. ...

NetLock

Lock Management with Programmable Switches



w/ Zhuolong Yu, Yiwen Zhang and others

SIGCOMM'20

Transactions

Transaction processing needs

- High throughput;
- Low latency; and
- Policy support

Existing approaches

- **Centralized:** low throughput
- **Decentralized:** limited policy support

Network-Assisted Lock Management

Transaction processing needs

- High throughput;
- Low latency; and
- Policy support

Challenges

- Limited memory to store the locks
- Limited functionalities to process the locks and realize the policies

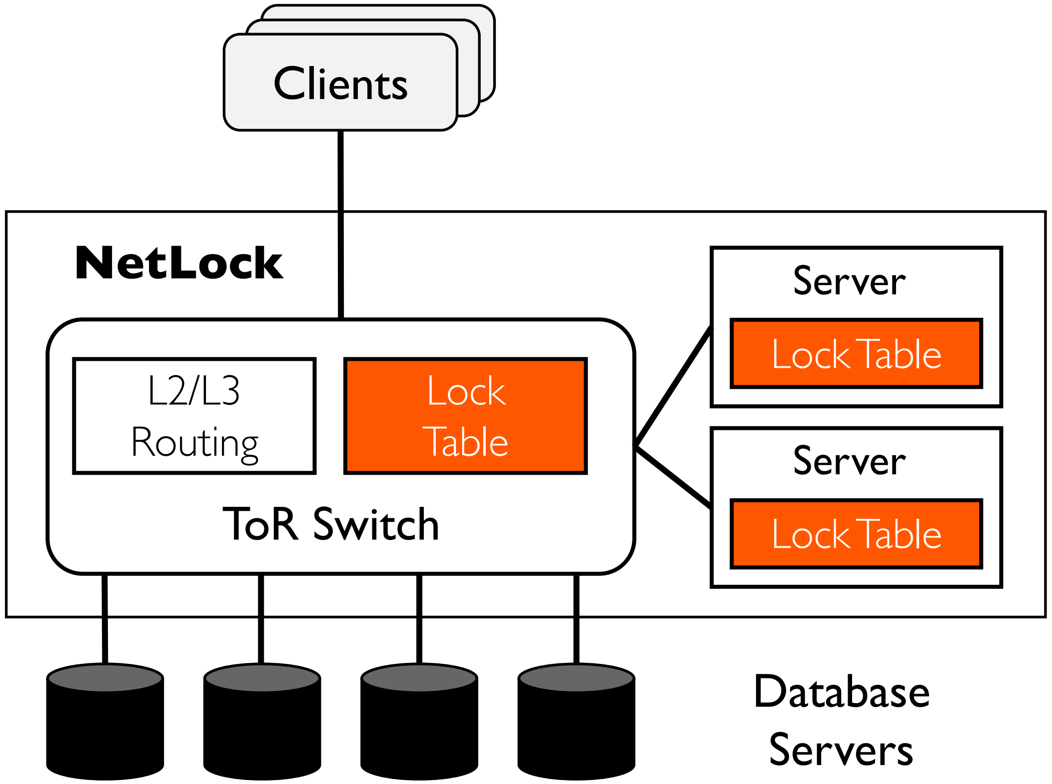
Programmable Switch

NetLock Architecture

NetLock processes lock requests with a combination of switch and servers

- The switch only stores and processes the requests on hot locks
- Servers do the rest

Implemented on a 6.5Tbps Barefoot Tofino switch



Switch Memory Disaggregation

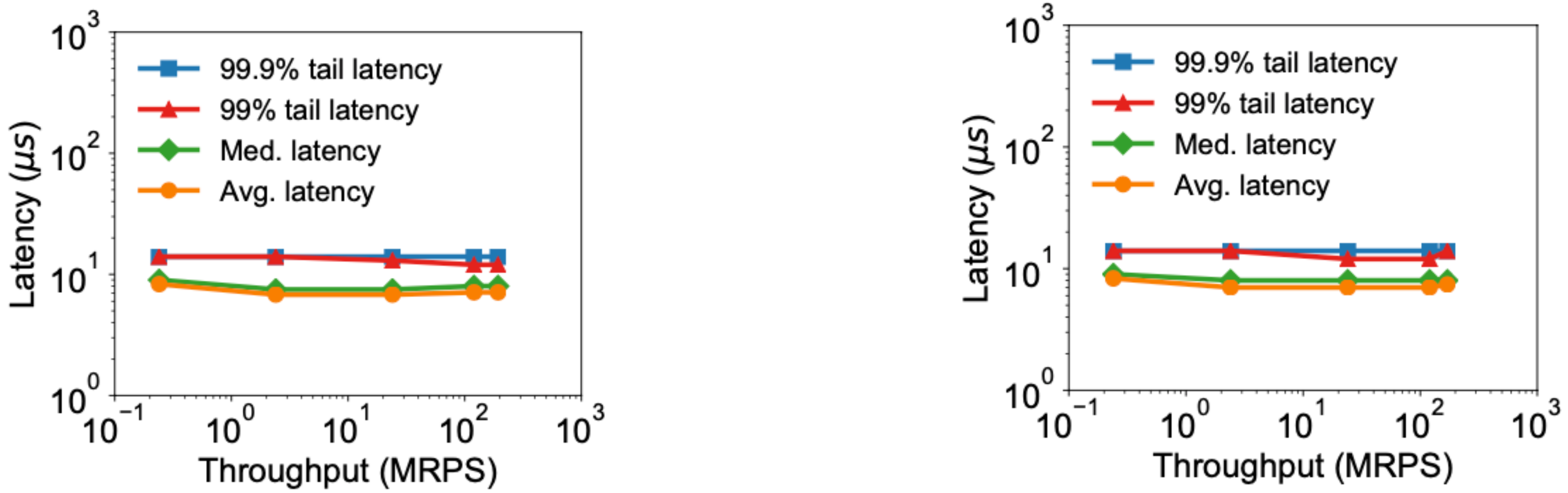
Determine how much switch memory is needed for a target throughput

- Formulated as a fractional knapsack problem
- Depends of expected contention

Handling overflow

- Move locks back and forth between switch and servers

Single μs Latency

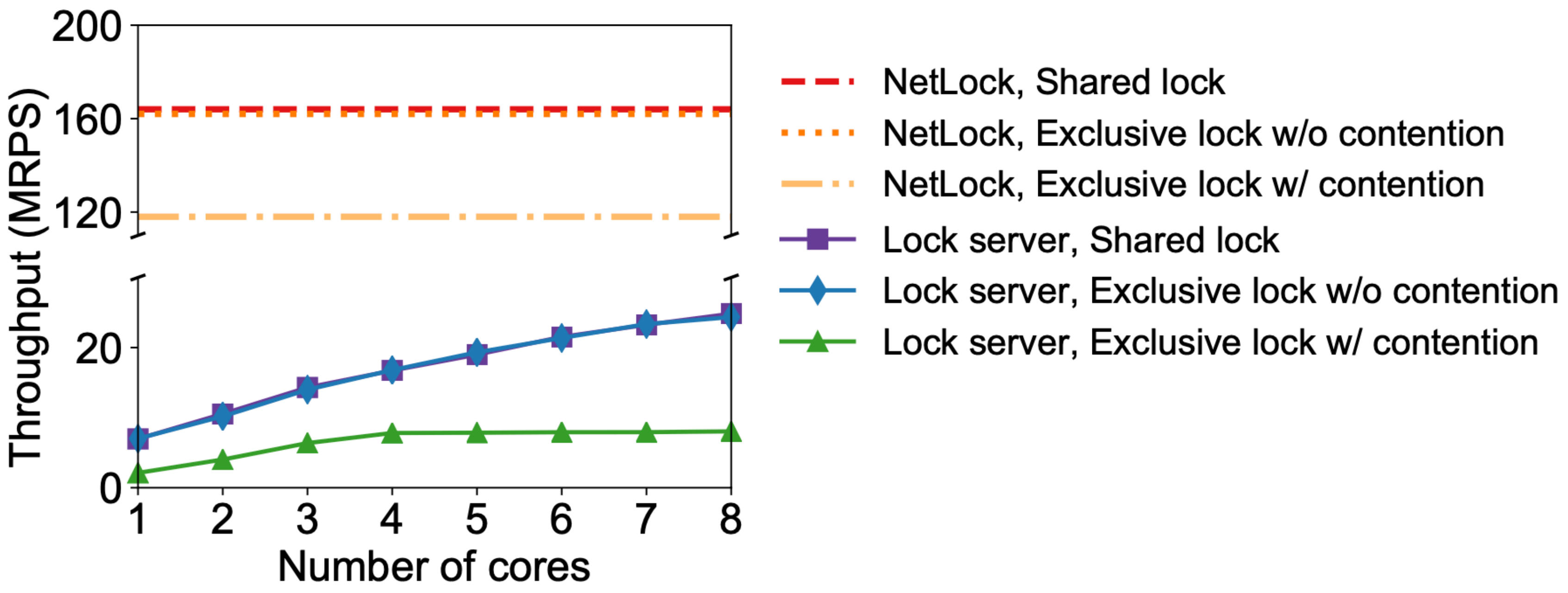


(a) Shared locks.

(b) Exclusive locks w/o contention.

20X lower latency for TPC-C over DSLR

Billions of Locks/Sec



18X higher throughput for TPC-C over DSLR

What is Practical Memory Disaggregation?

1. Applicability
2. Scalability
3. Efficiency
4. Performance
5. Isolation
6. Resilience
7. Security
8. Generality
9. …

Memory Disaggregation

AI/ML Systems

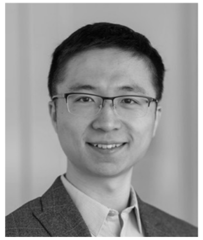
SymbioticLab

Wide-Area Computing

Big Data Systems

Network-Informed Data Systems Design

PhD Students

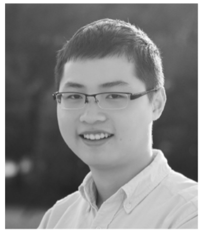


Juncheng Gu

Fan Lai

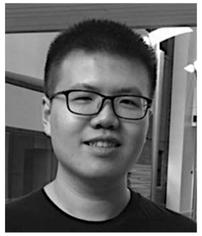
Jiachen Liu

Hasan Al Maruf



Jie You

Peifeng Yu



Yiwen Zhang

Undergraduate & Master's

Chris Chen
Yinwei Dai
Shuoren Fu
Songyuan Guan
Jack Kosaian
Qinye Li
Yang Liu
Yuze Lou
Alexander Neben
Wenting Tan
Yue Tan
Kaiwei Tu
Yuchen Wang
Yujia Xie
Yilei Xu
Jiaxing Yang
Yiwei Zhang
Jiangchen Zhu
Jingyuan Zhu
Xiangfeng Zhu

Collaborators

Aditya Akella
Ganesh Ananthanarayanan
Wei Bai
Vladimir Braverman
Shuchi Chawla
Kai Chen
Li Chen
Asaf Cidon
Yanhui Geng
Ali Ghodsi
Ayush Goel
Robert Grandl
Chuanxiong Guo
Matan Hamilis
Anthony Huang
Anand P. Iyer
Myeongjae Jeon
Xin Jin
Samir Khuller
Tan N. Le
Youngmoon Lee
Li Erran Li
Hongqiang Liu
Zhenhua Liu
Harsha V. Madhyastha
Kshiteej Mahajan
Barzan Mozafari
Linh Nguyen
Aurojit Panda
Manish Purohit
Junjie Qian
Kannan Ramchandran
K.V. Rashmi
Kang G. Shin
Scott Shenker
Brent Stephens
Ion Stoica
Xiao Sun
Muhammed Uluyol
Shivaram Venkataraman
Carl Waldspurger
Hongyi Wang
Jingfeng Wu
Sheng Yang
Bairen Yi
Dong Young Yoon
Zhuolong Yu
Hong Zhang
Junxue Zhang
Yuhong Zhong
Yibo Zhu