Using Spot VMs for a Remote Cache

Qizhen Zhang, Phil Bernstein, Daniel Berger,
Badrish Chandramouli, Vincent Liu, Boon Thau Loo

October 12, 2022

Why Use Remote Memory as a Cache?

- Database workload benefits from a larger cache
  - But VM's physical server has no available memory
  - Or workload changes periodically
- Data centers have lots of unused memory
  - Bin packing of VMs on a server leaves empty fragments
- Faster datacenter networks → disaggregated memory is coming

Unallocated Memory in Azure Across Clusters and time

- Median – 46%
- 10th percentile – 37%
- 1st percentile – 28%
- Daily peak-to-trough is 2x

- Extreme case is *stranded memory*, on servers with no available cores

Stranded Memory

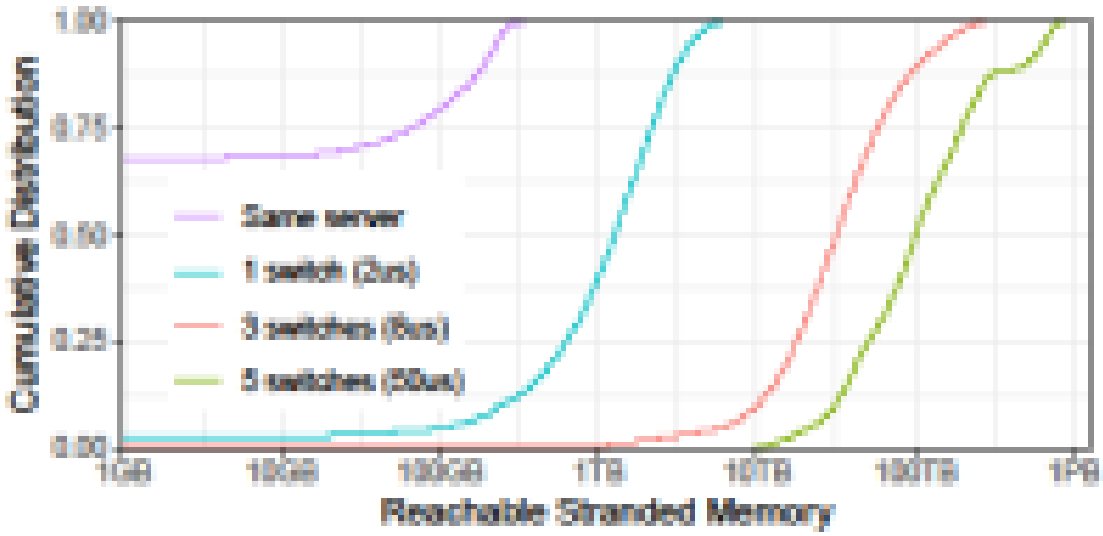
Fraction of memory stranded

- Median cluster, 8% stranded
- 10th percentile, ≥ 16% stranded
- 1st percentile, ≥ 23% stranded

Stranding duration

- 75th percentile – 22 minutes
- Median – 13 minutes
- 25th percentile – 6 minutes

Location



Outline

✓Motivation

- Redy, a remote-cache manager
  - Optimizing for throughput vs. latency
  - Using spot VMs
- Case study – FASTER key-value store
- CompuCache - Adding stored procedures

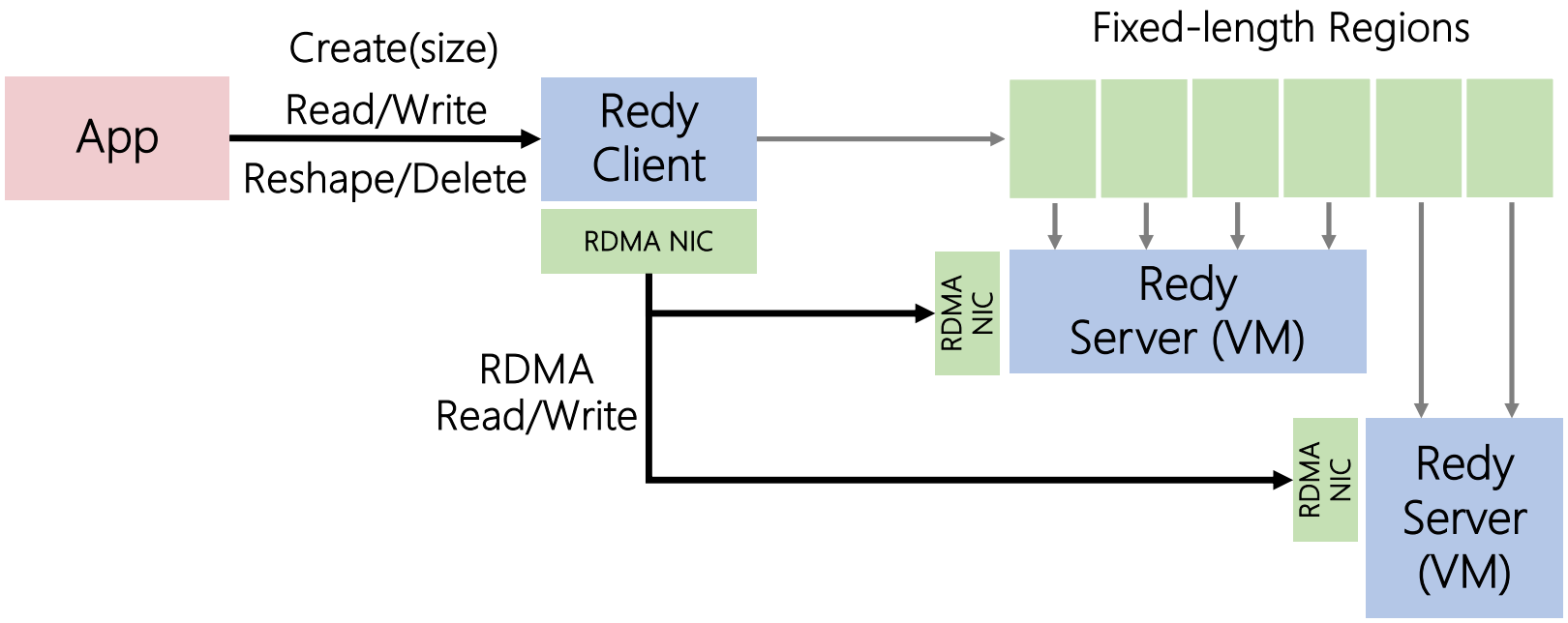
Use RDMA for Low-Latency Remote Access

Hardware latency (your mileage may vary)

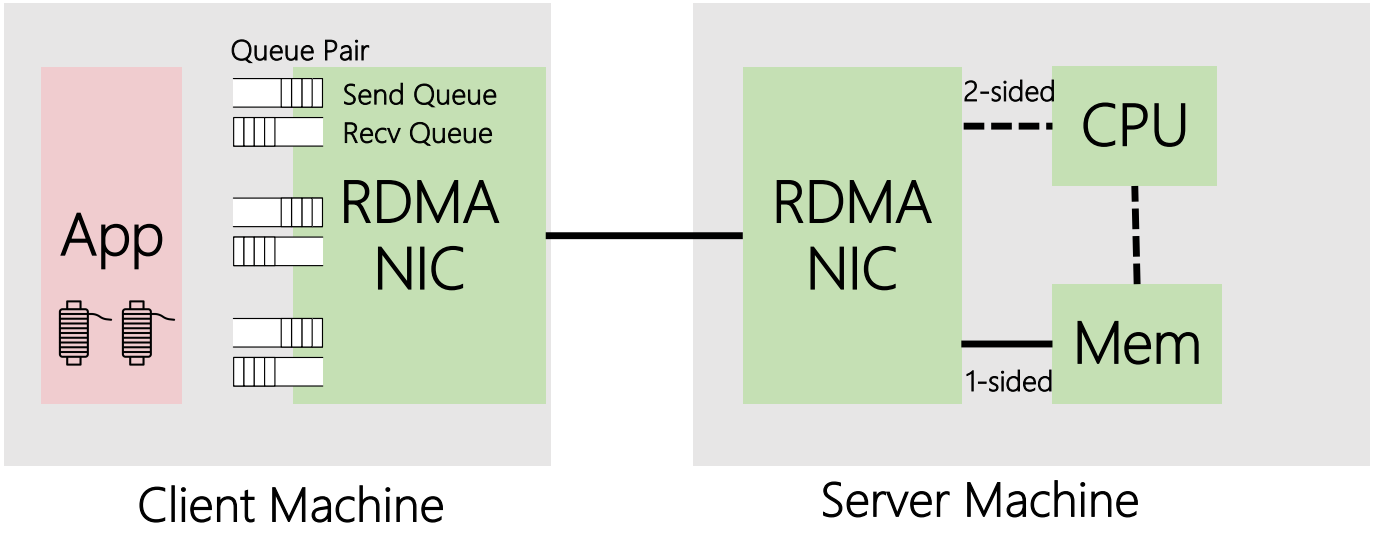
- Level 0 – DRAM 0.2 μs
- Level 1 – RDMA 2.0 μs
- Level 2 – SSD 100 μs

RDMA-accessible Cache

- Front End: an easy-to-use and general device abstraction
- Back End: efficient remote memory accesses via RDMA



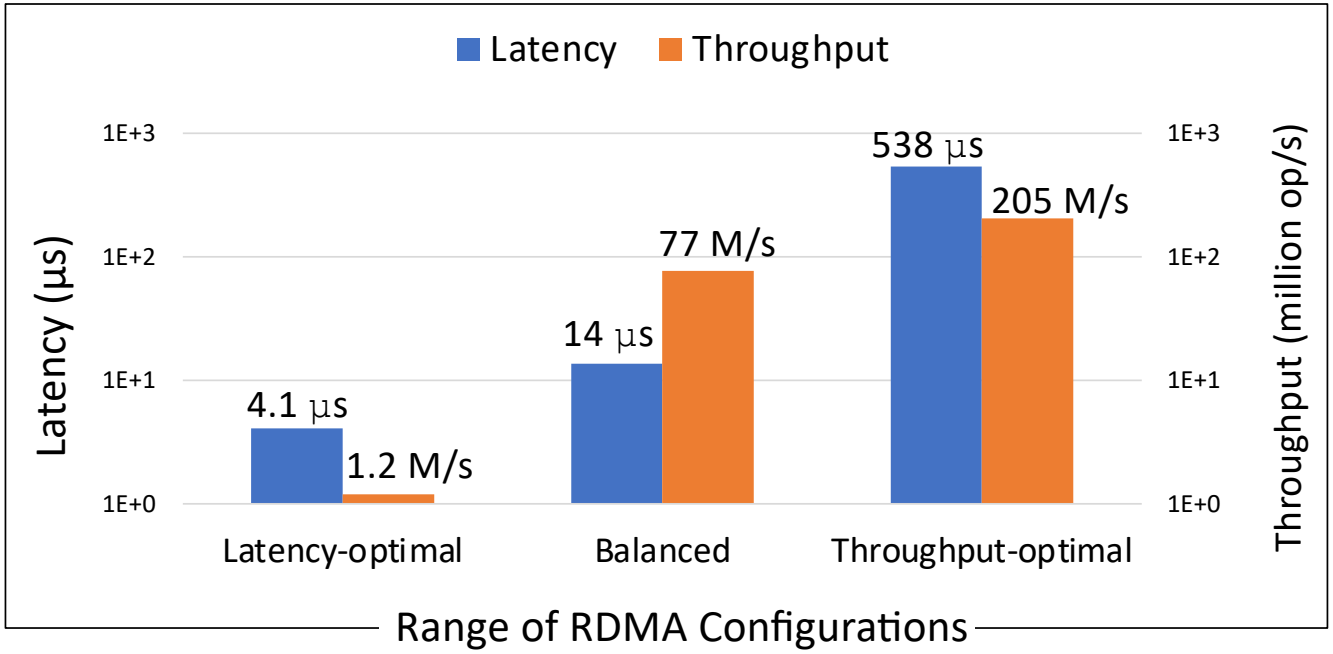
RDMA


Queue Pair
Send Queue
Recv Queue
App
RDMA NIC
RDMA NIC
2-sided
CPU
Mem
1-sided
Client Machine
Server Machine

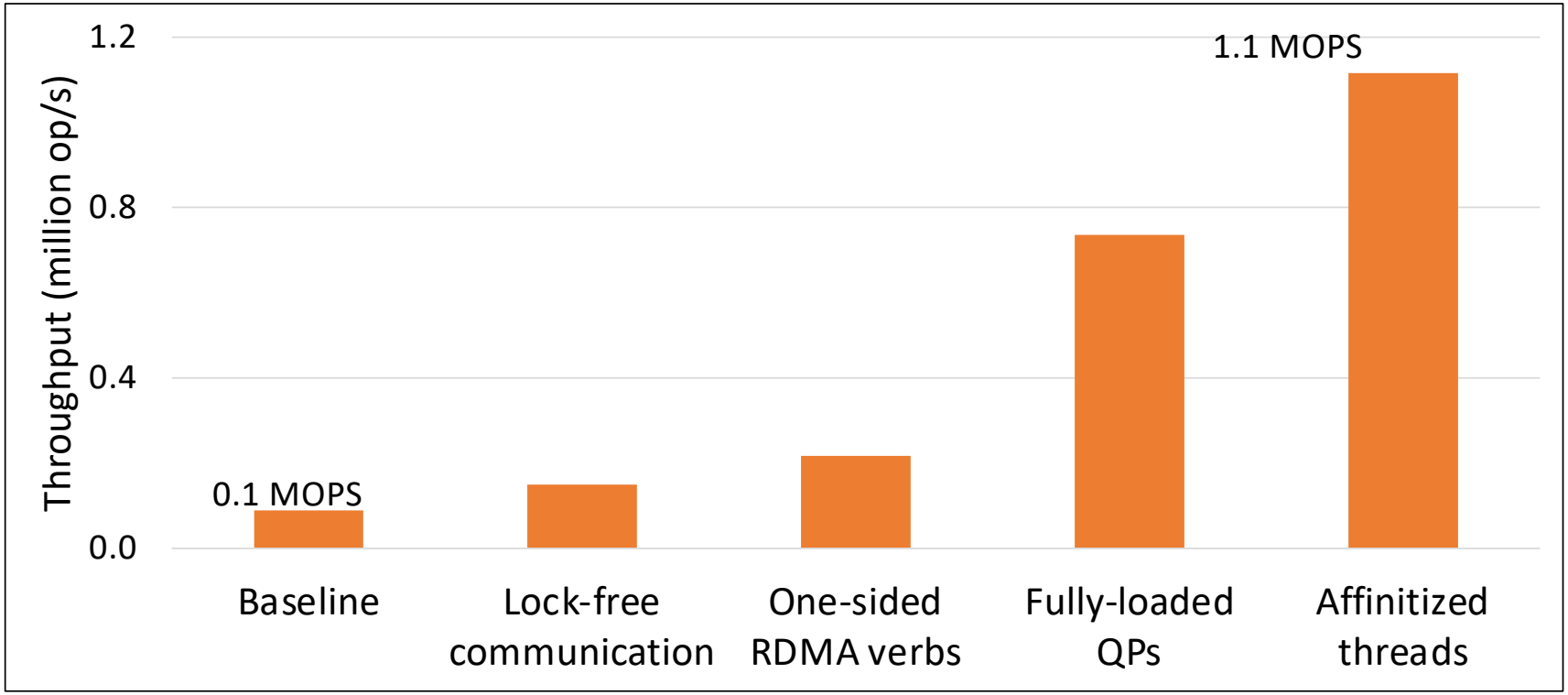
Research Challenge

Tuning RDMA performance

- Many RDMA tuning knobs for best latency or throughput
- Parallelization, asynchrony, batching, 1-sided/2-sided, …
- Optimal choice depends on record size, VM size, SLO, network configuration, …
- Experiment shows YCSB-like workload for 8-byte payloads

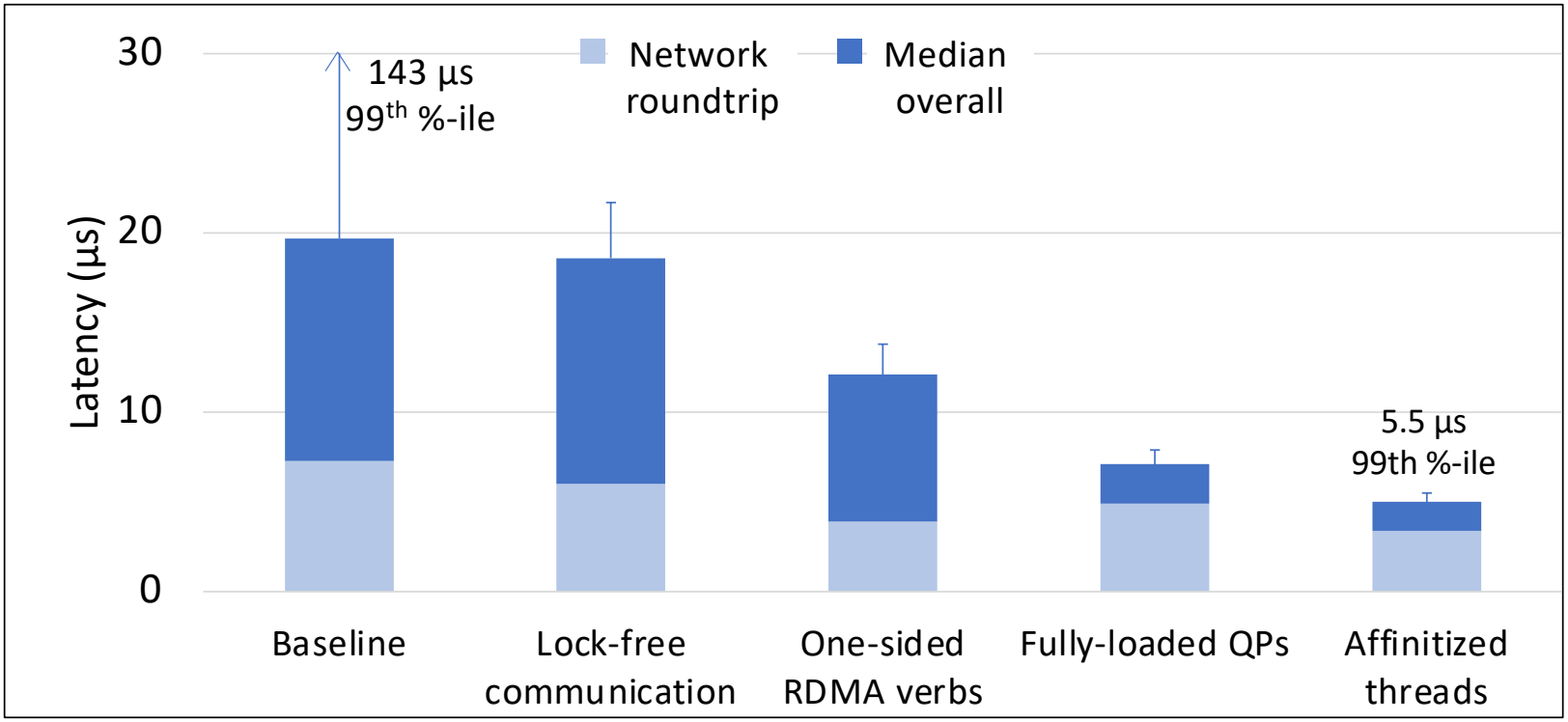


Throughput Benefit of Static Optimizations



- One client thread, one server thread, 8-byte read/write, no batching

Latency Benefit of Static Optimizations



- One client thread, one server thread, 8-byte read/write, no batching

Workload-dependent Optimization

- Primary determinants of Redy performance

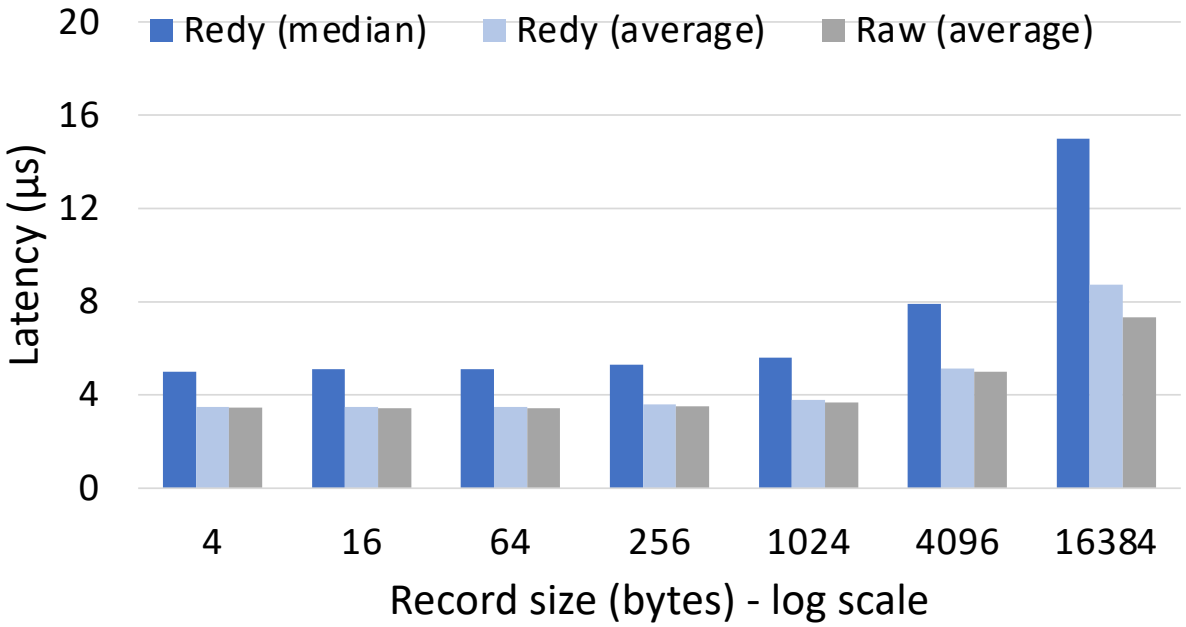
Variable	Description	Lower Bound	Upper Bound
c	# client threads	1	Client cores
s	# server threads	0	c
b	# requests per batch	1	⌈4KB / record-size⌉
q	# in-flight operations	Static opt	NIC spec

- Optimize these parameters to satisfy a given SLO.

Experiments

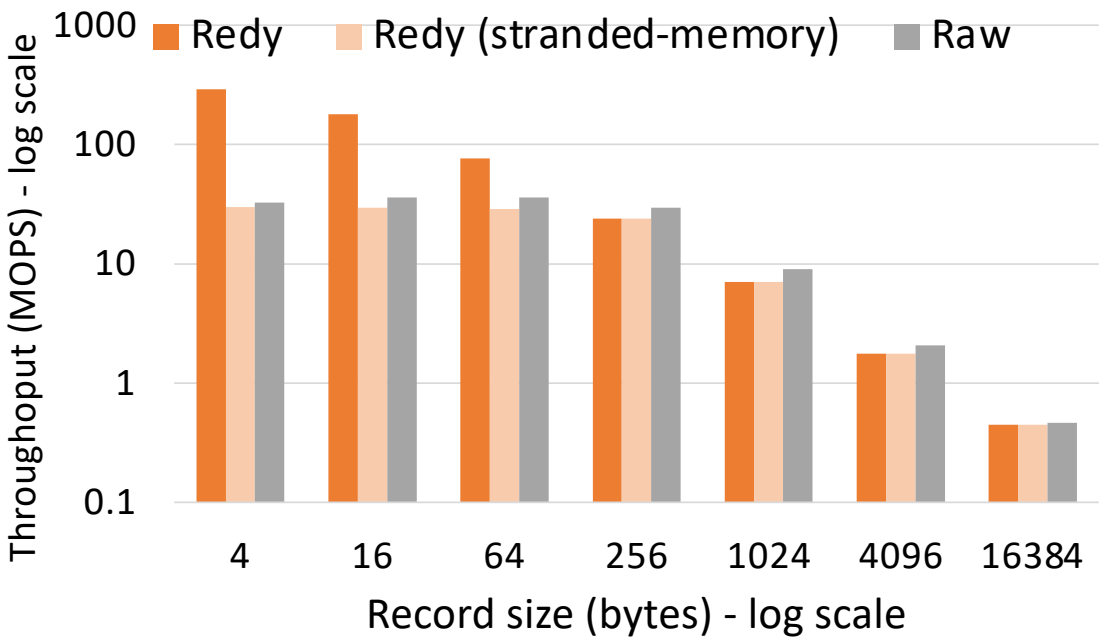
- Azure HPC Standard_HB60rs VMs.
- Each VM has 60 vCPUs, with 2.0 GHz AMD EPYC 7551
  - 228 GB of memory
  - 700 GB Azure premium SSD
- Mellanox ConnectX-5 NIC, supporting 100 Gb/s EDR Infiniband
- Windows 2019 Datacenter

Latency-Optimized Configuration (Reads)



- Reads are close to raw network latency of 3-4μs
- Small writes are a bit faster by in-lining them in the request (not shown)

Throughput-Optimized Configuration (Reads)



- For 16-byte records, throughput is 10x raw.
- Batching benefit stops at 256-bytes, the RDMA minimum packet size.
- Writes throughput is similar.

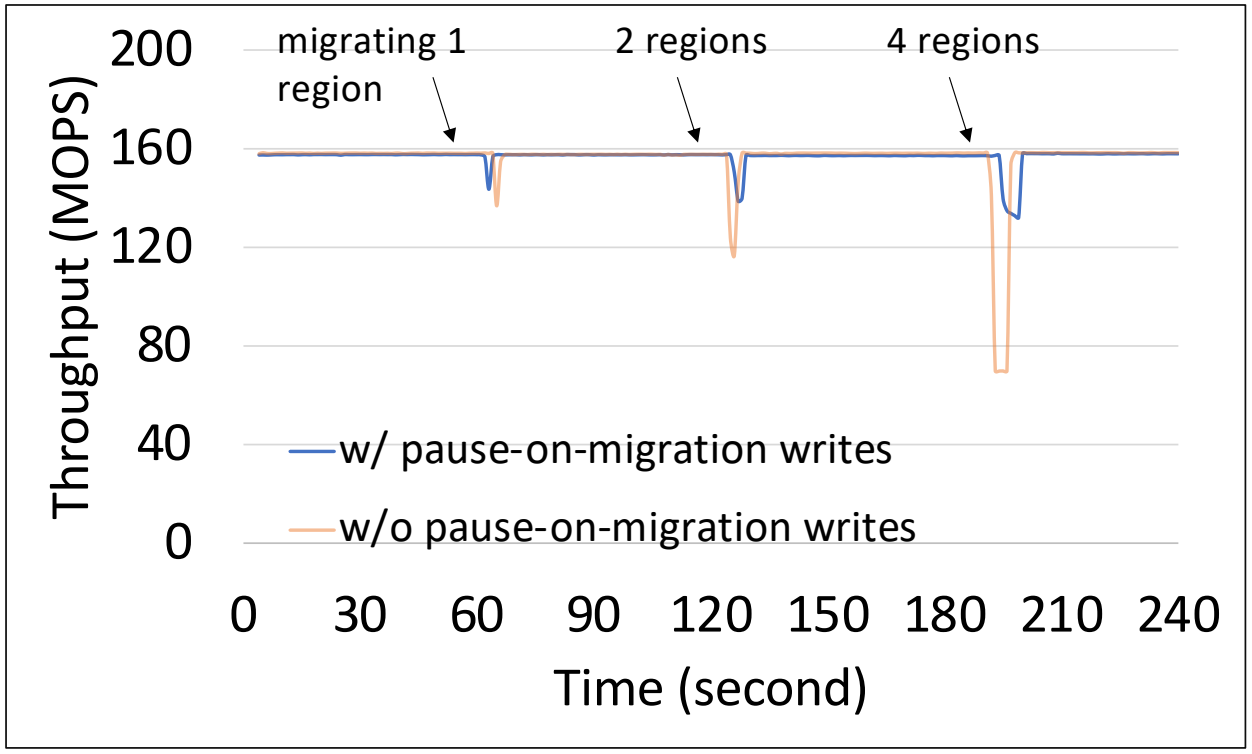
VM Allocation

- Cache manager chooses VMs to satisfy memory and core requirements
- Can use spot instances
  - Requires cache migration on short notice
- Periodically check for cheaper VMs
  - If successful, allocate and migrate

Dynamic Memory Management

- Cache failure – allocate a new cache [and populate it from a checkpoint]
- Spot instance reclamation – migrate to a newly allocated cache
  - Use bandwidth-optimized connection from new to old
  - Use one-sided reads to migrate the content
- Optimizations
  - The application reads the old cache during migration
  - Migrate region-by-region and stop writes only to the region being migrated

Optimizing Region Migration



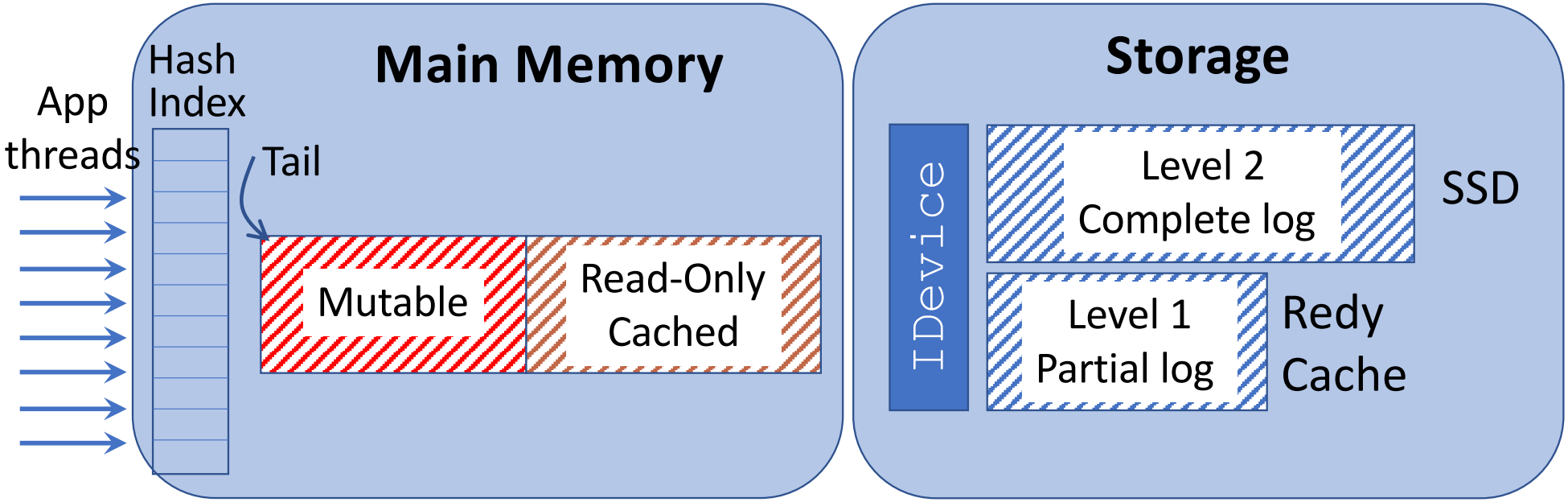
- Successively migrate 1GB regions

Outline

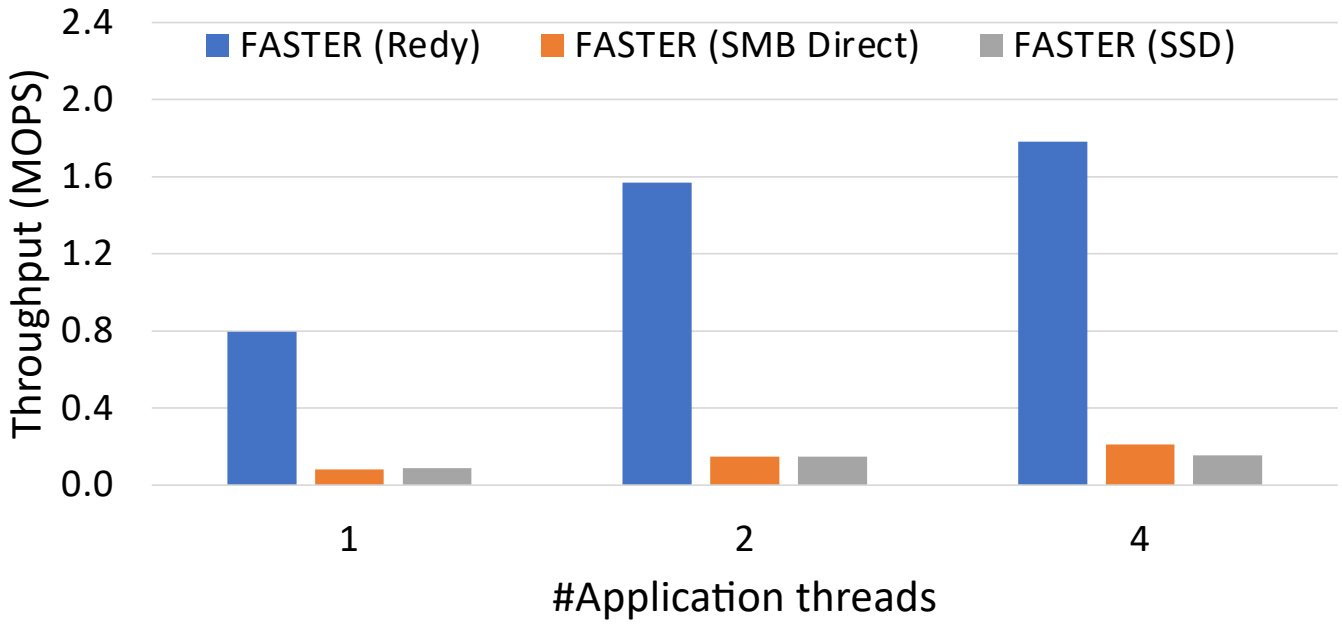
- ✓Motivation
- ✓Redy, a remote-cache manager
  - Optimizing for throughput vs. latency
  - Using spot VMs
- Case study – FASTER key-value store
- CompuCache - Adding stored procedures

FASTER's Multi-tier Storage

- The database is a log. Top storage tier stores the entire database.
- Lower tier replicates a tail of next higher tier.
- FASTER services cache misses from the lowest tier that has the data.



FASTER vs. SSD and SMB Direct



- FASTER local memory is 1GB
- 24-byte records
- Random reads from ~6GB database
- All devices can hold the complete log
- Redy uses batching

Outline

- ✓Motivation
- ✓Redy, a remote-cache manager
  - Optimizing for throughput vs. latency
  - Implementation
- ✓Experiments
- ✓Case study – FASTER key-value store
- CompuCache - Adding stored procedures

CompuCache

- Extend Redy with stored procedures

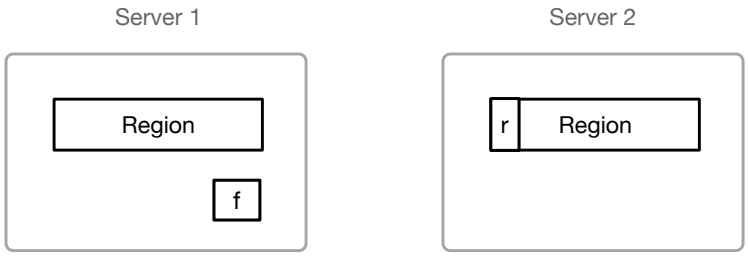
- Server-side pointer chasing
  - Apps know cache addresses, but chasing pointers requires physical addresses
  - Solution: LocalTranslator

```
l_addr, l_size ← Translate(c_addr, c_size)
```

physical address cache address

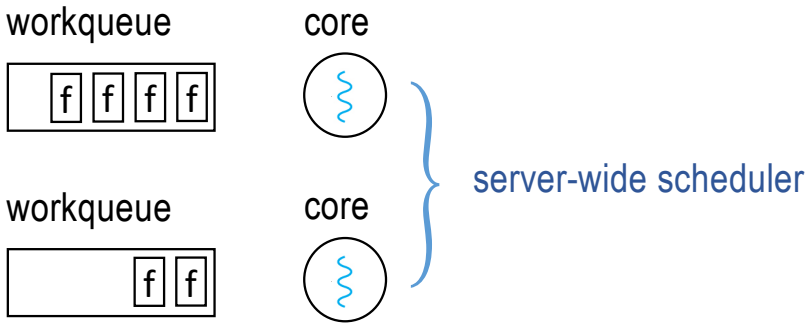
CompuCache Out-of-Bounds Exceptions

- A sproc may VM boundaries
  - Data Shipping - Flow the input data with Dflow: `l_addr ← DFlow(c_addr, c_size)`
  - Function Shipping - Flow the sproc function with Fflow: `FFlow(c_addr, c_size)`
  - Stop and return



CompuCache Execution

- Transport is eRPC over DPDK or RDMA
  - Batches all operation requests
  - Dynamically chooses batch size of responses
- Server uses one thread per core
  - Polls for requests and executes them on the same thread
  - Server-wide scheduler load-balances by moving requests between threads



CompuCache Execution (cont'd)

- On DFlow, the request becomes inactive and resumes after data transfer
- FFlow dispatches request to another thread

Server Migration

- Client maintains mapping for LocalTranslator
- When a server VM is reclaimed, client allocates new VM(s)
- For each region
  - Client pauses reads and writes for the region
  - Migrates the region
  - Updates LocalTranslator mapping
  - Broadcasts mapping to servers
  - Resumes reads and writes for the region
  - Server forwards future stale DFlow and FFlow requests to the new region
- After server is migrated, it sends remaining request to new VMs

Conclusion

- A remote cache is a must for data management
  - It uses memory that currently goes to waste
  - With RDMA, it offers big performance gains
  - Use stored procedures for pointer-chasing
- Remaining work
  - Integrate with Azure's VM allocator

Redy – VLDB 2022
https://arxiv.org/ftp/arxiv/papers/2112/2112.12946.pdf

CompuCache – CIDR 2022
https://www.cidrdb.org/cidr2022/papers/p31-zhang.pdf