Mitsume: an Object-Based Remote Memory System

Shin-Yeh Tsai, Yiying Zhang

One-Sided Remote Memory/NVM

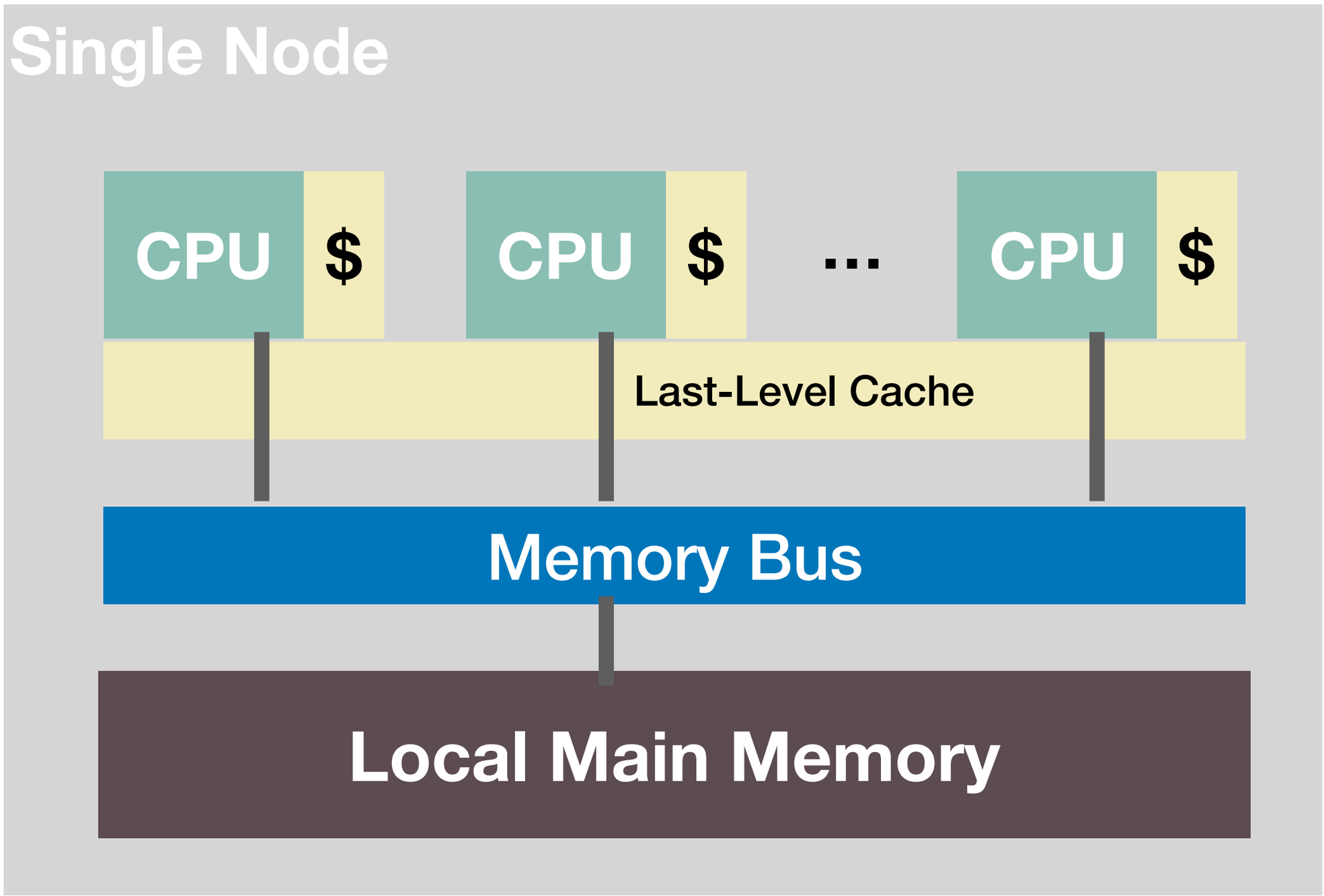
- One-sided devices
  - Devices without (general) computation power
  - Can only be read and written to with limited, low-level APIs
  - Disaggregated memory
  - NVMe over fabrics
- Cheap, low energy

Remote Memory Challenges and Opportunities

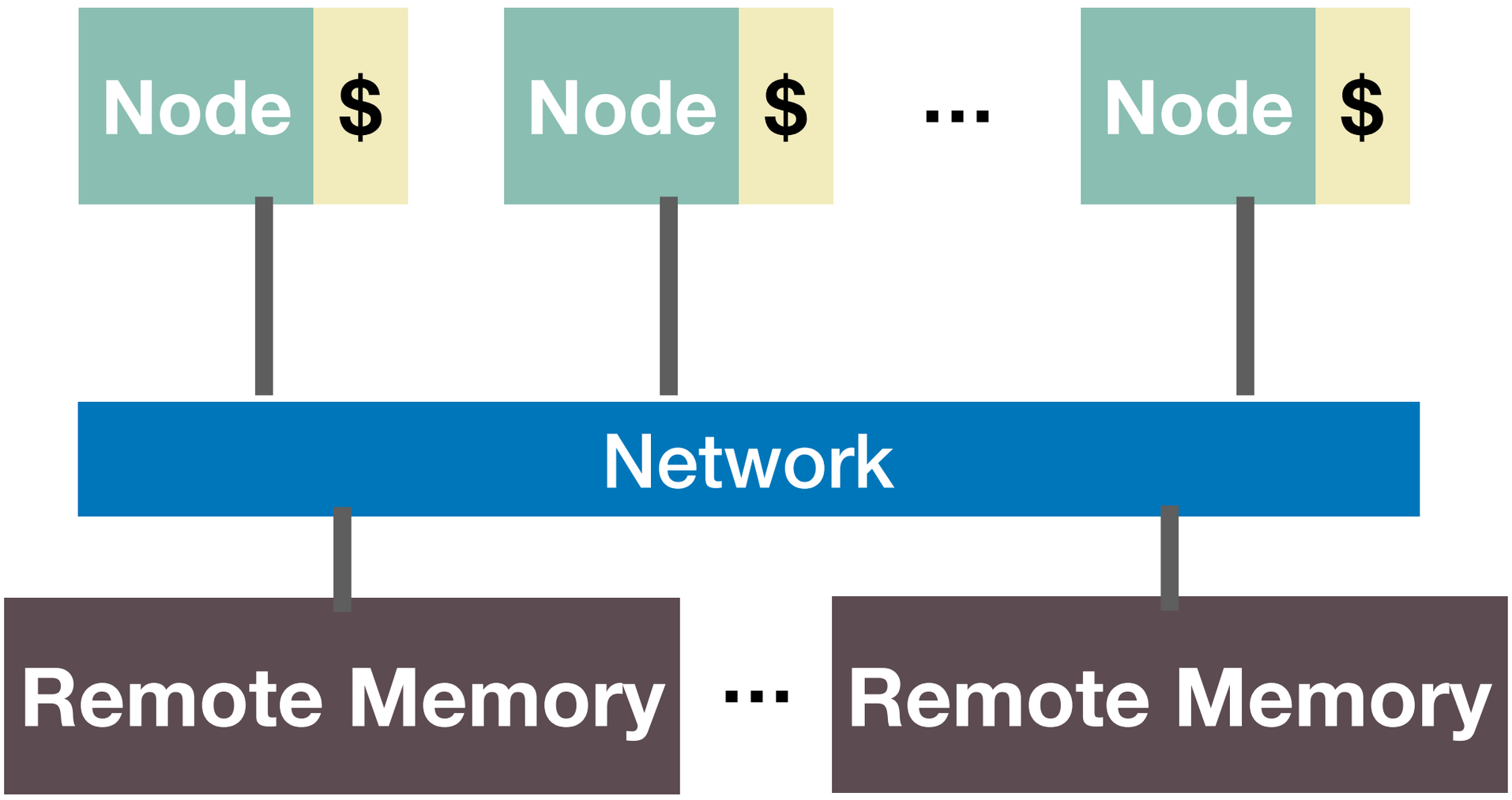
- Challenges
  - No computation power at memory
  - Remote memory can fail independently
- Opportunities
  - Can trace and control (remote) memory accesses
  - No local accesses that can violate atomicity

How to use one-sided remote memory/NVM devices?

Local Multiprocessor Shared Memory



Remote Memory



Remote Memory and Local Memory Comparison

- Similarities
  - No computation power
  - Multiple processors (cores) can read and write to
- Differences (remote memory)
  - No hardware coherence
  - Can fail independently (and more often)
  - Larger but slower than local memory

Our View of Remote Memory

- Treat remote memory as a raw data-store hardware
  - Similar to DRAM chips in local main memory
  - Fast, cheap data store

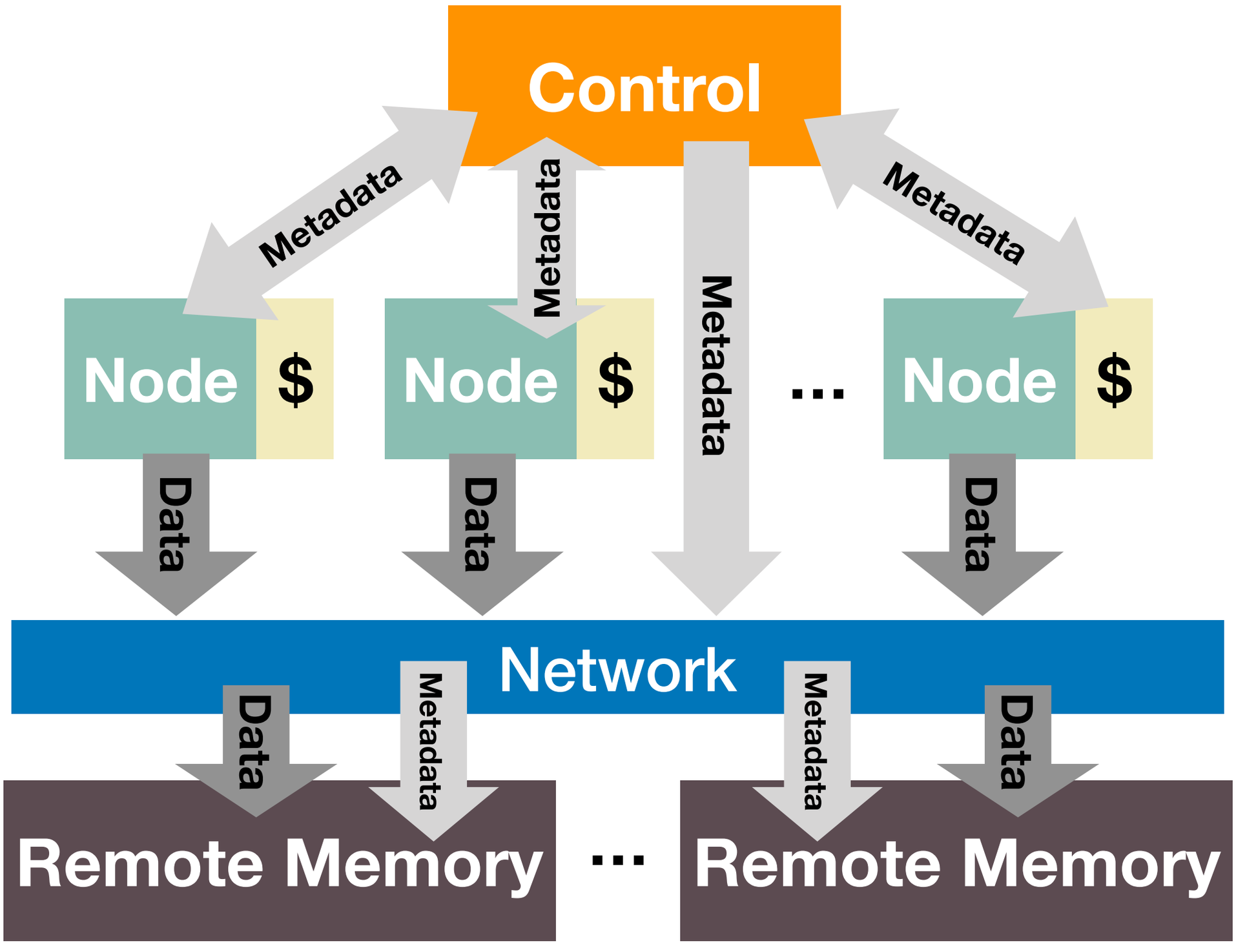
- Extract the control and intelligence apart
  - Similar to memory controller in local main memory
  - But managed in software

Mitsume*: an Object-Based Remote Memory System

- Separate data and control path
- Data: one-sided
  - Client nodes read/write to remote memory
  - Multiple processors (cores) can read and write to
- Control: two-sided
  - Global software controller manages remote memory and talks to clients via two-sided operations

* Mitsume means three eyes in Japanese and is from the manga and anime Mitsume ga Tooru (the Three-Eyed One)

Mitsume Architecture



Mitsume Data Organization

- Data stored and located by “object”
- Updates to an object guaranteed atomic and append-only
- Each object can have multiple versions
- Flexible physical locations of (versions) objects
- Each object can have their own replication factor

Global Control

- Allocate physical memory at remote memory
- Garbage collection
- Ensures QoS for different clients
- Resource management and load balancing
- Failure handling
- Security

Usage Models

- Key-value store
- Version system
- Remote swap
- Messaging system
- Pub/Sub

Conclusion

- One-sided memory/NVM devices are useful
- Learn from local memory system
- Separate data and control of remote memory
- Many usage possibilities of remote memory

Thank you
Questions?

WukLab

wuklab.io

PURDUE UNIVERSITY