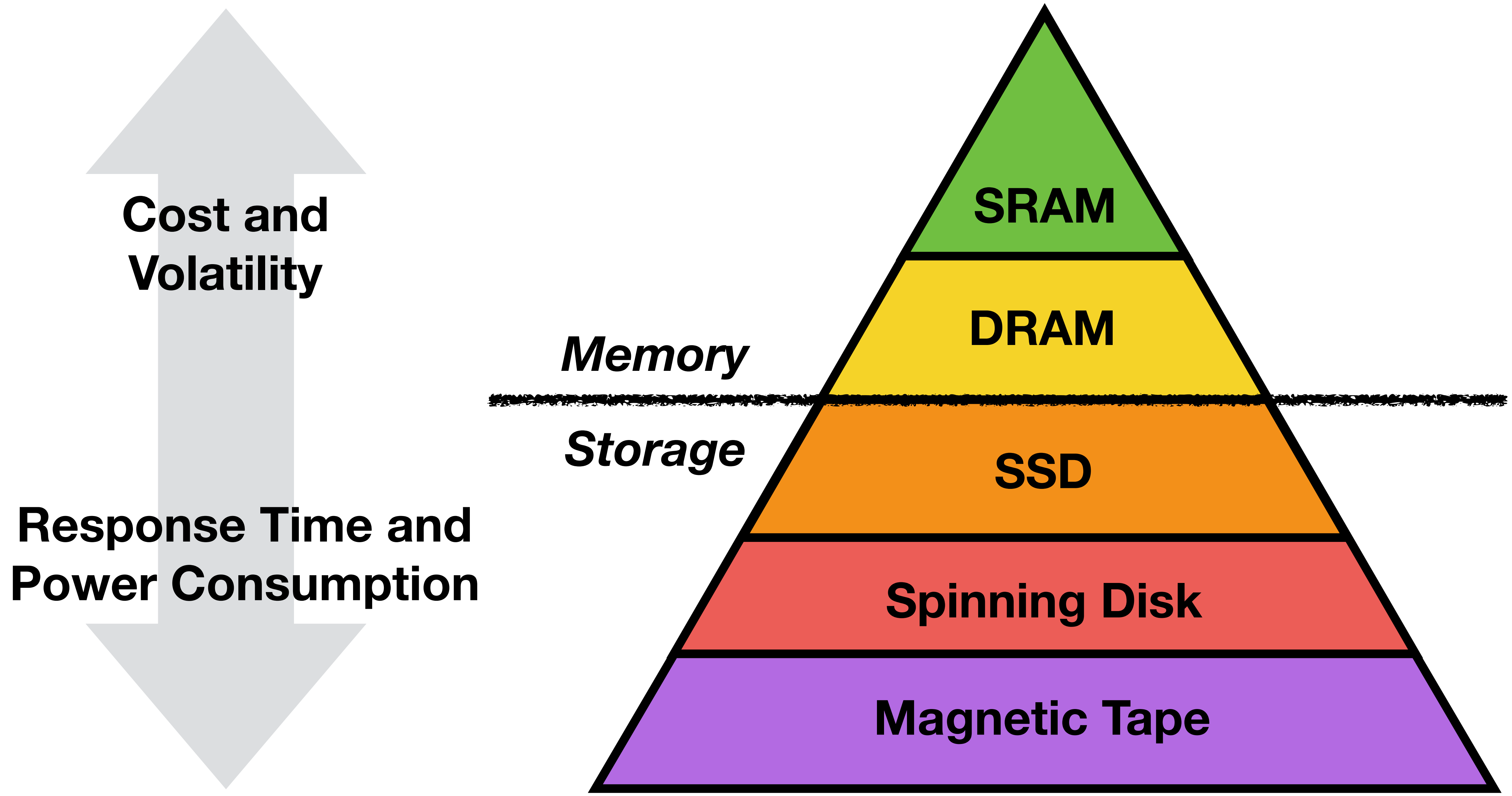
Consensus for Non-Volatile Main Memory

Huynh Tu Dang, Jaco Hofmann, Yang Liu, Marjan Radi,
Dejan Vucinic, Fernando Pedone, and Robert Soulé

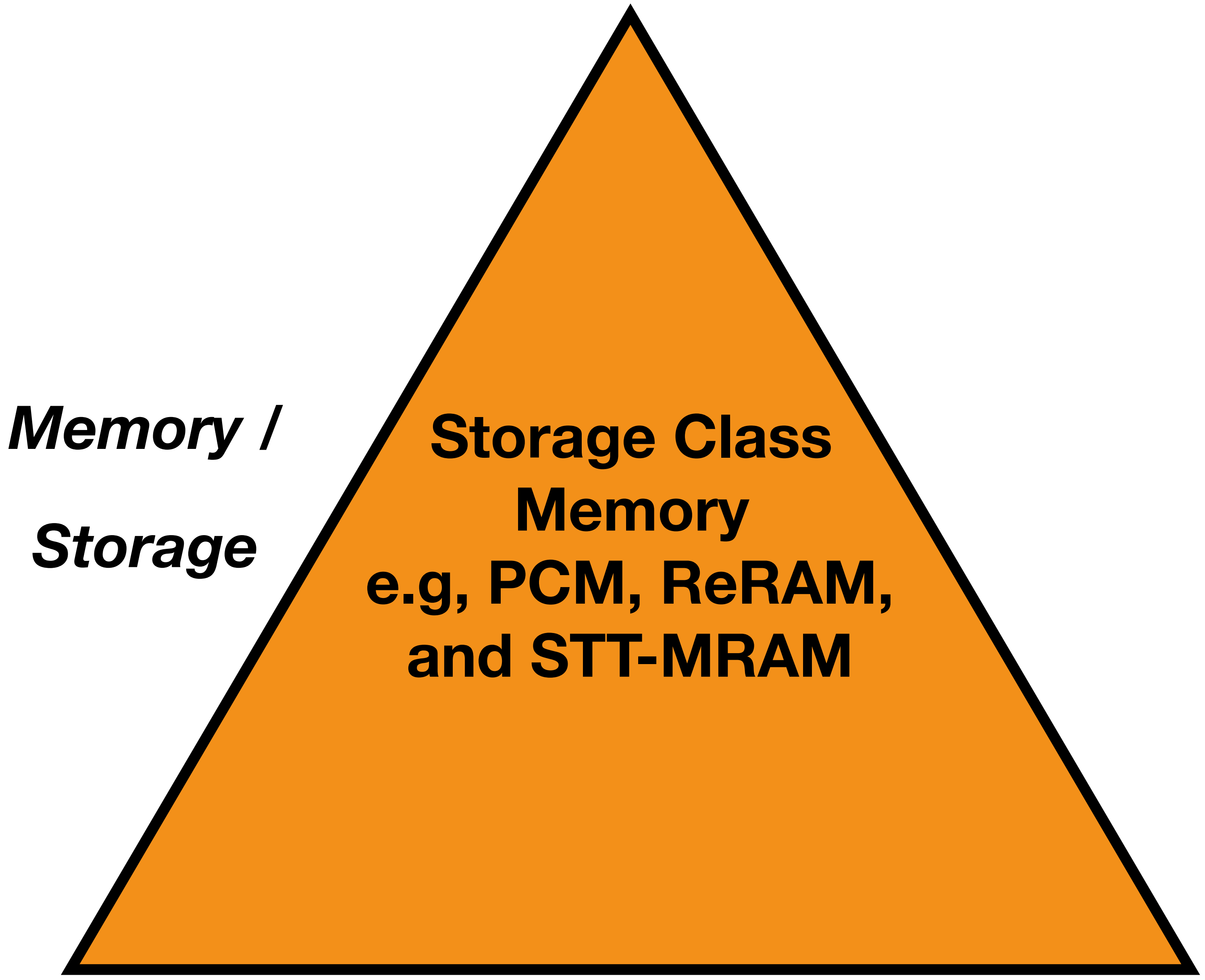
University of Lugano, TU Darmstadt, and Western Digital

Traditional Hierarchy



SCM is Changing the Hierarchy

- Non-volatile
- Byte-addressable
- Response time close to DRAM
- Simpler architecture, denser packing == lower cost

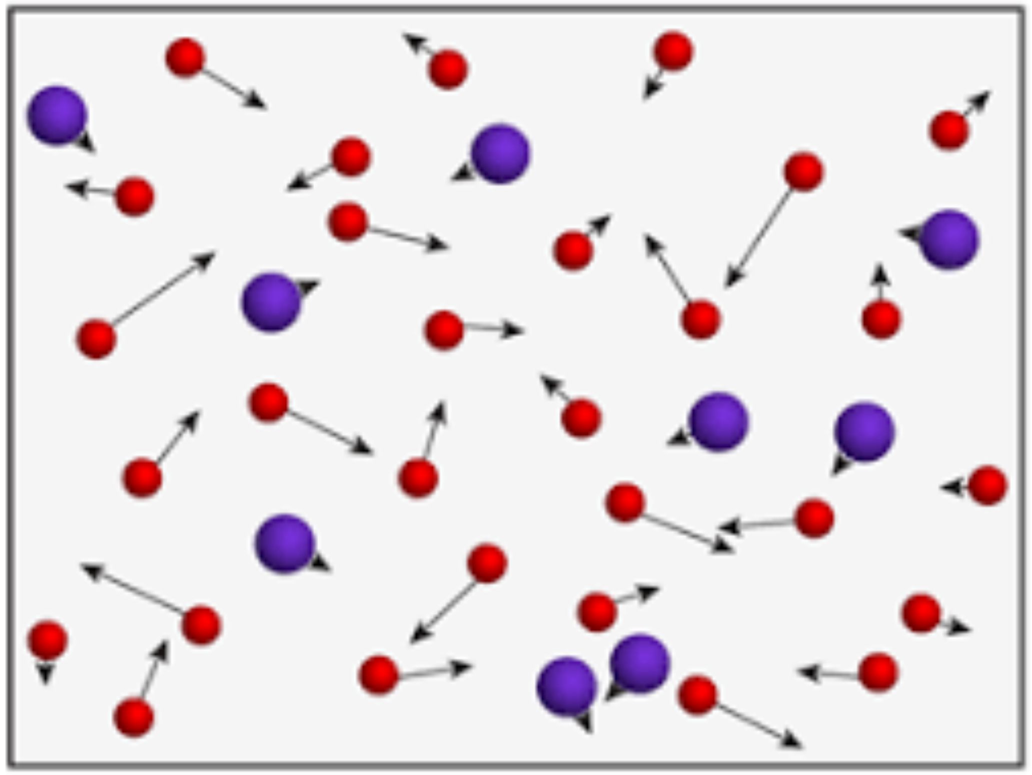


Benefits

- Architectual simplicity
    - No need to separate in-memory cache from persistent storage
- Scale storage and compute separately
    - Improve efficiency of storage utilization
- Reduces total cost of investment in the data center
    - Allows for pay-as-you-grow planning

The Problem with SCM

- All SCM technologies involve the movement of atoms
- Wear-out is unavoidable
- Imposes practical limits:
  - Single system with SCM as a replacement for DRAM
  - Scale-out size of storage systems built with SCM

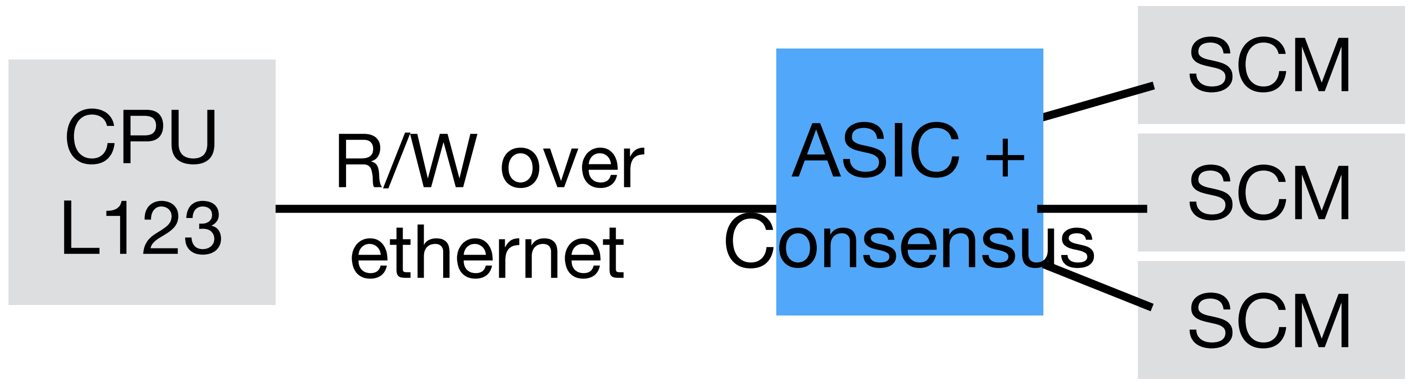


Handling Failures in Memory and Storage

Medium	Approach	Problem
CPU main memory	Ignore problem	System crashes
Super computer main memory	Checkpointing	Complicated management and cost
Disk and SSD	RAID	Centralized controller doesn’t scale

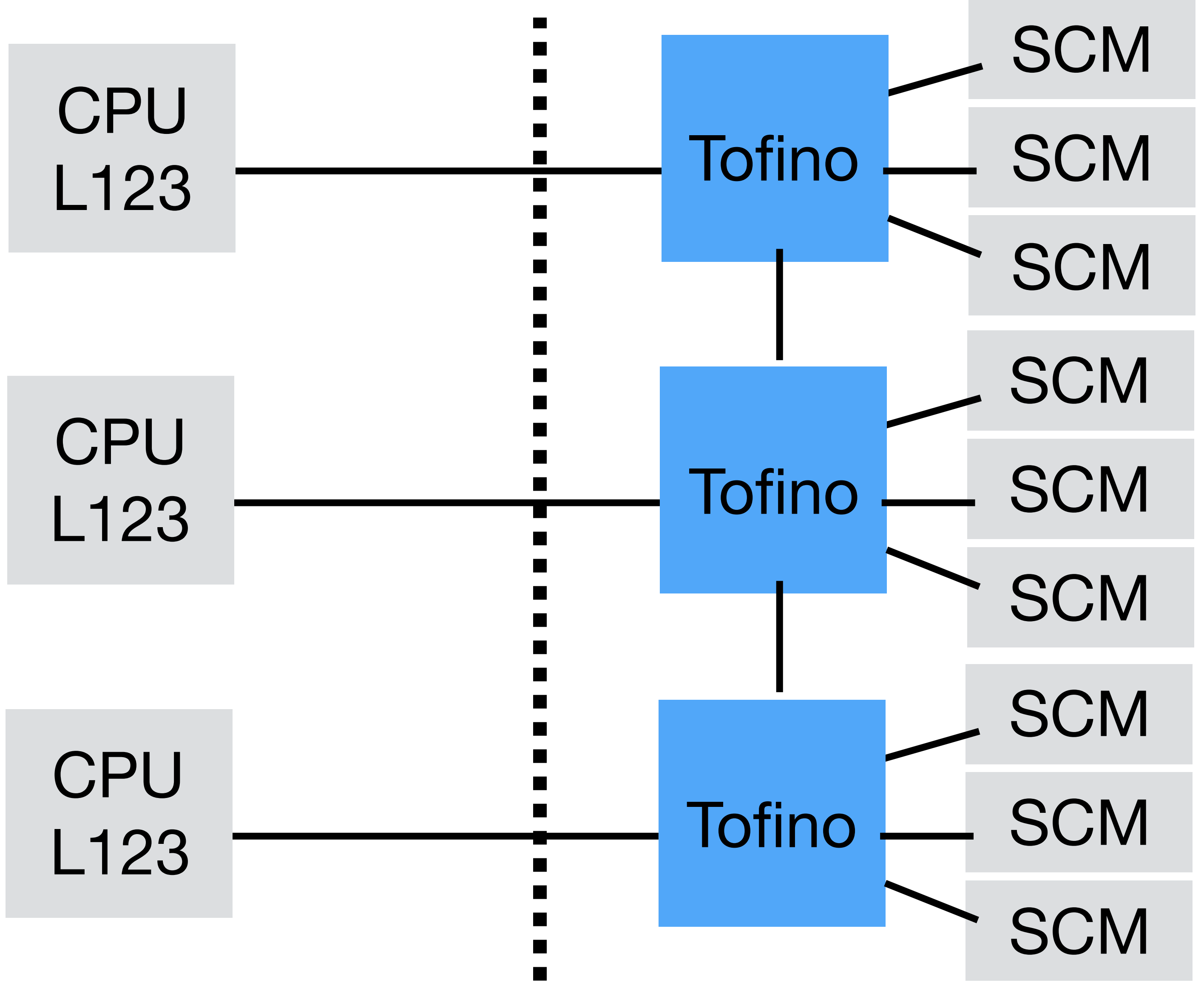
Key Idea

- Treat memory as a distributed storage system
- Replicate data to cope with failures
- Use consensus protocol for consistency



Long Term Goal

- Build a memory appliance
- Offer a petabyte or more of main memory
- Remote accessible from many cores



storage appliance

Consensus For Memory?

- Historically considered a performance bottleneck
- In-network shows great promise:
  - E.g., NetPaxos, NetChain, Speculative Paxos, Consensus in a Box
- Our approach: use a generalization of a protocol by **Attiya, Bar-Noy, Dolev (ABD)**

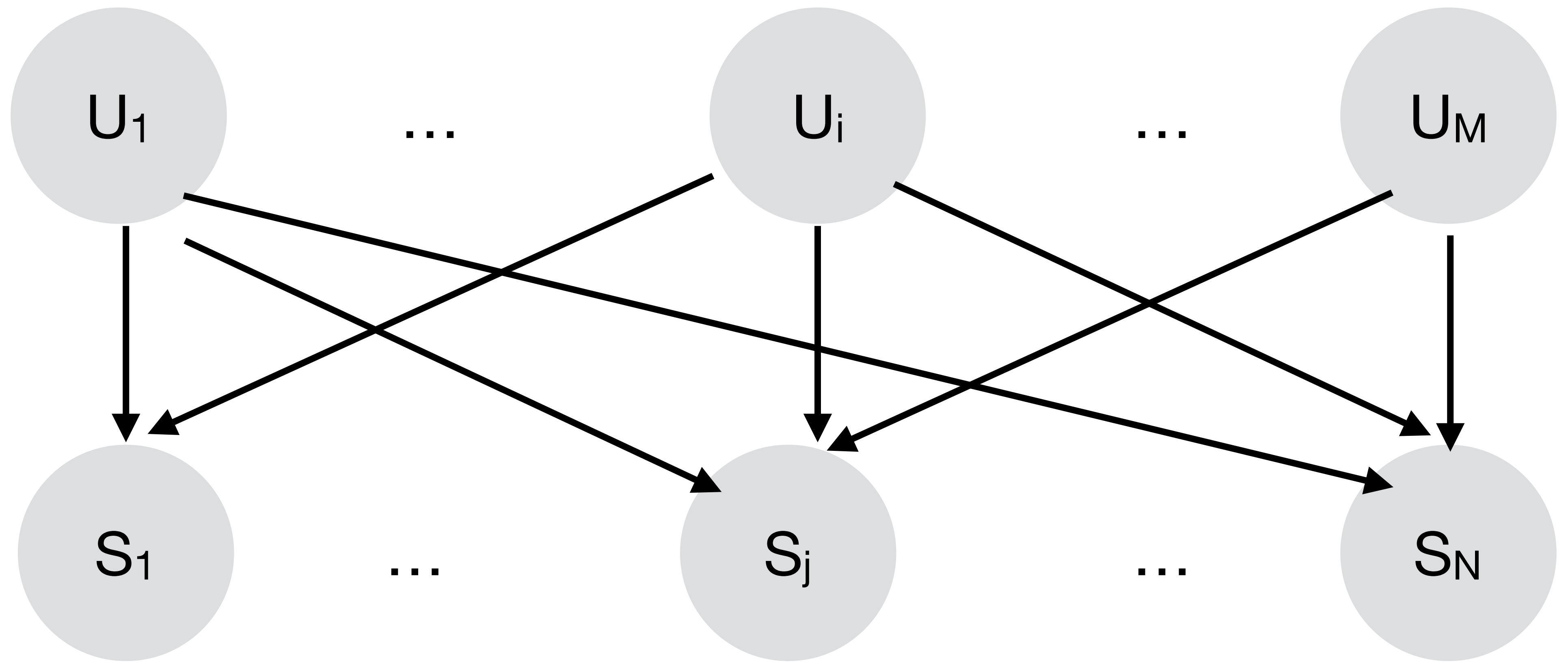
ABD vs. Paxos

- **Failure assumptions.** Paxos depends on election of non-faulty leader. ABD only depends on availability of majority.
- **Simpler protocol.** Paxos supports arbitrary operations. ABD only supports read/write operations. All we need for memory access.
- **Less state.** Paxos keeps replicated log at acceptor, which must be check-pointed frequently, adding overhead. ABD only has clients and servers.

Design Assumptions

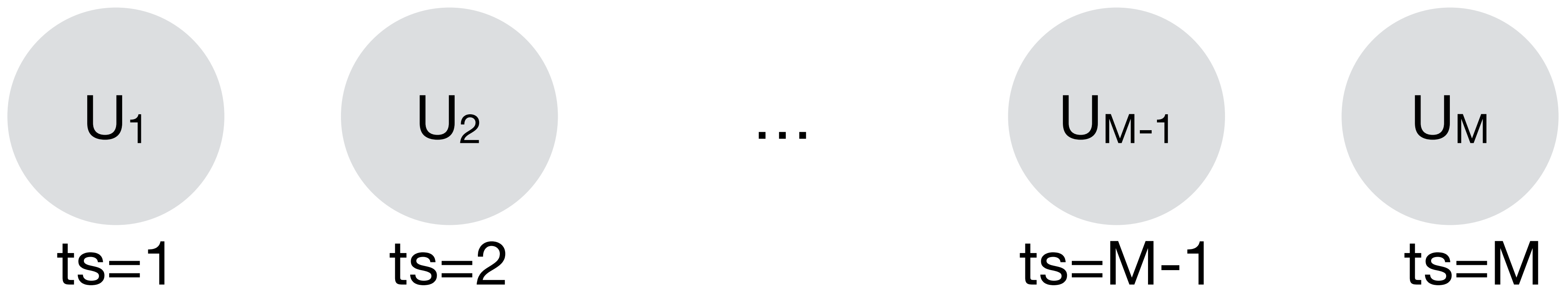
- Do not extend memory controller with logic for replication
- Cache lines are 64 bytes
- Switches do not fail (for now)
- Clients are directly connected to the switch, one client per port
- ~1000 CPUs, each issuing about 10 concurrent requests, so 10K concurrent requests (low bandwidth)

Protocol Setting



- M user processes
- N server processes
- Every user processes can send a message to every server process

Timestamps

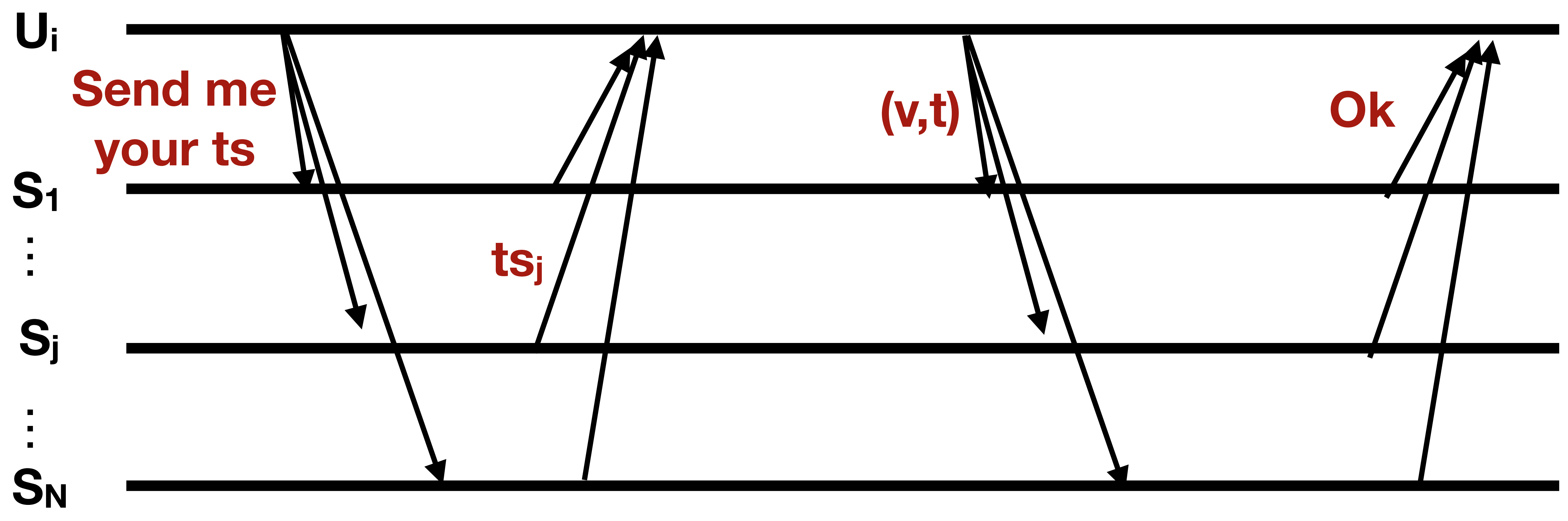


- Each U_i chooses timestamps of the form {i, M+i, 2M+i,…}
- For example, if M=32 (there are 32 user processes)

  U_1 chooses timestamps of the form {1, 33, 65,…}
- Allows us to easily identify which process issued a write

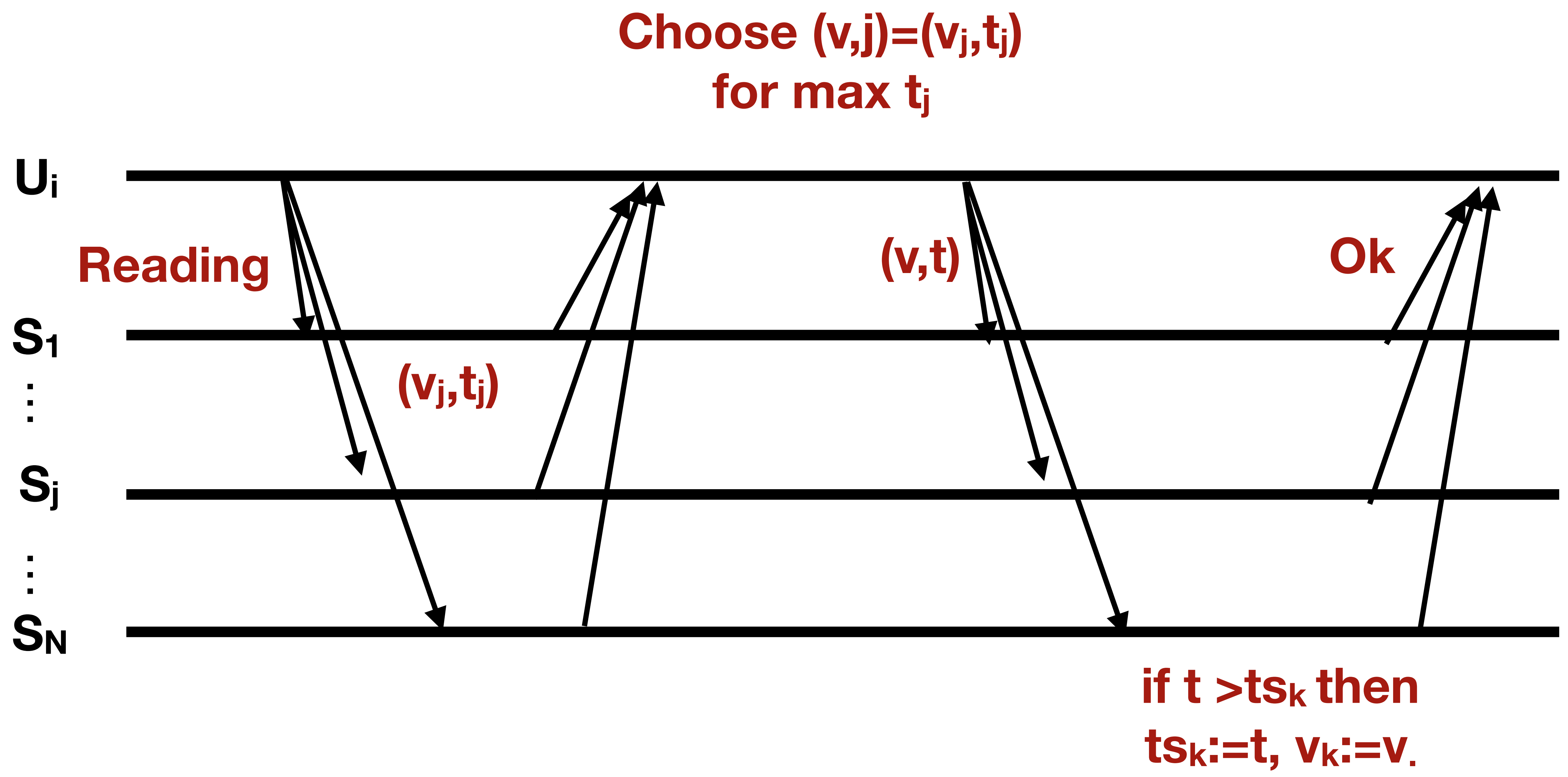
Write Operation

U_i chooses $t=pM+i$,
s.t. t is bigger than previous t
and any ts it received



if $t > ts_k$ then
$ts_k := t$, $v_k := v$.

Read Operation



Reads/Writes Complete Writes

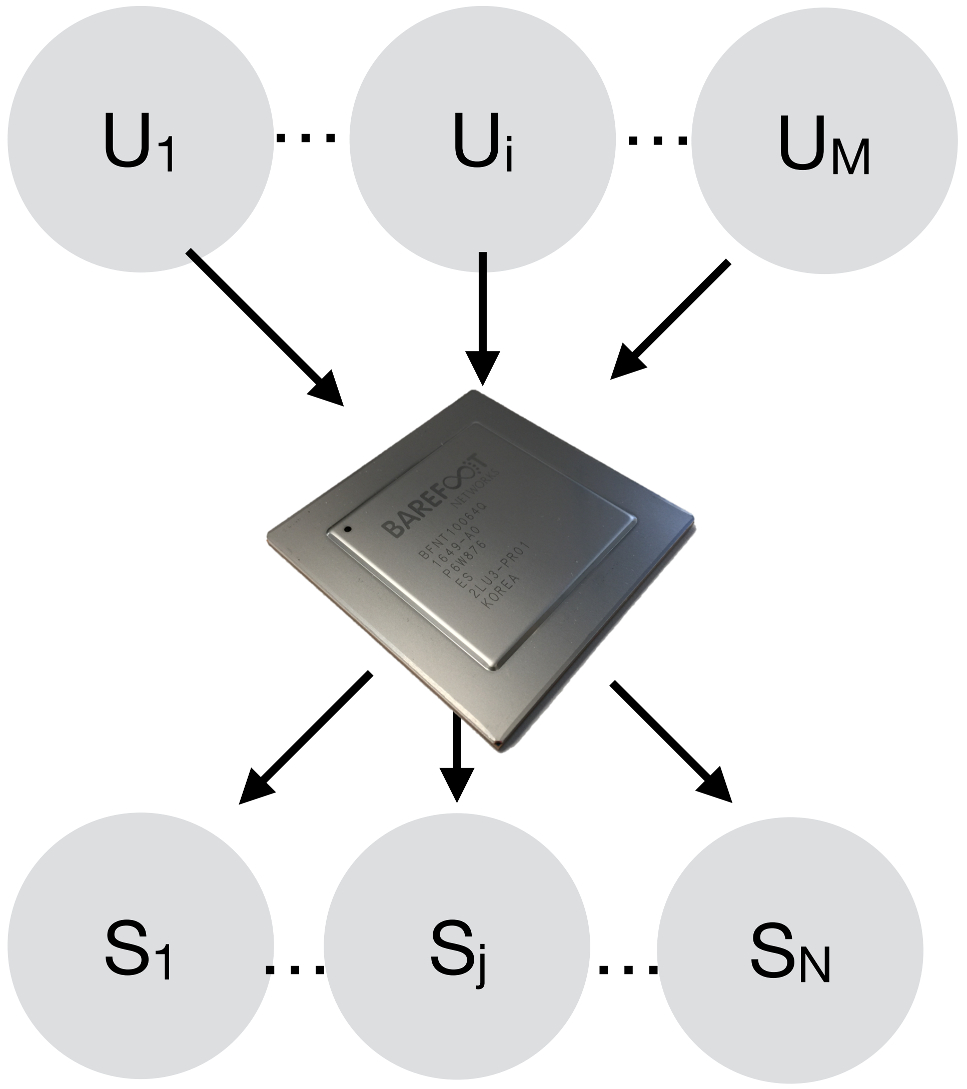
Time	S1	S2	S3	S4	S5	
W1	V1,1	V1,1	V0,0	V1,1	V1,1	***W1 on S1-S2,S4-S4; W1 complete***
W2	V1,1	V1,1	V0,0	V2,2	V2,2	***W2 on S4,S4; W2 incomplete***
R1	V1,1	V1,1	V1,1	V2,2	V2,2	***R1 on S1-S3***
R2	V1,1	V1,1	V2,2	V2,2	V2,2	***R2 on S3-S5; W2 complete***

ABD on Tofino: Challenges

- Original protocol designed for a single register.
  We need to generalize for multiple registers.
- Need to temporarily store the 64-bit value and timestamp
  - Need to keep per-port timestamps in switch instead of client
- Need to keep requests on a per-address basis
  - Address space is too big to have 1 register per address
  - May run into collisions with hashing

ABD on Tofino: Open Questions

- ABD only has clients and servers. Now, we have entity in middle
- How to we preserve same liveness guarantee if switch fails?
- What is the interplay between cache coherence and consistency?

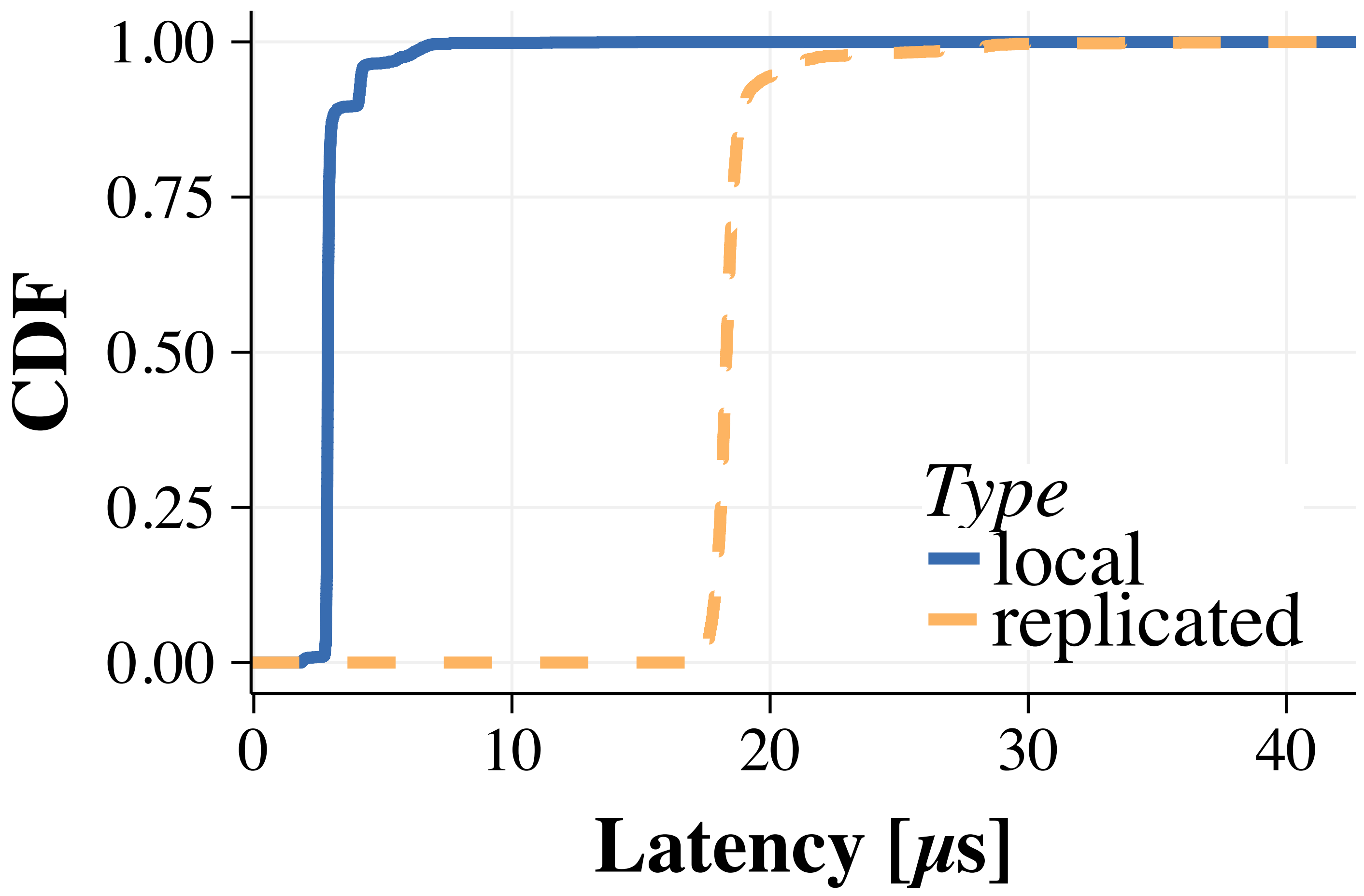


Memory Controller

- Don’t yet have a true hardware memory controller
- Emulate controller with special device drivers
  - Client side intercepts calls to malloc, invokes mmap on character device
  - Client allocates memory from the remote server
- When there is a page fault, issue ABD access to fetch remote page

Evaluation

- Remote RPC vs. local memory, 100K writes
- Local latency ~3µs, remote is ~18µs
- Includes parsing of the L2 header



Conclusions

- Storage Class Memory can transform the memory hierarchy
- In-network consensus helps solve a critical challenge for SCM
- Initial experiments demonstrate orders-of-magnitude faster than traditional storage systems and shows great promise as scalable memory

http://www.inf.usi.ch/faculty/soule/