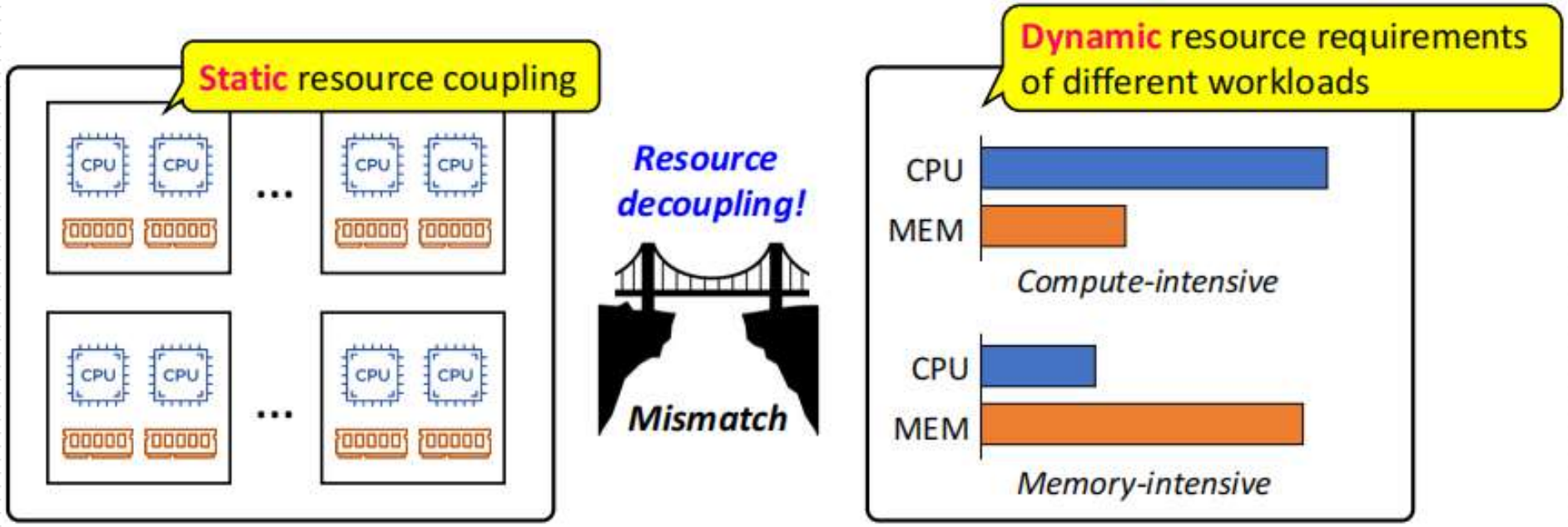
Motor: Enabling Multi-Versioning for Distributed Transactions on Disaggregated Memory

Ming Zhang, Yu Hua, Zhijun Yang
Huazhong University of Science and Technology, China

Presented by Sen Han

Background-DM

Insufficient Memory Utilization in Cloud (20%~60%[1-4])

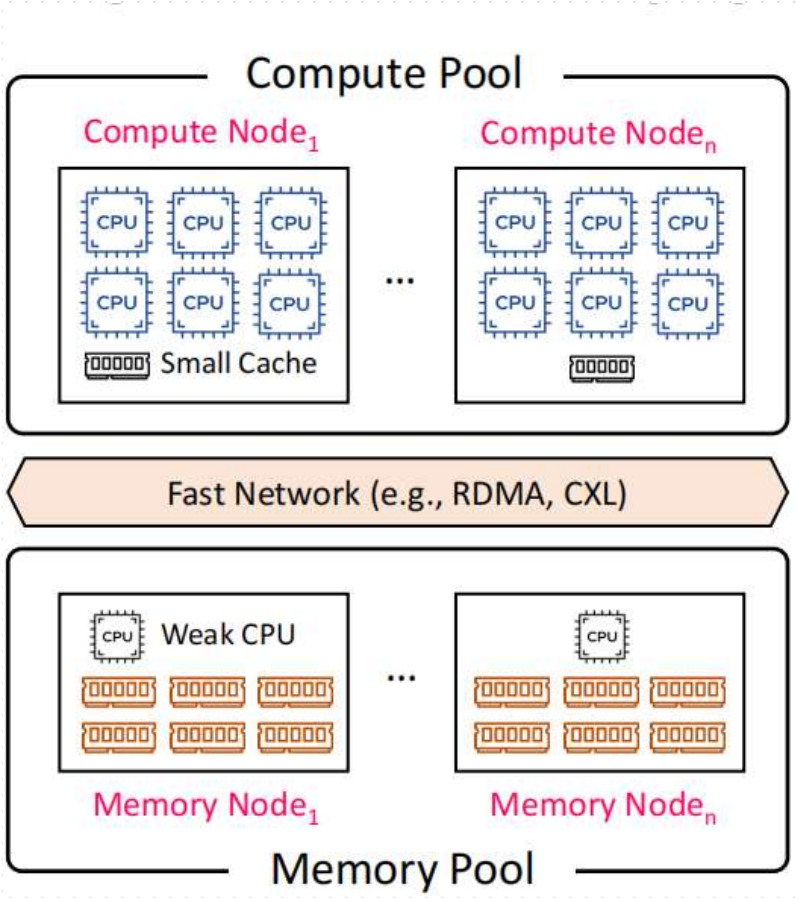


[1] MemTrade@SIGMERTICS'23, Borg@Eurosys'20, LegoOS@OSDI'18
[2] Google Production Cluster Trace. https://github.com/google/cluster-data
[3] Alibaba Production Cluster Trace. https://github.com/alibaba/clusterdata
[4] Snowflake Dataset. https://github.com/resource-disaggregation/snowset

Background-DM

USTC, CHINA ADSLAB

Disaggregated Memory(DM)

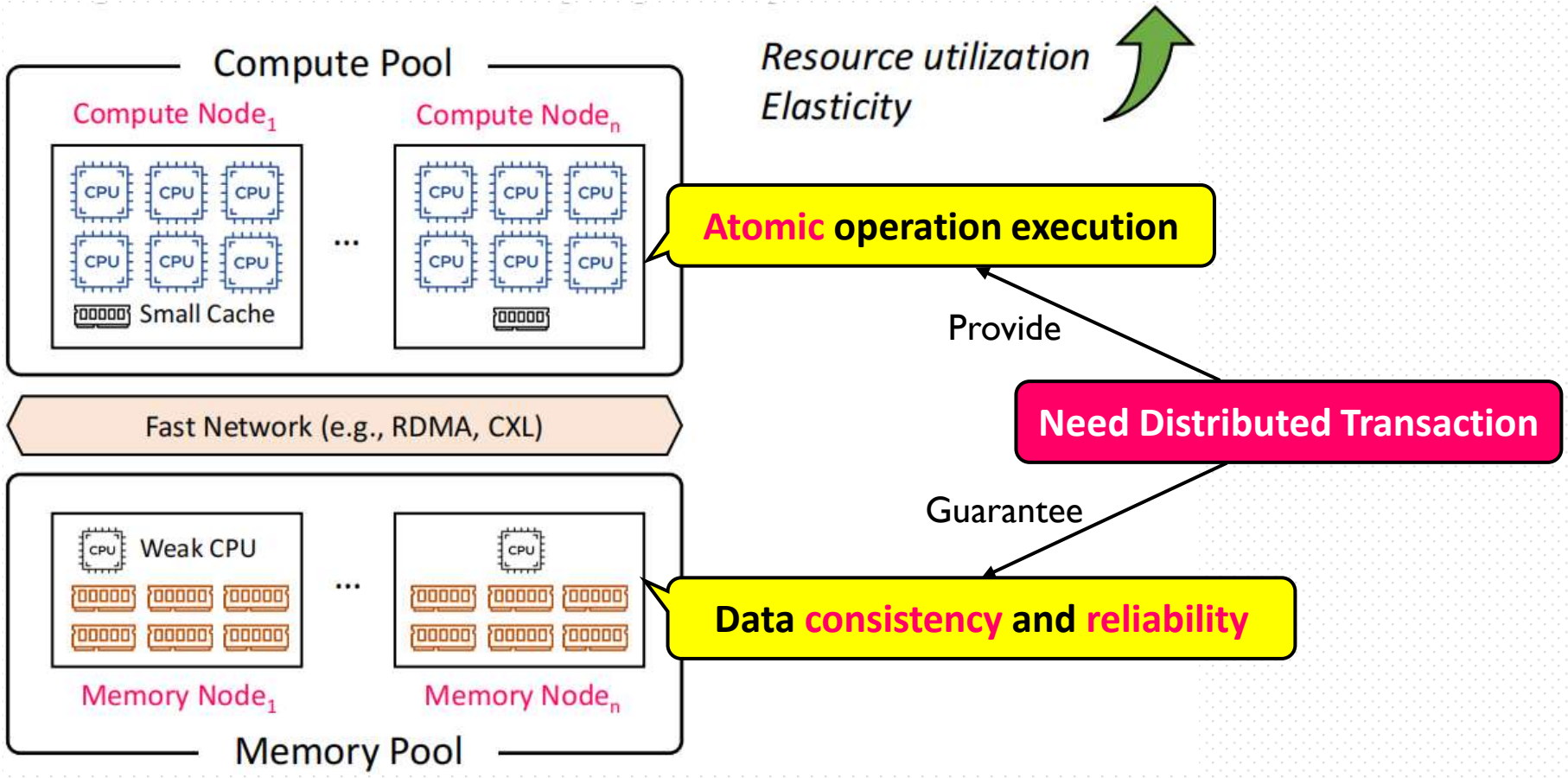


Resource utilization
Elasticity

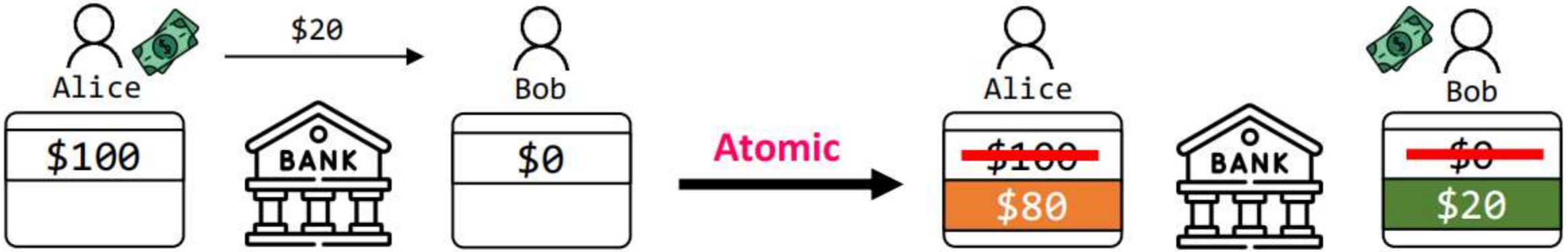
Background-DM

USTC, CHINA ADSLAB

Disaggregated Memory(DM)



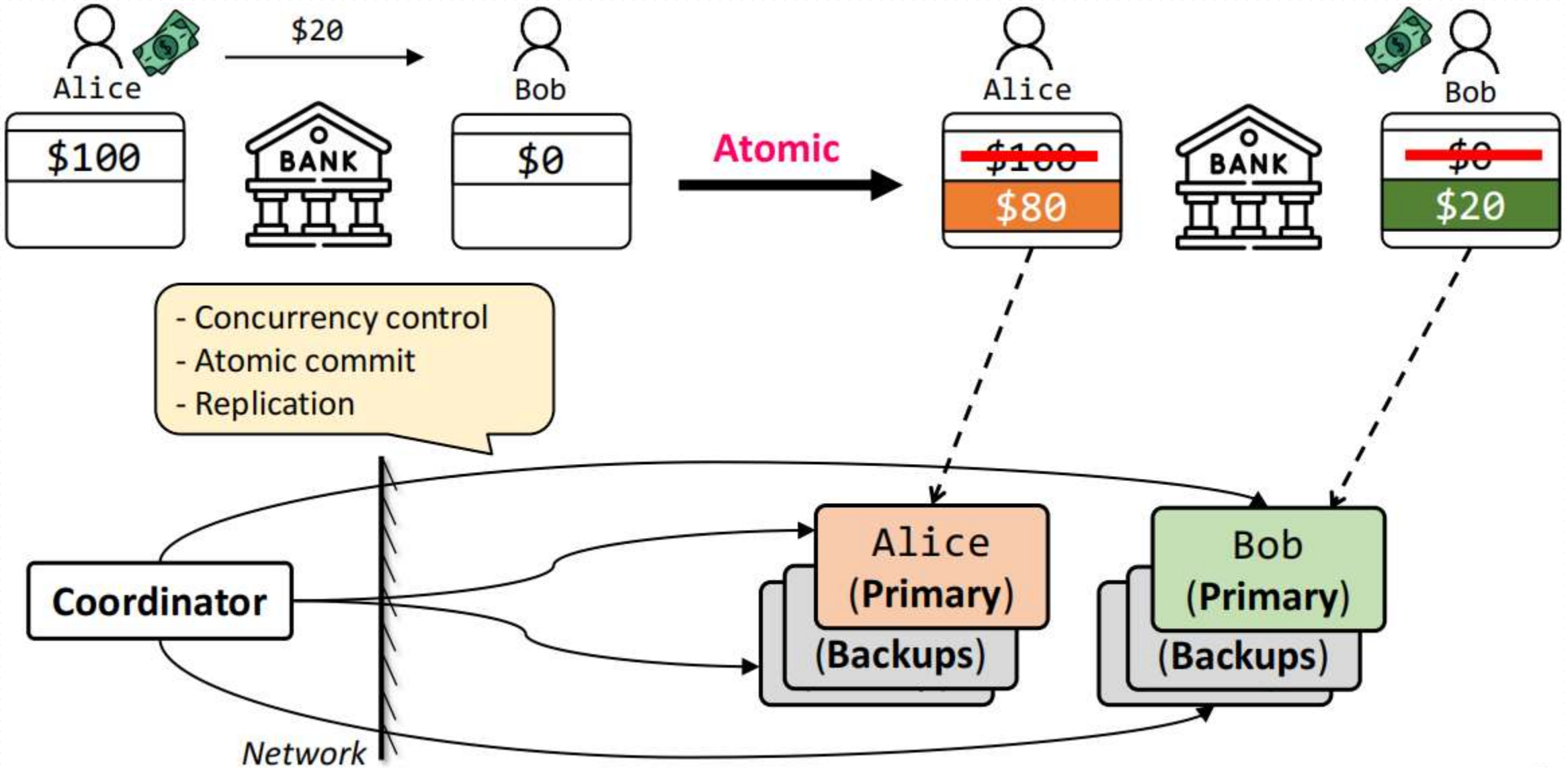
Background-Transaction



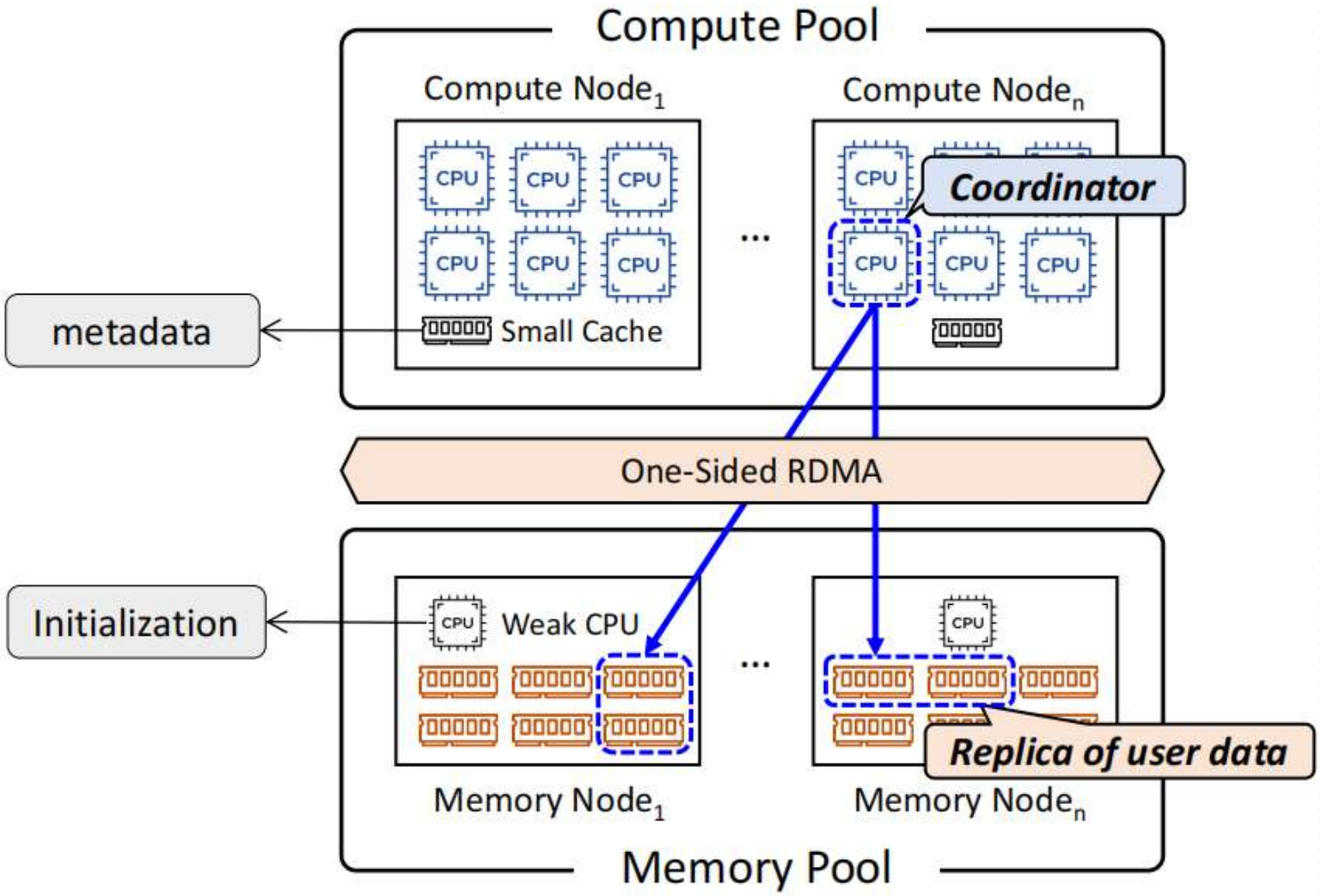

```
Txn begin
  Alice: $100 → $80
    Bob:   $0 → $20
Txn end
```

Transaction

Background-Transaction



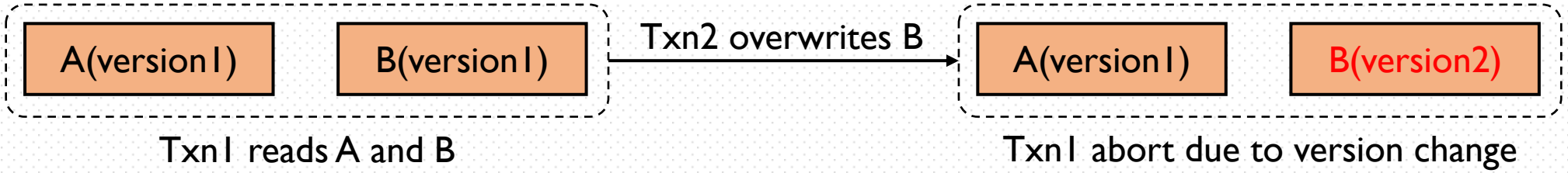
Background-Transaction on DM



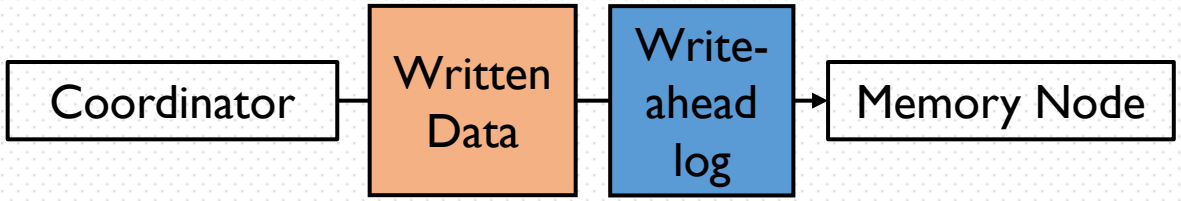
Existing Studies

Single-versioning distributed transaction system for DM[1]

limit concurrency



High logging overhead

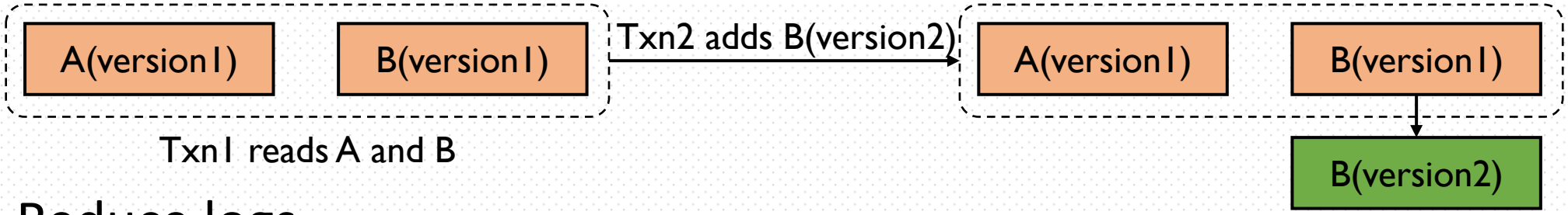


[1] FORD@FAST'22

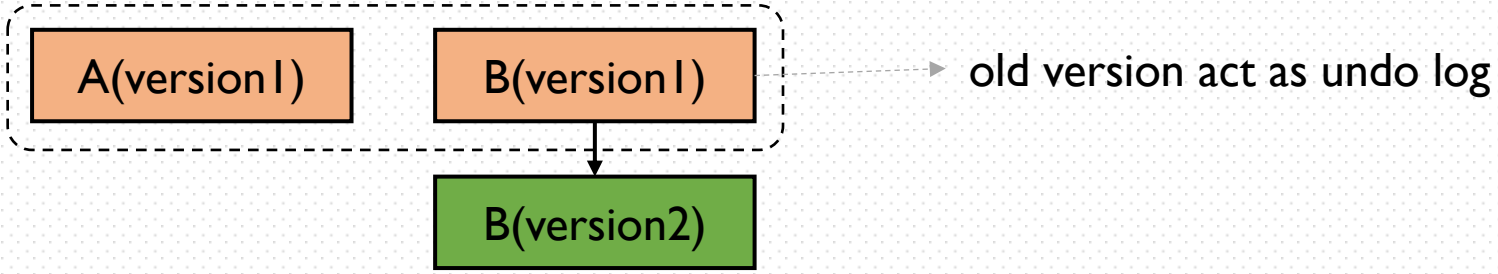
Existing Studies

Multi-versioning helps address limitations of single-versioning

Allow more concurrency



Reduce logs



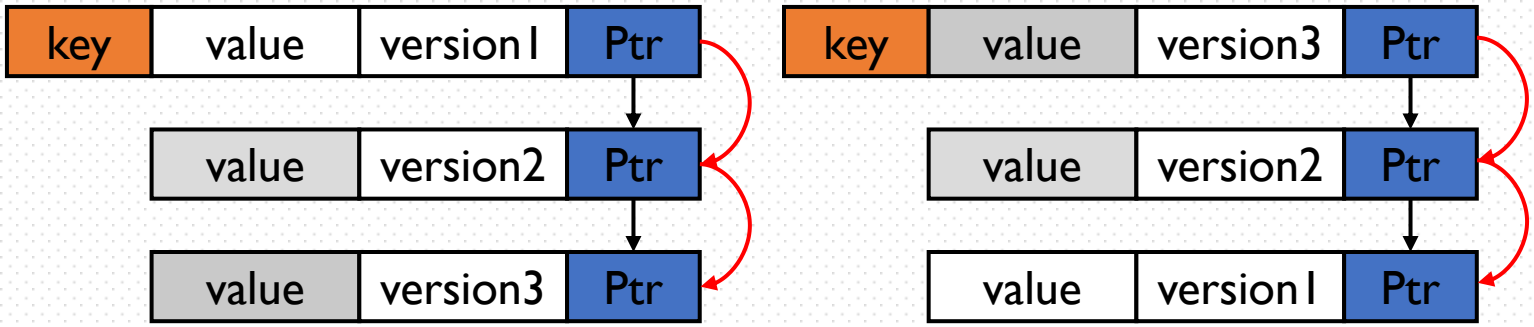
[1] FORD@FAST'22

Does Multi-Versioning Work on DM?

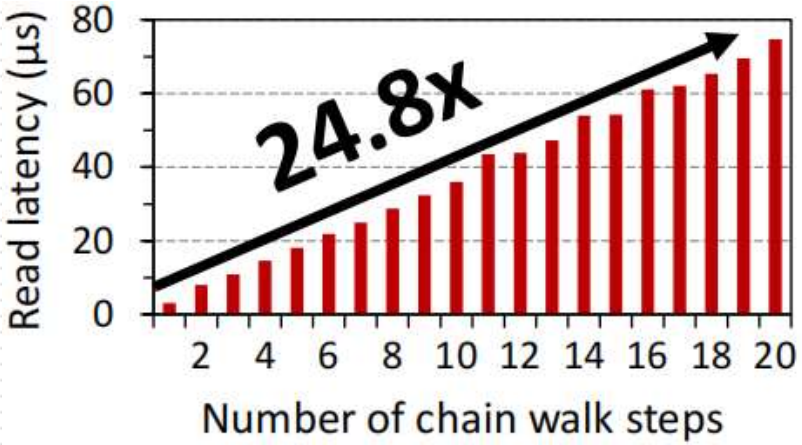
USTC, CHINA ADSLAB

Existing systems are based on monolithic servers

Inefficient linked version chain does not fit DM



Old-to-new chain[1-3] New-to-old chain[4-6]

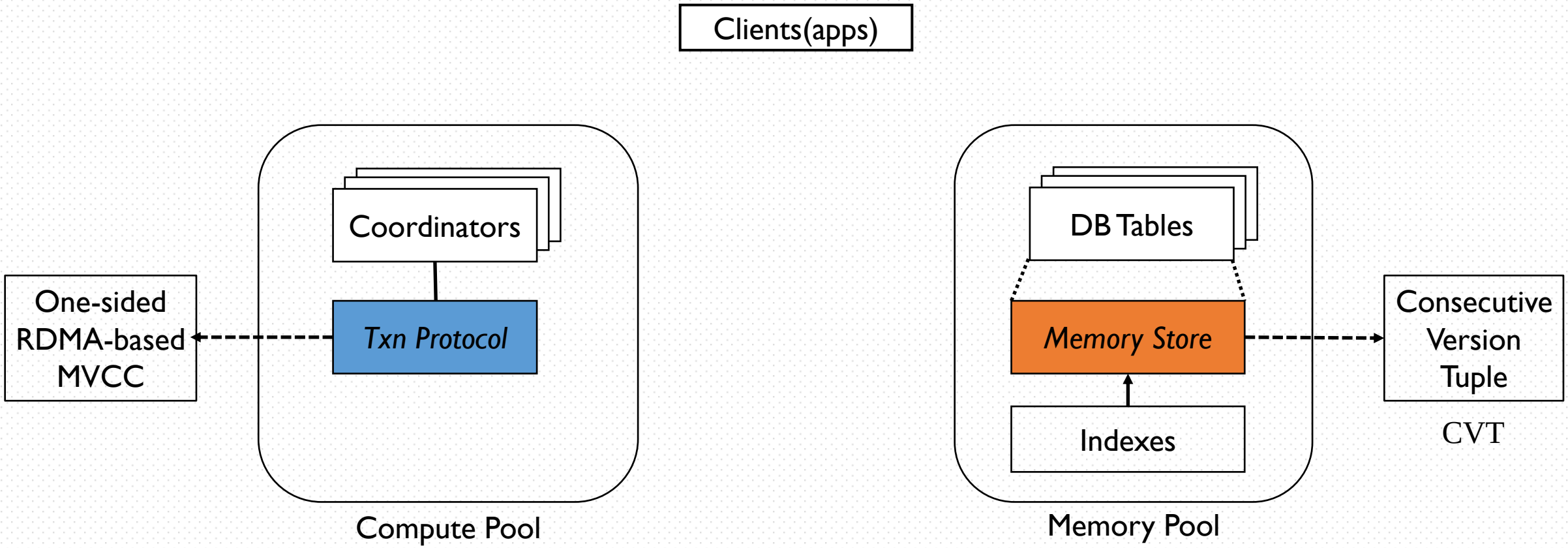


Incompatible transaction protocol

- Frequently consumes CPU of each data node[1,4,5]
- Memory node stores data but only has weak CPU

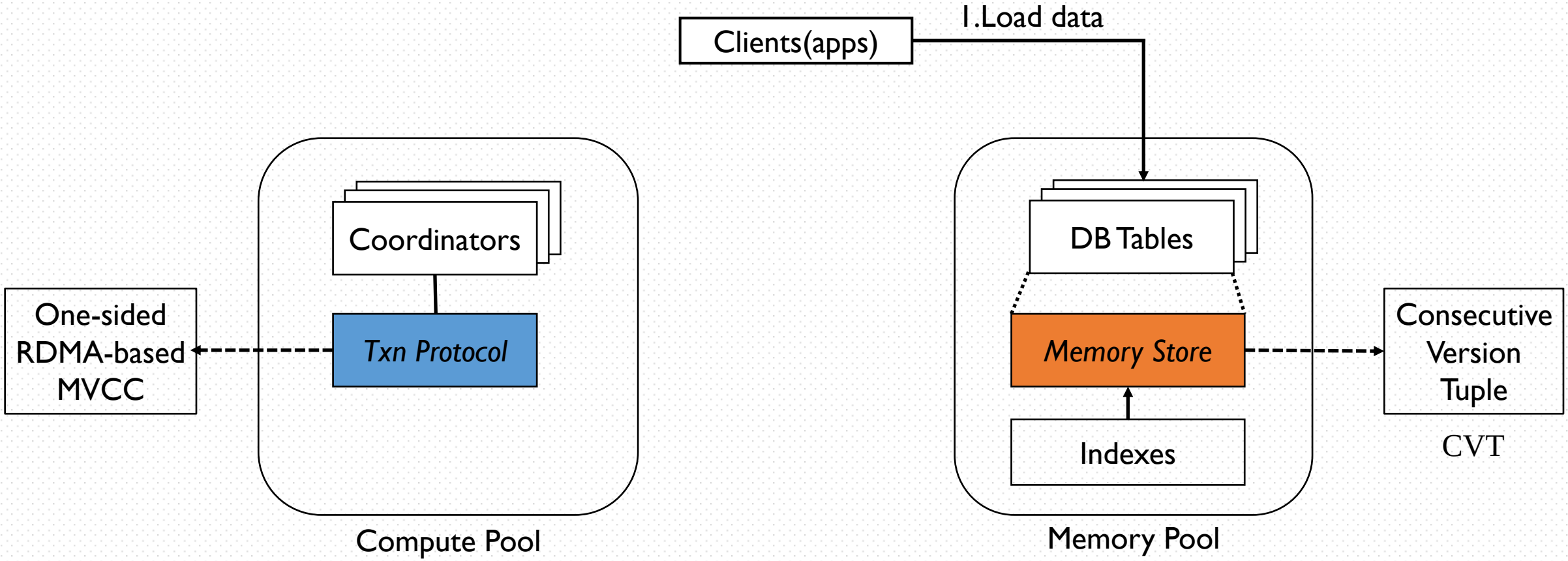
[1] DST@NSDI'21 [2] Hekaton@SIGMOD'13 [3] Aurogon@FAST'22
[4] FaRMv2@SIGMOD'19 [5] Neumann et al.@SIGMOD'15 [6] NAM-DB@VLDB'17

Design



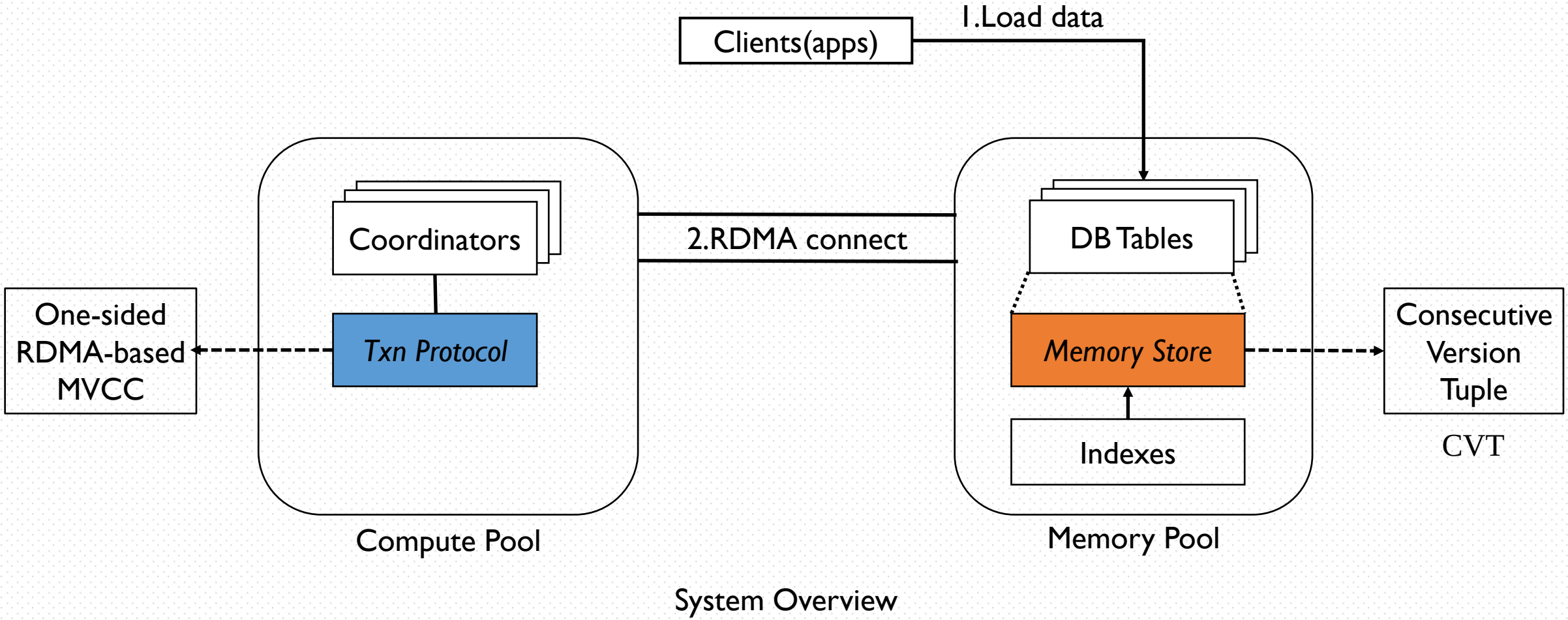
System Overview

Design



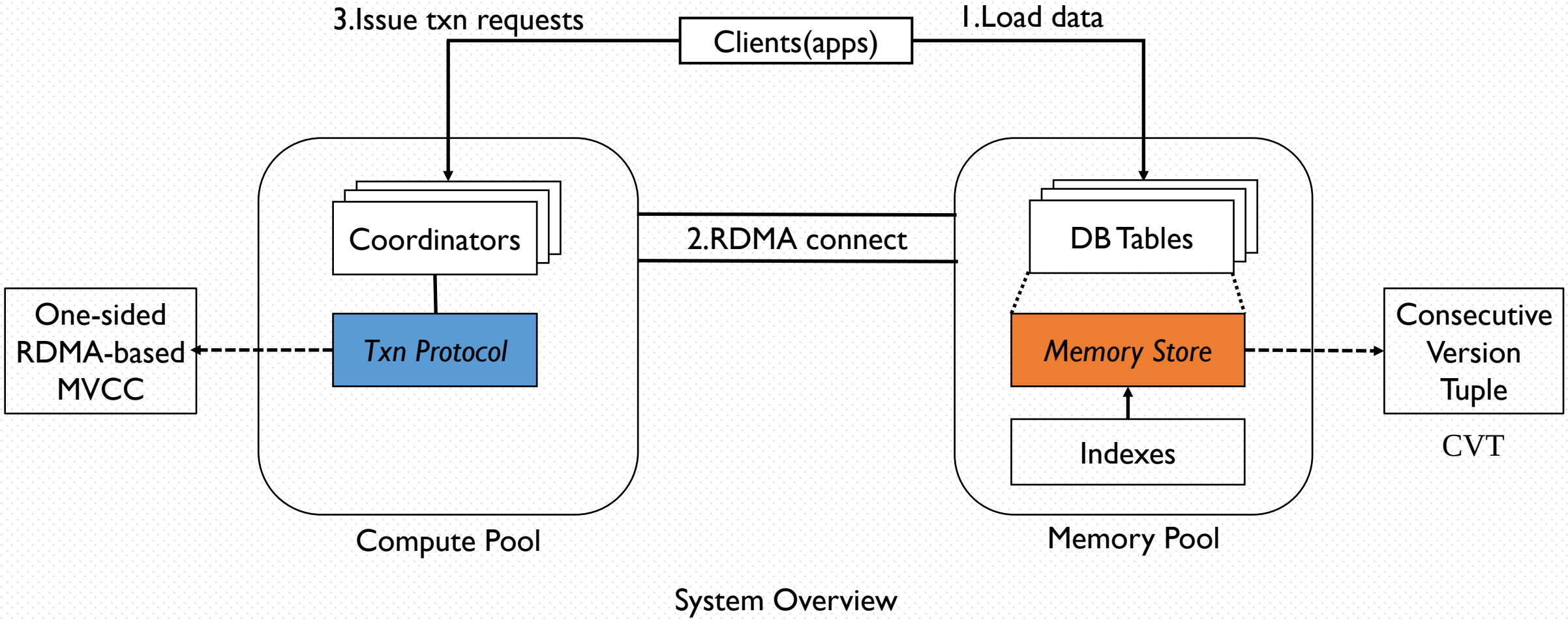
System Overview

Design



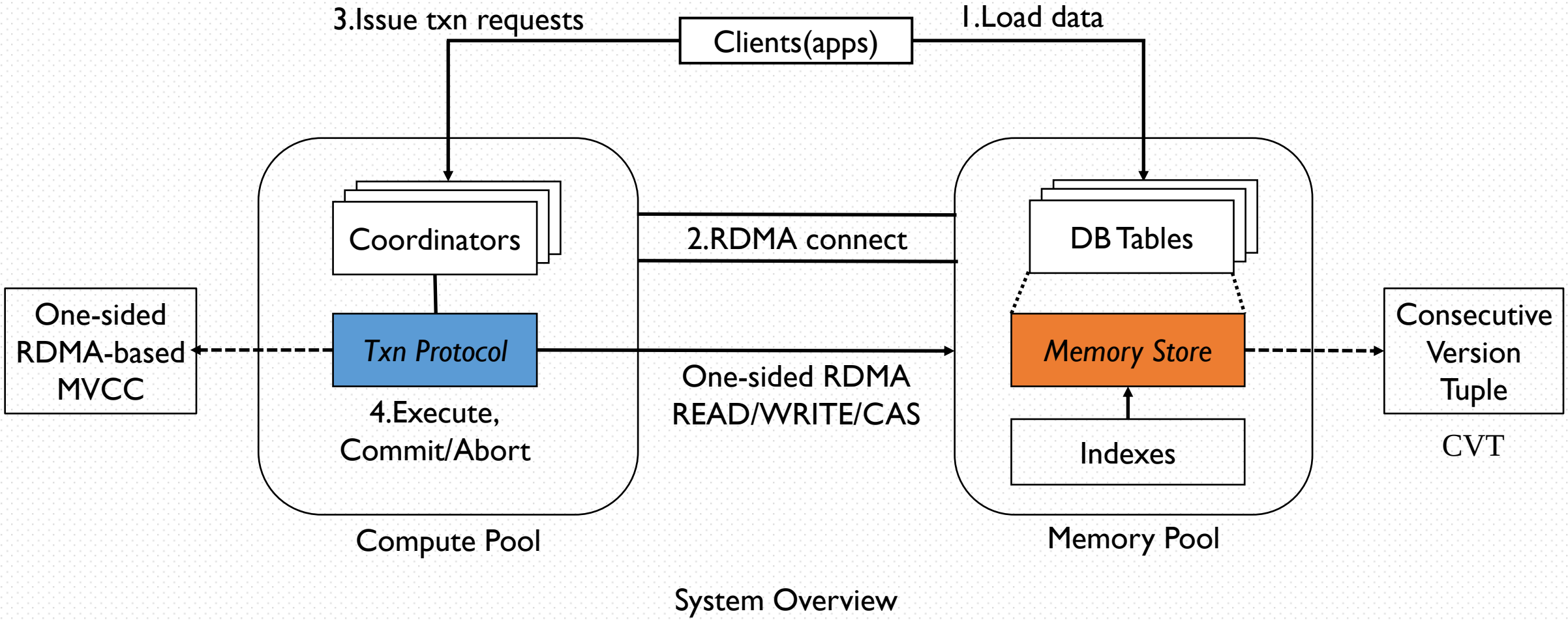
System Overview

Design



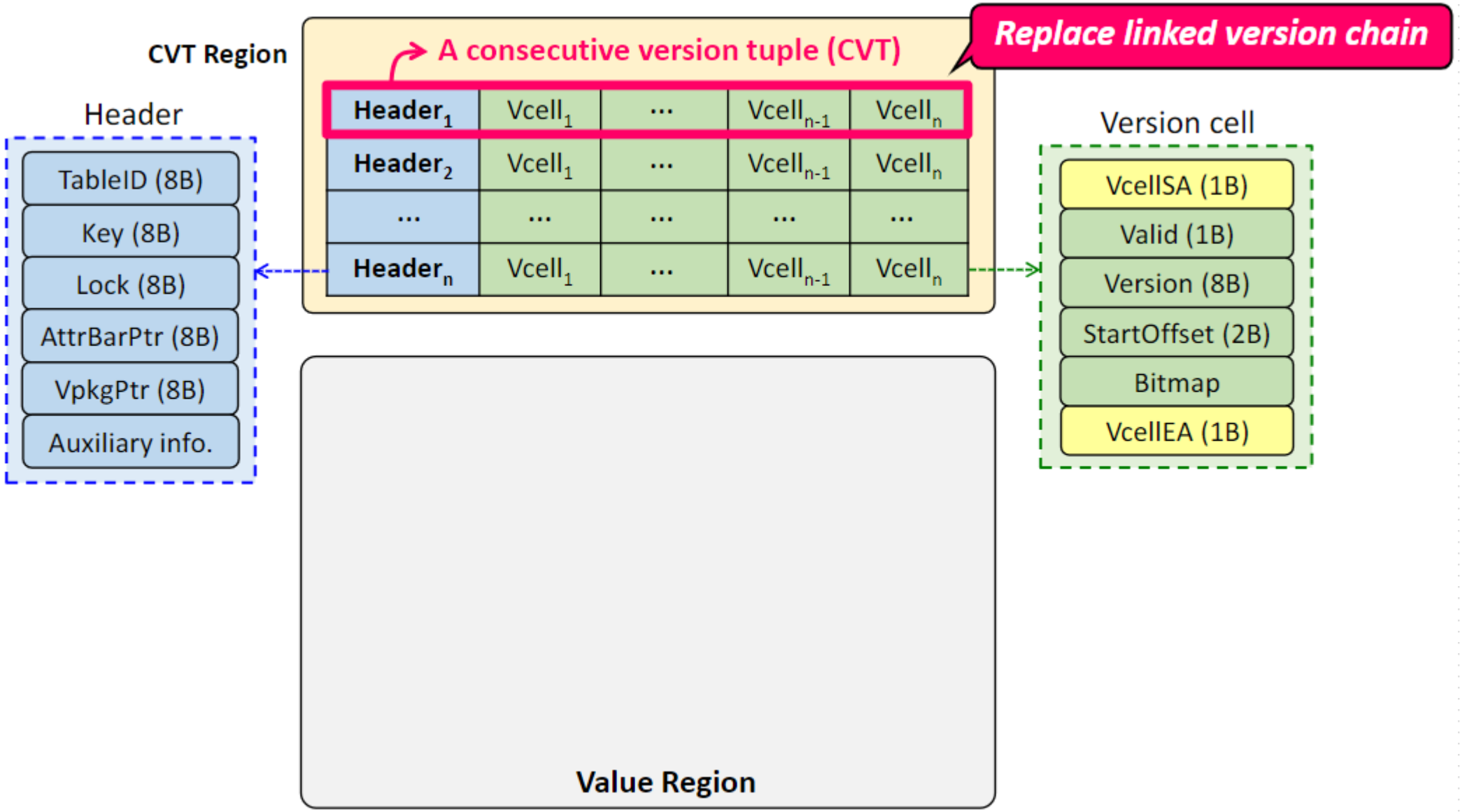
System Overview

Design

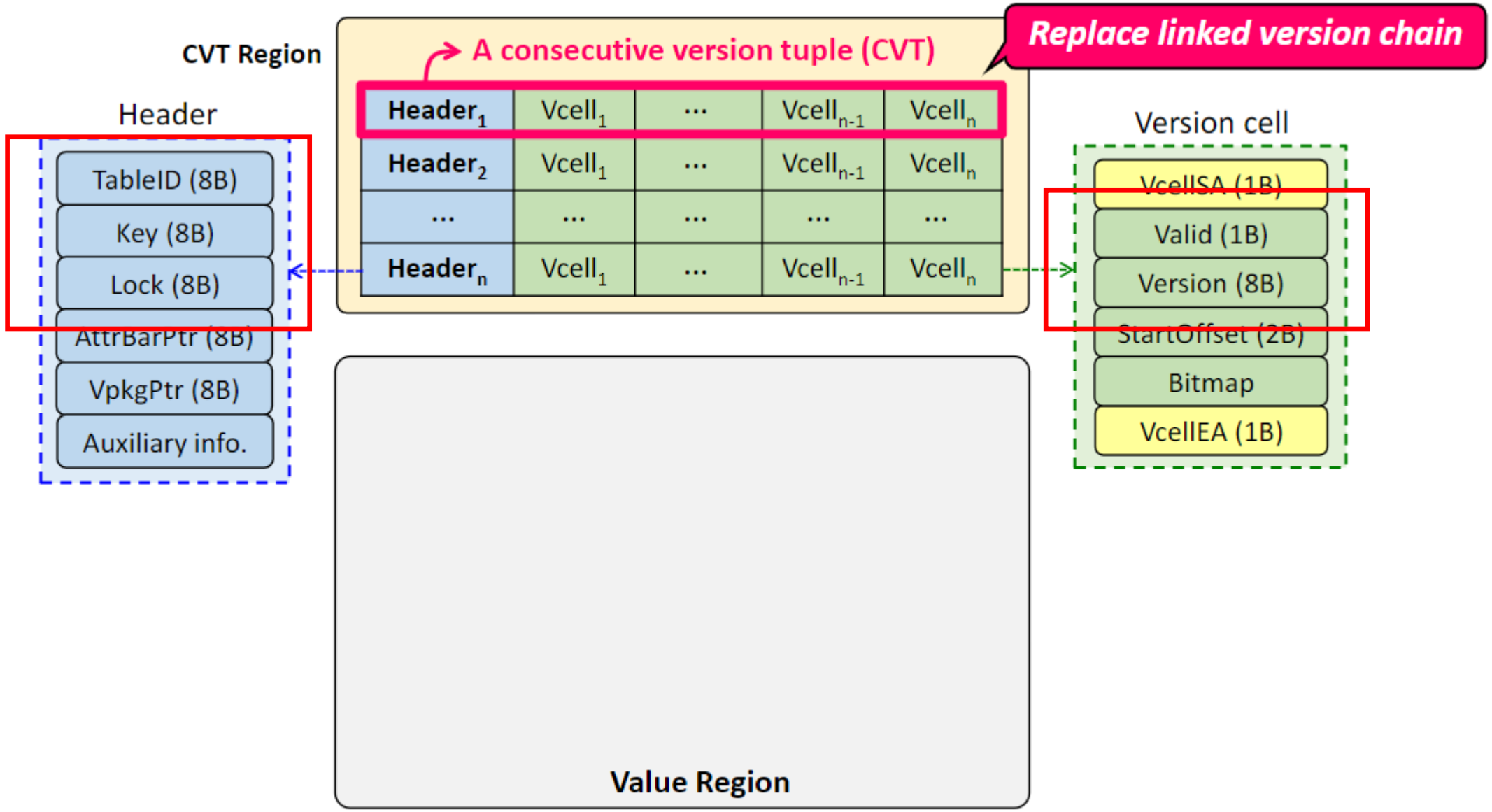


System Overview

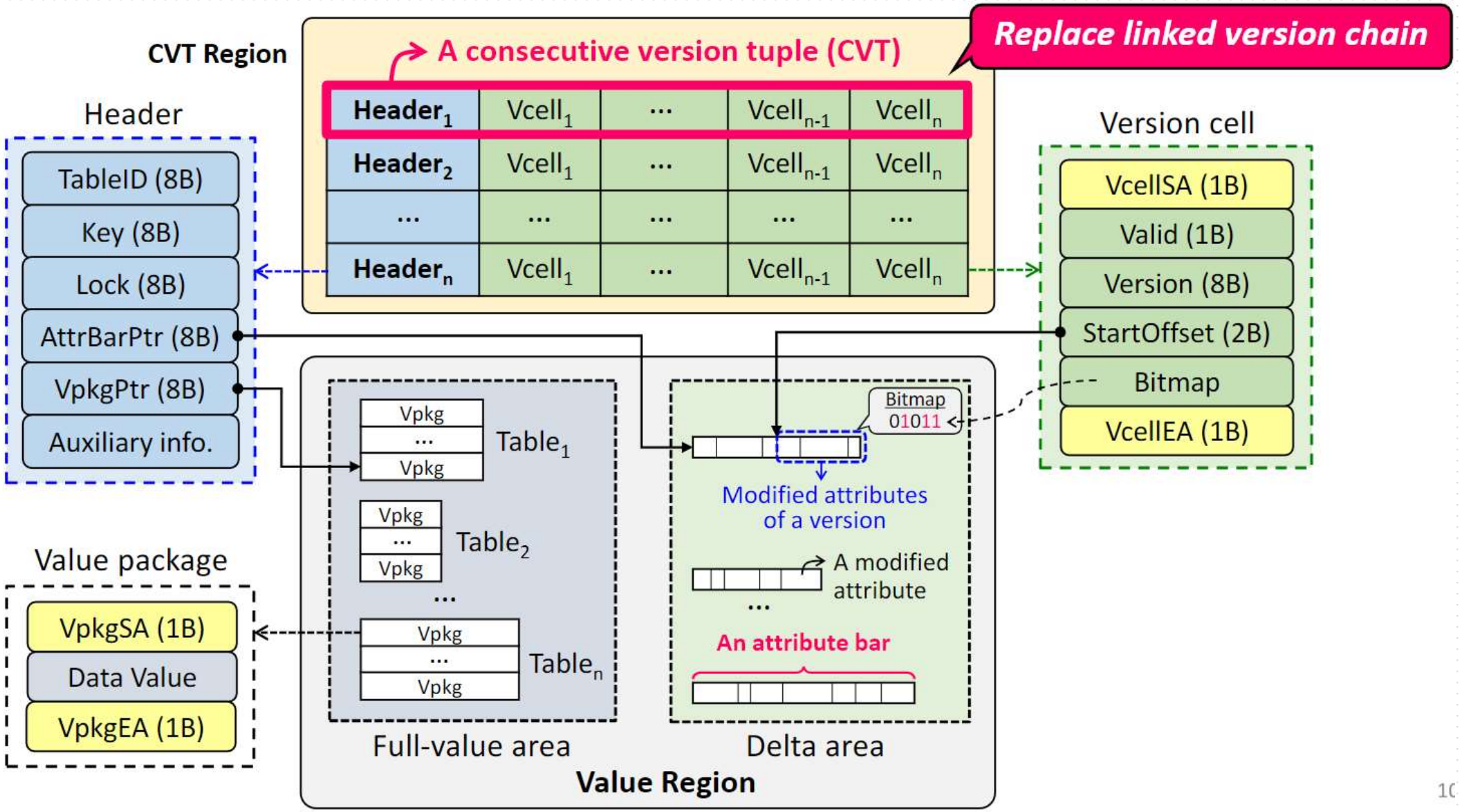
Consecutive Version Tuple



Consecutive Version Tuple



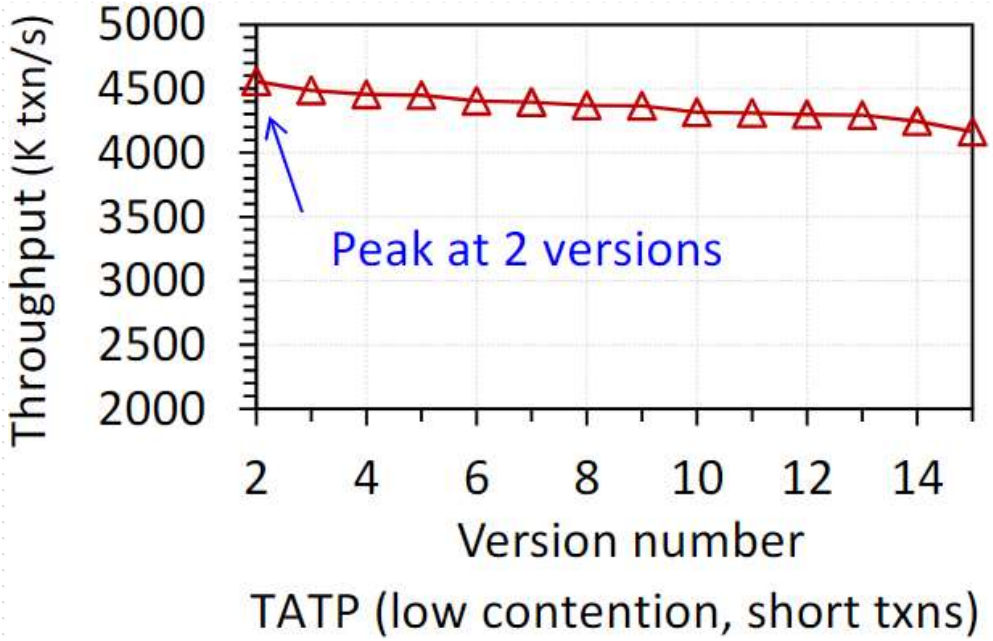
Reducing Memory Overhead



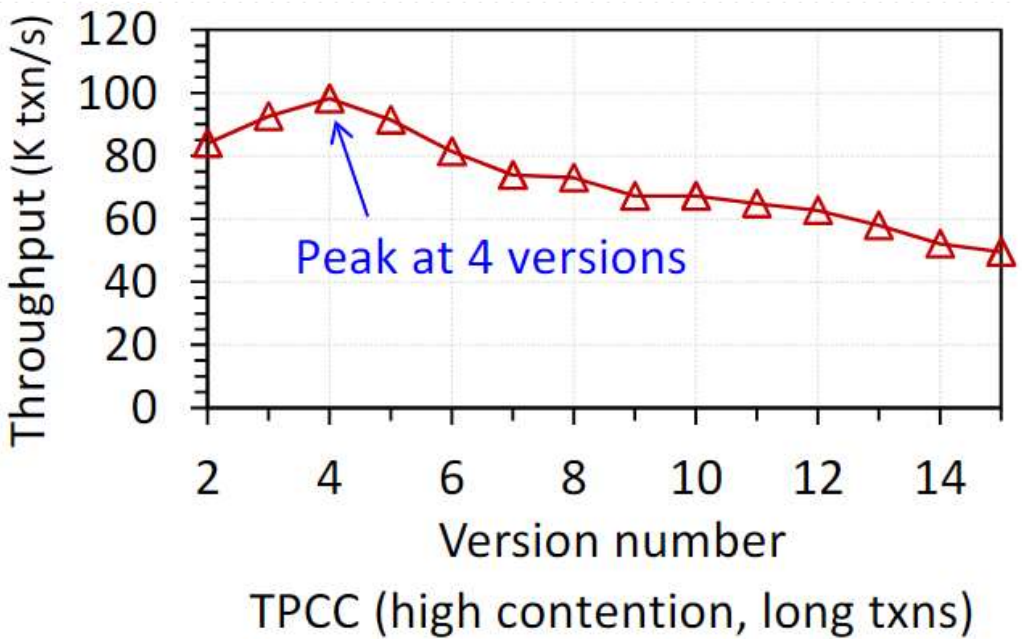
Number of versions to store

Number of Versions: depending on workload characteristics

- Read-write contention
- Number of accessed Records



TATP (low contention, short txns)



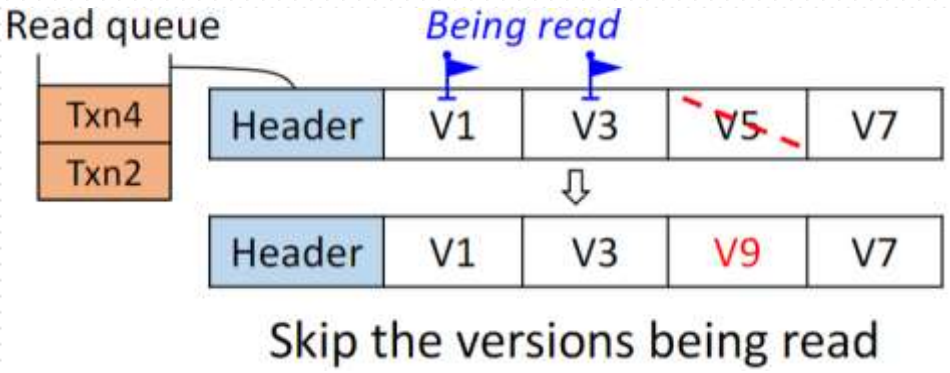
TPCC (high contention, long txns)

Coordinator-Active Garbage Collection

A CVT runs out of space - GC required

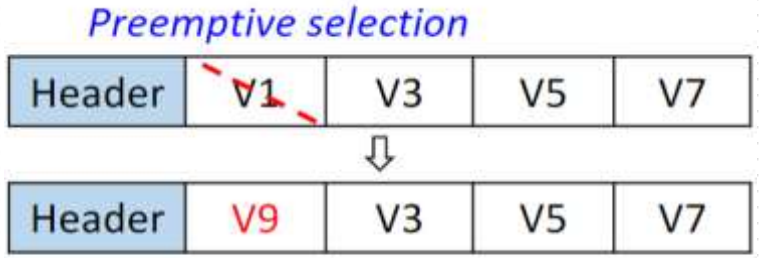
Prior systems track transaction states

- CPU in memory nodes is too weak to frequently track



Skip the versions being read

High overhead for compute nodes to maintain remote states

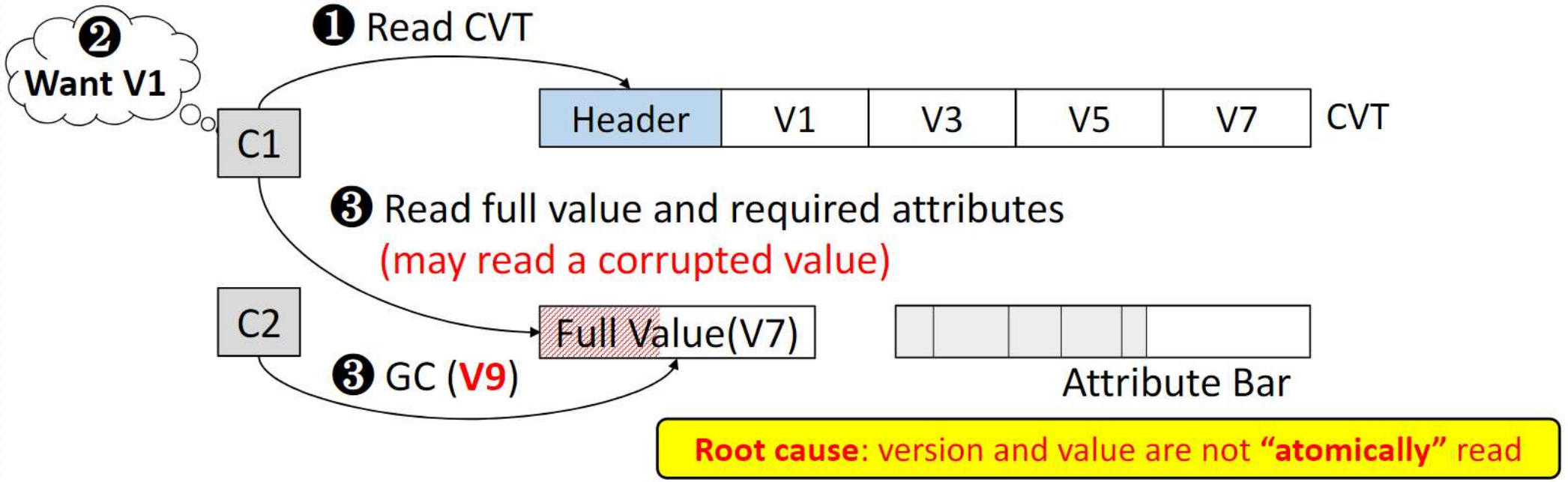


Overwrite the oldest version

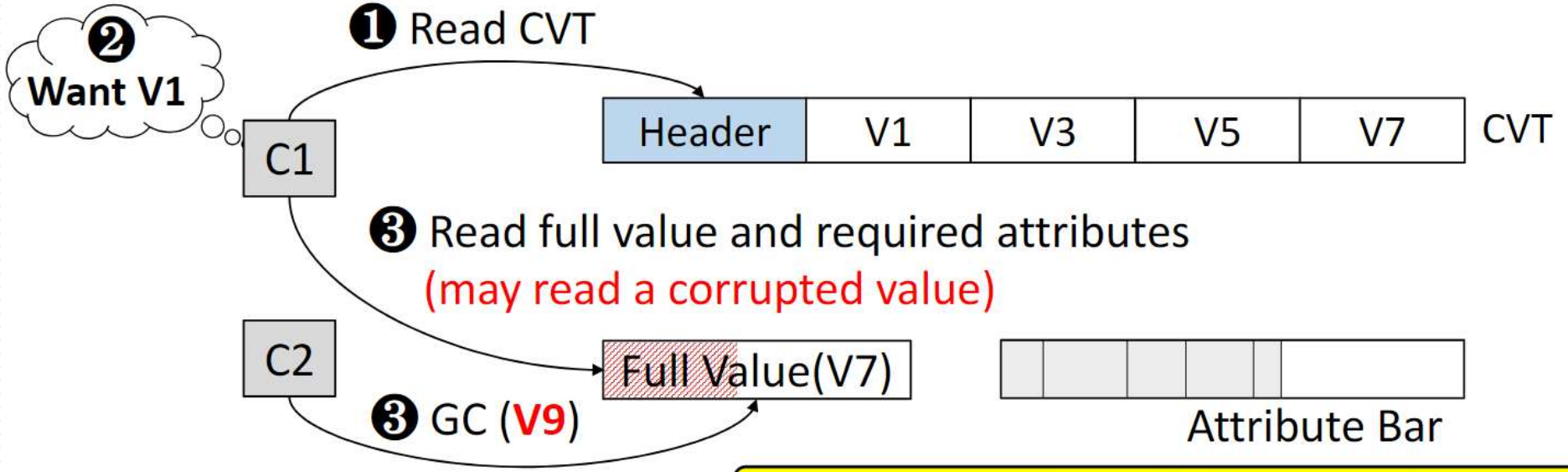
Simple, no tracking

Low abort rate with fast RDMA

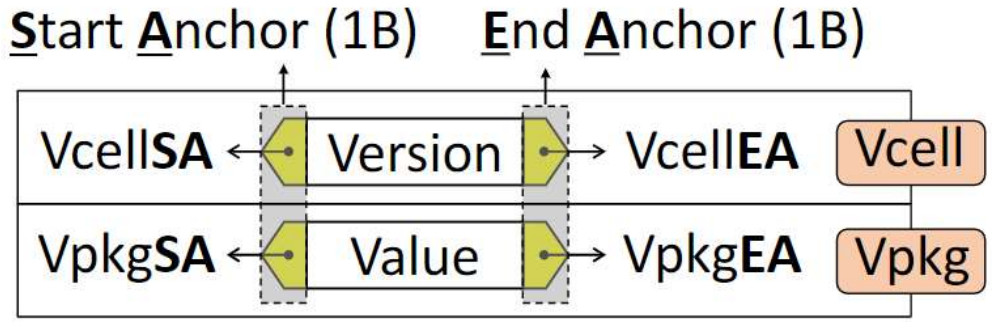
Anchor-Assisted Read



Anchor-Assisted Read

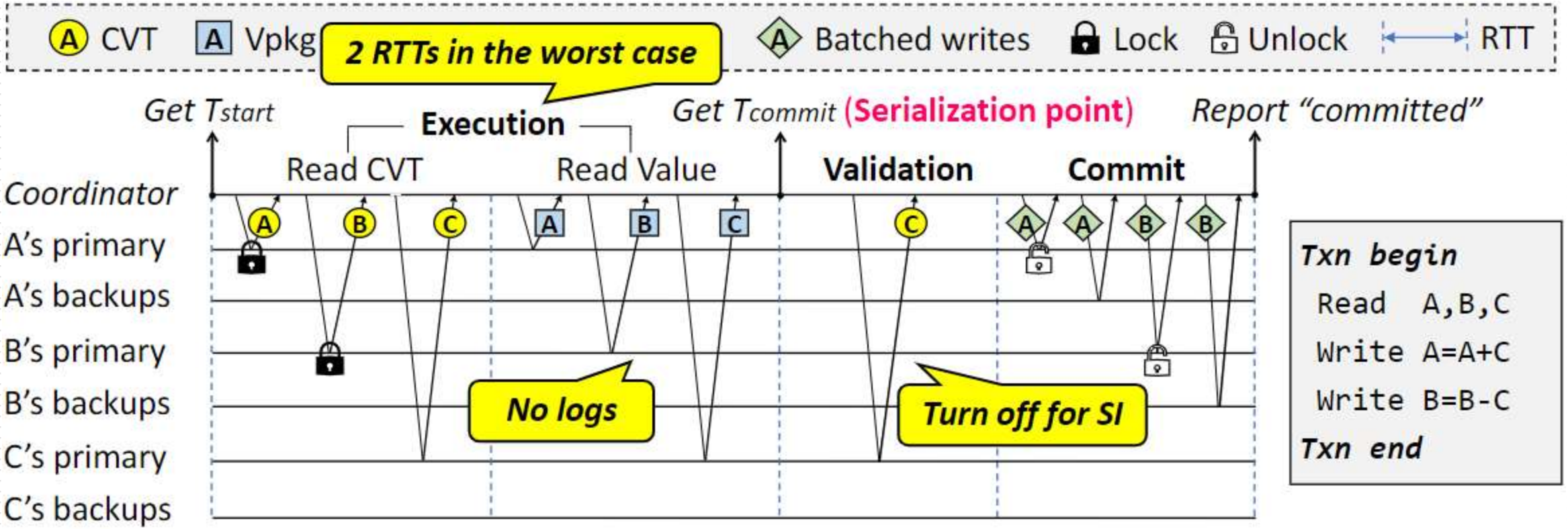


Root cause: version and value are not **"atomically"** read



- **Writer:** Vpkg → attributes → Vcell
- **Reader:** check "all anchors are equal"
  VcellSA = VcellEA = VpkgSA = VpkgEA ✓

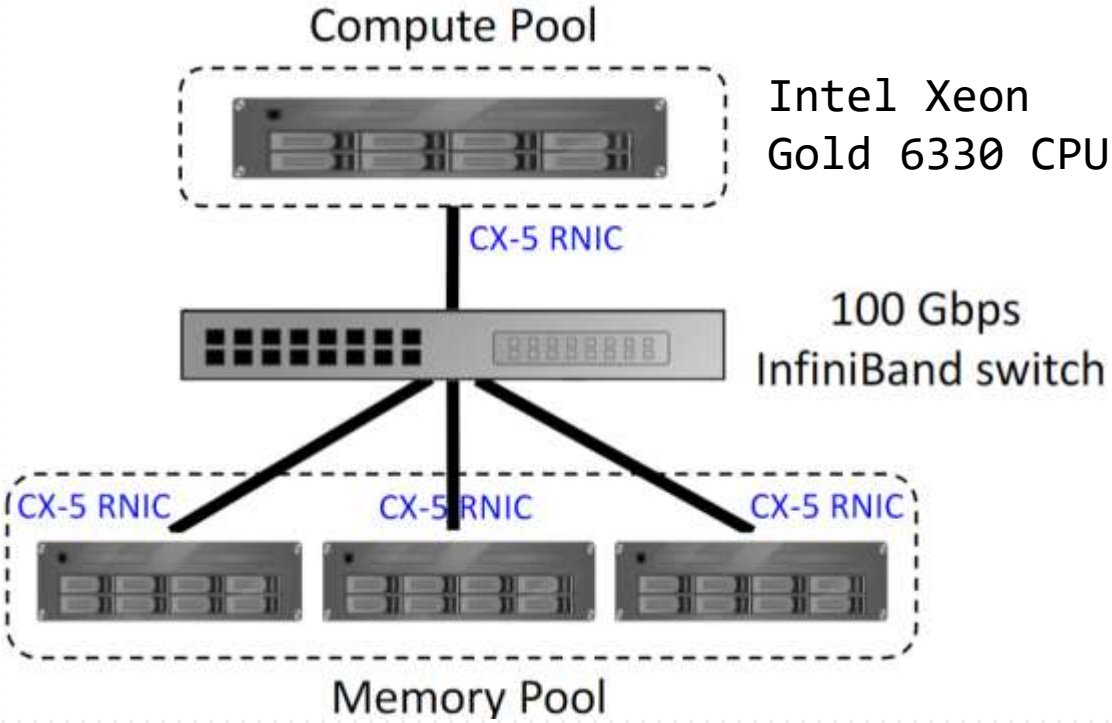
One-Sided RDMA-Based MVCC



Evaluation

Workloads

- KV Store
  - 8B key + 40B value
  - Skewed (skewness tunable)
- TATP
  - RO/RW: 80%/20%, max 48B
- SmallBank
  - RO/RW: 15%/85%, 16B
- TPCC
  - RO/RW: 8%/92%, max 672B



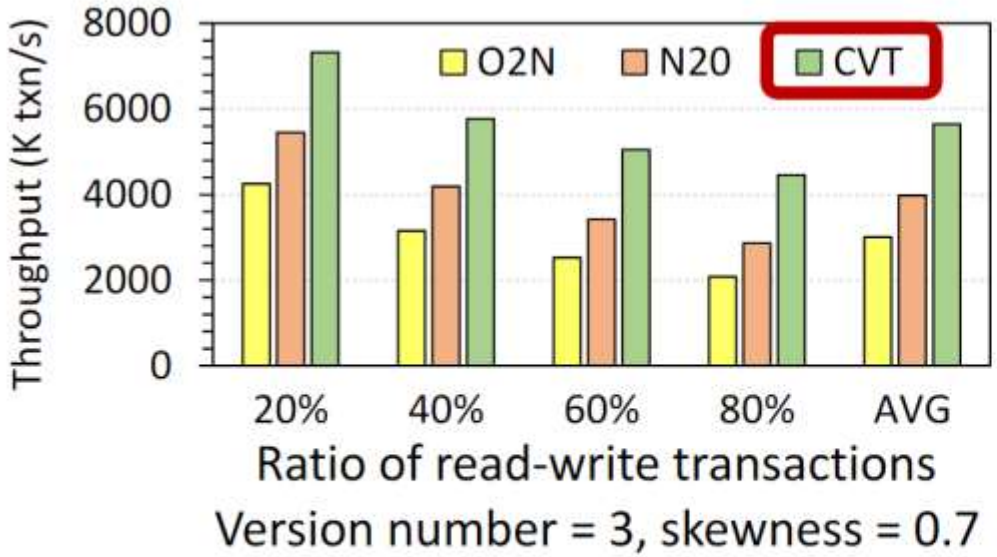
Comparisons

- FaRMv2@SIGMOD'19 (referred as FaRMv2-DM)
- FORD@FAST'22

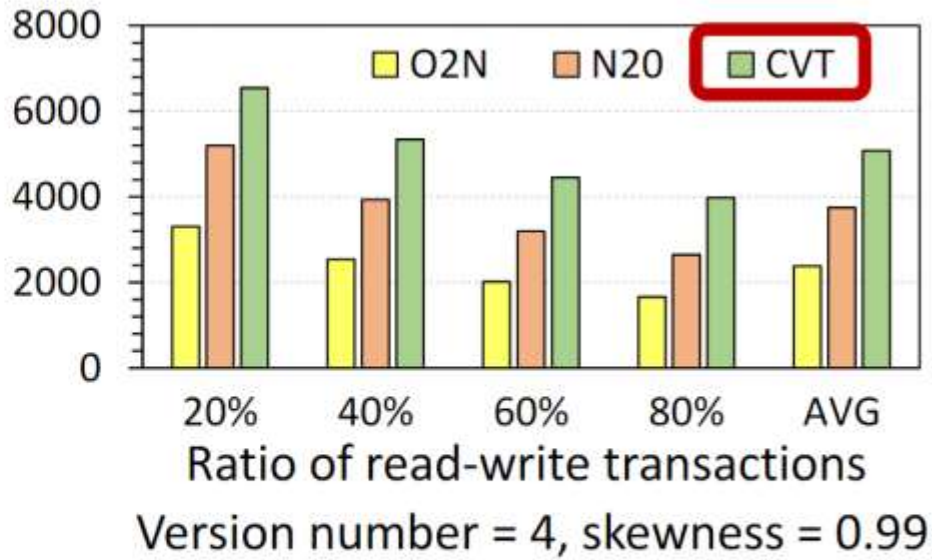
Performance of Version Structures

USTC, CHINA ADSLAB

KV Store



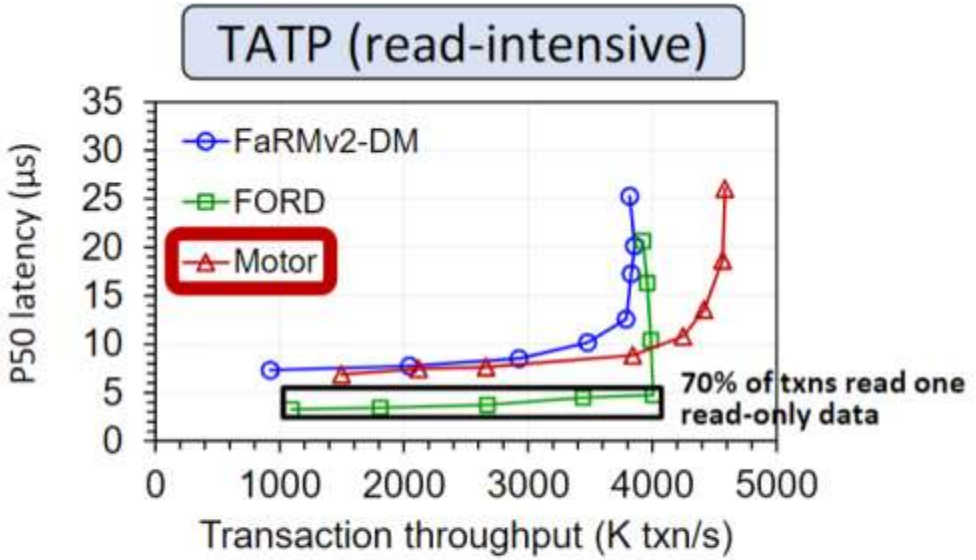
Version number = 3, skewness = 0.7

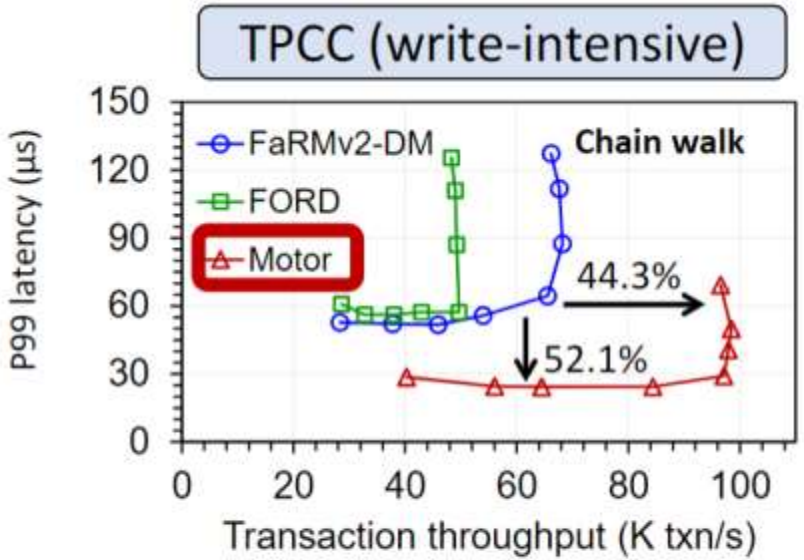


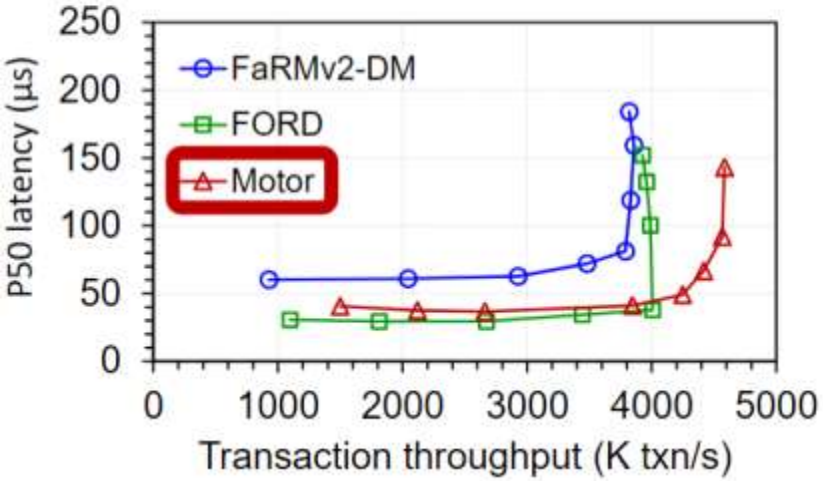
Version number = 4, skewness = 0.99

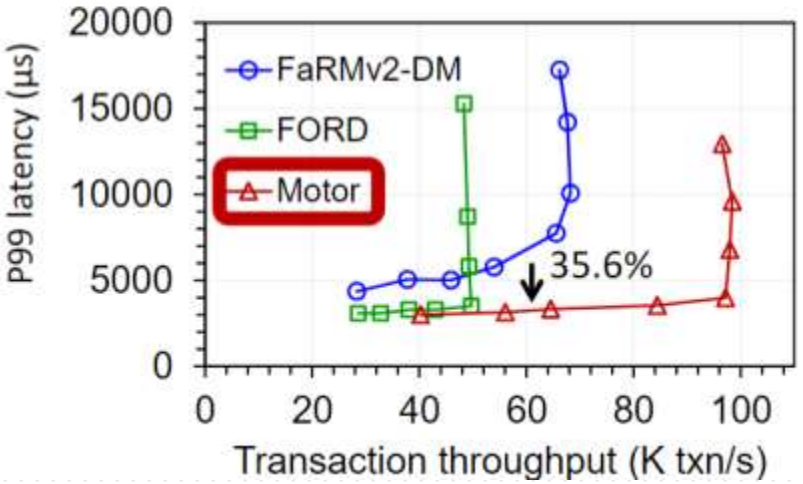
CVT improves throughput by
- 1.7–2.4x over O2N
- 1.3–1.6x over N2O

End-to-end Performance

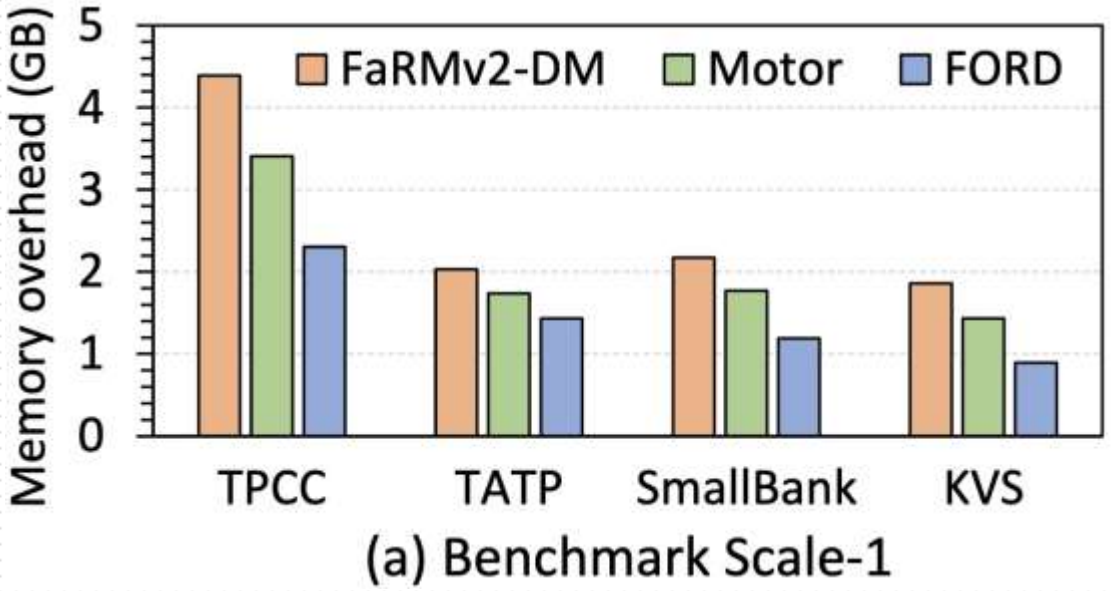




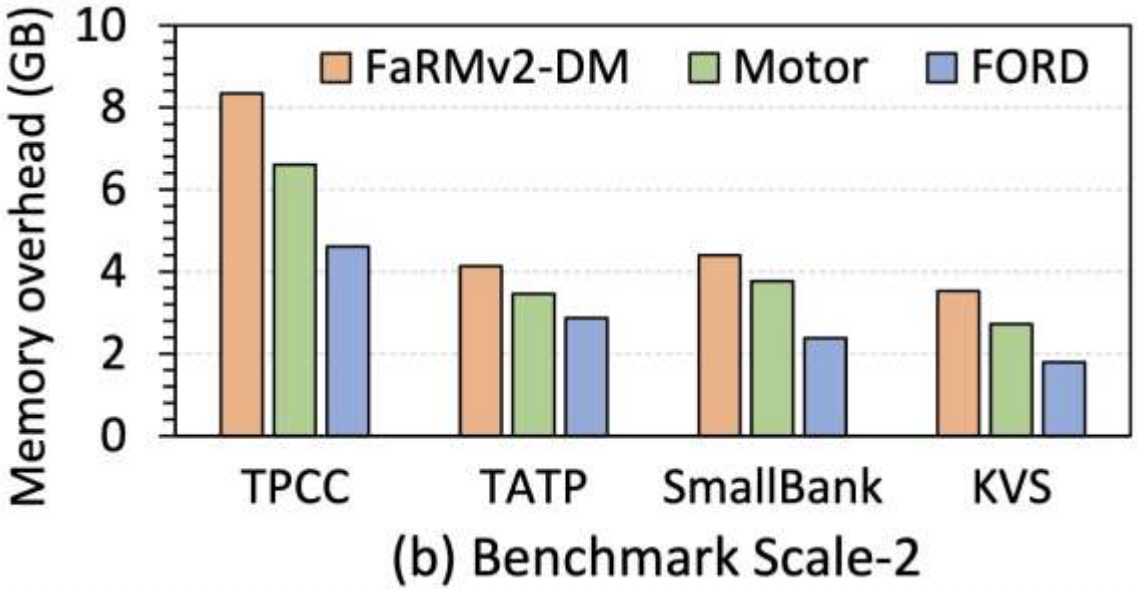




Memory Overhead



(a) Benchmark Scale-1



(b) Benchmark Scale-2

Conclusion

USTC, CHINA ADSLAB

Existing multi-version distributed transactions do not fit DM

- Inefficient linked version chain
- Incompatible transaction protocol

Motor: a holistic multi-version design for DM

- Consecutive version tuple structure (memory pool)
- One-sided RDMA MVCC based on CVT (compute pool)

Benefits

High Throughput | Low Latency | Low Memory Overhead

https://github.com/minghust/motor

Motor: Enabling Multi-Versioning for Distributed Transactions on Disaggregated Memory

Ming Zhang, Yu Hua, Zhijun Yang
Huazhong University of Science and Technology, China

Thanks you for your attention