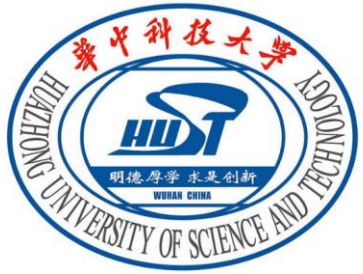


ROLEX: A Scalable RDMA-oriented Learned Key-Value Store for Disaggregated Memory Systems

Pengfei Li, Yu Hua, Pengfei Zuo, Zhangyu Chen, Jiajie Sheng

Huazhong University of Science and Technology

The USENIX Conference on File and Storage Technologies (FAST), 2023

Disaggregated Memory Systems (DMS)

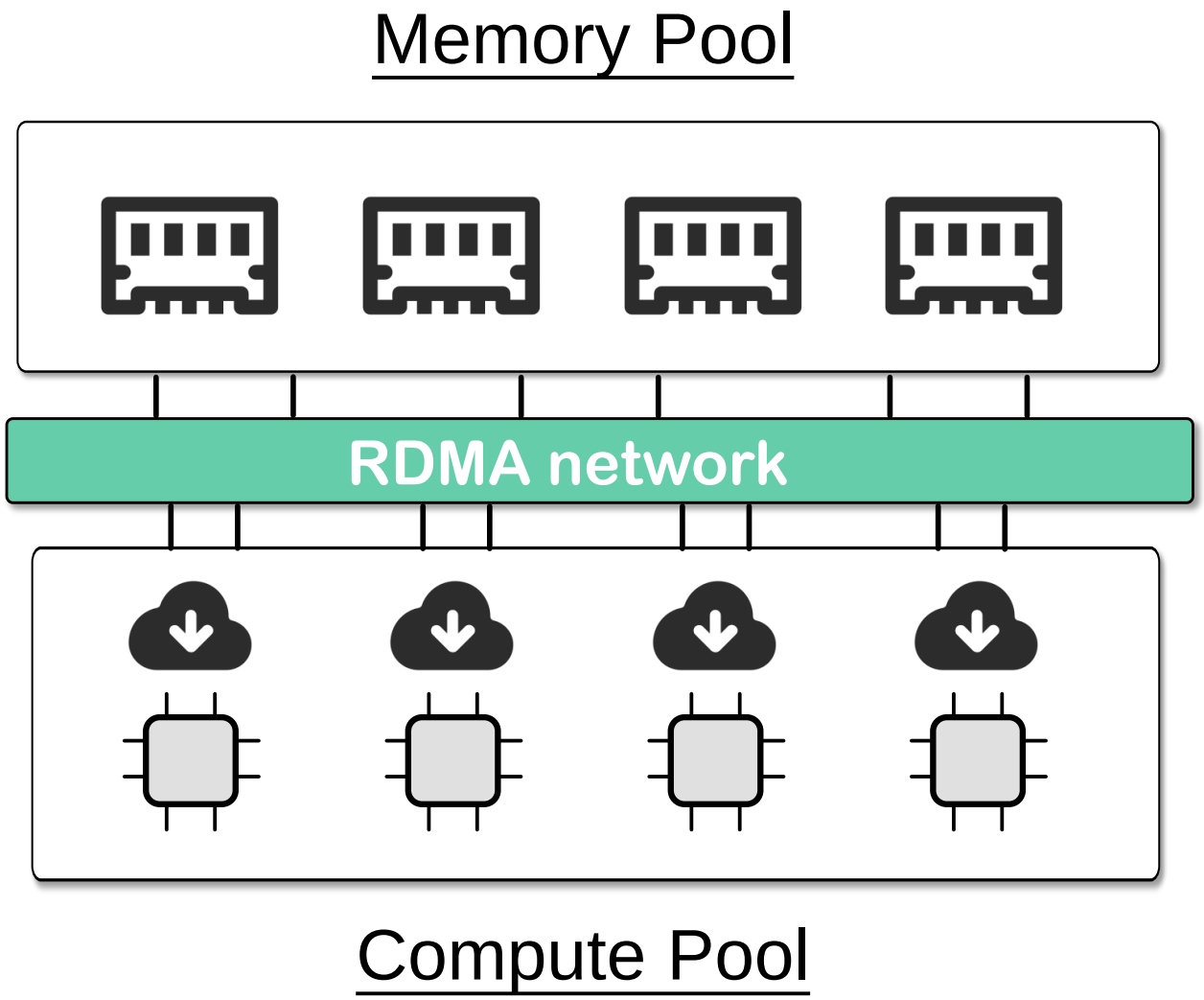
High resource utilization

Flexible hardware scalability

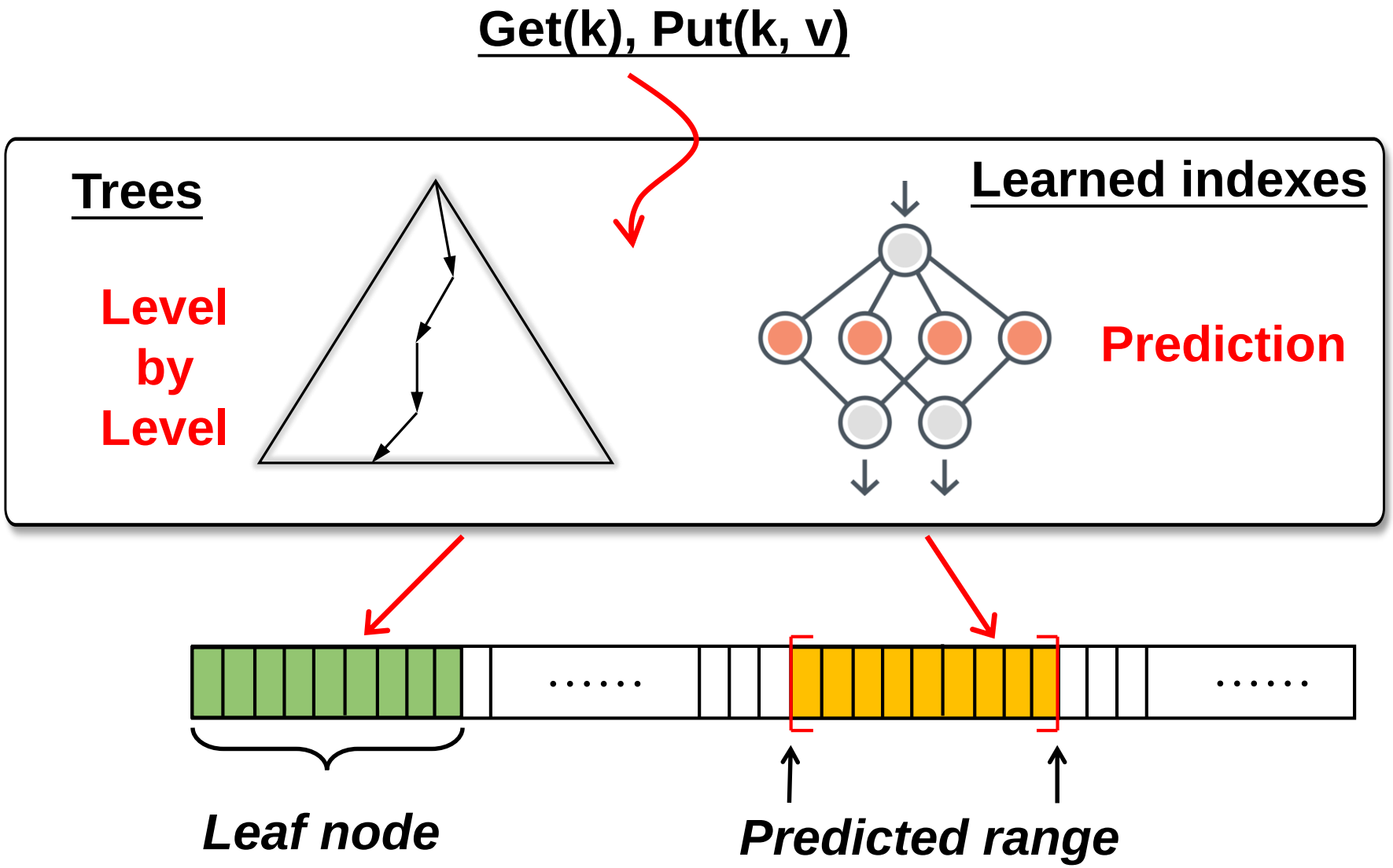
Efficient data sharing

40-400 Gbps

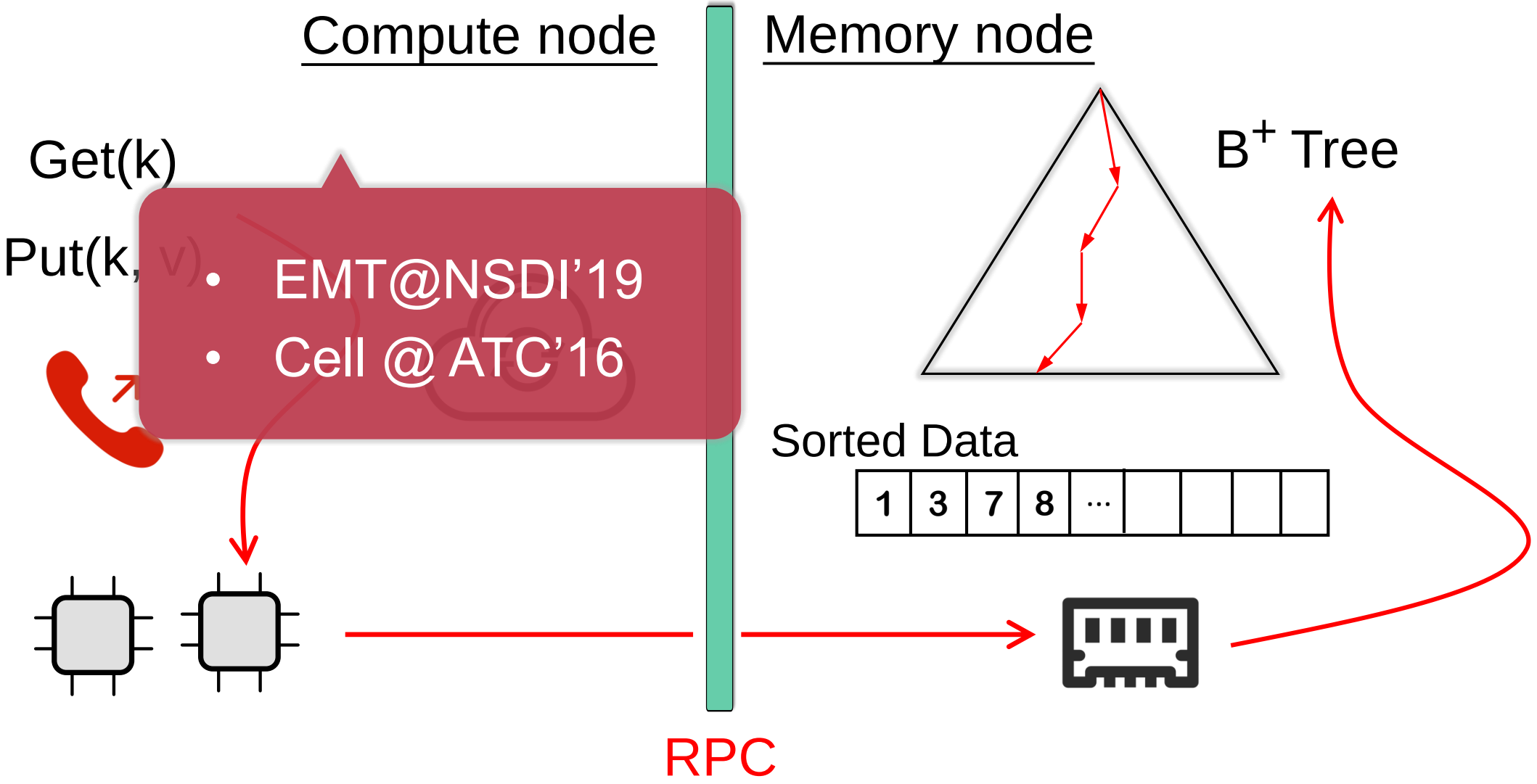
Remote CPU bypassing



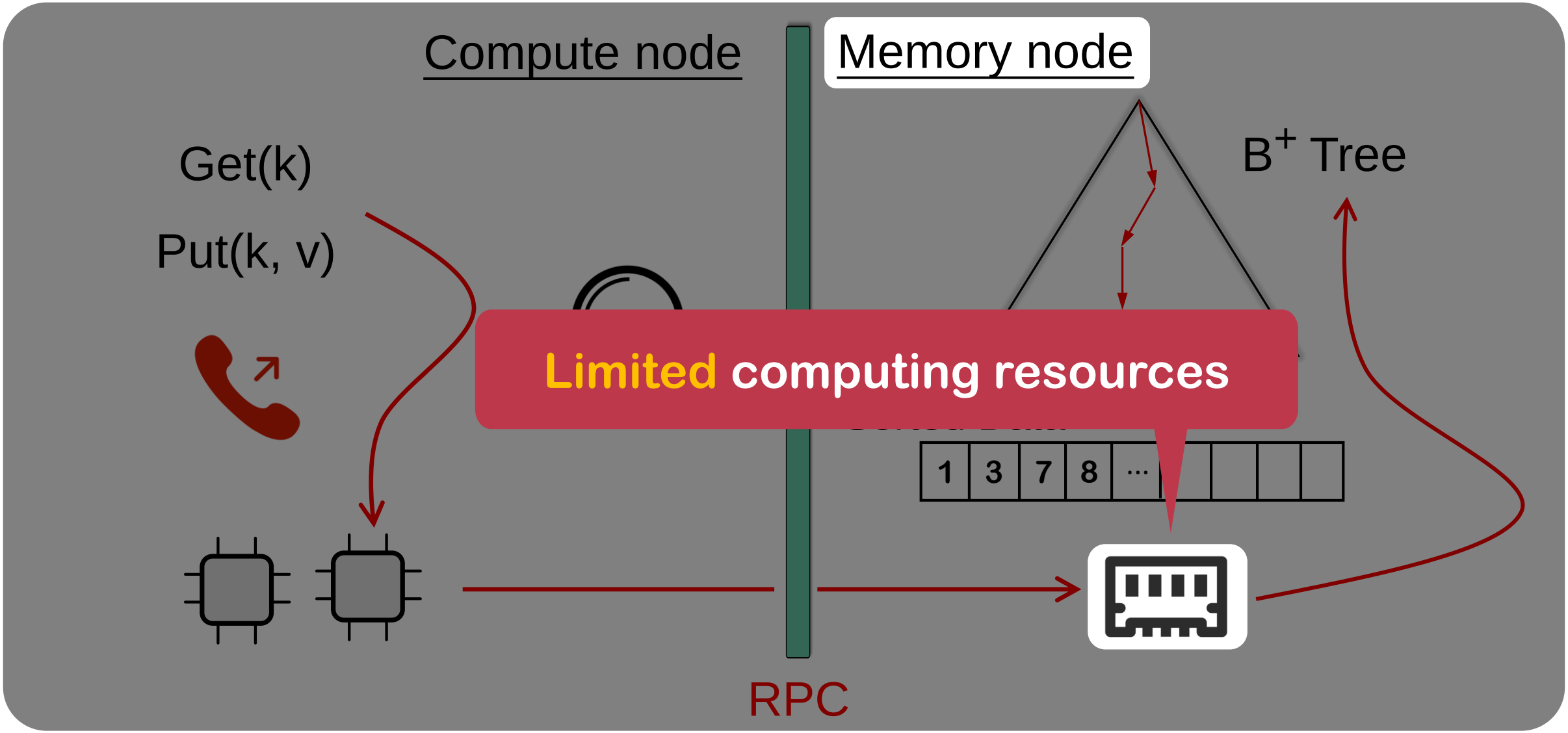
Ordered KV Store



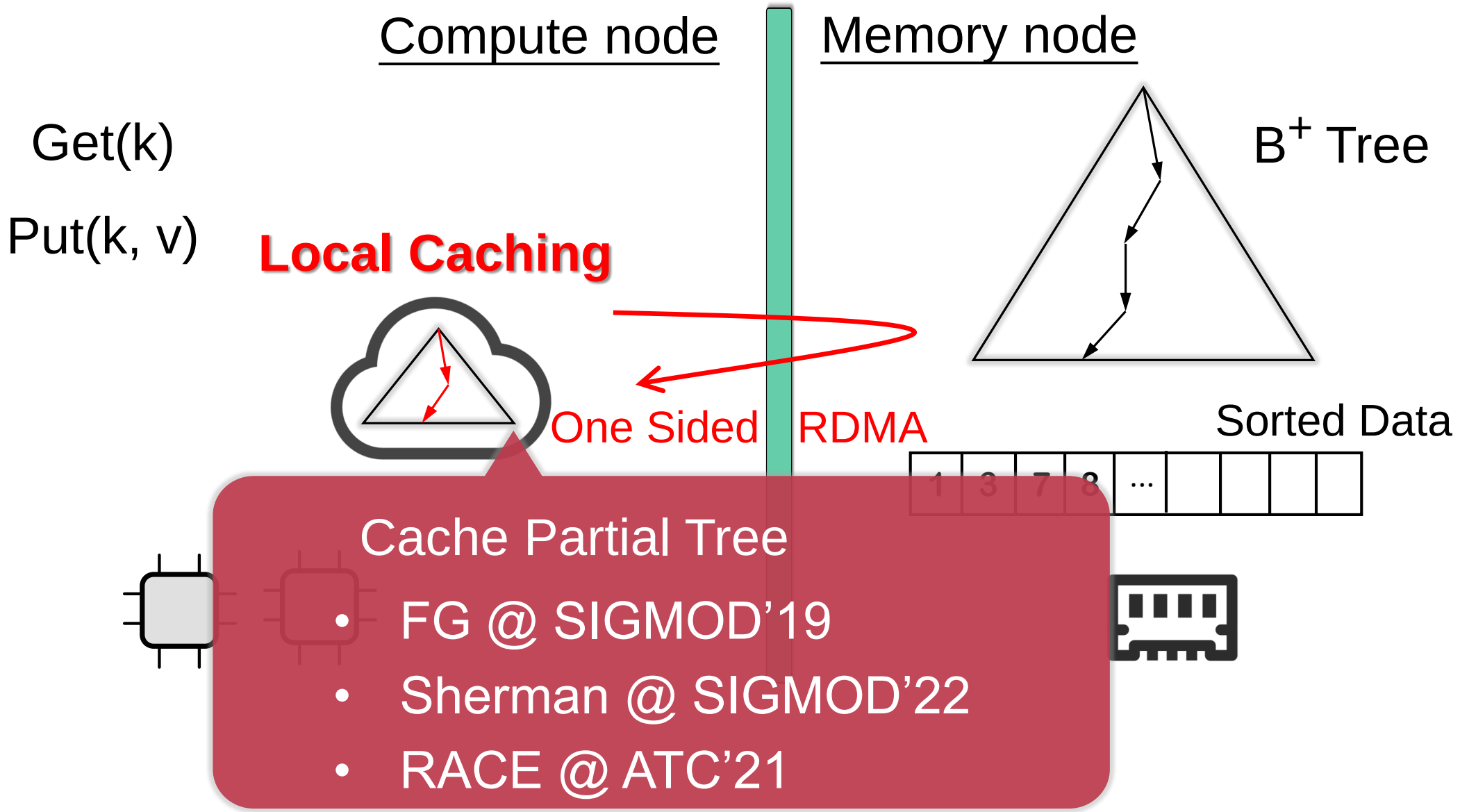
Tree-based Structures in DMS – Two Sided



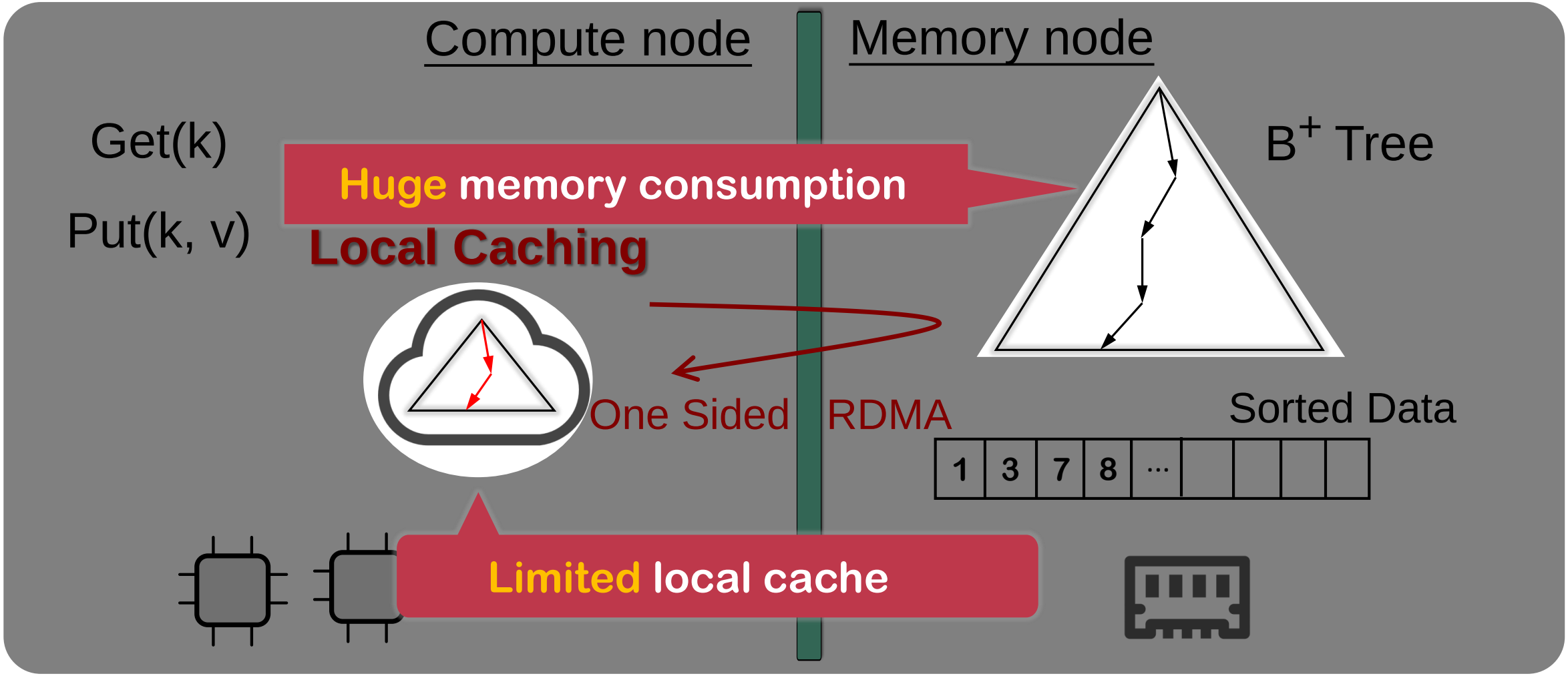
Tree-based Structures in DMS – Two Sided



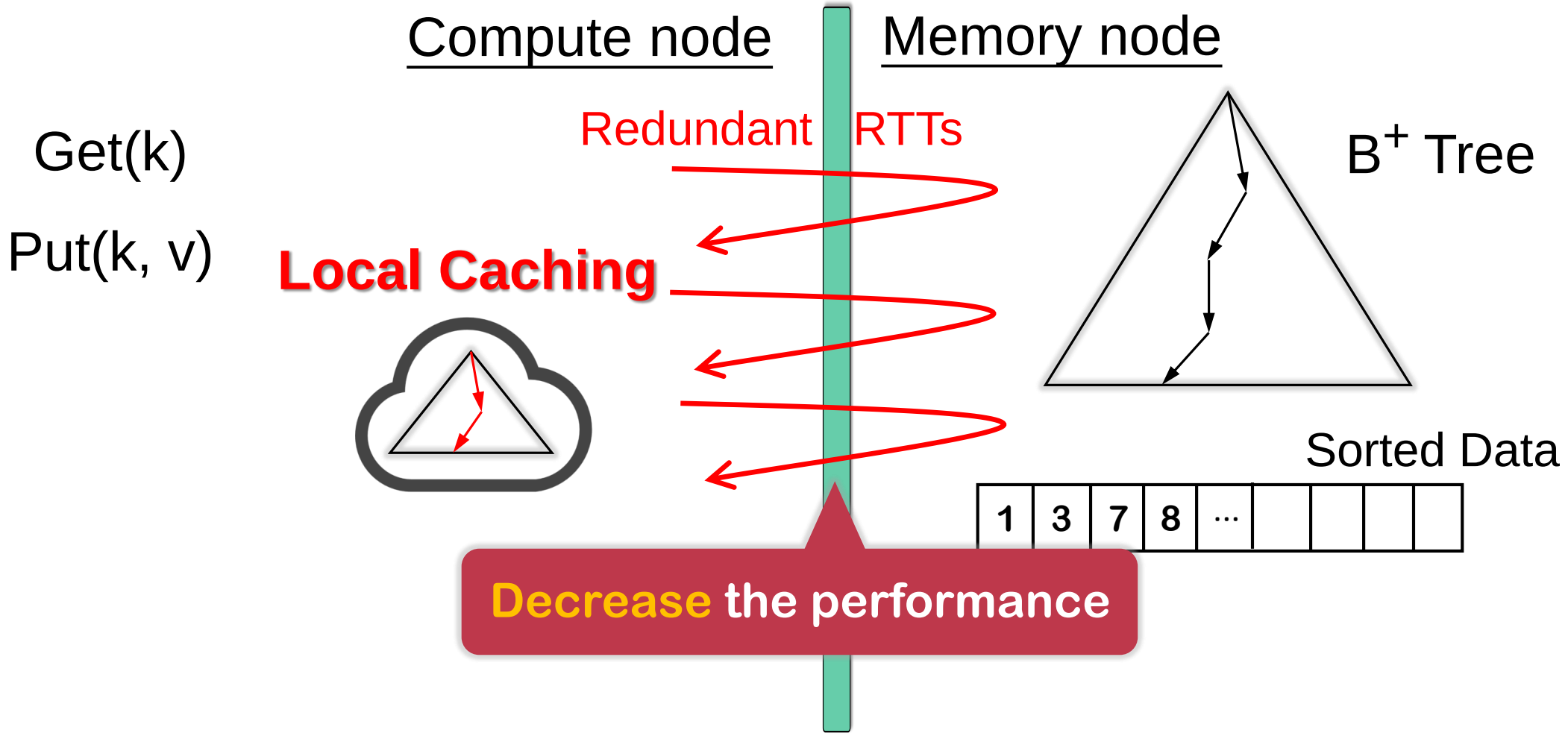
Tree-based Structures in DMS – One Sided



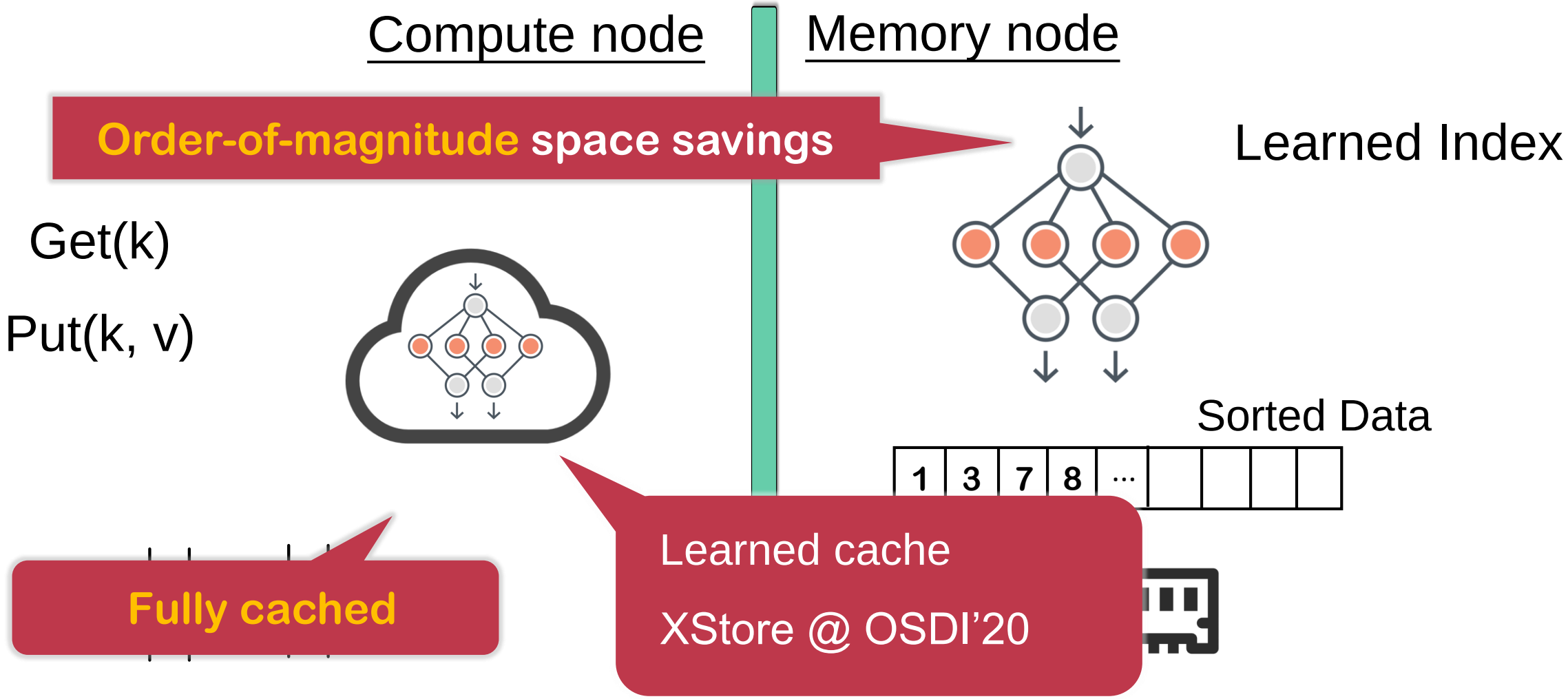
Tree-based Structures in DMS – One Sided



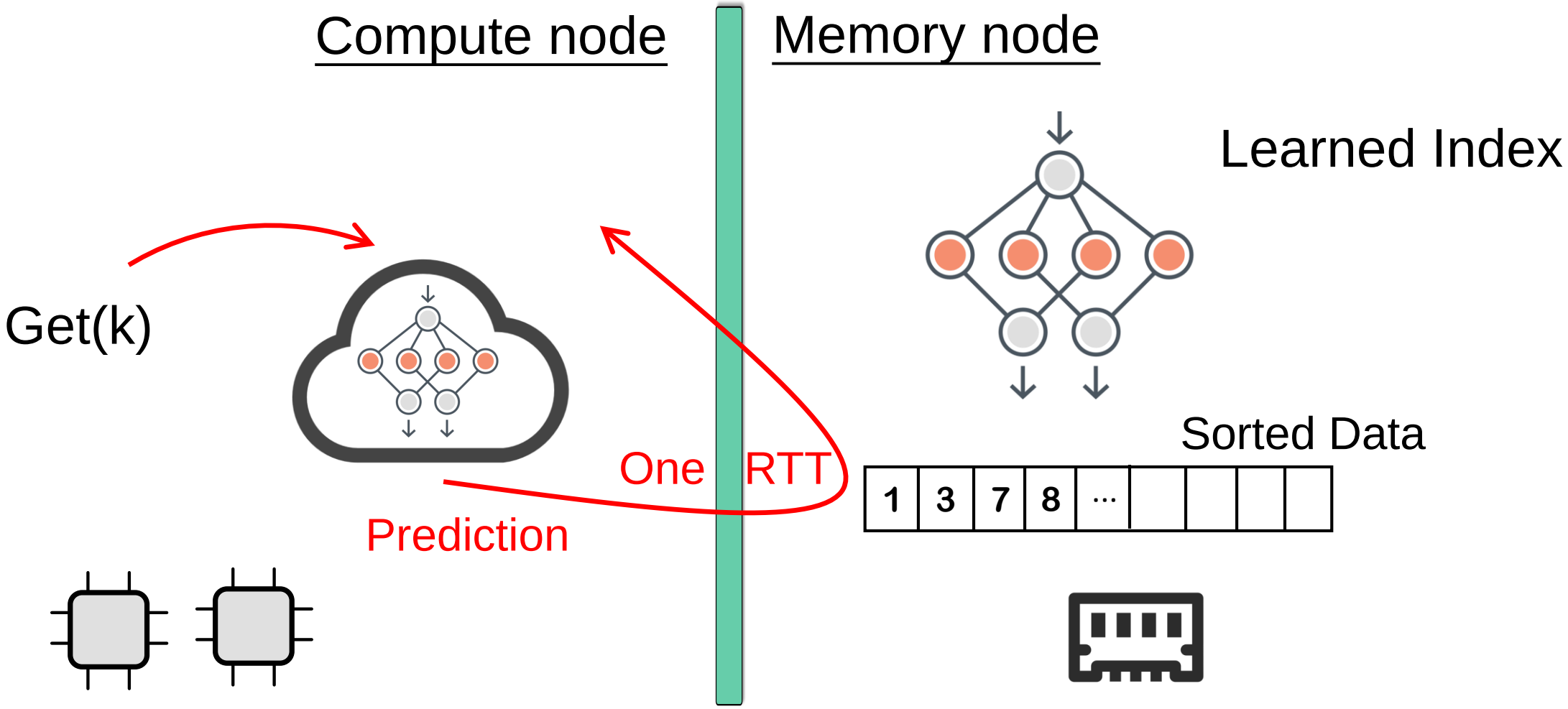
Tree-based Structures in DMS – One Sided



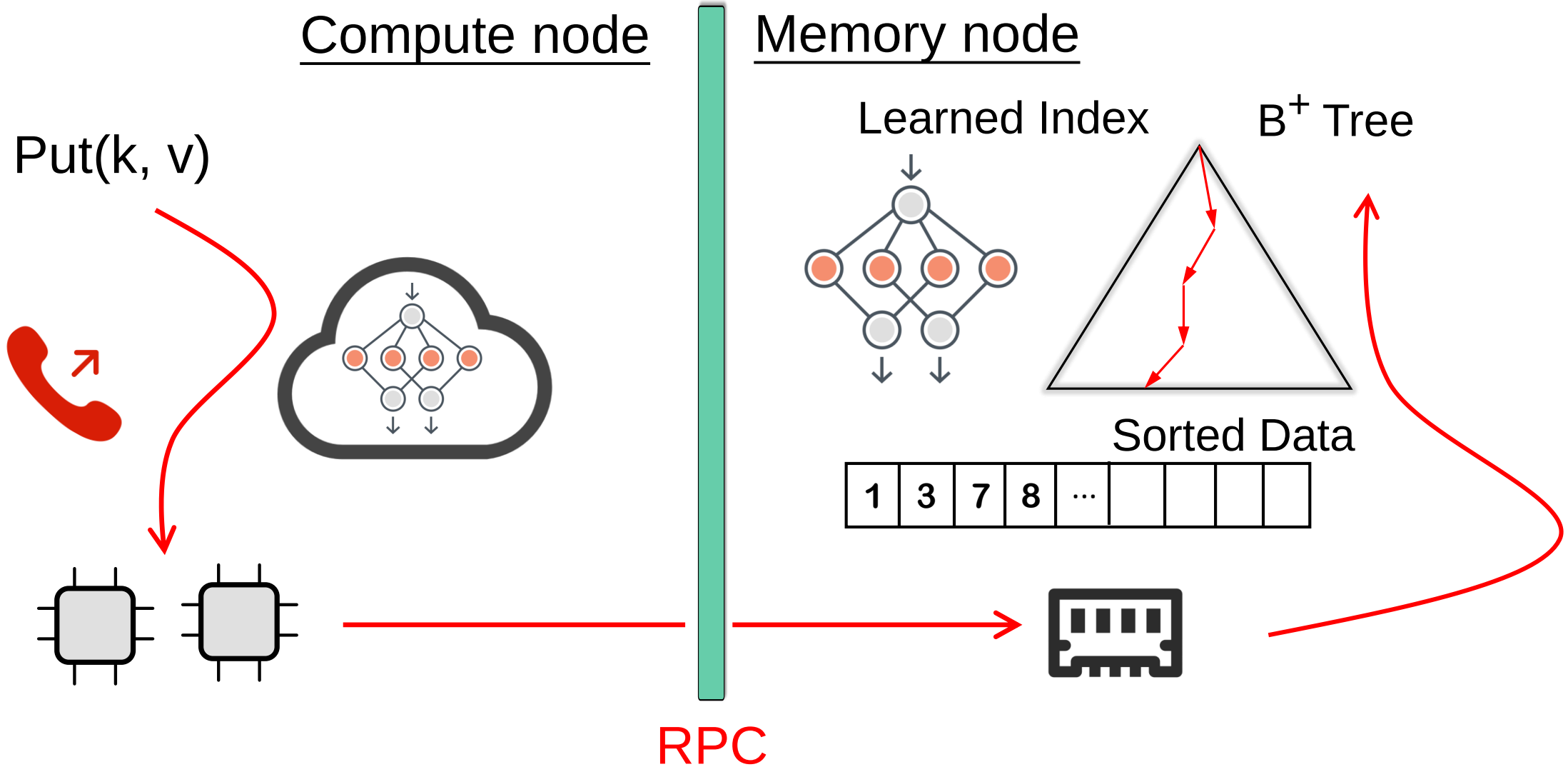
Learned-indexes in DMS



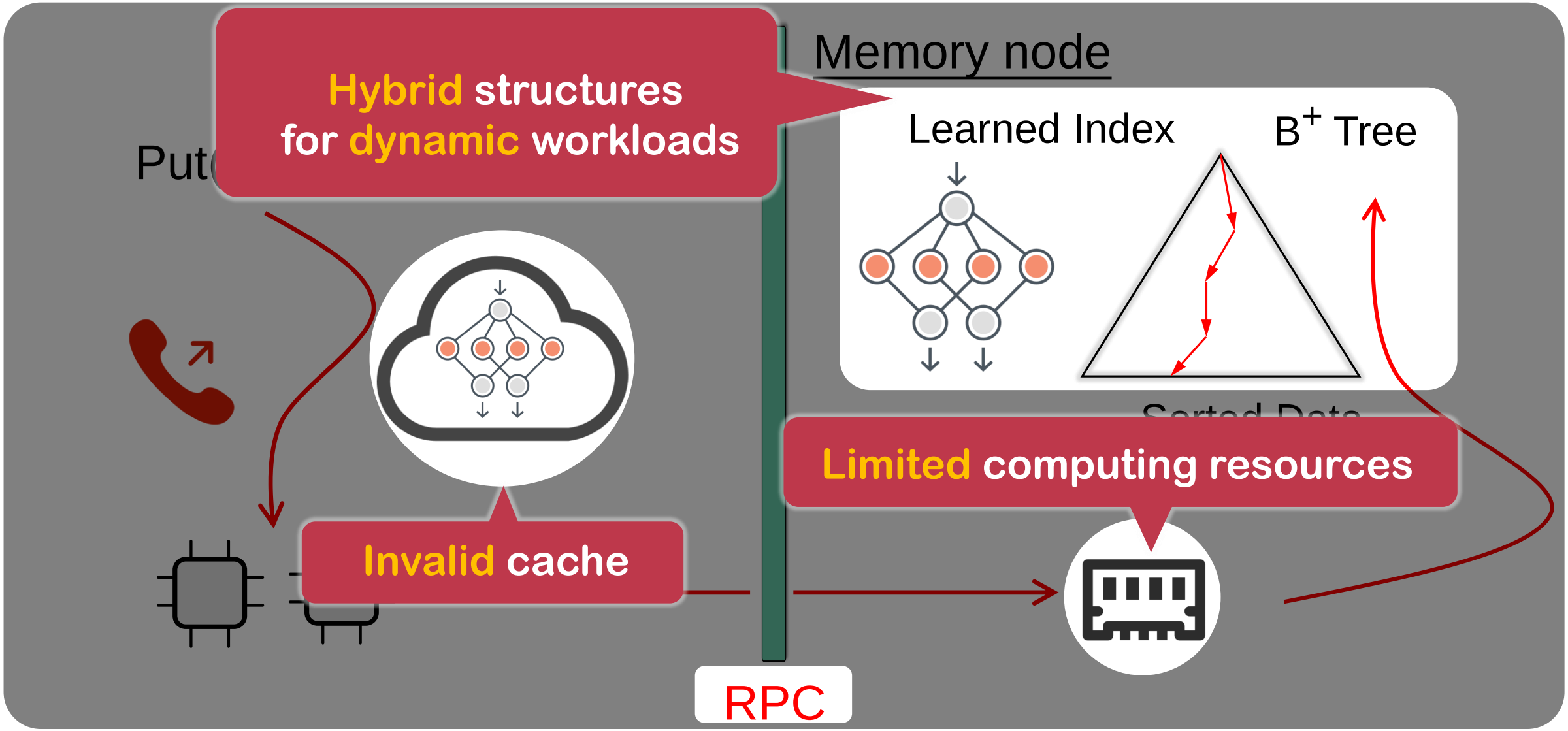
Learned-indexes in DMS



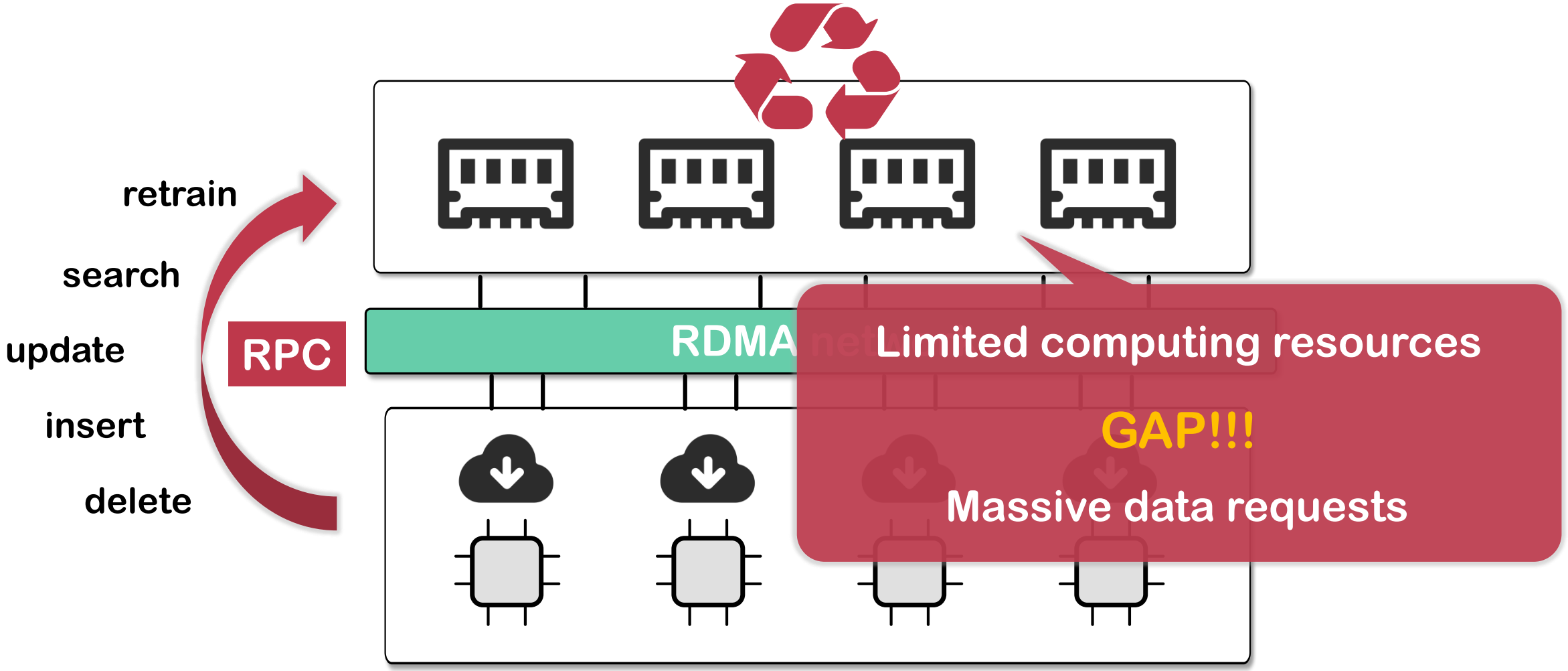
Learned-indexes in DMS



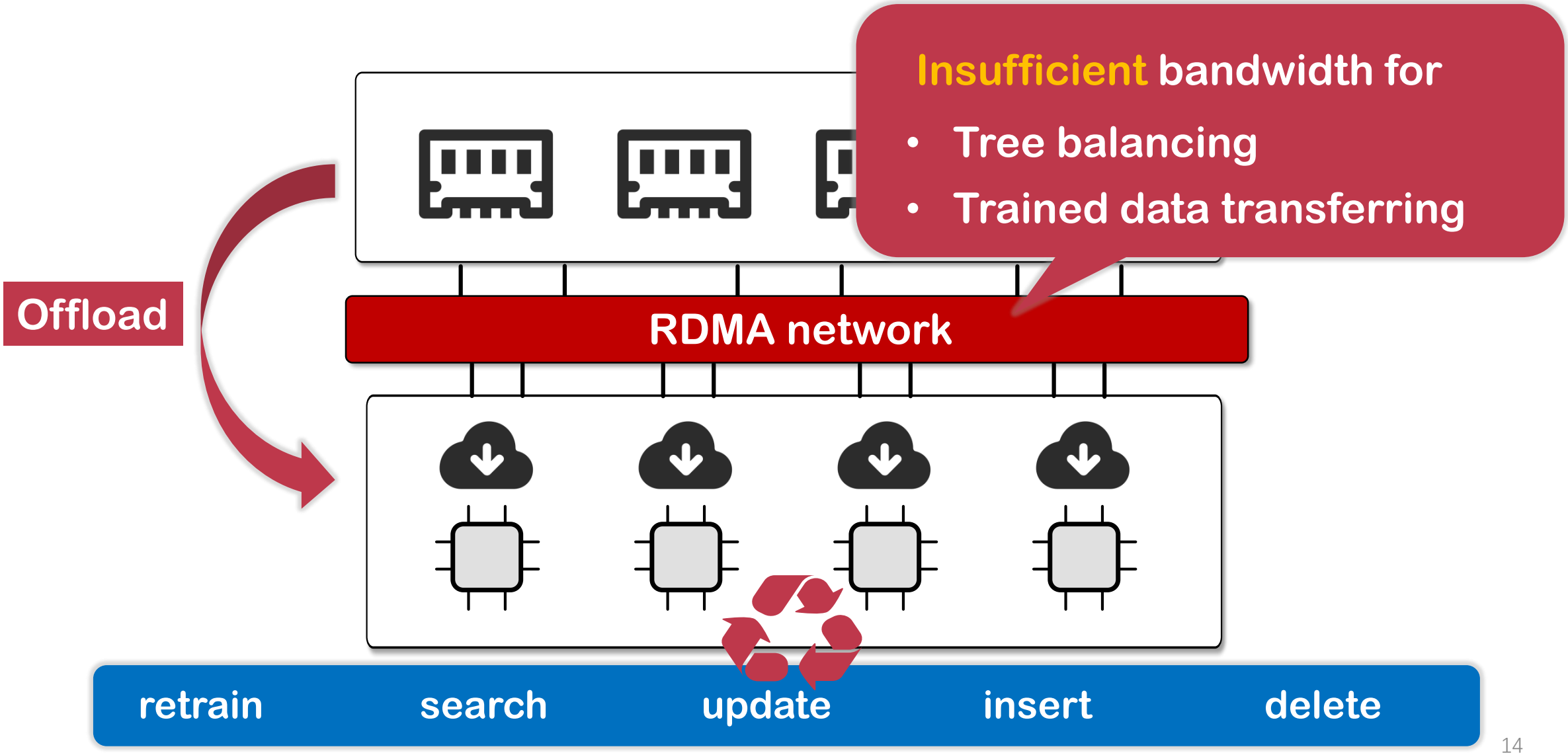
Learned-indexes in DMS



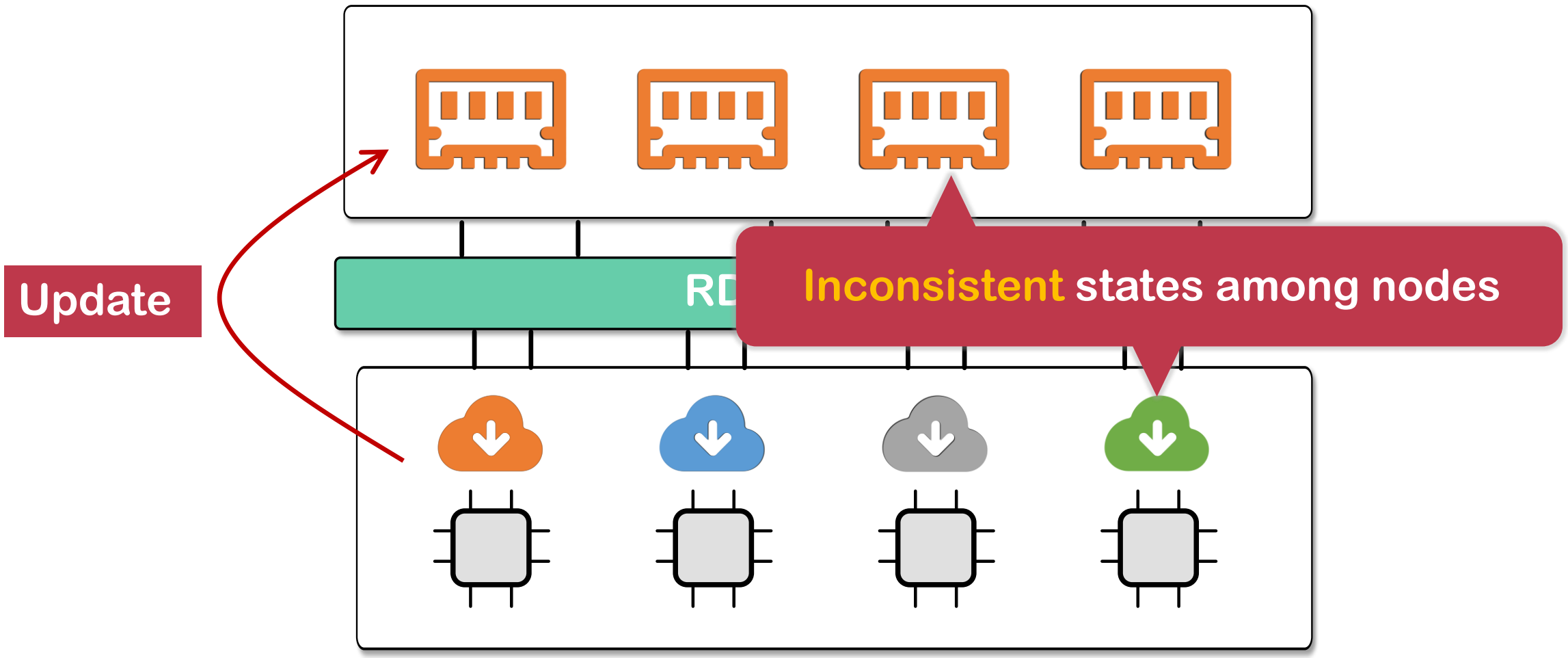
Challenge 1: Computing Bottleneck



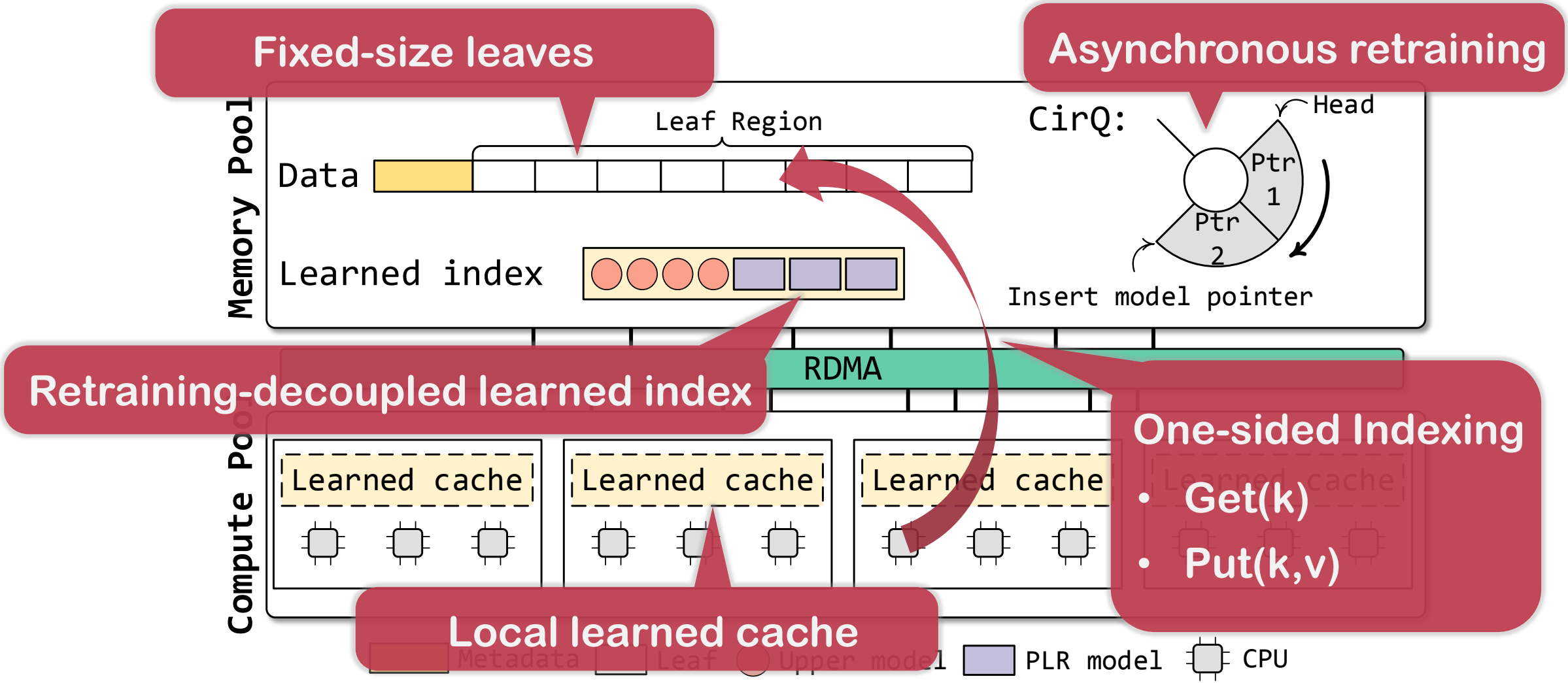
Challenge 2: Overloaded Bandwidth



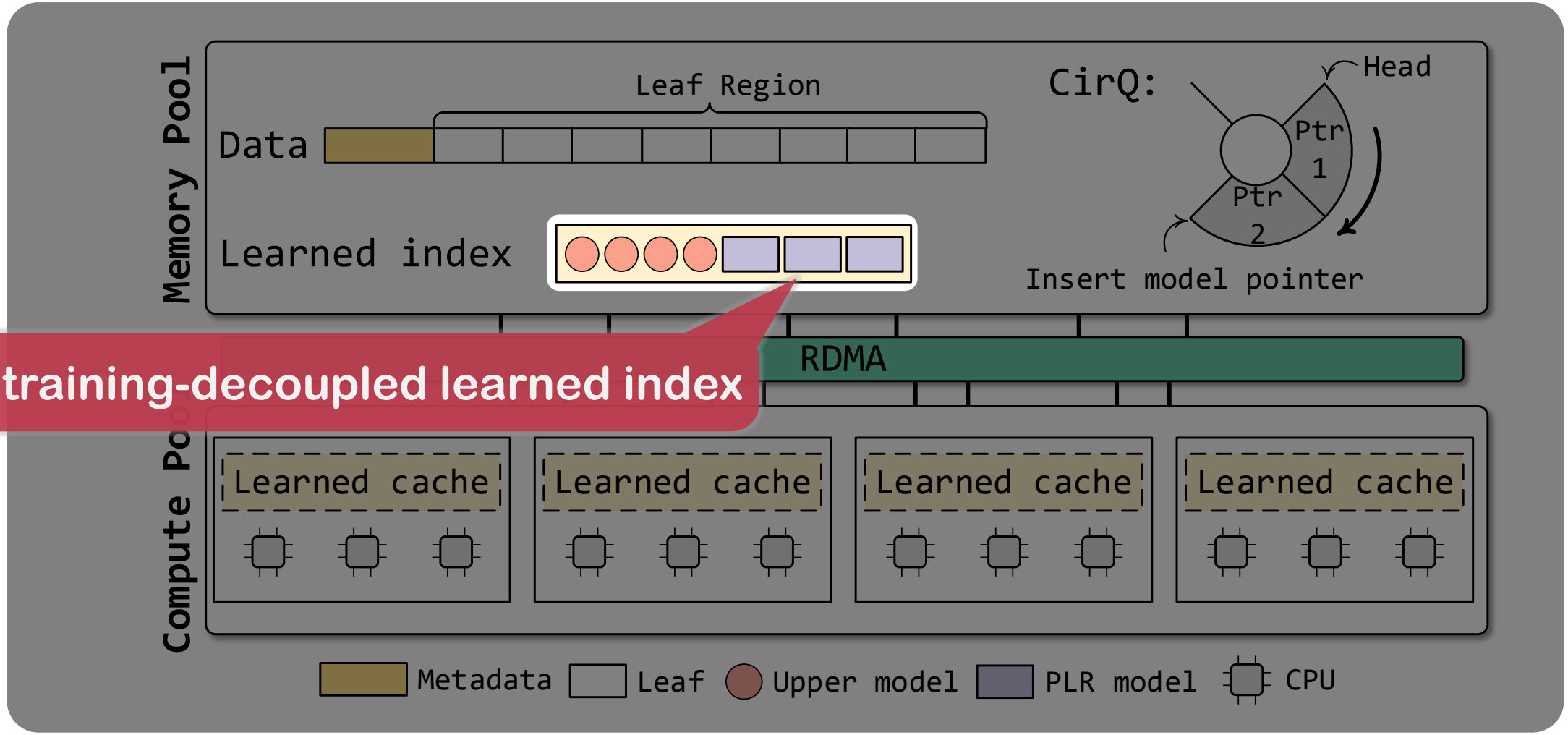
Challenge 3: Inconsistency Issues



RDMA-Oriented KVS Using Learned Indexes (ROLEX)

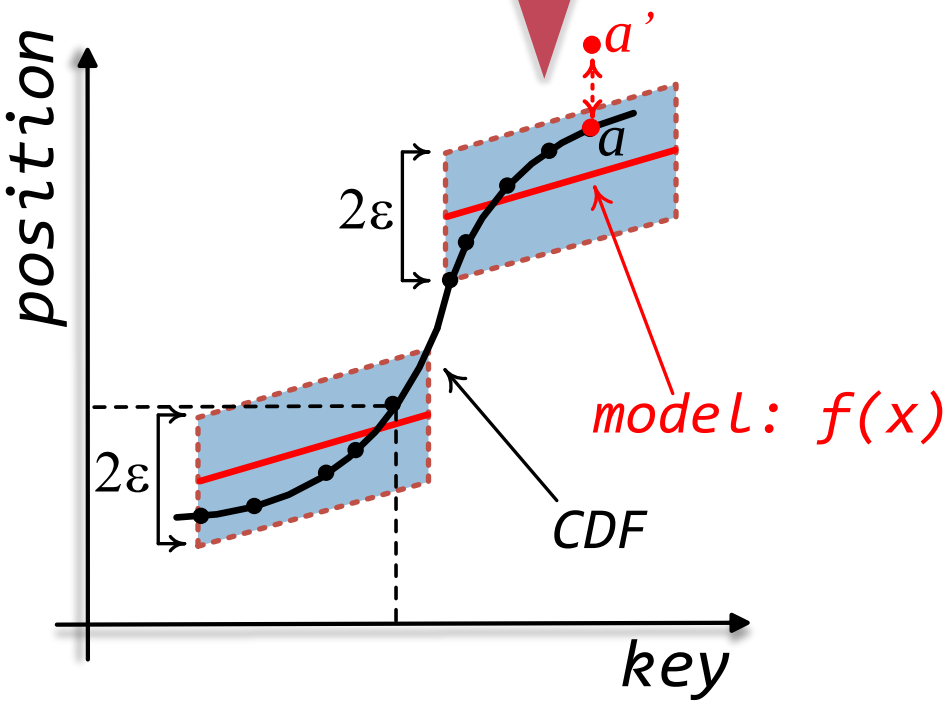


Retraining-Decoupled Learned Indexes

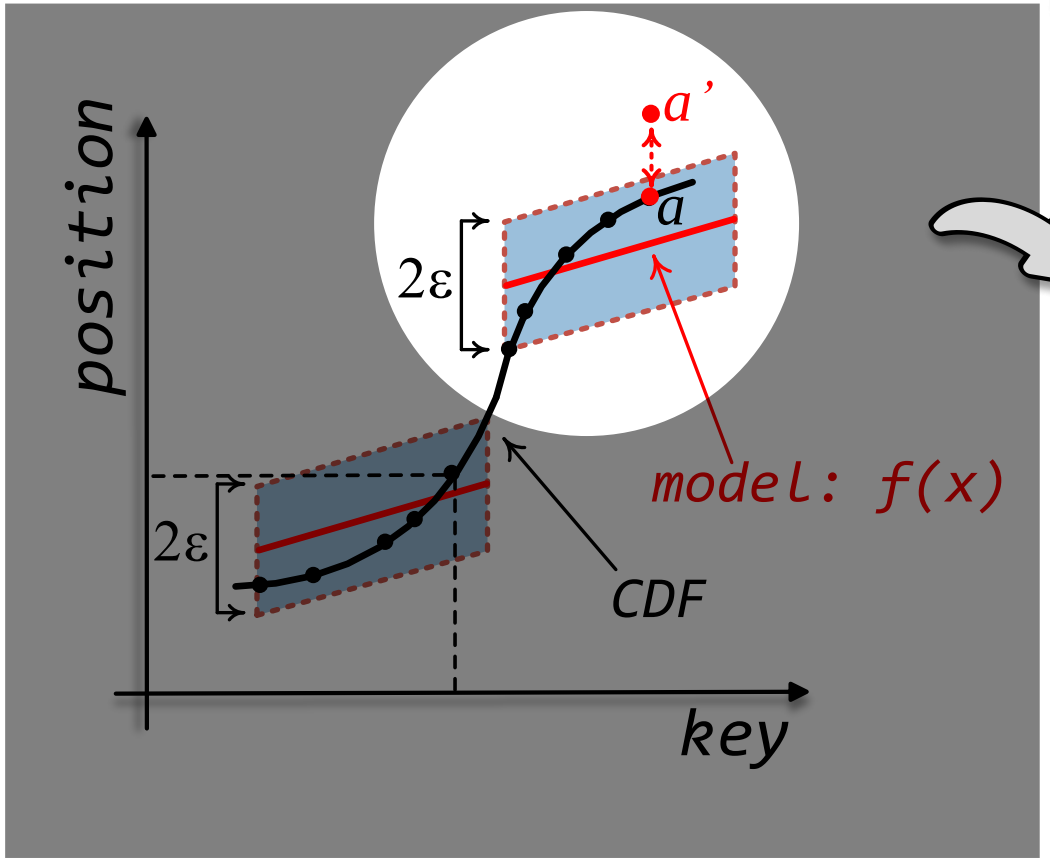


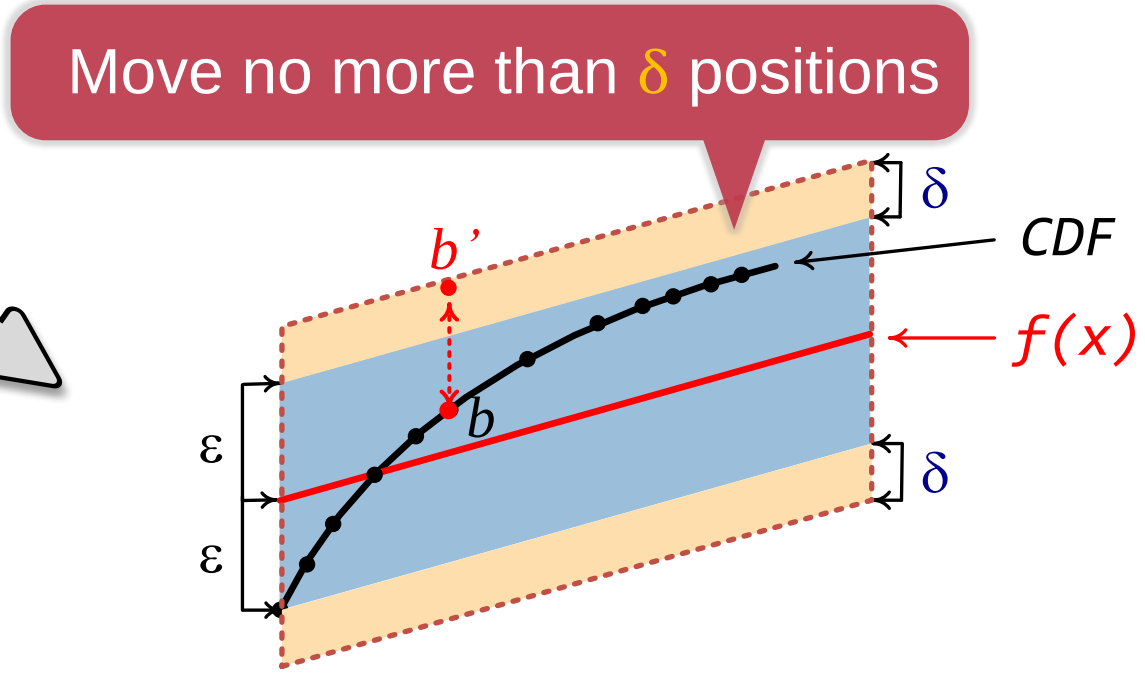
Retraining-Decoupled Learned Indexes

Error! out of prediction range

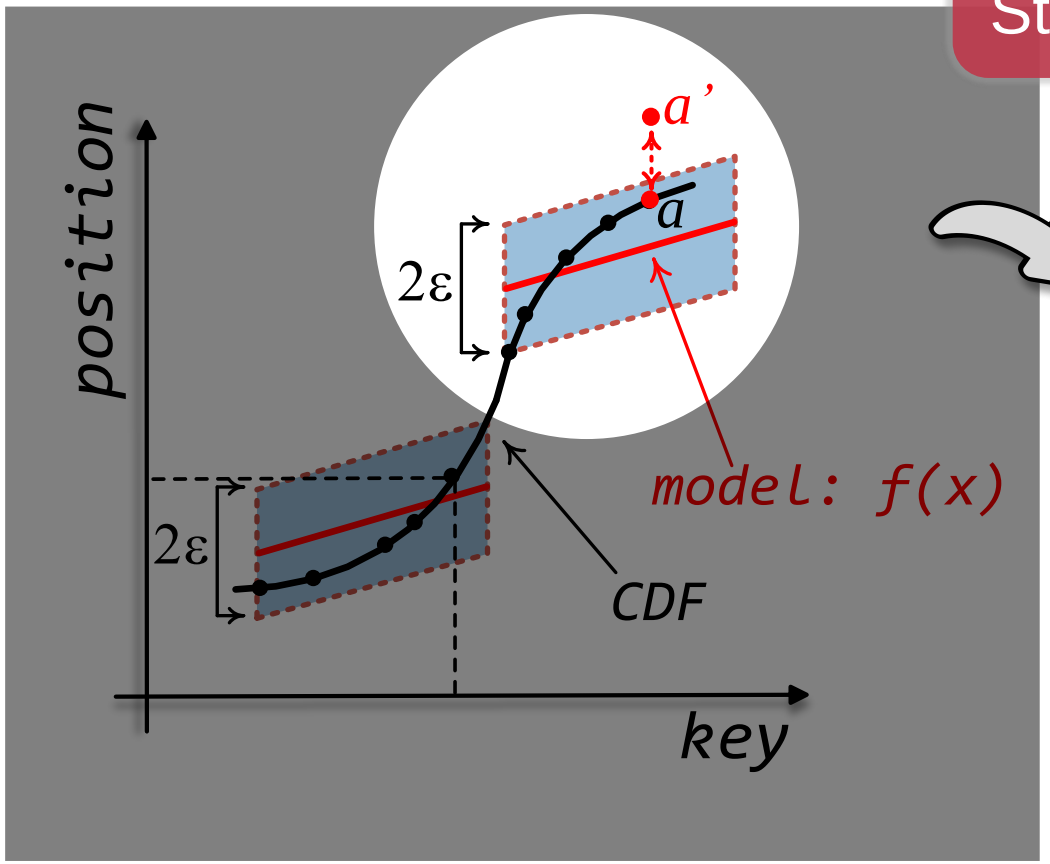


Retraining-Decoupled Learned Indexes

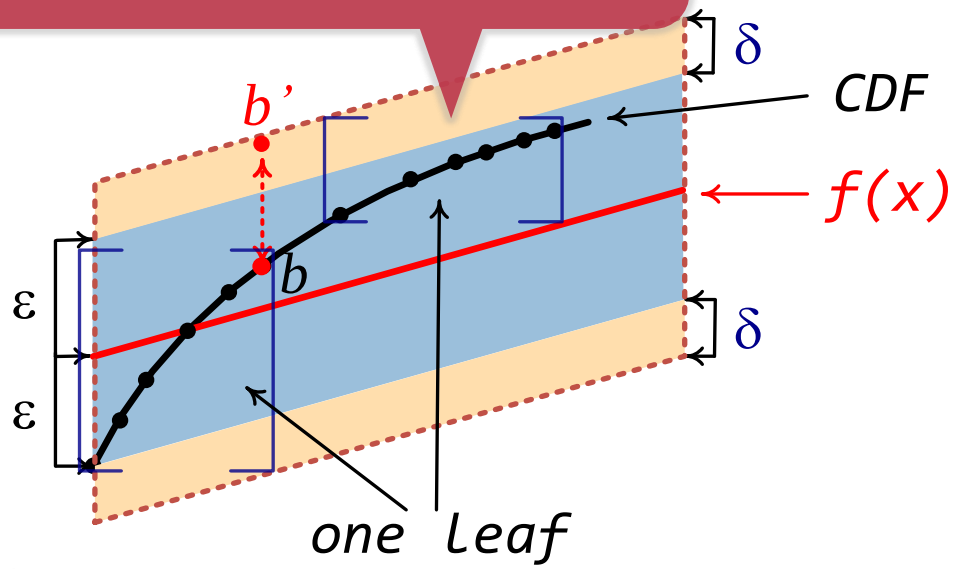




Retraining-Decoupled Learned Indexes



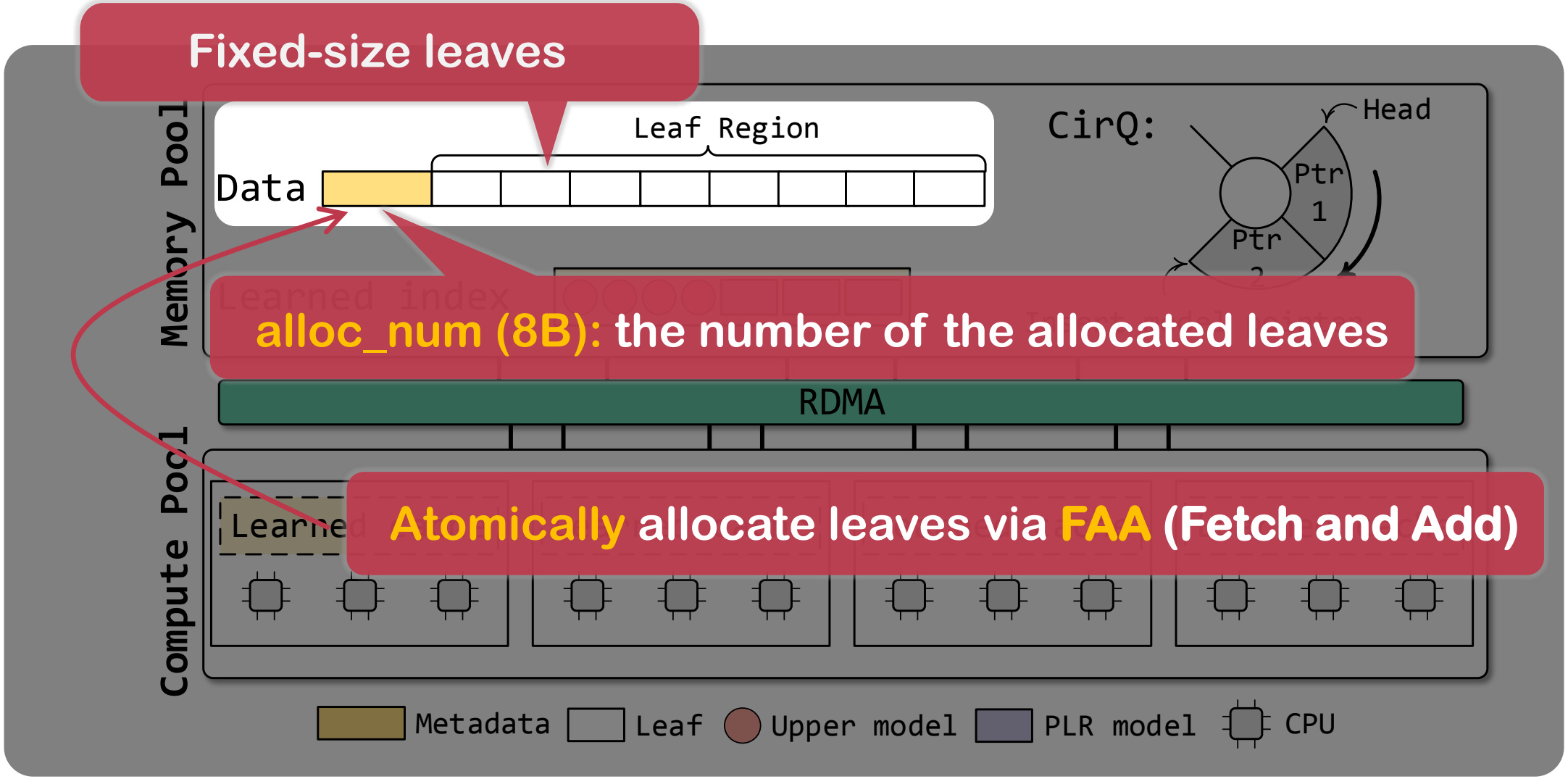
Store data into fixed-size leaves



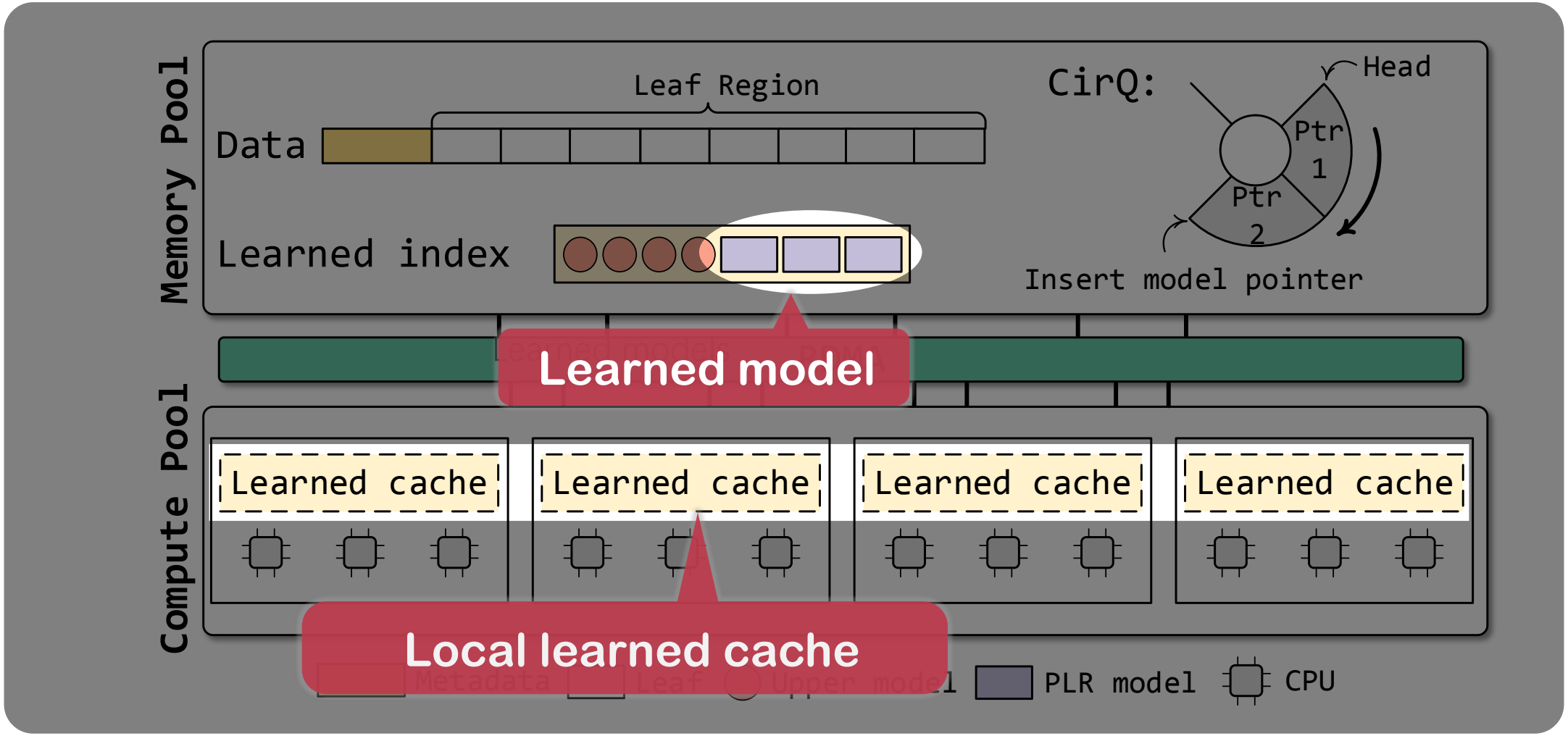
Data-movement constraints

- Moving within fixed-size leaves
- Synonym-leaf sharing

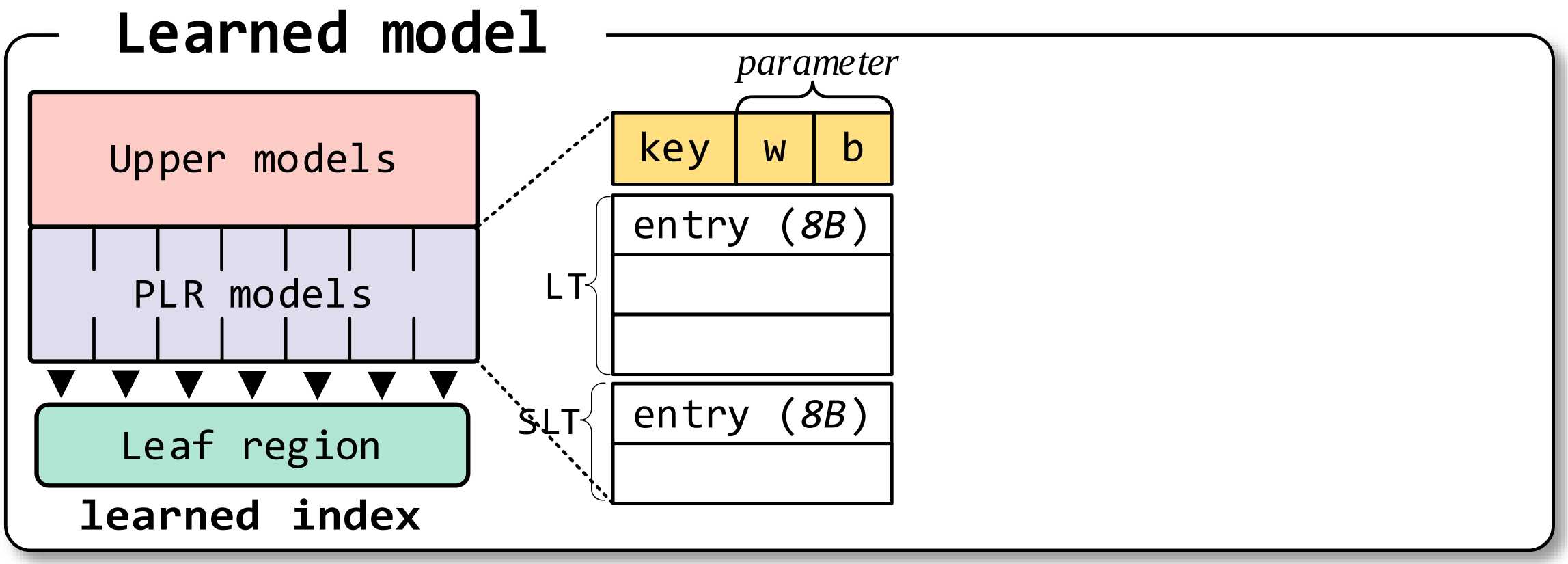
Memory Pool Stores Data



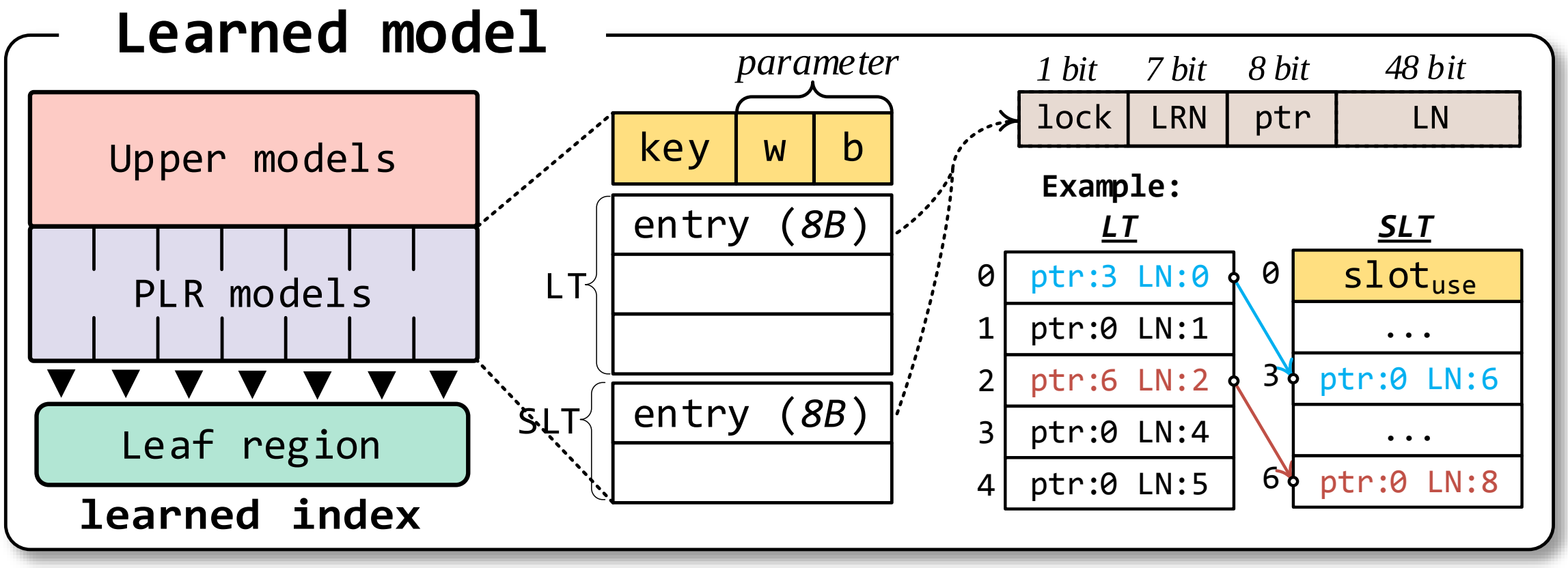
Learned Model



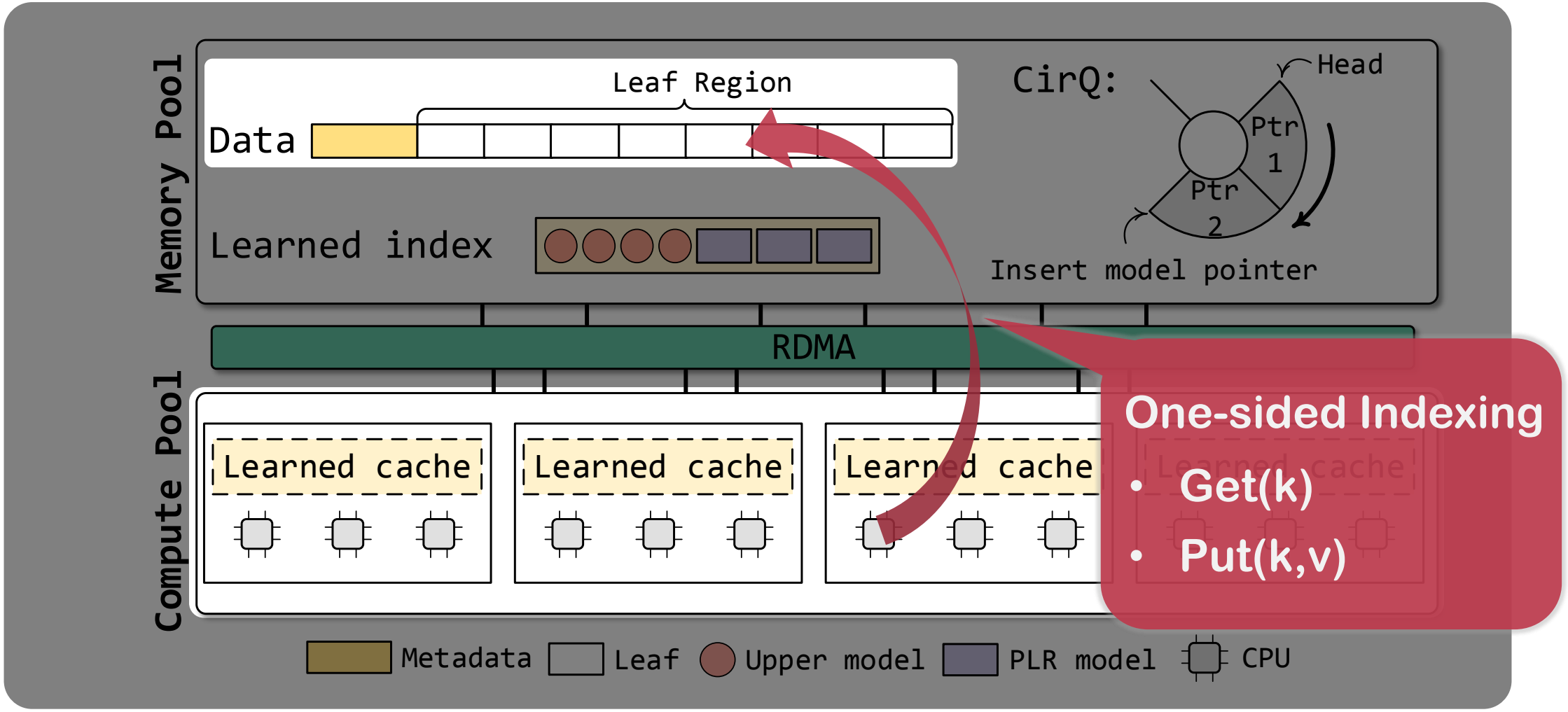
Learned Model – Piecewise Linear Regression Models



Learned Model – Leaf Table



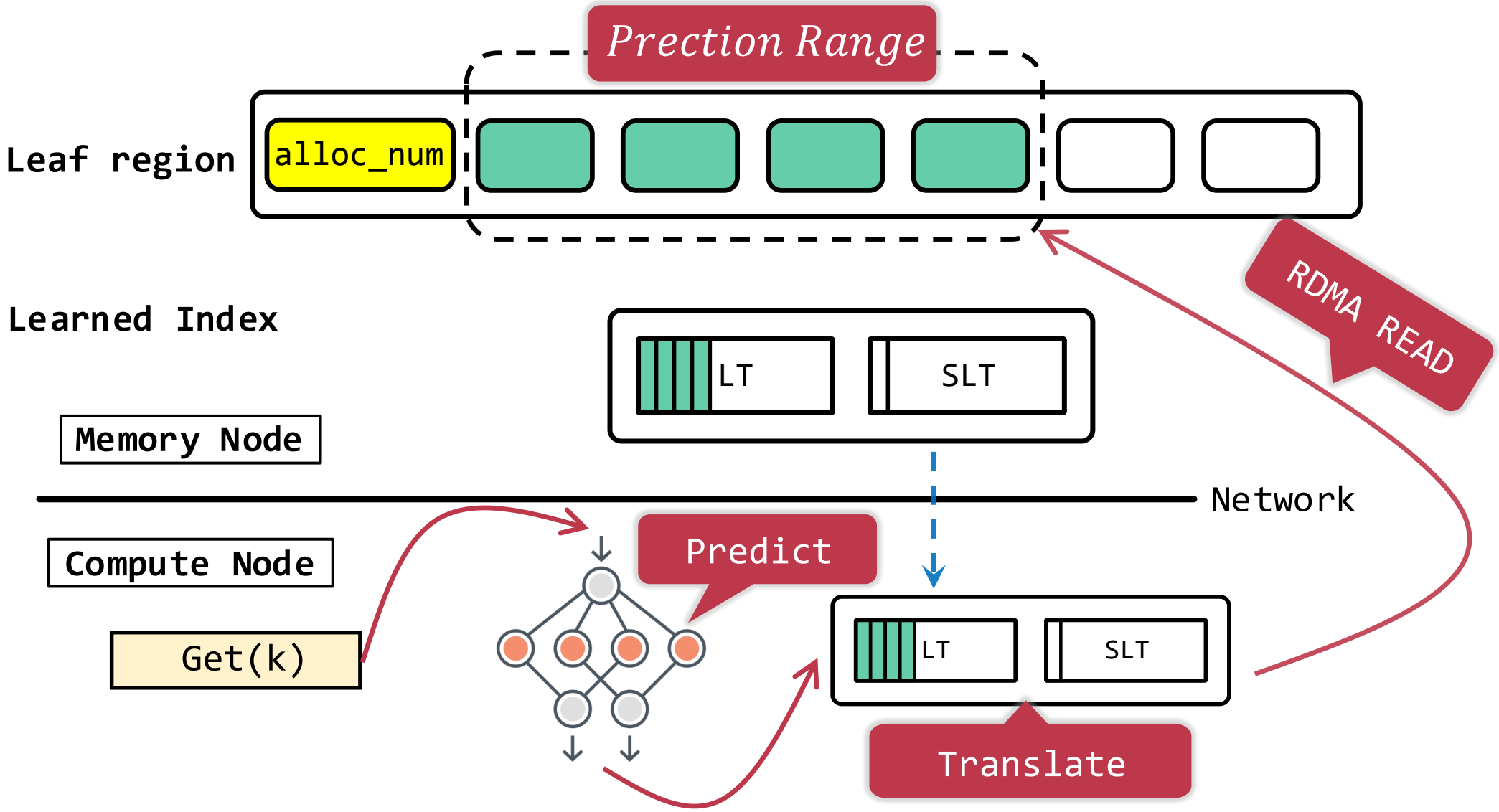
One-sided Indexing



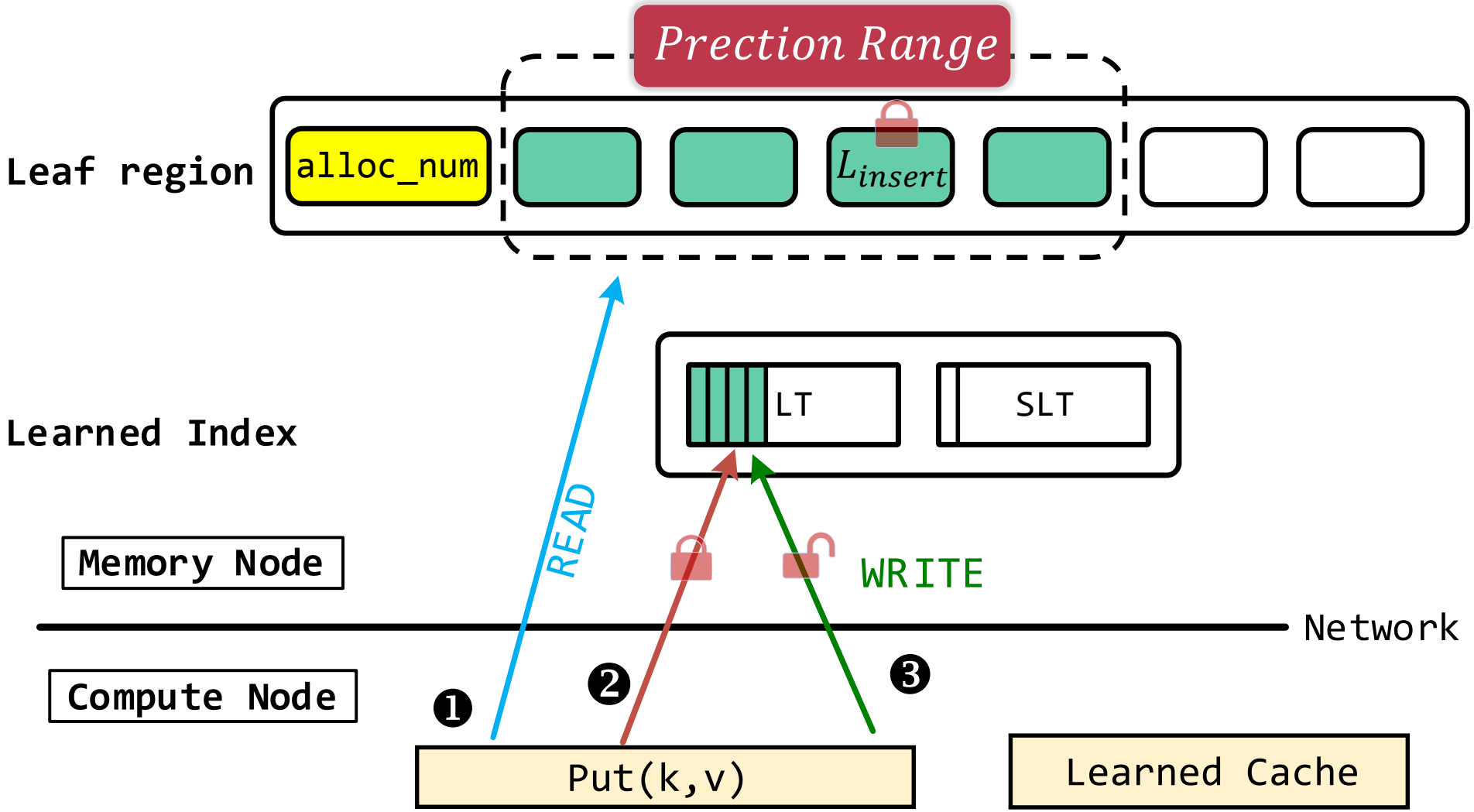
One-sided Indexing

- Get(k)
- Put(k,v)

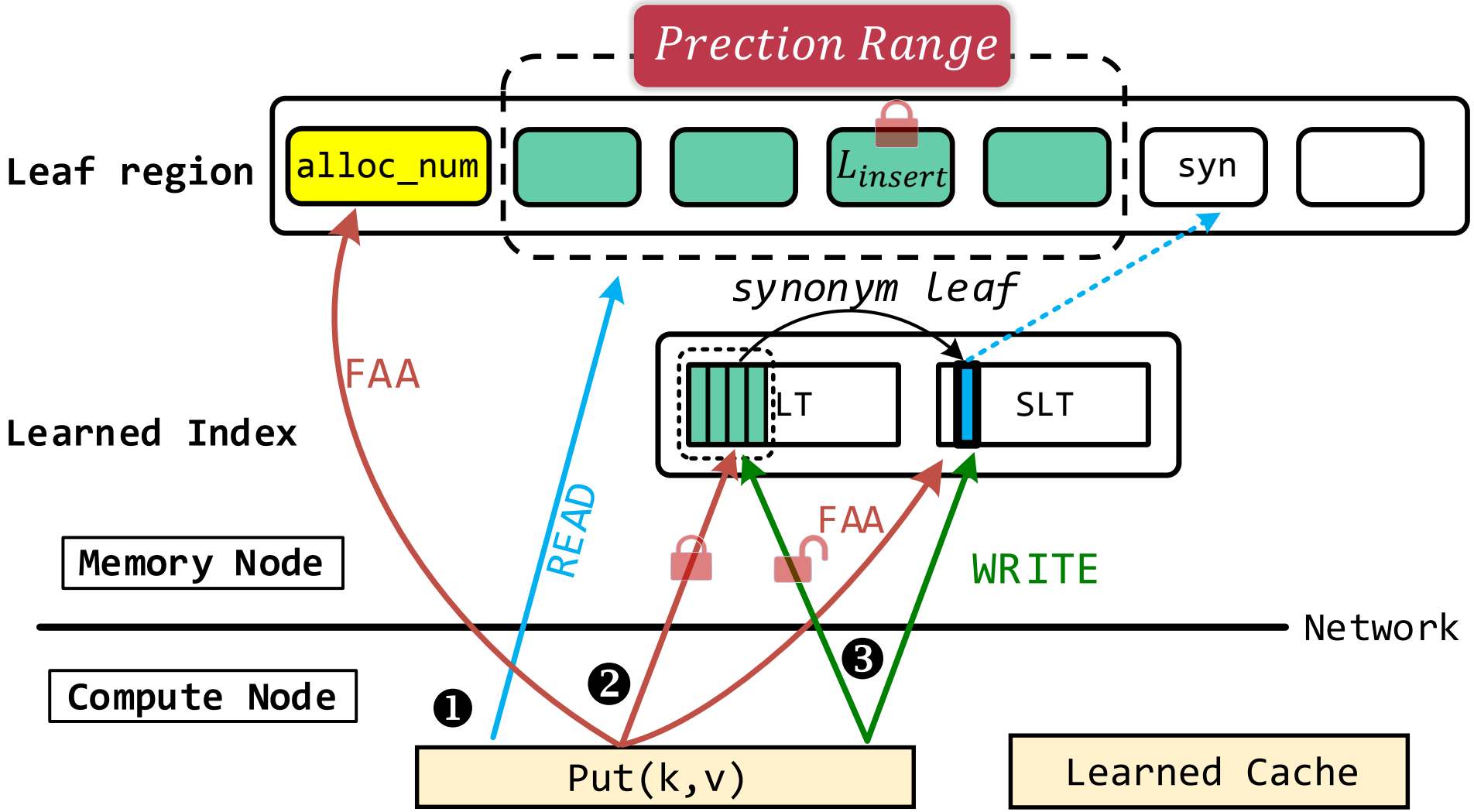
One-sided Indexing – Get (k)



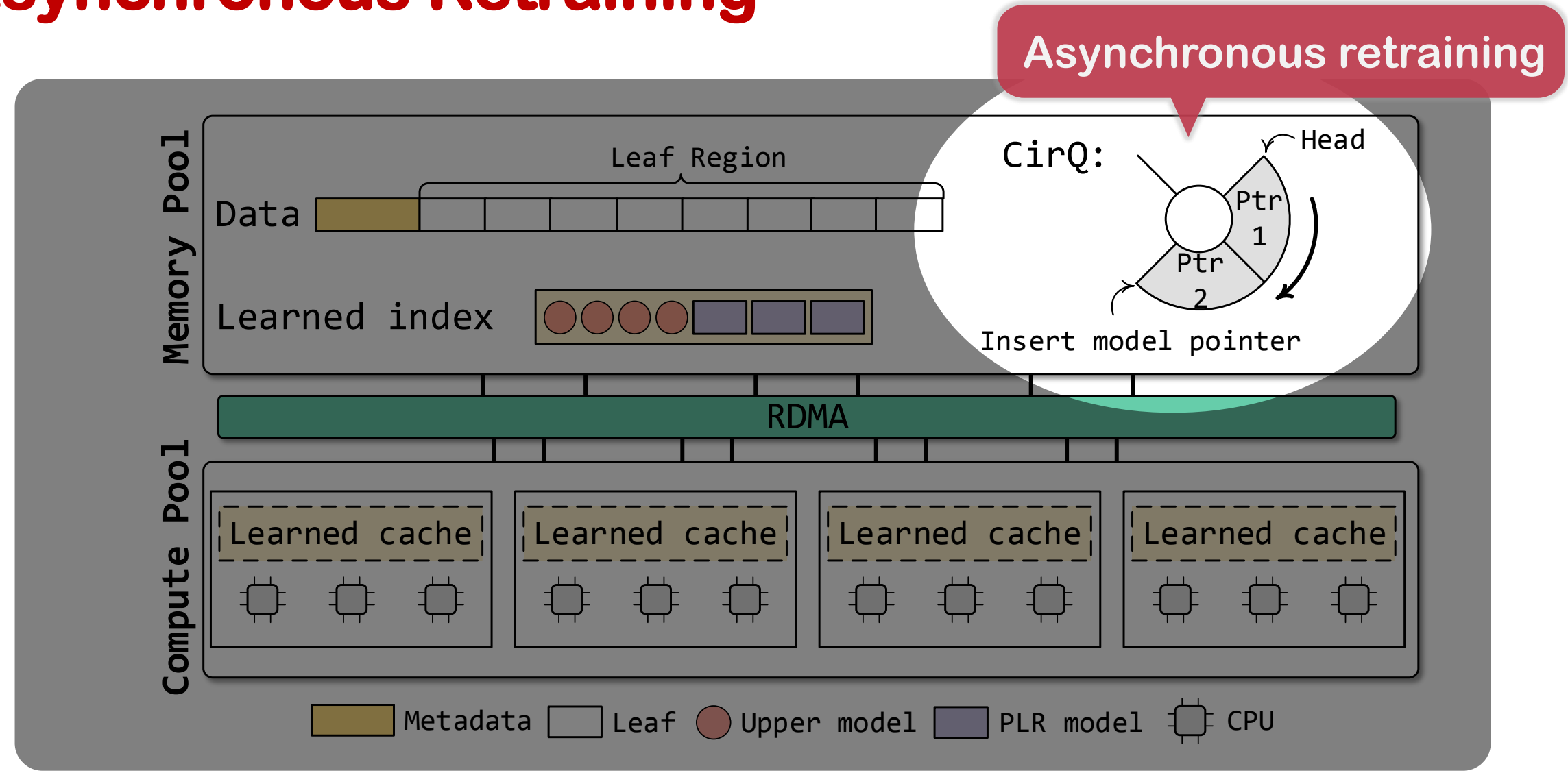
One-sided Indexing – Put (k, v)



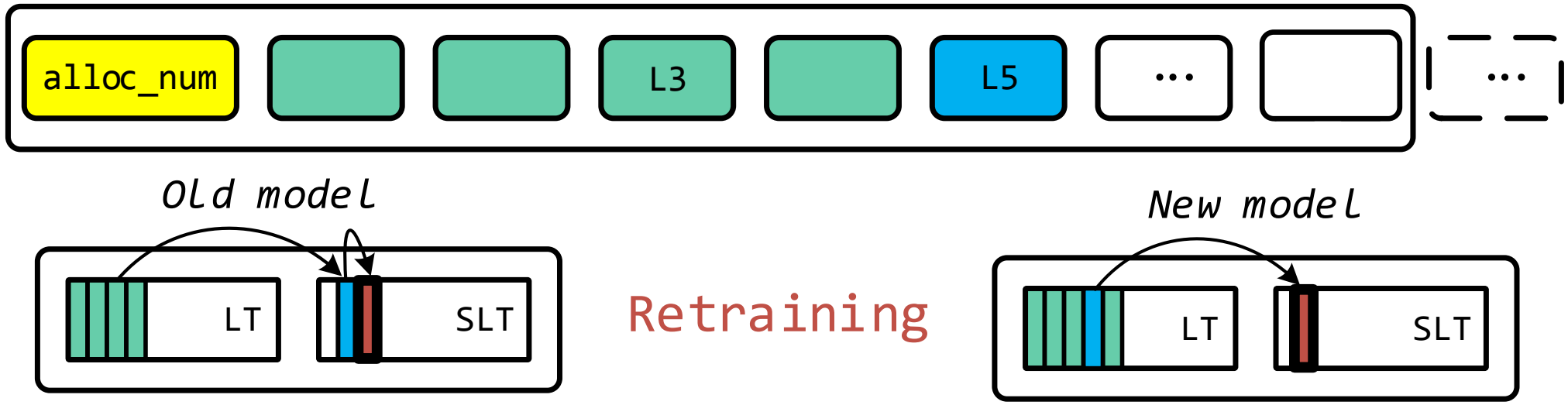
One-sided Indexing – Put (k, v)



Asynchronous Retraining



Asynchronous Retraining – Consistency Guarantee

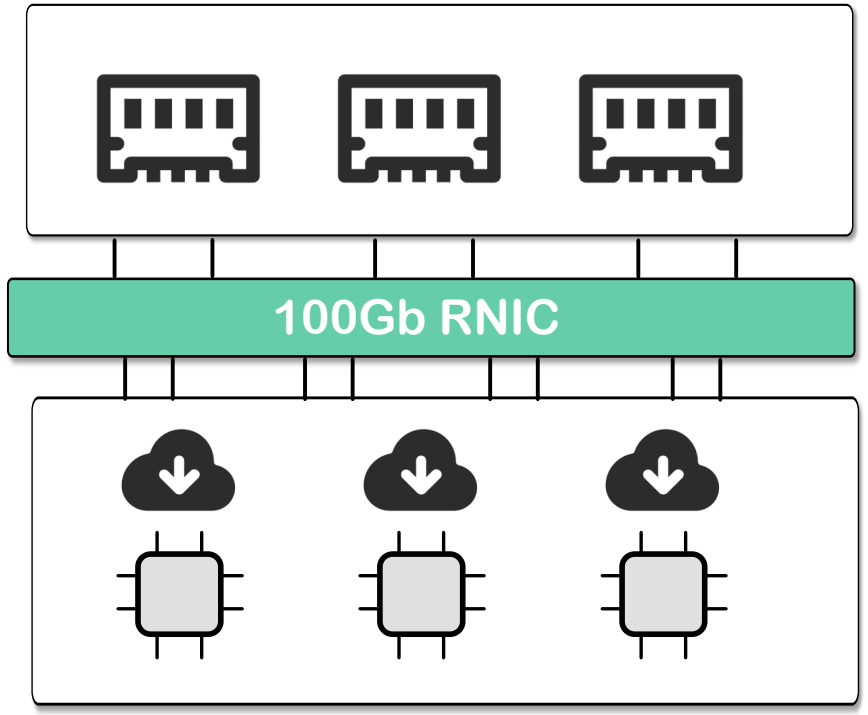


Identify the modified data

- Checking old LT and SLT
- Insert new leaf into new SLT

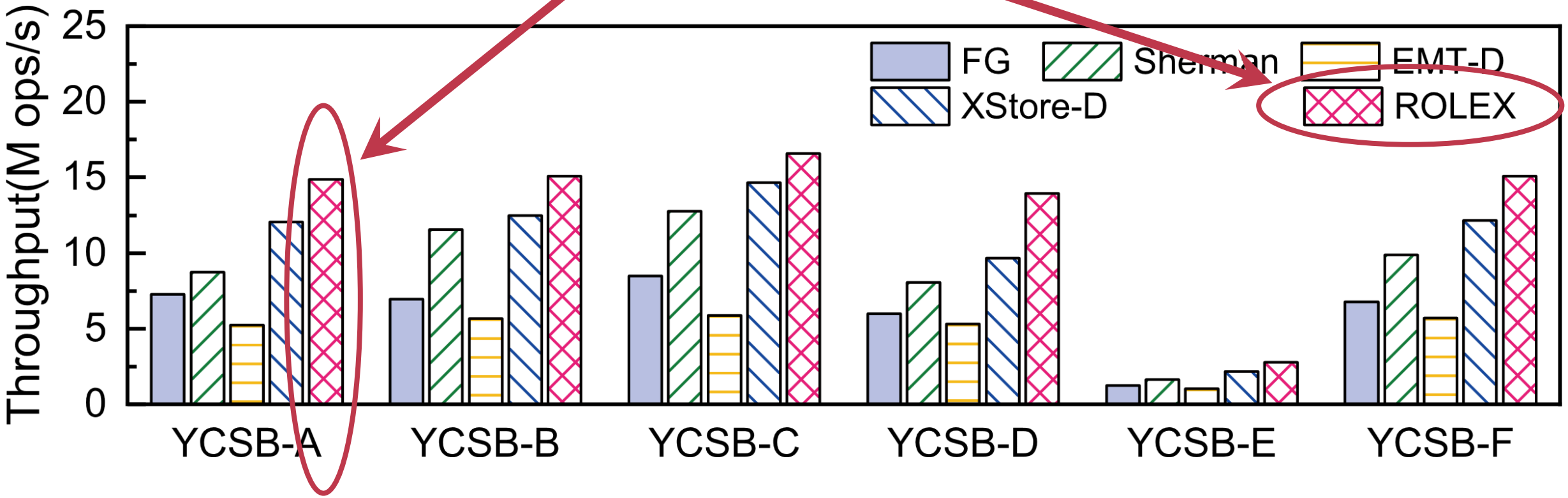
Experimental Setup

- **Testbed**
  - 3 compute nodes + 3 memory nodes
  - 100Gb Mellanox ConnectX-5 IB RNIC
- **Workloads**
  - YCSB; Lognormal & Normal distributions
  - 8B keys and values
- **Comparisons**
  - XStore-D [OSDI'20]
  - Sherman [SIGMOD'22]
  - EMT-D (eRPC + Masstree) [NSDI'19]
  - FG (Fine-grained B-link Tree) [SIGMOD'19]



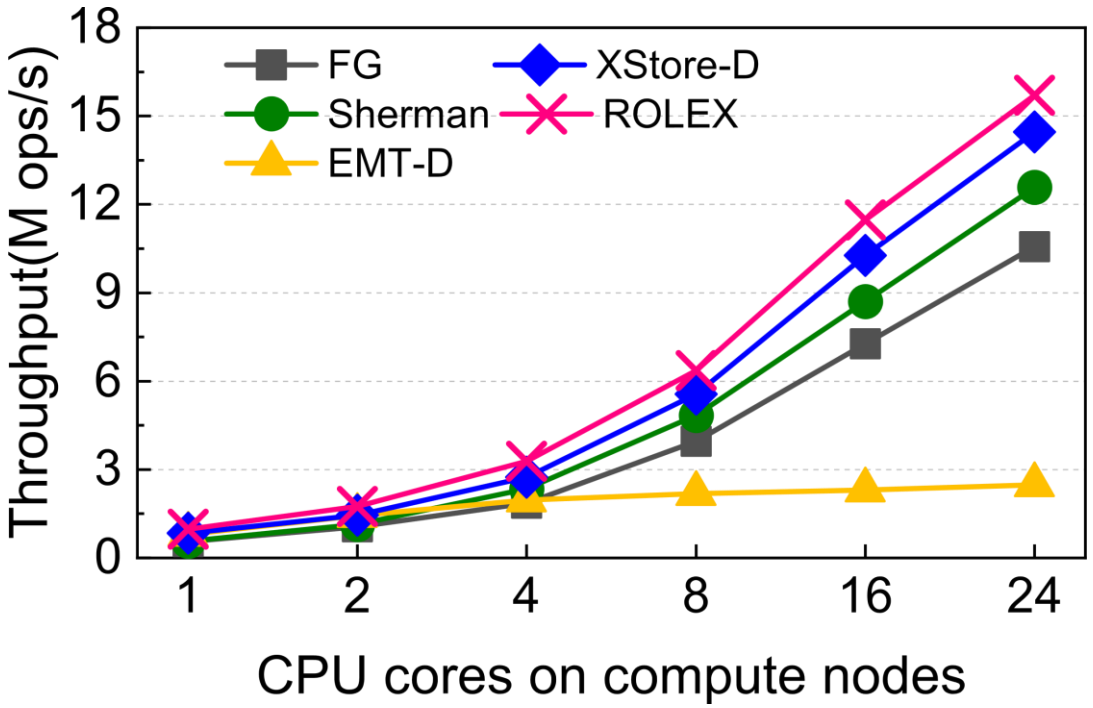
Performance on YCSB

- Competitive performance on static workloads
- 1.3x~2.8x improvements on dynamic workloads

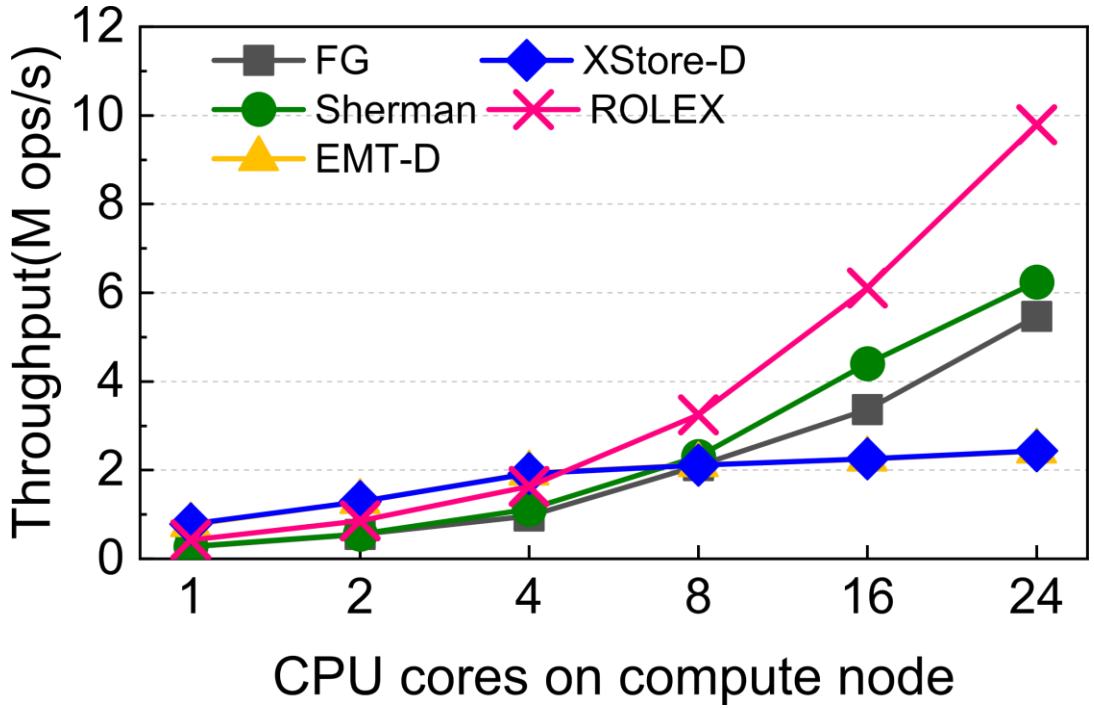


Scale with CPU cores on compute nodes

- ROLEX efficiently scale with computing resources



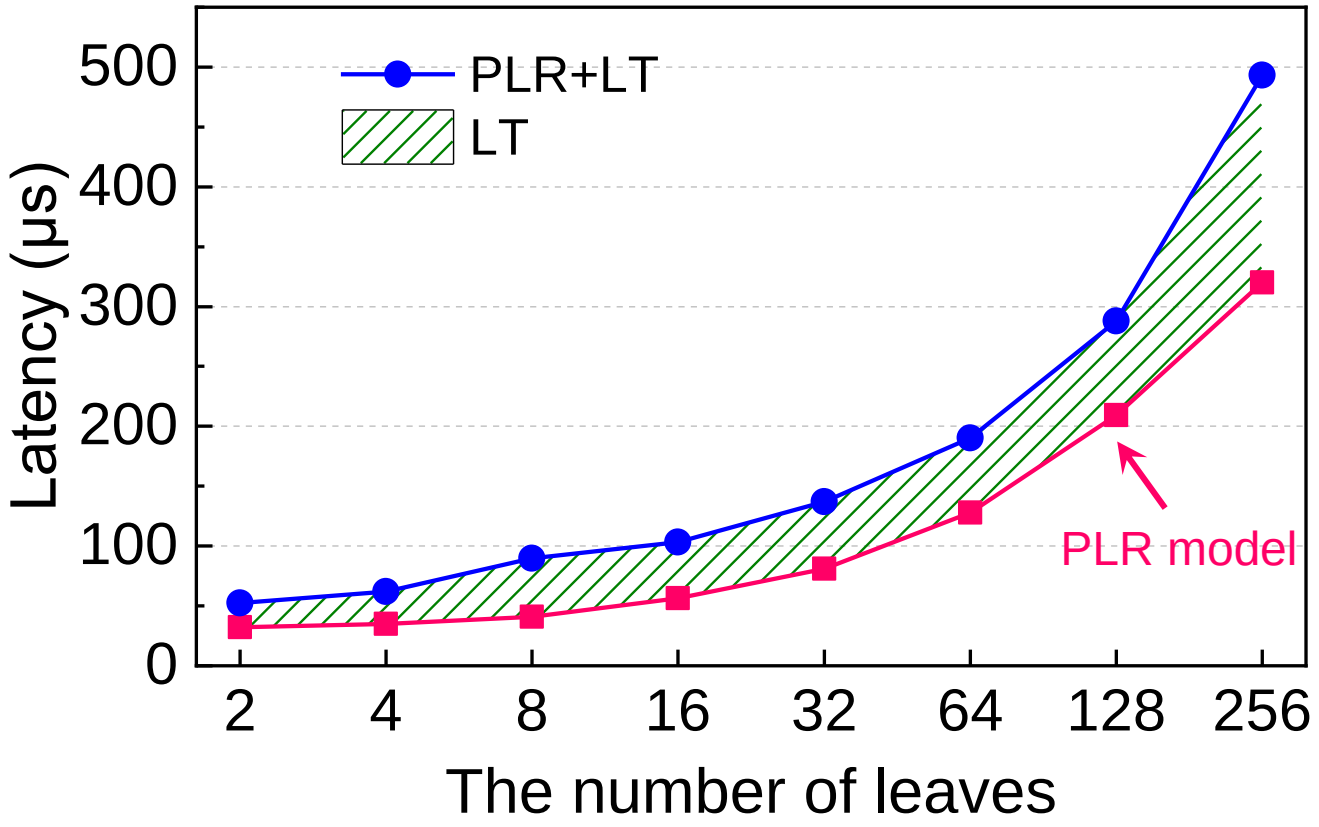
(a) Read performance



(b) Write performance

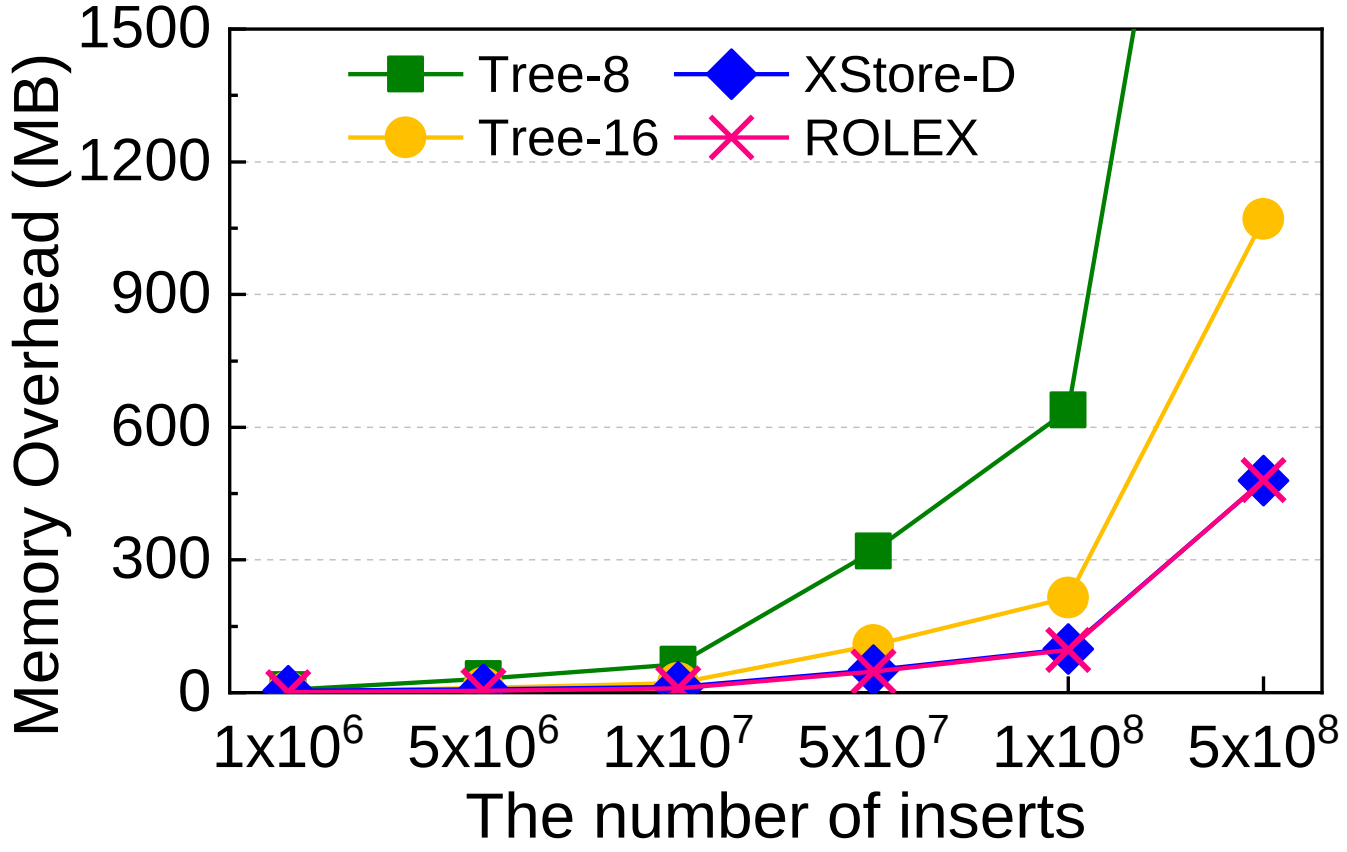
Training Latency

- Training 128 leaves consumes about 300μs
- Retraining a PLR model is efficient



Memory Overhead

- (S)LT accounts for 98% memory overhead
- Learned models save order-of-magnitude space



Conclusion

- **Disaggregated memory systems require efficient one-sided operations**
  - Tree-based structures incur multiple RTTs
  - Learned indexes fail to dynamically change with one-sided operations
- **ROLEX: a scalable RDMA-oriented KV store using learned indexes**
  - Operation decoupling
  - One-sided indexing
  - Asynchronous retraining
- **1.3x~2.8x improvements on the dynamic workloads**
  - Check ROLEX @ https://github.com/iotlpf/ROLEX

Thanks!

Q & A

cspfli@hust.edu.cn