Towards Cost-Effective and Elastic Cloud Database Deployment via Memory Disaggregation

Yingqiang Zhang[1], **Chaoyi Ruan[1,2]**, Cheng Li[2], Xinjun Yang[1], Wei Cao[1], Feifei Li[1], Bo Wang[1], Jing Fang[1], Yuhui Wang[1], Jingze Huo[1,2], Chao Bi[1,2]

Alibaba Cloud
Worldwide Cloud Services Partner

USTC.CHINA
ADSLAB
先进数据系统实验室

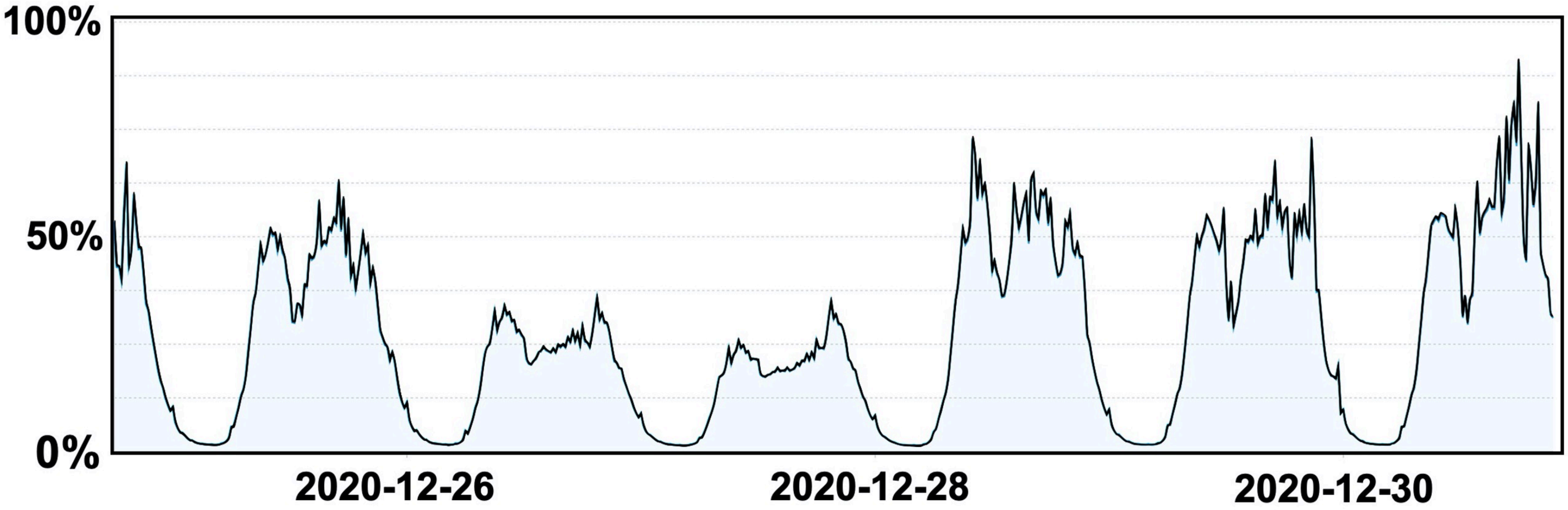
The Rise of Cloud-Native Databases

- Cloud-native databases
  - leverage modern cloud infrastructures
  - target high performance, high elasticity, and low price
  - serve as building block for cloud applications

Amazon Aurora POLARDB Cloud Spanner Azure GaussDB

Elastic Resource Demands by Cloud-native databases

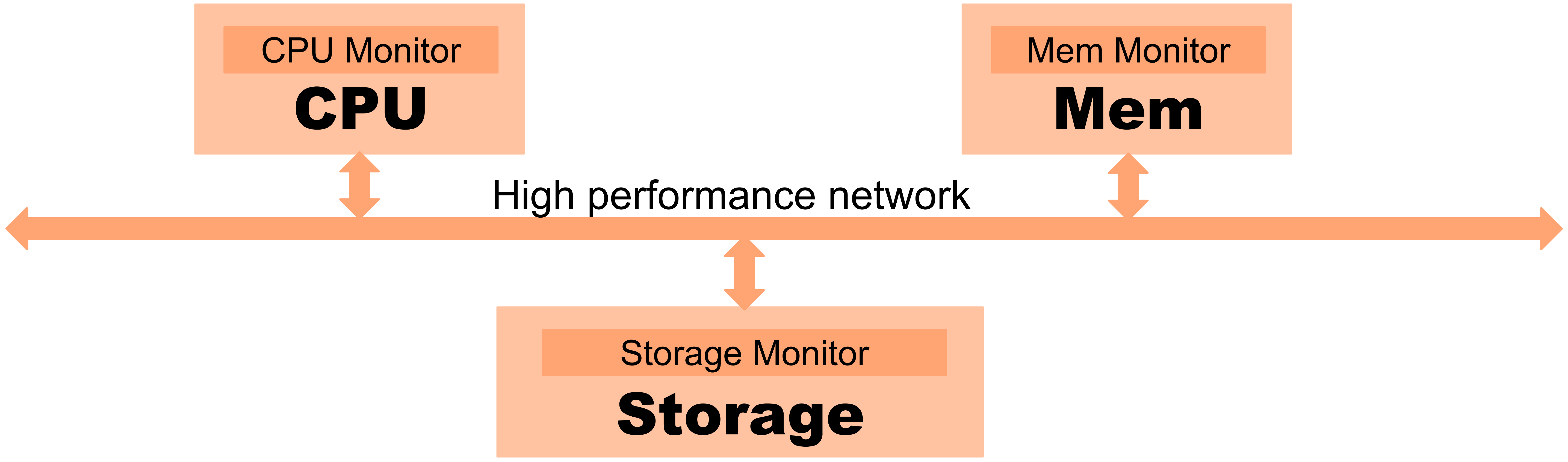
- Most of time, the utilization of CPU is below 50%
- However, for short time periods, it reaches up to 91.27% at peak



The traditional monolithic setup fails to meet the demands

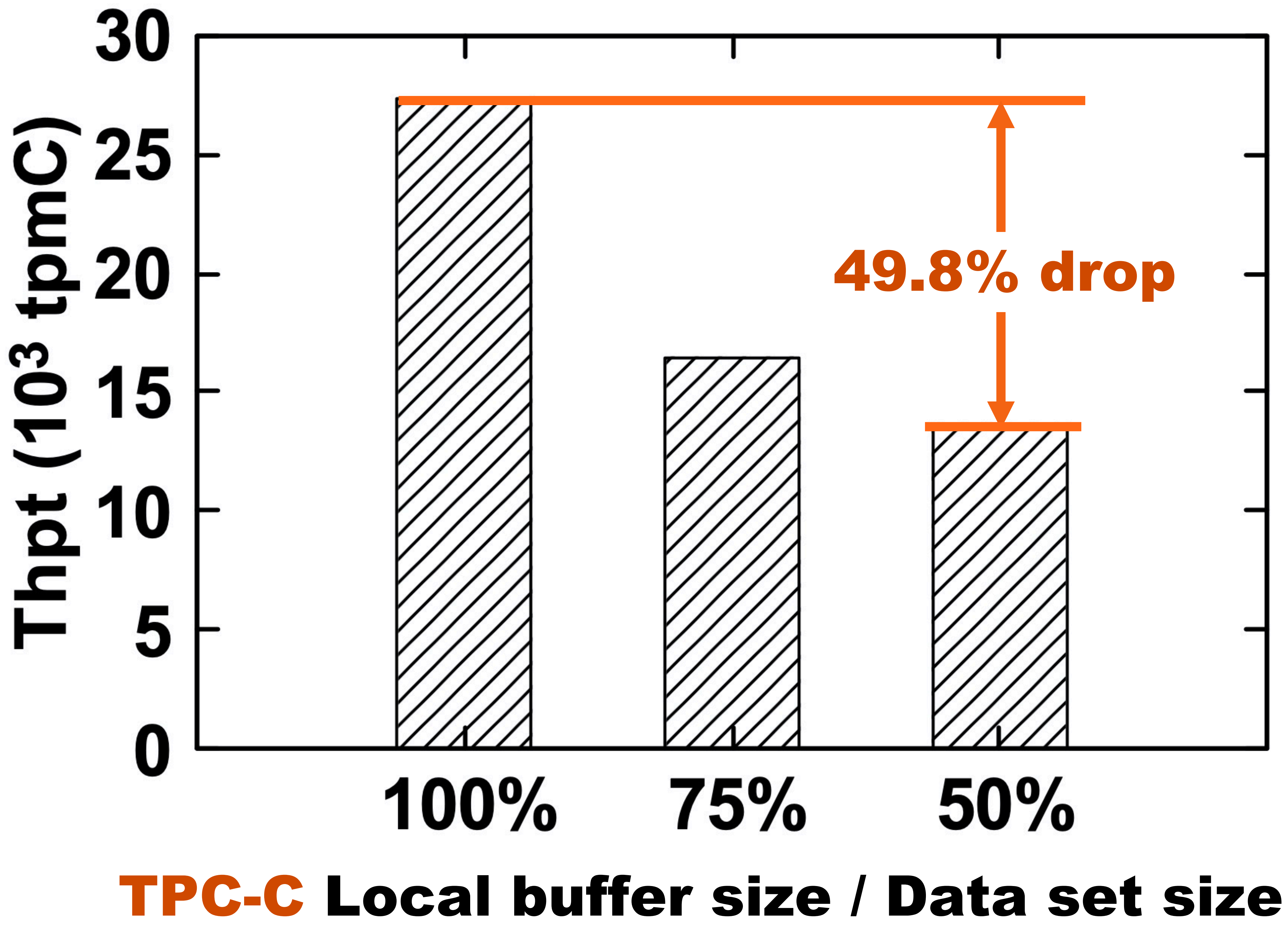
Related Work: Resource Disaggregation

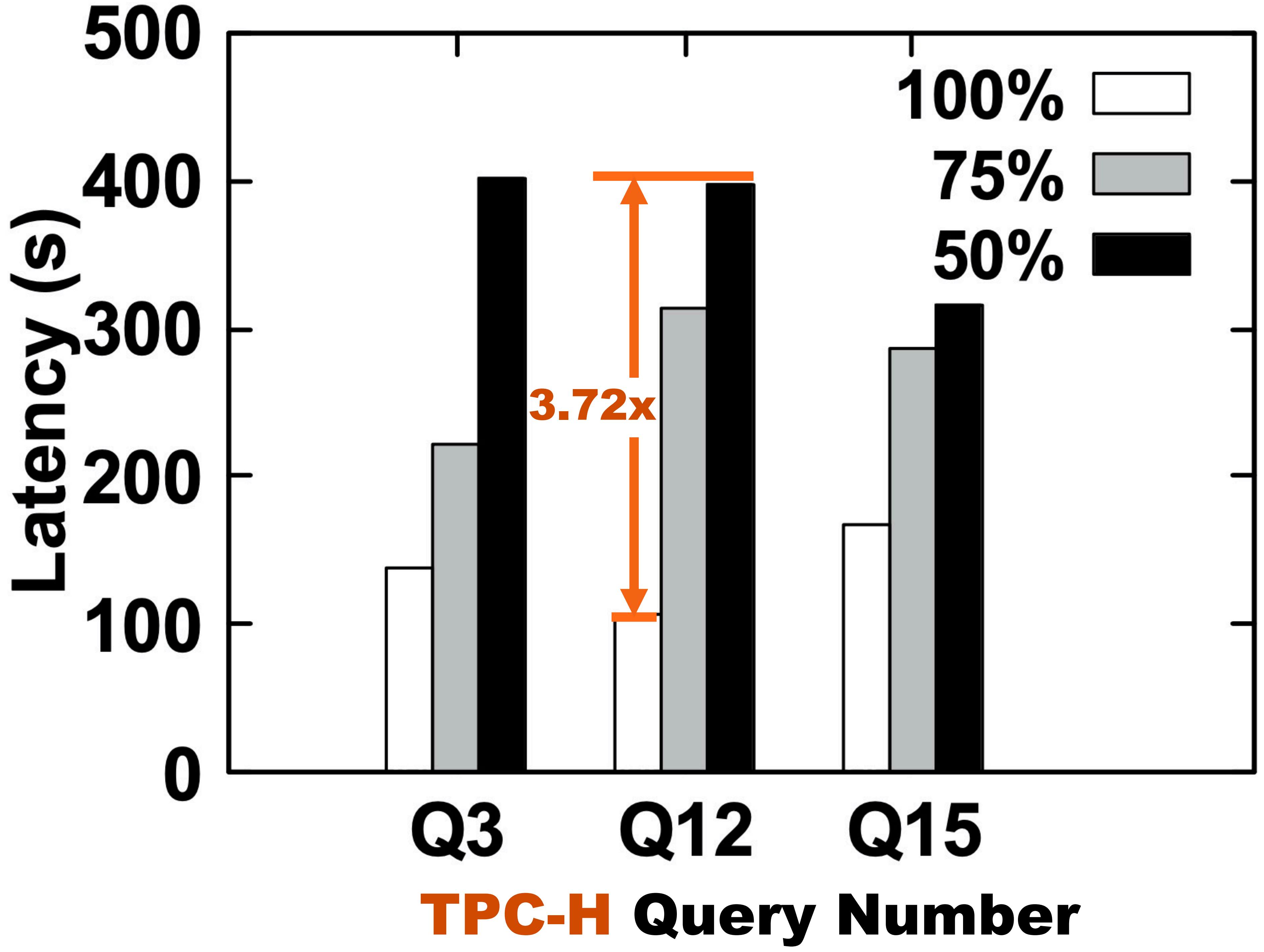
- Disaggregated databases
  - Amazon Aurora [SIGMOD' 17], PolarDB [SIGMOD' 21]
  - They only focus on compute and storage disaggregation.
- General disaggregation approaches
  - LegoOS [OSDI' 18], Infiniswap [NSDI' 17]
  - High memory access and failure recovery overhead, due to being oblivious to database I/O characteristics and going through traditional I/O stack



Challenge 1: High disaggregation overhead

- Run workloads with Infiniswap
- Remote memory pool is accessed via 25Gbps network

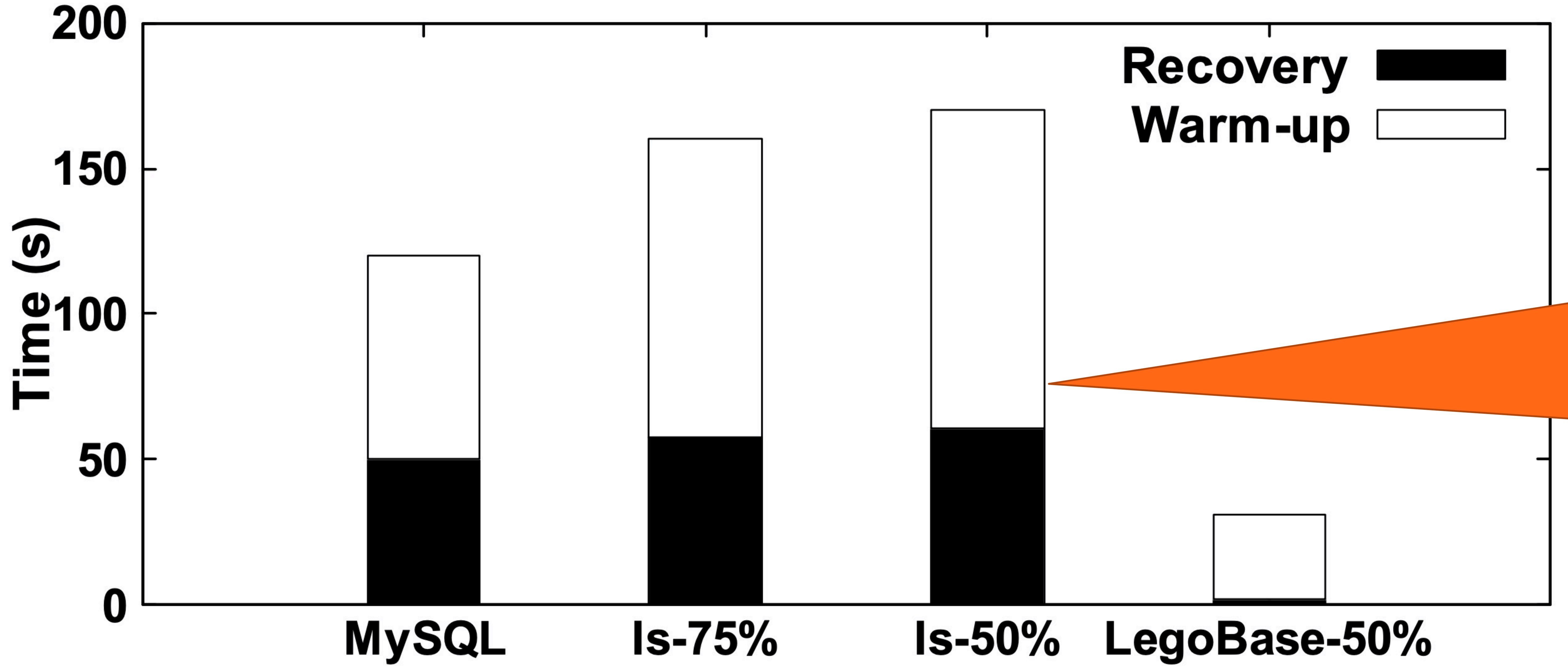




Takeaway 1: Fast (remote) memory access advocates the co-design of database kernel and memory disaggregation.

Challenge 2: High fault recovery cost

- Run MySQL atop Infiniswap with TPC-C and varied remote memory ratios
- Crash the MySQL instance and measure the fail-over time costs

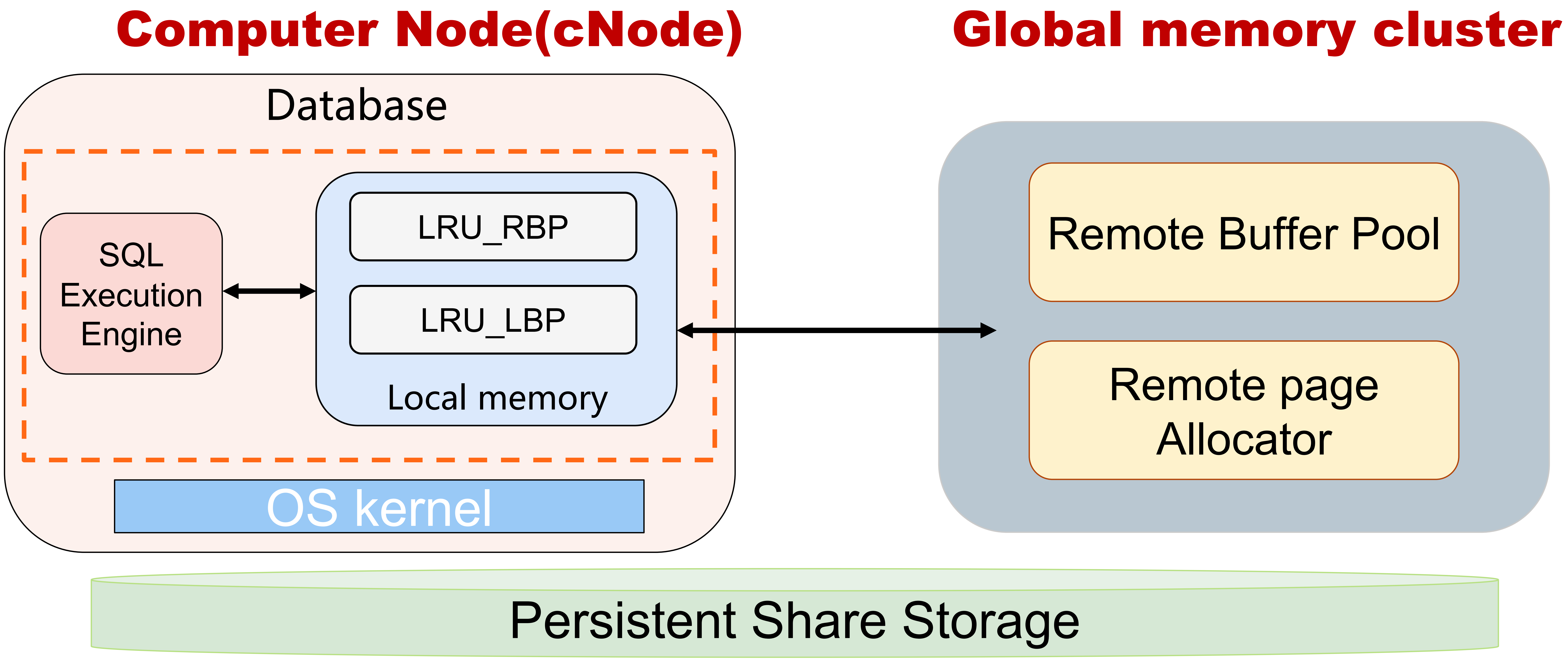


The pages cached remotely are not used, thus precluding fast recovery for local crashes.

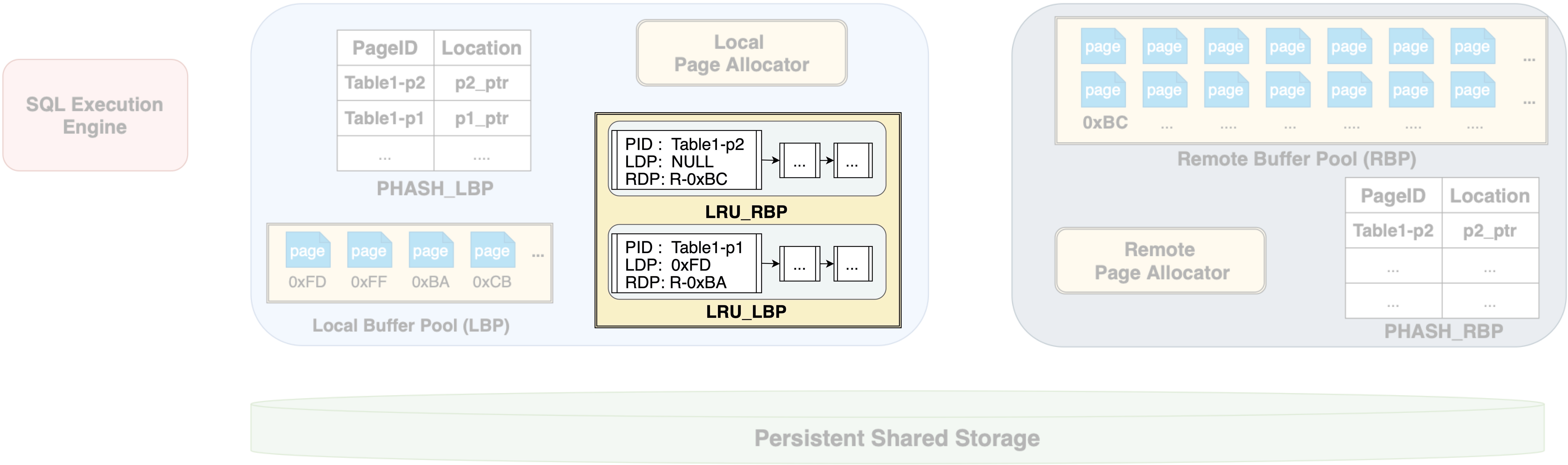
Takeaway 2: Independent fault handling requires to bridge the gap between the monolithic fault tolerance protocol and memory disaggregation.

Database co-design

- Bypass OS kernel
- Retain MySQL sophisticated LRU
- Leverage DB access patterns and data layout

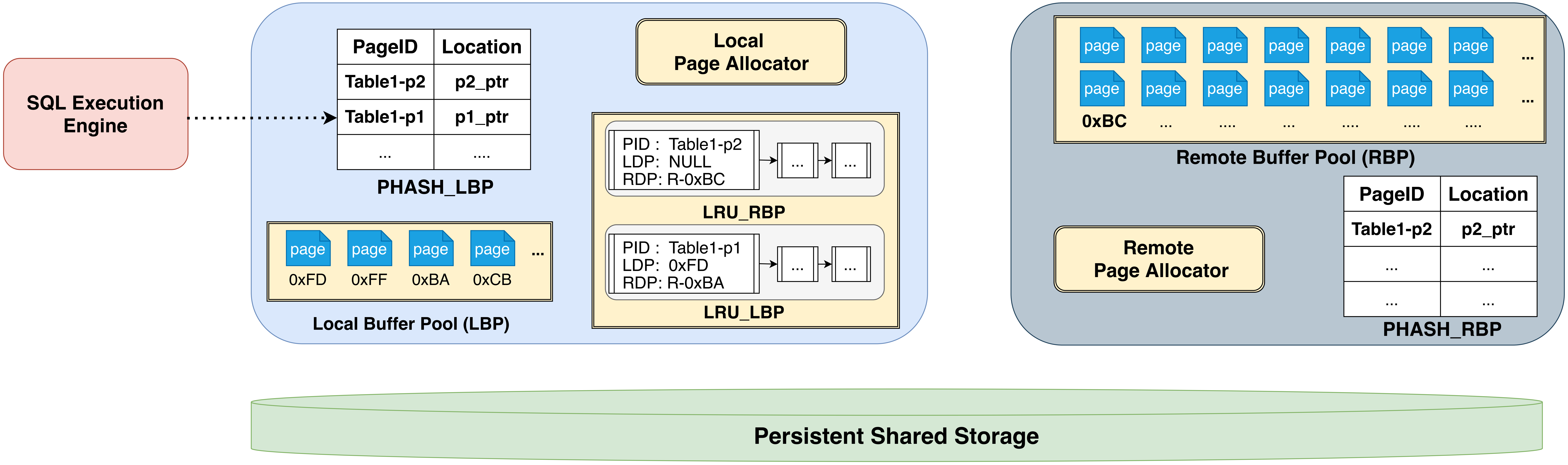


Local and Remote Memory Management



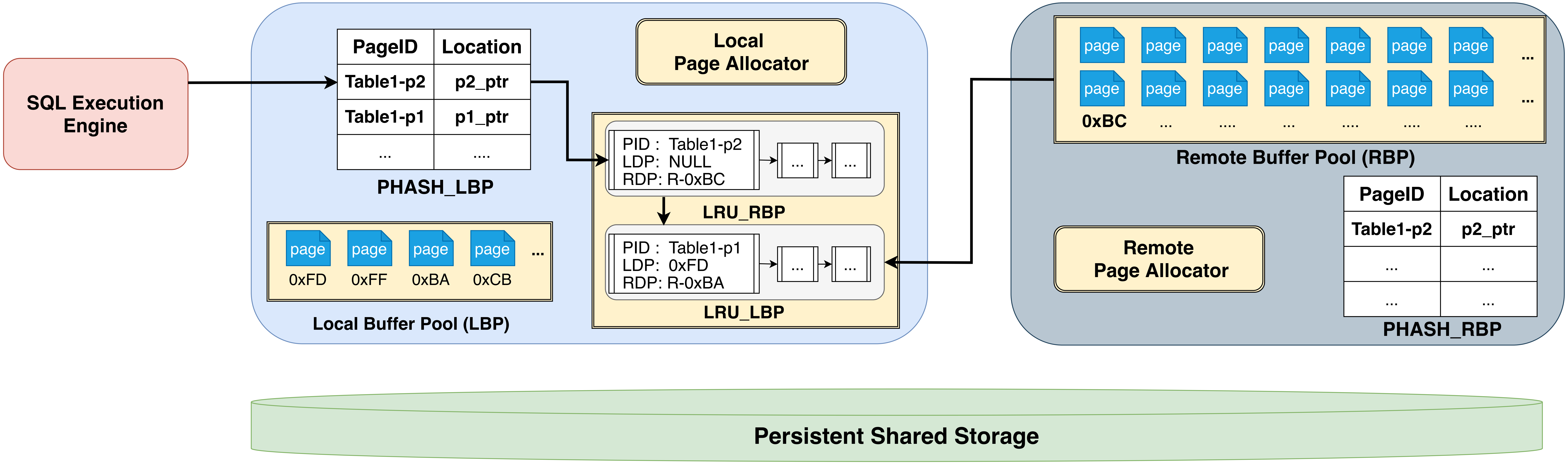
Two-level LRU : hold all metadata of local and remote pages，reduce communications with remote memory node

Local and Remote Memory Management



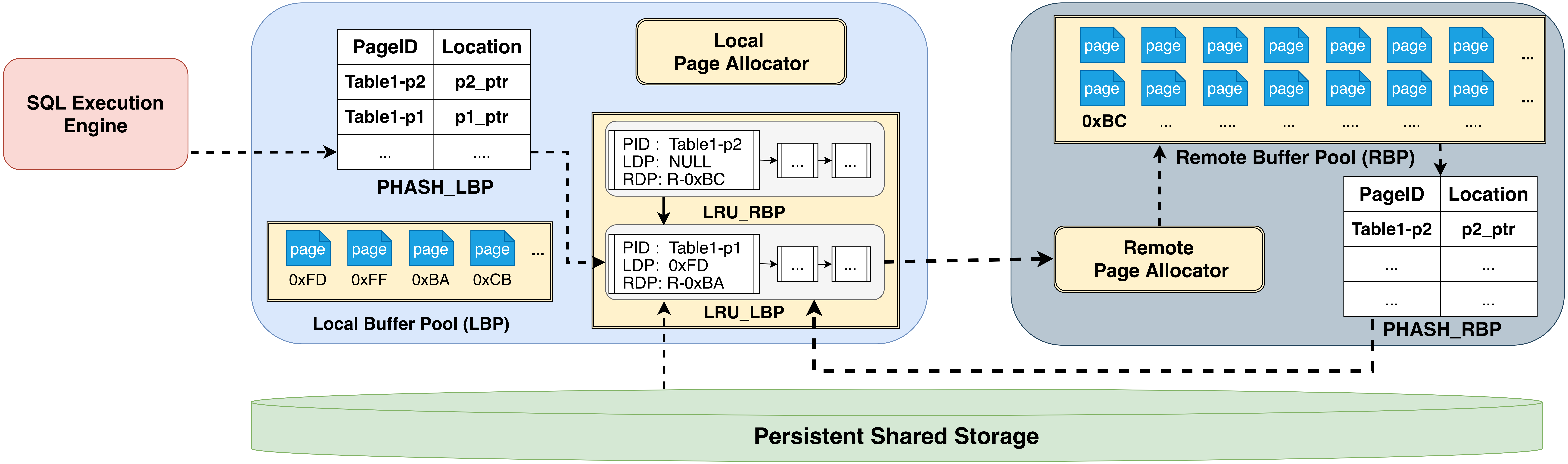
Page access from local buffer pool

Local and Remote Memory Management



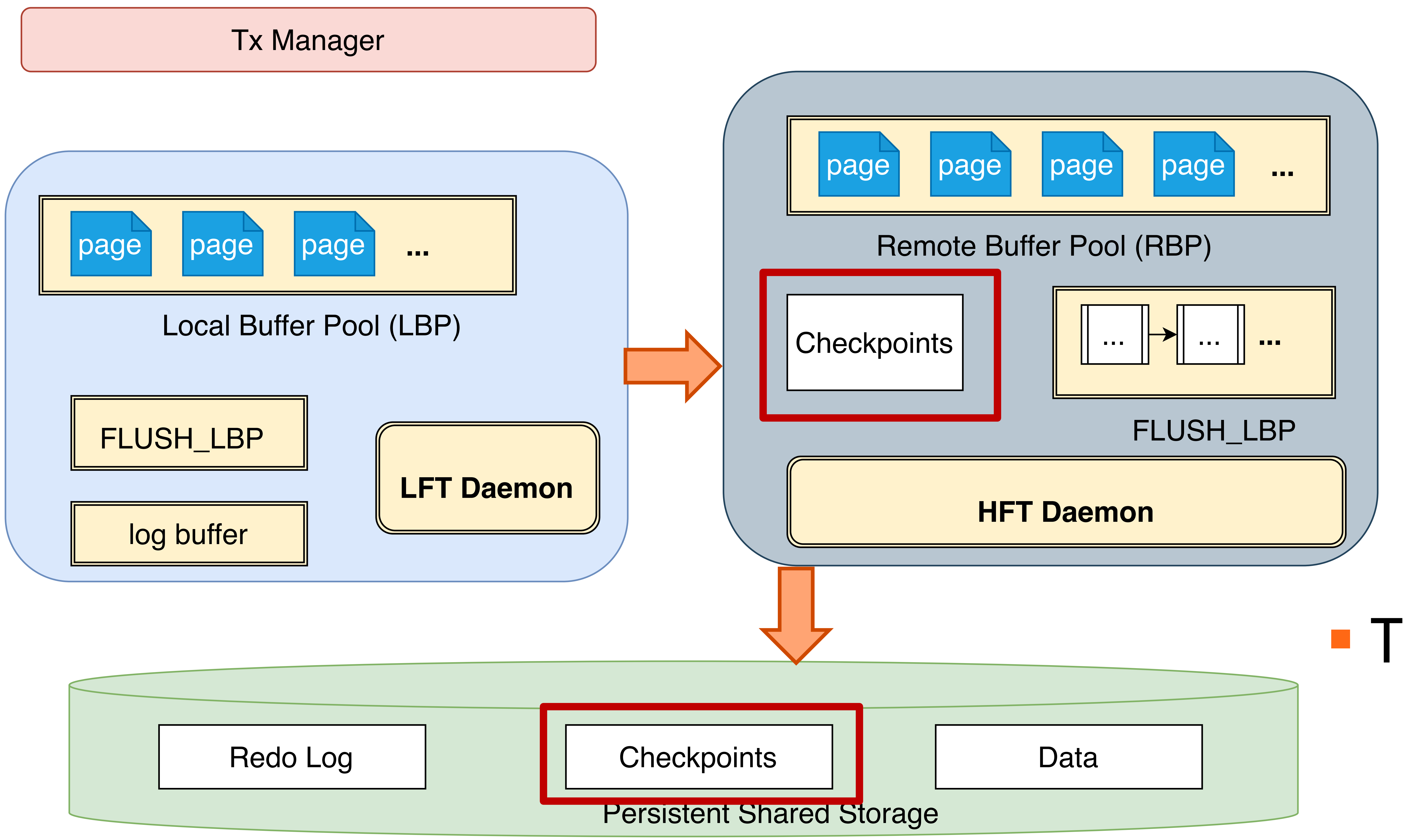
Page access from remote buffer pool

Local and Remote Memory Management



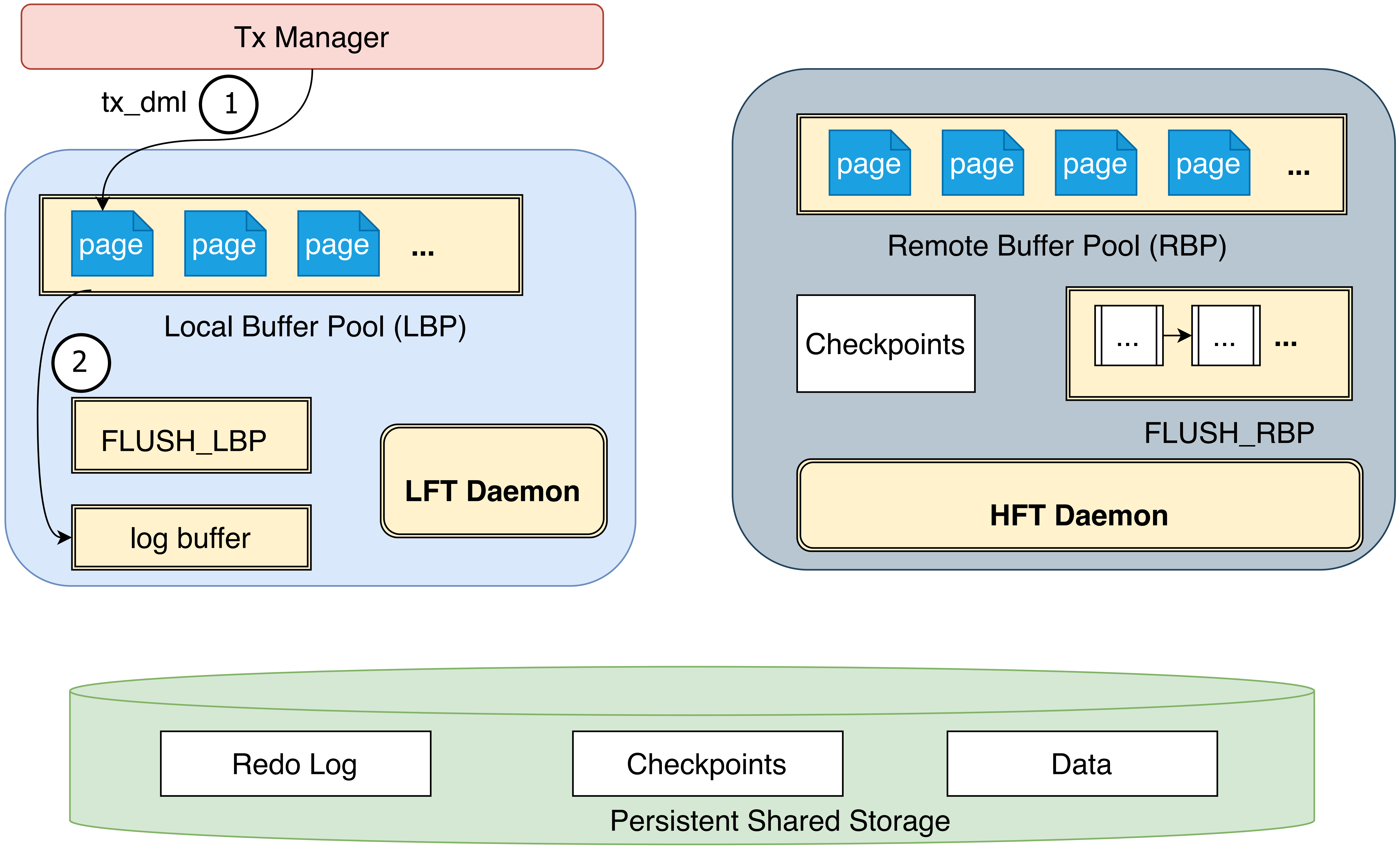
Page access from persistent shared storage

New Fault Tolerance: Two-tier ARIES Protocol

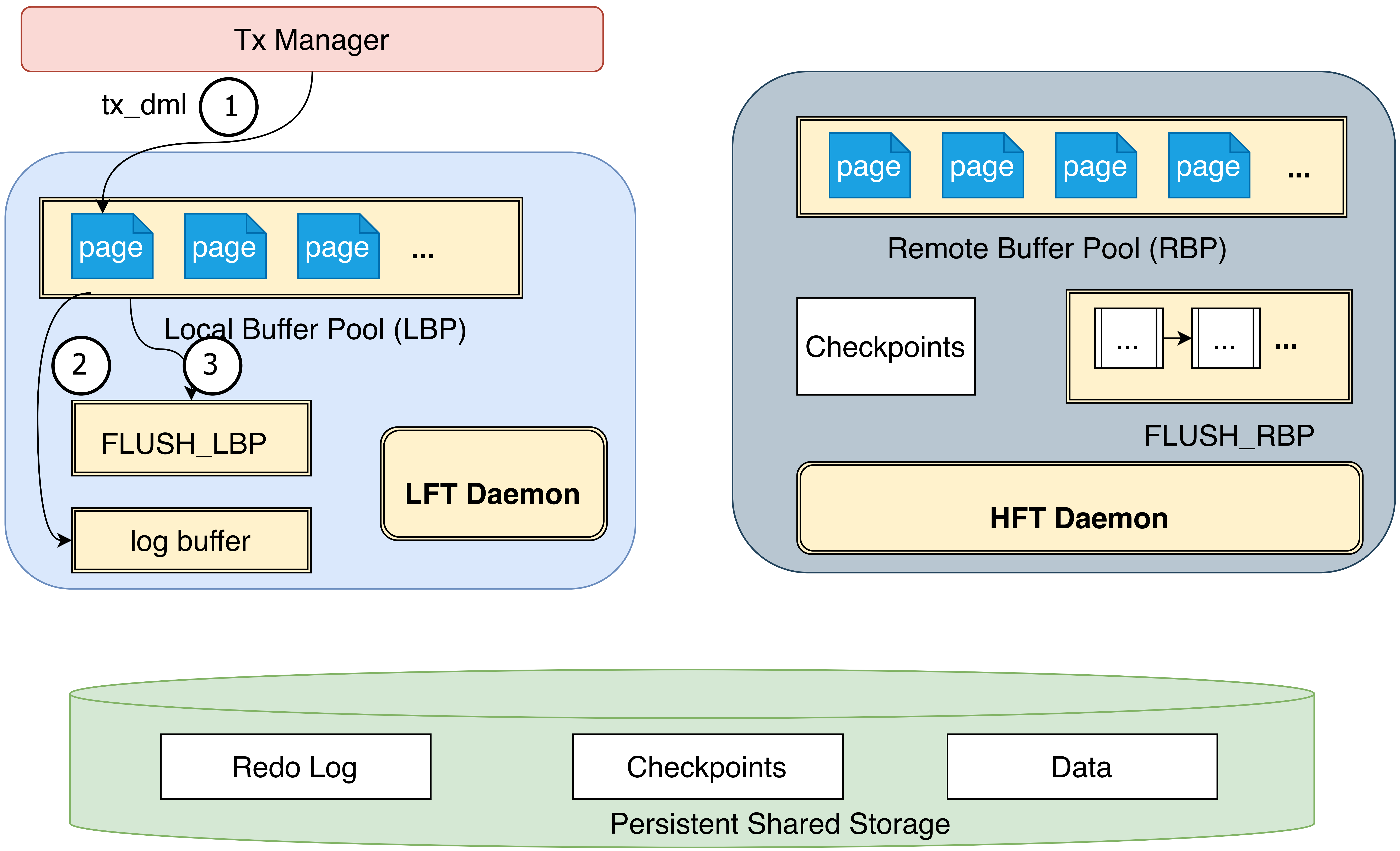


- Tier-1 checkpoints
  - Compute node flushes fine-grained checkpoints to remote buffer at high speed via RDMA
- Tier-2 checkpoints
  - Remote mem node flushes big checkpoints to storage as usual for persistence

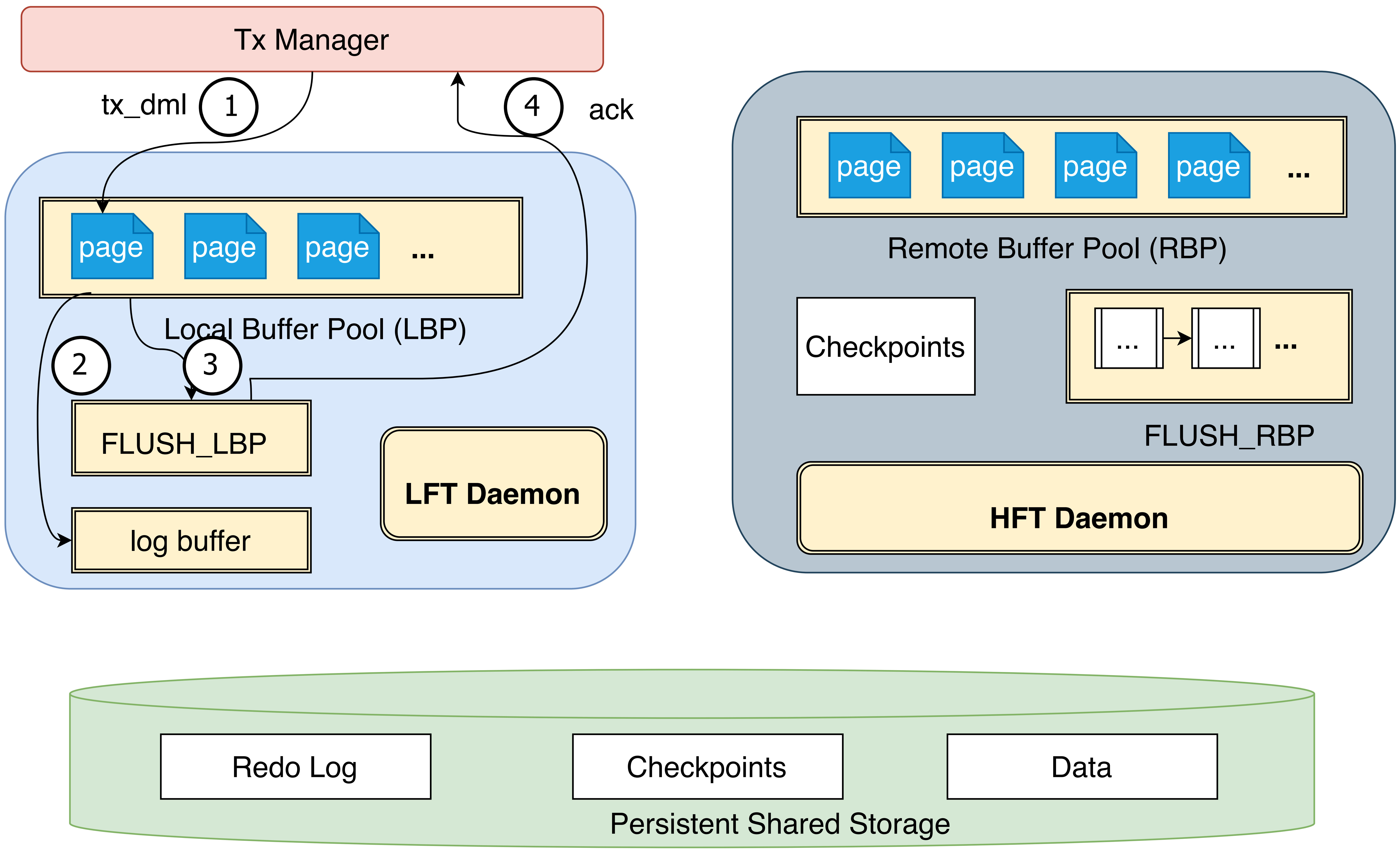
New Fault Tolerance: Two-tier ARIES Protocol



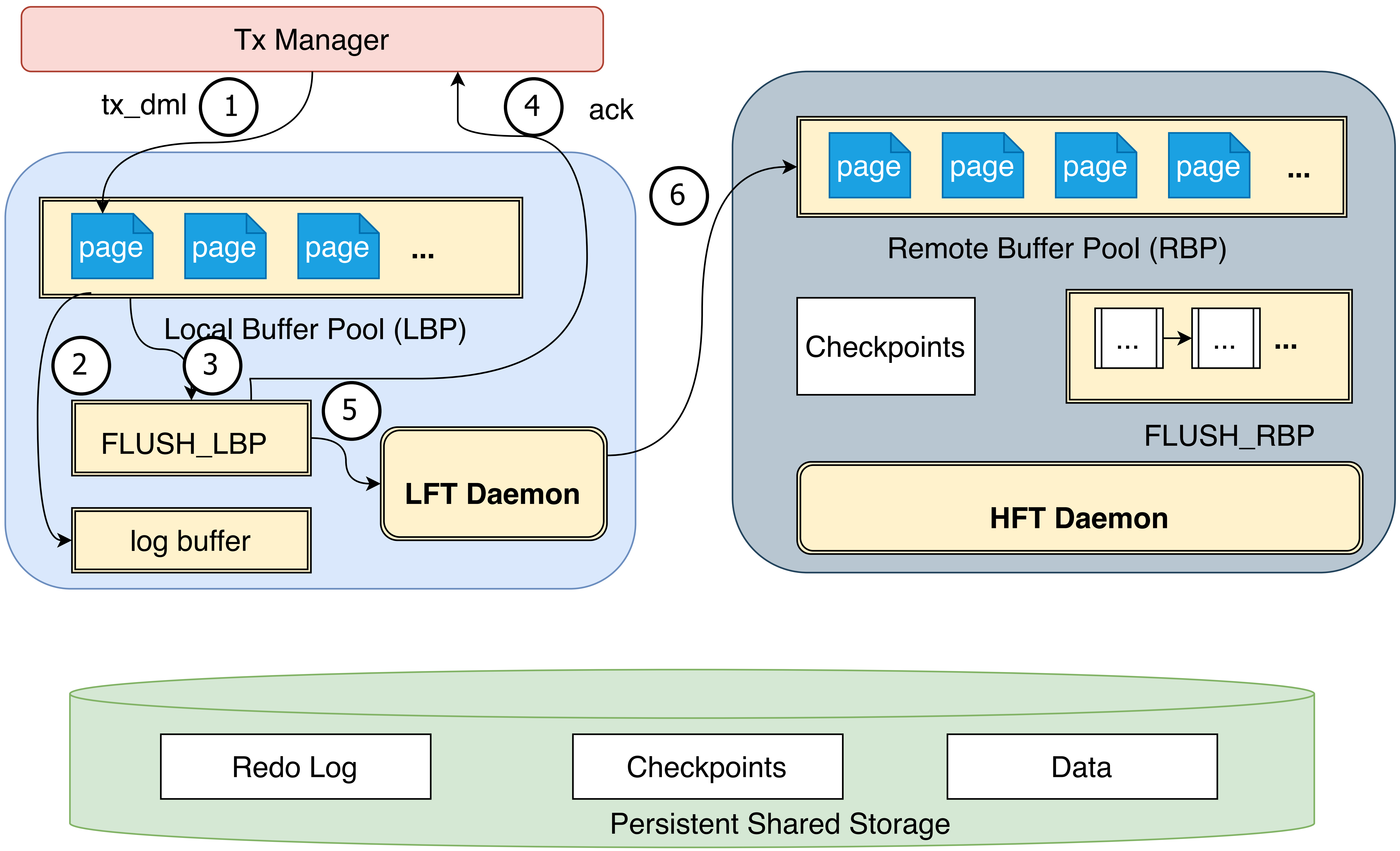
New Fault Tolerance: Two-tier ARIES Protocol



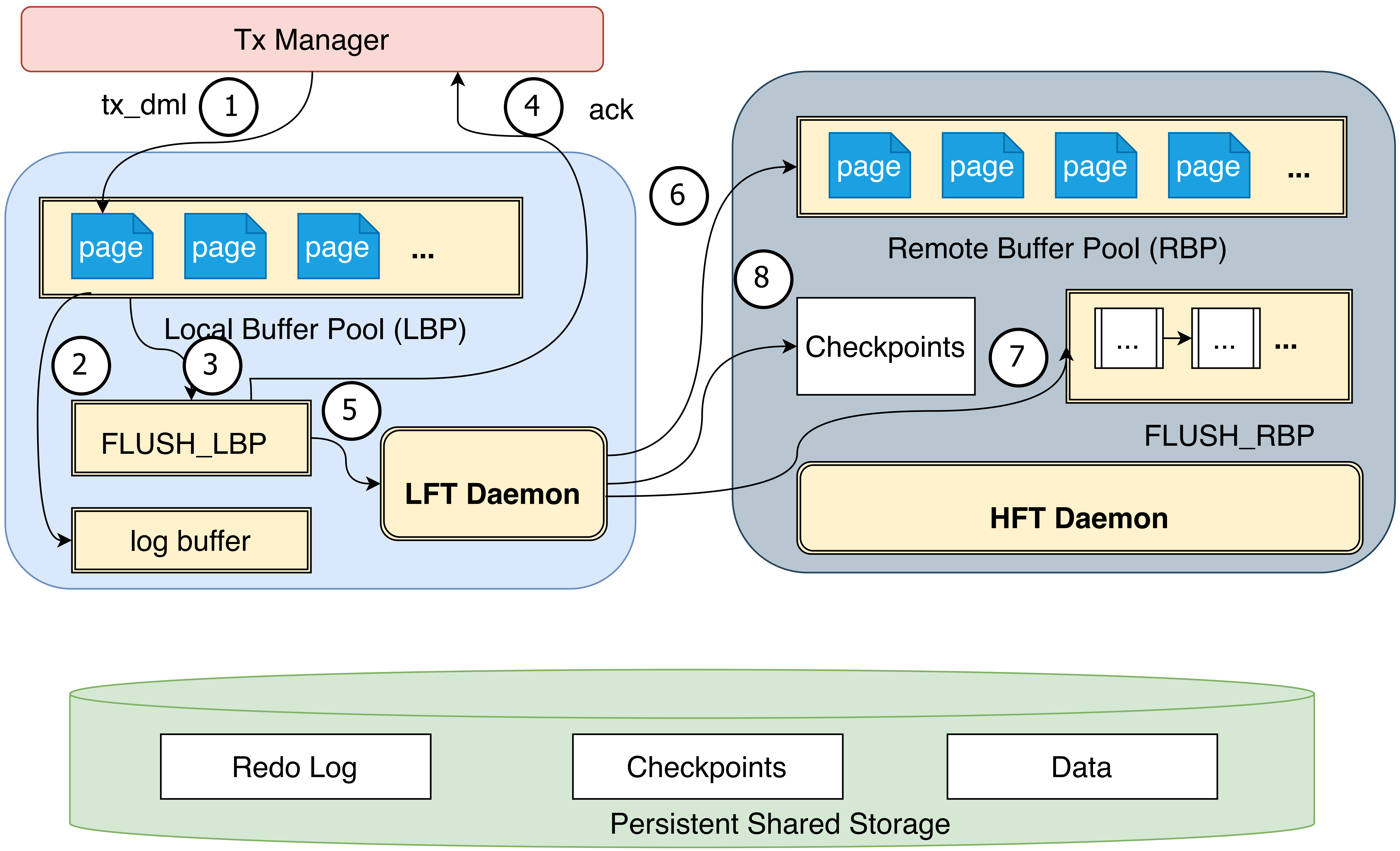
New Fault Tolerance: Two-tier ARIES Protocol



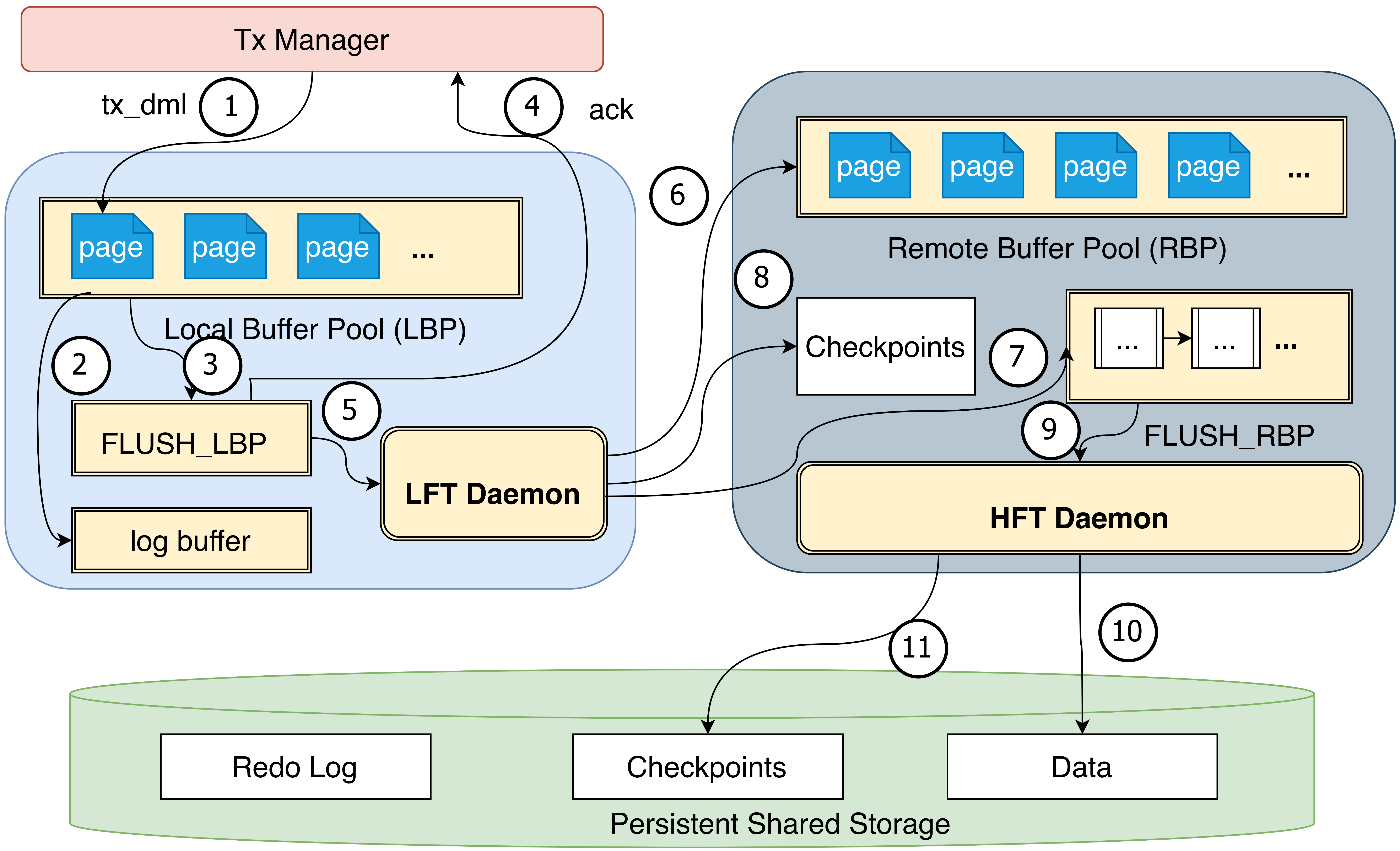
New Fault Tolerance: Two-tier ARIES Protocol



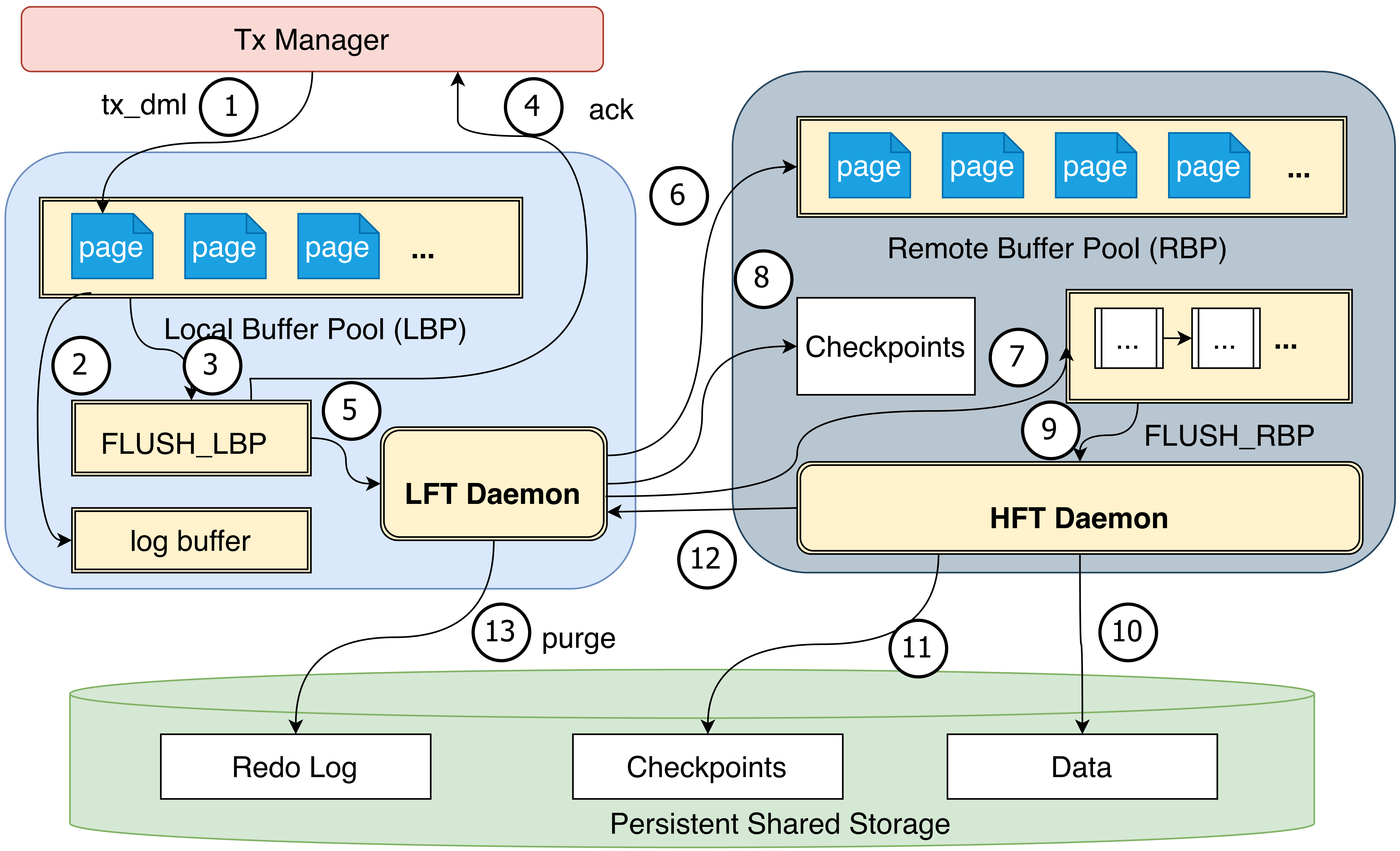
New Fault Tolerance: Two-tier ARIES Protocol



New Fault Tolerance: Two-tier ARIES Protocol



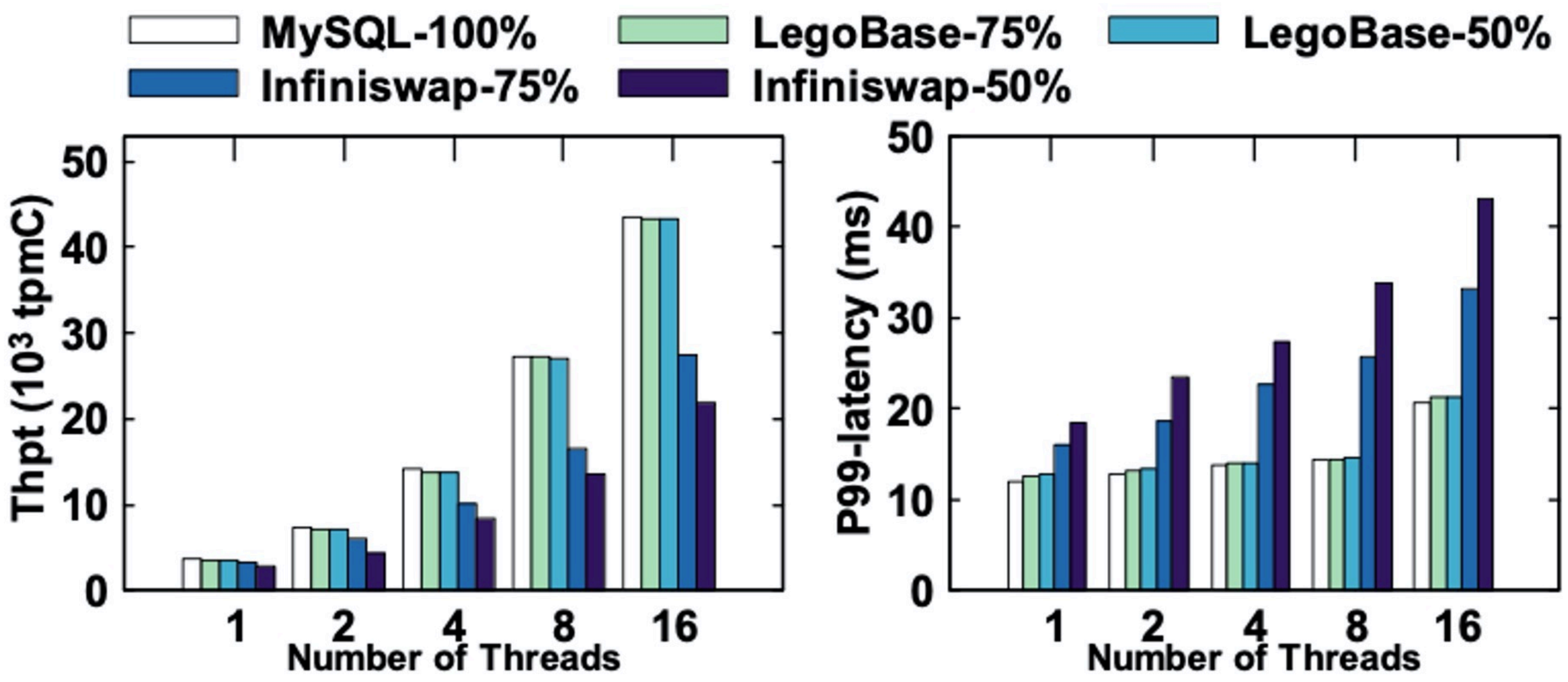
New Fault Tolerance: Two-tier ARIES Protocol



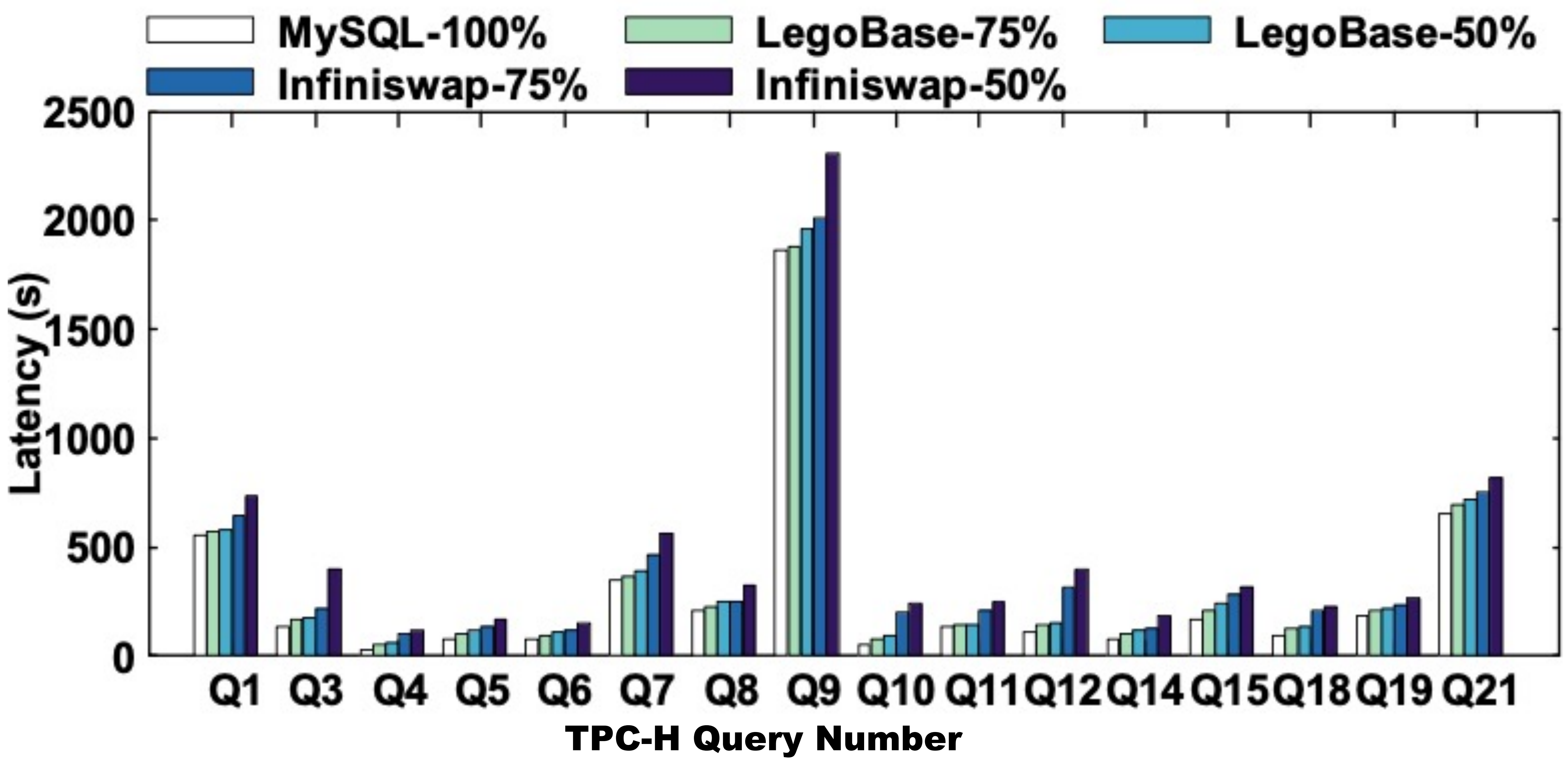
Experimental Setup

- Hardware configuration
  - Two Intel Xeon CPU E5-2682 v4 processors
  - 512GB DDR4 DRAM
  - 25Gbps Mellanox ConnectX-4 network adapter
- Workloads
  - TPC-C (20GB) / TPC-H (40GB) / Sysbench / Production workloads
- Baseline systems
  - Monolithic MySQL-8.0
  - MySQL over Infiniswap

TPC-C & TPC-H

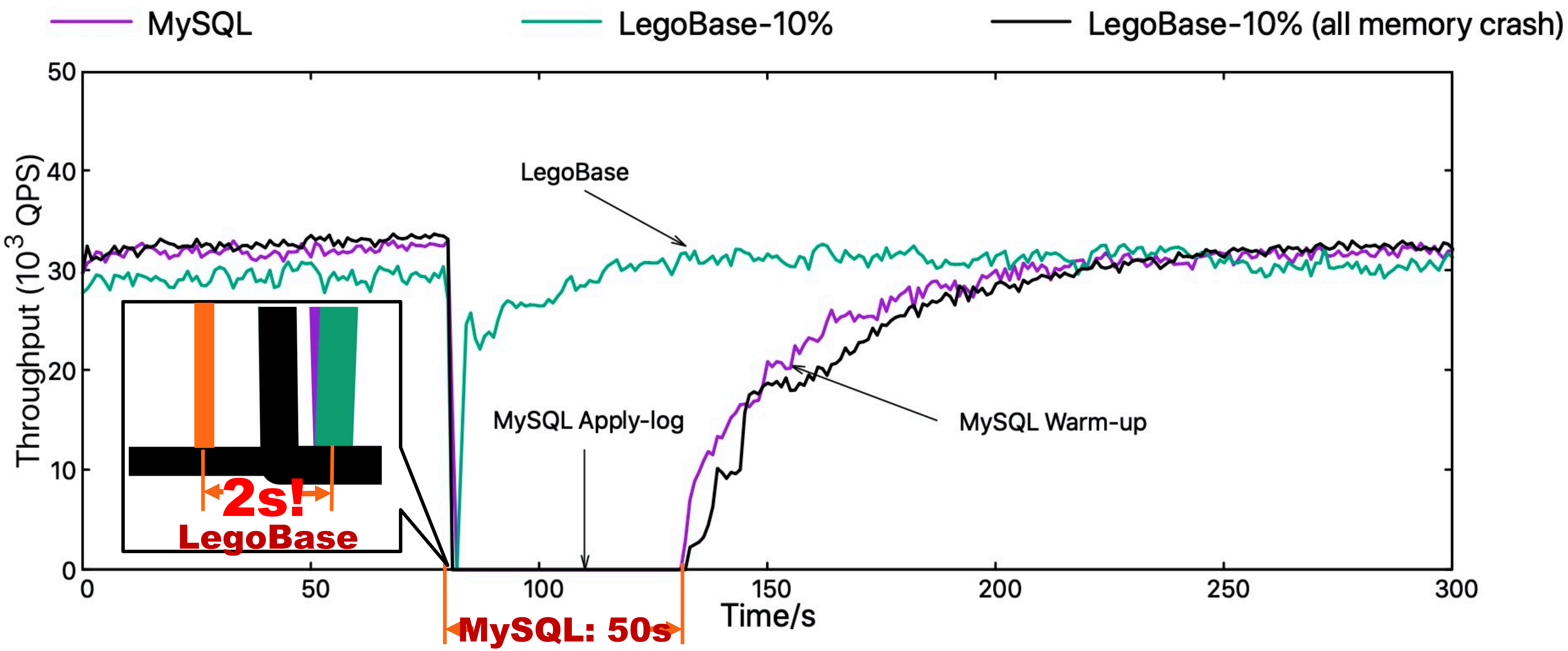


- TPC-C
  - Compared to the monolithic MySQL setup, Infiniswap worsen by up to 2.01x (2.35×) in Throughput (P99 Latency)
  - But, LegoBase brings only up to 3.82% (4.42%) loss in Throughput (P99 Latency)



- TPC-H
  - LegoBase-75%’ s latencies are closer to the ones of MySQL-100%
  - LegoBase outperforms MySQL over infiniswap: e.g., runs up to 75.7% faster for Q11

Fast State Recovery and Warm-Up



- When all/remote memory crashes, LegoBase resorts to the normal MySQL recovery process.
- But, with only local node crashes, LegoBase can re-use vast majority of remote pages, and significantly reduces the state recovery and warm-up time costs.
- This can also benefit efficient planned hardware re-configurations.

Conclusion

- LegoBase is a novel memory-disaggregated cloud-native database architecture
- LegoBase is able to scale CPU and memory capacities independently with comparable performance as the monolithic setup without using remote memory
- LegoBase can achieve faster state recovery and is more cost-effective than state-of-the-art baselines

Thank you !

Alibaba Cloud
Worldwide Cloud Services Partner

Yingqiang Zhang

yingqiang.zyq@alibaba-inc.com

USTC.CHINA ADSLAB
先 进 数 据 系 统 实 验 室

Chaoyi Ruan

rcy@mail.ustc.edu.cn