CS 839: Design the Next-Generation Database

Lecture 20: OLTP in Cloud

Xiangyao Yu

4/2/2020

Discussion Highlights

SmartNIC for join

- Filtering, hash table, indexing
- Network traffic scheduling for shuffling (reduce the problem of bursty traffic)
- Hash table in SmartNIC?
- Sort in SmartNIC
- Data partitioning

Parameters [million tuples per second]
RHJ: $P_{\text{scan}} = 225$, $P_{\text{Part}} = 120$, $P_{\text{net}} = 1024$, $P_{\text{build}} = 120$, $P_{\text{probe}} = 225$
SMJ: $P_{\text{part}} = 78$, $P_{\text{sort}} = 75$, $P_{\text{net}} = 1024$, $P_{\text{merge}} = 45$, $P_{\text{scan}} = 225$

HW/SW techniques to improve performance of sort-merge join

- Equivalent performance after removing bottlenecks? (Not necessarily)
- Hardware acceleration for the sort and merge

Radix join to achieve theoretical maximum performance

- Communication powered by SmartNICs/RDMA (network scheduling for shuffling)
- Hash partitioning logic in SmartNIC

Today's Paper

Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases

Alexandre Verbitski, Anurag Gupta, Debanjan Saha, Murali Brahmadesam, Kamal Gupta, Raman Mittal, Sailesh Krishnamurthy, Sandor Maurice, Tengiz Kharatishvili, Xiaofeng Bao

Amazon Web Services

ABSTRACT

Amazon Aurora is a relational datab
workloads offered as part of Amazon W
this paper, we describe the architecture o
considerations leading to that architecture
constraint in high throughput data proc
compute and storage to the network.
architecture to the relational database to
most notably by pushing redo processing
out storage service, purpose-built for A
doing so not only reduces network traffic
crash recovery, failovers to replicas wi
fault-tolerant, self-healing storage. We th
achieves consensus on durable state a

SIGMOD 2017

Amazon Aurora development team wins the 2019 ACM SIGMOD Systems Award*

By Werner Vogels on 04 July 2019 10:00 AM | Permalink | Comments (2)

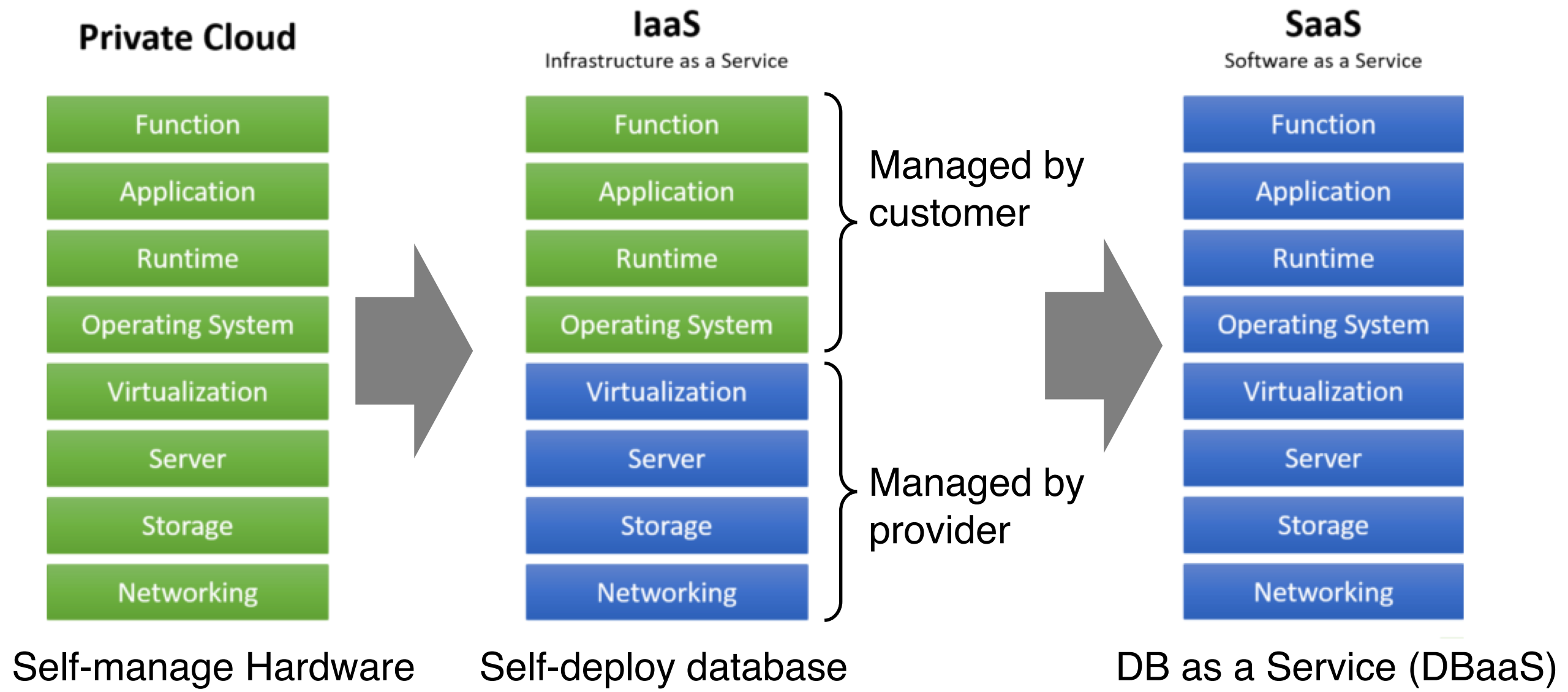
Cloud Computing

aws

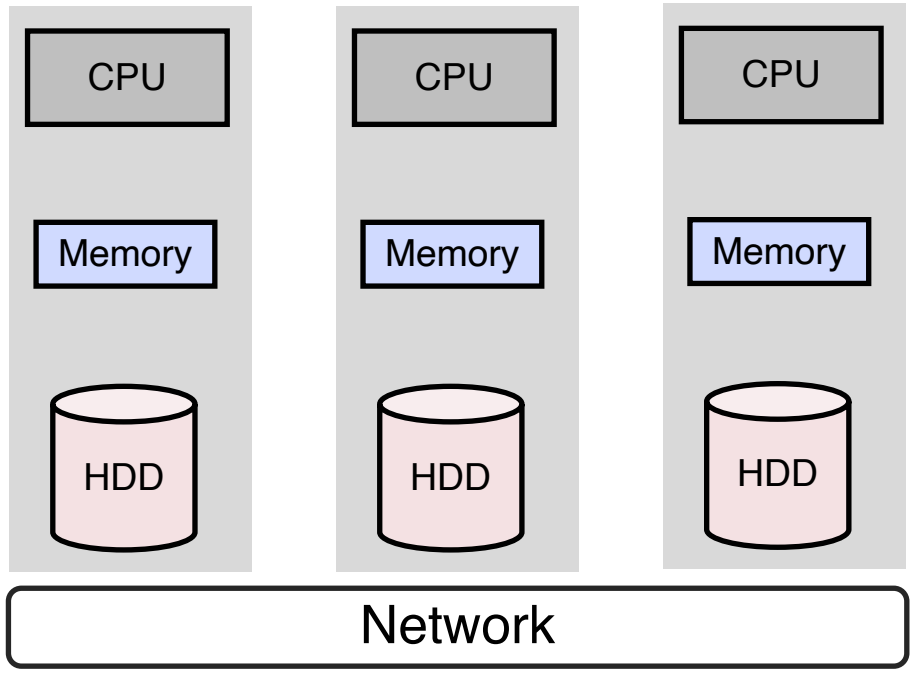
Microsoft Azure

Google Cloud

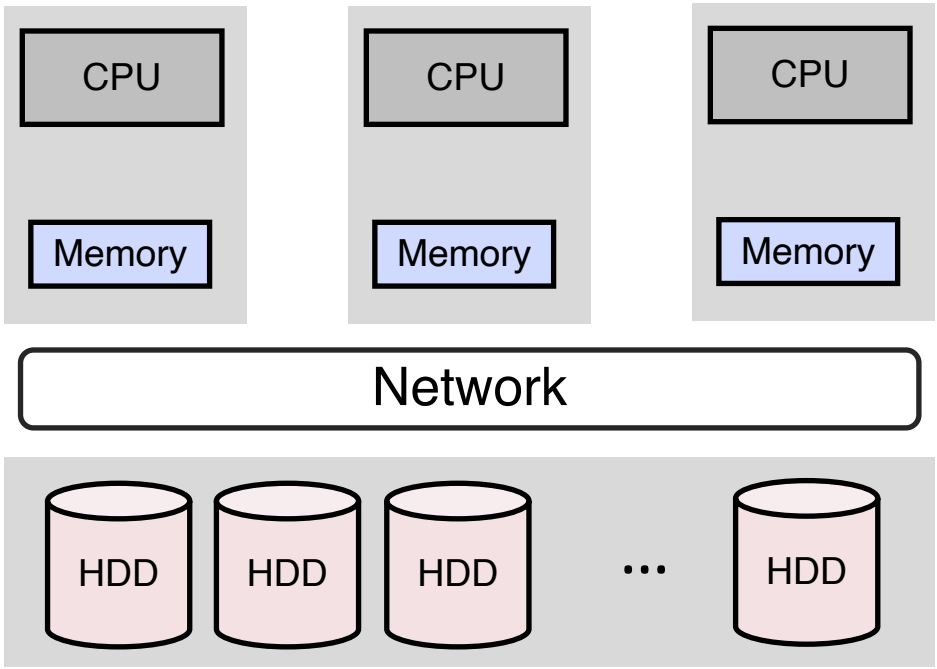
Cloud Computing



Shared Nothing vs. Shared Disk

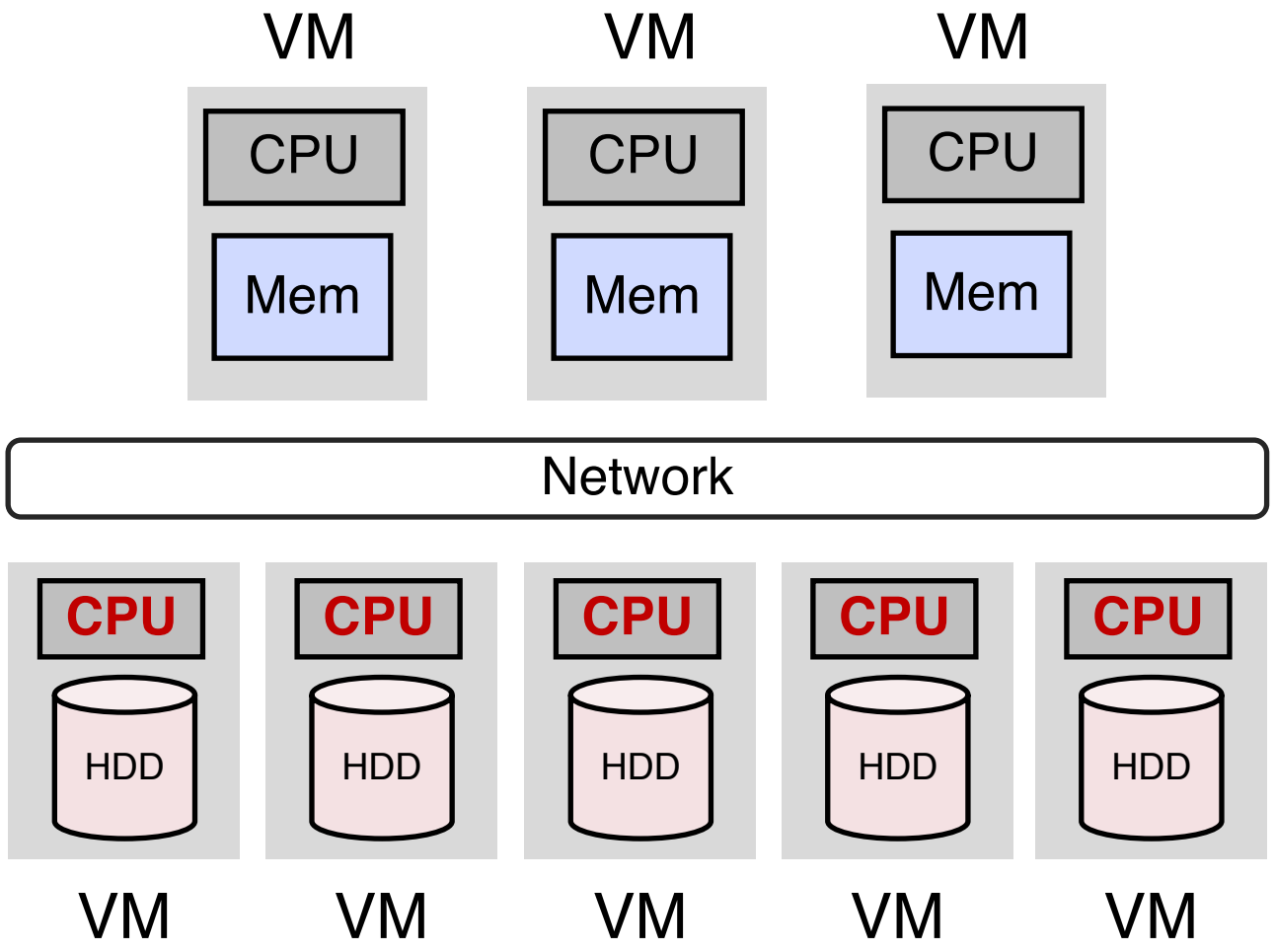


Shared Nothing



Shared Disk

Cloud Storage Disaggregation



Storage disaggregation

- Independent management and scaling of compute and storage
- Cost reduction

Smartness in Storage

- Storage nodes contain CPUs for computation

Computation Pushdown in Cloud OLTP

Pushdown to cloud storage?

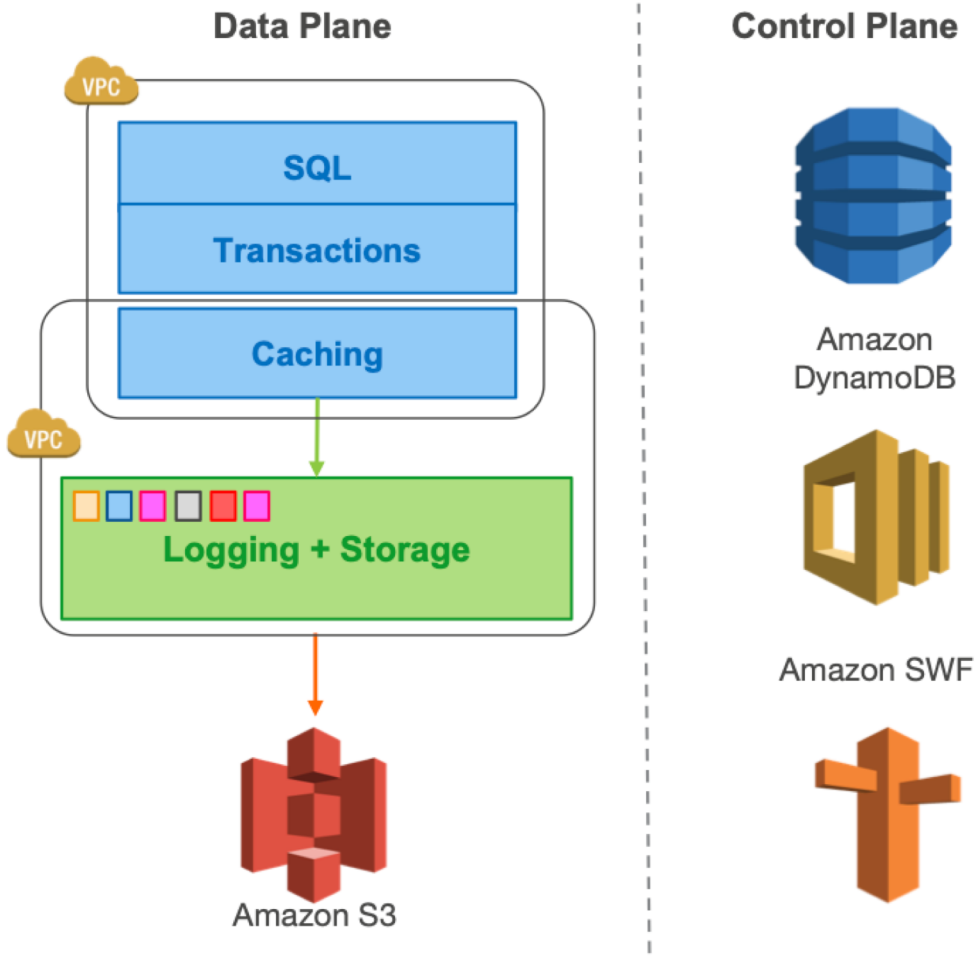
- Concurrency control
- Indexing
- Buffer manager
- Logging

Computation Pushdown in Cloud OLTP

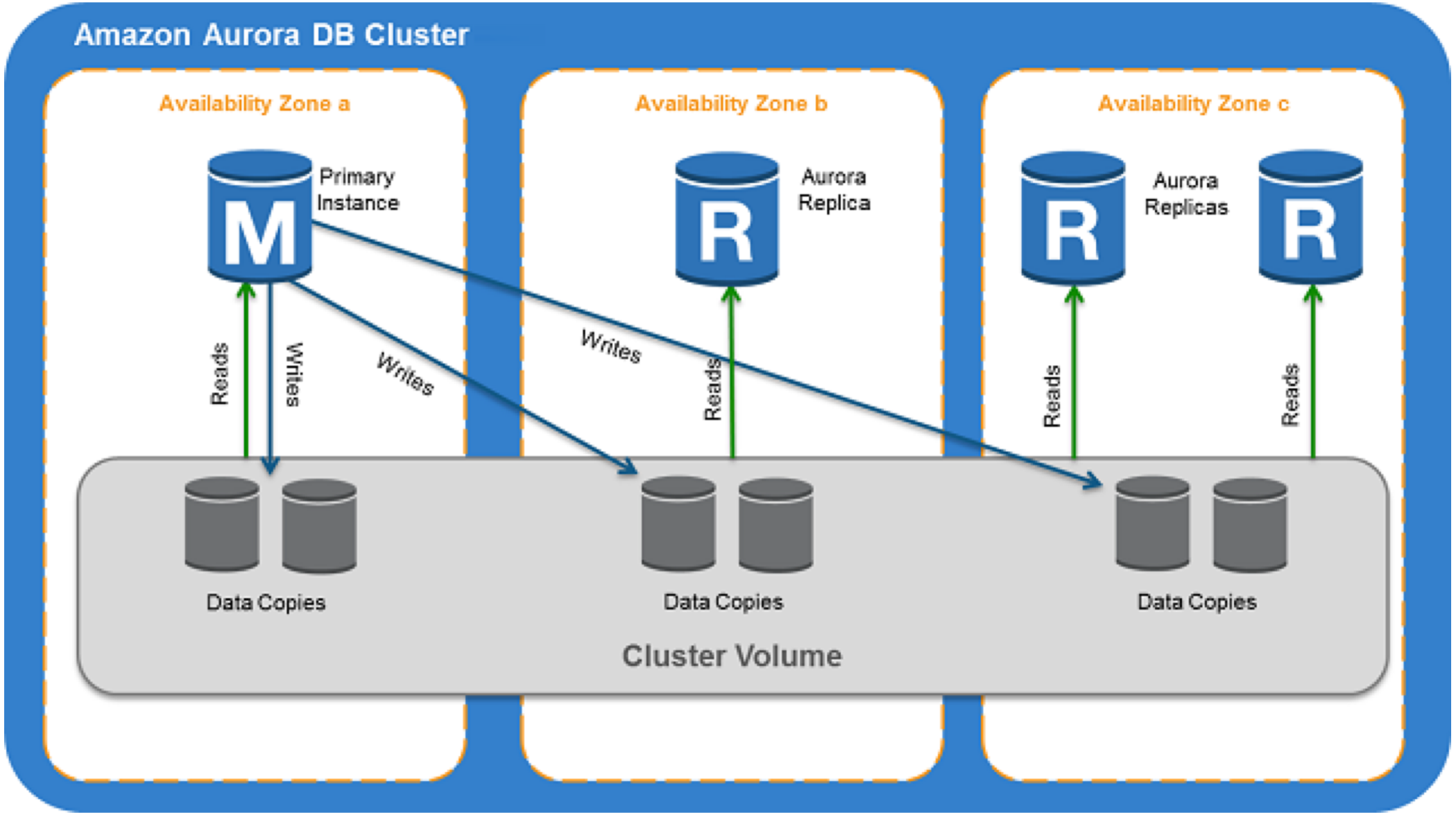
Pushdown to cloud storage?

- Concurrency control
- Indexing
- Buffer manager
- Logging

Push redo processing into the storage service



Aurora – Single Master



Quorum-Based Voting Protocol

Data replicated into V copies

A write must acquire votes from V_w copies

A read must acquire votes from V_r copes

$V_w + V_w > V$ => $V_w > V / 2$

$V_r + V_w > V$

For three copies

$V_w \geq 2$

$V_r \geq 2$

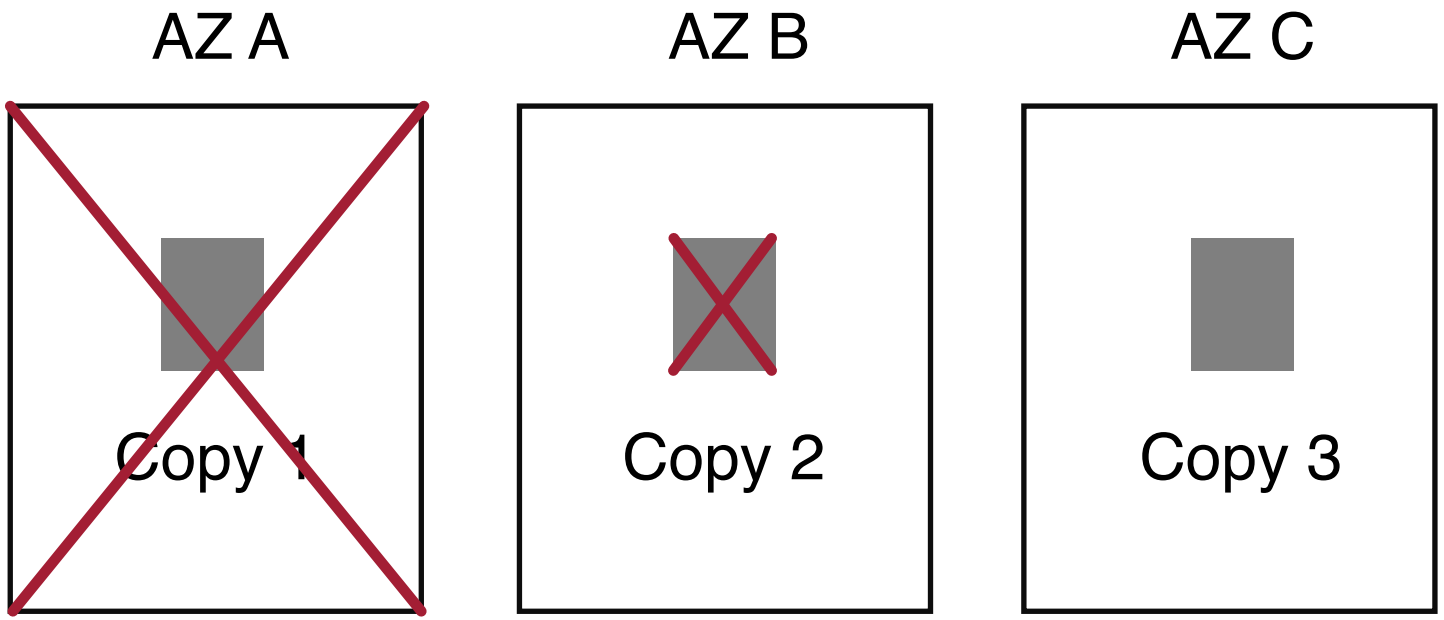
Copy 1 Copy 2 Copy 3

For six copies

$V_w \geq 4$

$V_r \geq 3$

3-Way Replication

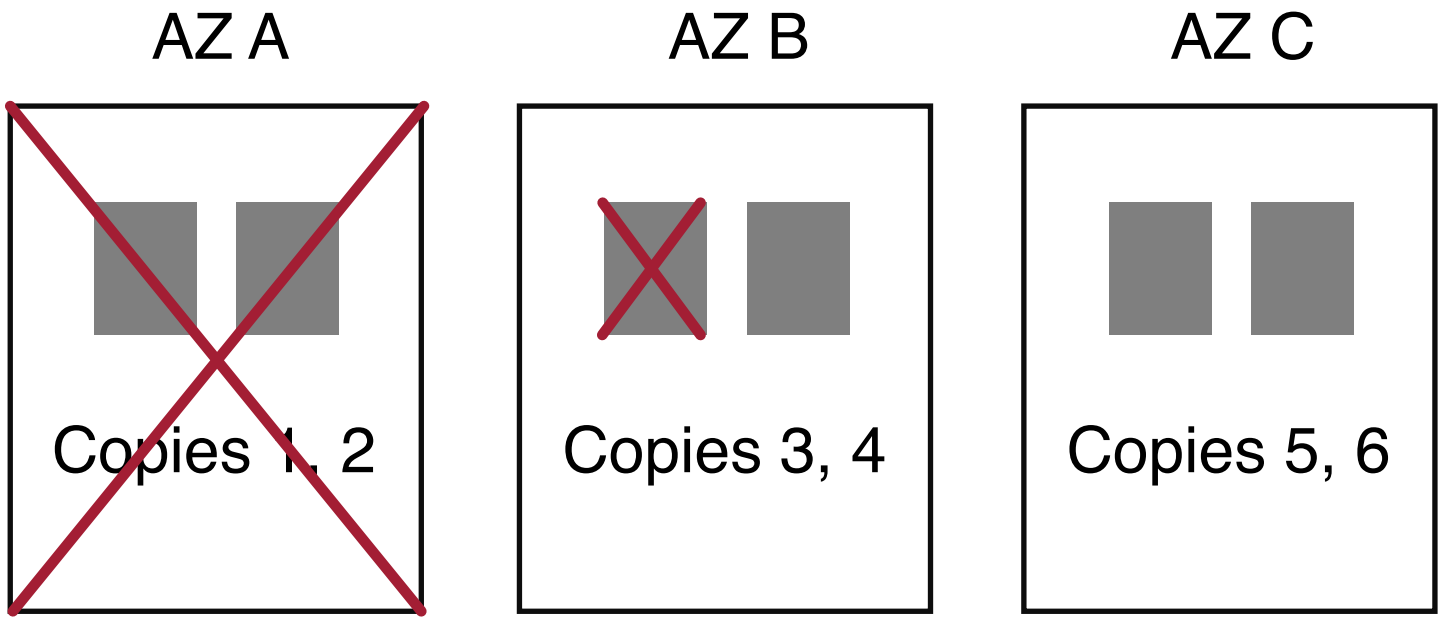


AZ: Availability zone

- AZs fail independently

Data is unavailable if one AZ is unavailable and one other copy is unavailable

6-Way Replication



Can read if one AZ fails and one more node fails

- Allow to rebuild a write quorum by adding additional replica

Can write if one AZ fails

Segmented Storage

Availability is determined by

- MTTF: Mean time to failure
- MTTR: Mean time to repair

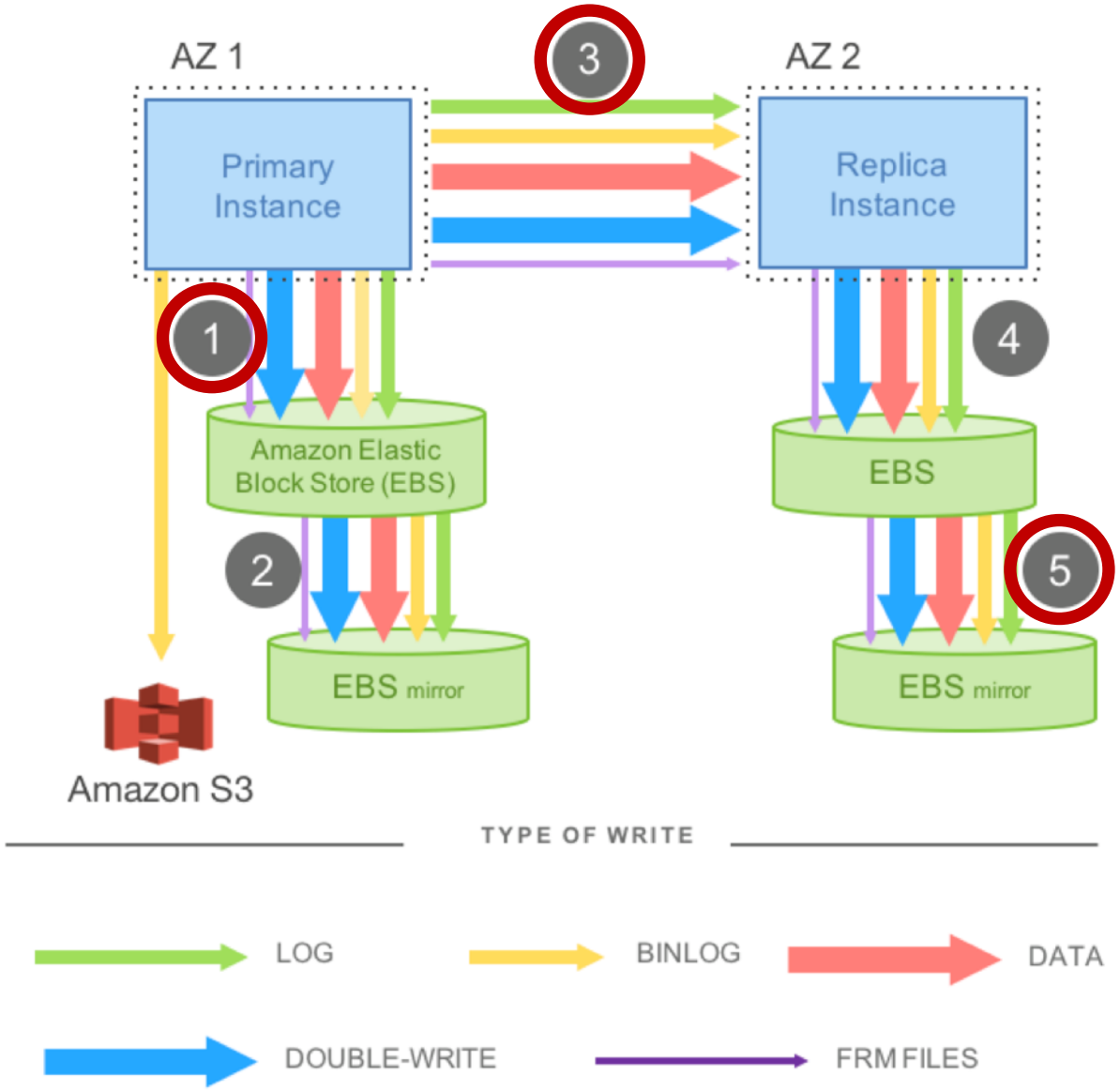
Maximize availability

=> Minimize MTTR (MTTF is hard to reduce)

Segment: 10 GB block. Basic unit of failure and repair

Protection Group (PG): Six replication copies of a segment

Network IO in MySQL



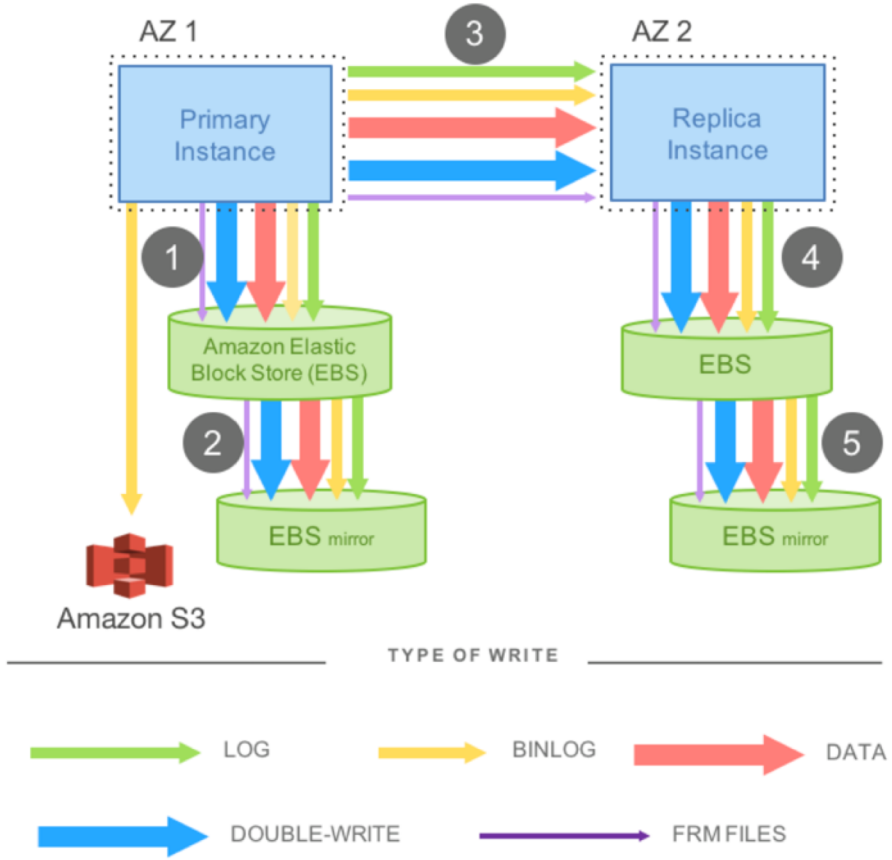
IO traffic

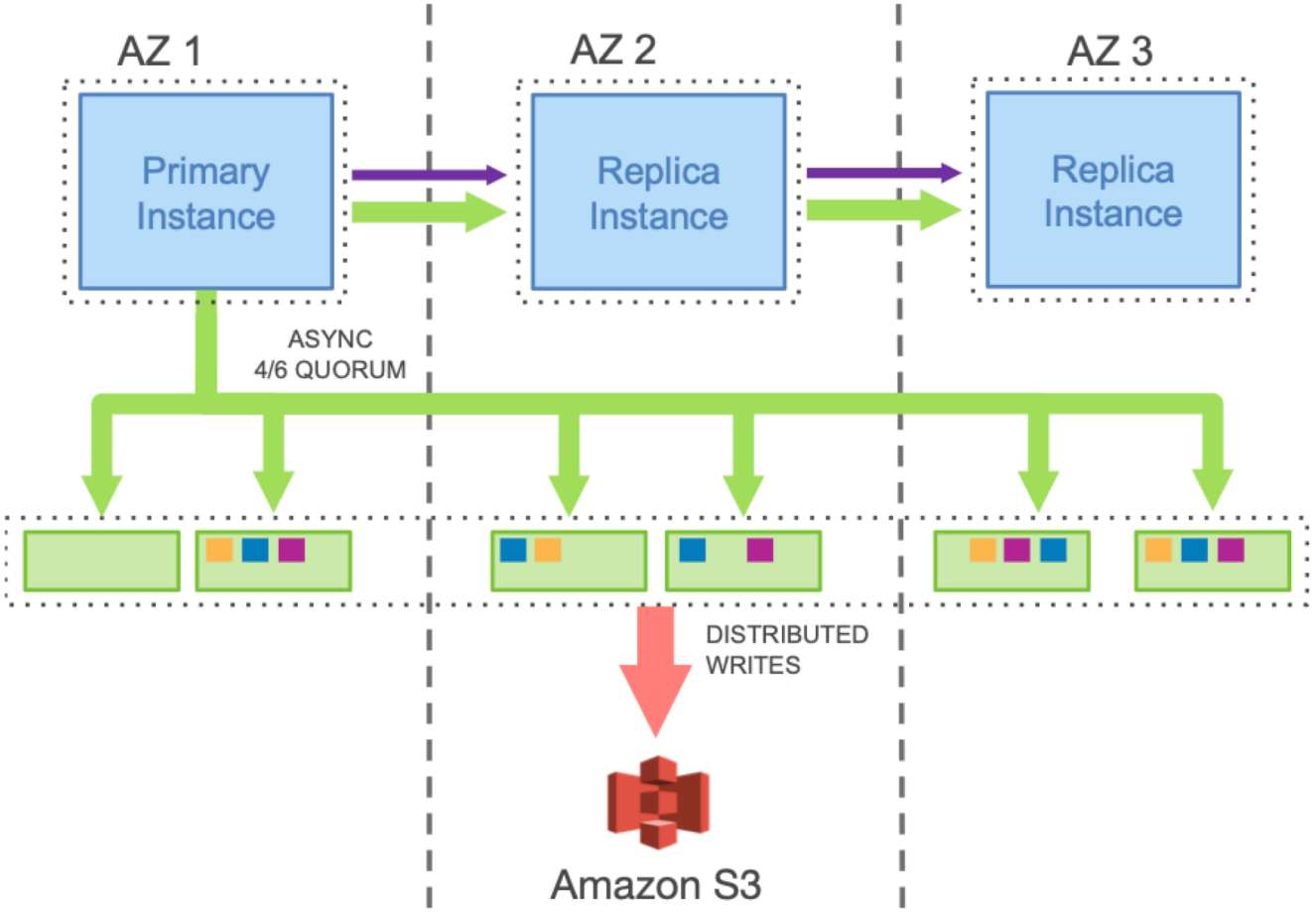
- **REDO Log**
- Binary log
- **Data**
- Double-write
- metadata (FRM)

Latency

- Steps 1, 3, and 5 are sequential and synchronous

MySQL vs. Aurora





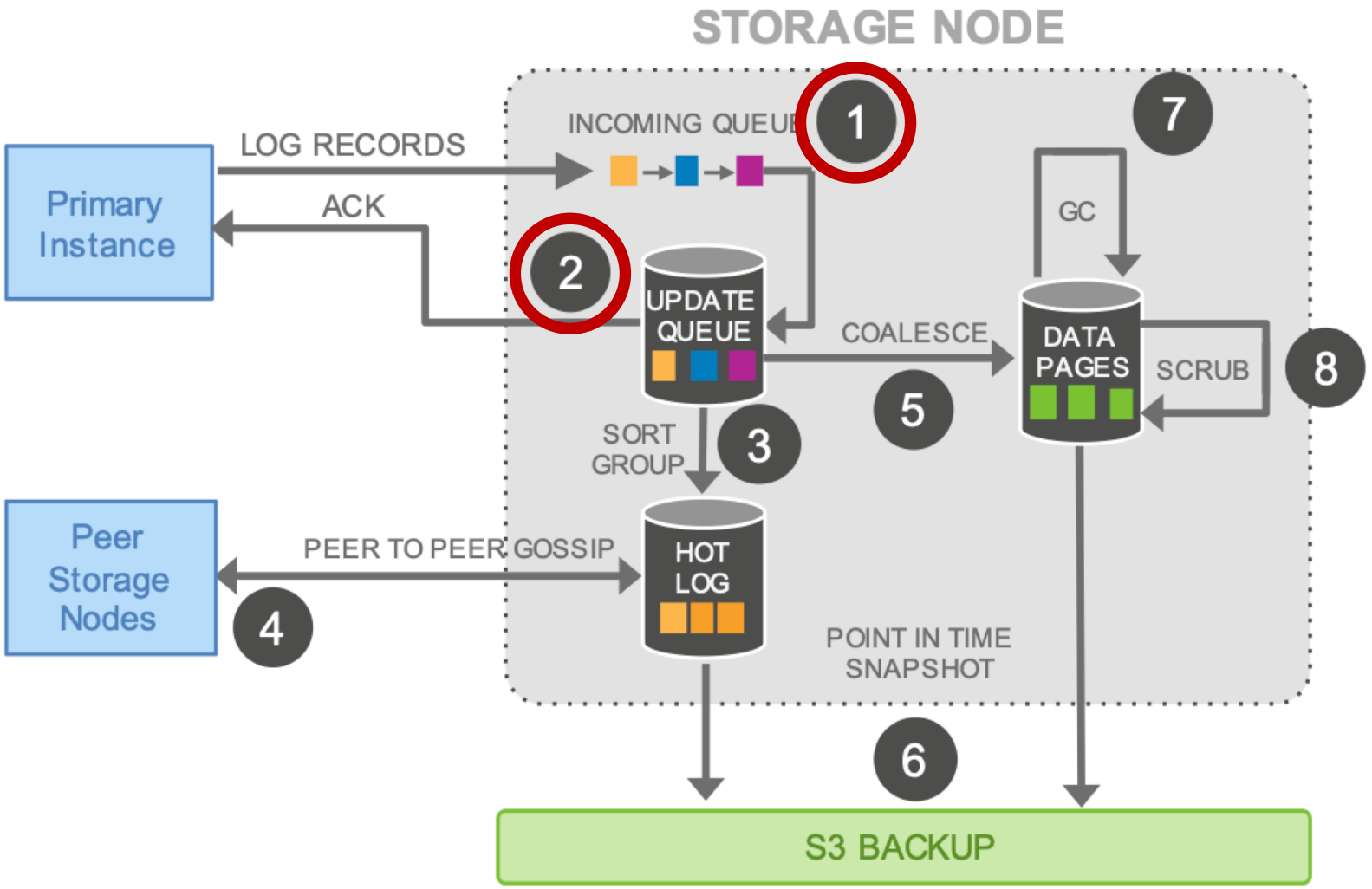
Aurora: send only REDO log to storage

MySQL vs. Aurora – Network IO

Table 1: Network IOs for Aurora vs MySQL

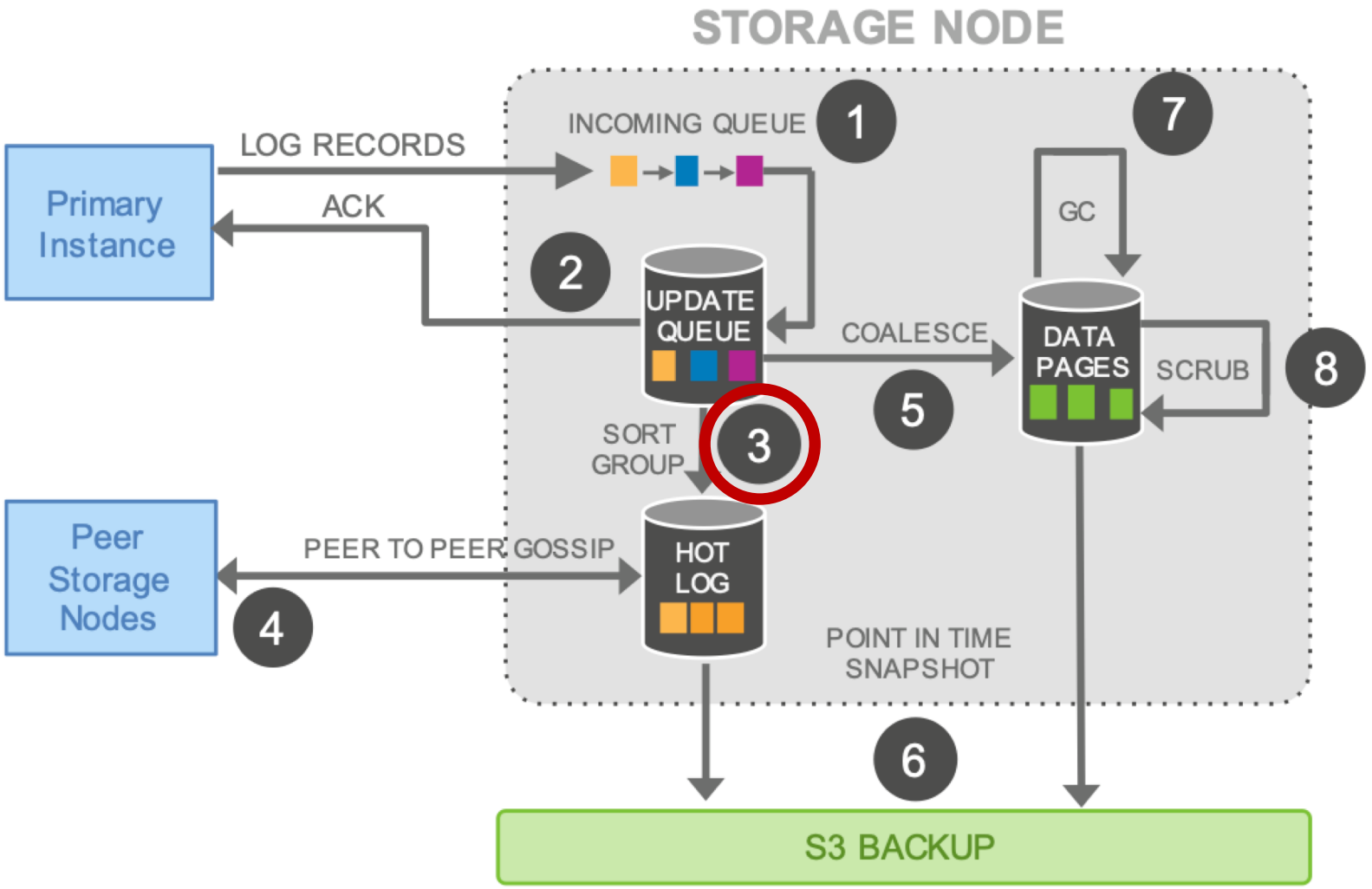
Configuration	Transactions	IOs/Transaction
Mirrored MySQL	780,000	7.4
Aurora with Replicas	27,378,000	0.95

Storage Node



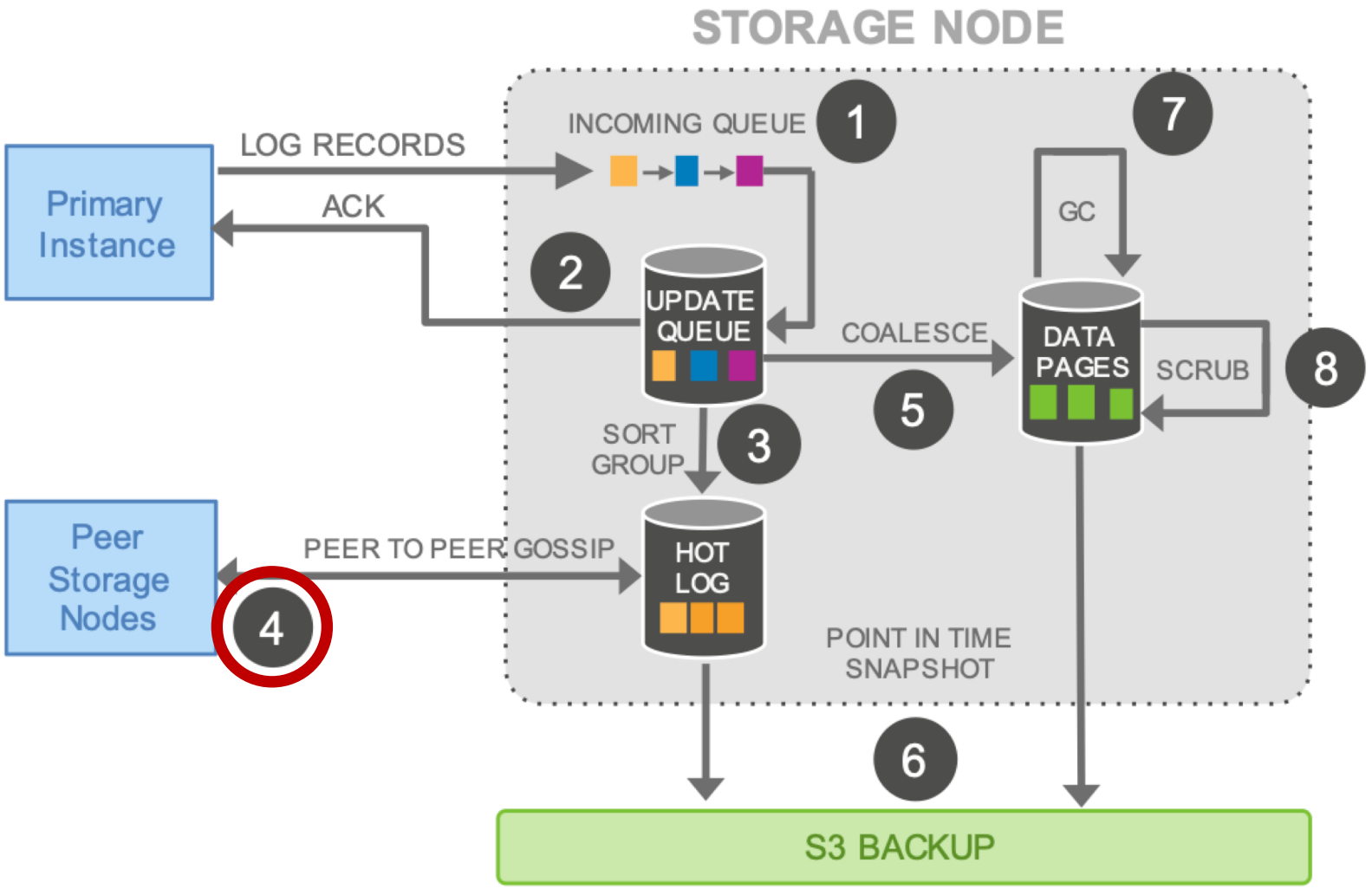
Only Steps 1 & 2 are in the foreground path

Storage Node



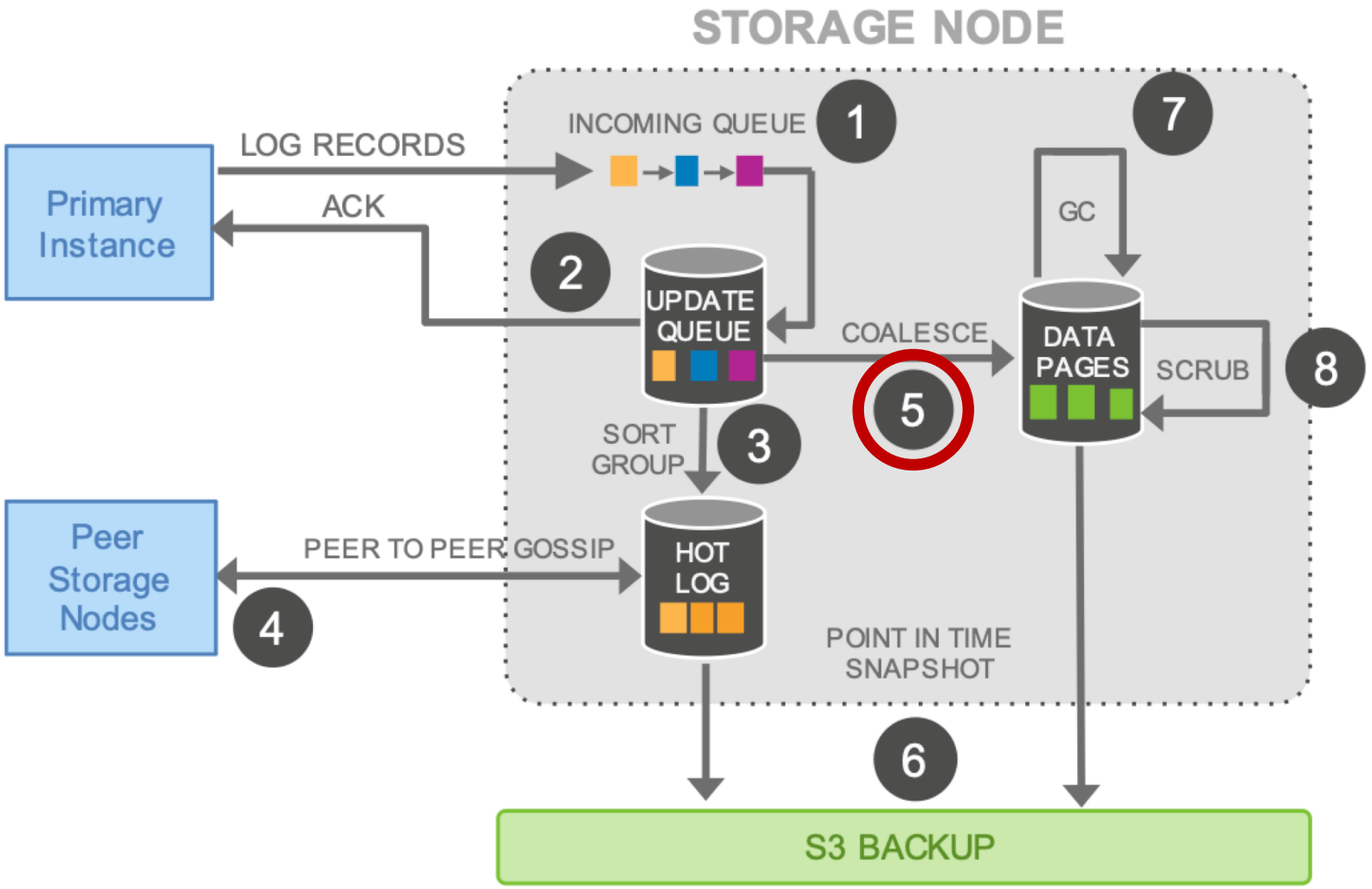
Identify gaps in the log

Storage Node



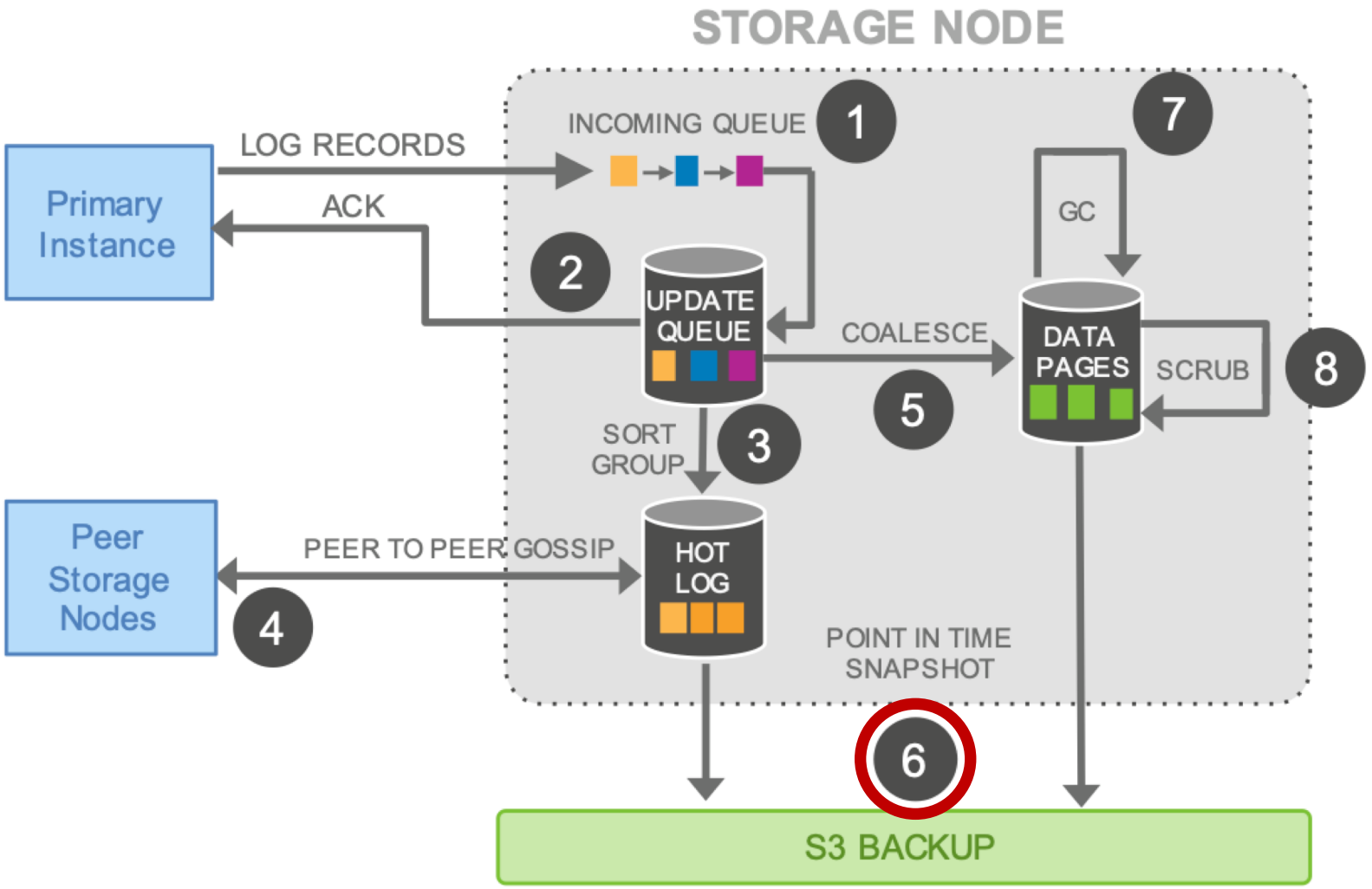
Gossip with peers to fill gaps

Storage Node



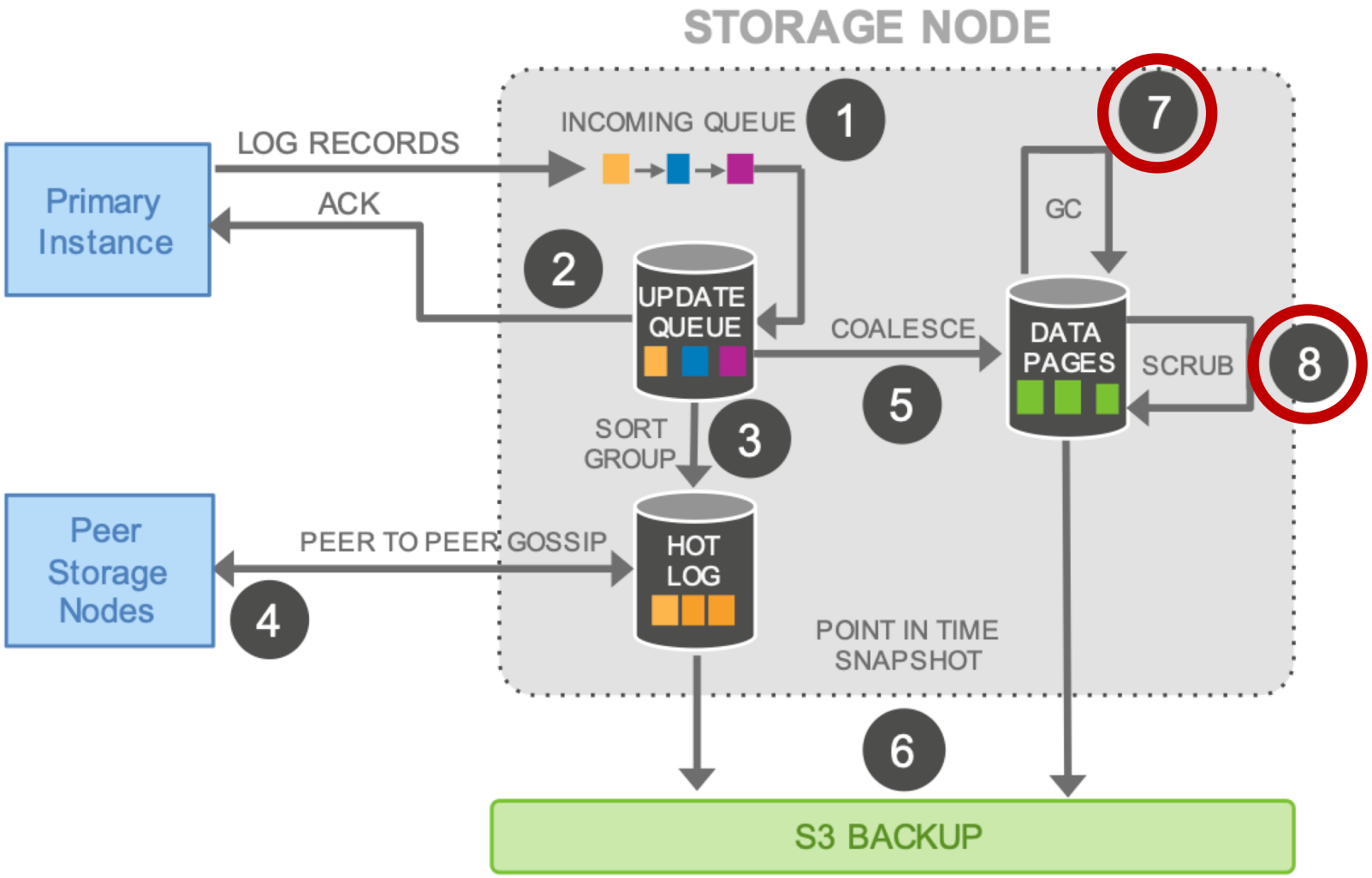
Coalesce log records into data pages

Storage Node



Periodically stage log and pages to S3

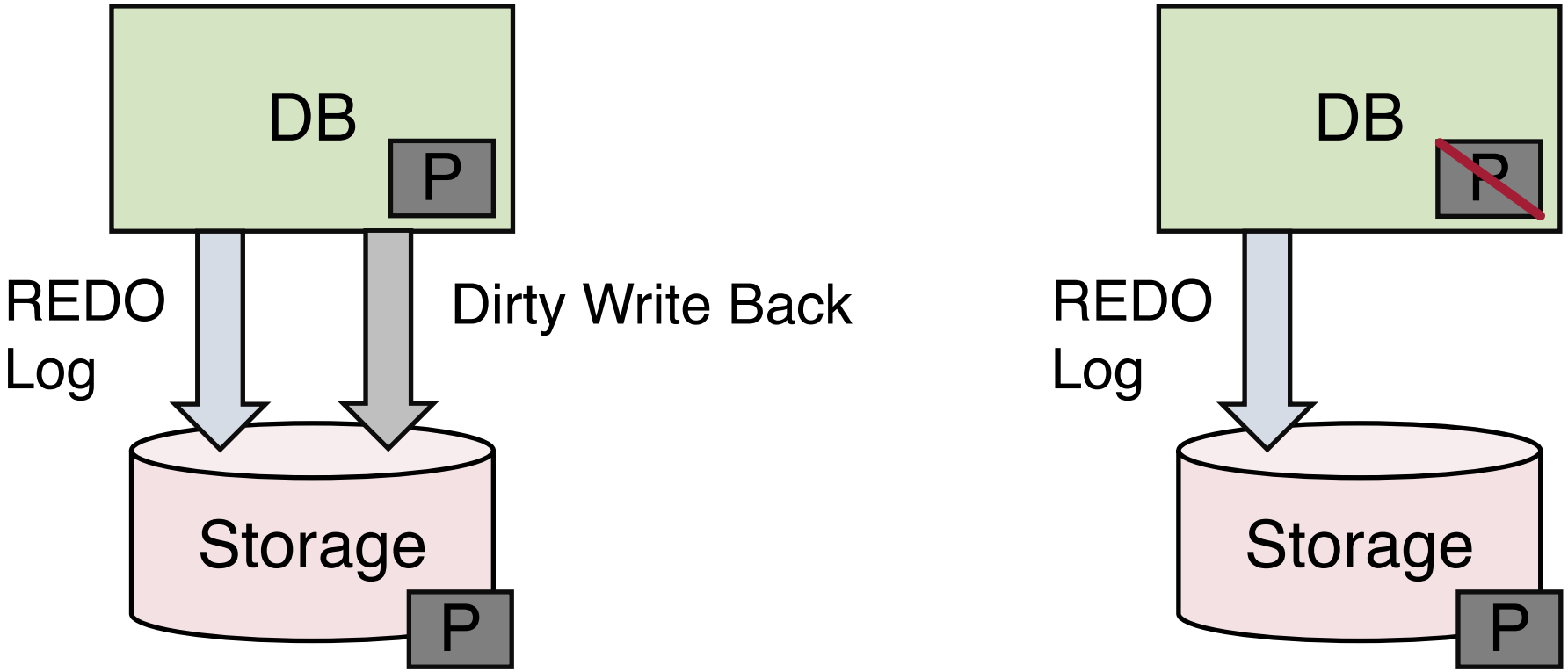
Storage Node



Periodically garbage collect old versions and periodically validate CRC code on pages

* Cyclic redundancy check (CRC) is an error-detecting code

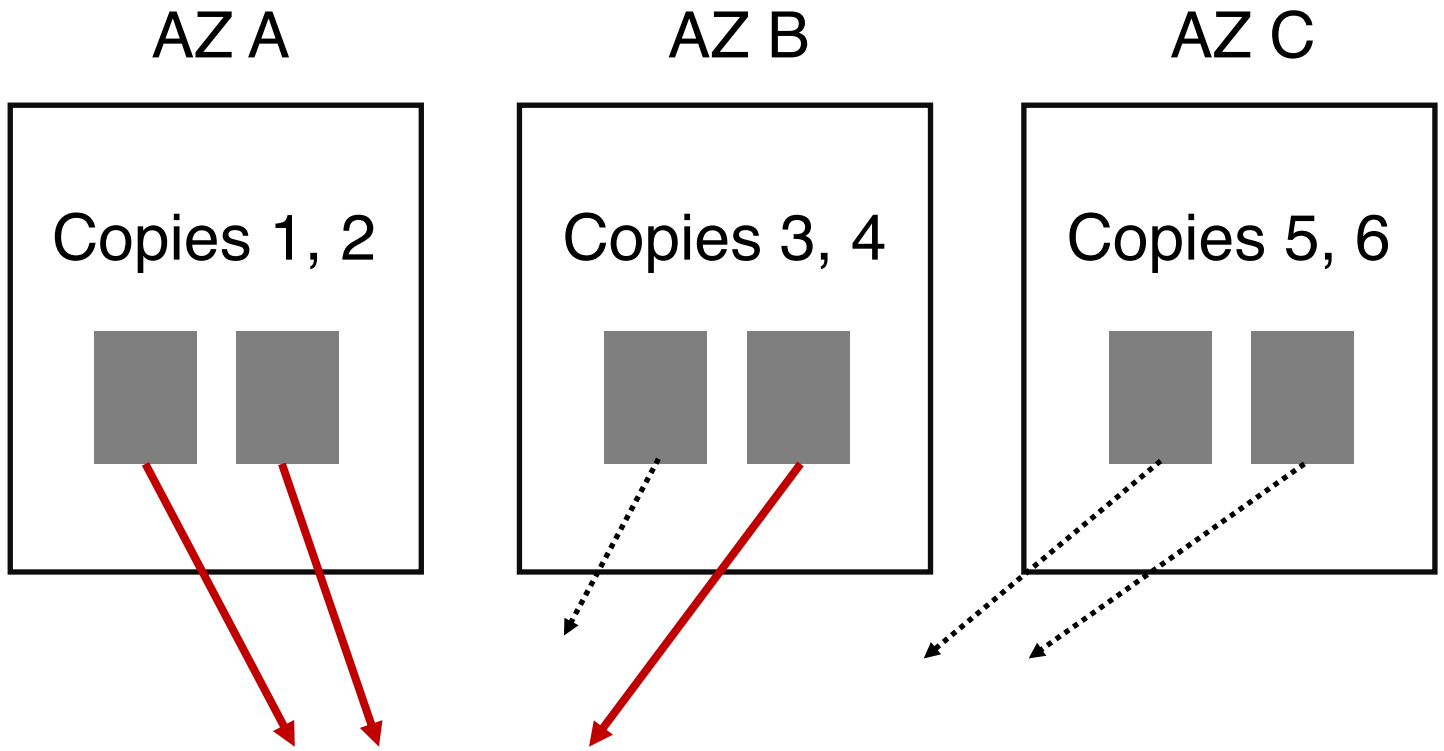
Dirty Evict



A dirty page can be evicted if all changes in the page have been hardened in the log

Read from storage upon a cache miss

Read from One Quorum

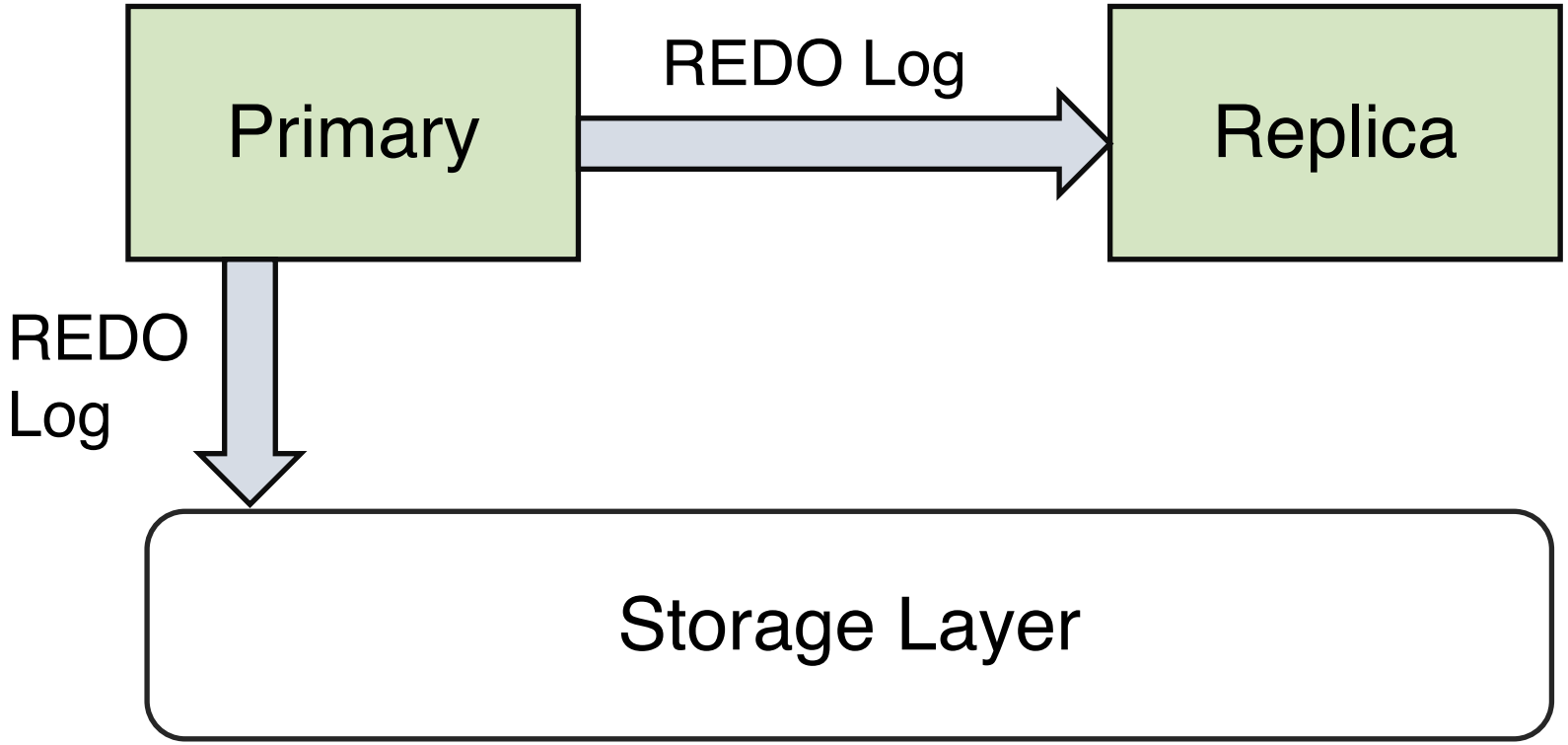


Three votes to read data

The DB server knows which node contains the latest value

=> A single read from the update-to-date node

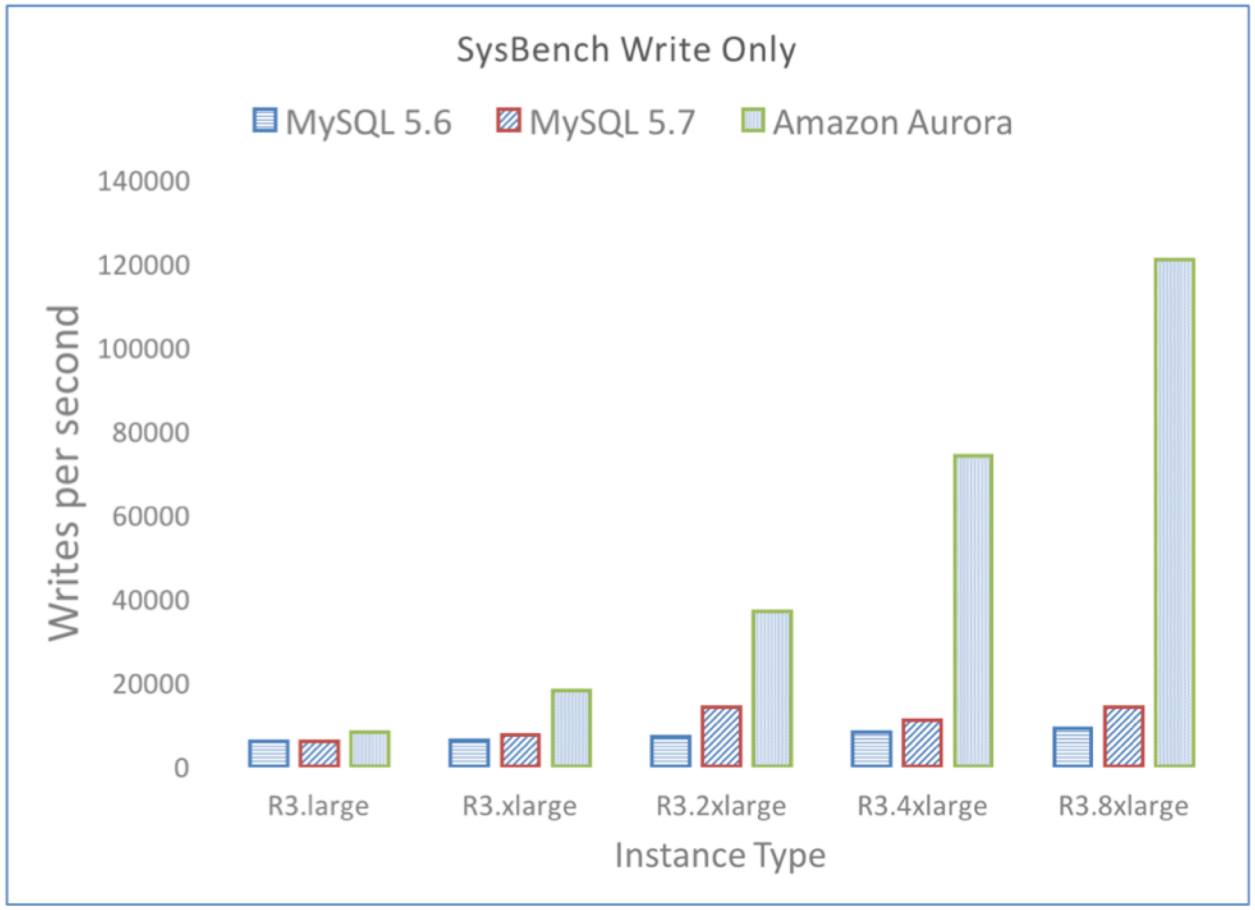
Replication

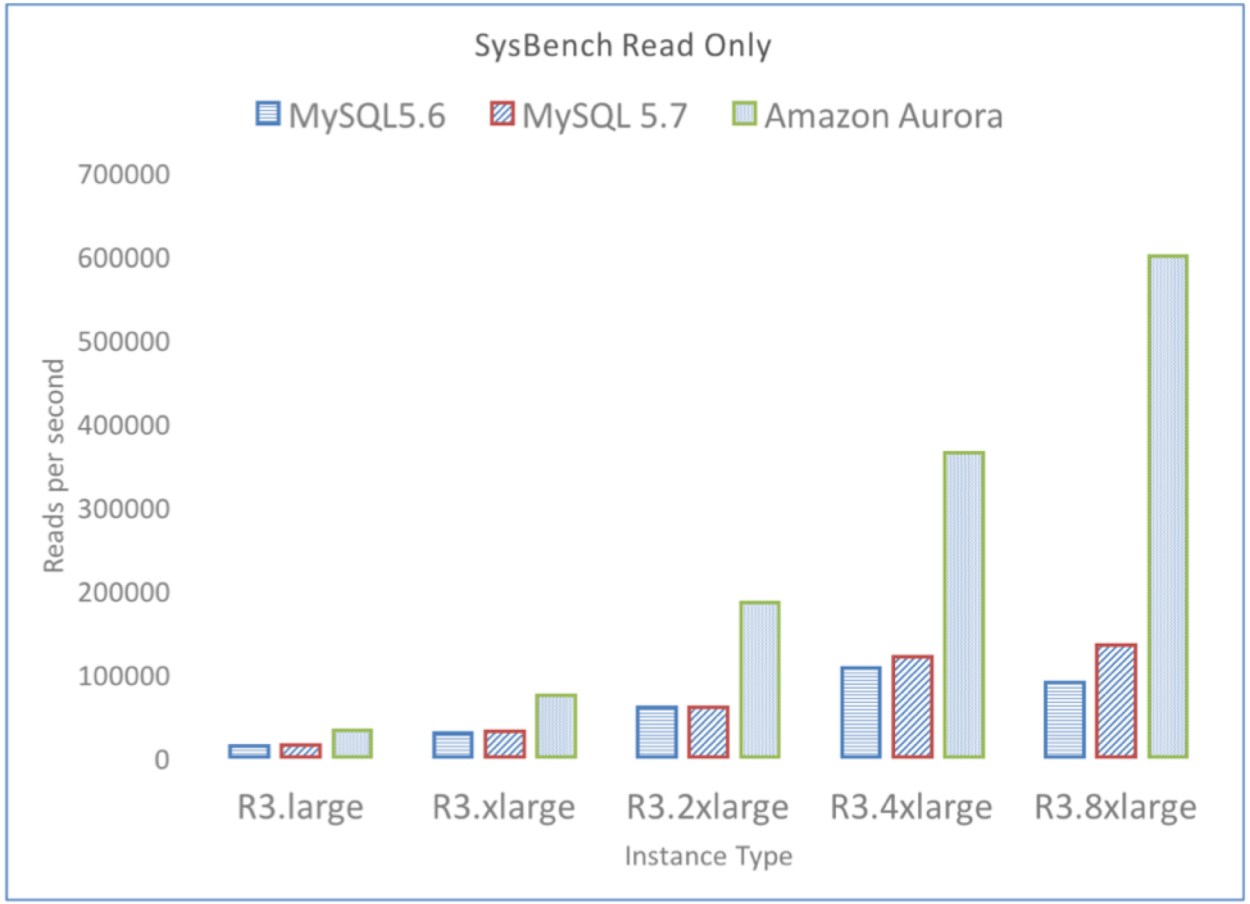


If page is in replica's local buffer, update the page

Otherwise, discard the log record

Evaluation – Aurora vs. MySQL





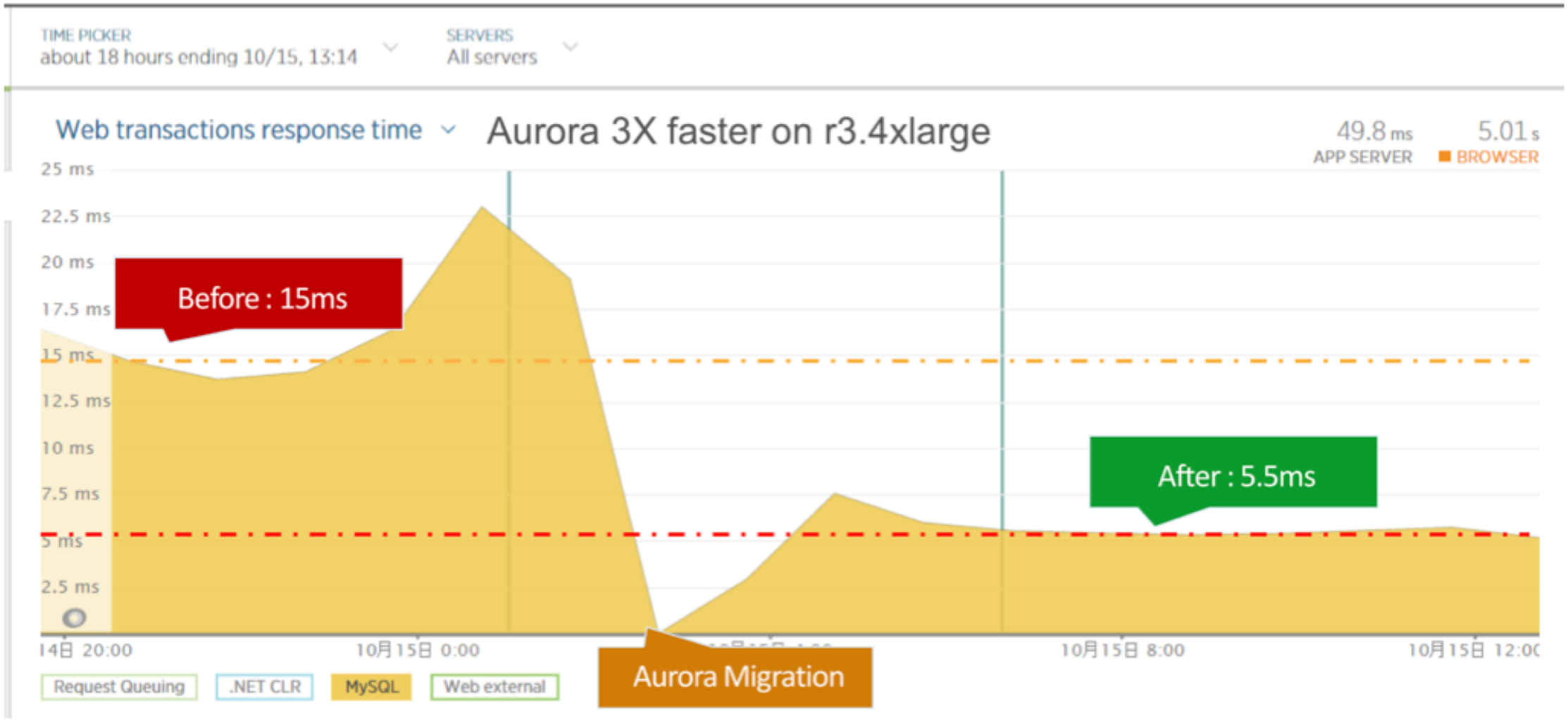
Evaluation – Varying Data Sizes

Table 2: SysBench Write-Only (writes/sec)

DB Size	Amazon Aurora	MySQL
1 GB	107,000	8,400
10 GB	107,000	2,400
100 GB	101,000	1,500
1 TB	41,000	1,200

Performance drops when data does not fit in main memory

Evaluation – Real Customer Workloads



Evaluation – Real Customer Workloads

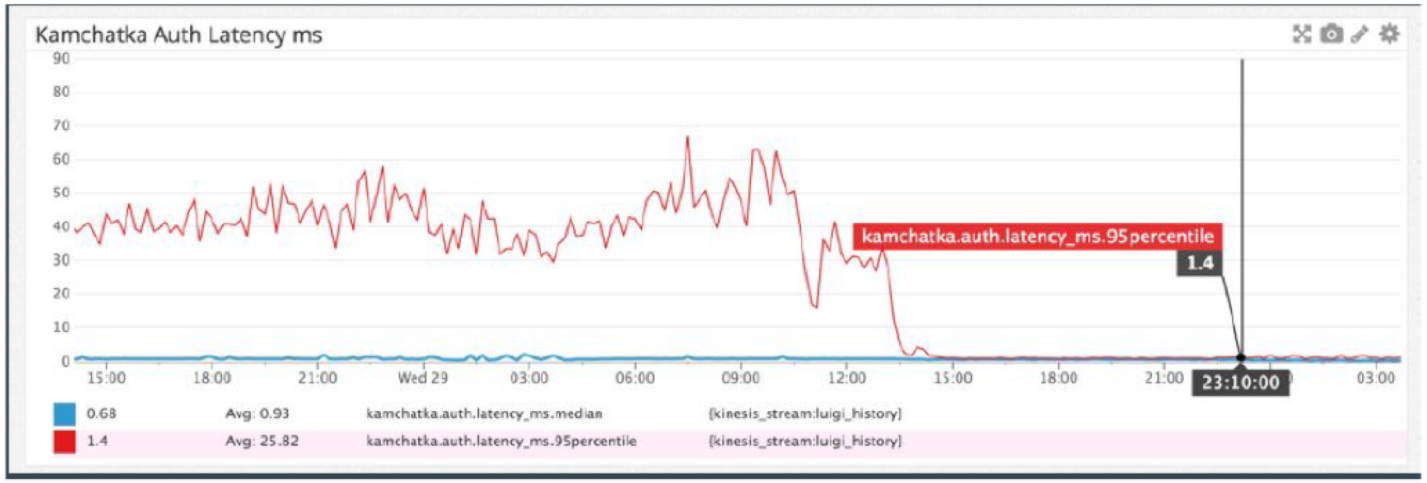
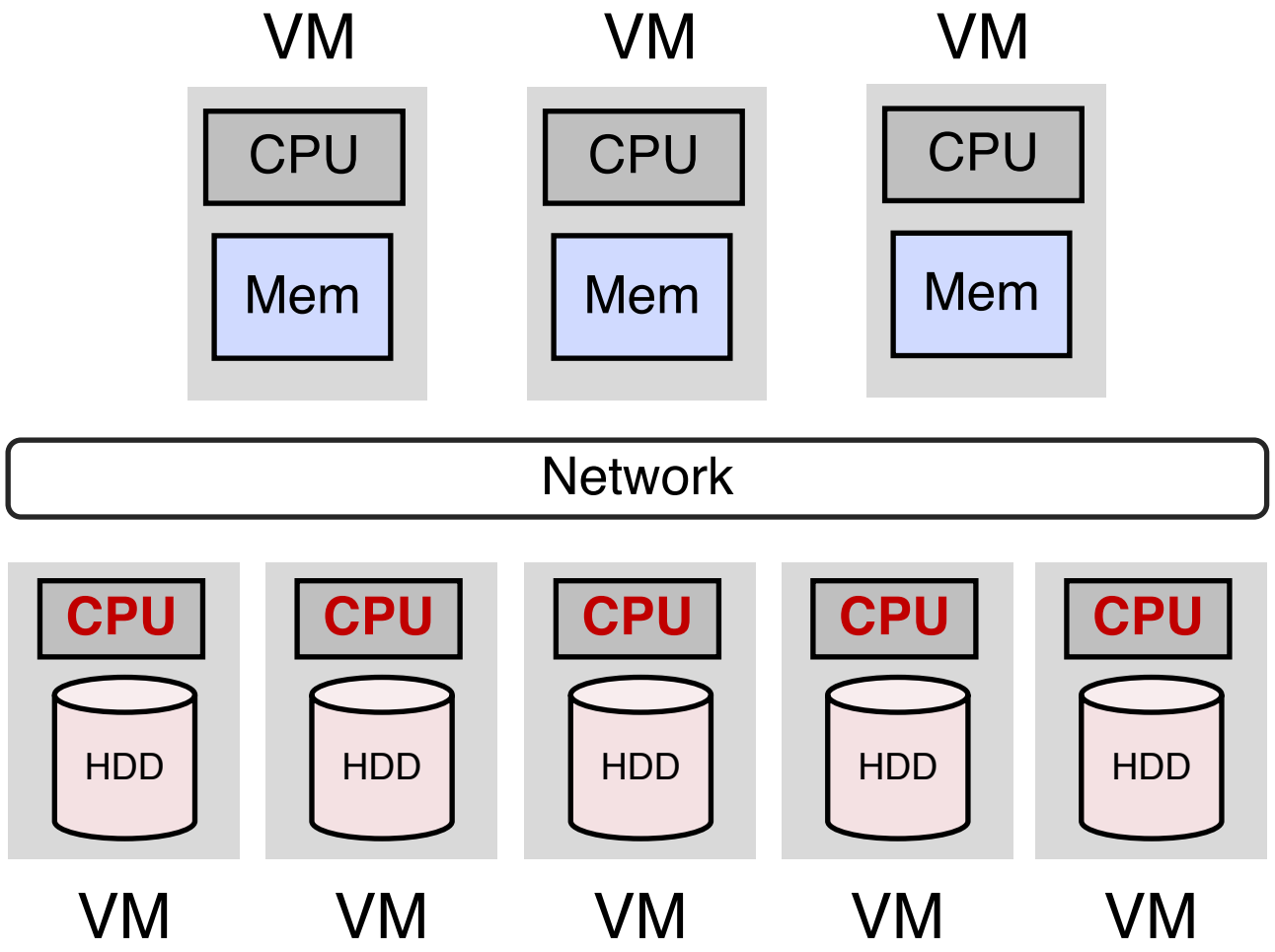


Figure 9: SELECT latency (P50 vs P95)

Figure 10: INSERT per-record latency (P50 vs P95)

Discussion



Quantify the cost of computation in the storage layer

Which fraction of speedup comes from improving MySQL vs. the new disaggregation design

OLTP in Cloud – Q/A

How the log can be considered as database?

Mini-transactions (MTR)?

Transaction not durable when client receives ack?

Other bigger companies have similar offerings?

Adoption of *log as a database*?

Global LSN bottleneck

Similar to logging shipping

Aurora for OLAP?

Serverless and multi-master?

Group Discussion

Cloud storage and Smart SSD are similar in that both push computation to the data. What do you see as the key differences between the two?

The initial version of Aurora (i.e., the one presented in this paper) supports only a single master. What are the challenges of moving to a multi-master setting?

Can you think of other applications that can benefit from a smart and disaggregated storage service in the cloud?

Before Next Lecture

Submit discussion summary to https://wisc-cs839-ngdb20.hotcrp.com

- **Deadline: Friday 11:59pm**

Submit review for

- Choosing A Cloud DBMS: Architectures and Tradeoffs
- [optional] Amazon Redshift and the Case for Simpler Data Warehouses