

**facebook**

# MySQL + RocksDB

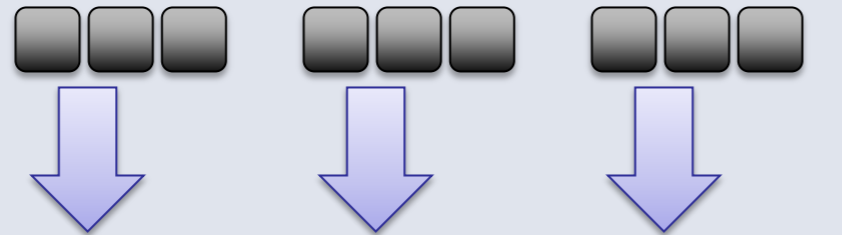
## For Better Storage Efficiency Than InnoDB

XLDB 2015 Lightning Talk  
Siying Dong, Software Engineer  
Database Engineering Team @ Facebook

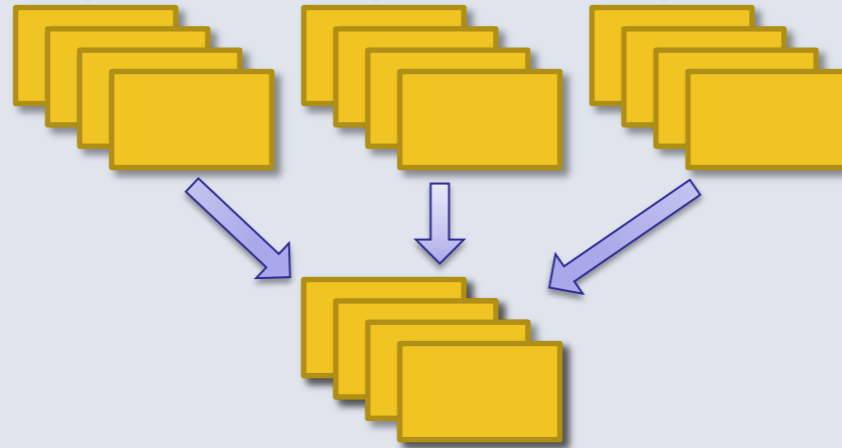
# Facebook Website Architecture

Data center

Web servers



Caches



MySQL  
Databases



# Facebook Website Architecture

Data center

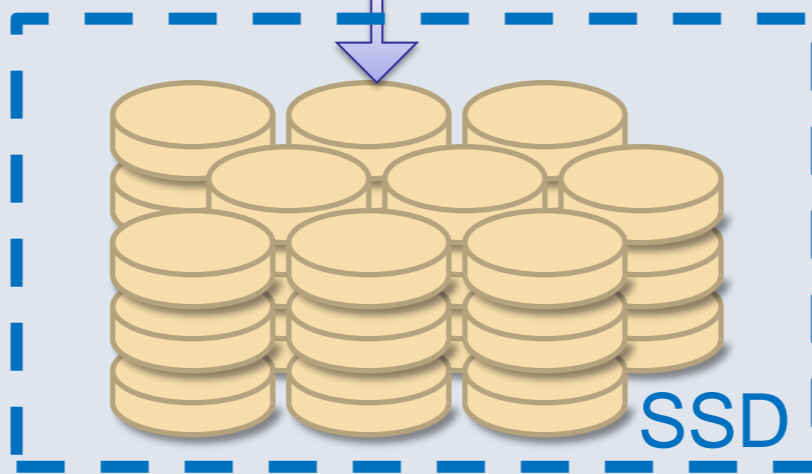
Web servers



Caches



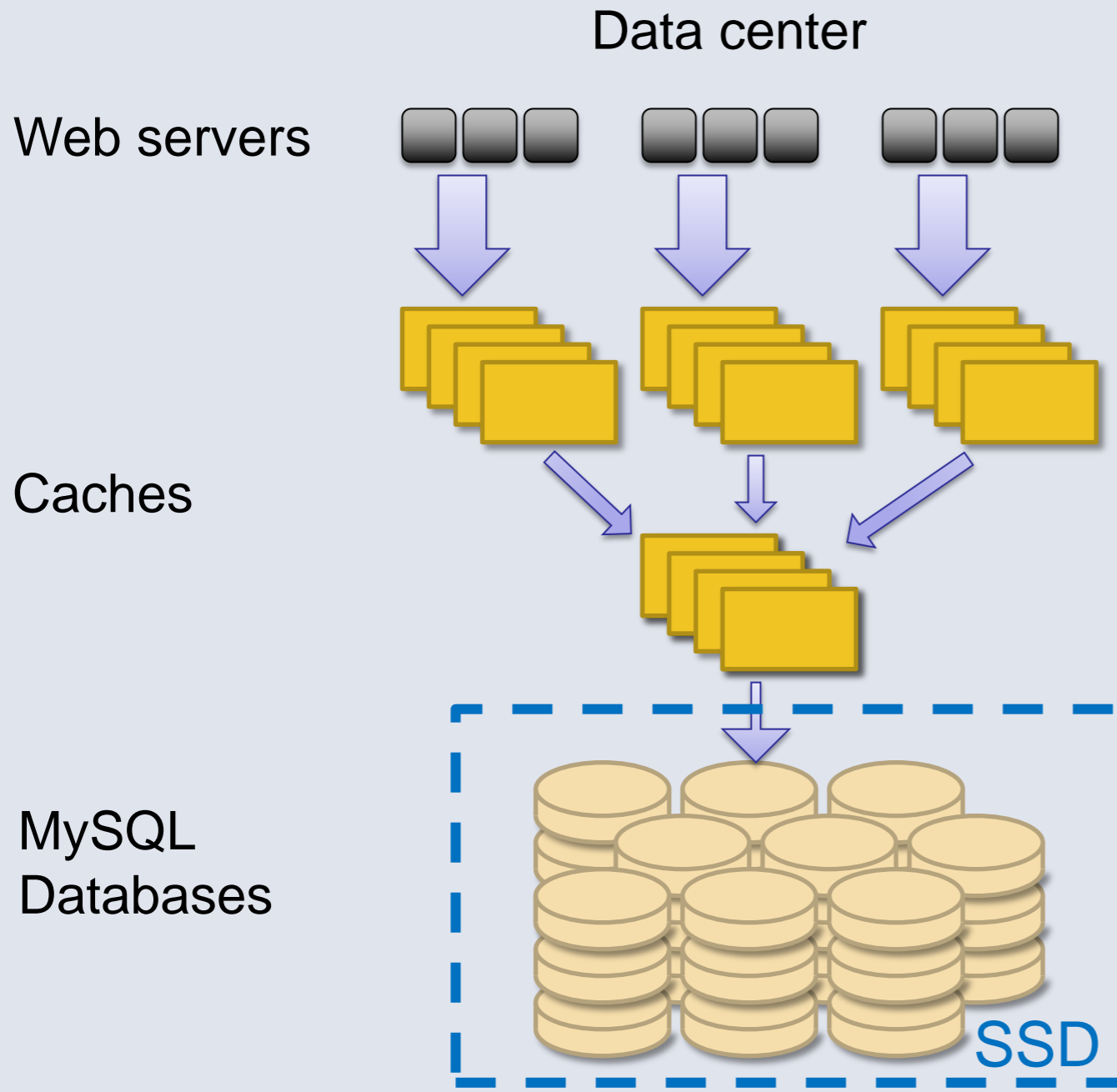
MySQL  
Databases



SSD

S

# Facebook Website Architecture



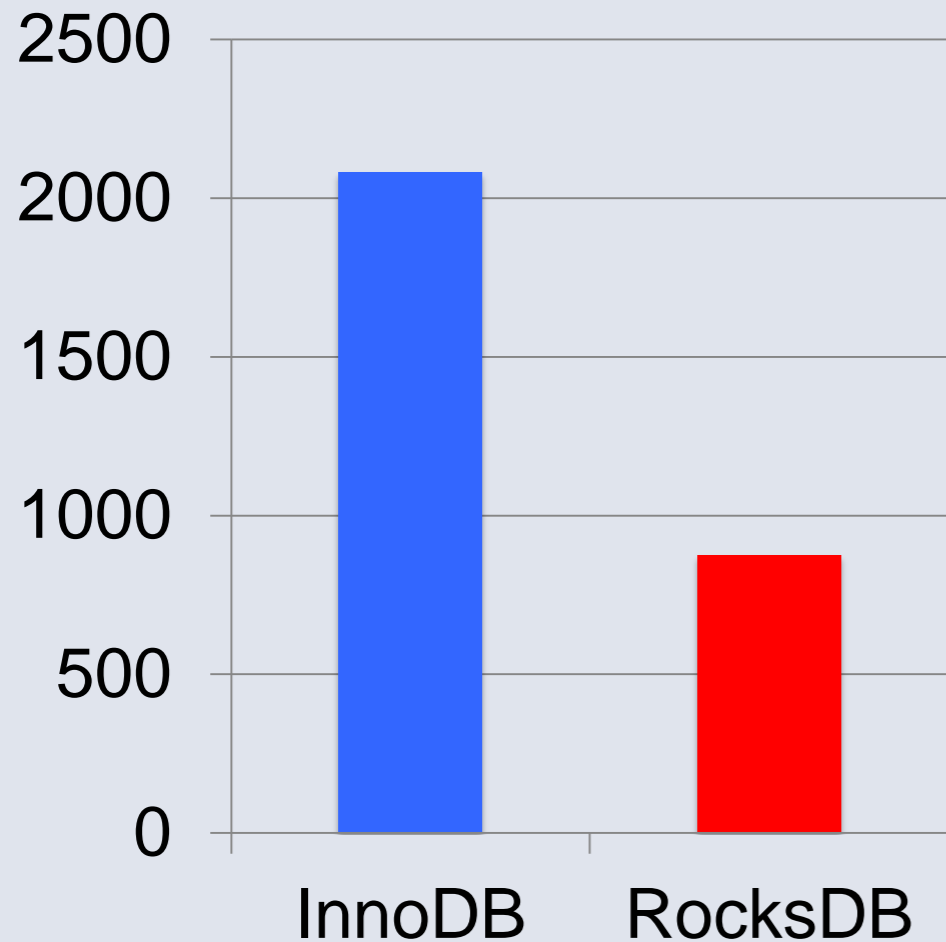
- SSD Capacity is the bottleneck
- Also want to write fewer bytes to SSDs

# MySQL + RocksDB (MyRocks)

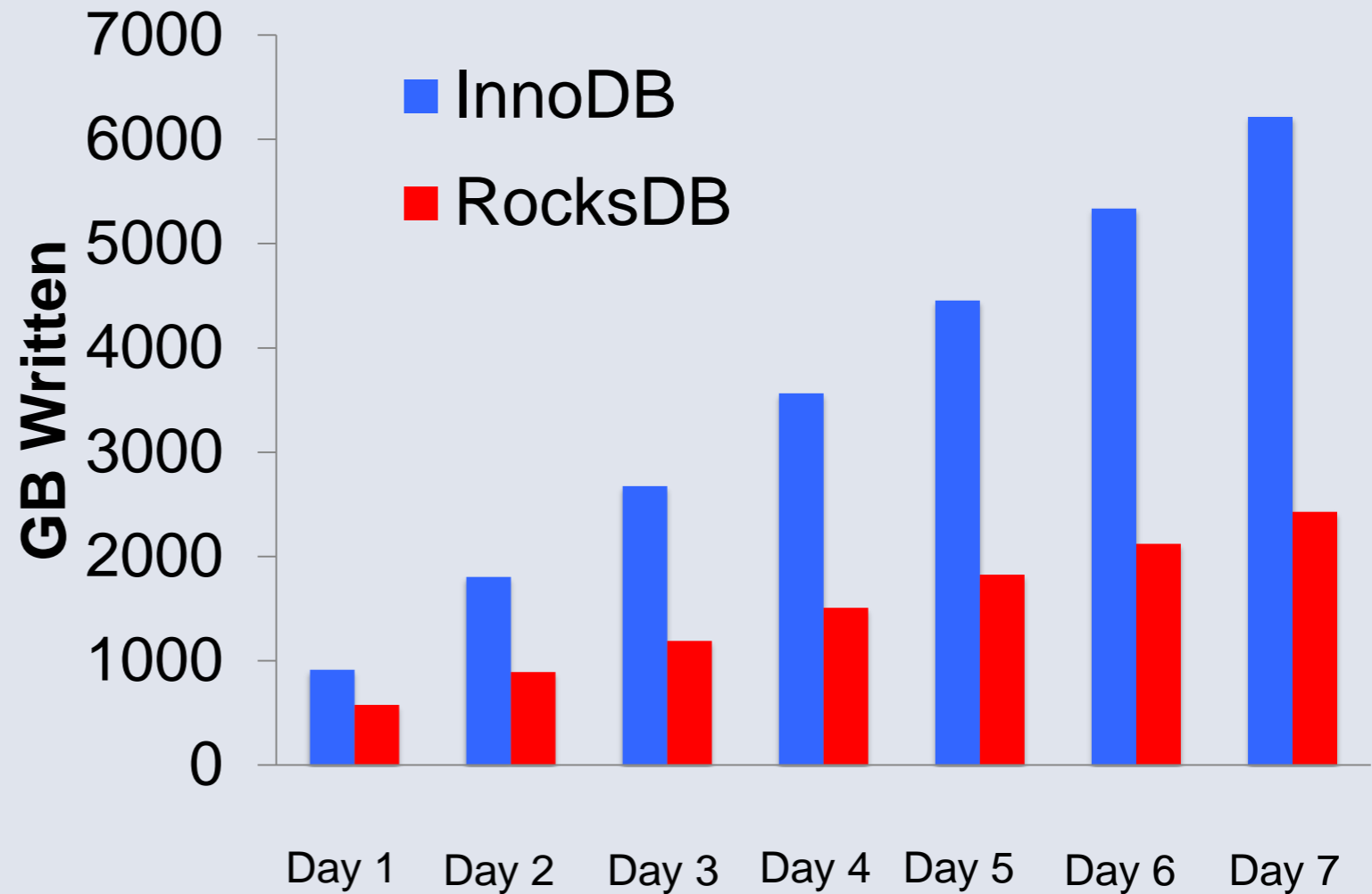
- RocksDB
  - Embedded persistent key-value store
  - Log-Structure-Merge tree
  - Open-Sourced
- RocksDB Storage Engine in MySQL  
<https://github.com/MySQLOnRocksDB/mysql-5.6>

# InnoDB vs. RocksDB Storage Engines

## DB Size (GB)



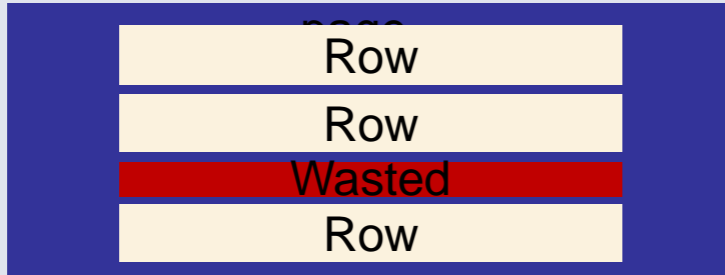
## Cumulative GB Written



- Workload: Social Graph Data (Edges + Vertexes)
- RocksDB: zlib compression from L3
- InnoDB: edges use zlib, vertexes are pre-compressed.

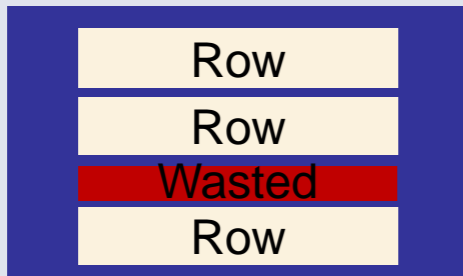
# Lower Space Amplification

Uncompressed 16KB

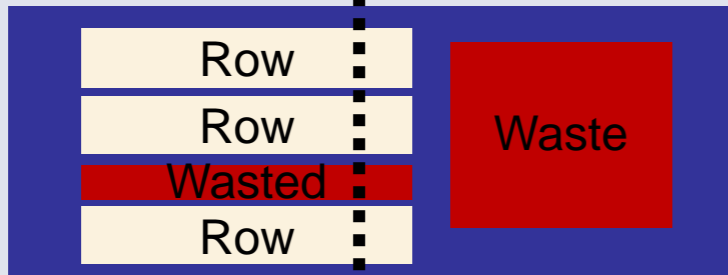


InnoDB

Compressed to 5KB



Using 8KB space on storage



4KB

4KB

Level 0

Level 1

Level 2

Level 3

Level 4

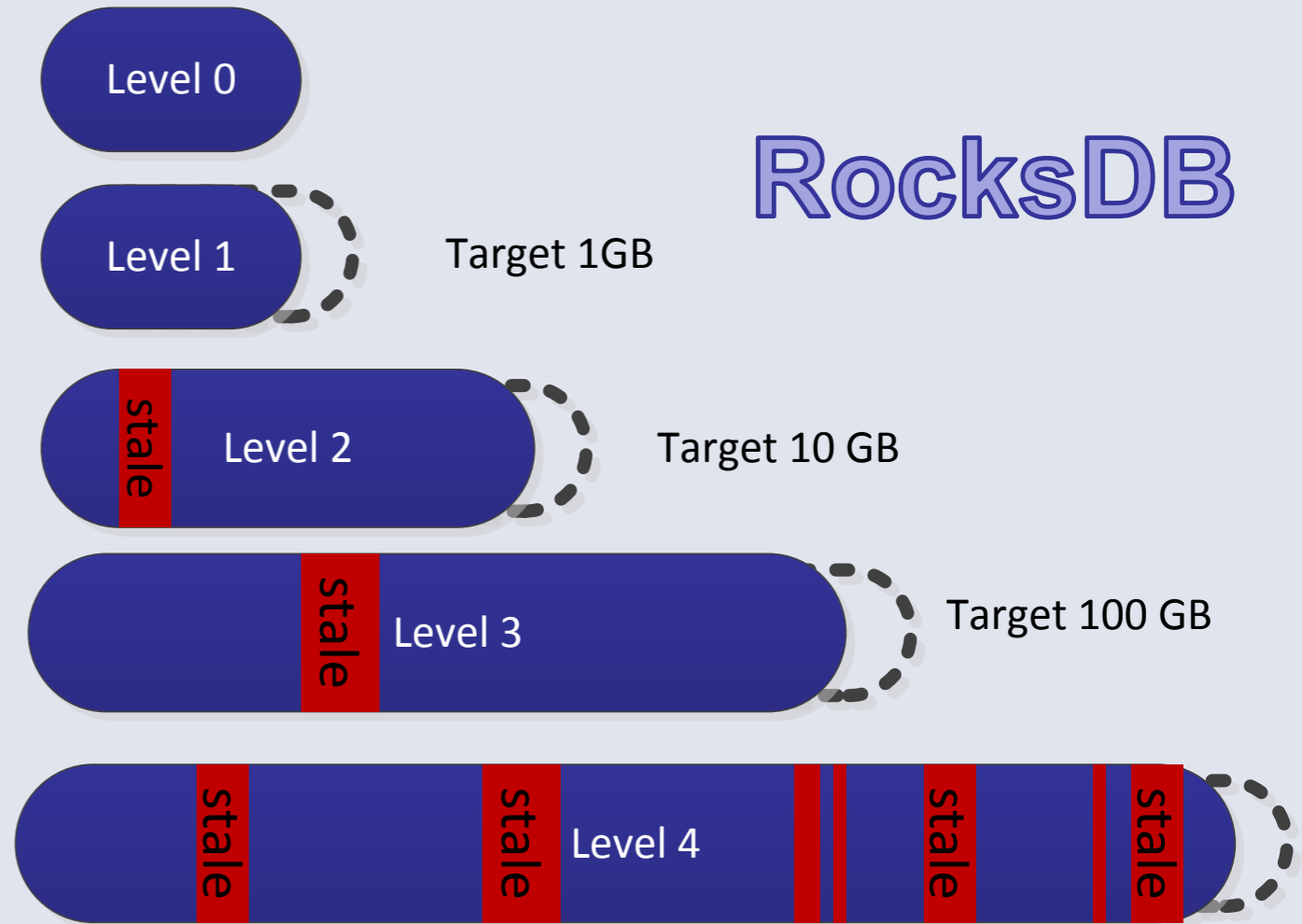
Target 1GB

Target 10 GB

Target 100 GB

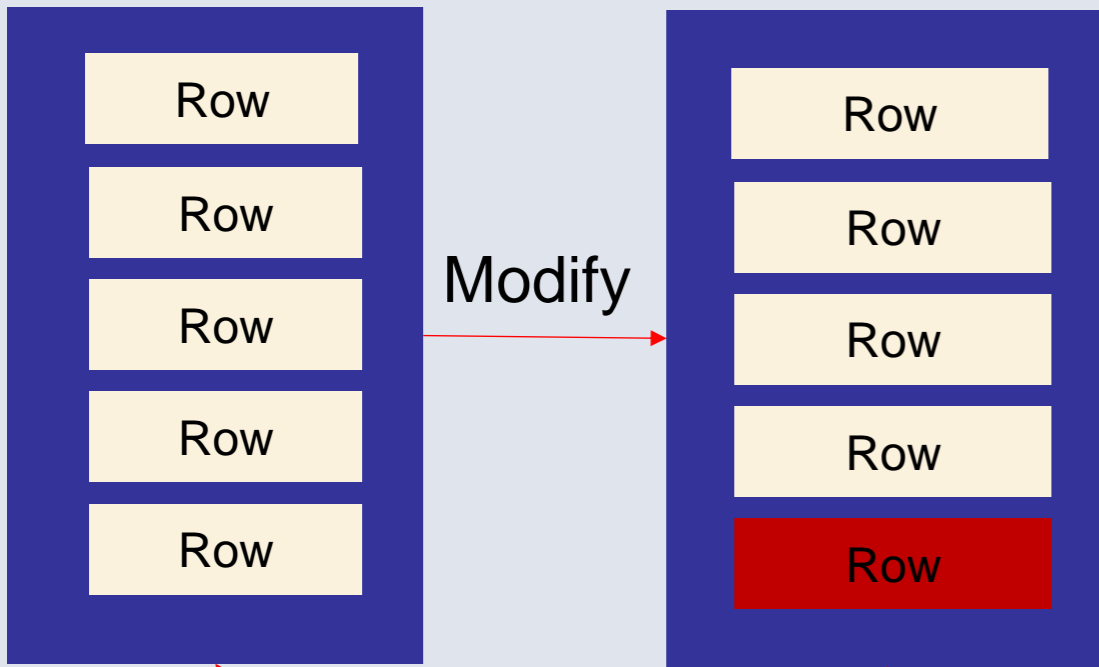
Target 1000 GB

RocksDB



# Lower Write Amplification (Worst Case)

## InnoDB

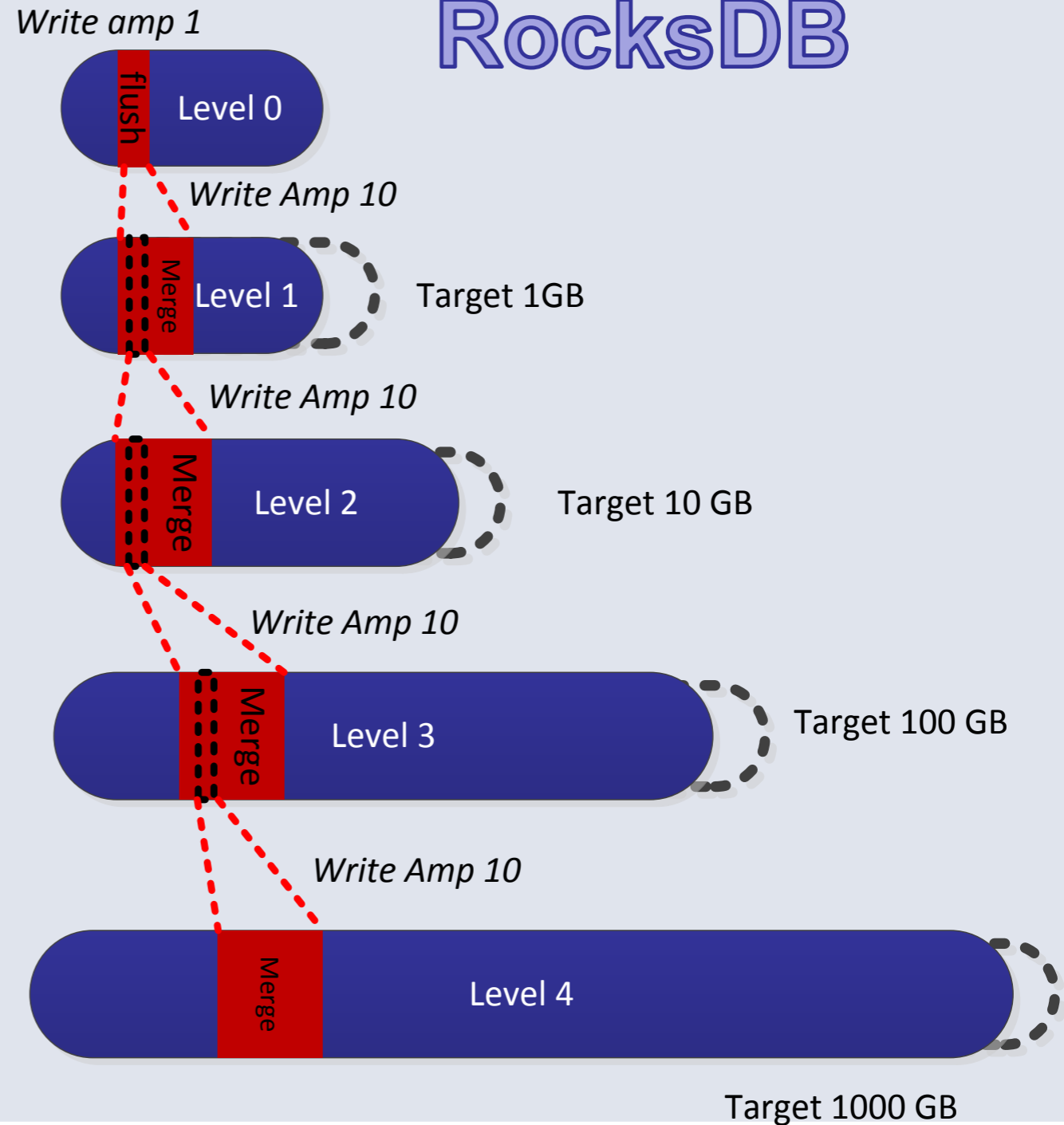


Read

Write

Write Amp = Page size / row size

## RocksDB



Write amp 1

Level 0

Write Amp 10

Level 1

Target 1GB

Write Amp 10

Level 2

Target 10 GB

Write Amp 10

Level 3

Target 100 GB

Write Amp 10

Level 4

Target 1000 GB



# Conclusion

- Current bottlenecks of our MySQL databases:
  - disk capacity
  - bytes written
- MySQL + RocksDB vs. InnoDB
  - compresses better
  - writes less data to storage devices
- We anticipate RocksDB will be able to replace many uses of InnoDB after some more work

# facebook

(c) 2009 Facebook, Inc. or its licensors. "Facebook" is a registered trademark of Facebook, Inc.. All rights reserved. 1.0