FAST'23
FEBRUARY 21-23, 2023
SANTA CLARA, CA

CJFS : Concurrent Journaling for Better Scalability

Joontaek Oh*, Seung Won Yoo*, Hojin Nam*, Changwoo Min†, Youjip Won*

*KAIST †Virginea Tech

Outline

- ✓ Background and Motivation
- ✓ Design
  - ➢ Dual Thread Journaling
  - ➢ Multi-Version Shadow Paging
  - ➢ Opportunistic Coalescing
  - ➢ Compound Flush
- ✓ Evaluation
- ✓ Conclusion

Background and Motivation

Hardware and Software @2023+

Hardware:

2 cores
Intel Core 2 Duo
@2006

877 IOPS
Western Digital Caviar SE16
@2006

Software:

Hardware and Software @2023+

Hardware:

2 cores
Intel Core 2 Duo
@2006

128 cores
AMD EPYC 7763
@2021

877 IOPS
Western Digital Caviar SE16
@2006

700K IOPS
Seagate FireCuda 530
@2021

Software:

Hardware and Software @2023+

Hardware:

2 cores
Intel Core 2 Duo
@2006

64X

128 cores
AMD EPYC 7763
@2021

877 IOPS
Western Digital Caviar SE16
@2006

798X

700K IOPS
Seagate FireCuda 530
@2021

Software:

Hardware and Software @2023+

Hardware:

64X

2 cores
Intel Core 2 Duo
@2006

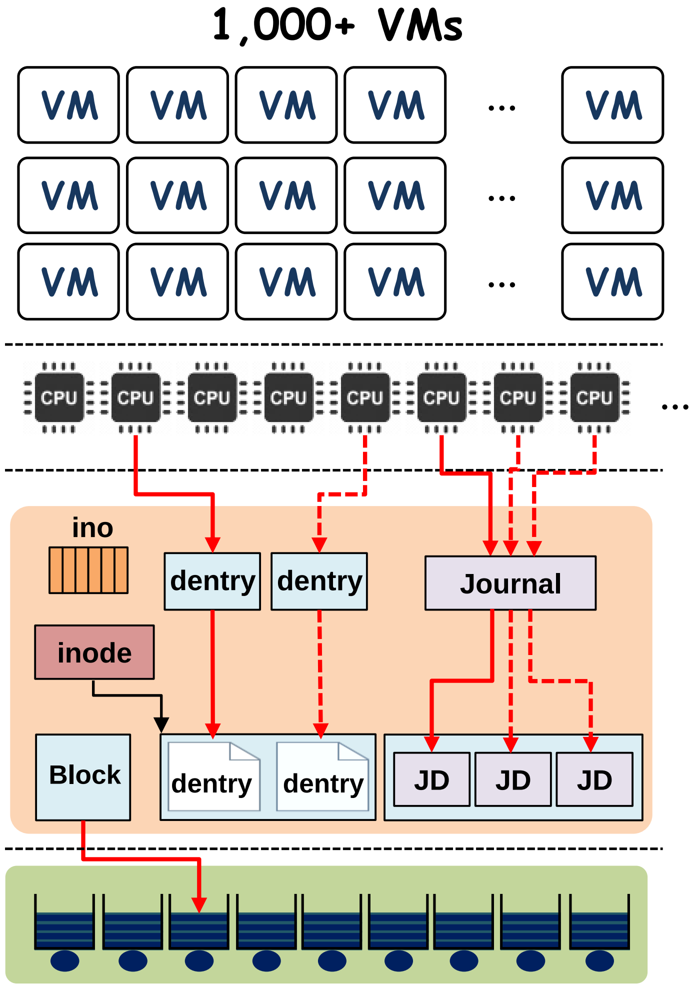
128 cores
AMD EPYC 7763
@2021

798X

877 IOPS
Western Digital Caviar SE16
@2006

700K IOPS
Seagate FireCuda 530
@2021

Software:



Hardware and Software @2023+

Hardware:

64X

2 cores
Intel Core 2 Duo
@2006

128 cores
AMD EPYC 7763
@2021

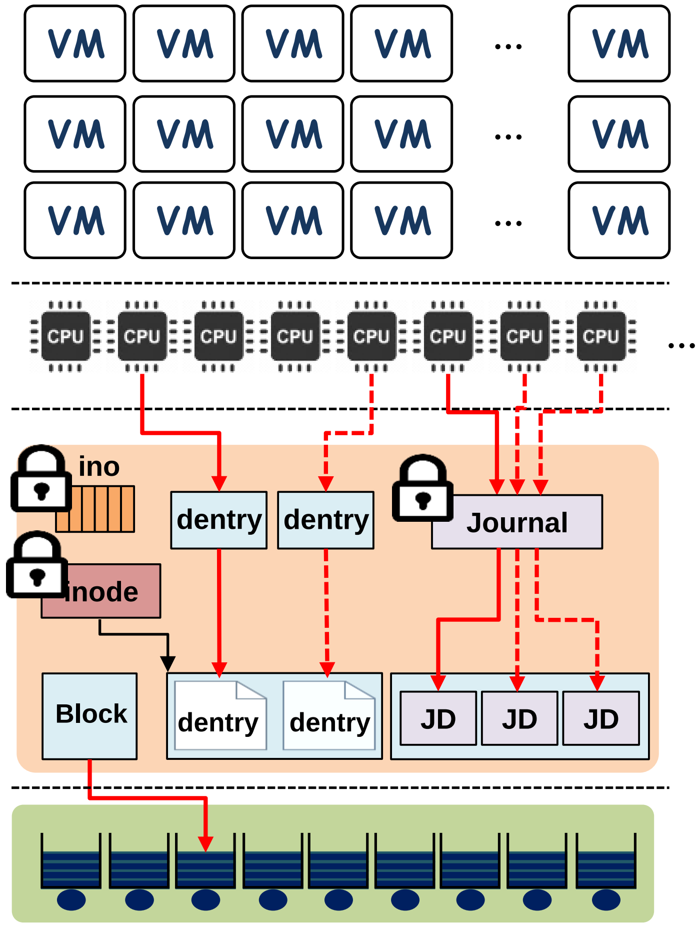
798X

877 IOPS
Western Digital Caviar SE16
@2006

700K IOPS
Seagate FireCuda 530
@2021

Software:

1,000+ VMs



Hardware and Software @2023+

Hardware:

64X

2 cores
Intel Core 2 Duo
@2006

128 cores
AMD EPYC 7763
@2021

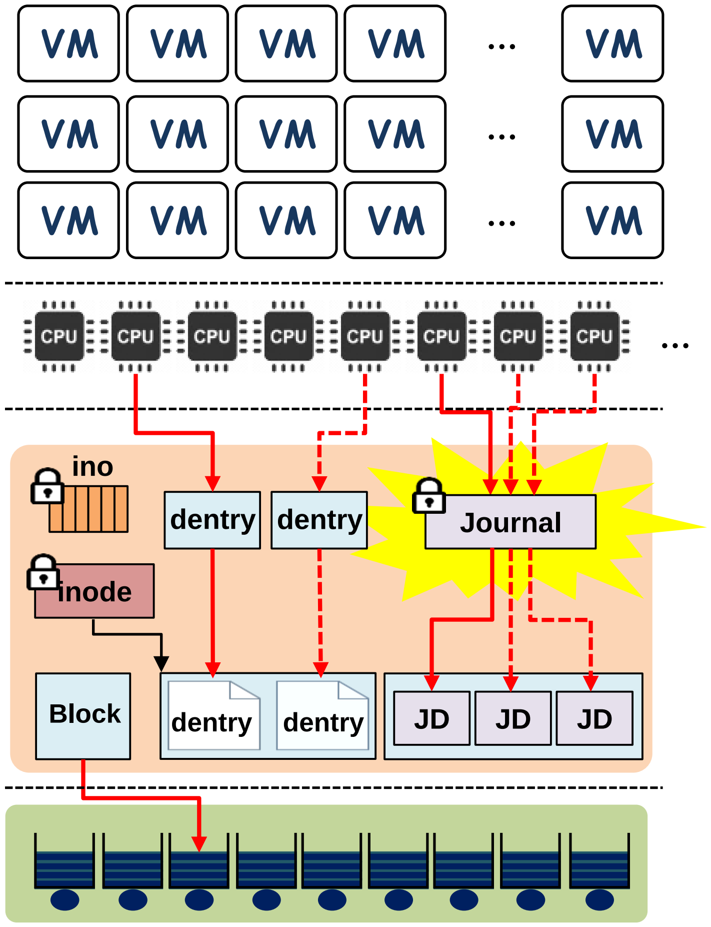
798X

877 IOPS
Western Digital Caviar SE16
@2006

700K IOPS
Seagate FireCuda 530
@2021

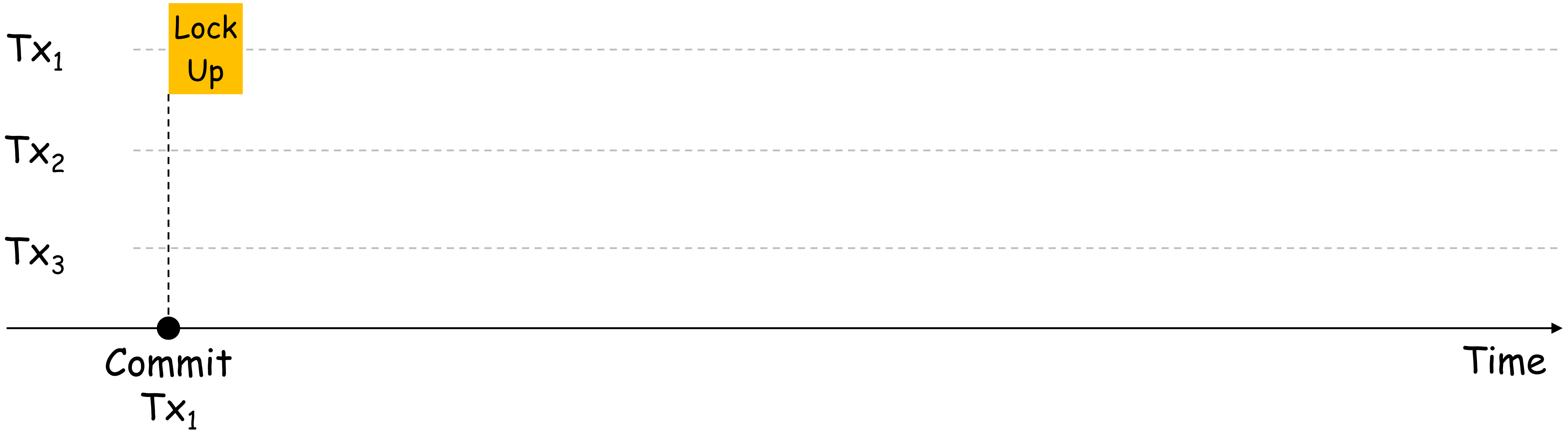
Software:

1,000+ VMs



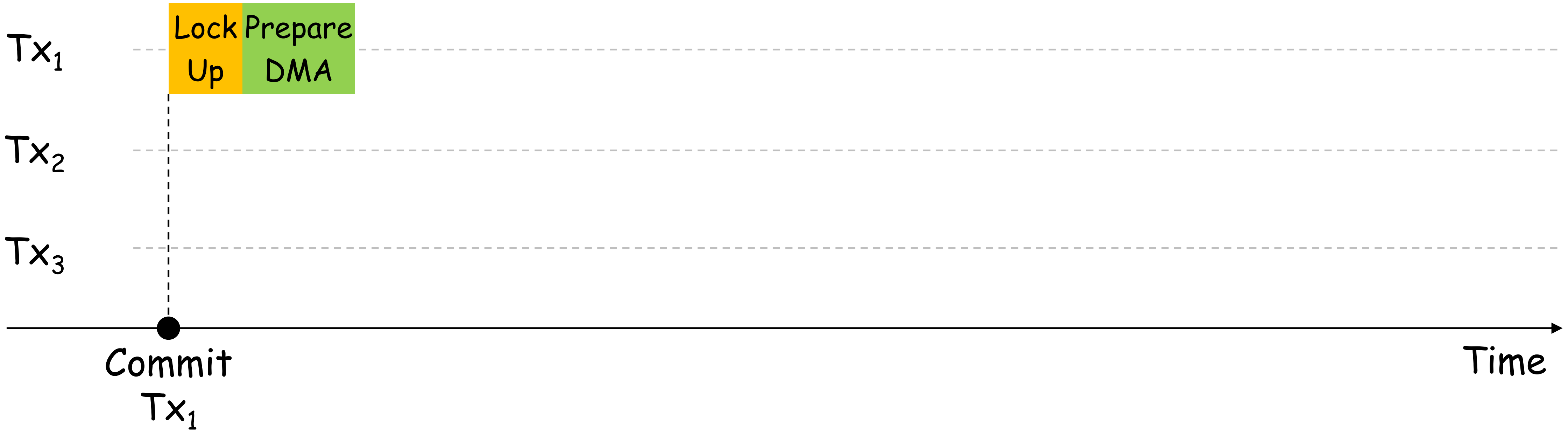
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
    - Lock-Up: Lock the running transaction and waiting for remained file operation



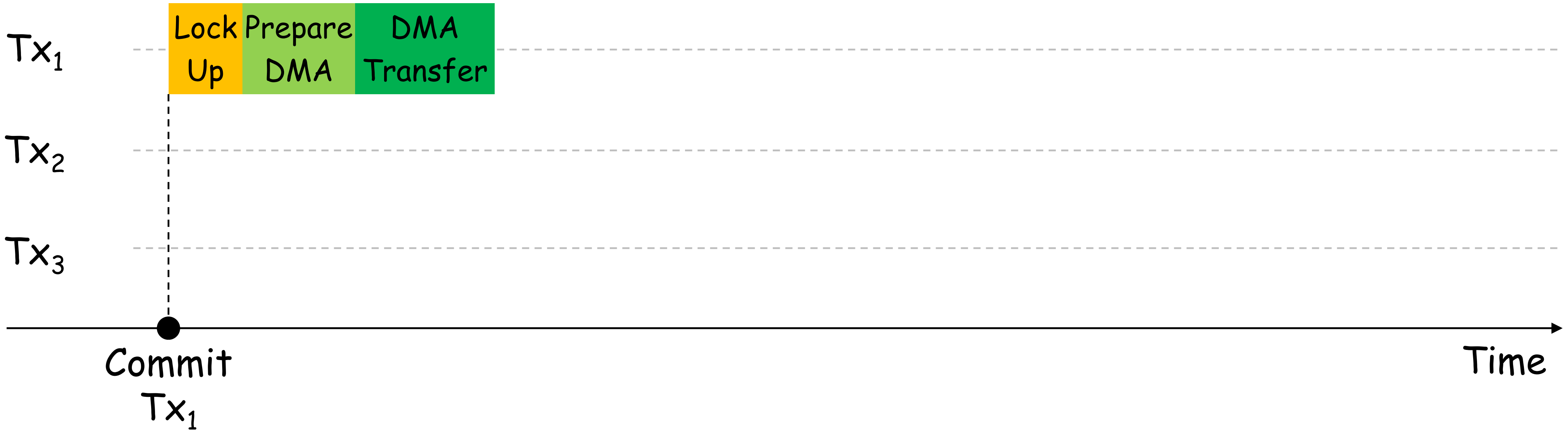
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction



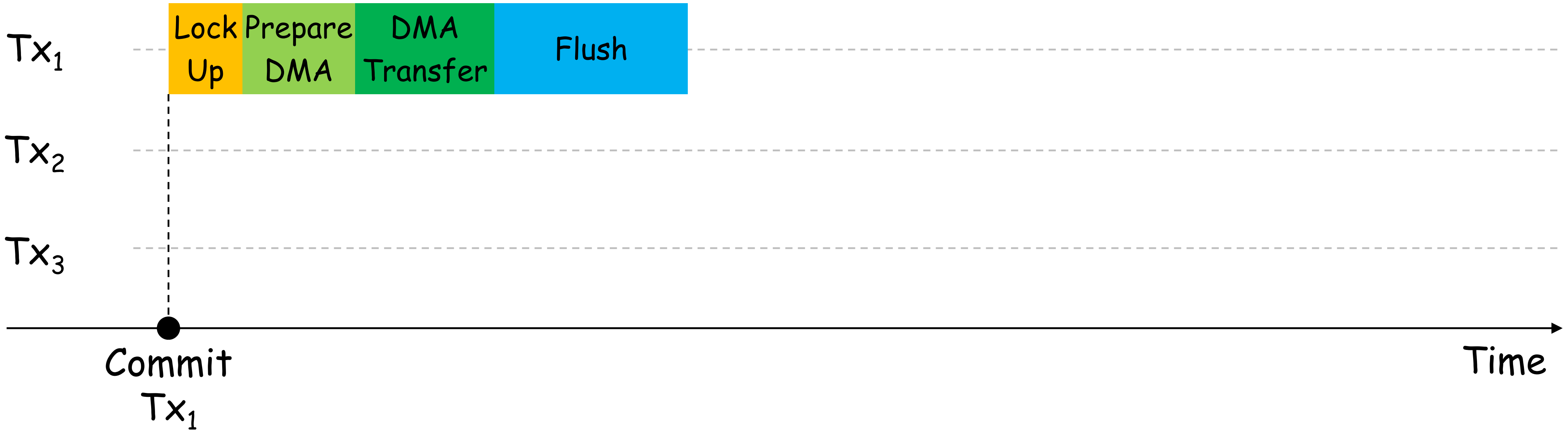
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction
  - DMA Transfer: Waiting for the completion of DMA Transfer of the transaction



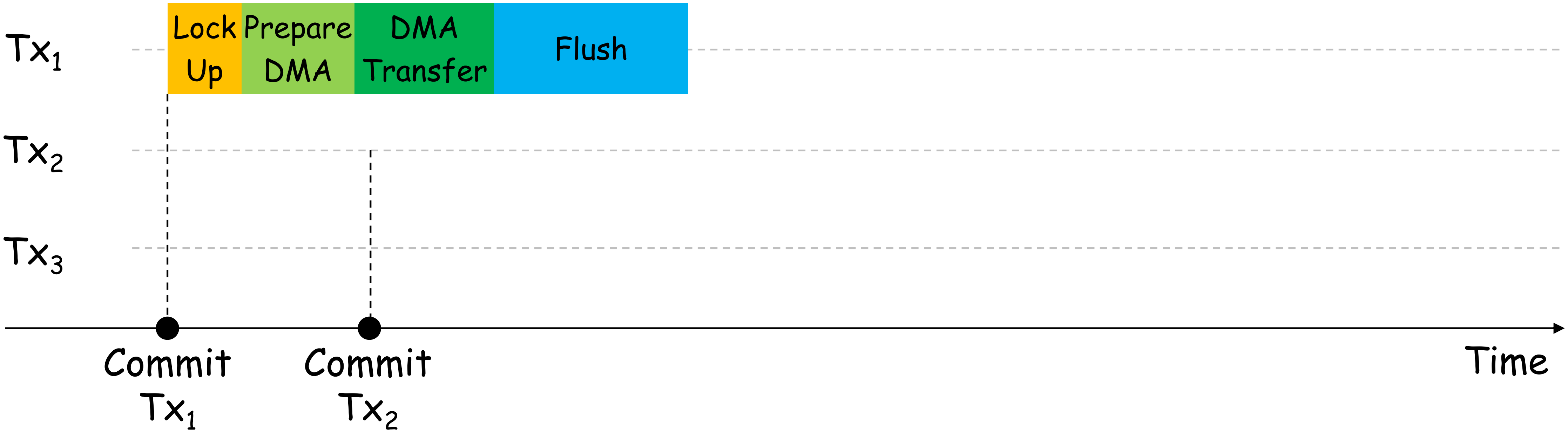
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction
  - DMA Transfer: Waiting for the completion of DMA Transfer of the transaction
  - Flush: Flush transferred data



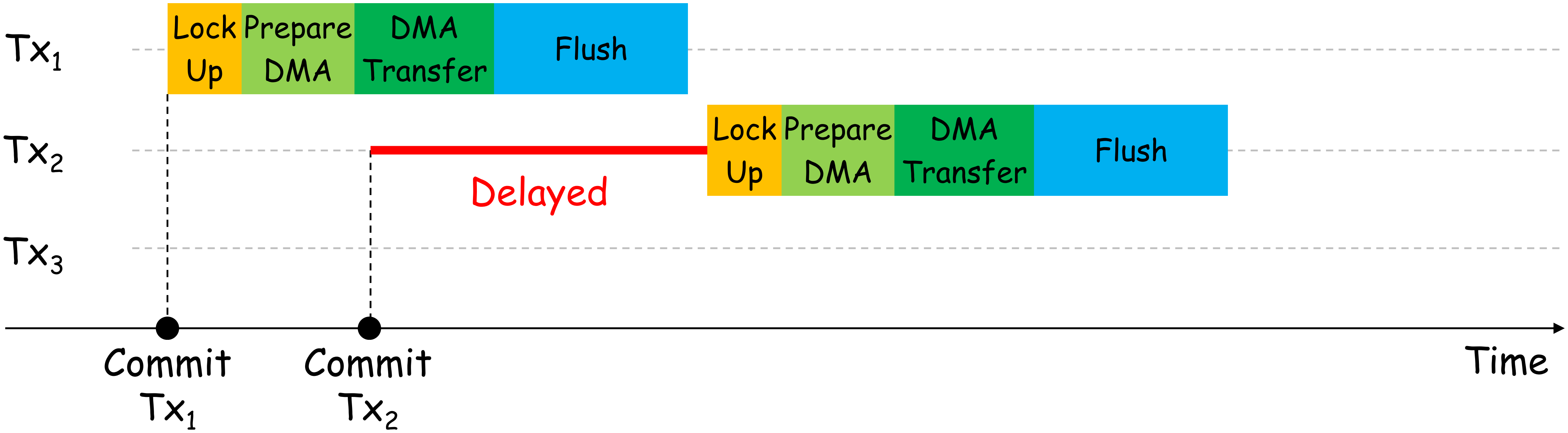
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction
  - DMA Transfer: Waiting for the completion of DMA Transfer of the transaction
  - Flush: Flush transferred data



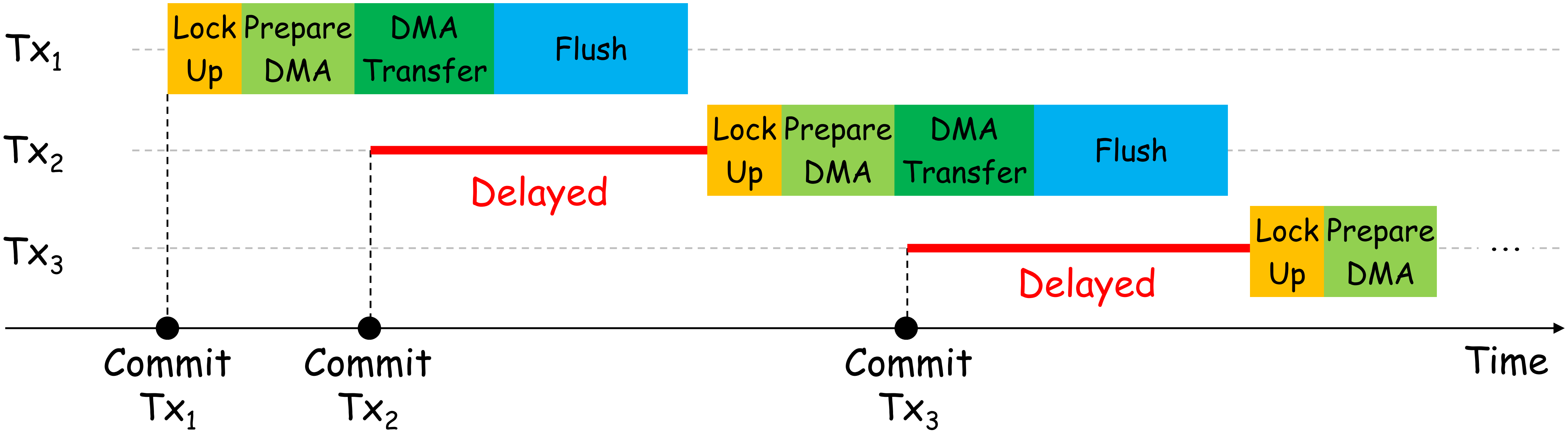
Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction
  - DMA Transfer: Waiting for the completion of DMA Transfer of the transaction
  - Flush: Flush transferred data



Serial Commit in EXT4 Journaling

- All steps of journal commit are serialized
  - Lock-Up: Lock the running transaction and waiting for remained file operation
  - Prepare DMA: Create and dispatch write command for the transaction
  - DMA Transfer: Waiting for the completion of DMA Transfer of the transaction
  - Flush: Flush transferred data

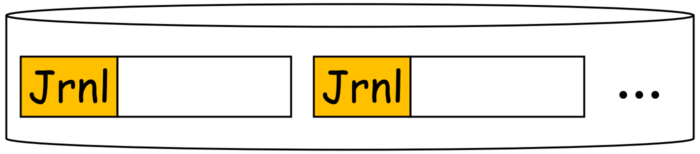


Existing Works

Existing Works

Multiple journal regions:

IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

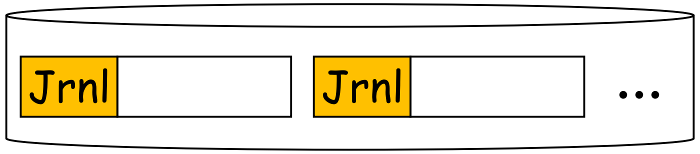


Existing Works

Multiple journal regions:

IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

Still serial transaction commit in each journal region

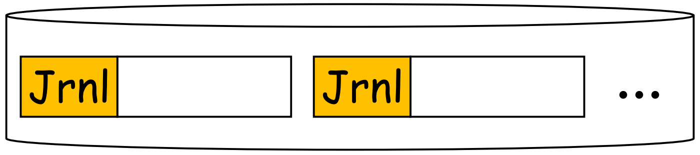


Existing Works

Multiple journal regions:

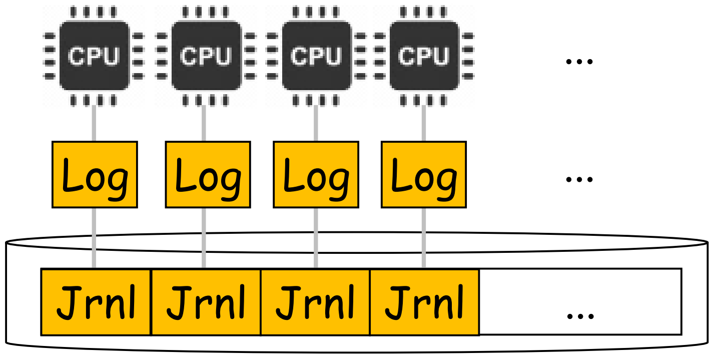
IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

Still serial transaction commit in each journal region



Per-core running transaction:

ScaleFS (SOSP '17), MQFS (SOSP '21)

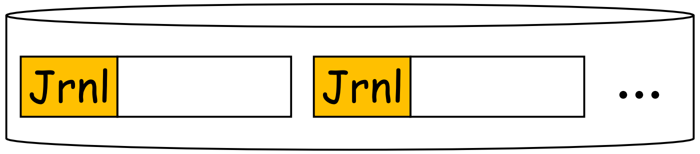


Existing Works

Multiple journal regions:

IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

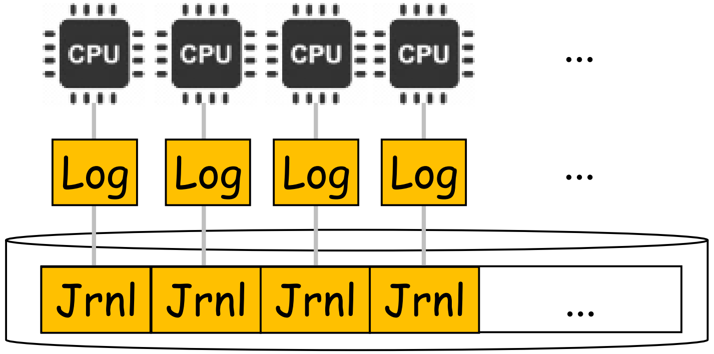
Still serial transaction commit in each journal region



Per-core running transaction:

ScaleFS (SOSP '17), MQFS (SOSP '21)

Conflict between multiple transactions and Still serial commit

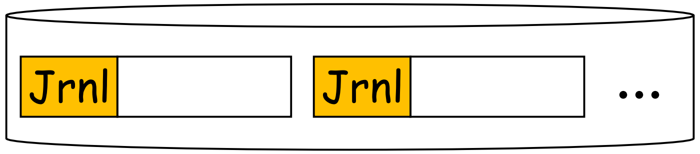


Existing Works

Multiple journal regions:

IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

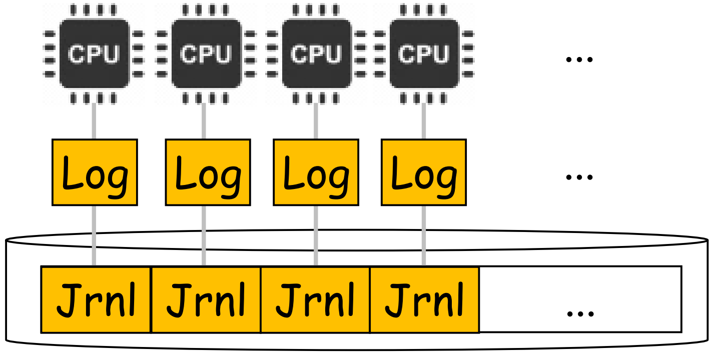
Still serial transaction commit in each journal region



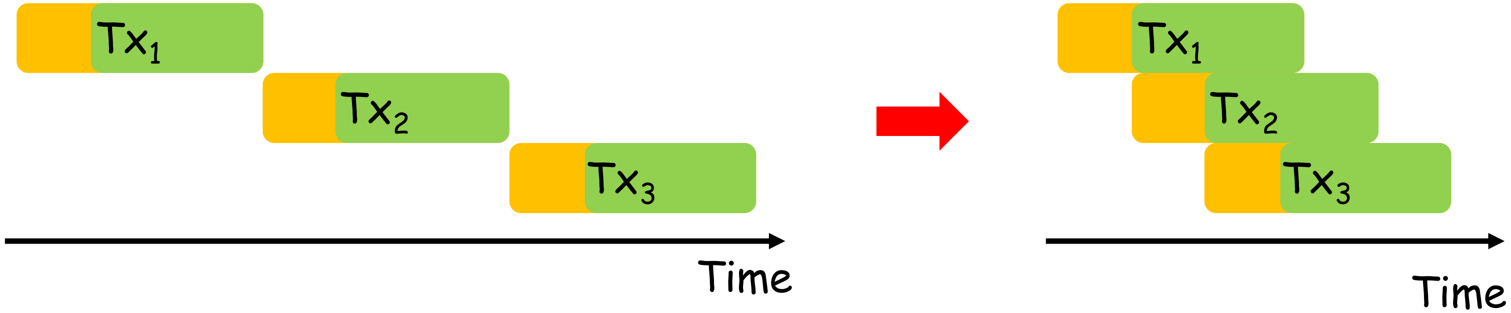
Per-core running transaction:

ScaleFS (SOSP '17), MQFS (SOSP '21)

Conflict between multiple transactions and Still serial commit



Parallel journal commit: BarrierFS (FAST '18)

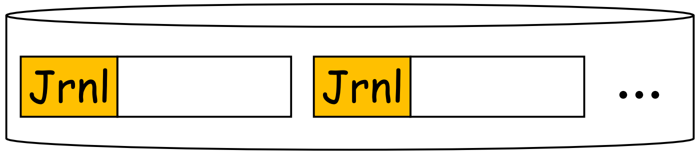


Existing Works

Multiple journal regions:

IceFS (OSDI '14), SpanFS (ATC '15), Z-journal (ATC'21)

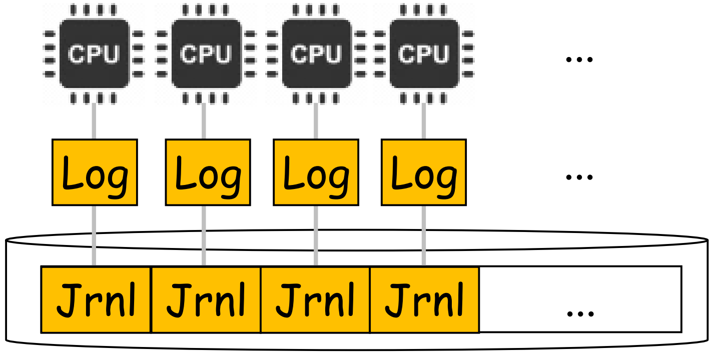
Still serial transaction commit in each journal region



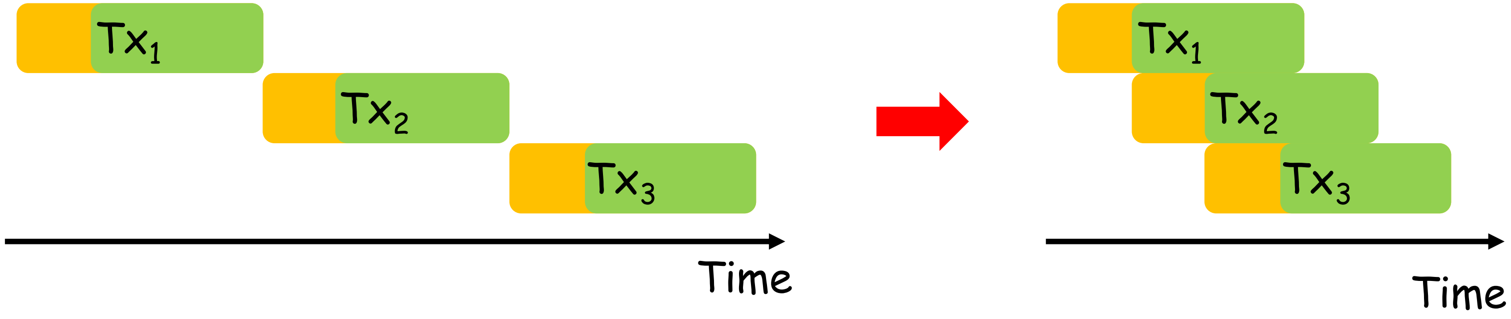
Per-core running transaction:

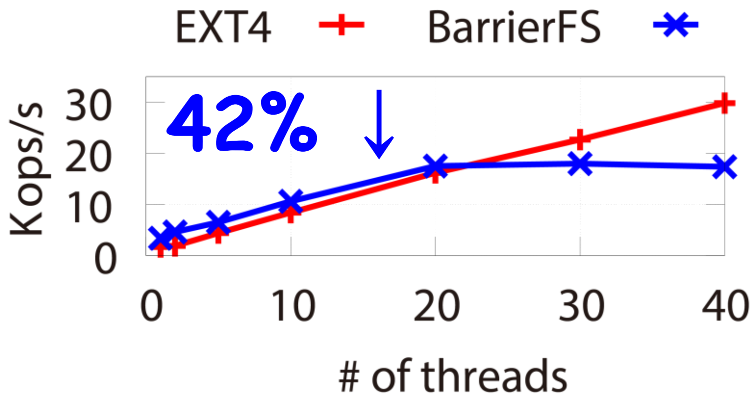
ScaleFS (SOSP '17), MQFS (SOSP '21)

Conflict between multiple transactions and Still serial commit



Parallel journal commit: BarrierFS (FAST '18)





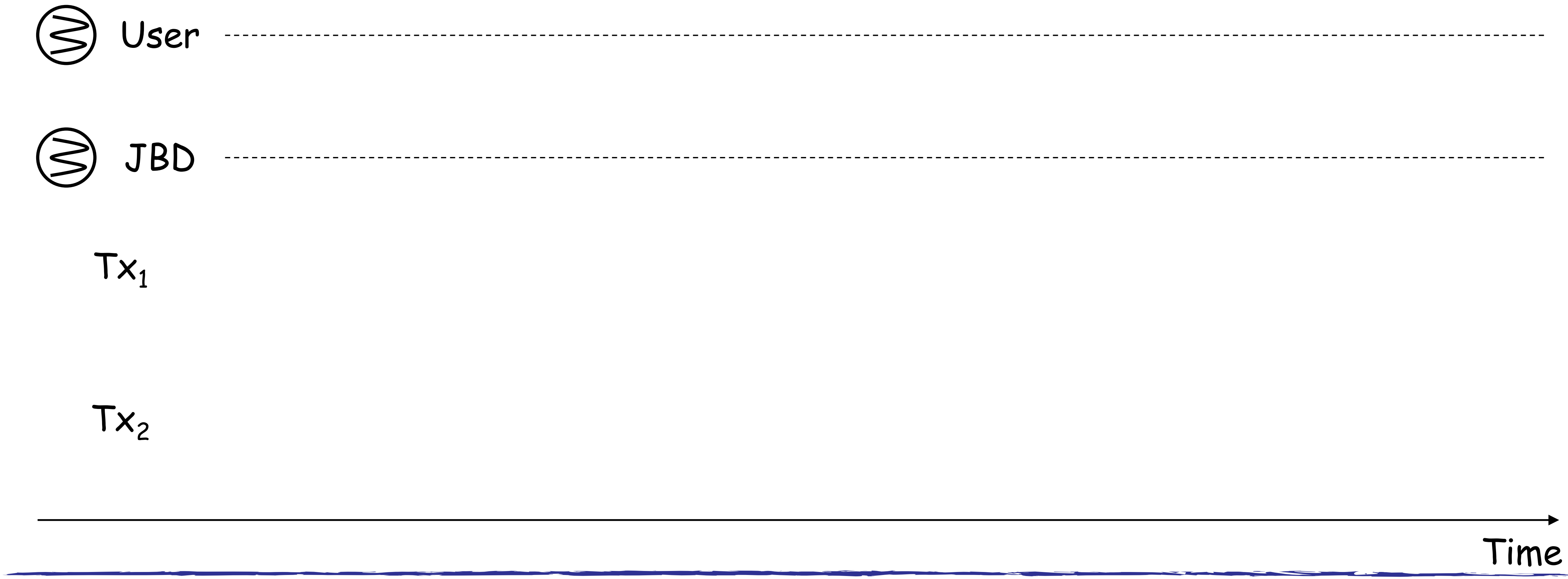
Main reasons

- Transaction conflict

- Transaction Lock-Up

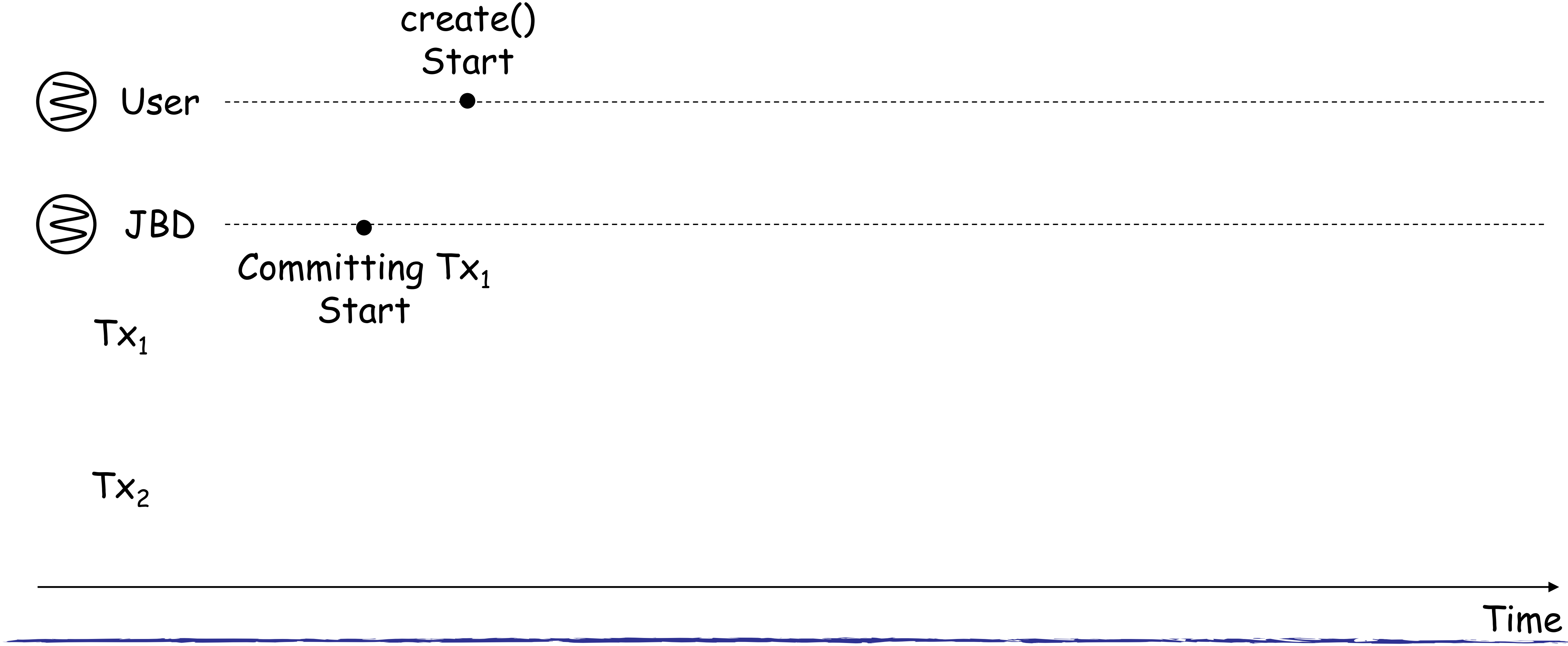
Transaction conflict

The situation that a file operation modifies a page which is being committed



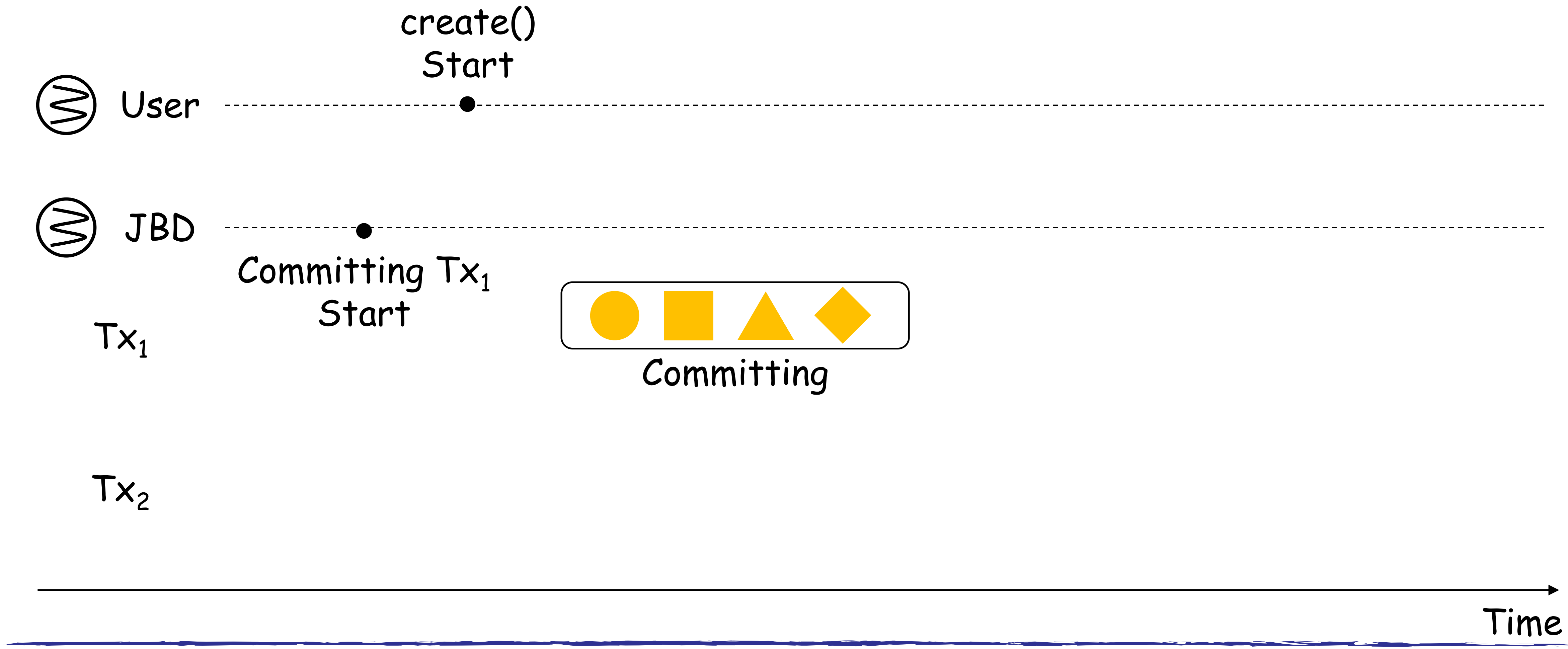
Transaction conflict

The situation that a file operation modifies a page which is being committed



Transaction conflict

The situation that a file operation modifies a page which is being committed



create()
Start

User

JBD

Committing Tx_1
Start

Tx_1

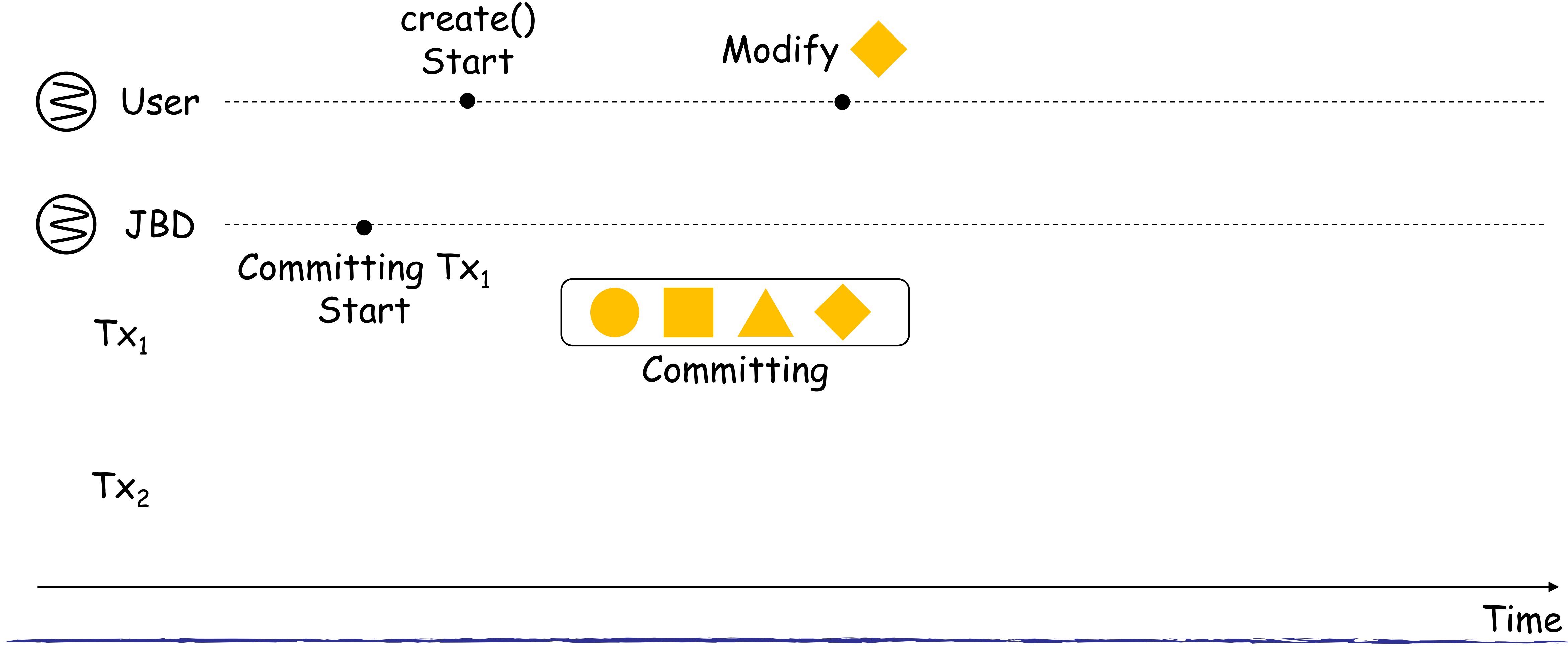
Committing

Tx_2

Time

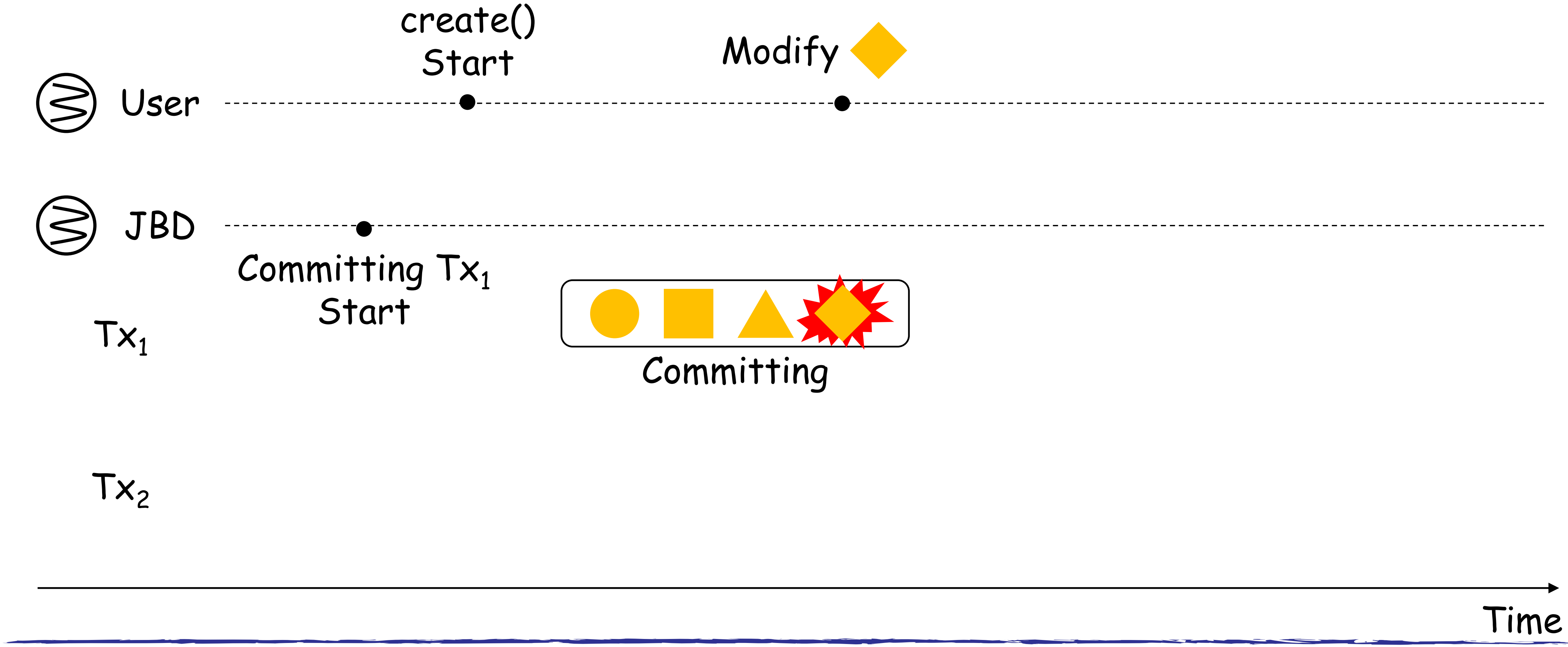
Transaction conflict

The situation that a file operation modifies a page which is being committed



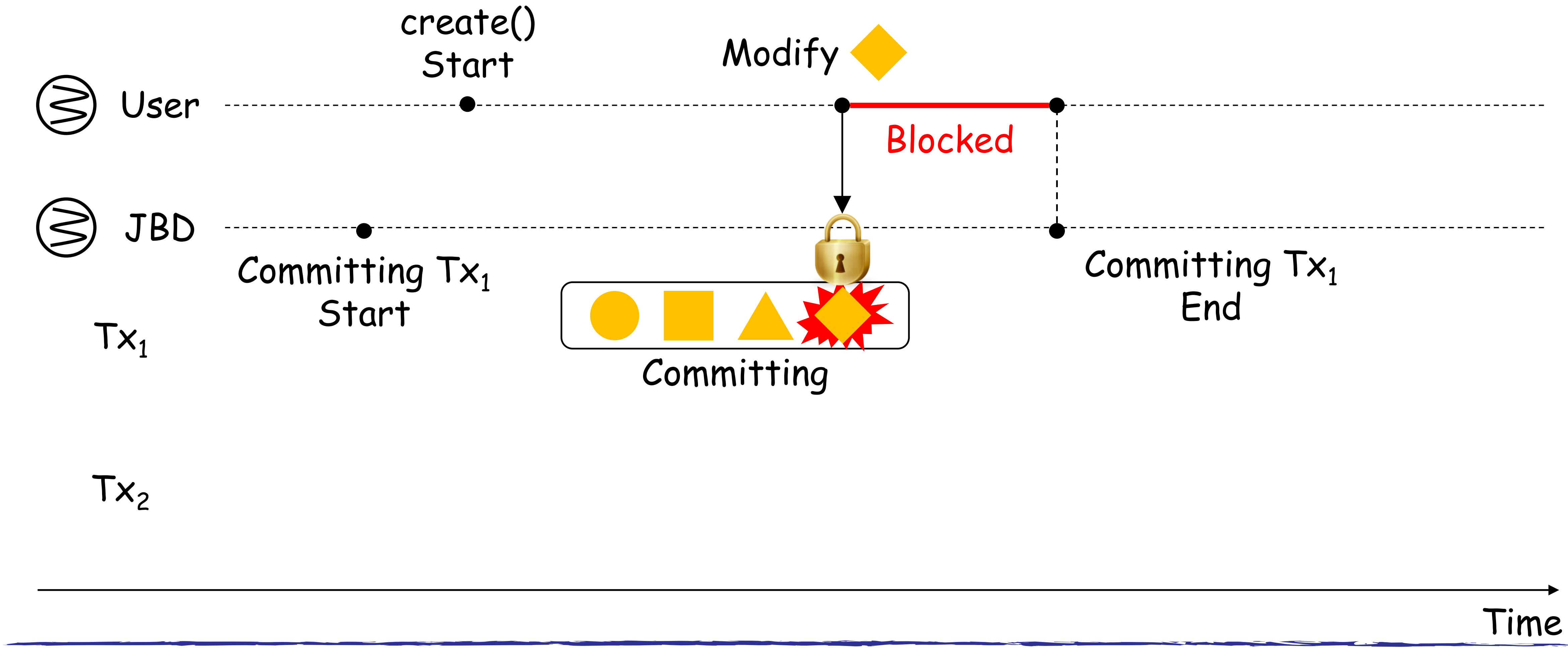
Transaction conflict

The situation that a file operation modifies a page which is being committed



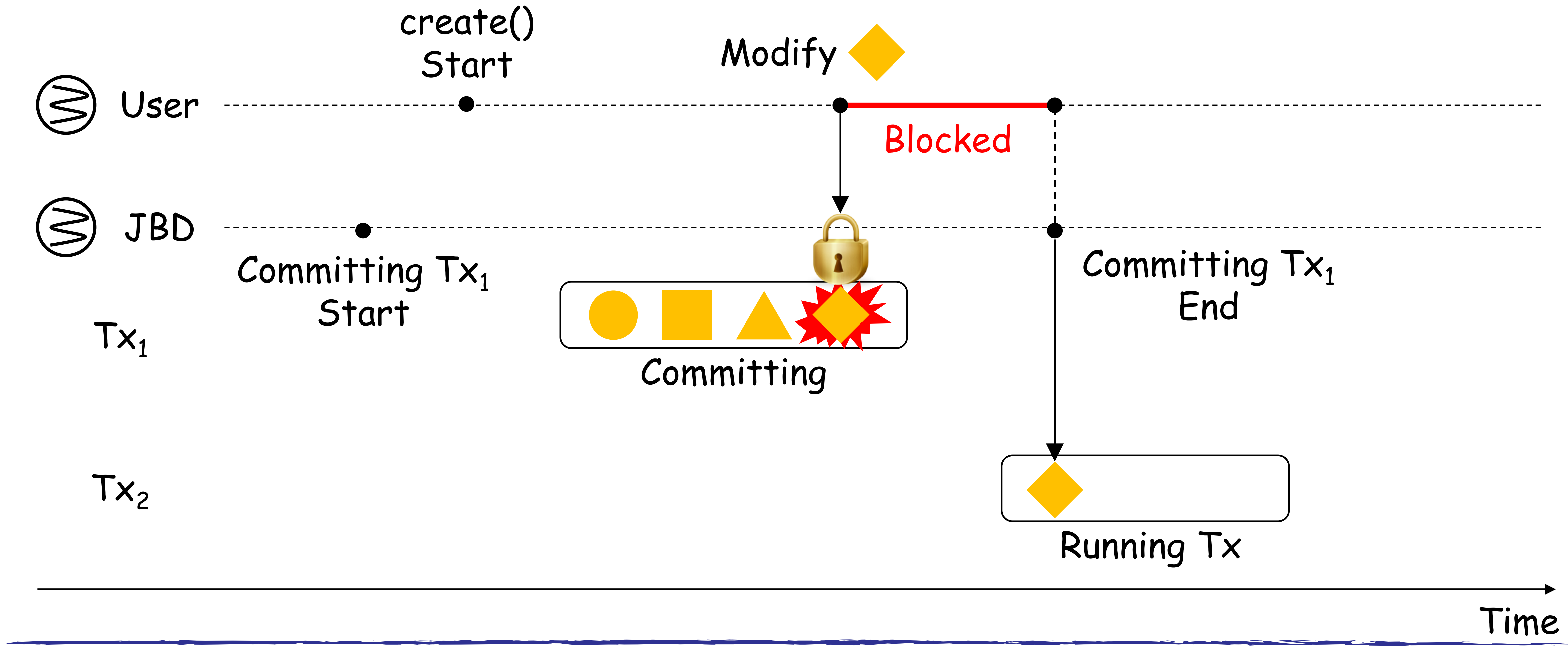
Transaction conflict

The situation that a file operation modifies a page which is being committed



Transaction conflict

The situation that a file operation modifies a page which is being committed



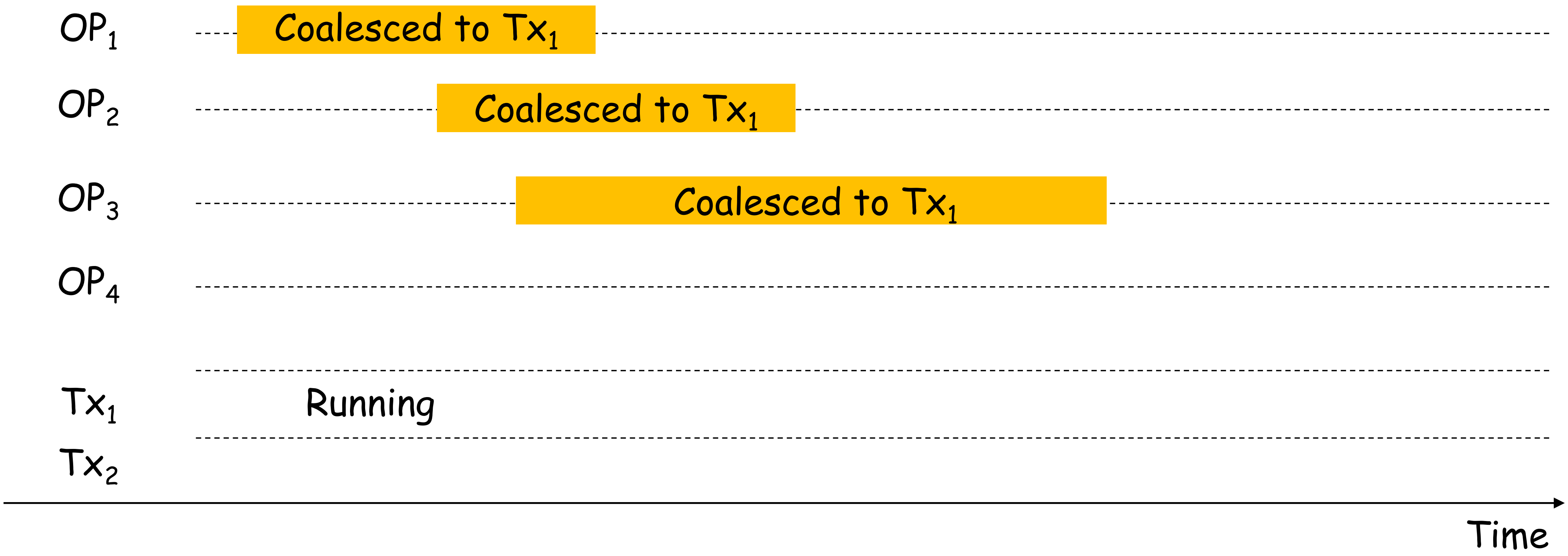
Transaction Lock-Up

the locked period for isolating the running transaction from file operations

OP_1
OP_2
OP_3
OP_4
Tx_1
Tx_2
Time

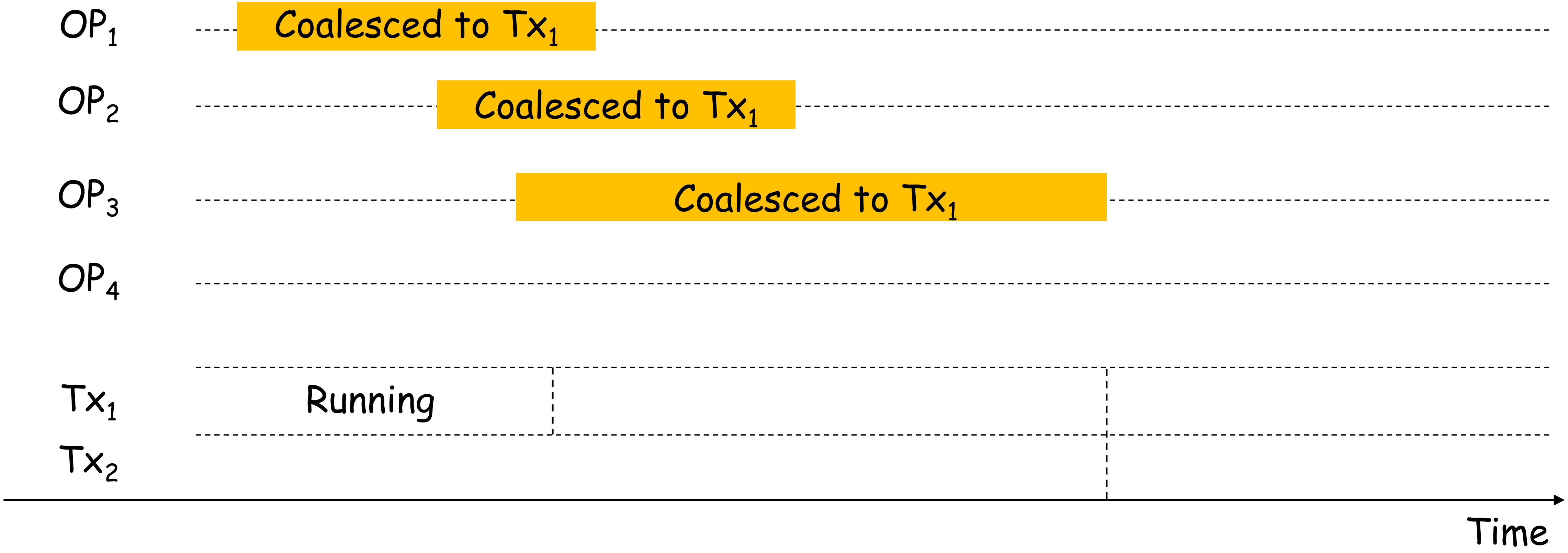
Transaction Lock-Up

the locked period for isolating the running transaction from file operations



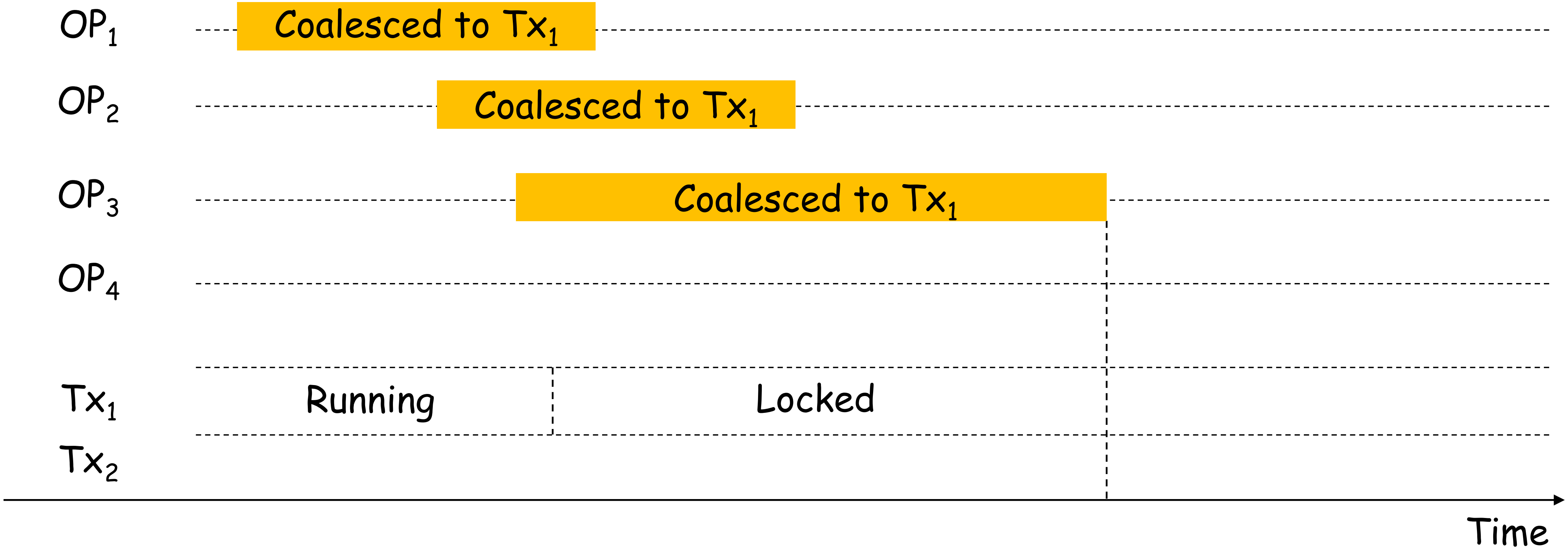
Transaction Lock-Up

the locked period for isolating the running transaction from file operations



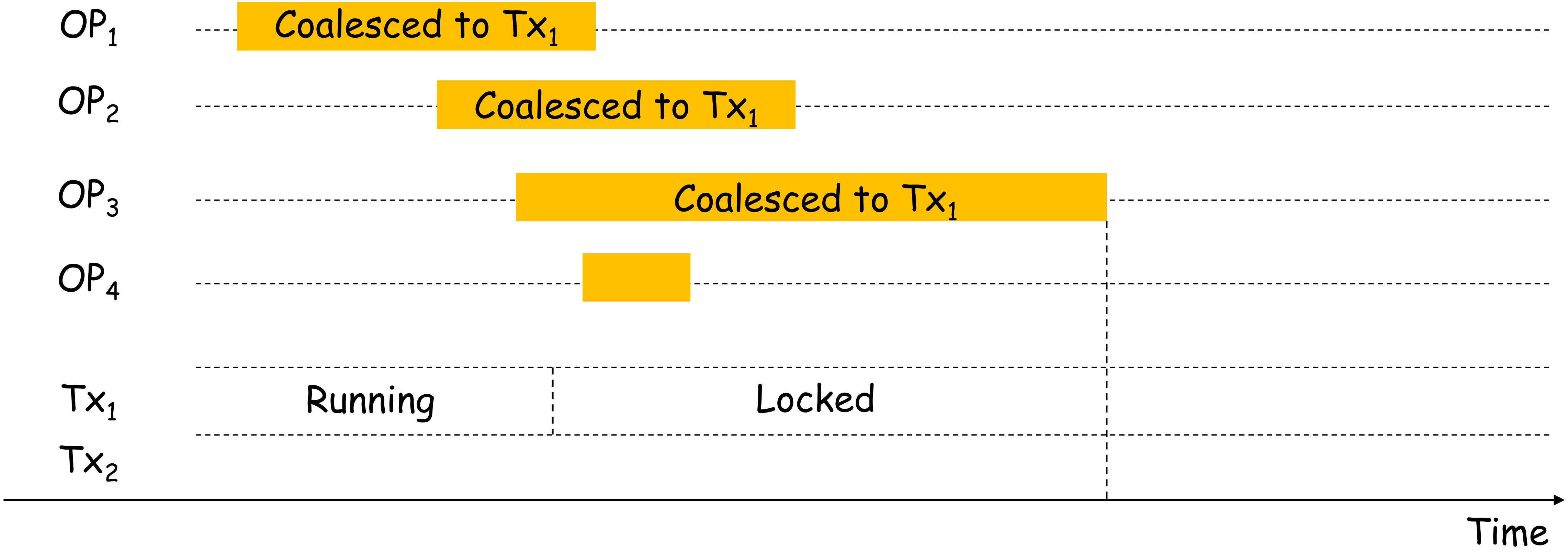
Transaction Lock-Up

the locked period for isolating the running transaction from file operations



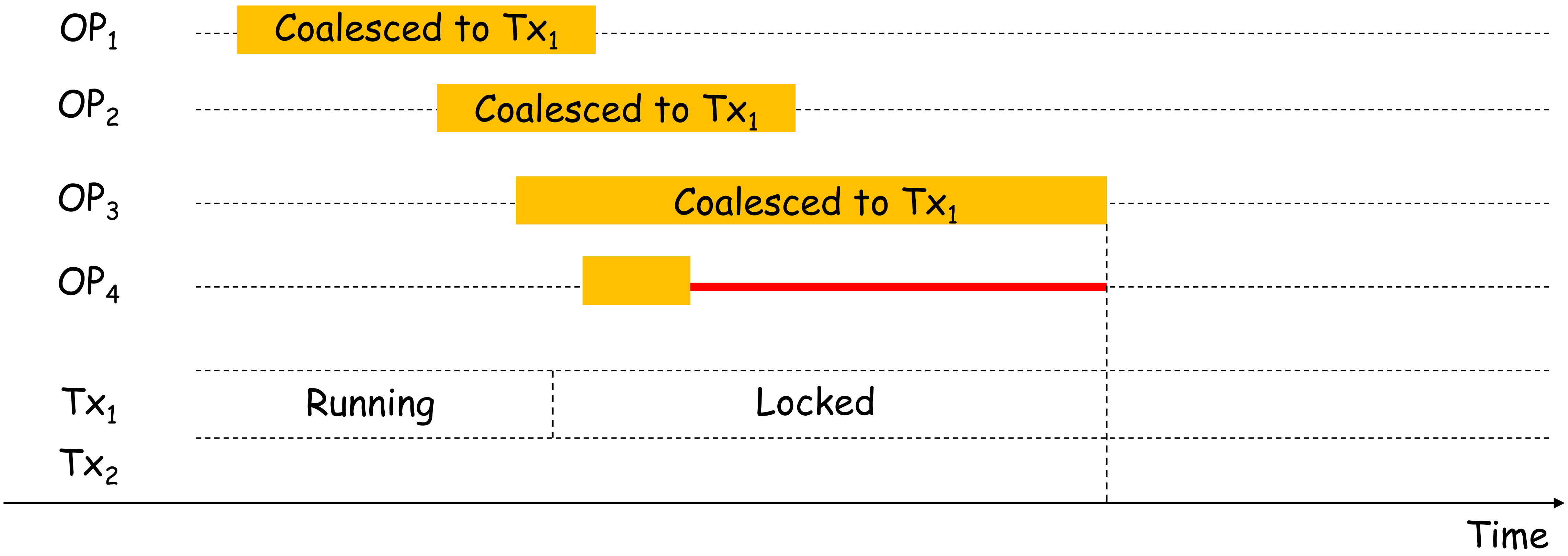
Transaction Lock-Up

the locked period for isolating the running transaction from file operations



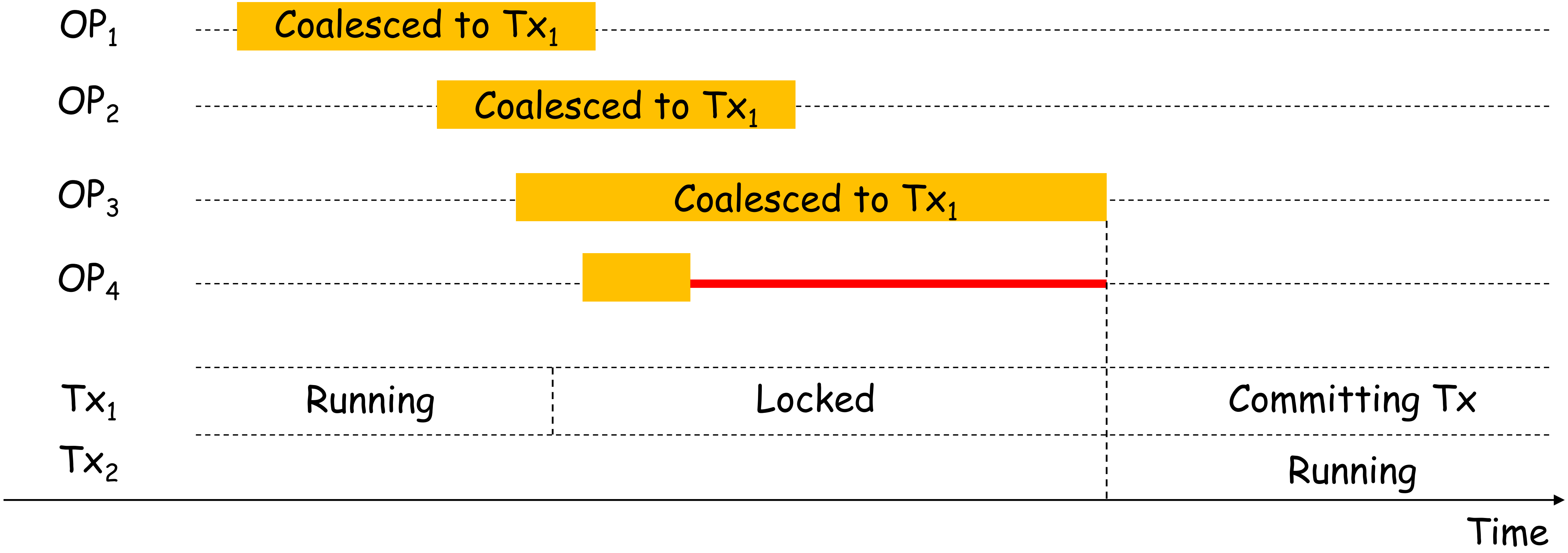
Transaction Lock-Up

the locked period for isolating the running transaction from file operations

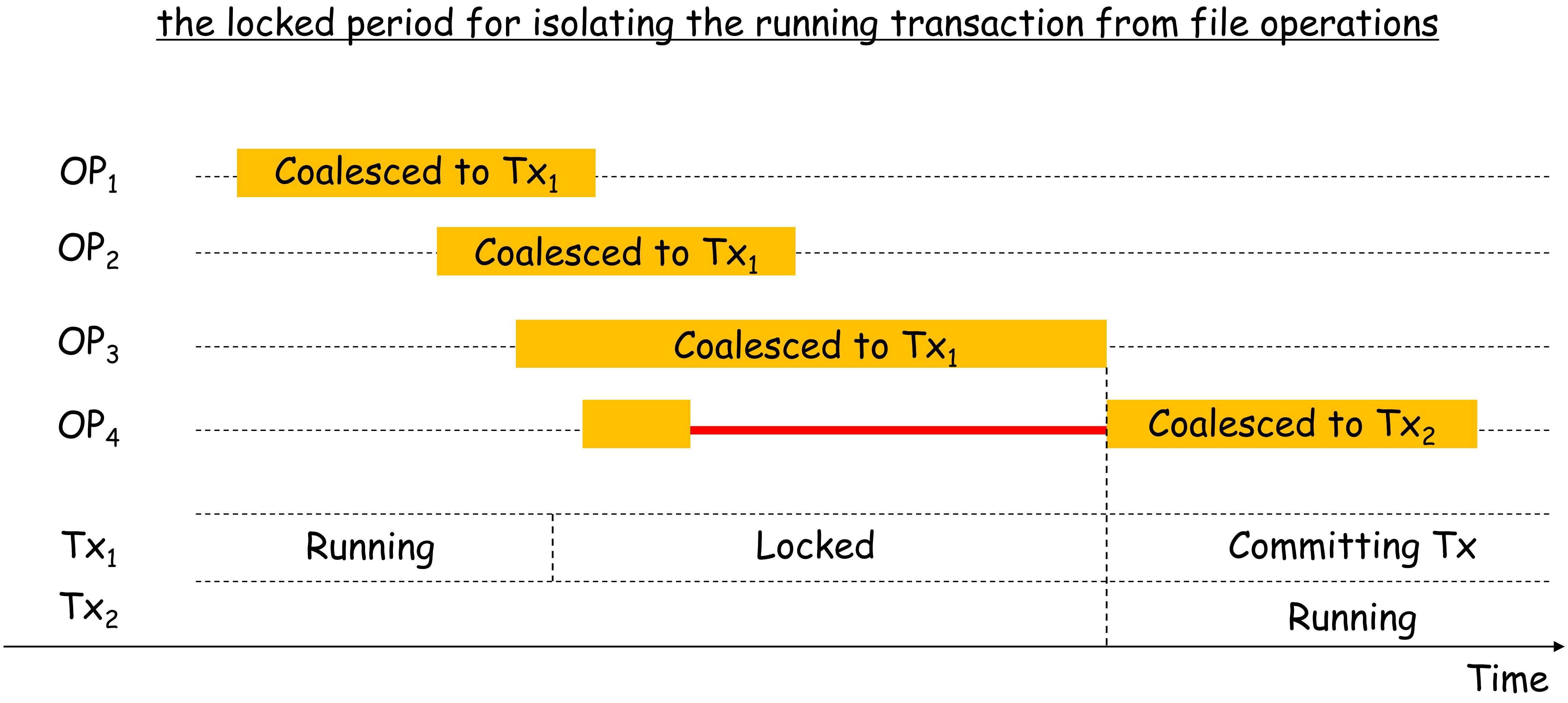


Transaction Lock-Up

the locked period for isolating the running transaction from file operations



Transaction Lock-Up

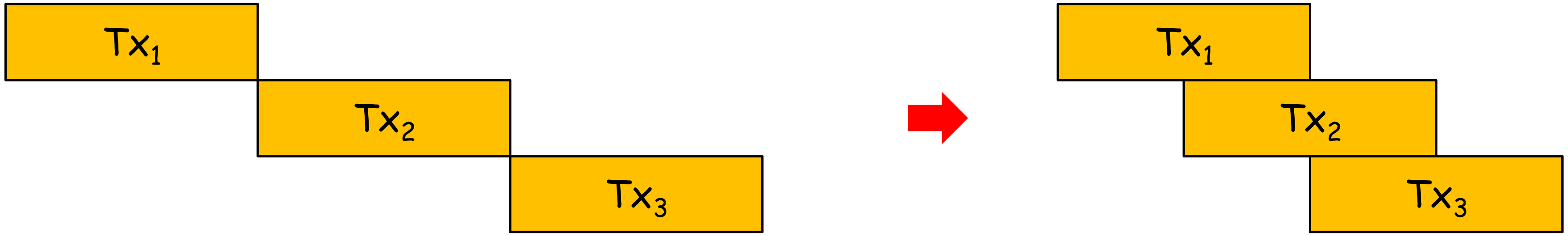


Design:
Concurrent Journaling Filesystem (CJFS)

Design Goals

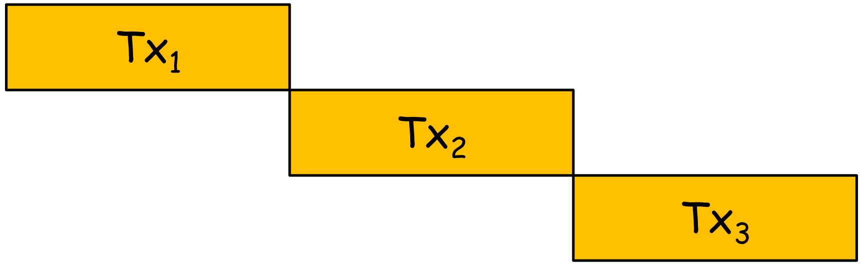
EXT4:

Concurrent Journaling Filesystem (CJFS):

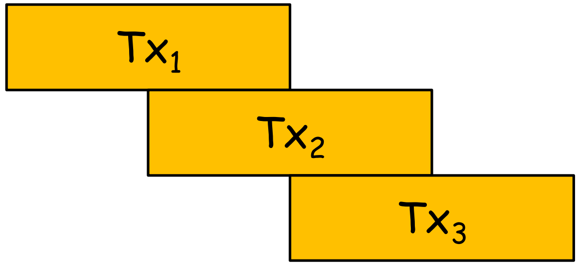


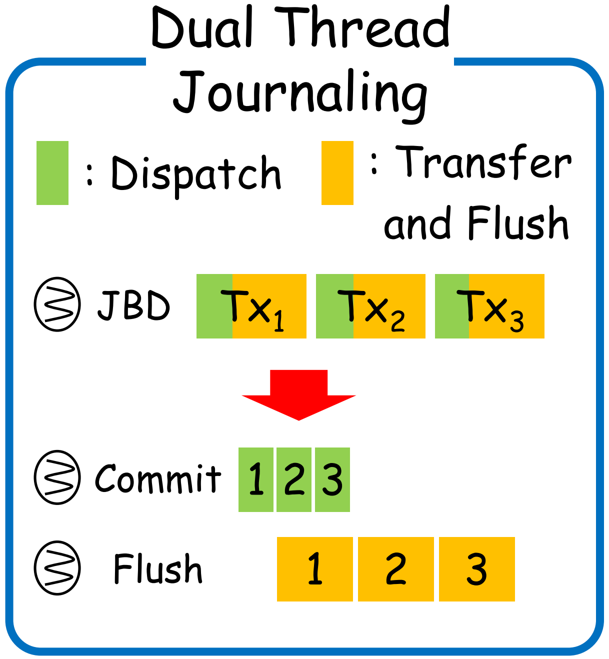
Design Goals

EXT4:



Concurrent Journaling Filesystem (CJFS):

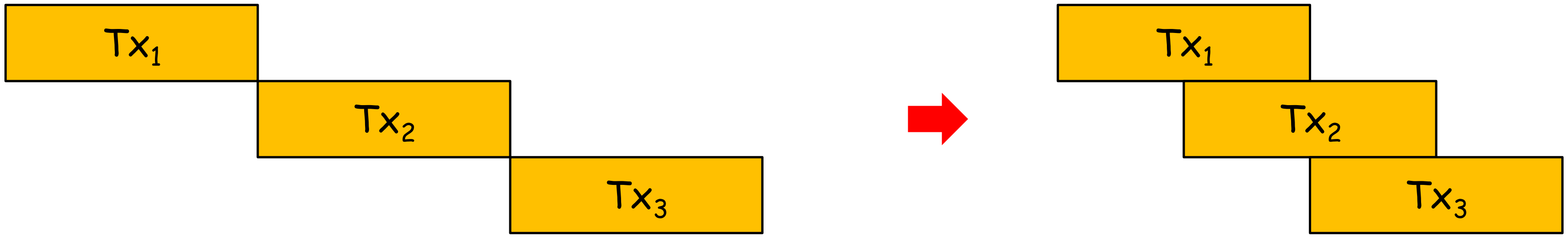


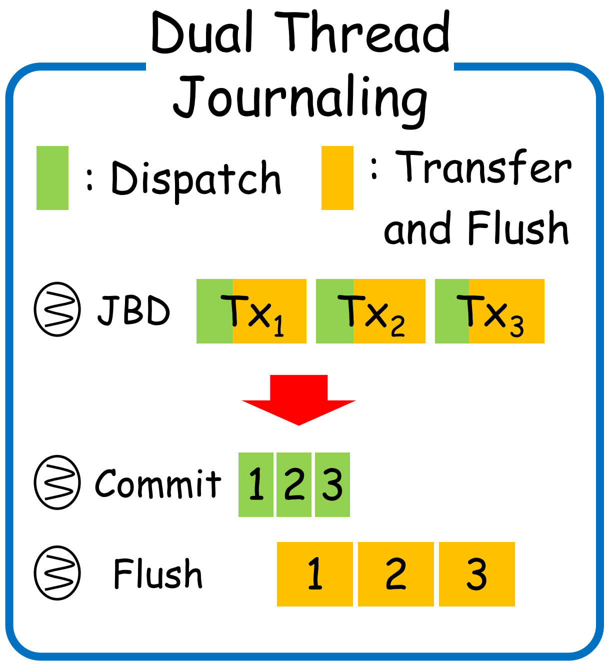


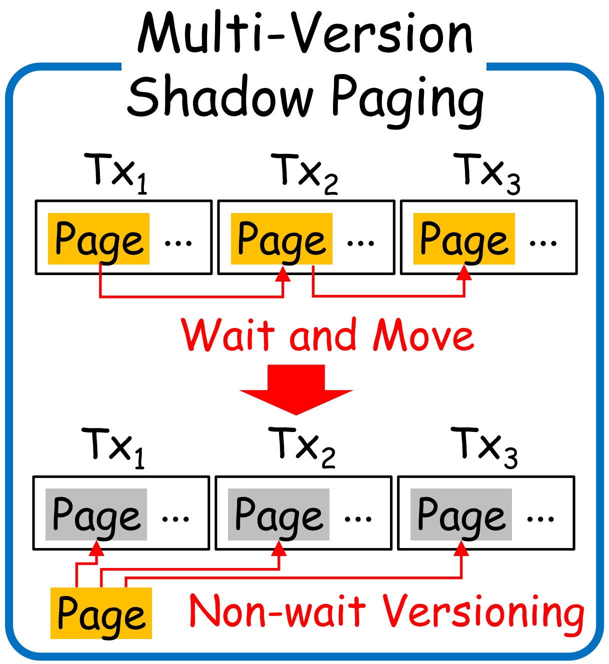
Design Goals

EXT4:

Concurrent Journaling Filesystem (CJFS):



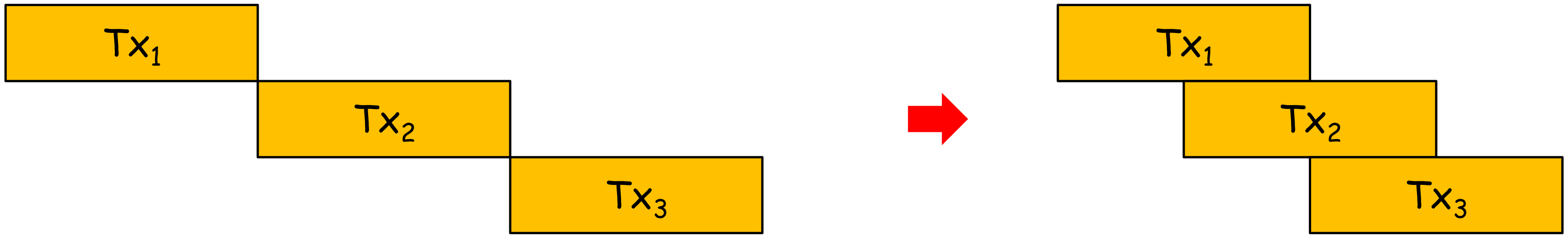


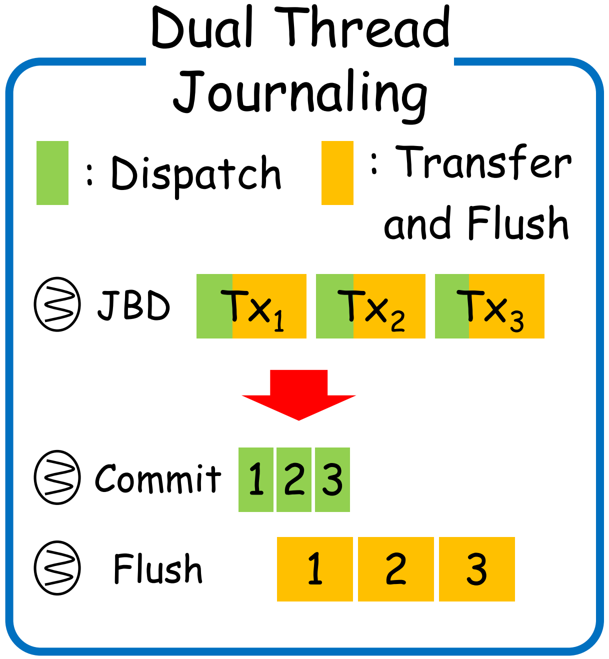


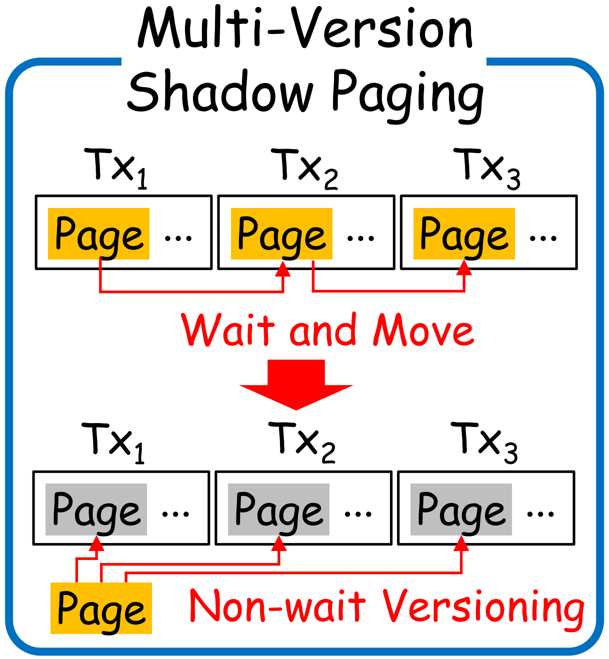
Design Goals

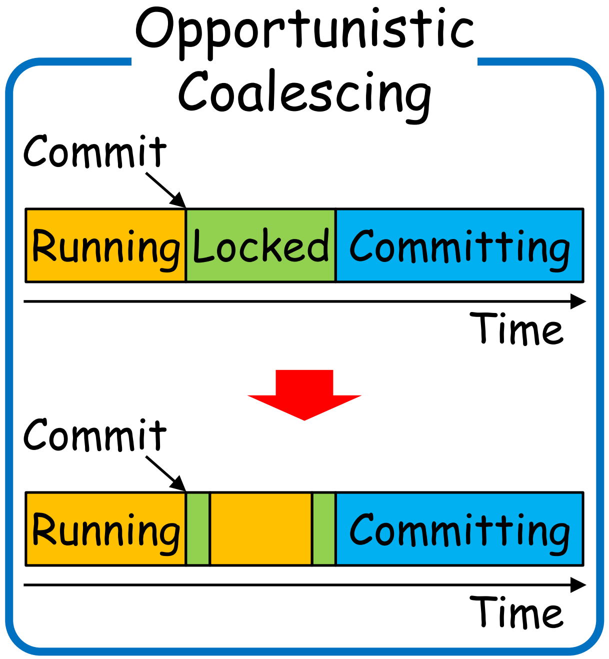
EXT4:

Concurrent Journaling Filesystem (CJFS):





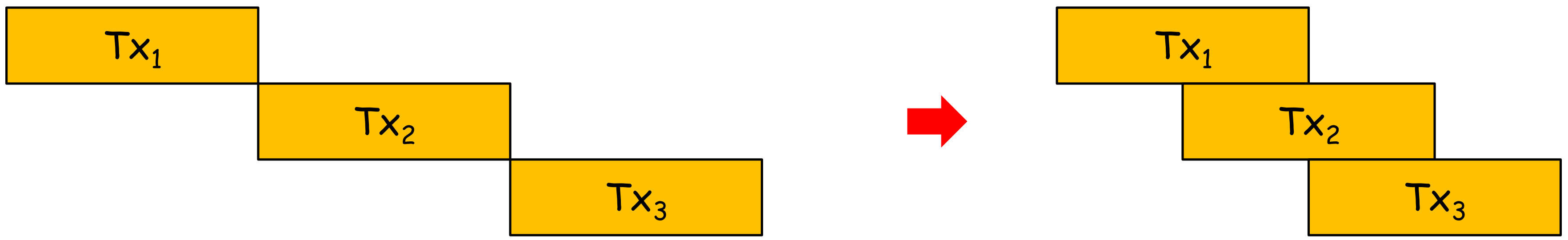




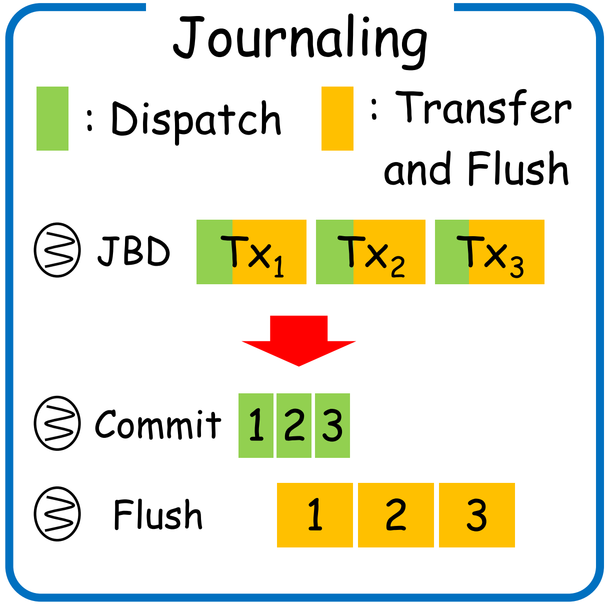
Design Goals

EXT4:

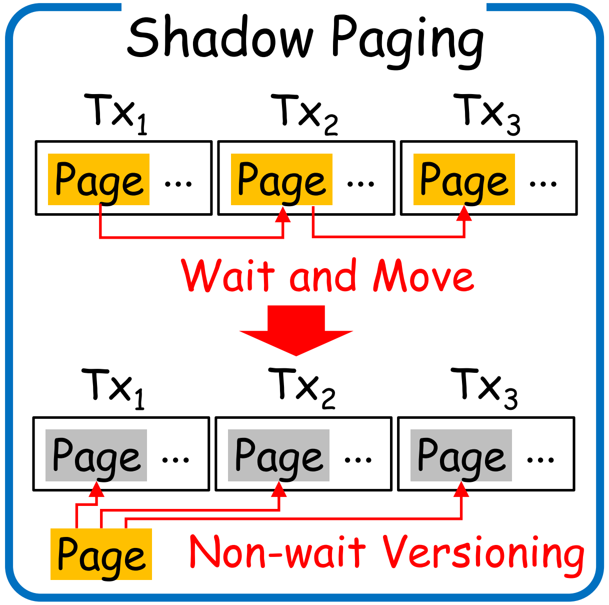
Concurrent Journaling Filesystem (CJFS):



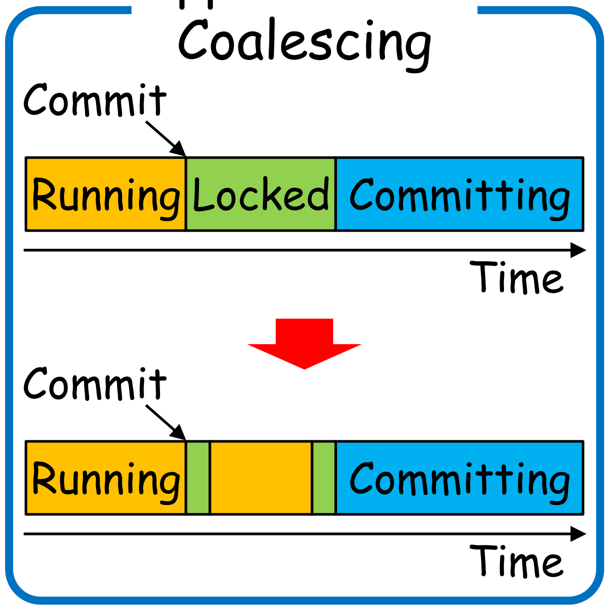
Dual Thread Journaling



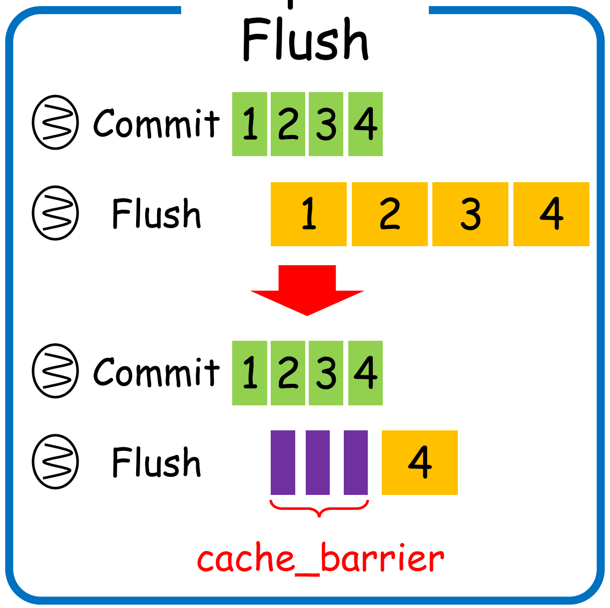
Multi-Version Shadow Paging



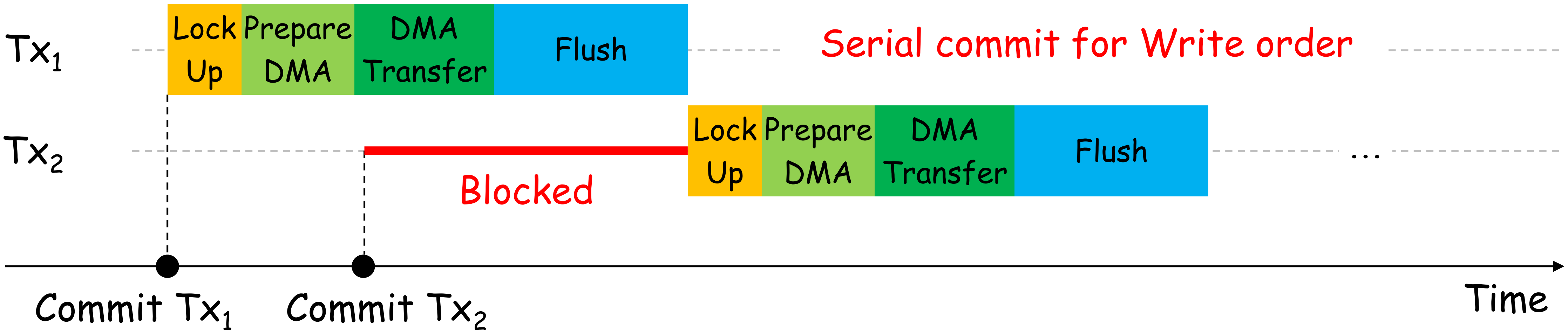
Opportunistic Coalescing



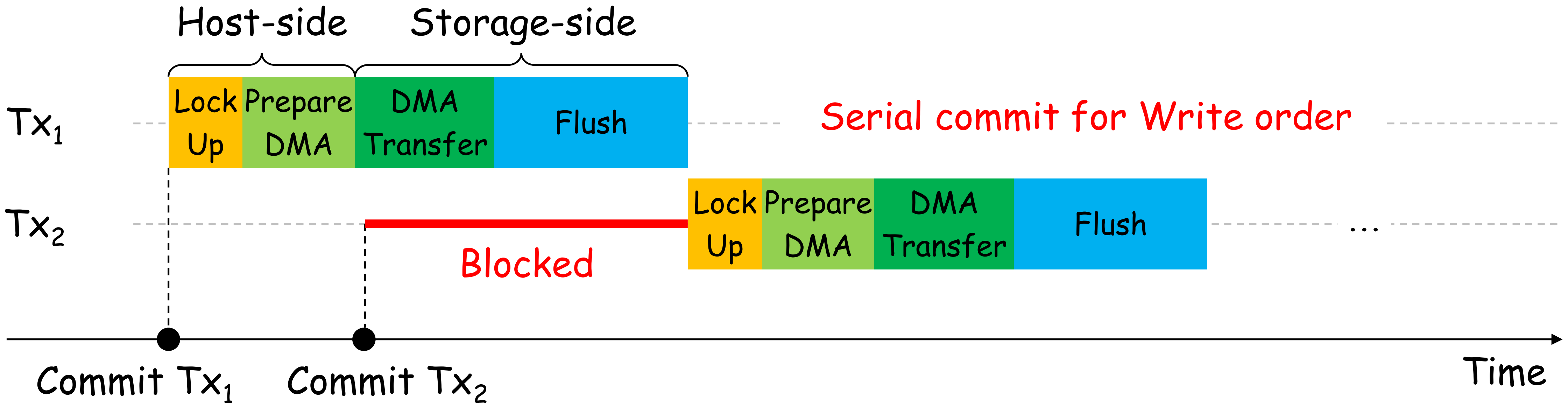
Compound Flush



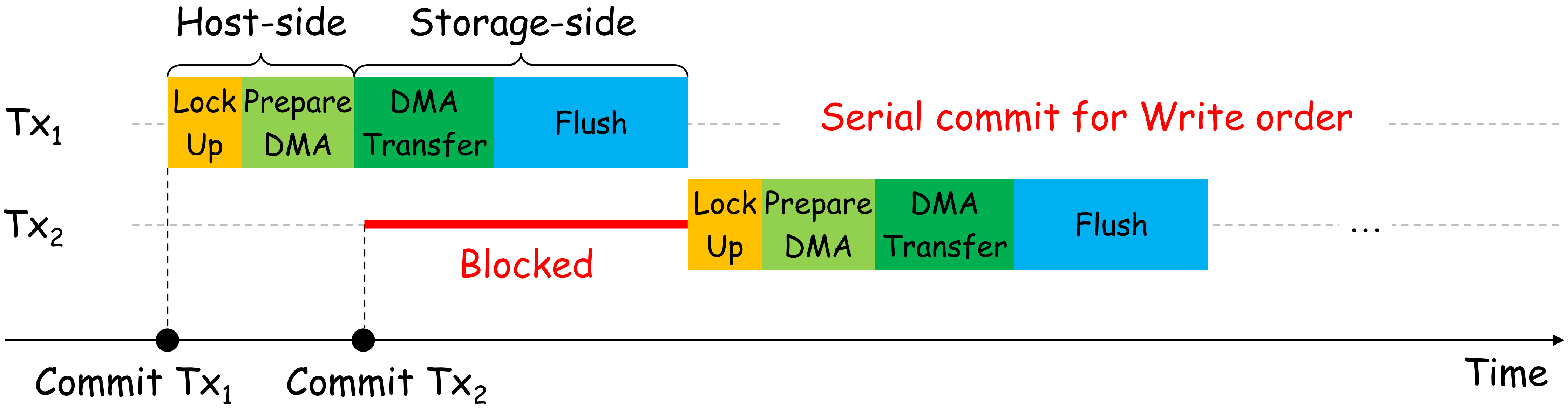
Dual Thread Journaling



Dual Thread Journaling



Dual Thread Journaling



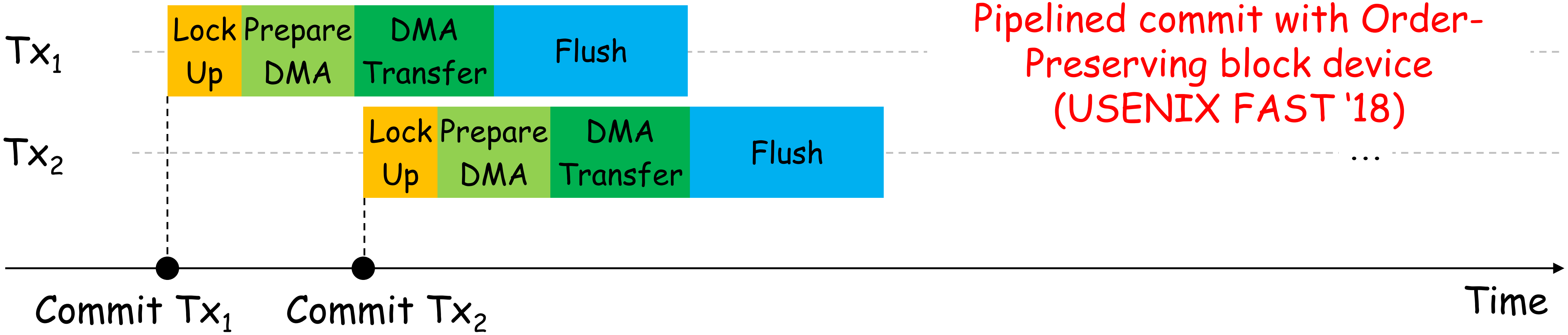
Host-side Storage-side

Tx$_1$: Lock Up | Prepare DMA | DMA Transfer | Flush

Tx$_2$: Blocked | Lock Up | Prepare DMA | DMA Transfer | Flush …

Commit Tx$_1$ Commit Tx$_2$ Time

Serial commit for Write order



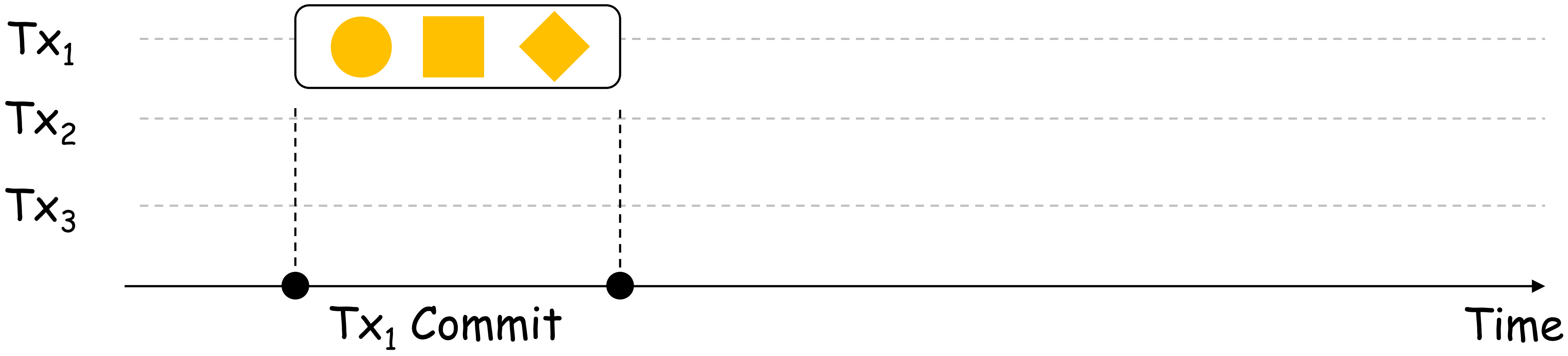
Tx$_1$: Lock Up | Prepare DMA | DMA Transfer | Flush

Tx$_2$: Lock Up | Prepare DMA | DMA Transfer | Flush …

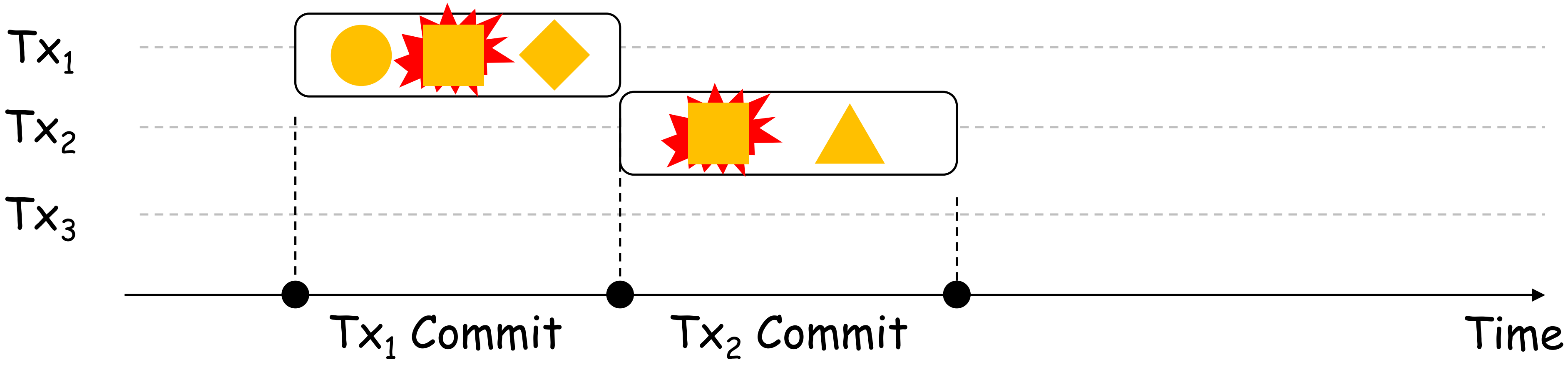
Commit Tx$_1$ Commit Tx$_2$ Time

Pipelined commit with Order-Preserving block device (USENIX FAST '18)

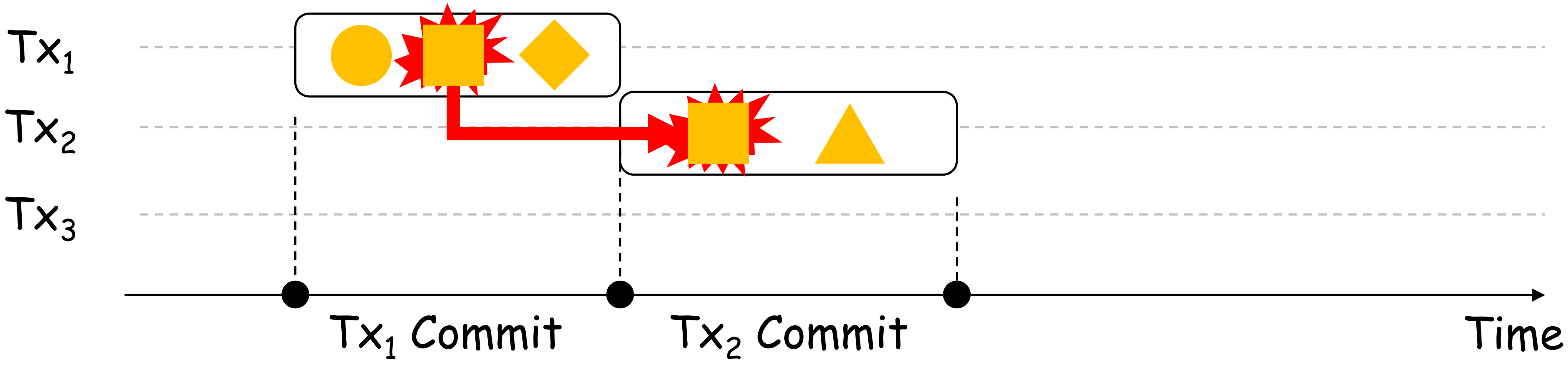
Multi-Version Shadow Paging



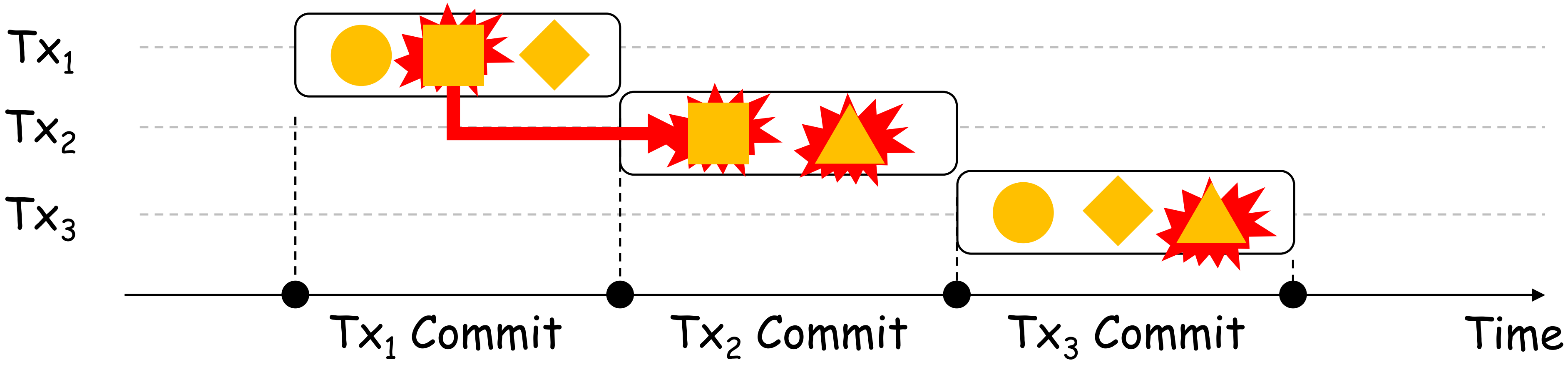
Multi-Version Shadow Paging



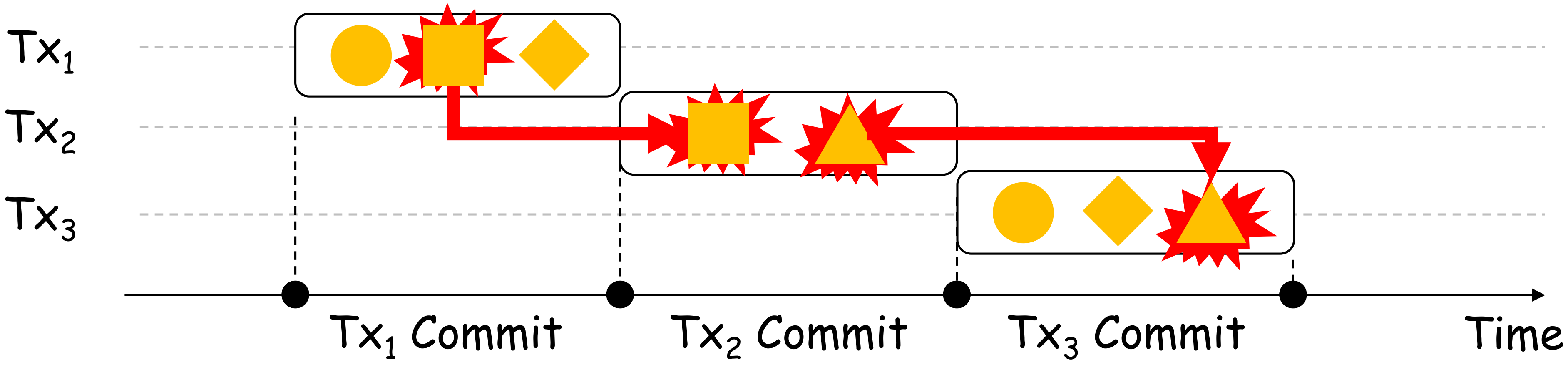
Multi-Version Shadow Paging



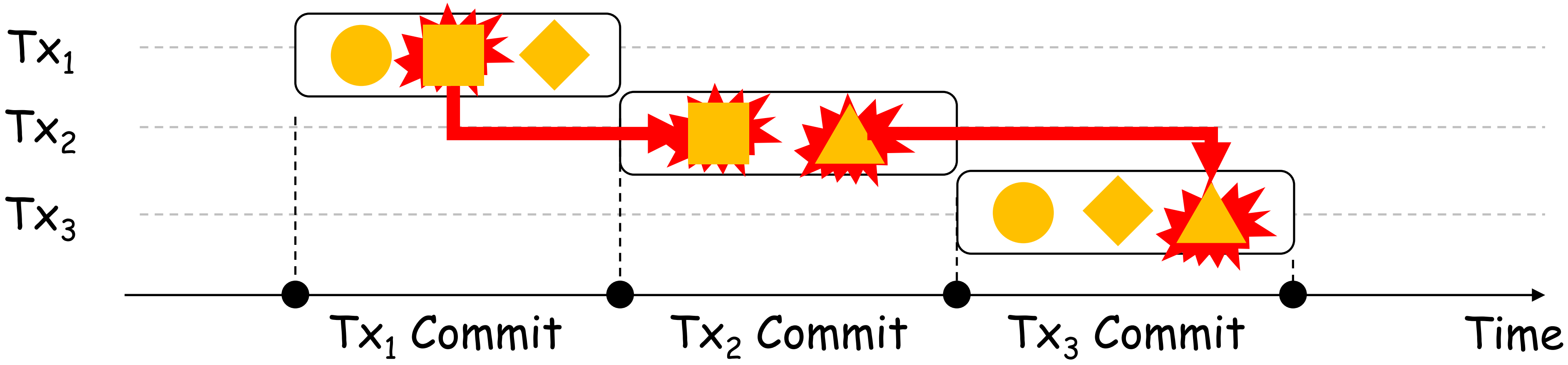
Multi-Version Shadow Paging



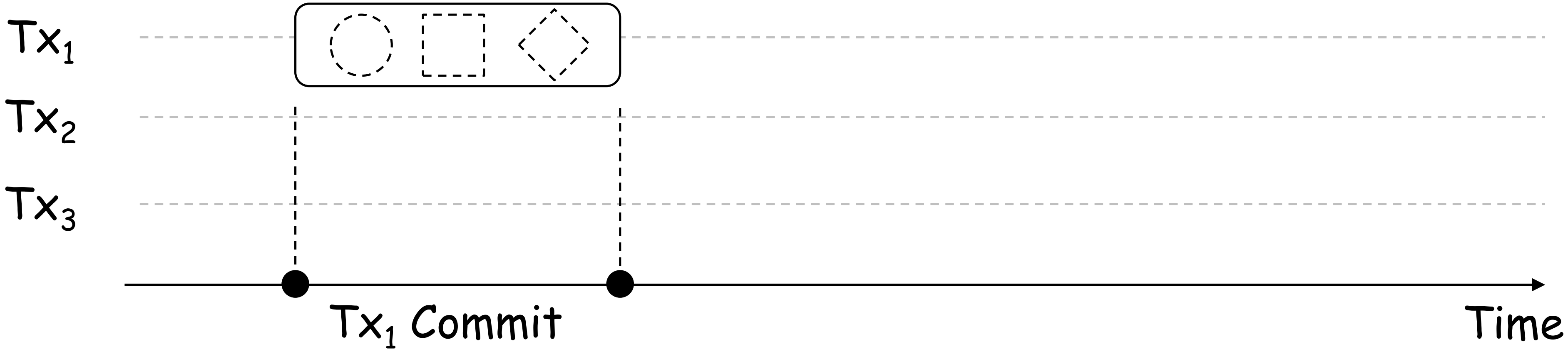
Multi-Version Shadow Paging



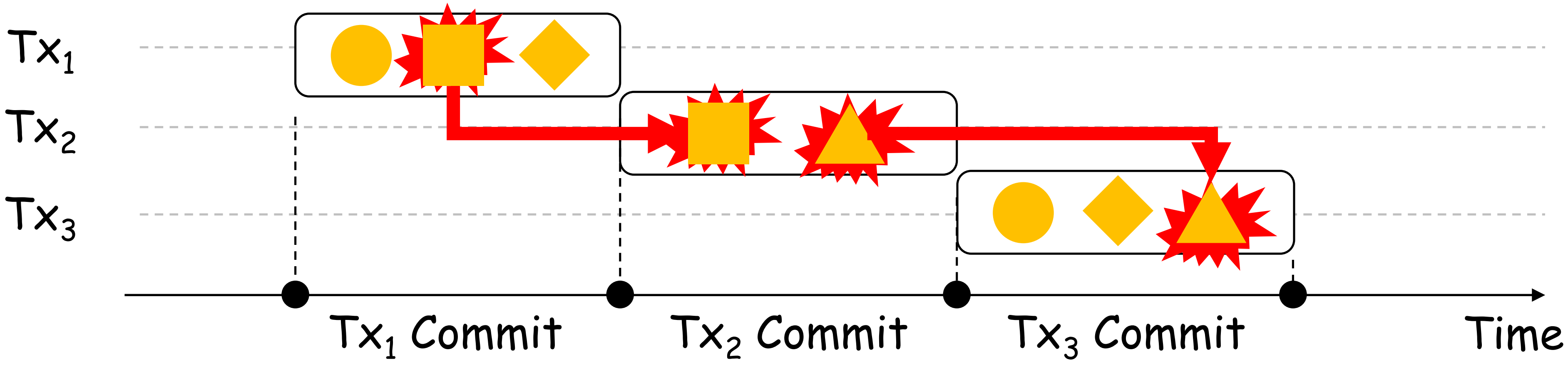
Multi-Version Shadow Paging

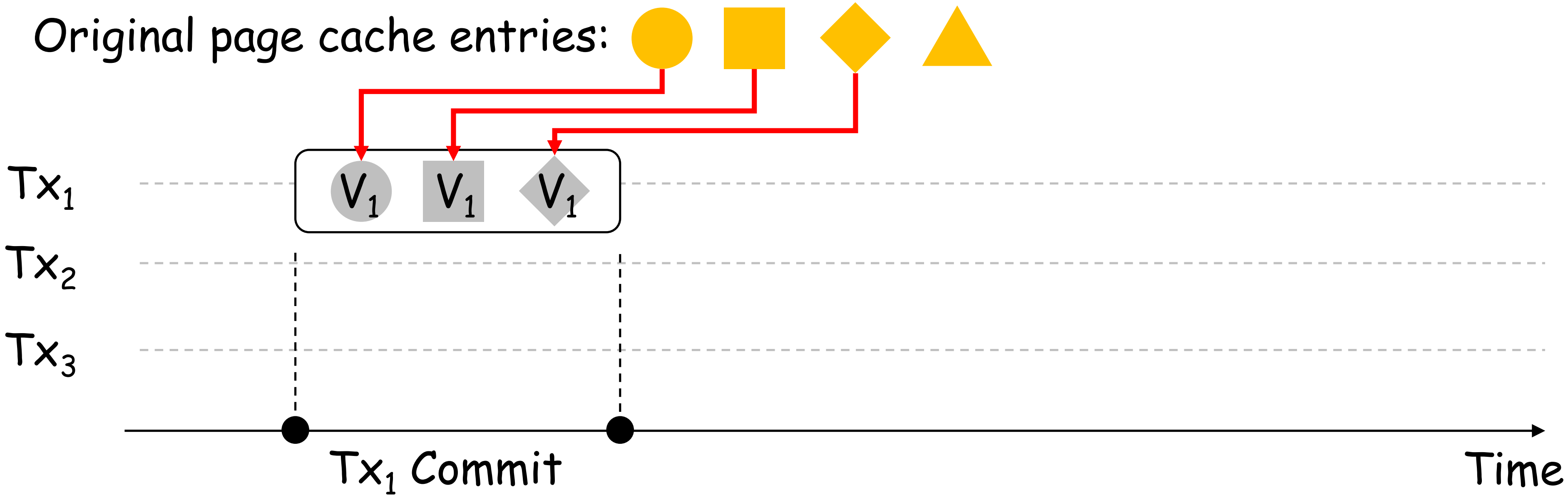


Original page cache entries: ● ■ ◆ ▲

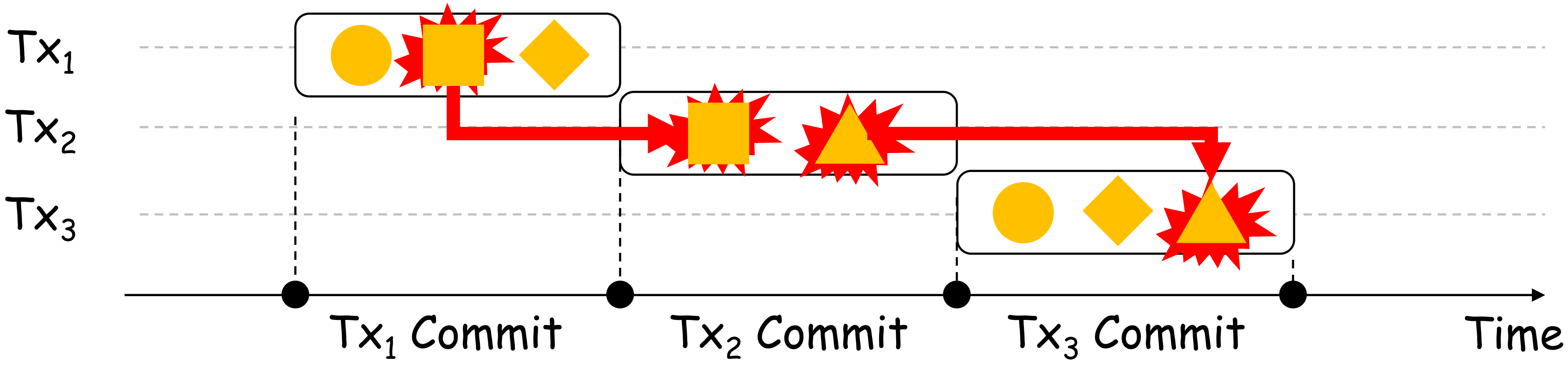


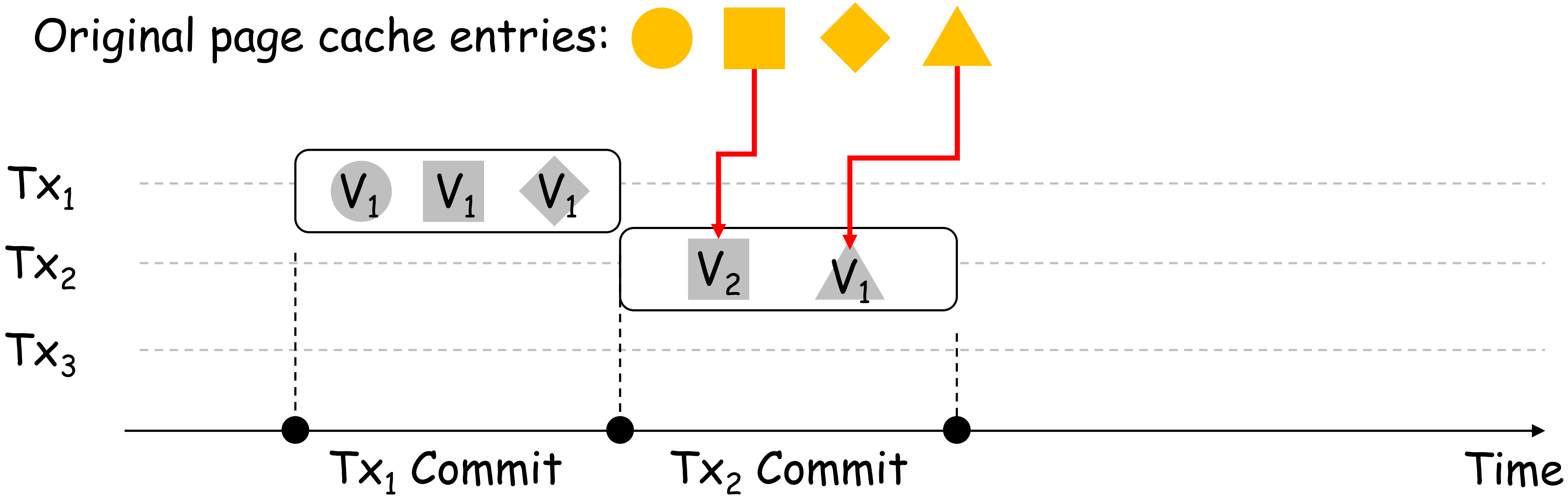
Multi-Version Shadow Paging



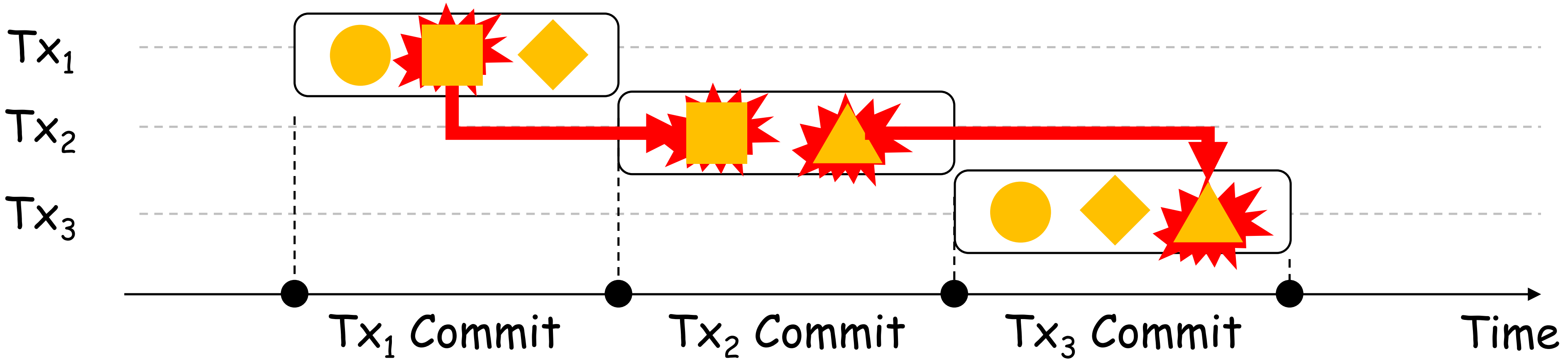


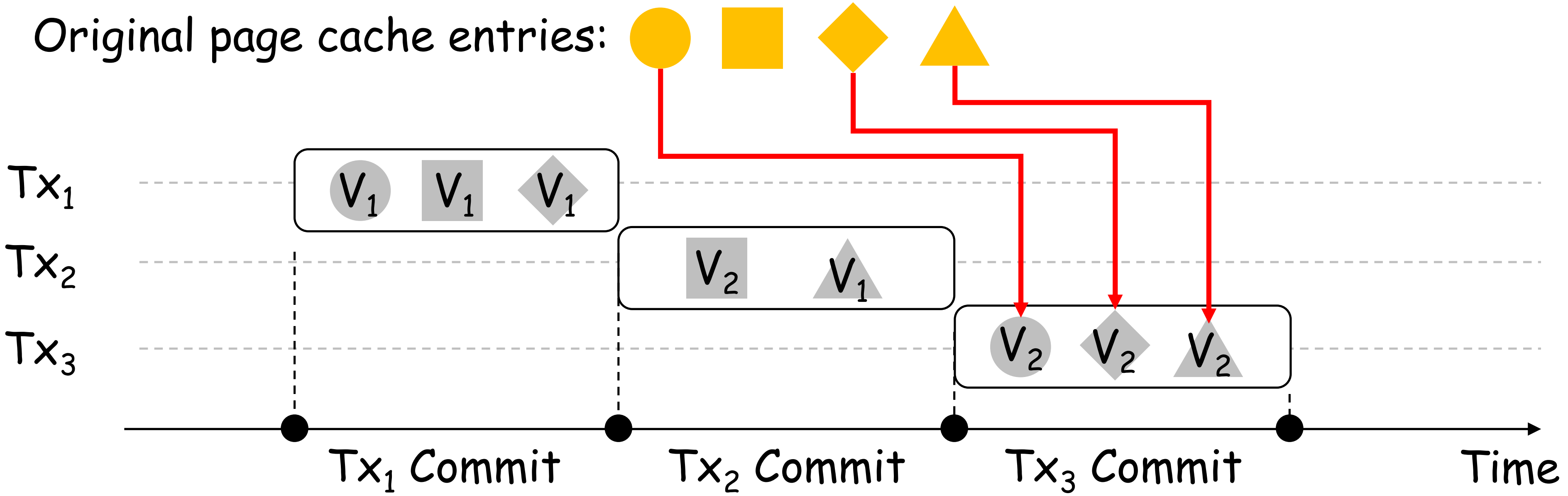
Multi-Version Shadow Paging



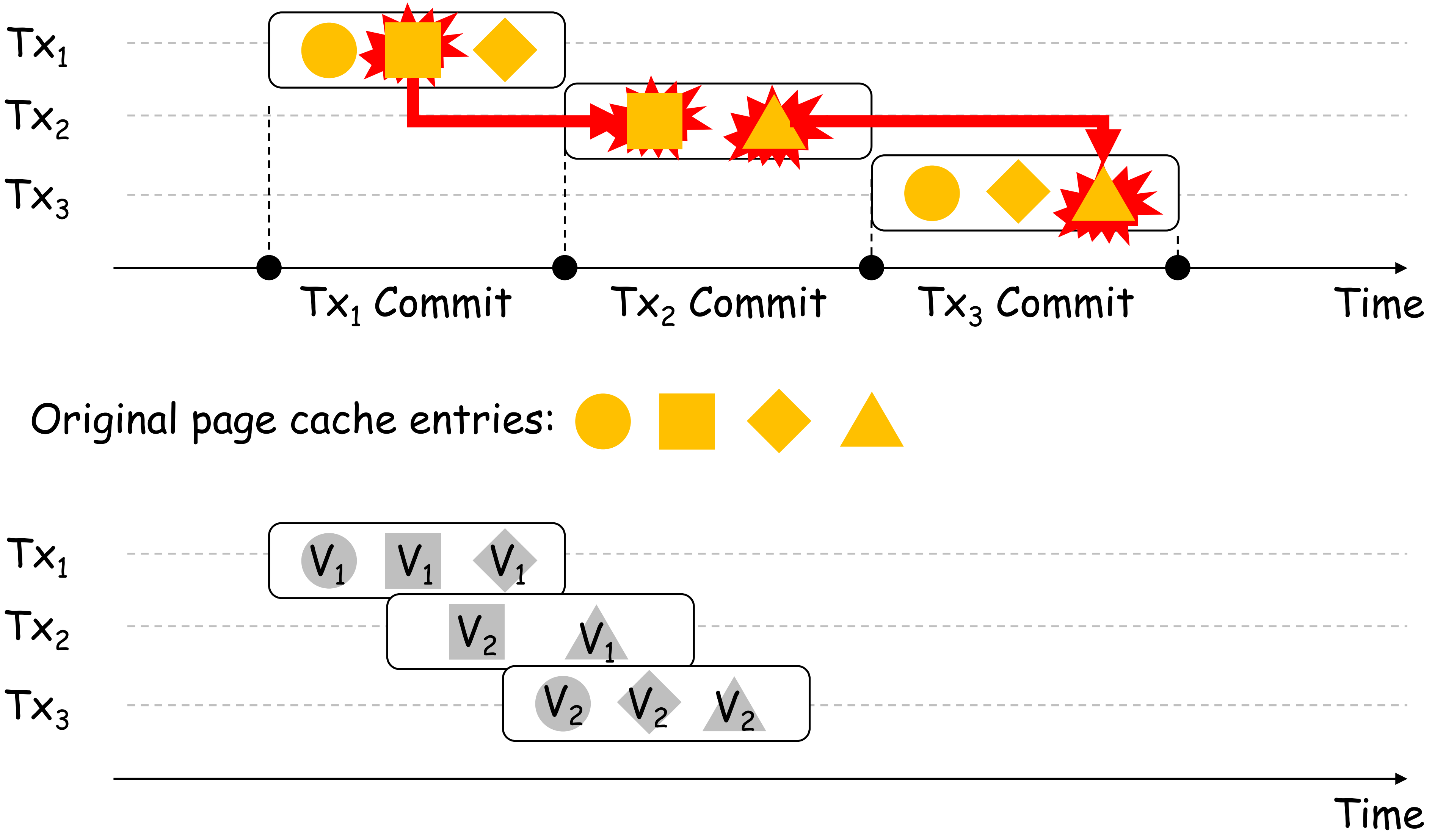


Multi-Version Shadow Paging





Multi-Version Shadow Paging



Tx_1

Tx_2

Tx_3

Tx_1 Commit

Tx_2 Commit

Tx_3 Commit

Time

Original page cache entries:

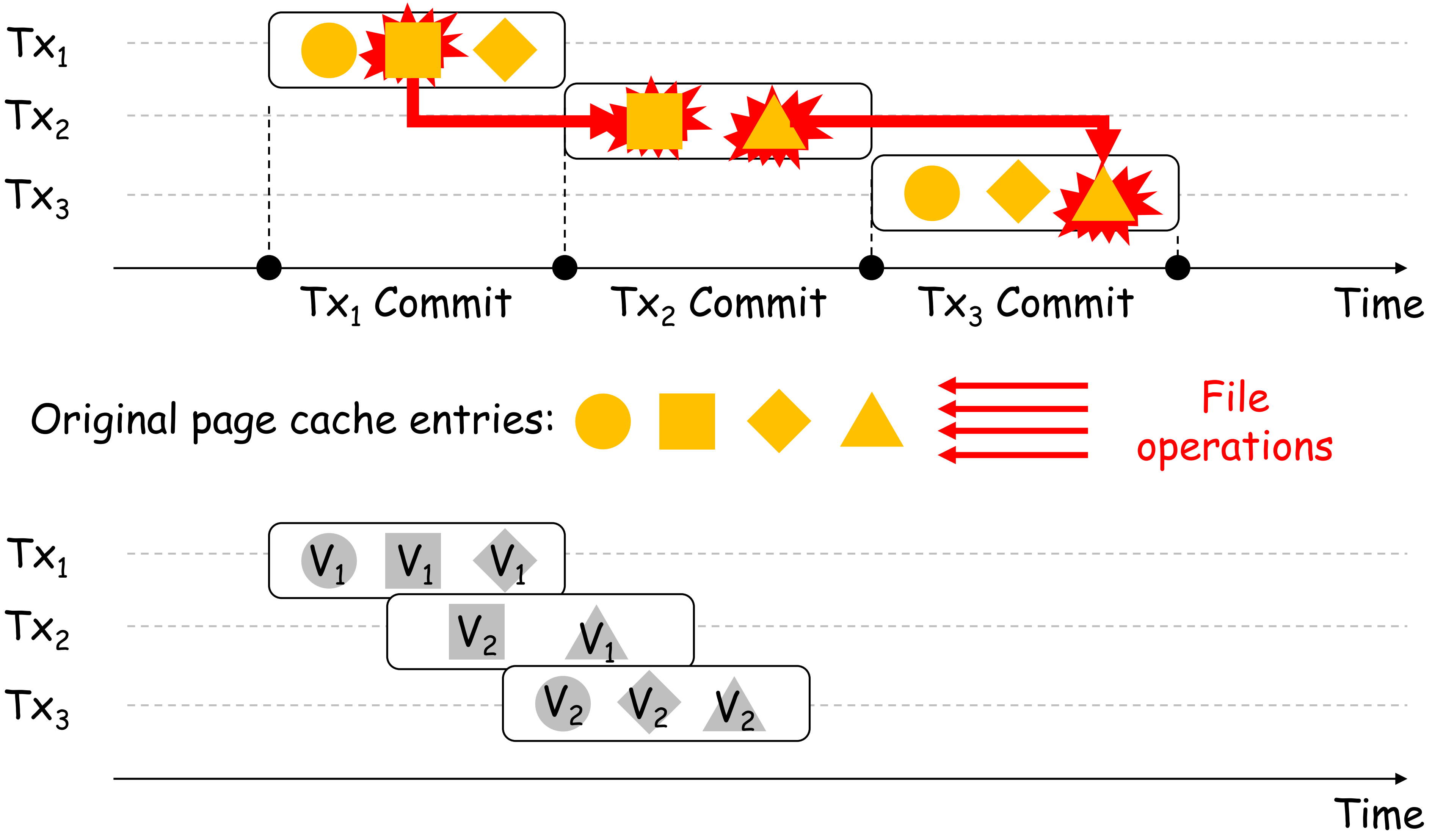
Tx_1 V_1 V_1 V_1

Tx_2 V_2 V_1

Tx_3 V_2 V_2 V_2

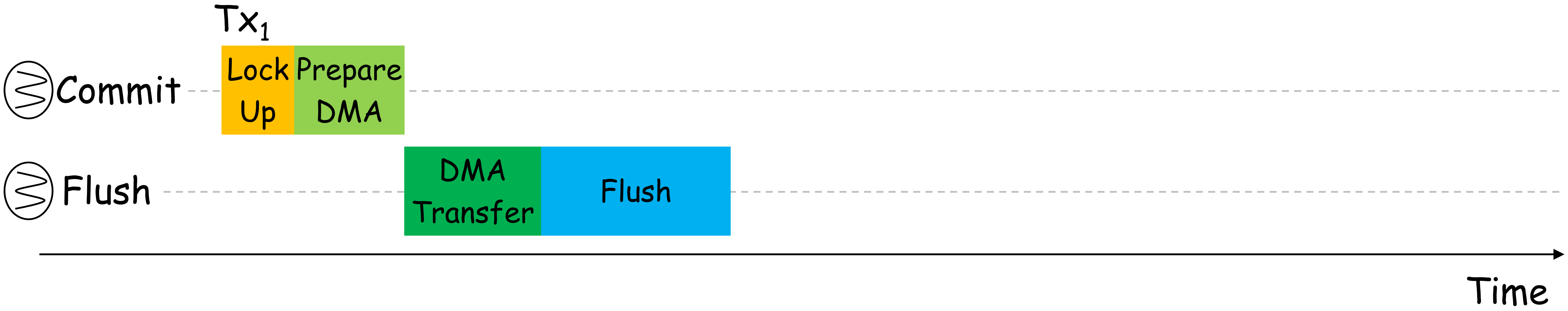
Time

Multi-Version Shadow Paging



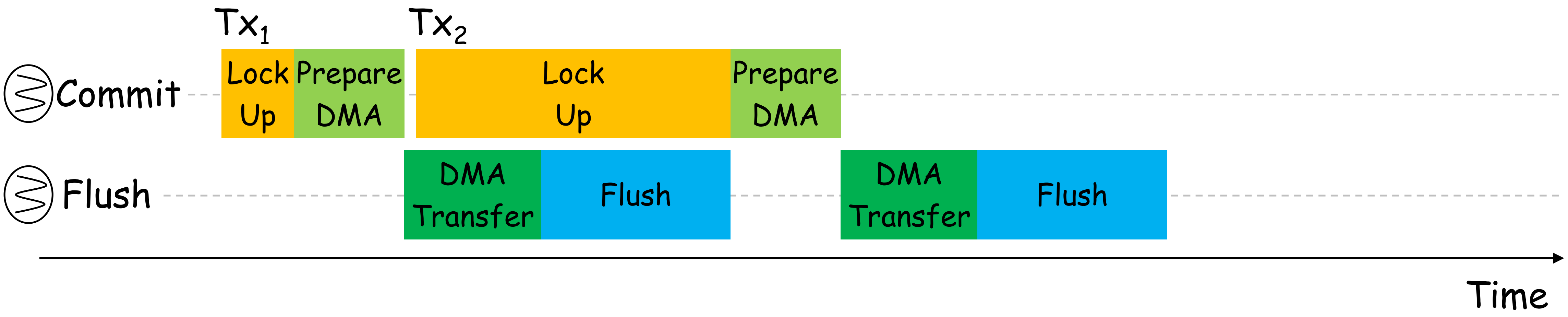
Opportunistic Coalescing

- When versions are exhausted, transaction commits are serialized
- The running transaction is locked and waits for preceding transaction commits



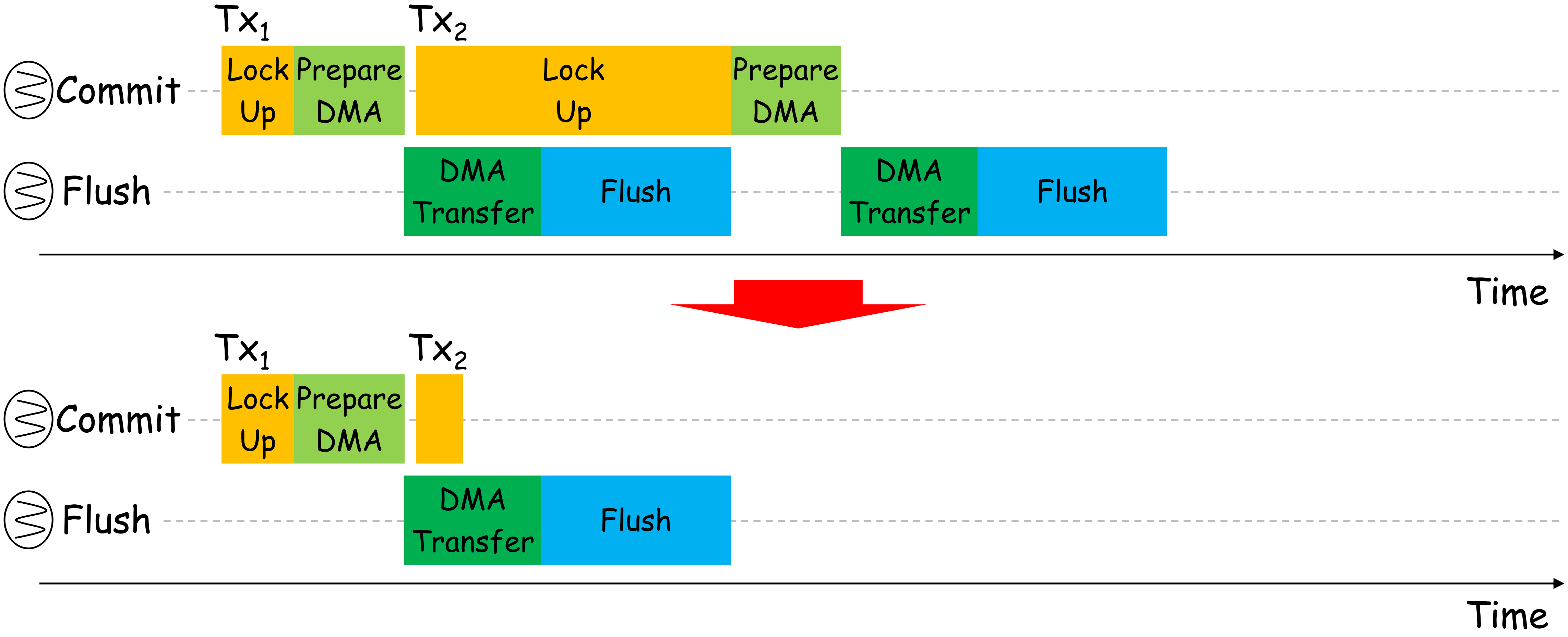
Opportunistic Coalescing

- When versions are exhausted, transaction commits are serialized
- The running transaction is locked and waits for preceding transaction commits



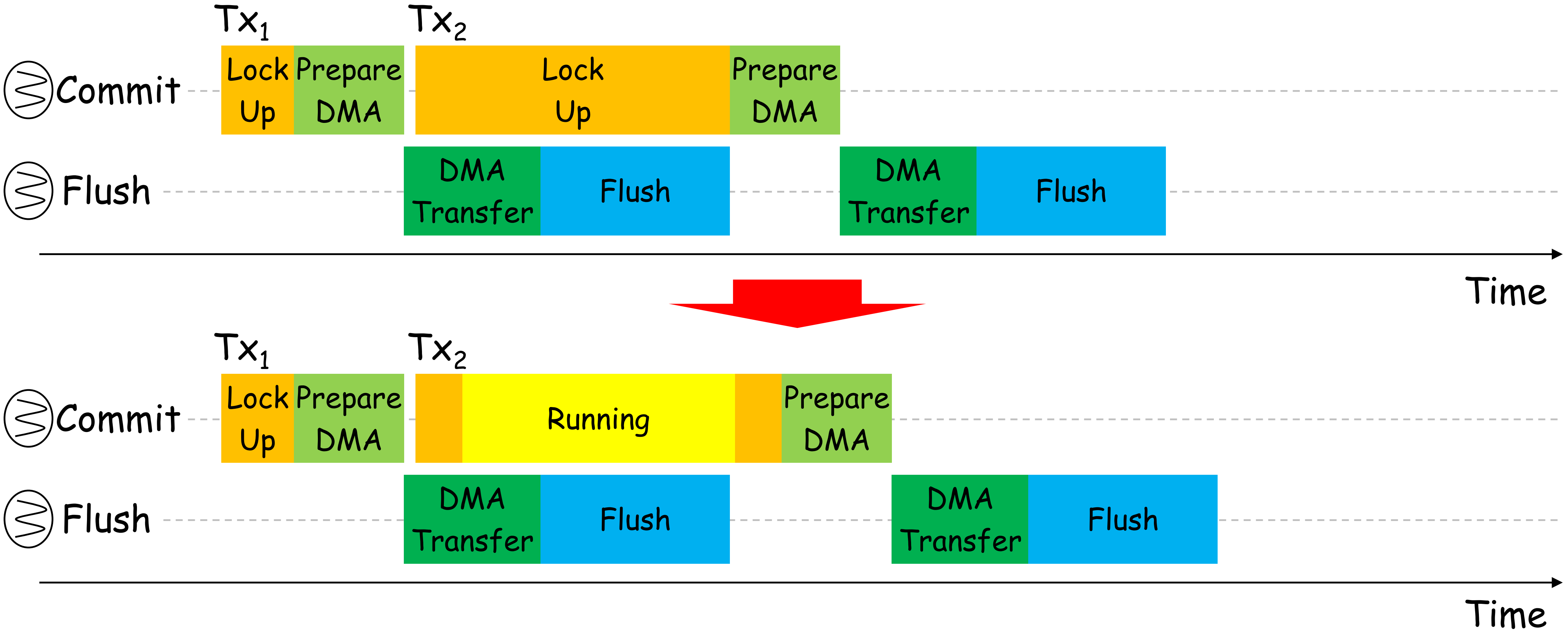
Opportunistic Coalescing

- When versions are exhausted, transaction commits are serialized
- The running transaction is locked and waits for preceding transaction commits

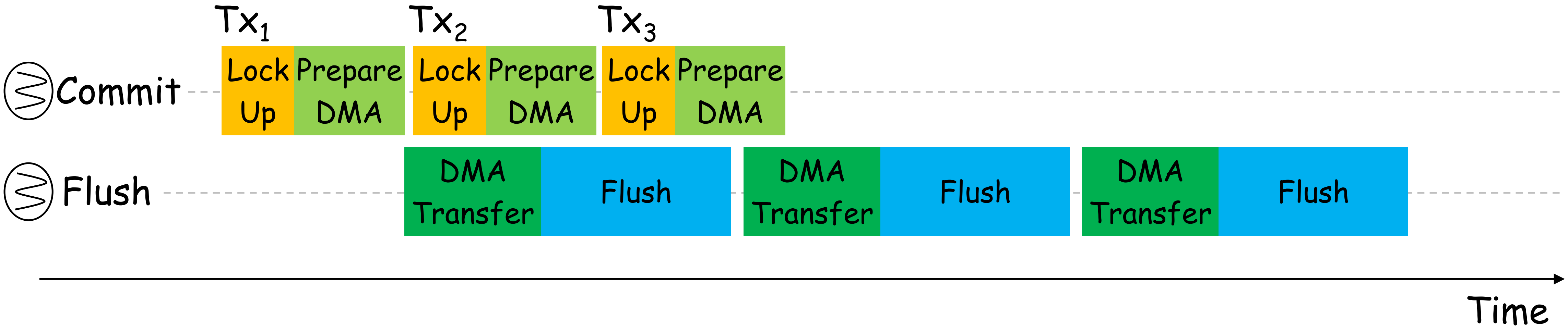


Opportunistic Coalescing

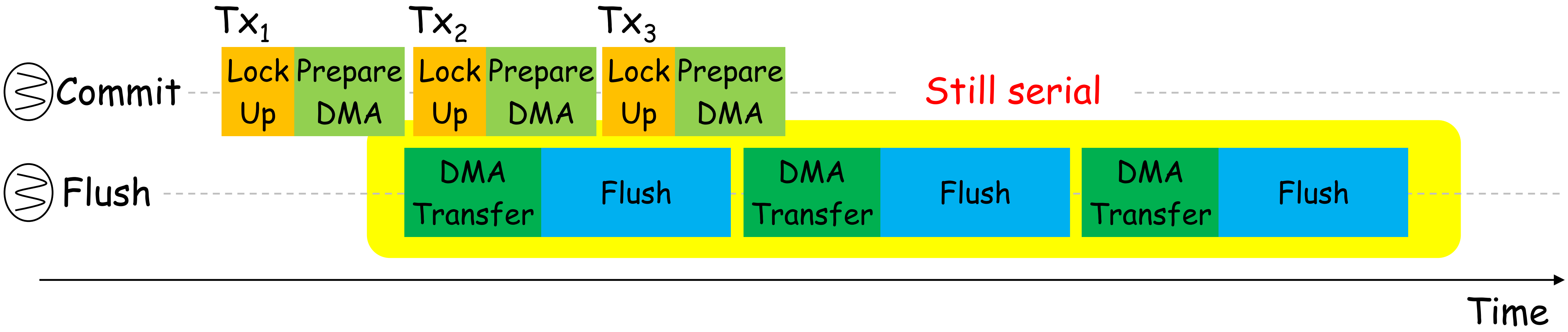
- When versions are exhausted, transaction commits are serialized
- The running transaction is locked and waits for preceding transaction commits



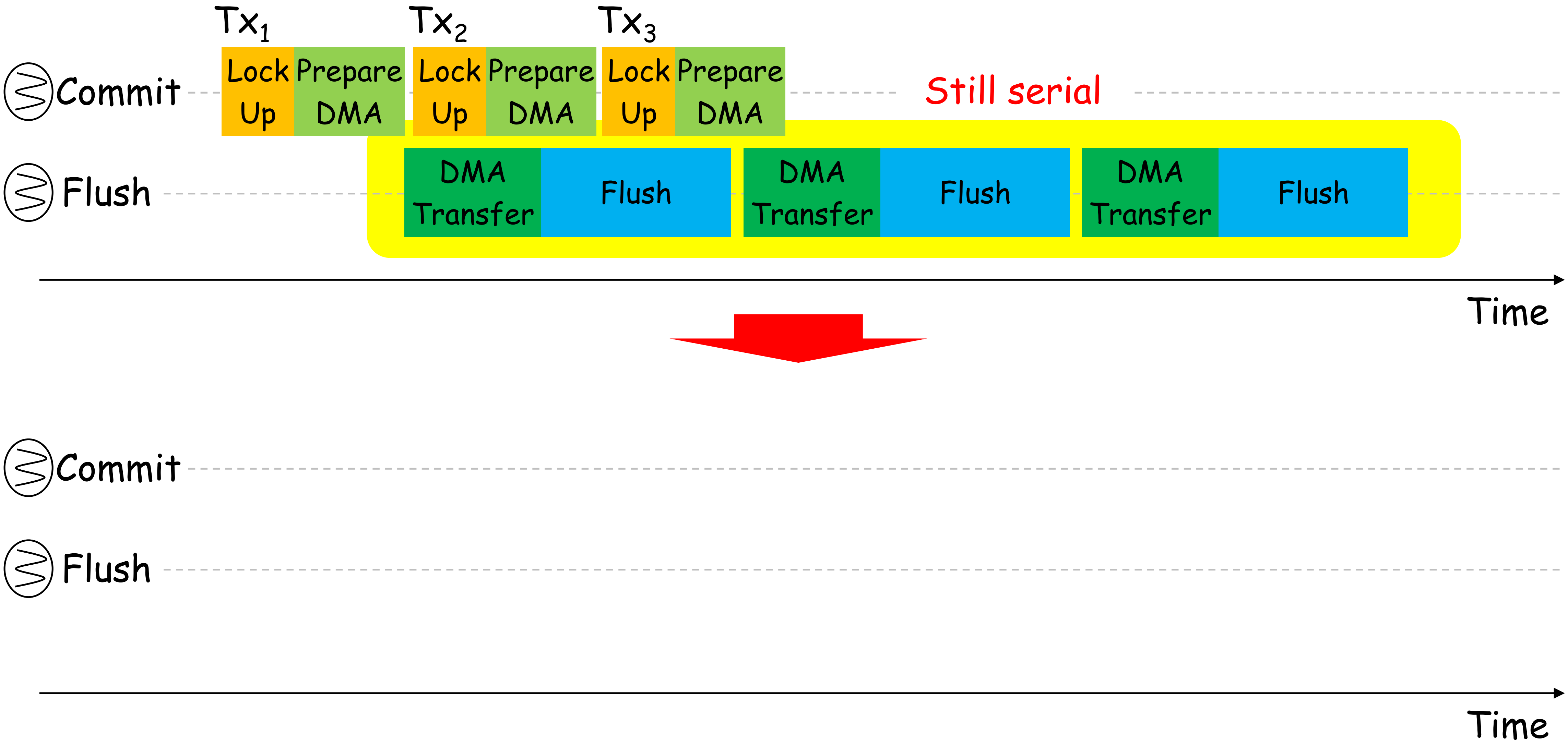
Compound Flush



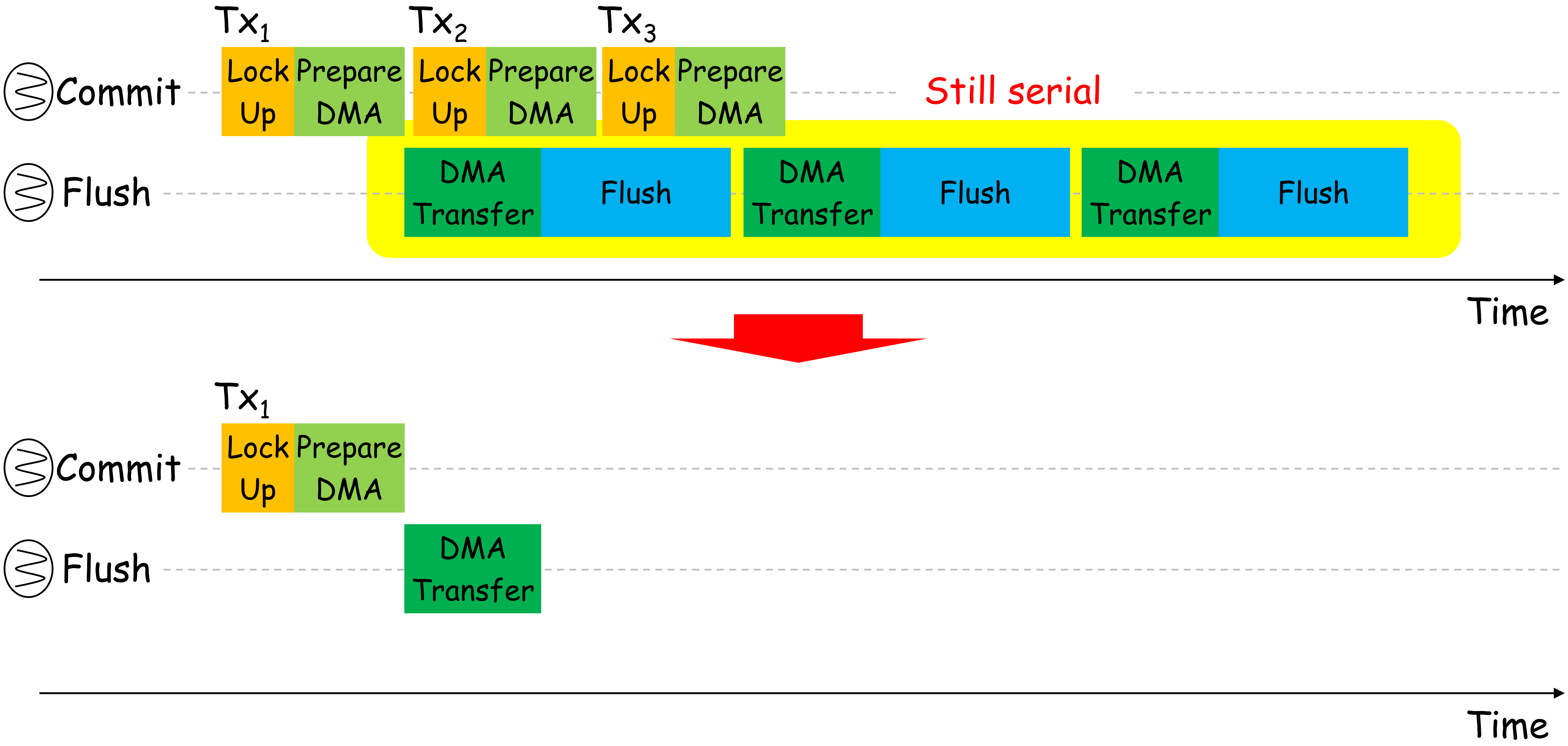
Compound Flush



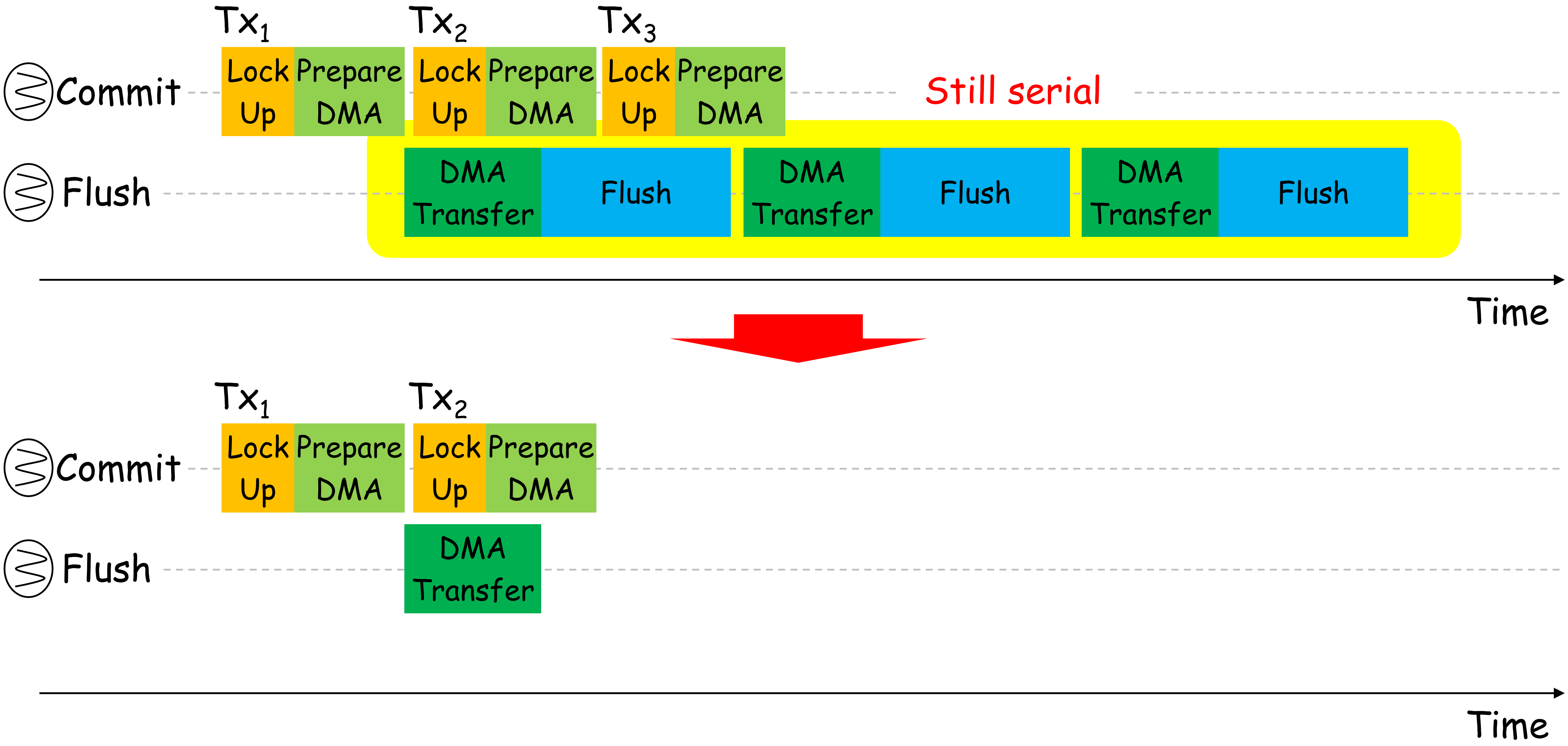
Compound Flush



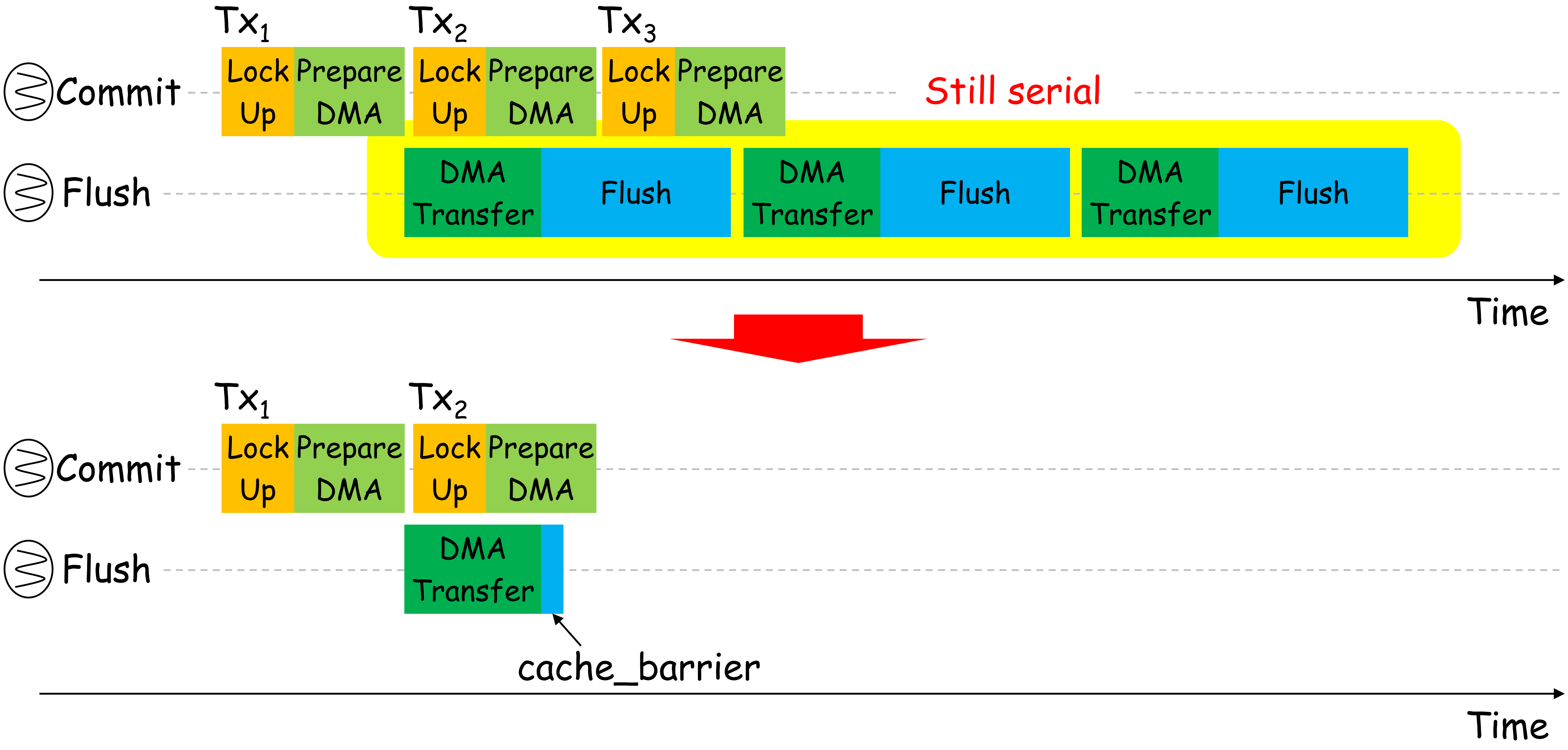
Compound Flush



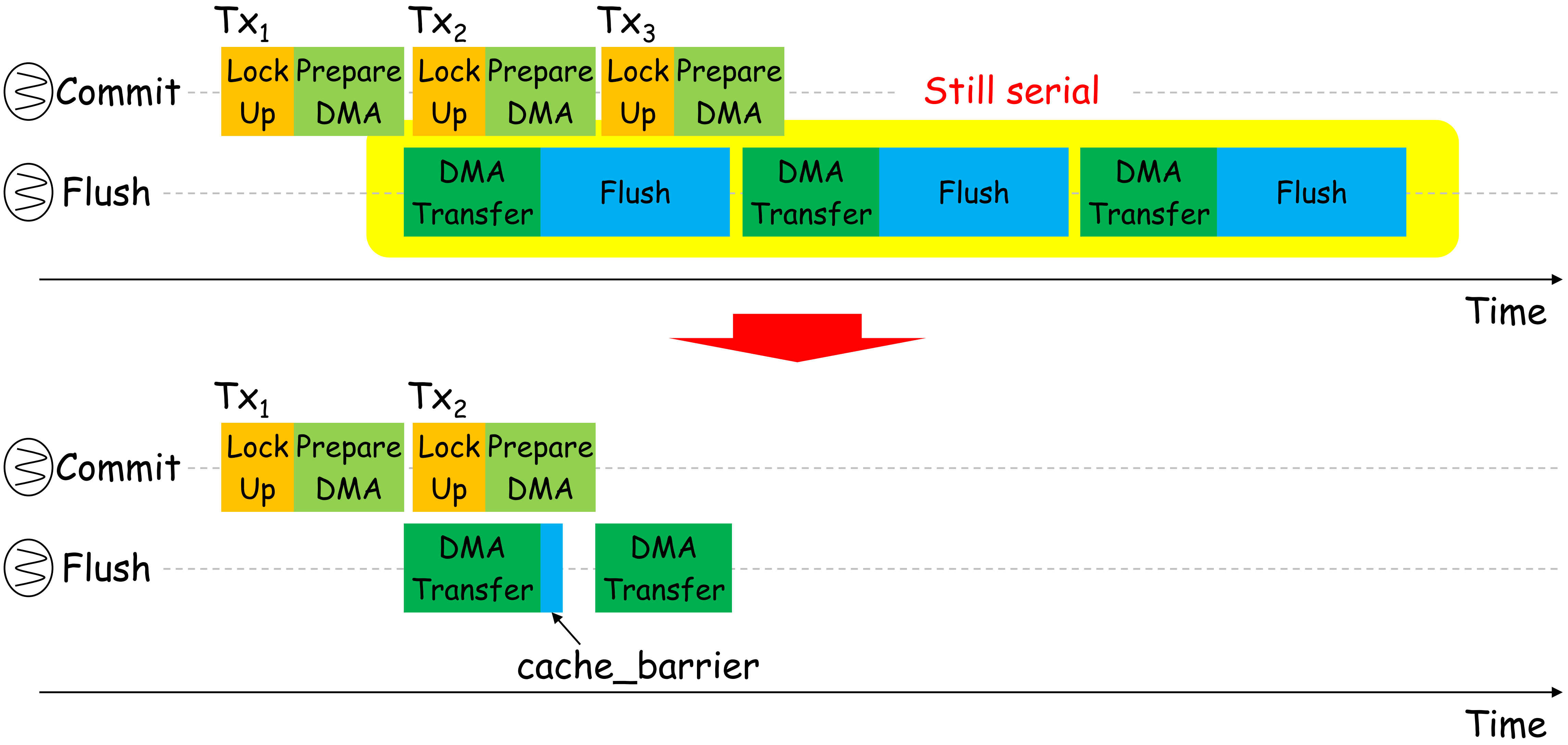
Compound Flush



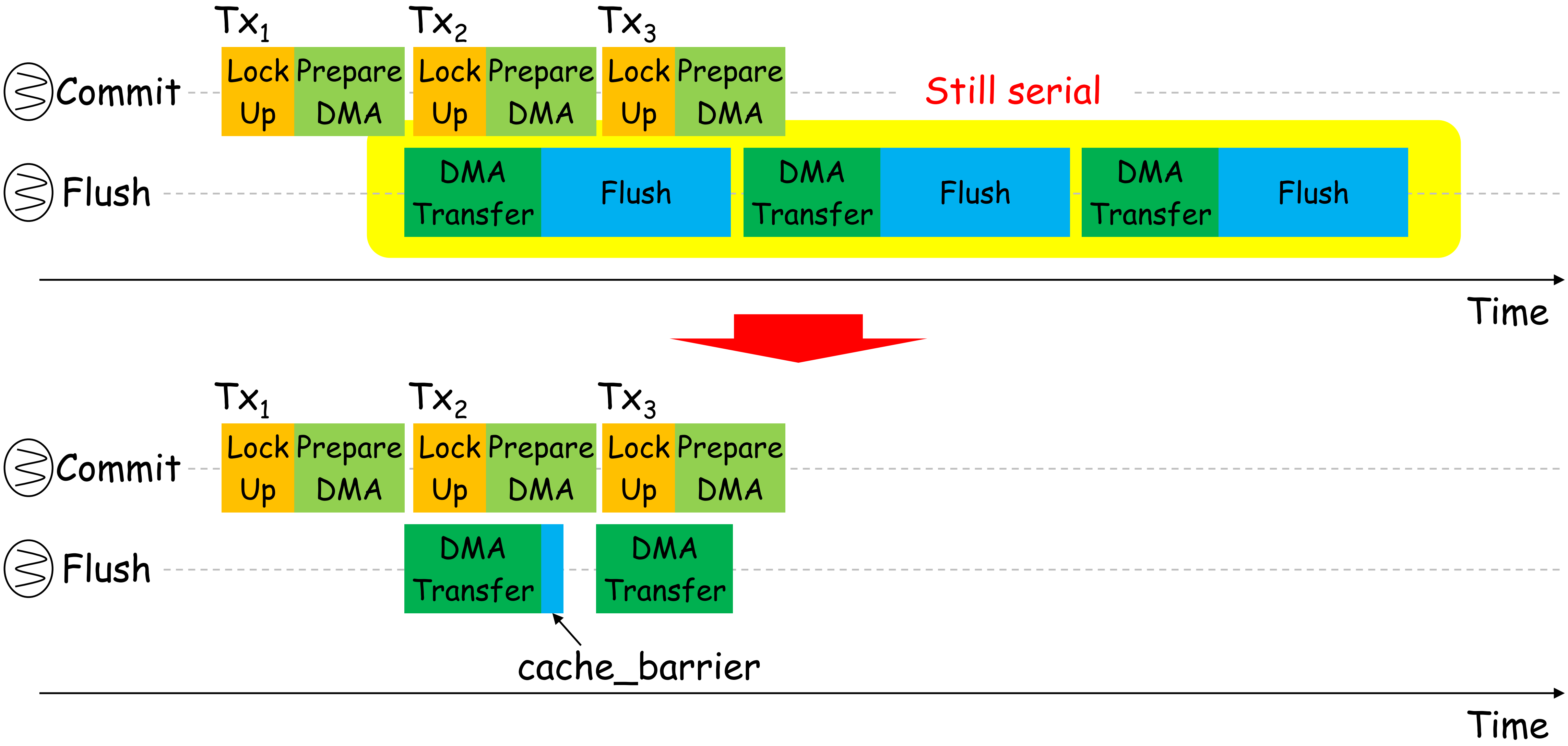
Compound Flush



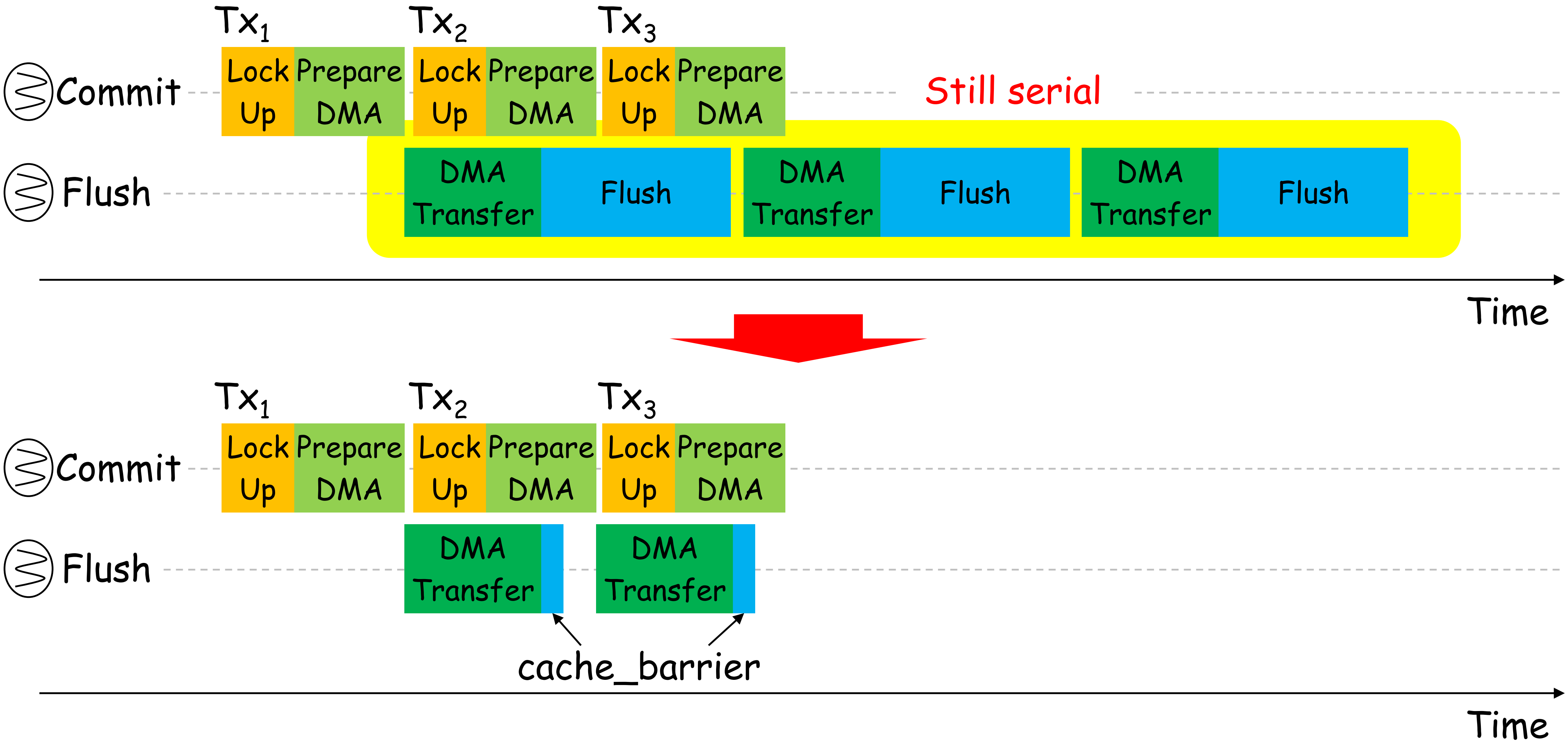
Compound Flush



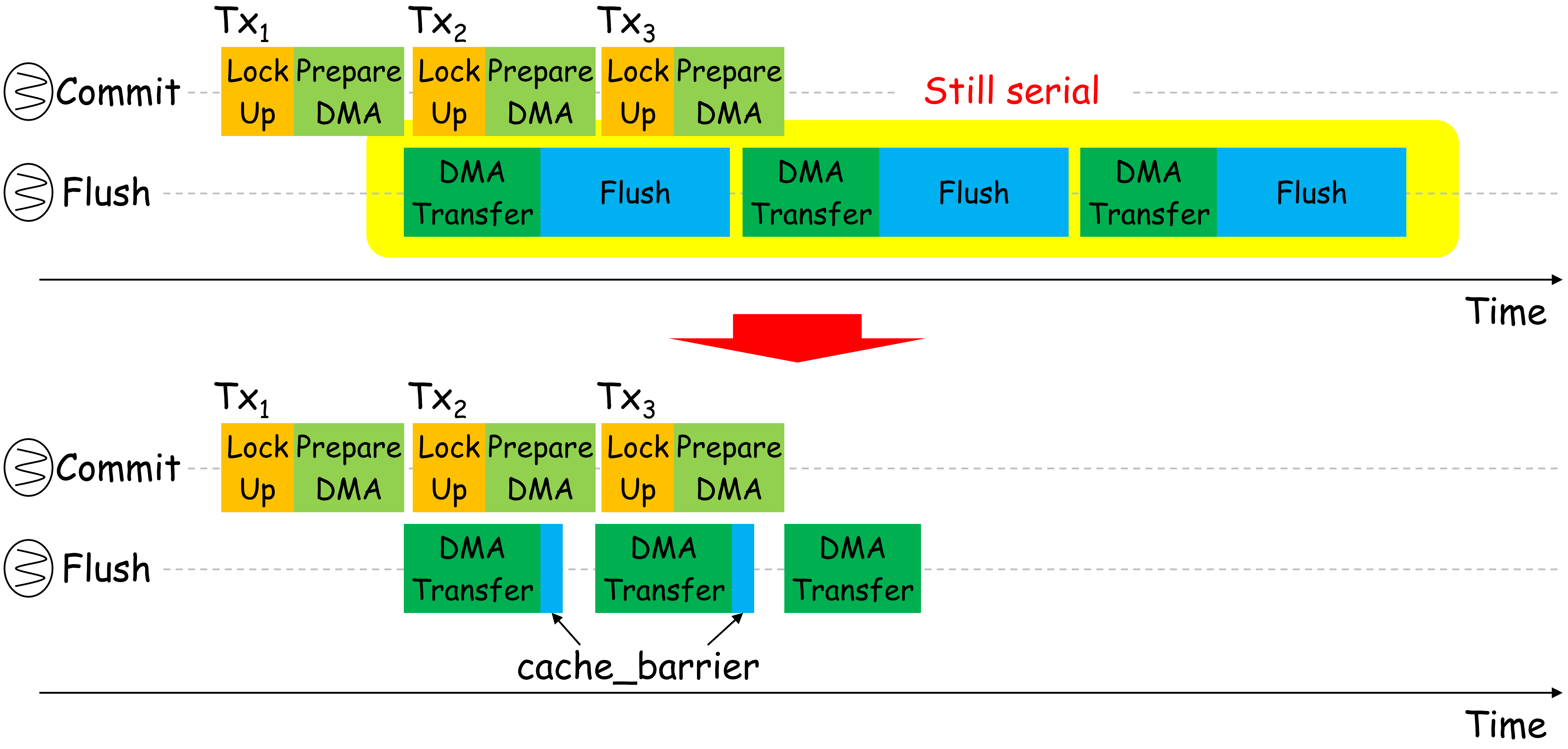
Compound Flush



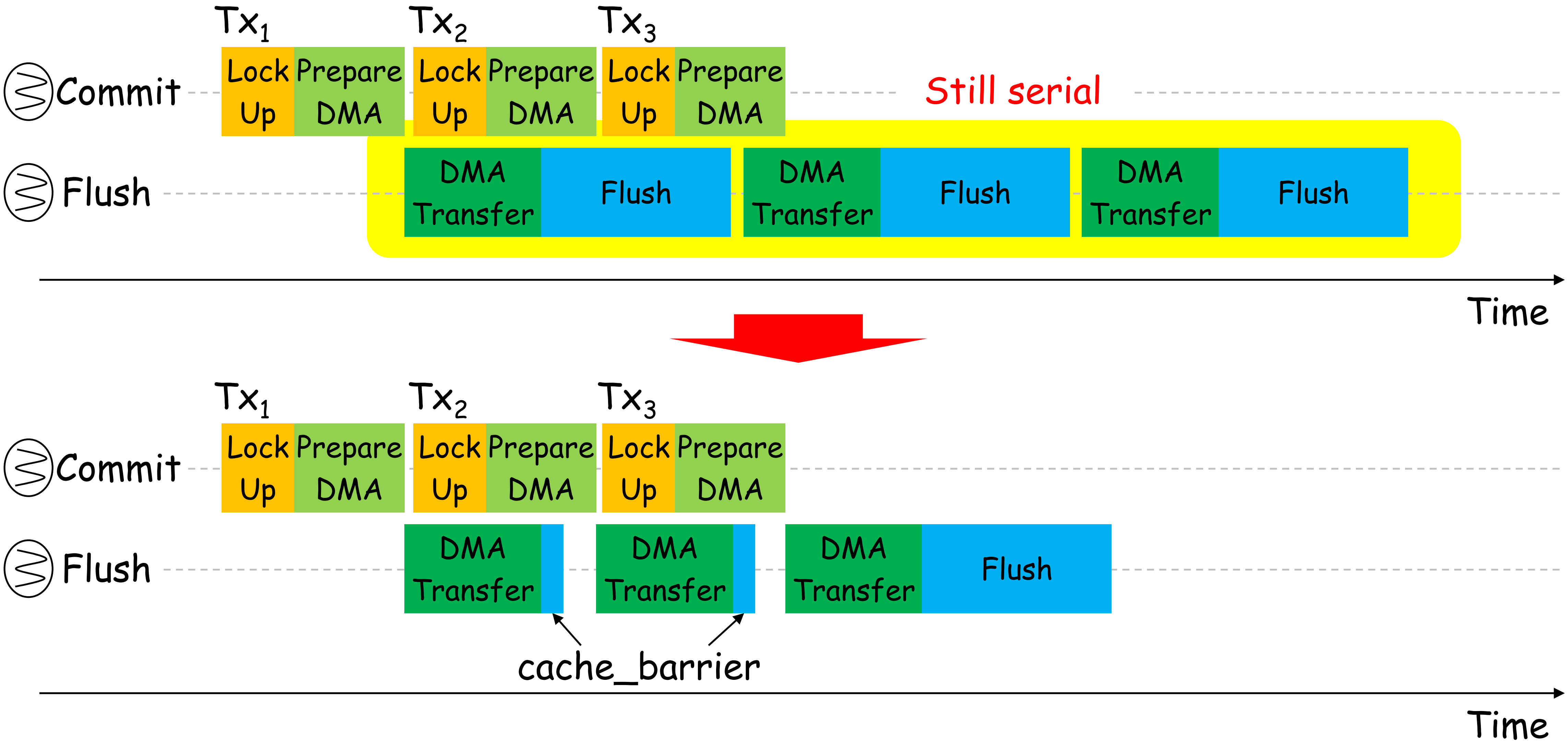
Compound Flush



Compound Flush



Compound Flush



Evaluation

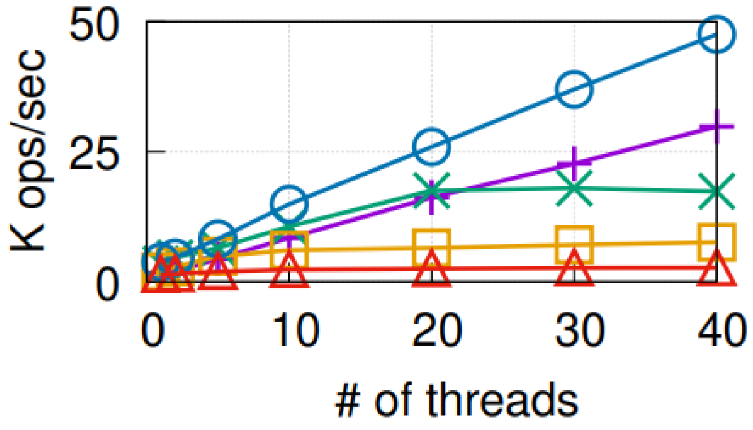
Evaluation Setup

- CPU : Intel Xeon Gold 6320 (2.1 GHz, 2 Socket X 20 core = 40 core)
- Memory : 512GB DRAM
- Storage: Samsung 970 Pro (MLC, NVMe)
- OS (Kernel)
  - CentOS 7.4 (Linux Kernel 5.18.18)
- Filesystem: EXT4, BarrierFS, EXT4 with fast commit, SpanFS, CJFS
- Workloads: Varmail-shared, Varmail-split, Dbench, OLTP-Insert
  - Varmail-shared: Varmail with a shared directory
  - Varmail-split: Varmail with a per-thread directory

Macro Benchmarks

EXT4 + BarrierFS × Fast Commit ⊟ SpanFS ▵ CJFS ⊖

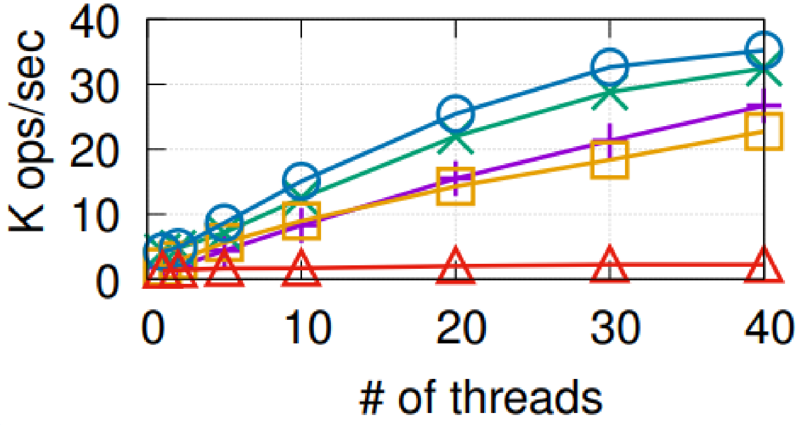
Varmail-shared



Compared to

EXT4: 1.6X
BarFS: 2.7X
FC: 6.3X
SpanFS: 17X

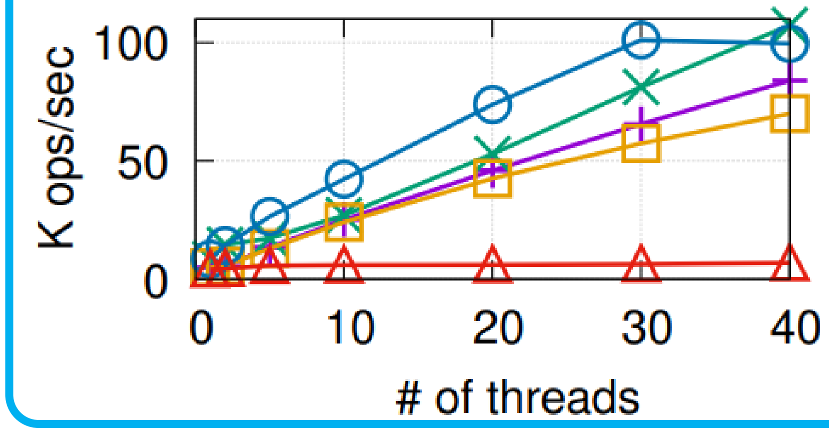
Varmail-split



Compared to

EXT4: 1.3X
BarFS: 1.1X
FC: 1.6X
SpanFS: 16X

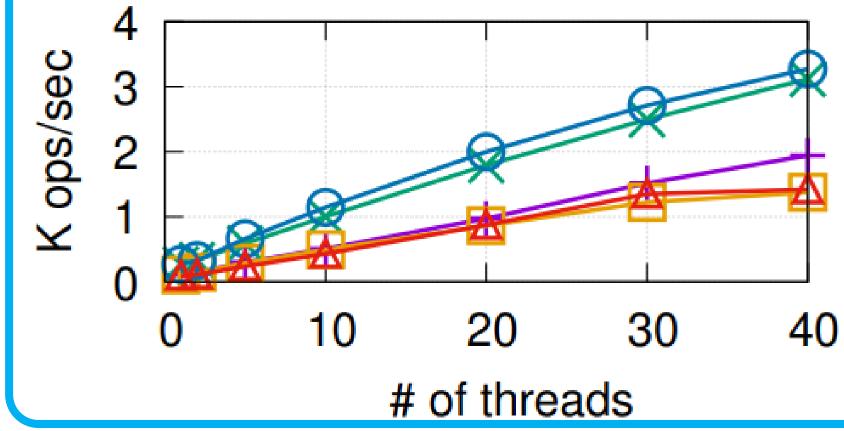
Dbench



Compared to

EXT4: 1.2X
BarFS: 1X
FC: 1.4X
SpanFS: 15X

OLTP-Insert

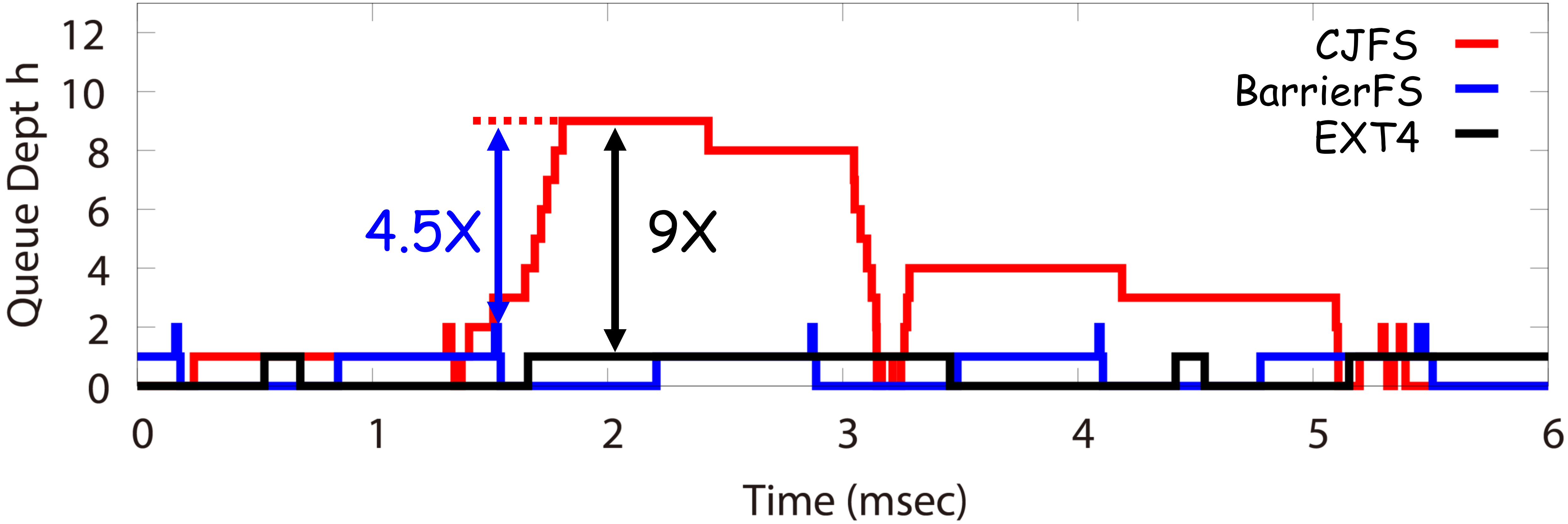


Compared to

EXT4: 1.7X
BarFS: 1X
FC: 2.4X
SpanFS: 2.3X

Command Queue Depth

- Workload: Varmail with 40 threads



Transactions are transferred and flushed concurrently

Conclusion

- We propose CJFS, Concurrent Journaling Filesystem
- CJFS achieves concurrent transaction commit with four techniques
  - Dual Thread Journaling
  - Multi-Version Shadow Paging
  - Opportunistic Coalescing
  - Compound Flush
- CJFS improves the throughput in macro benchmarks
  - Varmail-shared: 1.6X, Varmail-split: 1.3X, Dbench: 1.2X, OLTP-Insert: 1.7X

Question
&
Answer

https://github.com/ESOS-Lab/cjfs