

CONCURRENCY CONTROL

Two-Phase Locking

Part (1)



AMR ELHELW

Schedule Serializability

All Schedules

Serializable Schedules

Serial Schedules

Begin

Lock (X)

Read (X)

Write (X)

Unlock (X)

Commit

T1

T2

Time

Begin

Lock (X)

Read (X)

Write (X)

Unlock (X)

Commit

Begin

Lock (X)

Read (X)

Write (X)

Unlock (X)

Commit



X: Locked by T1

Begin

Lock (X)

Read (X)

Write (X)

Unlock (X)

Commit

Amr Elhelw's

TECH

VAULT

- **S**hared lock (**S-Lock**)
 - a.k.a. *Read Lock*
- e**X**clusive lock (**X-lock**)
 - a.k.a. *Write Lock*

Compatibility Matrix

	Shared lock	Exclusive lock
Shared lock	✓	✗
Exclusive lock	✗	✗

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

Unlock (A)

T1

T2

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

Unlock (A)

X-Lock (A)

Write (A)

Unlock (A)

T1

T2

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

Unlock (A)

X-Lock (A)

Write (A)

Unlock (A)

Time

T1

T2

Time

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

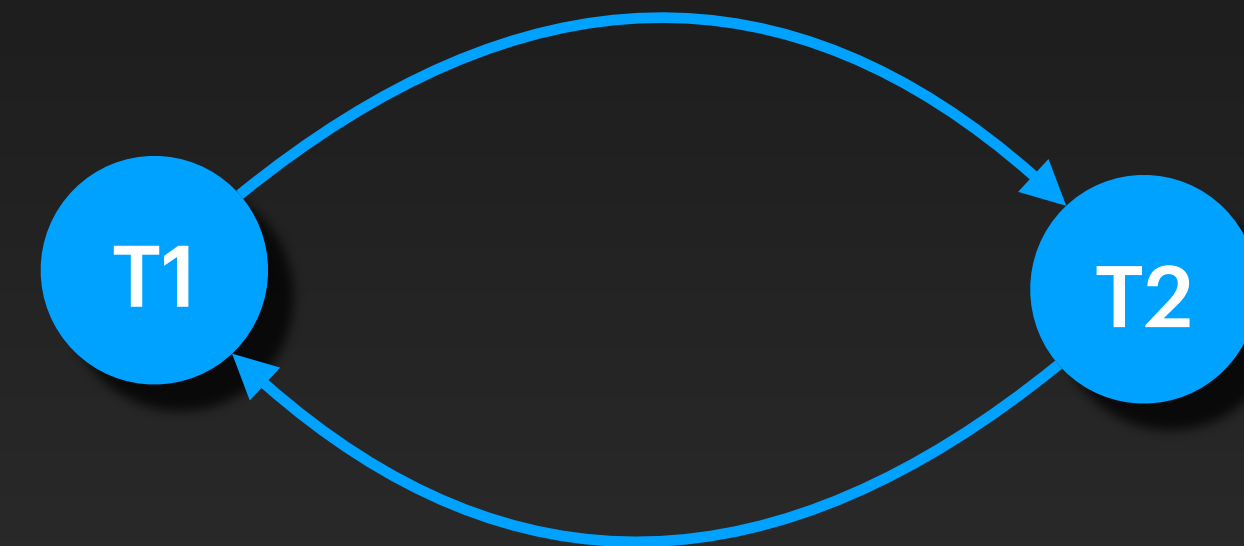
Unlock (A)

X-Lock (A)

Write (A)

Unlock (A)

Dependency Graph



Not serializable!

T1

T2

Time

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

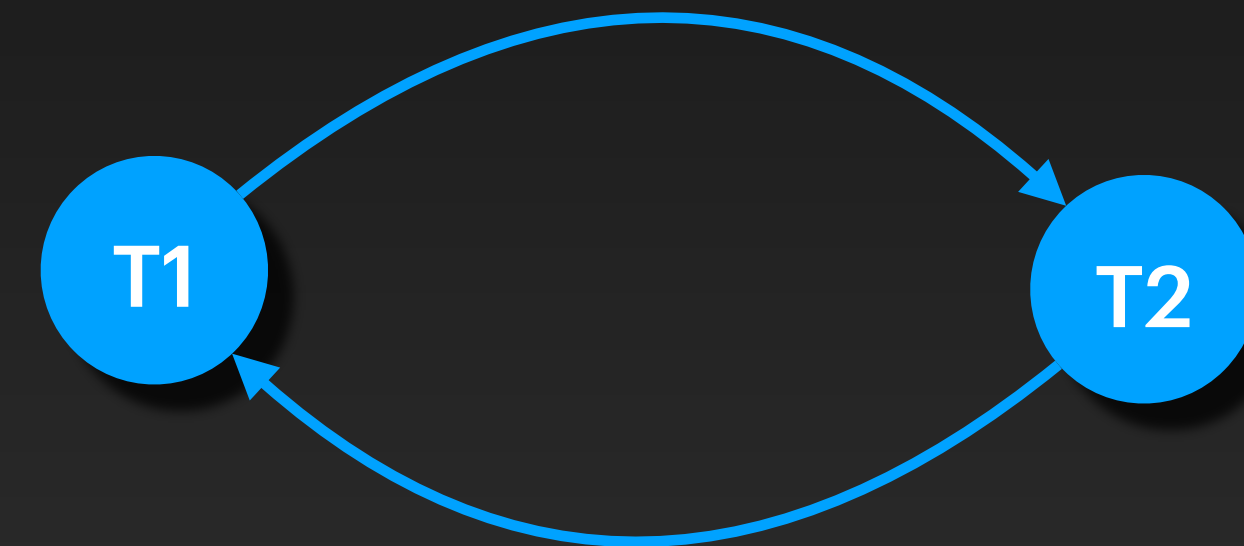
Unlock (A)

X-Lock (A)

Write (A)

Unlock (A)

Dependency Graph



T1

X-Lock (A)

Write (A)

Unlock (A)

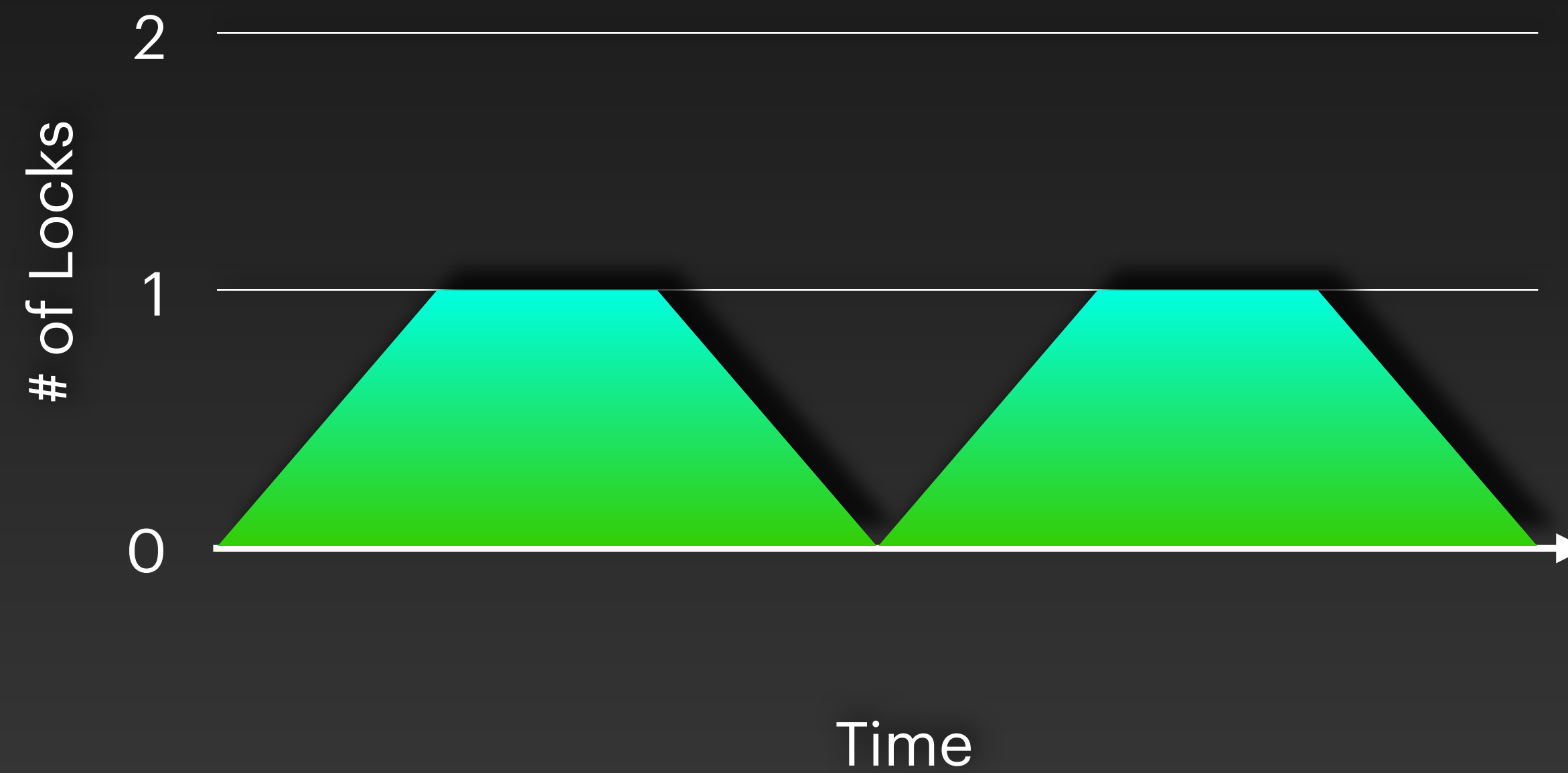
...

S-Lock (A)

Read (A)

Unlock (A)

Time



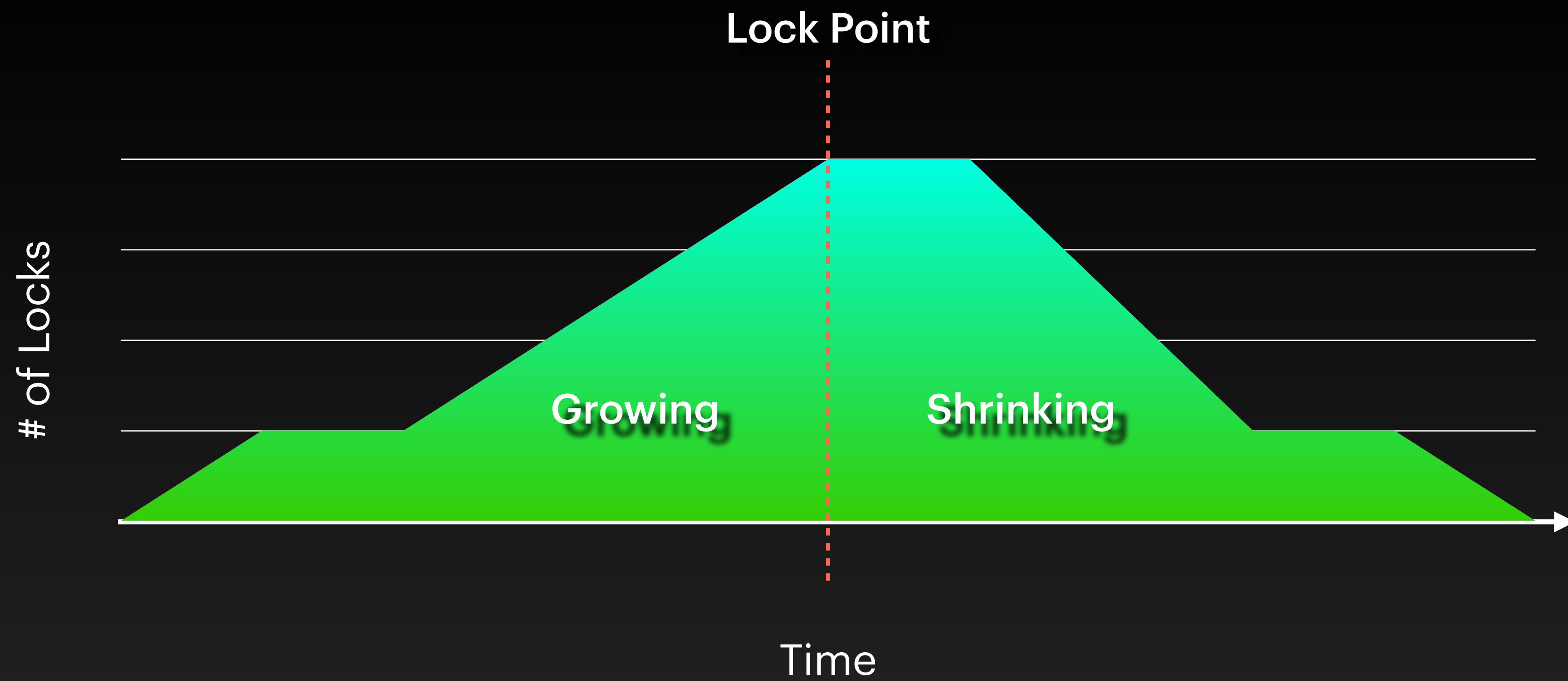
Two-Phase Locking (2PL)

- Phase 1 - Growing

- Transaction can acquire/upgrade locks, without releasing/downgrading any locks

- Phase 2 - Shrinking

- Transaction can release/downgrade locks. It cannot acquire/upgrade locks.



2PL



Not 2PL

X-Lock (A)

Write (A)

Unlock (A)

...

S-Lock (A)

Read (A)

Unlock (A)

Not 2PL

X-Lock (A)

Write (A)

...

Read (A)

Unlock (A)

2PL

T1

T2

Time

X-Lock (A)

X-Lock (B)

Read (A)

Write (A)

Unlock (A)

Write (B)

Unlock (B)

Commit

S-Lock (A)

Read (A)

Unlock (A)

Commit

Dirty Read

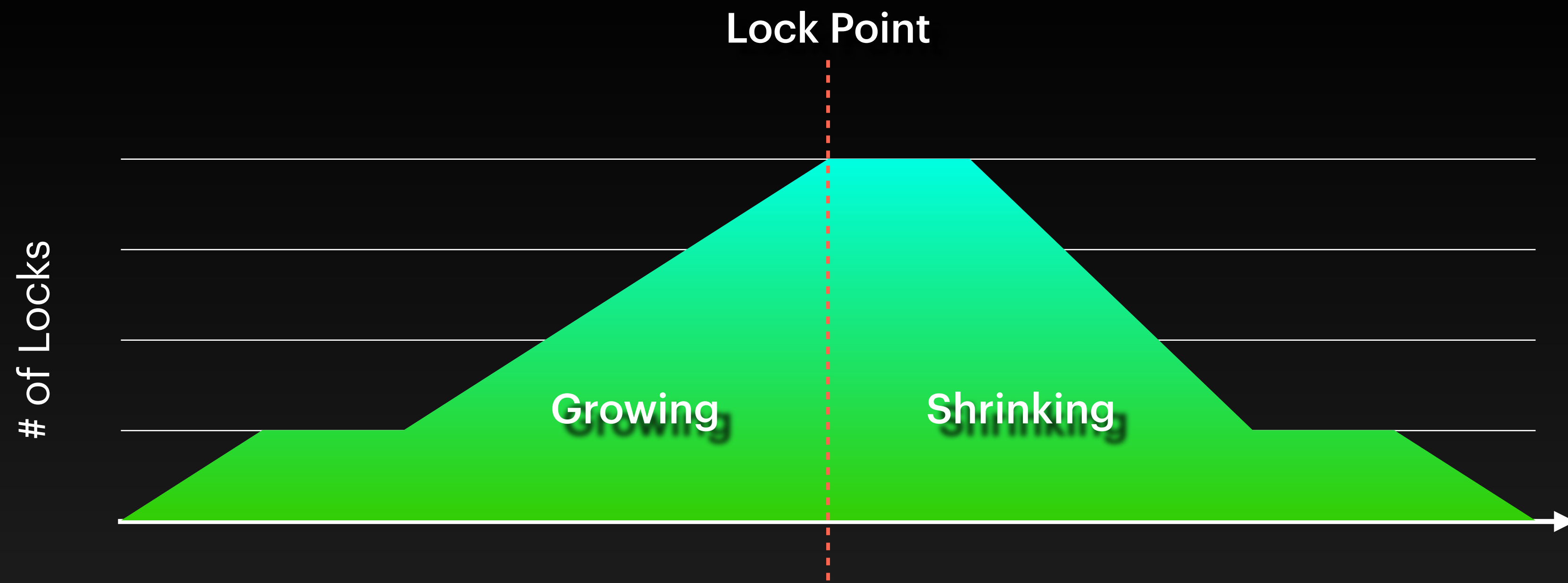
- Valid 2PL
- Cascading Rollback

Strict 2PL

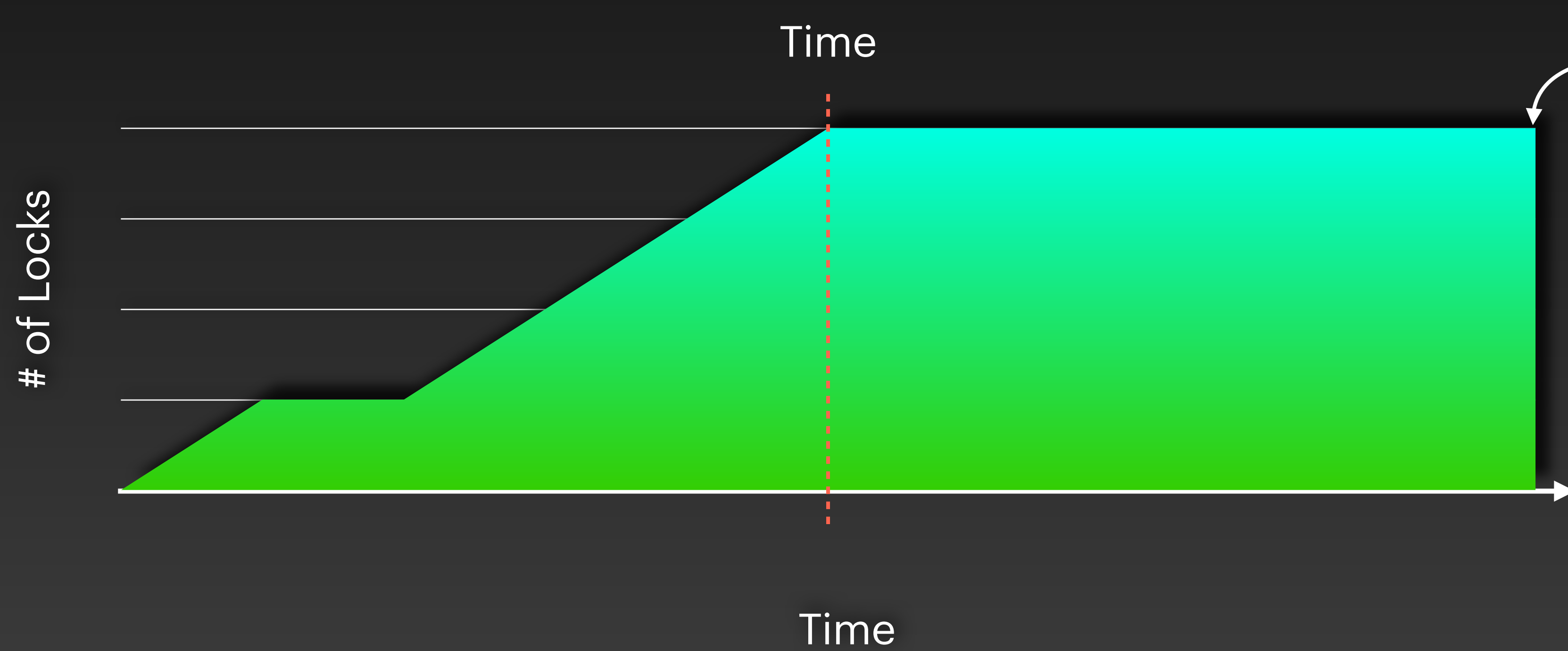
- Transaction can only release **exclusive (write) locks** after it has ended (committed or rolled back)
- Prevents dirty reads
- Prevents cascading rollbacks
 - Produces **Strict** schedules

Strong Strict (Rigorous) 2PL

- Transaction can only release **ALL locks** after it has ended (committed or rolled back)
- **More restrictive** than Strict 2PL
- **Easier to implement** than Strict 2PL (and more common)



2PL



Strong Strict 2PL

T1

T2

Time

X-Lock (A)

Read (A)

Write (A)

X-Lock (B)

Write (B)

Commit

Unlock (A)

Unlock (B)

S-Lock (A)

Read (A)

Commit

Unlock (A)