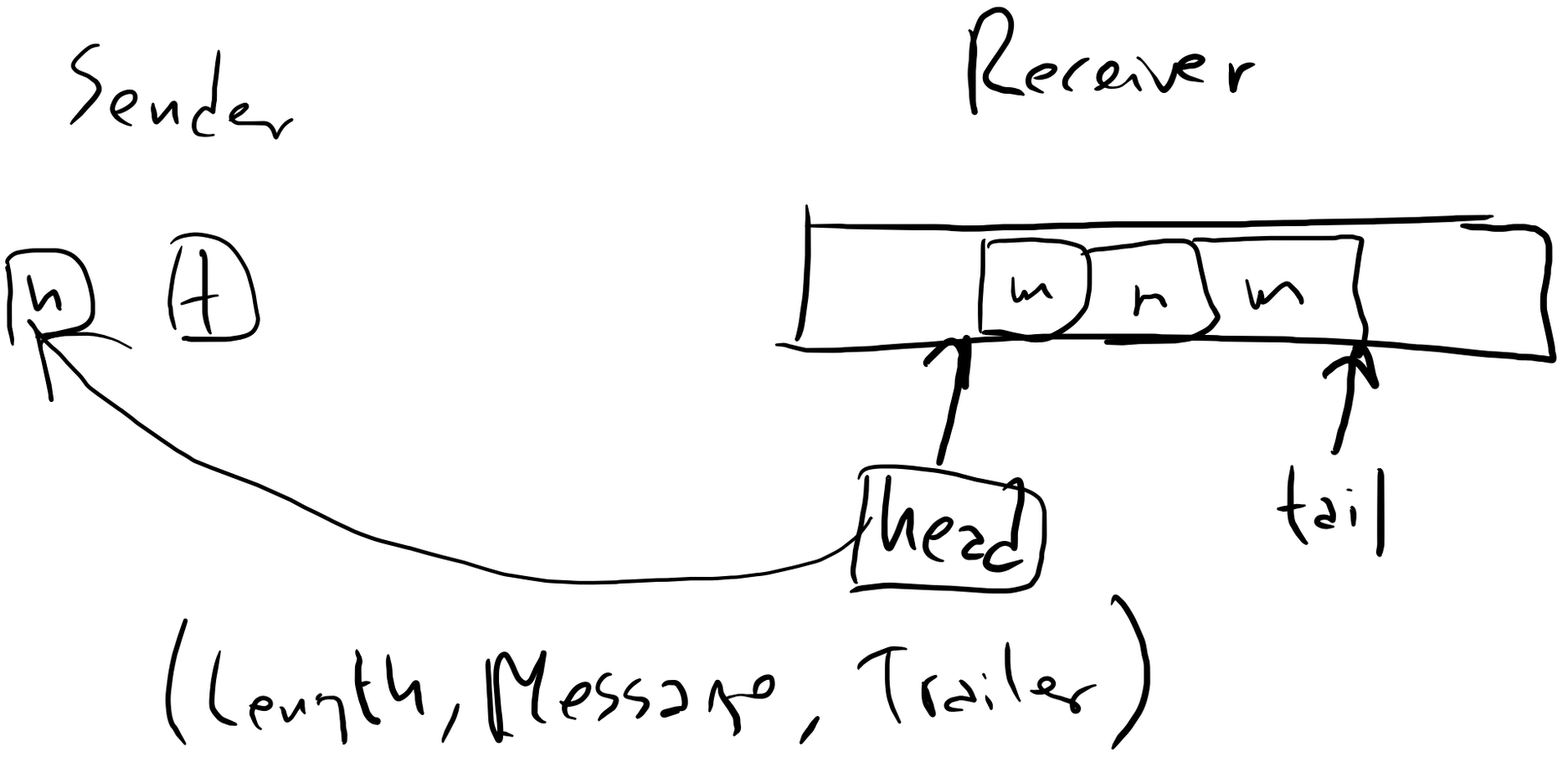
CS244b — FaRM

Learning Goals:

- µsec-scale systems
- High-end hardware capabilities
  kernel-bypass, RDMA, NV DRAM
- More practice reasoning about
  transaction serializability & recovery


Sender
Receiver
h
t
m
m
m
head
tail
(Length, Message, Trailer)

FaRM API

- ACID transactions
  - On global shared mem.
    Begin, Commit, Alloc, Free
    Read, Write
- Lock-free single object reads
- Function shipping

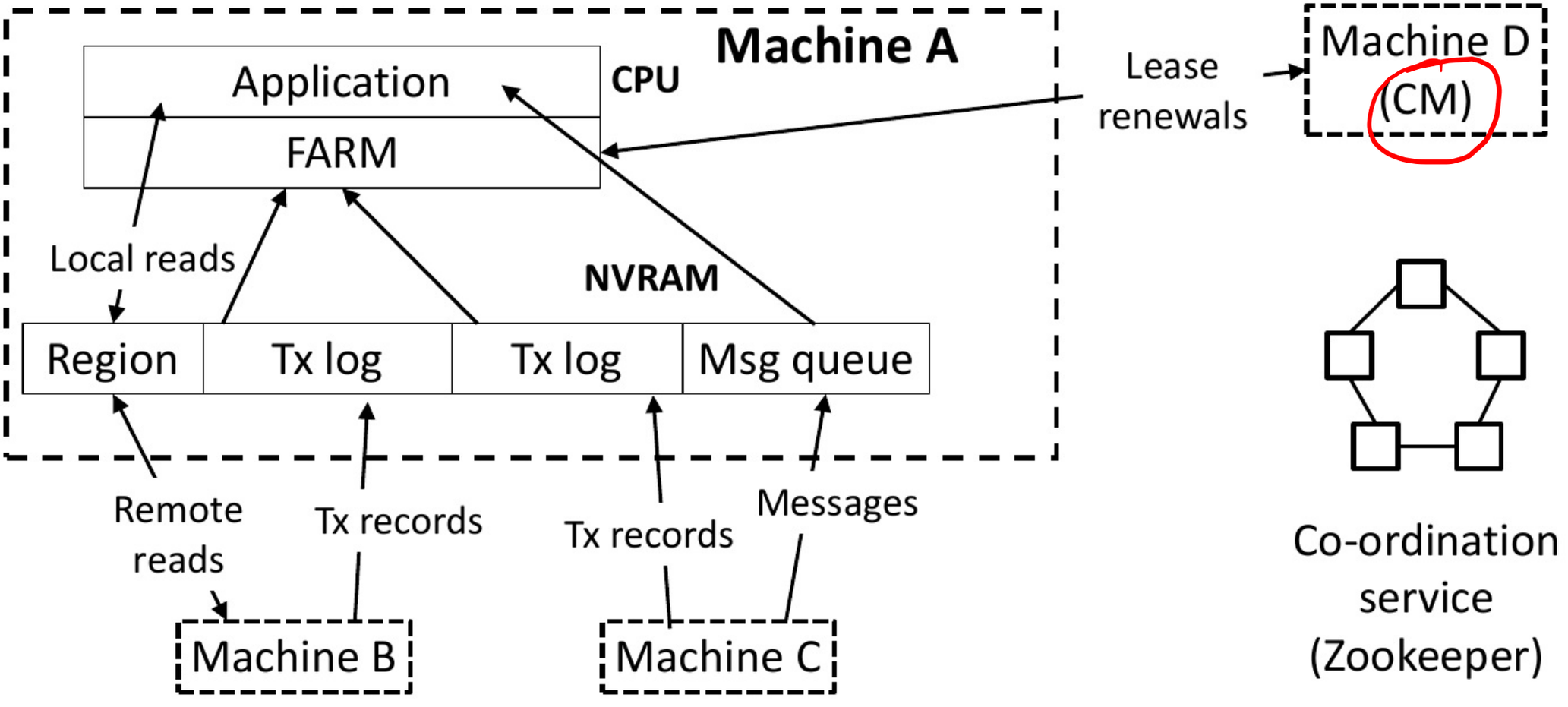


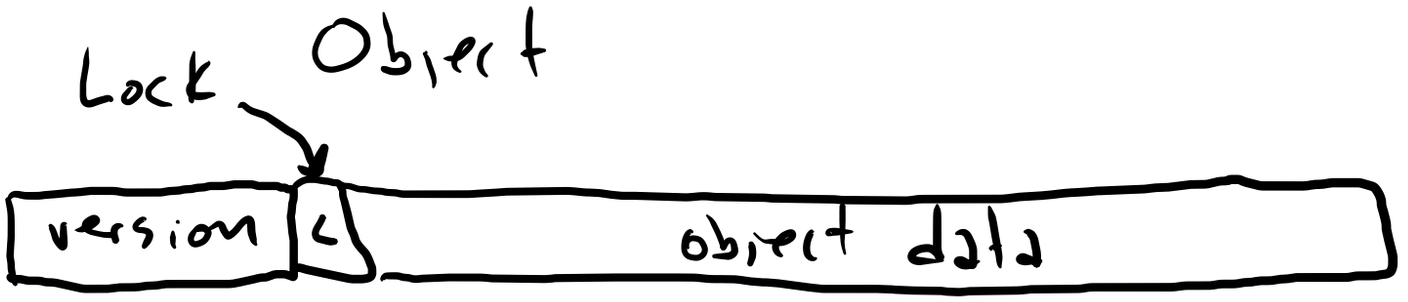
Figure 3. FaRM architecture

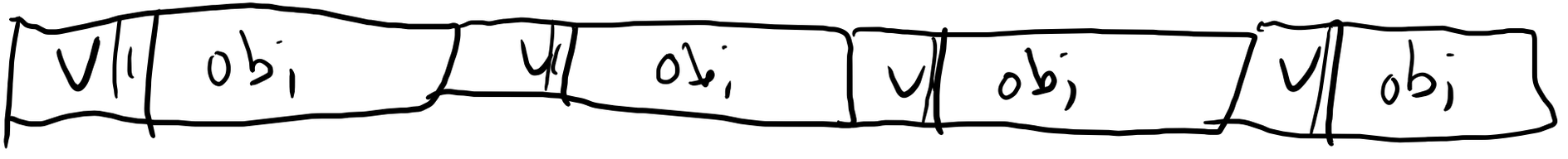
Set of machines S, 2GiB regions

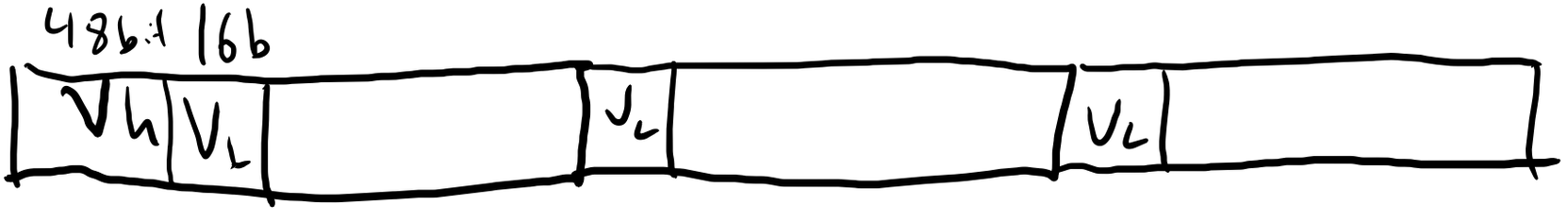
⟨C, S, F, CM⟩

```
void T1()
{
   if (x2 == 0)
      x1 = 1;
}
```

```
void T2()
{
   if (x1 == 0)
      x2 = 1;
}
```


Lock
Object
version
L
object data


V
obj
V
obj
V
obj
V
obj


48bit 16b
Vh
VL
VL
VL

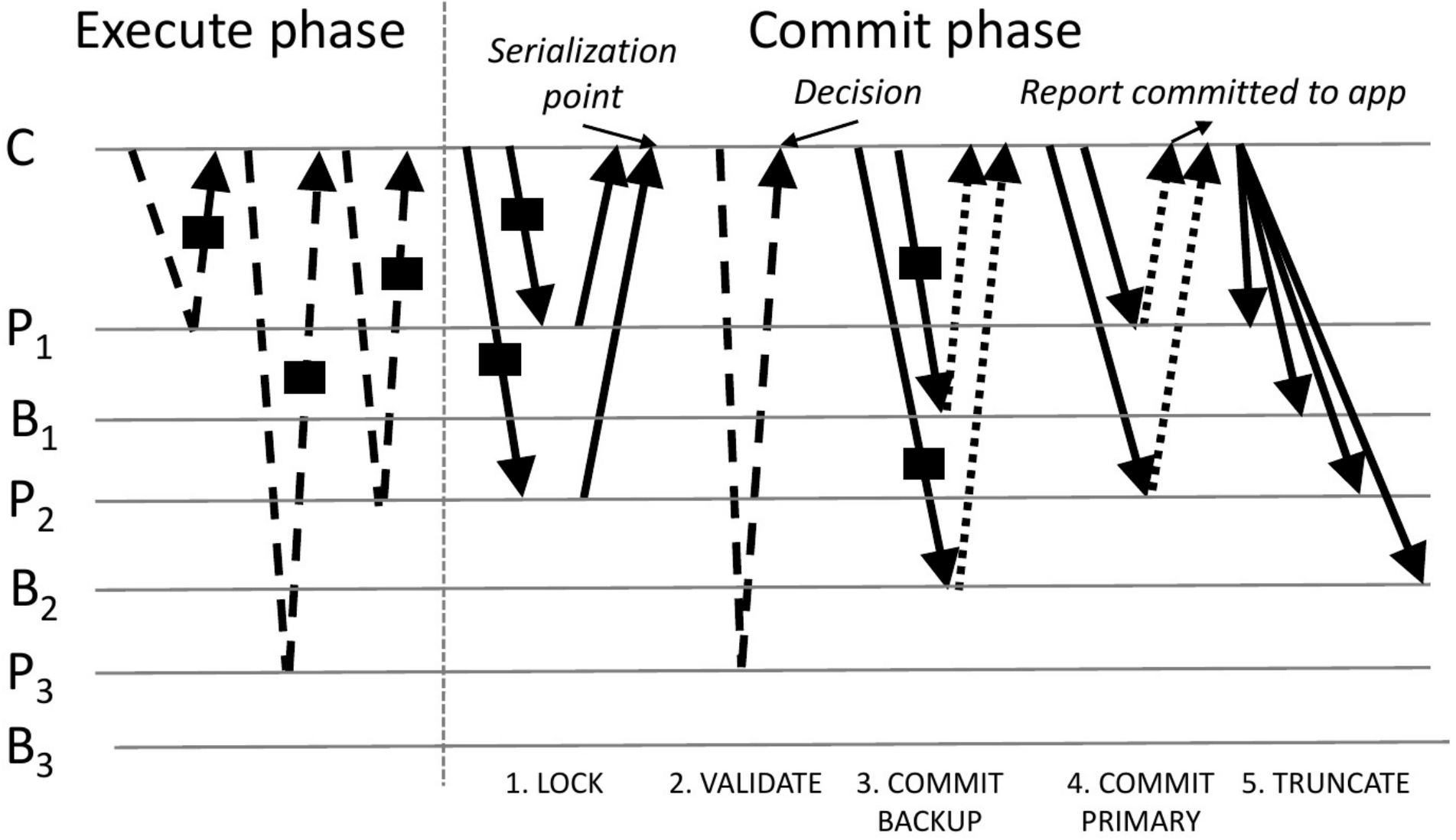


Figure 4. FaRM commit protocol with a coordinator C, primaries on P_1, P_2, P_3, and backups on B_1, B_2, B_3. P_1 and P_2 are read and written. P_3 is only read. We use dashed lines for RDMA reads, solid ones for RDMA writes, dotted ones for hardware acks, and rectangles for object data.

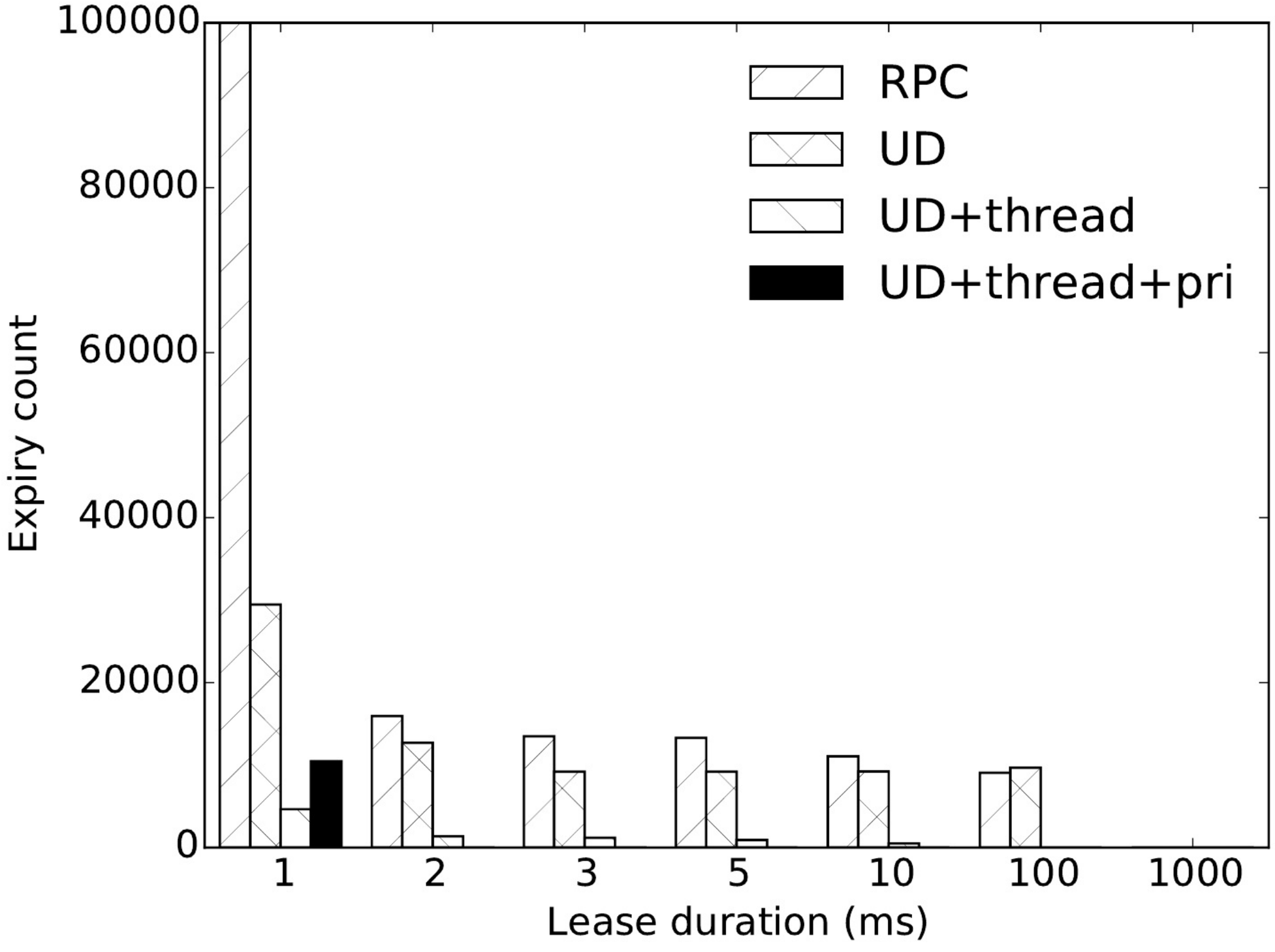


Figure 16. False positives with different lease managers

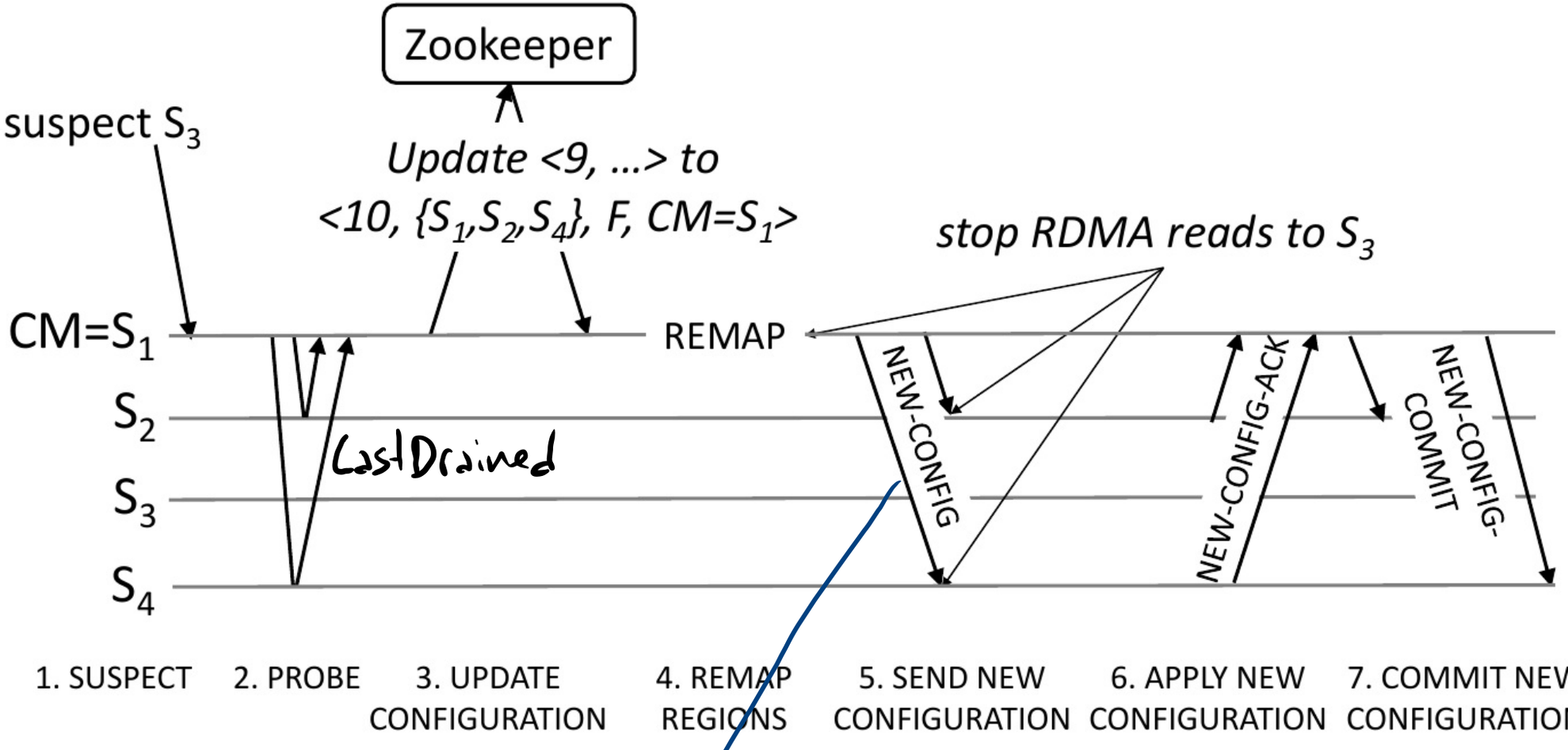


Figure 5. Reconfiguration

LastPrimaryChange[r]
LastReplicaChange[r]

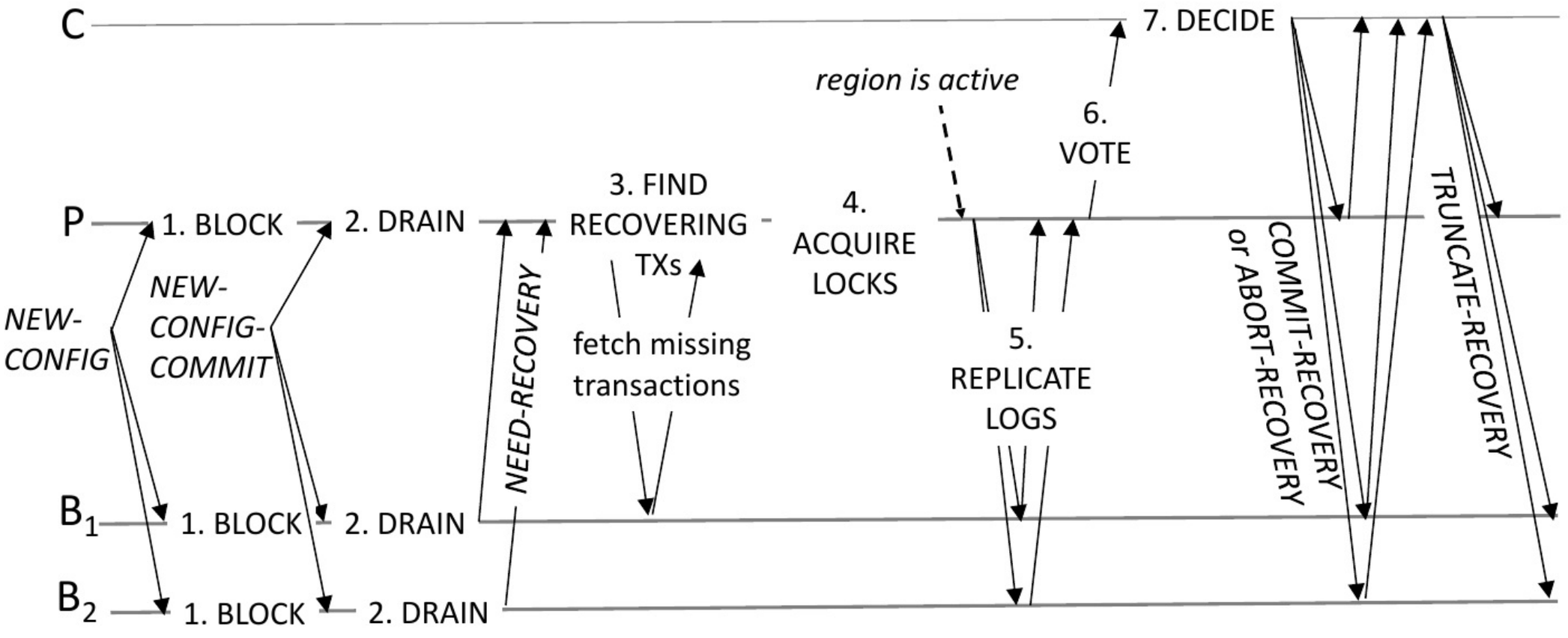


Figure 6. Transaction state recovery showing a coordinator C, primary P, and two backups B_1 and B_2

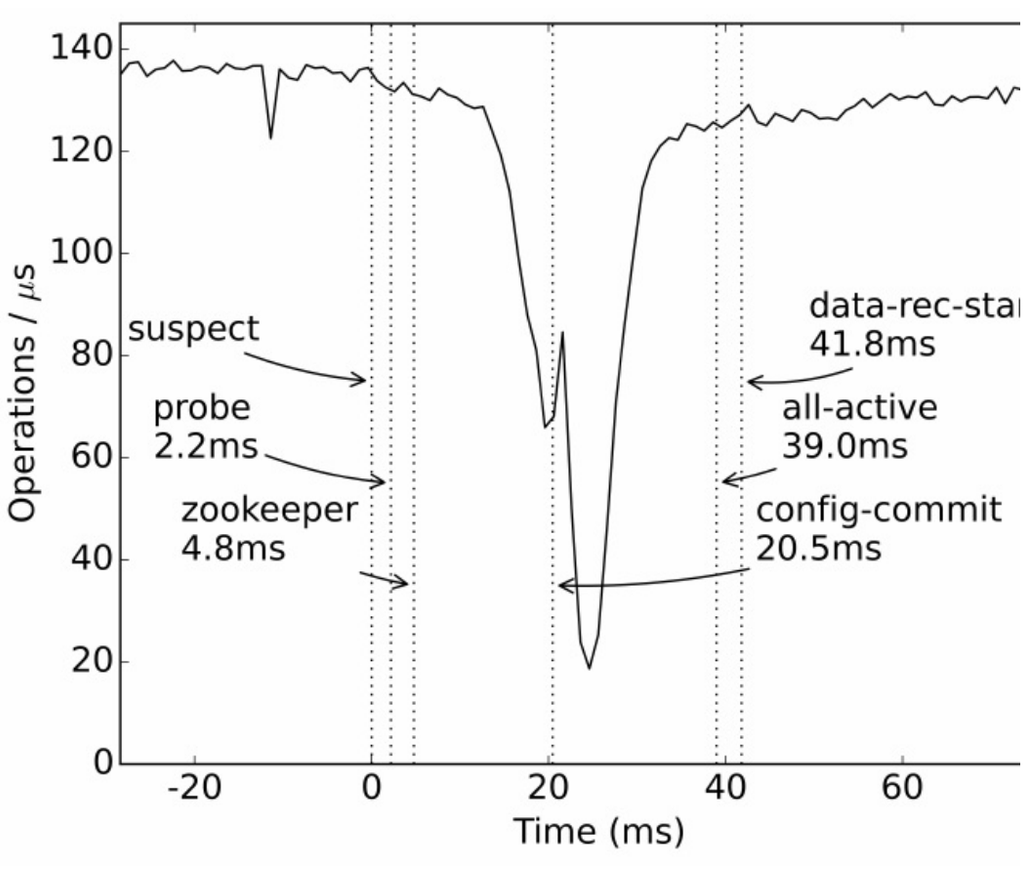
Vote

- commit-primary
- commit-backup
- lock
- abort

- truncated
- unknown

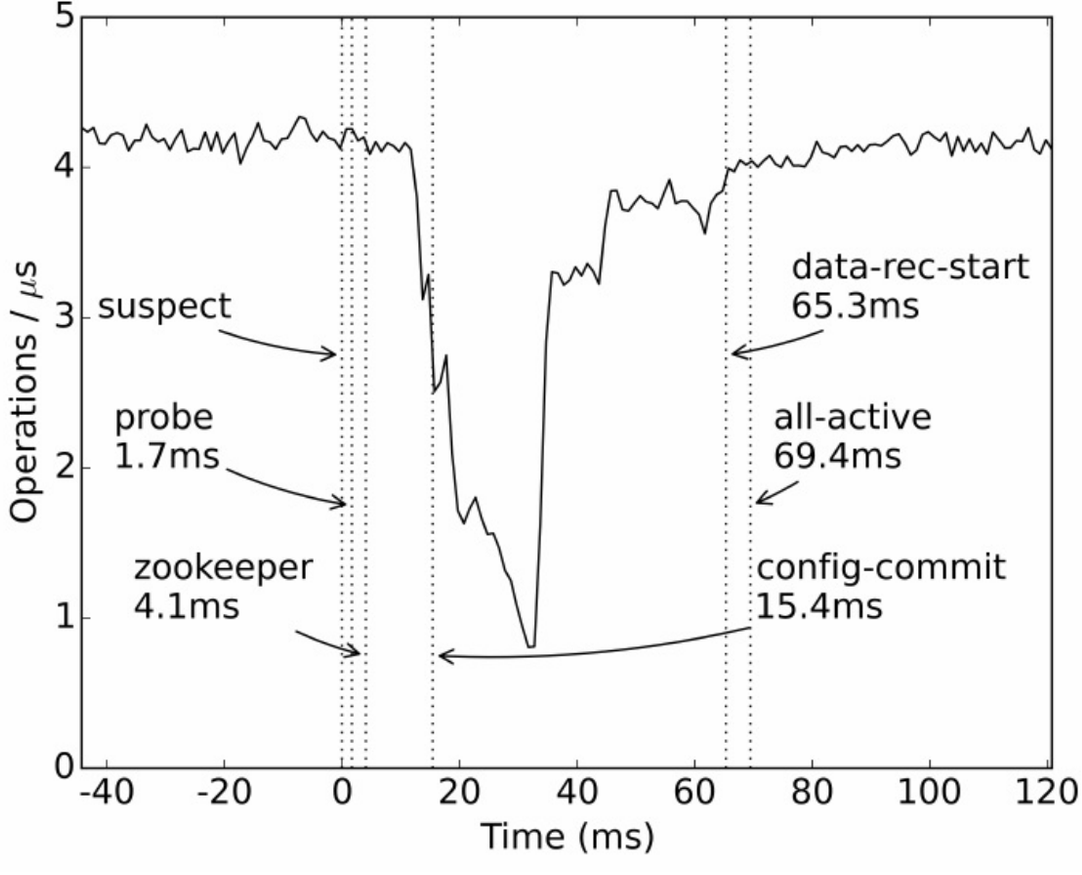
When to commit?

- If you see one commit Primary
- One commit-backup and <u>no</u> abort/unknown



(a) Time to full throughput

Fig. 9 (TATP)



(a) Time to full throughput

Fig. 10 (TPC-C)

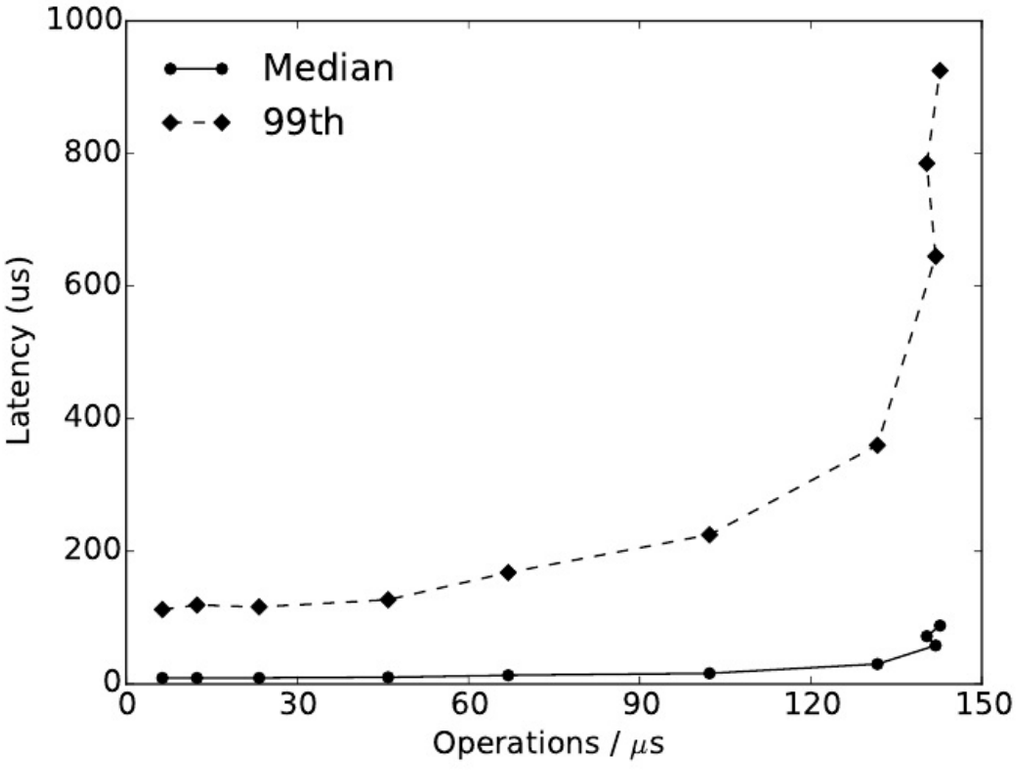


Figure 7. TATP performance

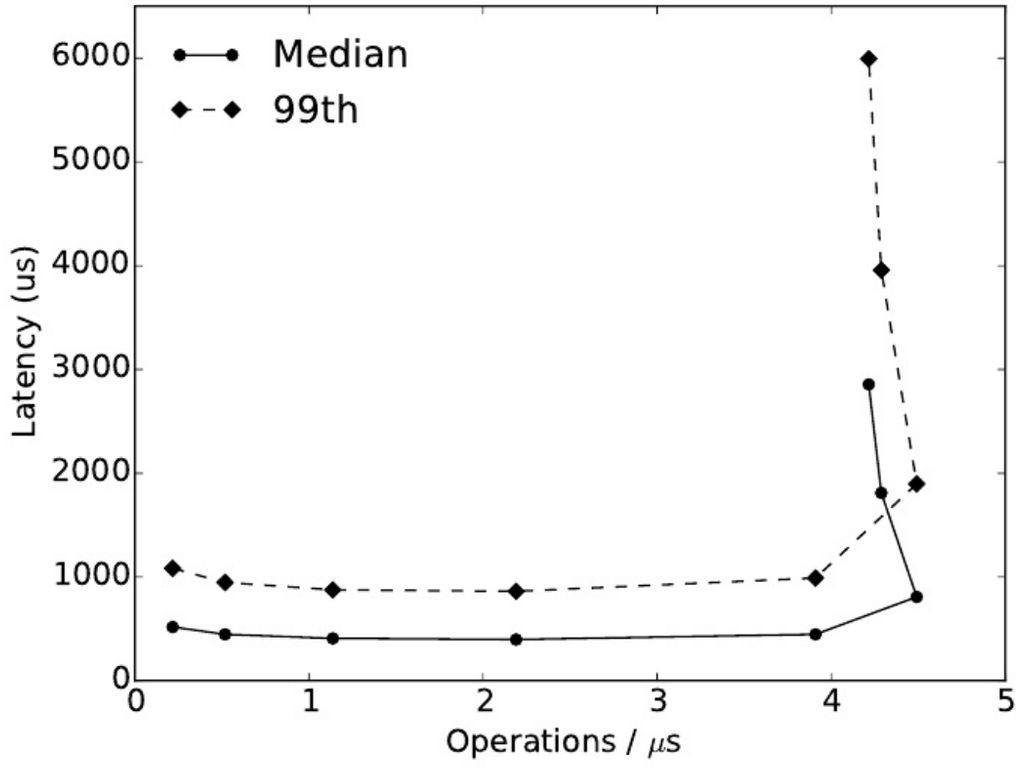


Figure 8. TPC-C performance