Student Presentation

Just say NO to Paxos Overhead: Replacing Consensus with Network Ordering

Benjamin Kuykendall

COS 518: Advanced Computer Systems

February 18, 2019

Problem statement

Consensus is slow.

Paxos achieves consensus with a 2-message delay.

This is optimal when the network is asynchronous [Lam06b].

Previous solutions

Asynchronous unordered:

- Paxos [Lam98]
- Raft [OO14]

Usually reliable & ordered network:

- Fast Paxos [Lam06a]
- Speculative Paxos [PLL+15]

Totally ordered atomic broadcast:

- Trivial [BJ87]

Key idea

Separate concerns of ordering and reliable delivery.

Ordered unreliable multicast:

- Network-Ordered Paxos

To get OUM: add sequence numbers using new hardware.

To get NOPaxos: execute non-dropped operations in order.

Key challenges

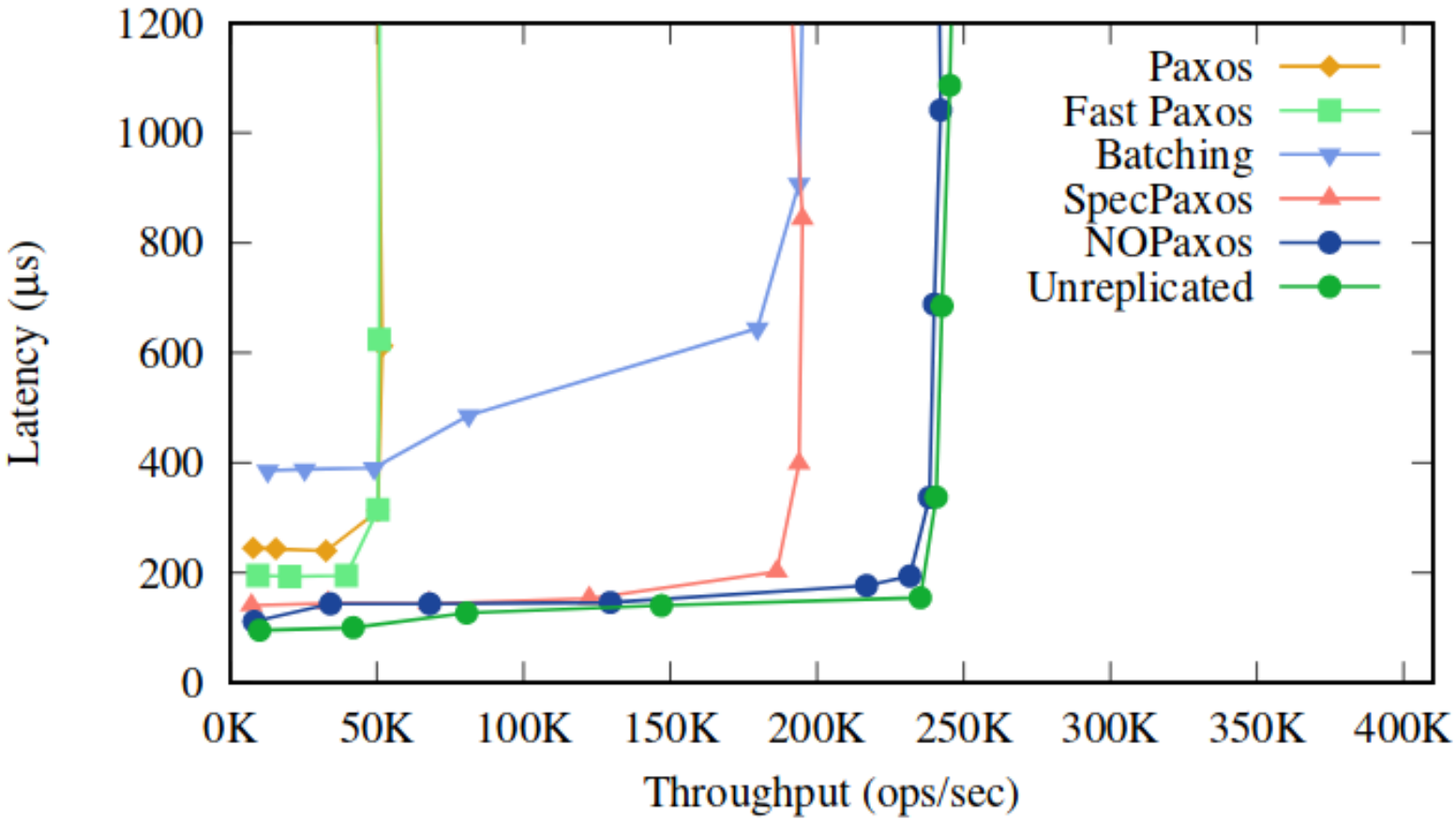
Build a fast sequencer.

Detect dropped messages and agree not to execute.

Deal with failures.

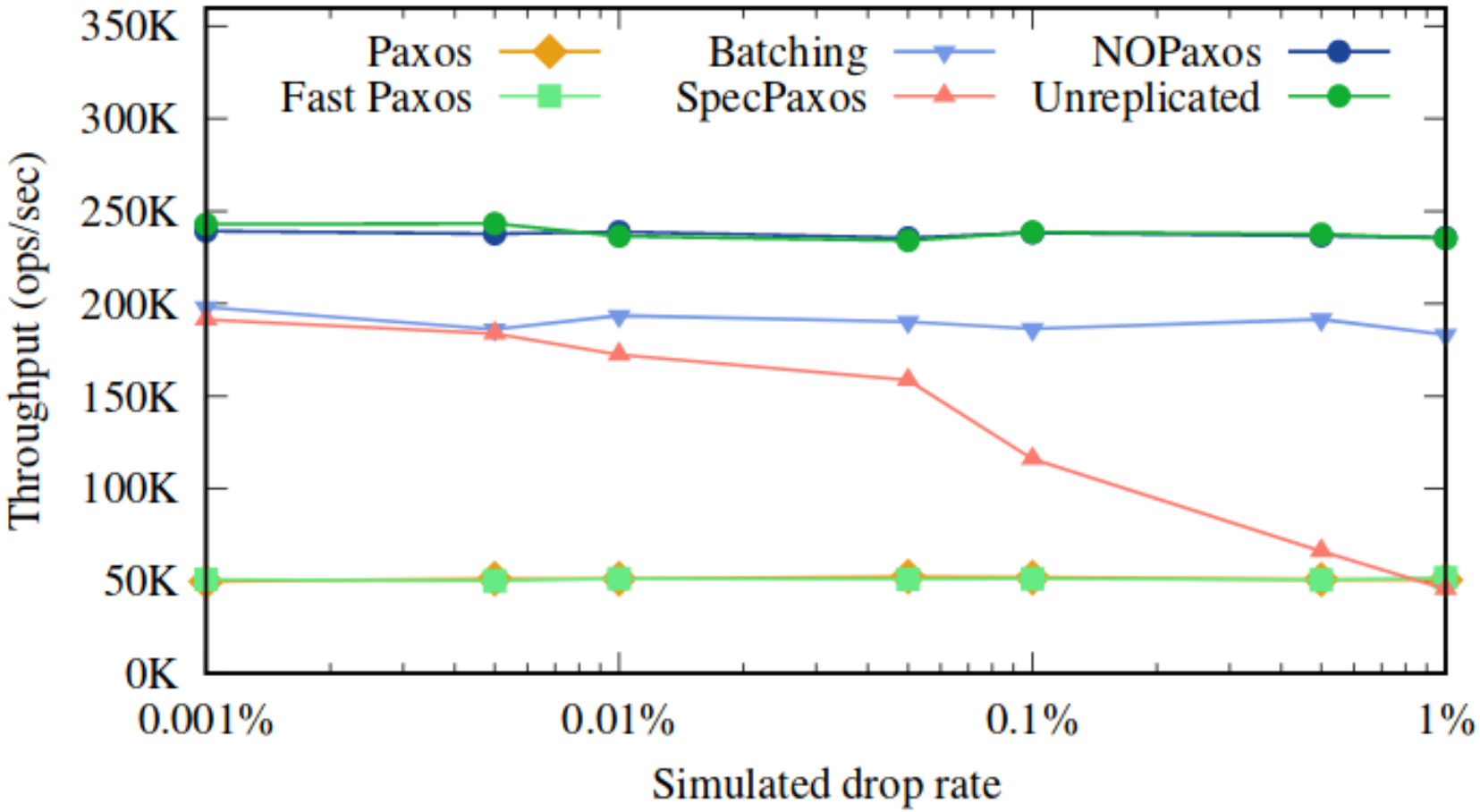
Key result

Fast sequencing $\Longrightarrow$ consensus in one round trip.



Key result

Dropped packets resolved quickly.



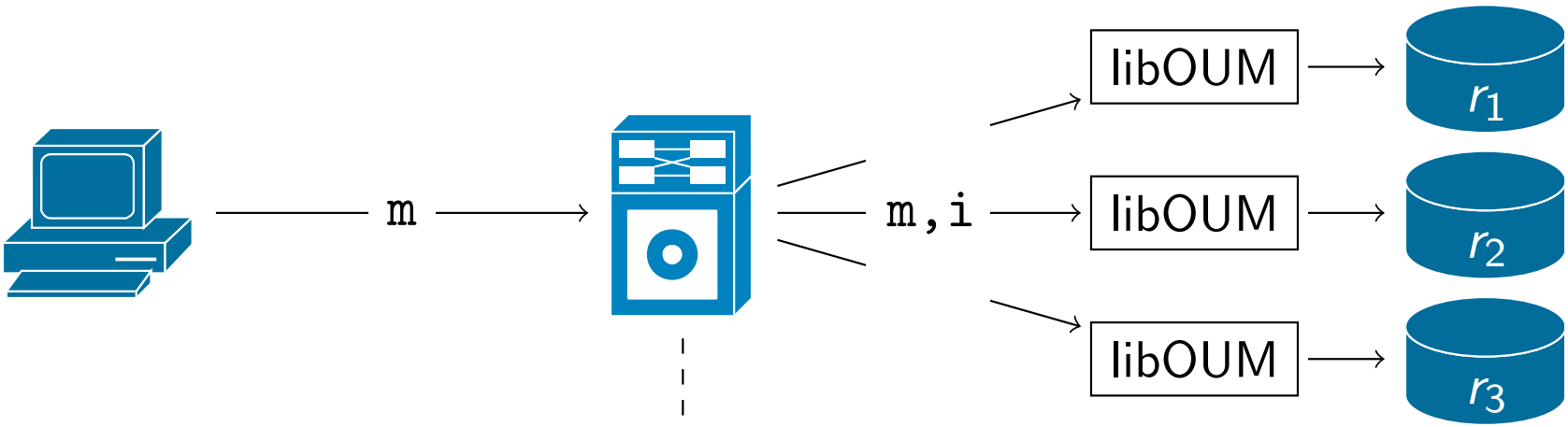
Impact

Useful functionality for programmable network switches.

Idea of accomplishing functionality “in-network”.

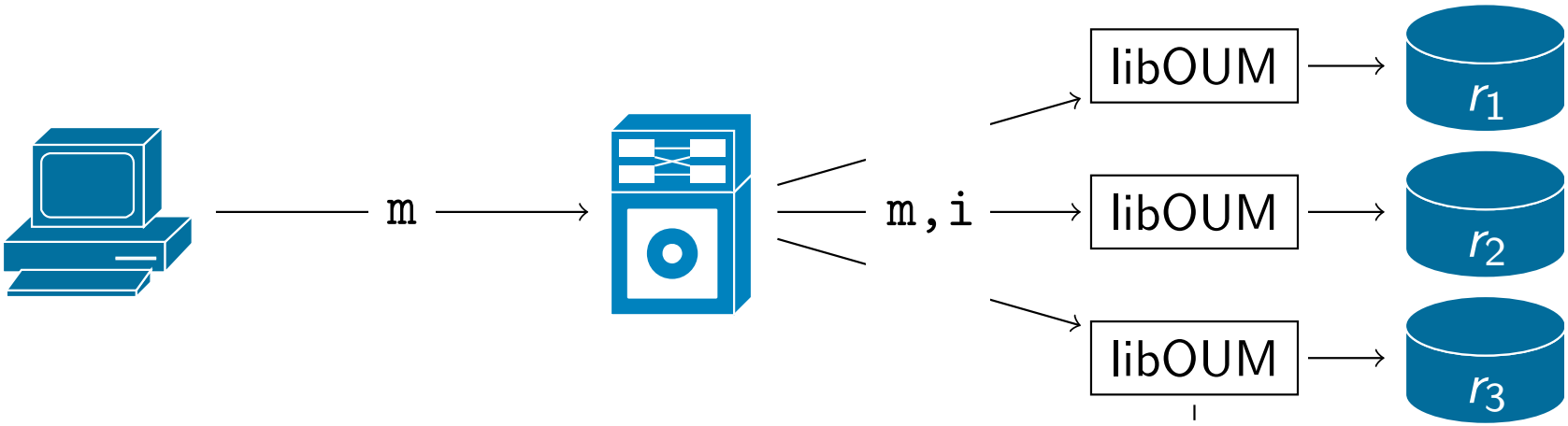
Technical details

OUM Implementation



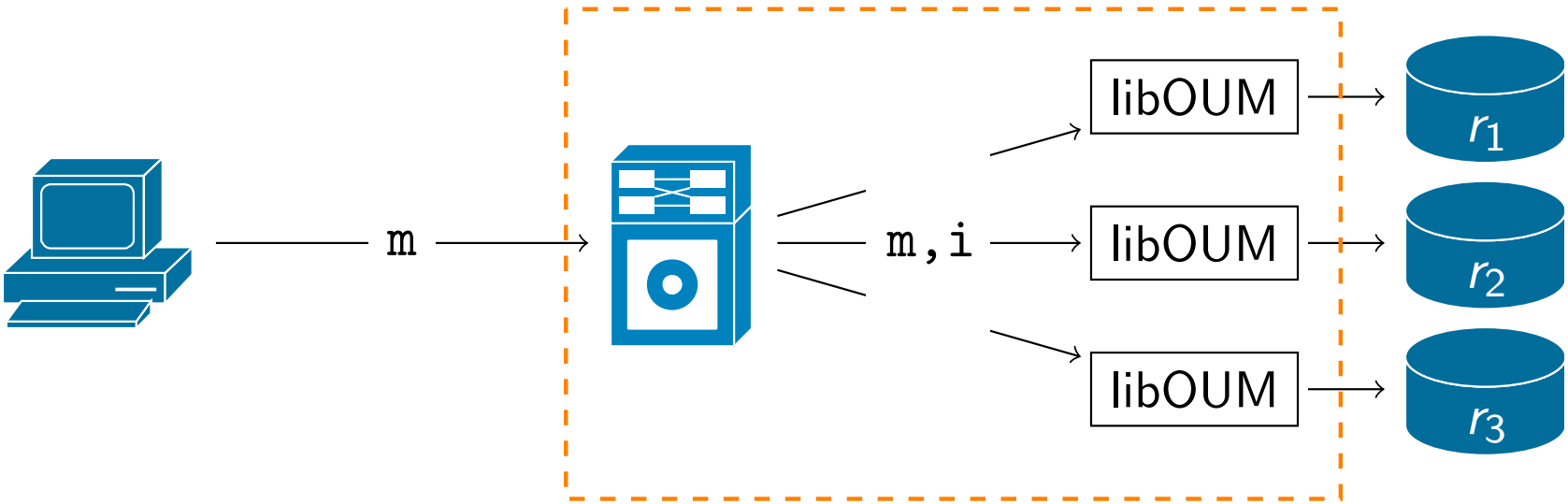

```
counter <- 0
on input m:
    i <- counter++
    send (m,i) to each replica
```

OUM Implementation

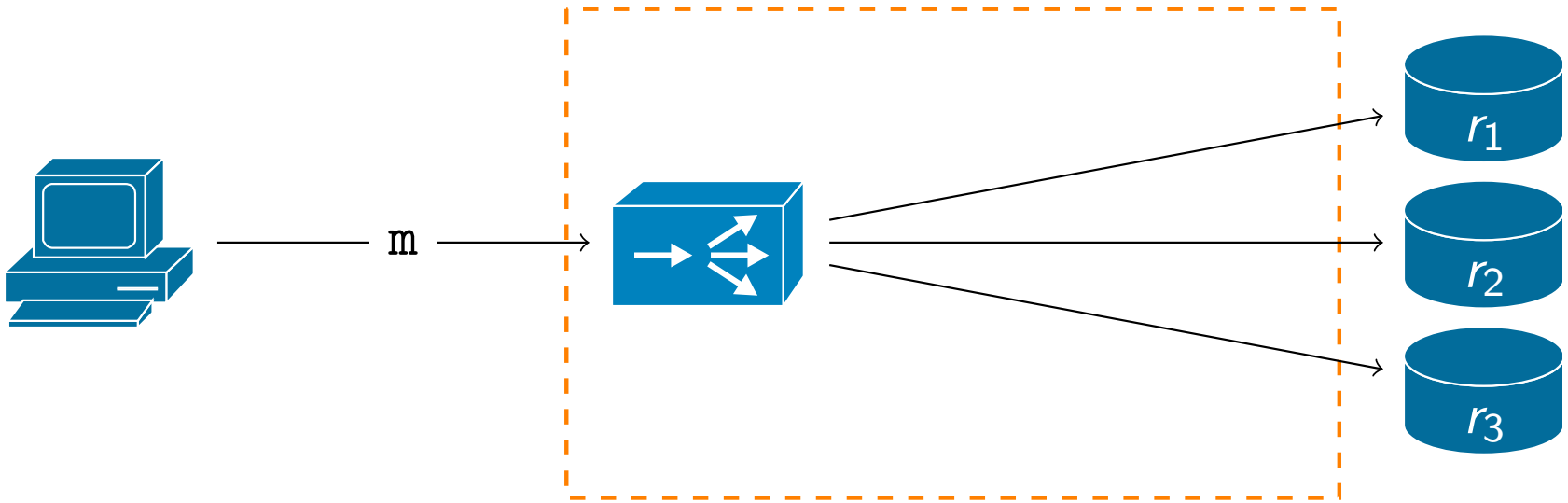



```
counter <- 0
on input (m,i):
    for j from counter to i-1:
        send ⊥ to app
    if i == counter:
        send m to app
    counter <- max(i+1, counter)
```

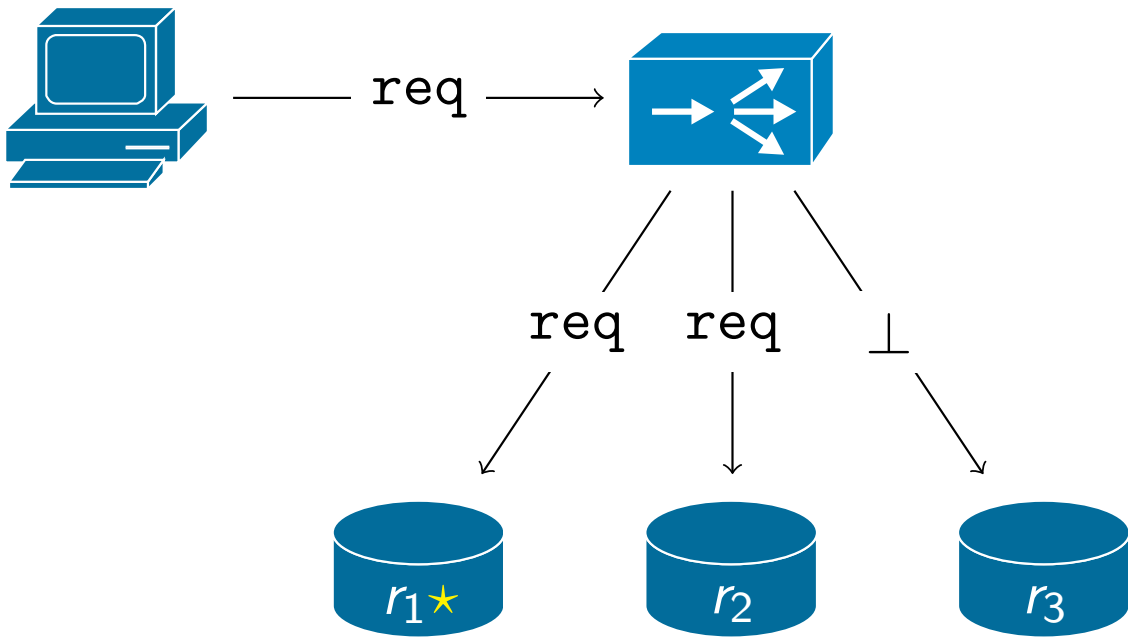
OUM Implementation



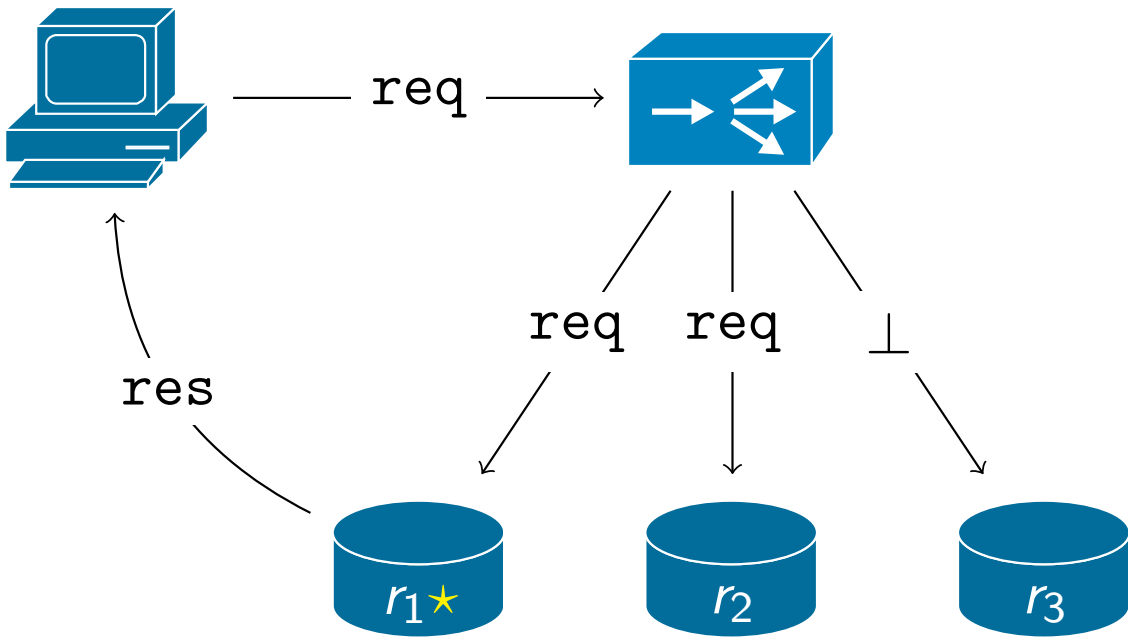
OUM Implementation



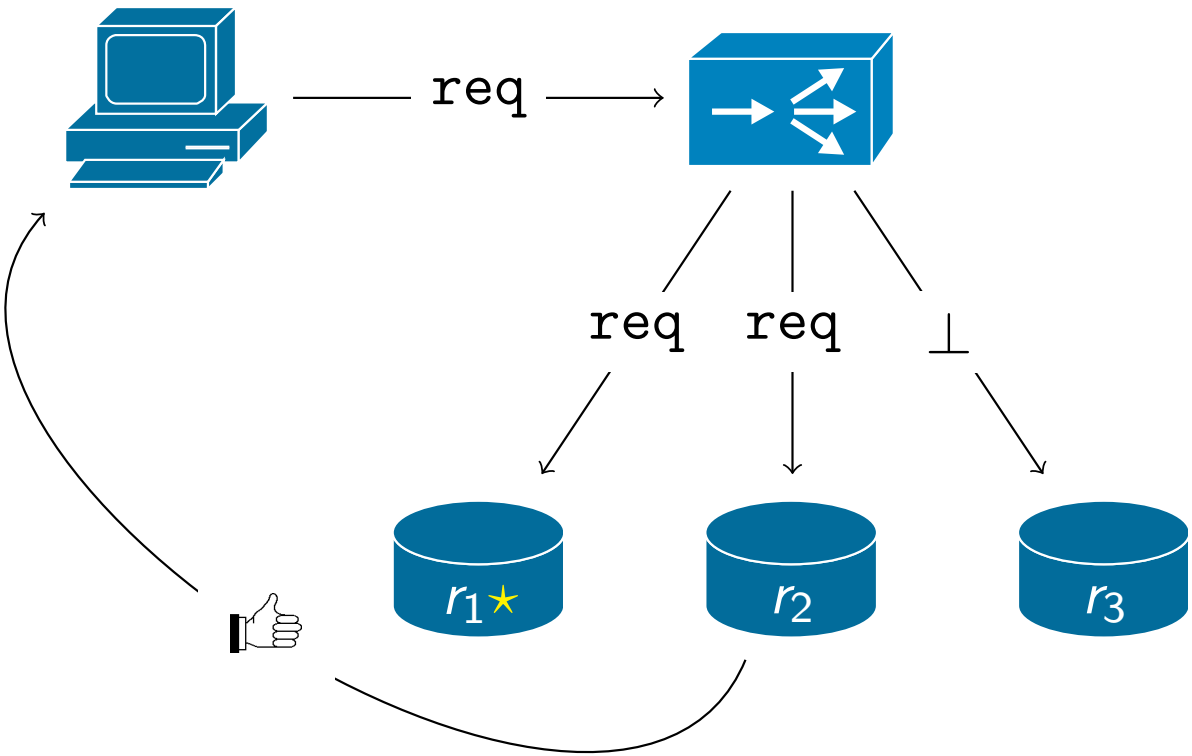
NOPaxos Implementation



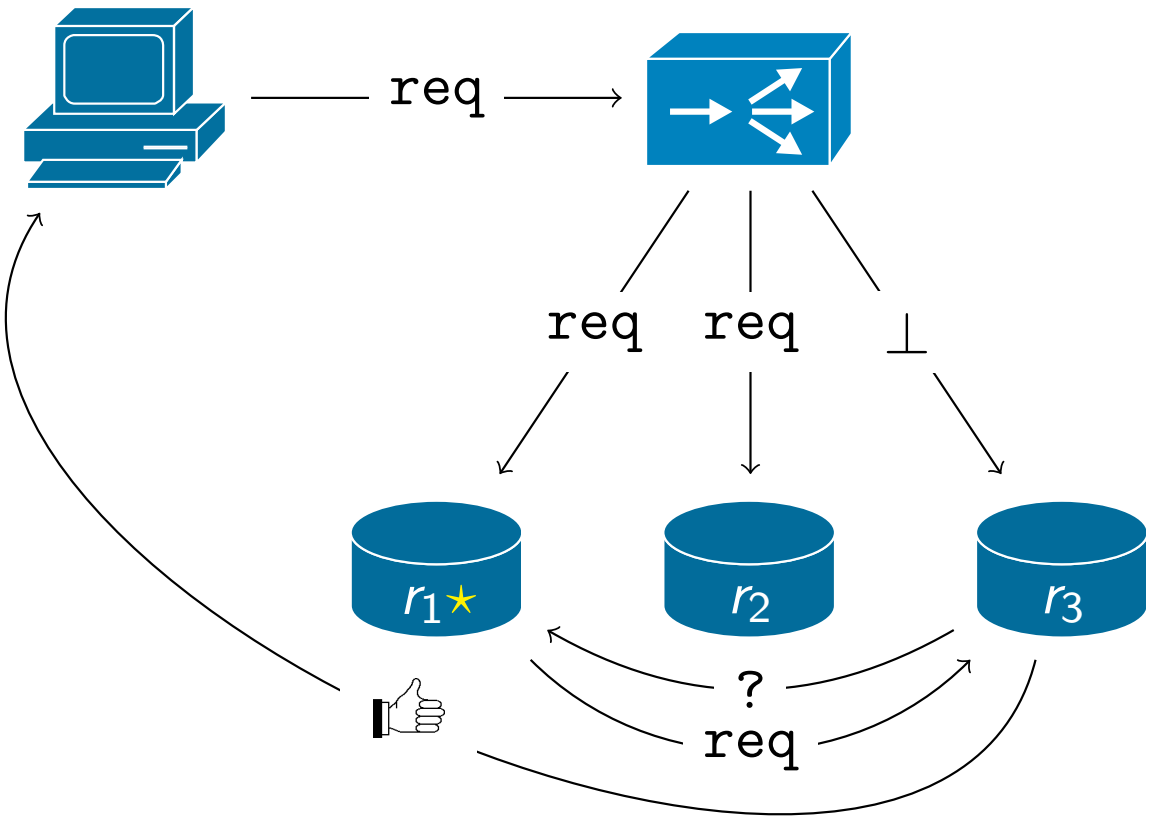
NOPaxos Implementation



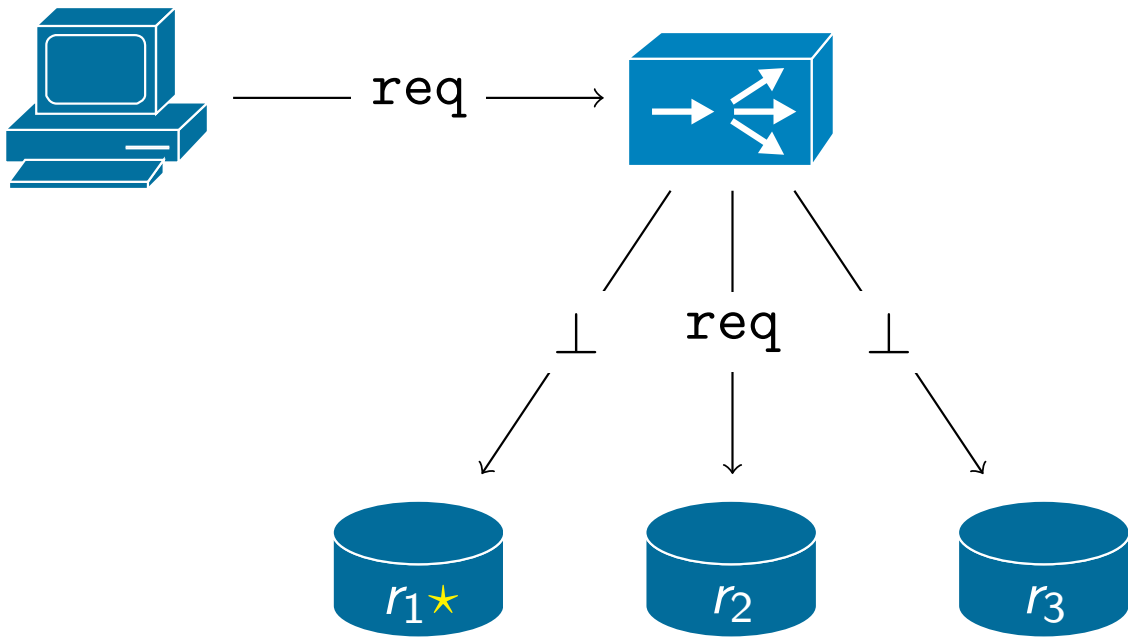
NOPaxos Implementation



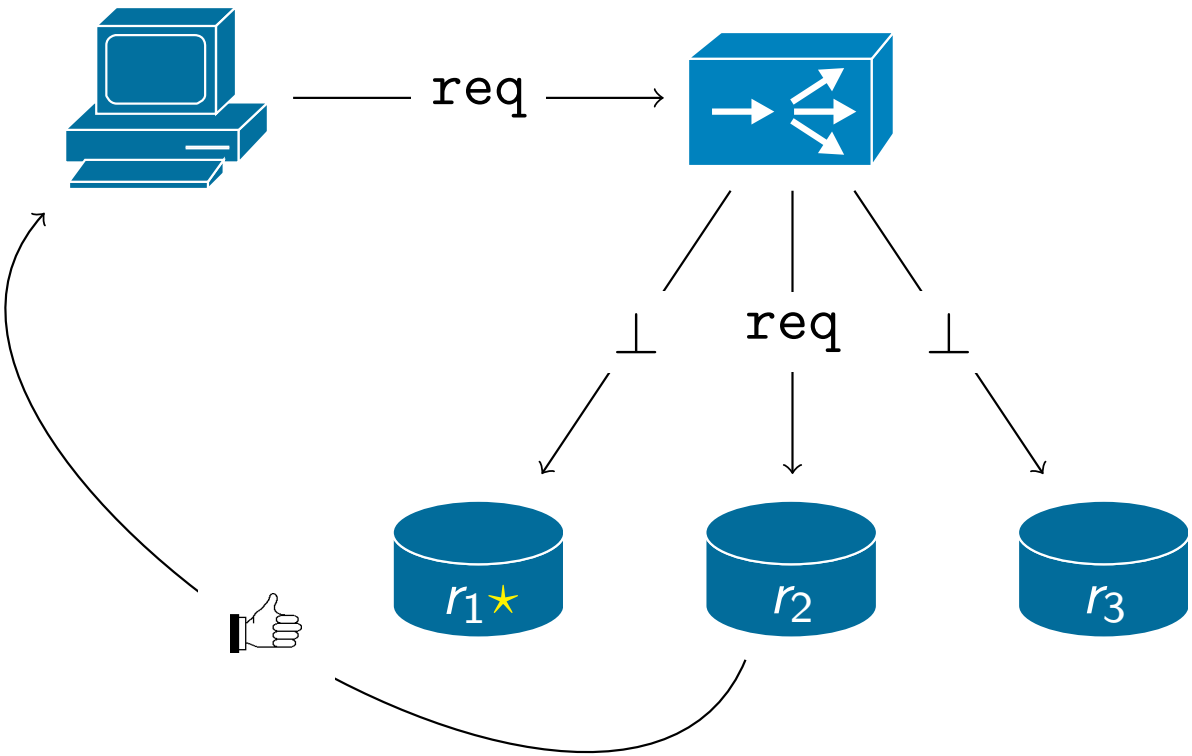
NOPaxos Implementation



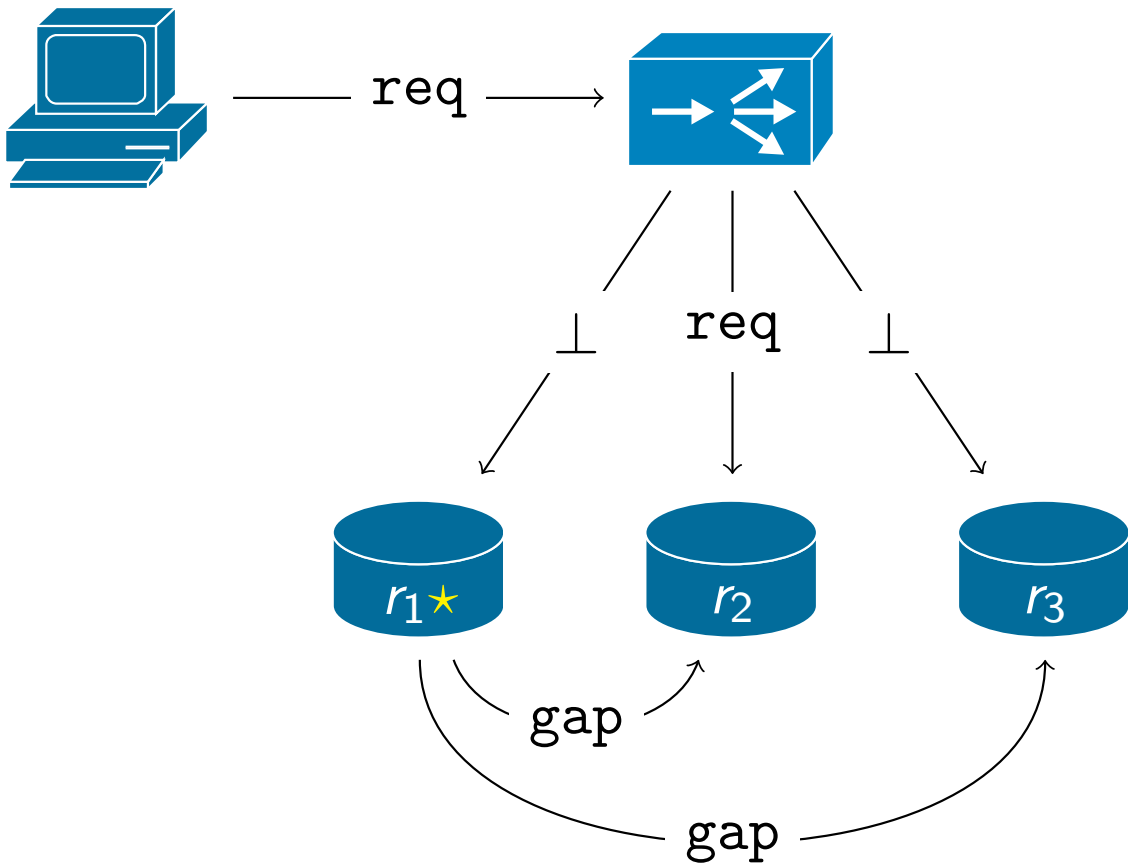
NOPaxos Implementation



NOPaxos Implementation

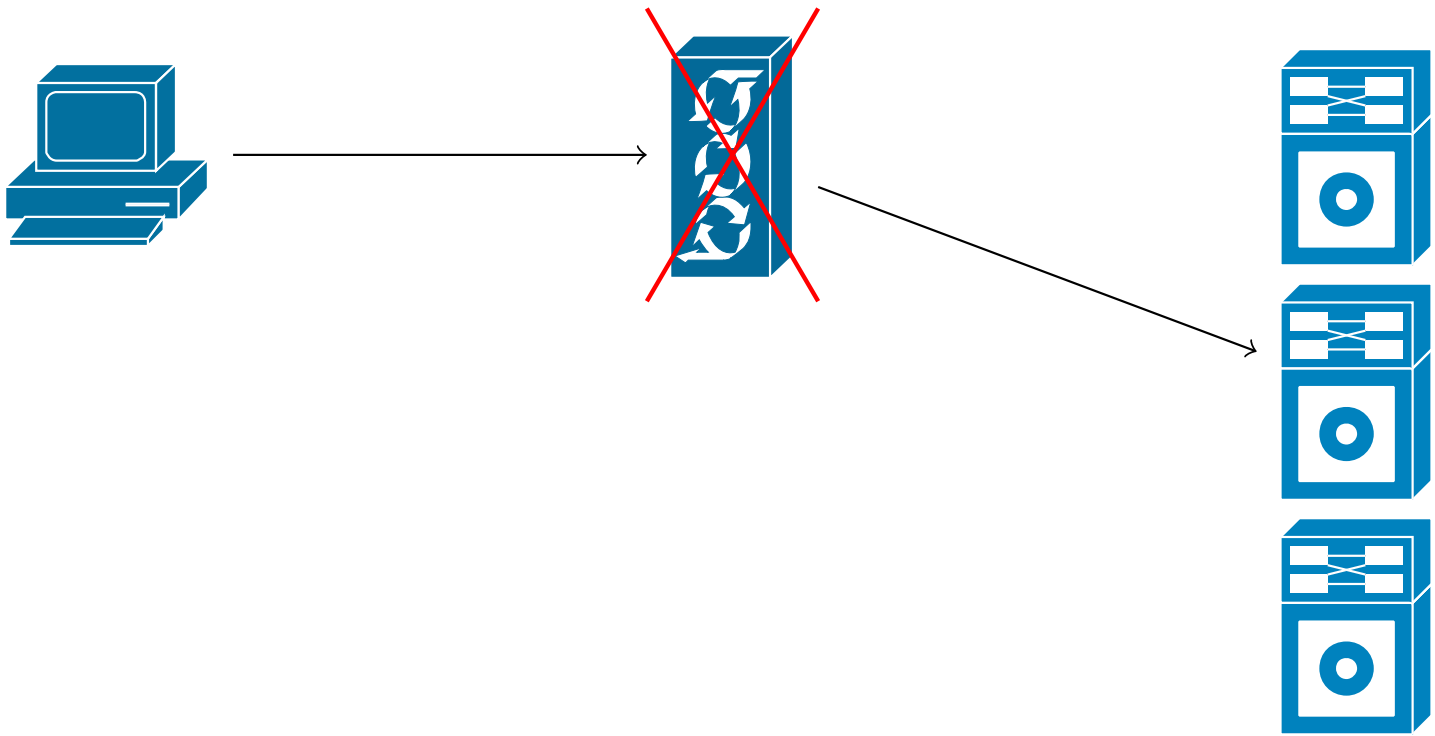


NOPaxos Implementation

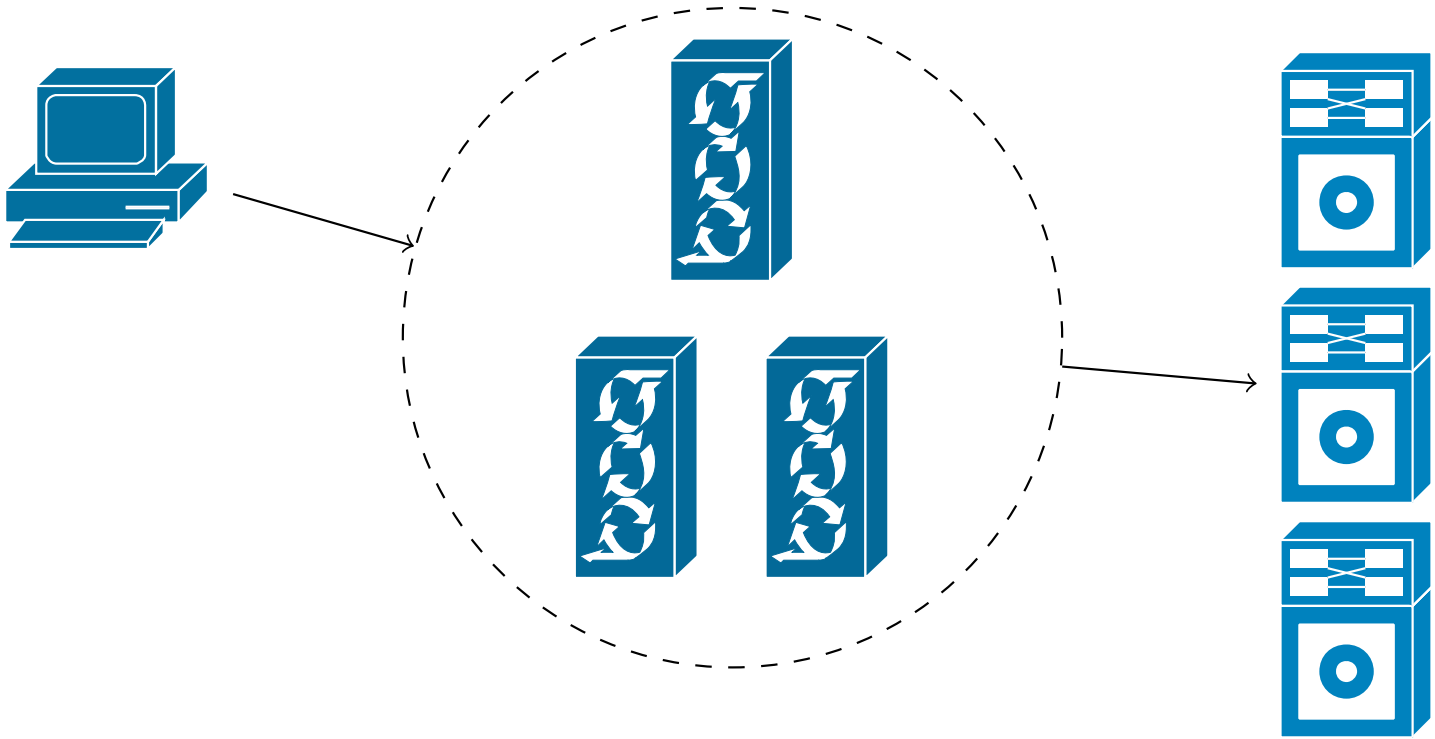


Sequencer failure

Sequencer failure



Sequencer failure



References

K. Birman and T. Joseph.

Exploiting virtual synchrony in distributed systems.
SIGOPS Oper. Syst. Rev., 21(5):123–138, November 1987.

Leslie Lamport.

The part-time parliament.
ACM Trans. Comput. Syst., 16(2):133–169, May 1998.

Leslie Lamport.

Fast paxos.
Distributed Computing, 19:79–103, October 2006.

Leslie Lamport.

Lower bounds for asynchronous consensus.
Distributed Computing, 19(2):104–125, 2006.

Jialin Li, Ellis Michael, Naveen Sharma, Adriana Szekeres, and Dan Ports.

Just say no to paxos overhead: Replacing consensus with network ordering.
OSDI, pages 467–483, 2016.

Diego Ongaro and John Ousterhout.

In search of an understandable consensus algorithm.
USENIX, pages 305–320, 2014.

Dan Ports, Jialin Li, Vincent Liu, Naveen Sharma, and Arvind Krishnamurthy.

Designing distributed systems using approximate synchrony in data center networks.
NSDI, pages 43–57, 2015.