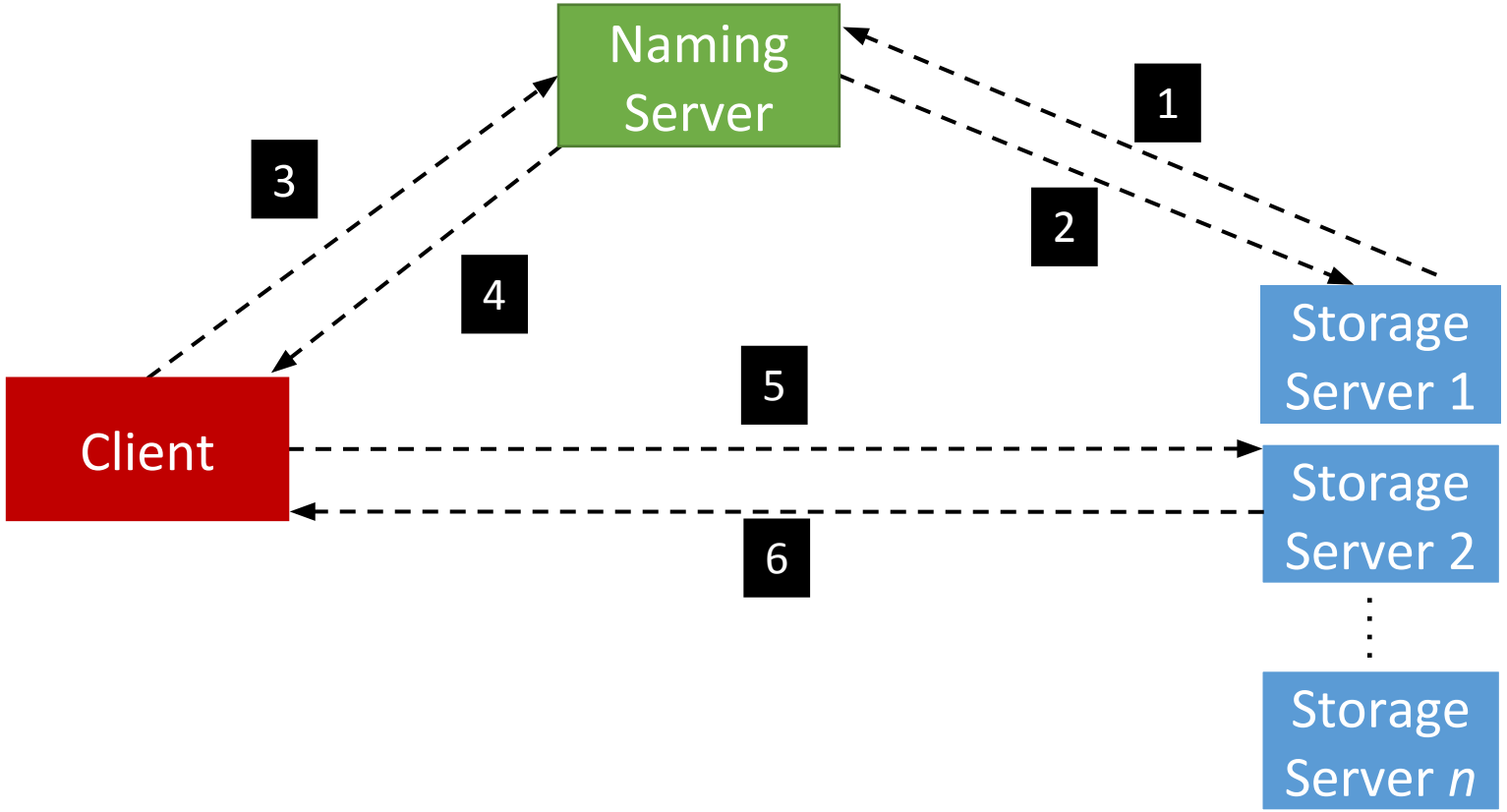
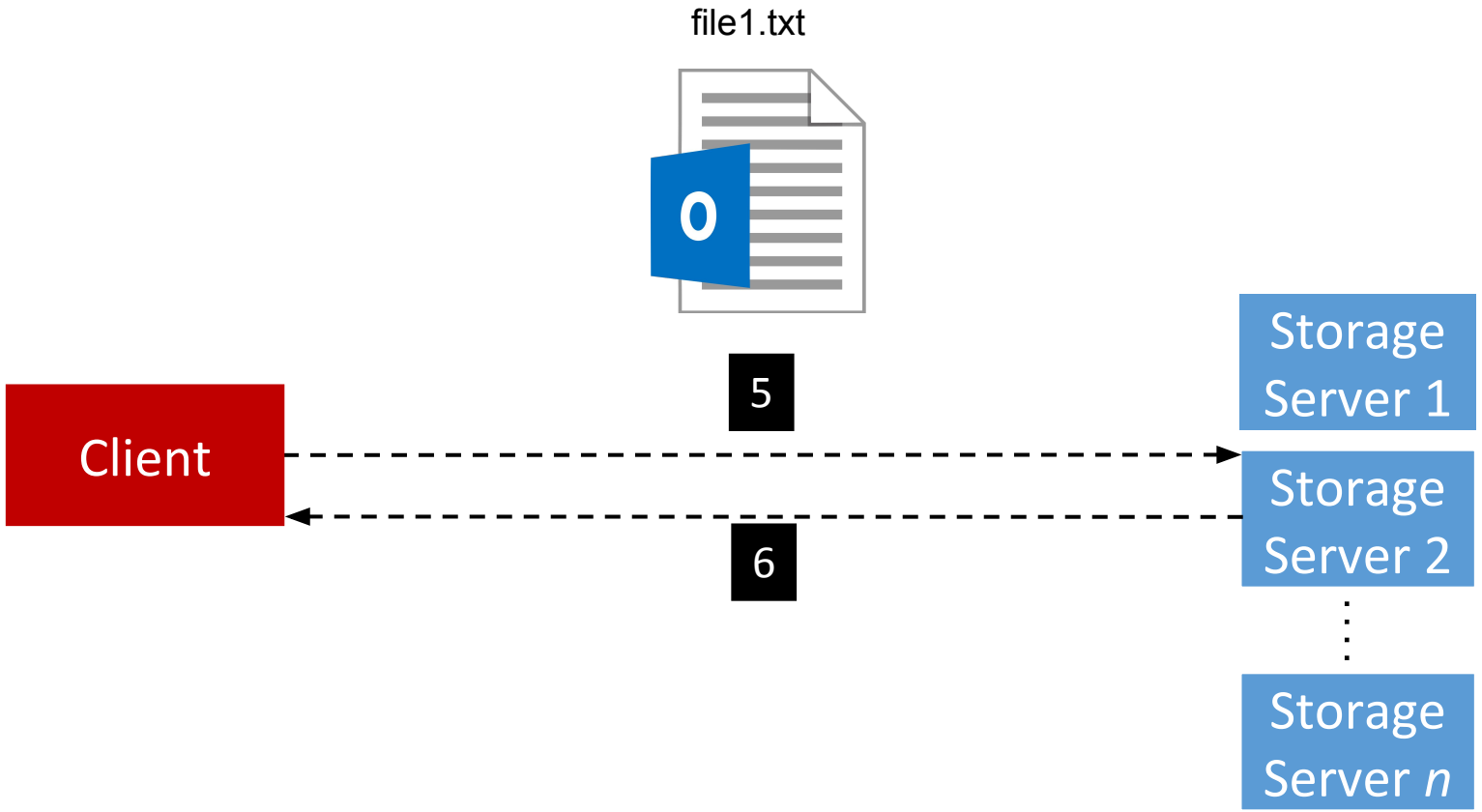
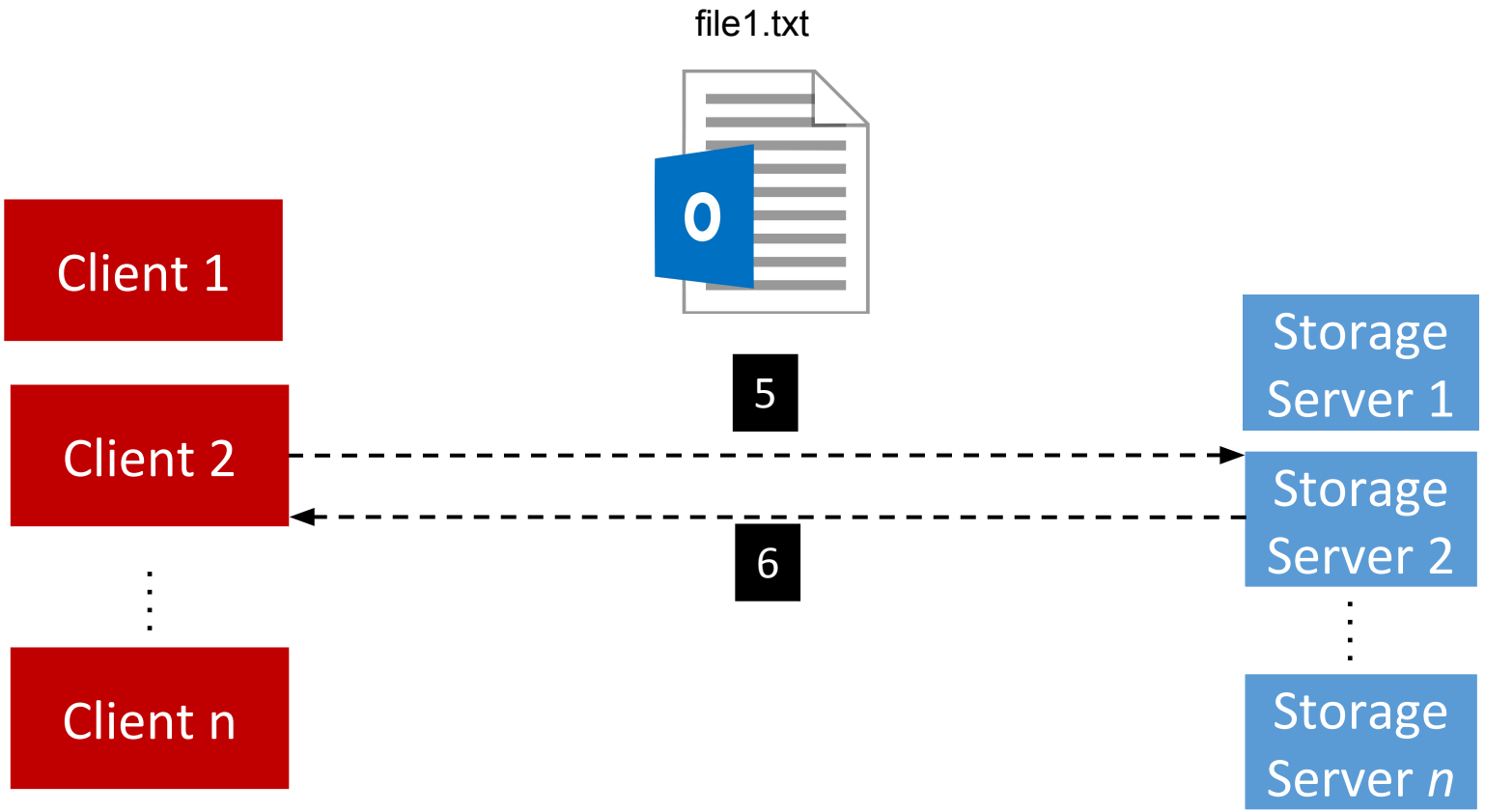
Recitation 6

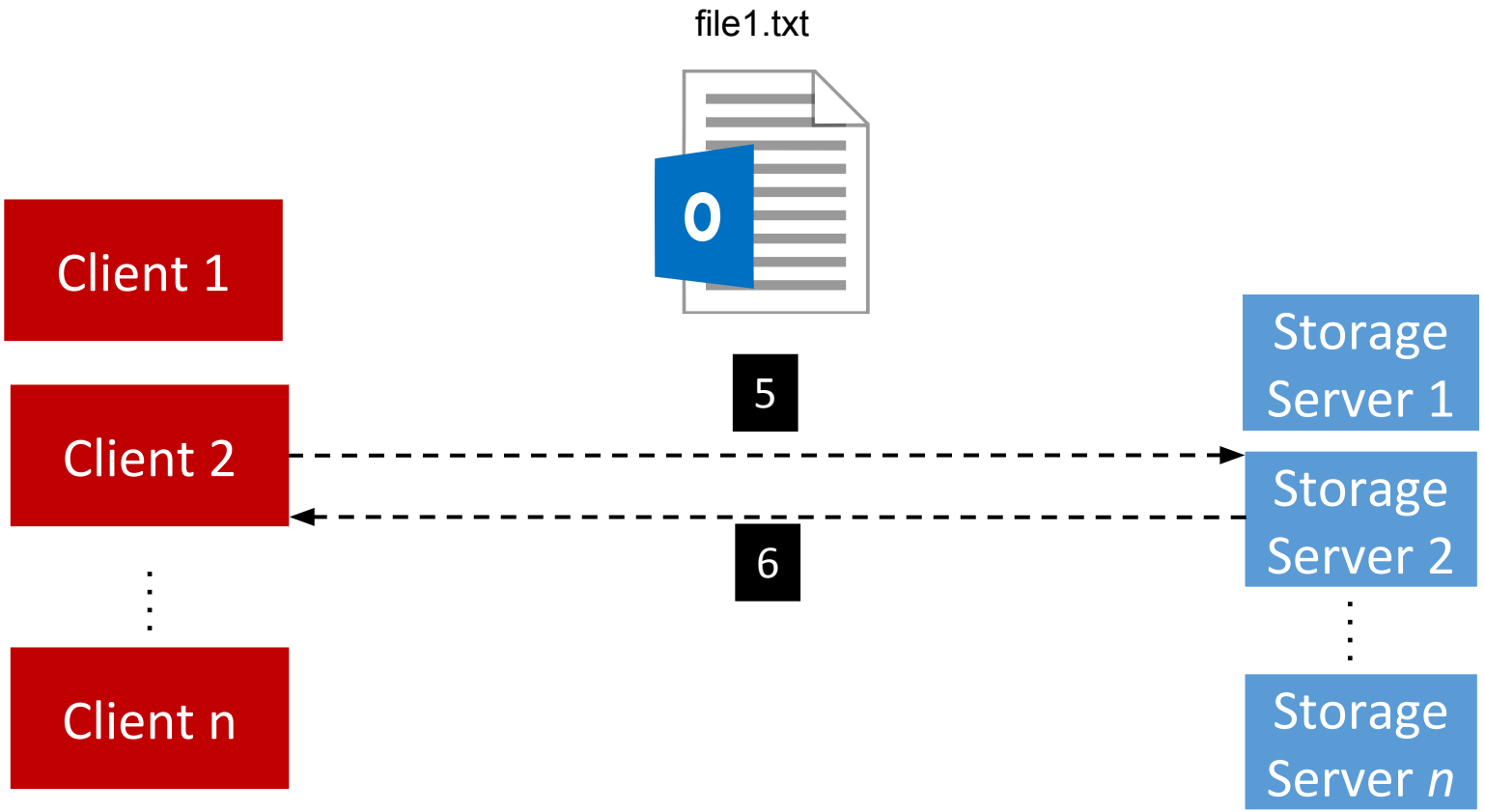
Zeinab Khalifa
October 1st, 2020

Carnegie Mellon University Qatar

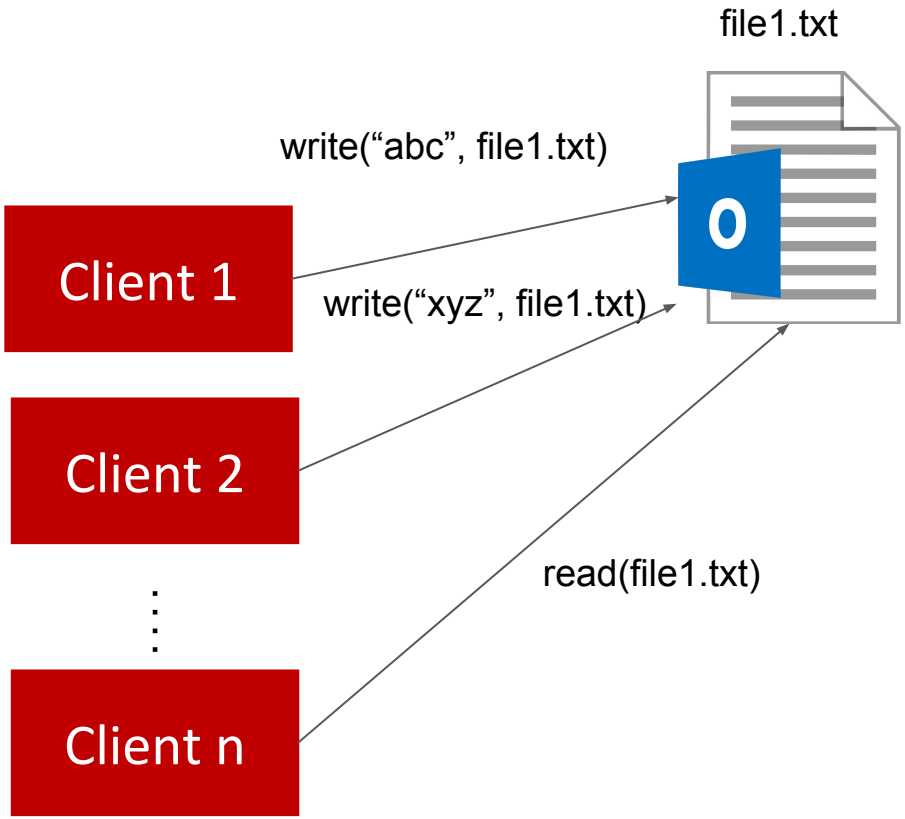

Naming Server
1
2
3
4
5
6
Client
Storage Server 1
Storage Server 2
Storage Server n

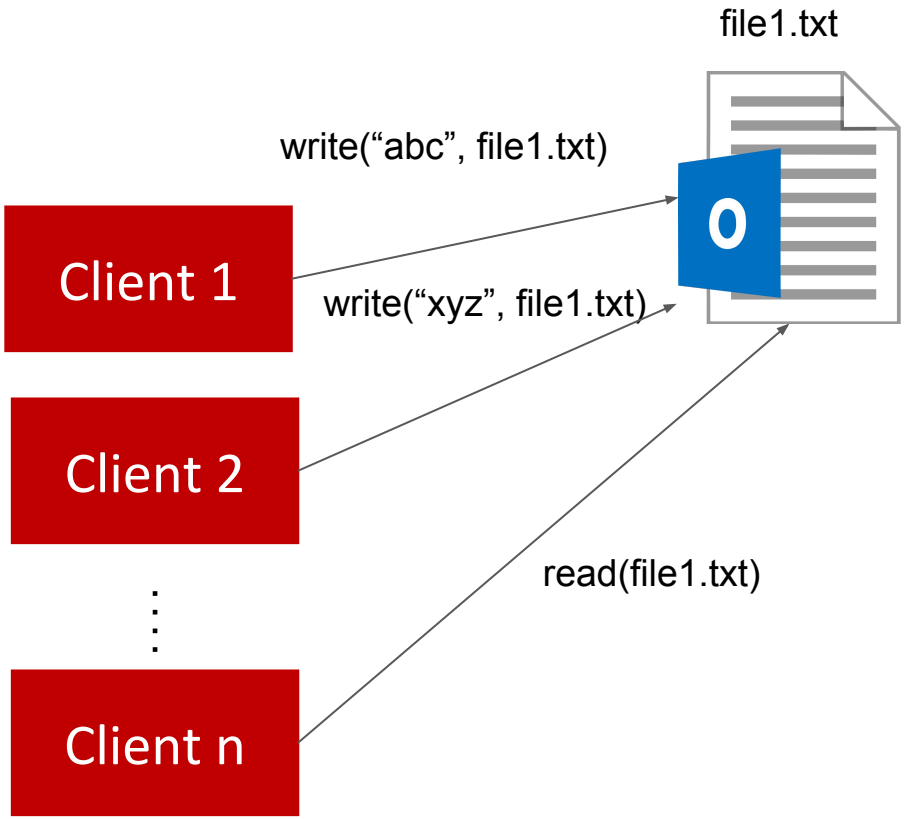

file1.txt
Client
5
6
Storage Server 1
Storage Server 2
Storage Server n


file1.txt
Client 1
Client 2
Client n
5
6
Storage Server 1
Storage Server 2
Storage Server n


file1.txt
Client 1
Client 2
Client n
5
6
Storage Server 1
Storage Server 2
Storage Server n

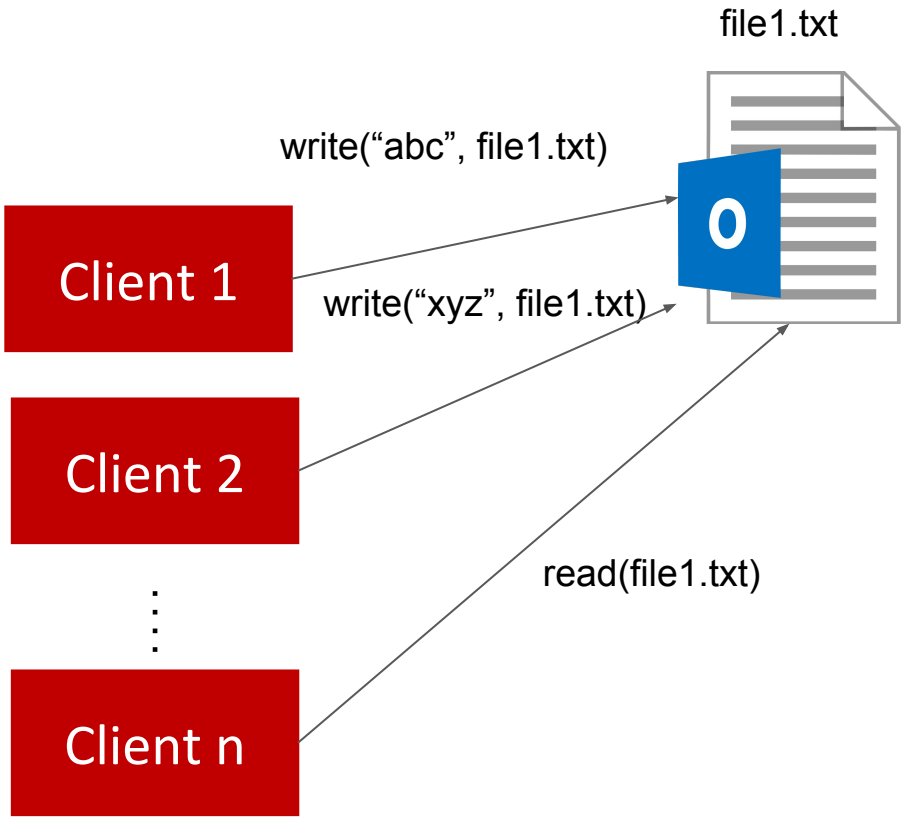
What might go wrong?





What might go wrong?

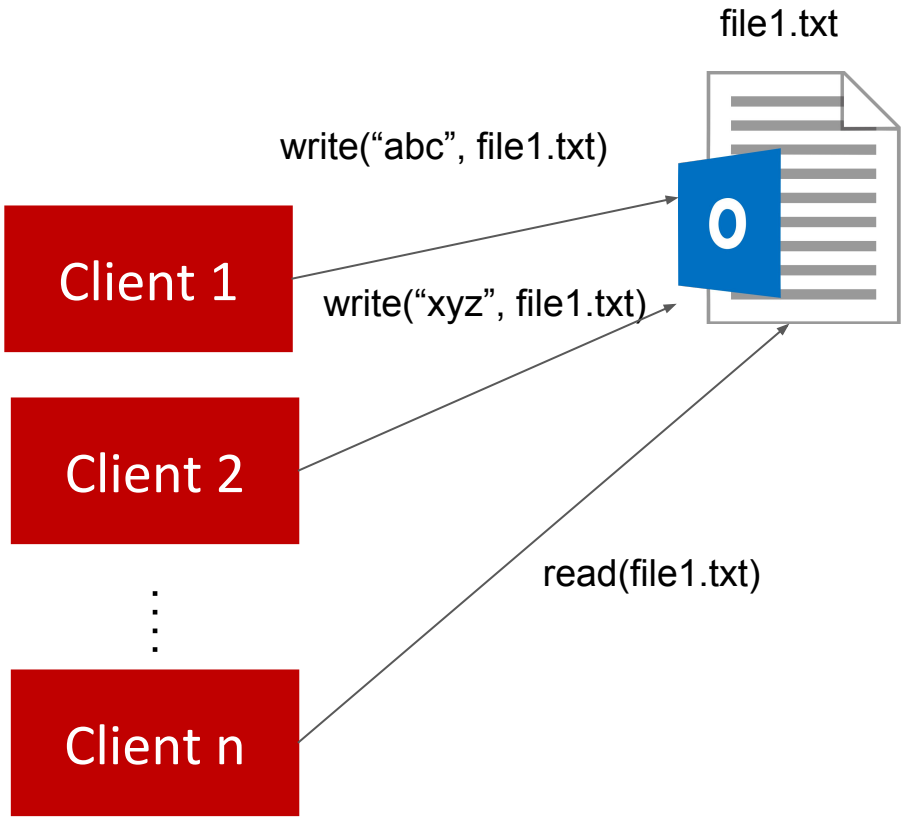
- Synchronization



What might go wrong?

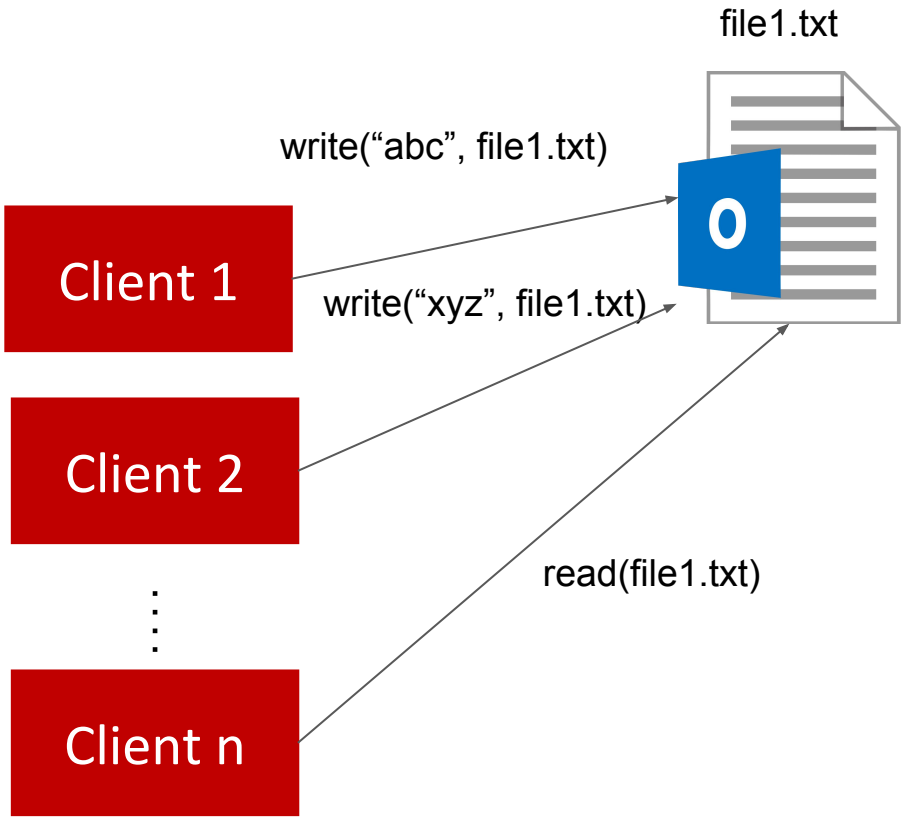
- Synchronization

file1.txt is hosted on SS9, and it's gets 5000 reqs/ sec. As opposed to file2.txt which gets 1000 reqs / month on SS3



What might go wrong?

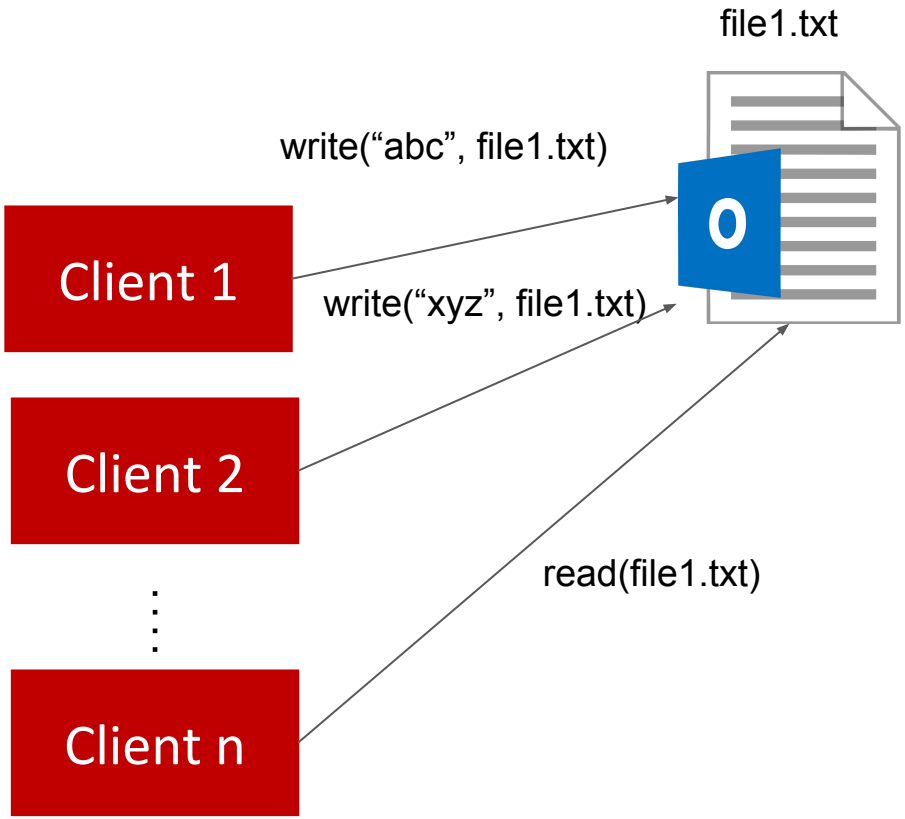
- Synchronization
- Load-balancing



What might go wrong?

- Synchronization
- Load-balancing

Replicate file1.txt on multiple Storage Servers

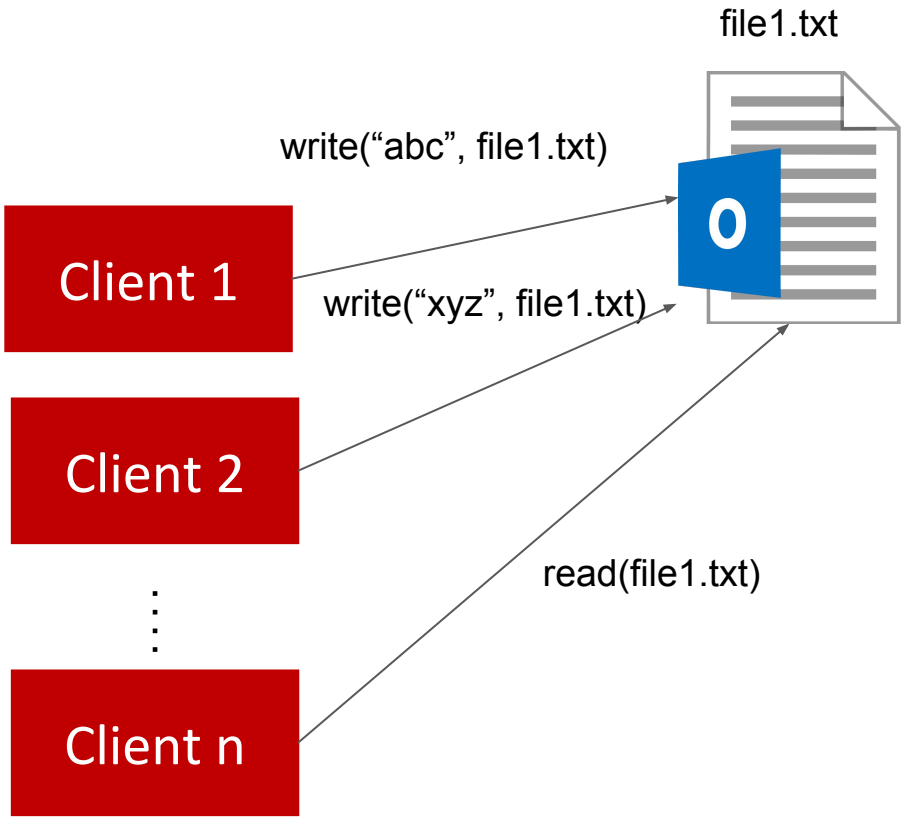


What might go wrong?

- Synchronization
- Load-balancing

Replicate file1.txt on multiple Storage Servers

- Consistency



What might go wrong?

- Synchronization
- Load-balancing
- Consistency

Project 2 Objectives

Synchronization

Readers

Writers

Synchronization

Readers

Shared Lock

Synchronization

Readers

Shared Lock

Can multiple readers read the file simultaneously?

Can multiple readers read the file while someone is writing to it?

Synchronization

Readers

Shared Lock

Can multiple readers read the file simultaneously?

Can multiple readers read the file while someone is writing to it?

Writers

Exclusive Lock

Synchronization

Readers

Shared Lock

Can multiple readers read the file simultaneously?

Can multiple readers read the file while someone is writing to it?

Writers

Exclusive Lock

Can multiple people write to the same file?

Can multiple people write to a file while someone is reading it?

Synchronization

Readers

Shared Lock

Can multiple readers read the file simultaneously?

Can multiple readers read the file while someone is writing to it?

Writers

Exclusive Lock

Can multiple people write to the same file?

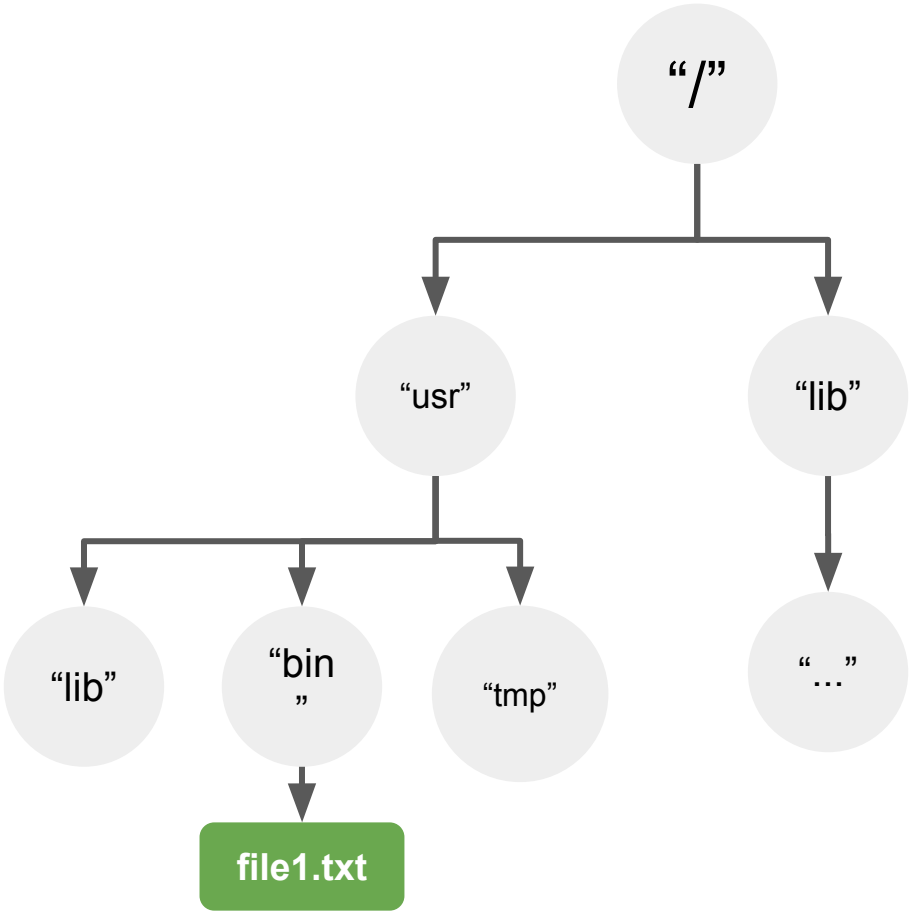
Can multiple people write to a file while someone is reading it?

Where are these locks implemented?

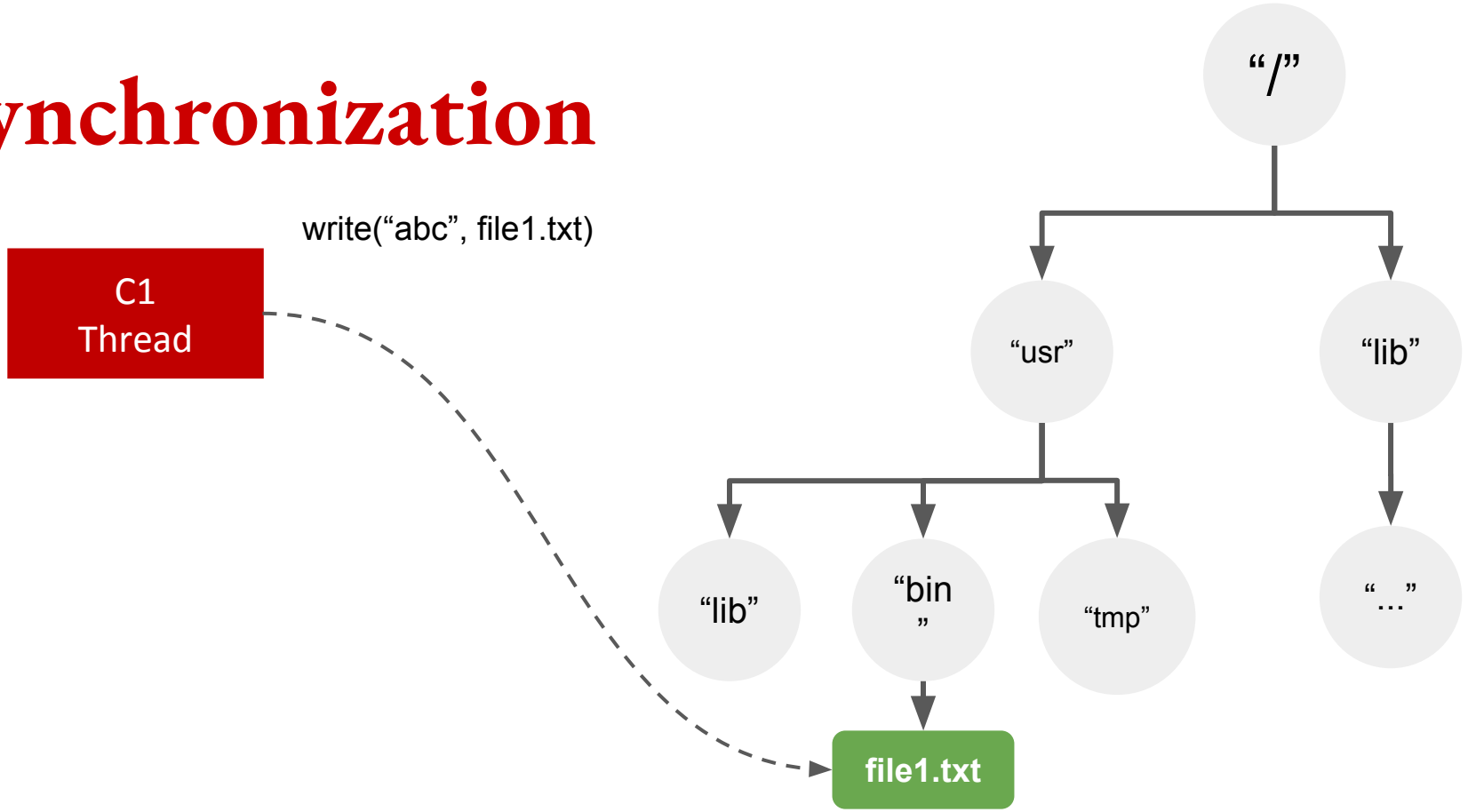
Synchronization

Where are these locks implemented?

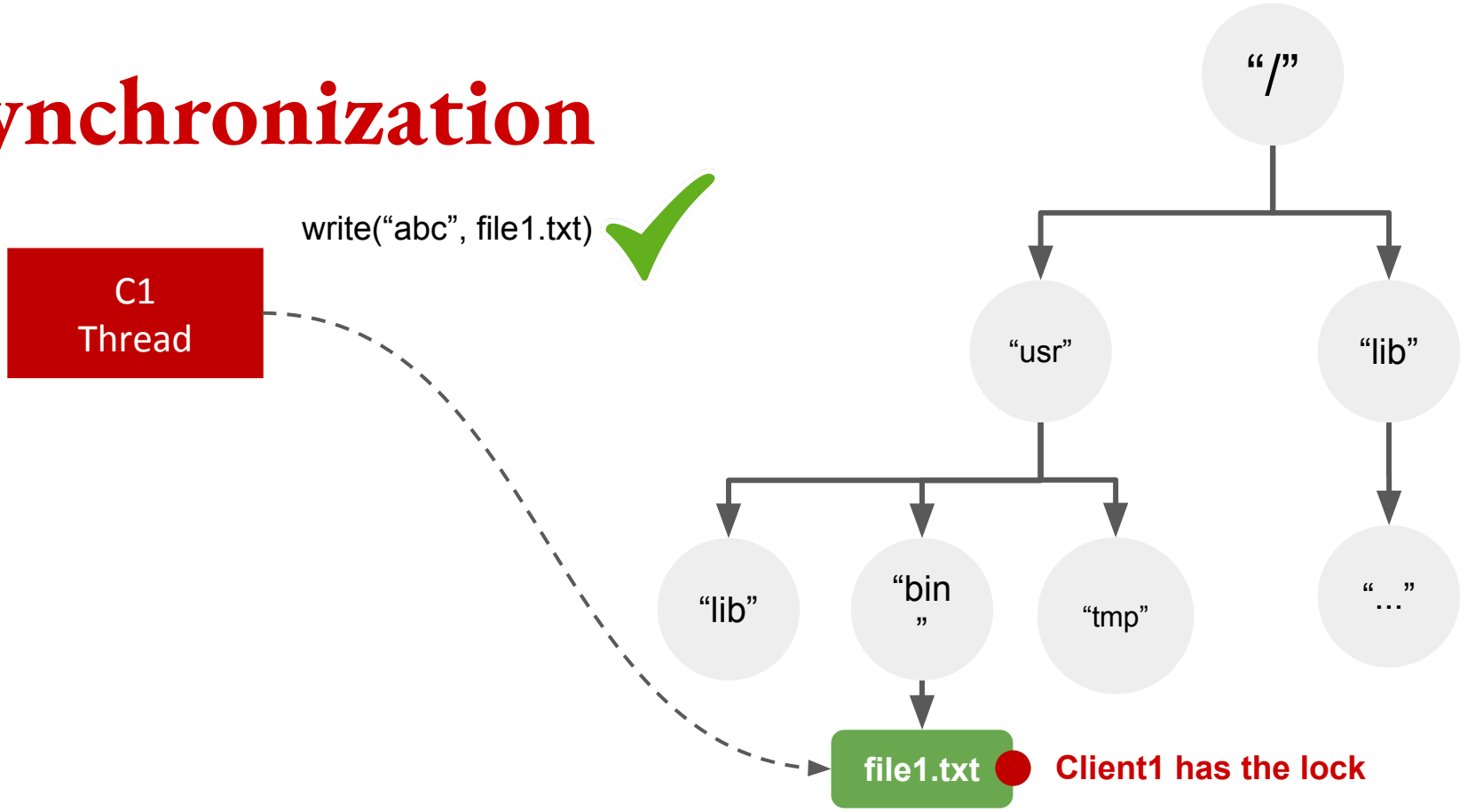
Synchronization



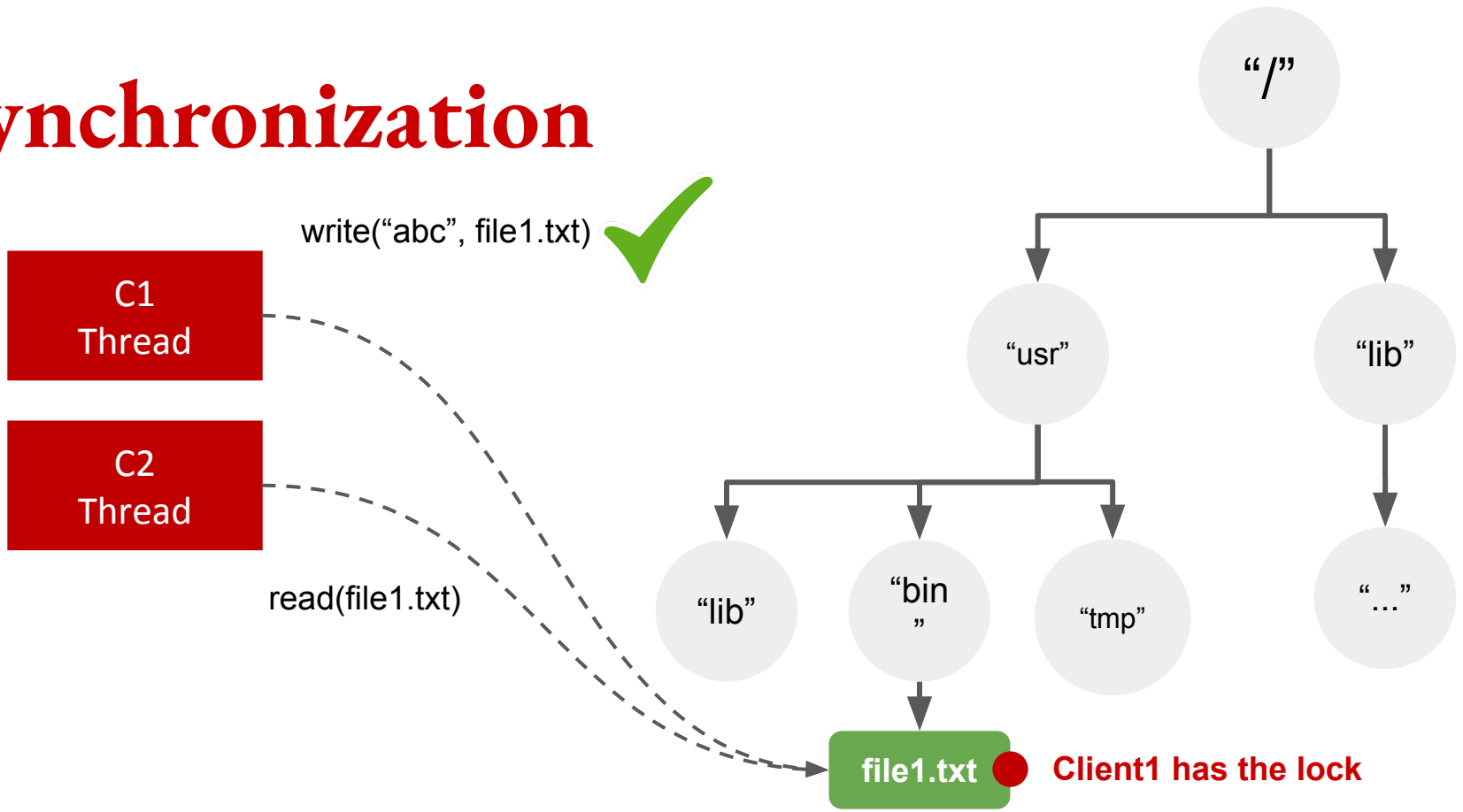
Synchronization



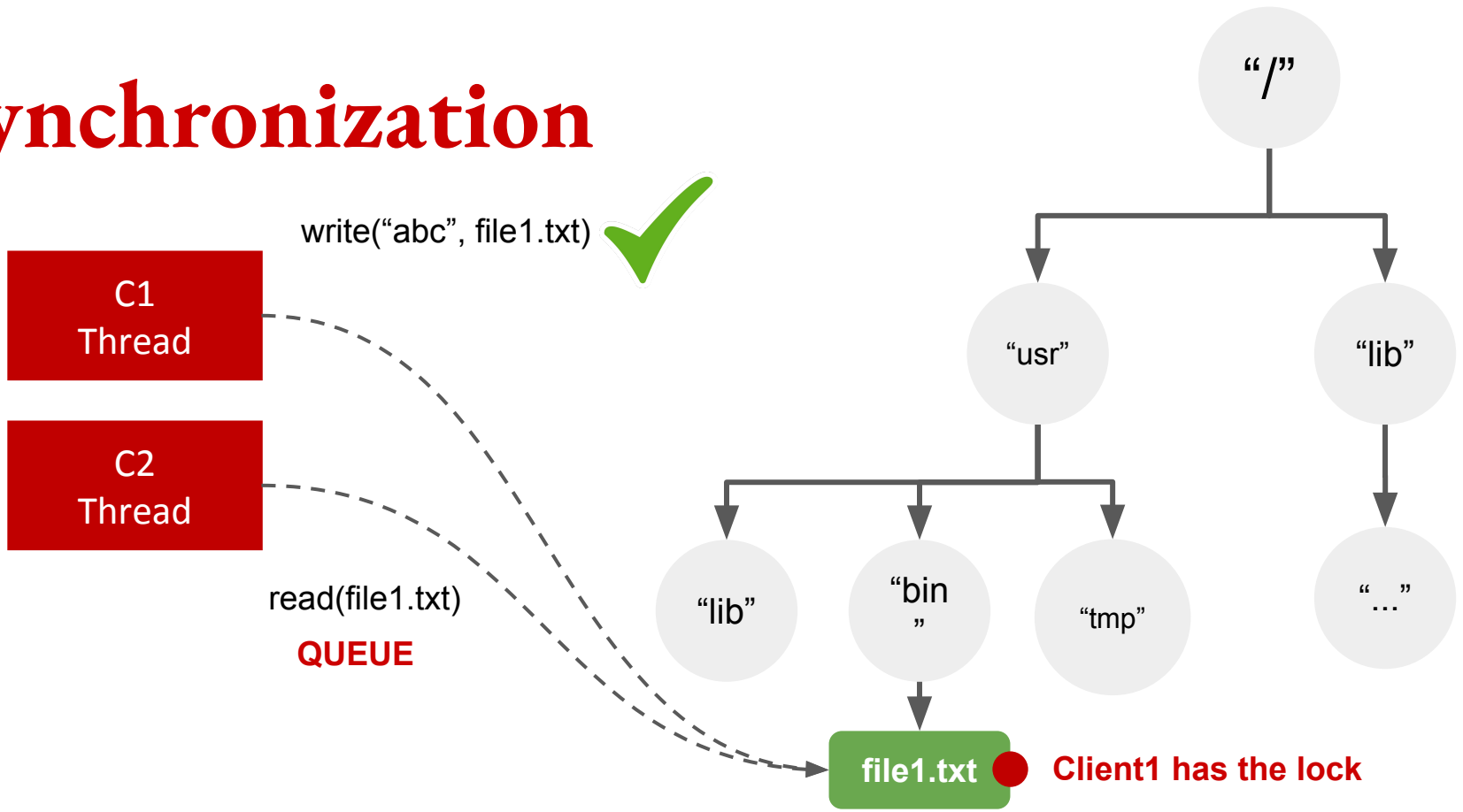
Synchronization



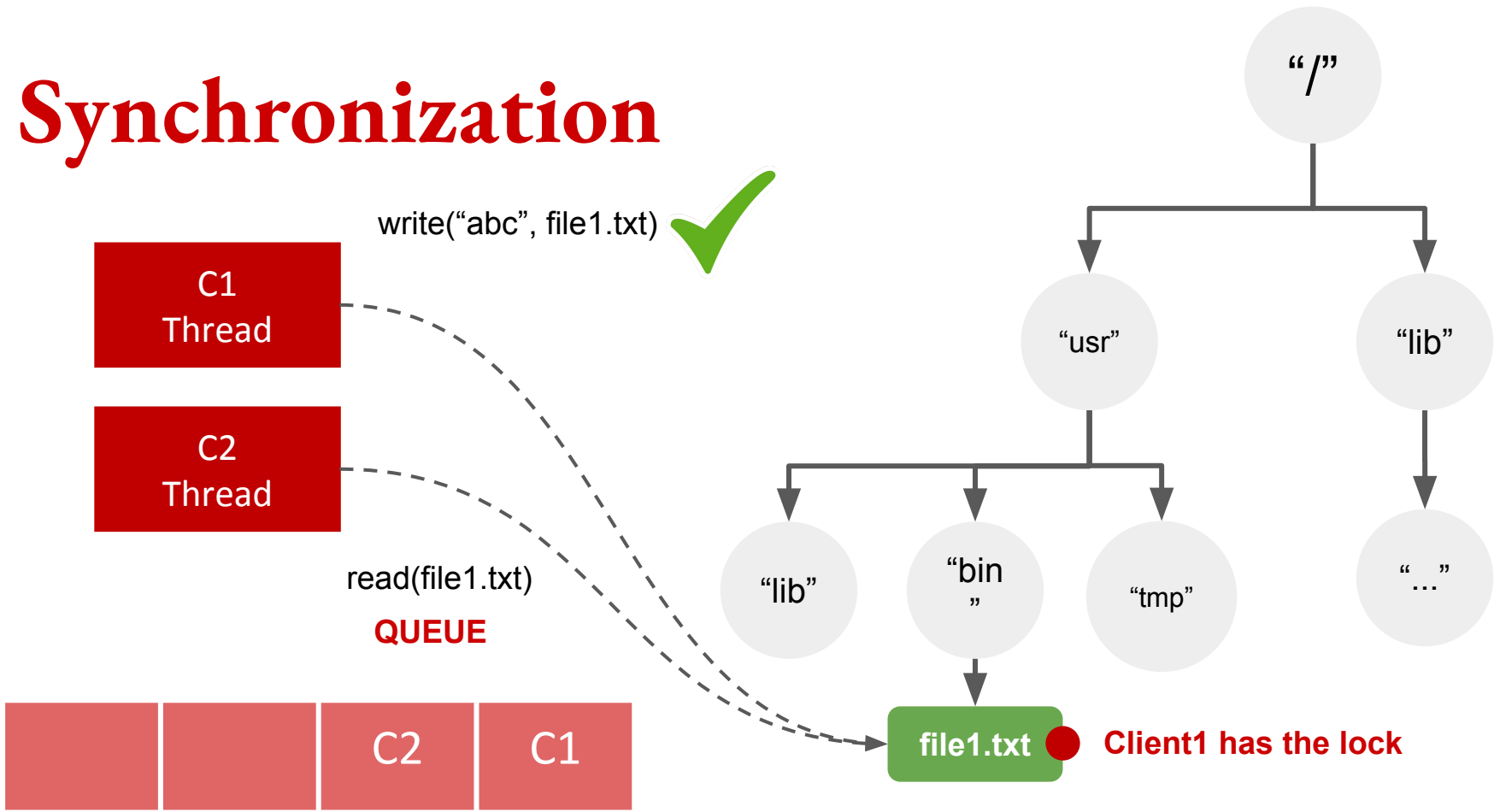
Synchronization



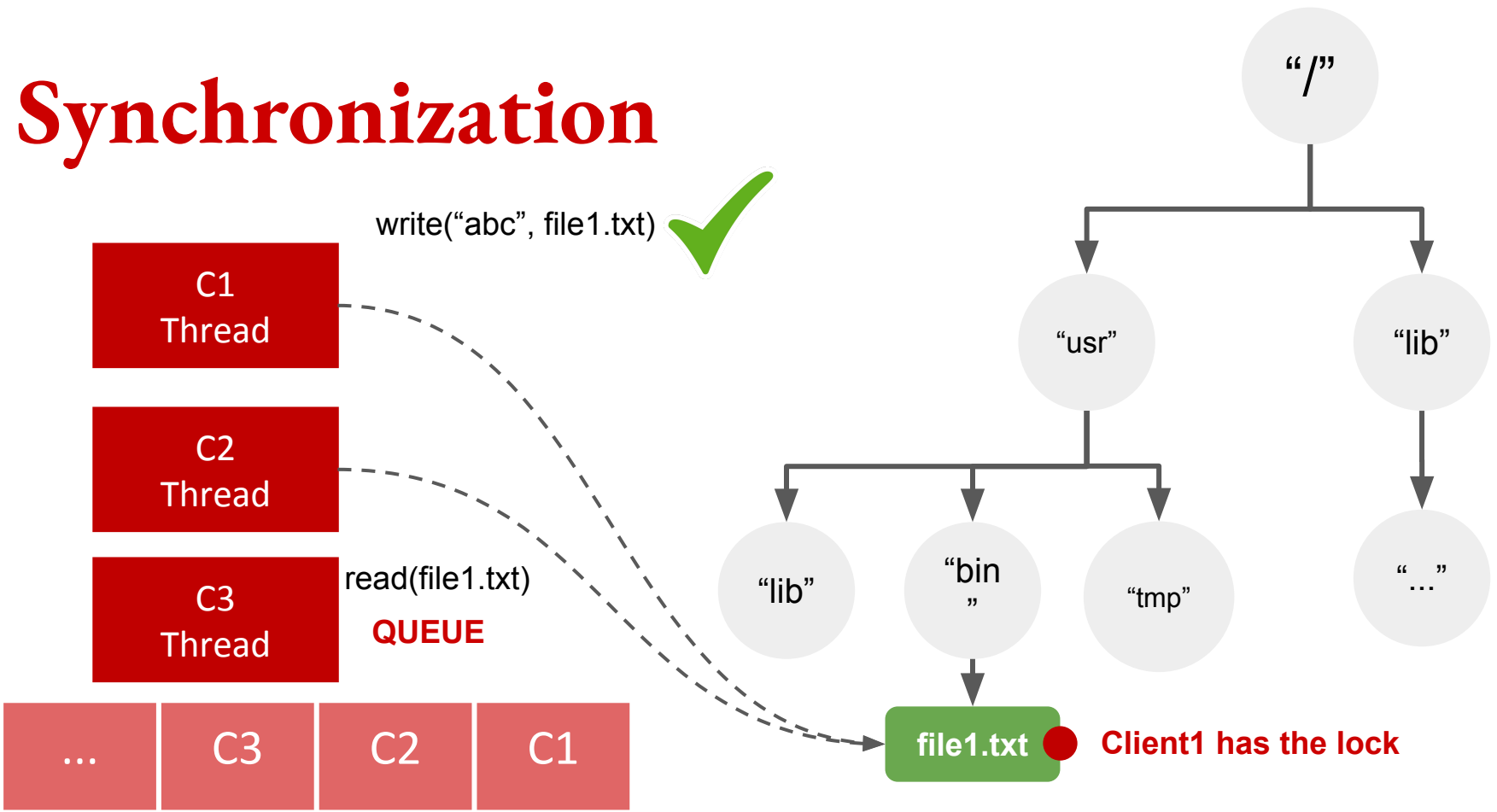
Synchronization



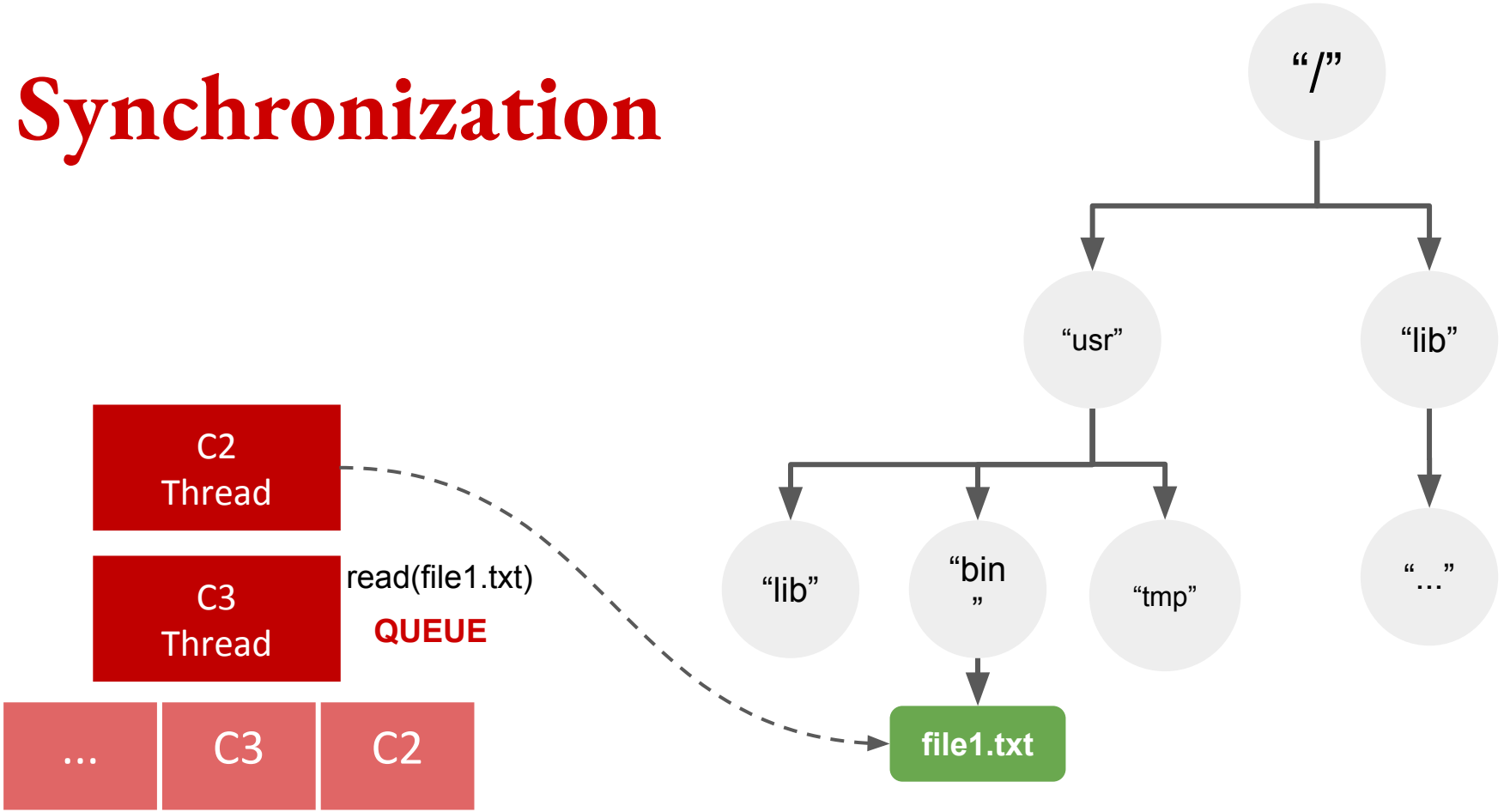
Synchronization



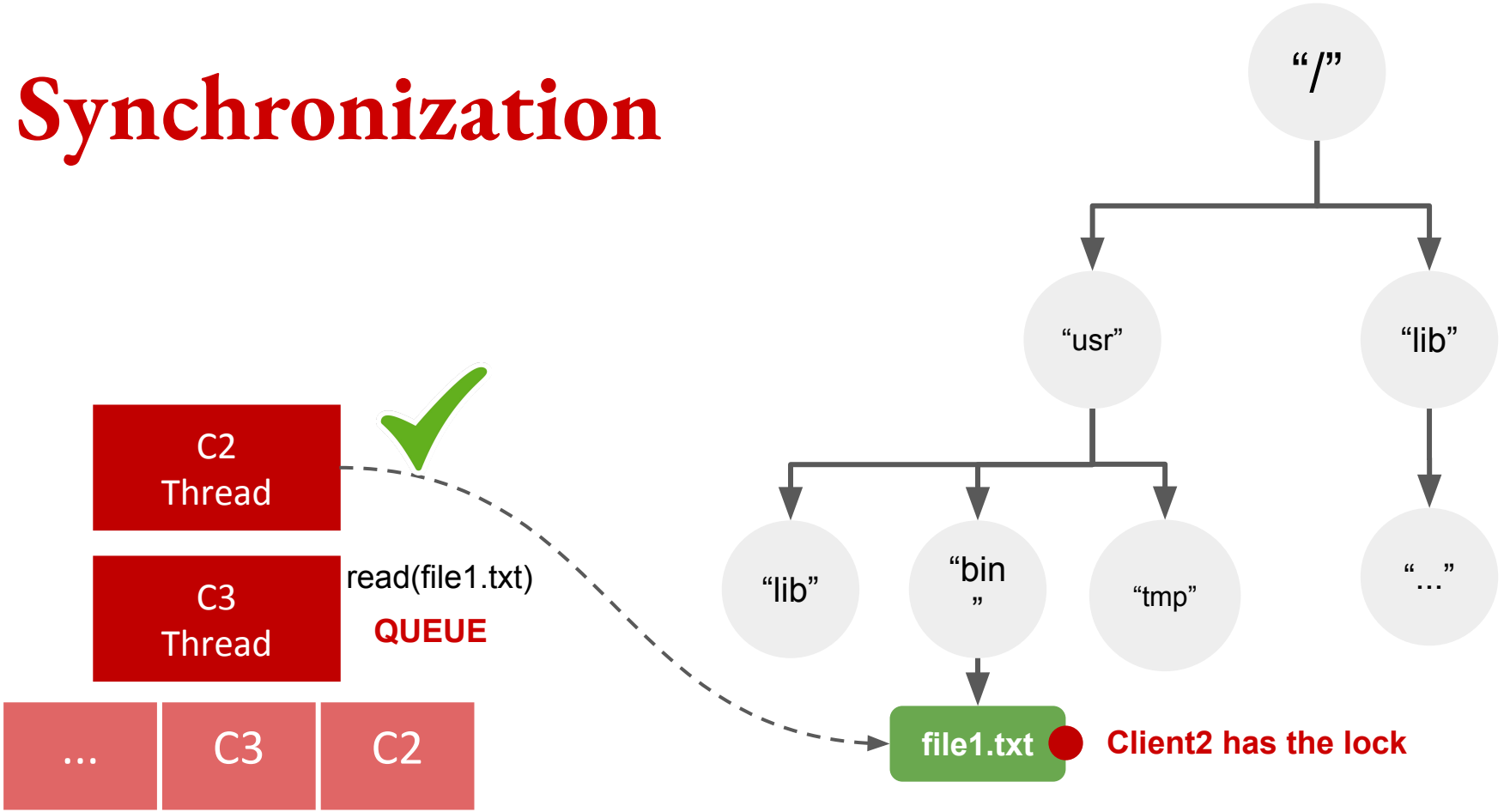
Synchronization



Synchronization

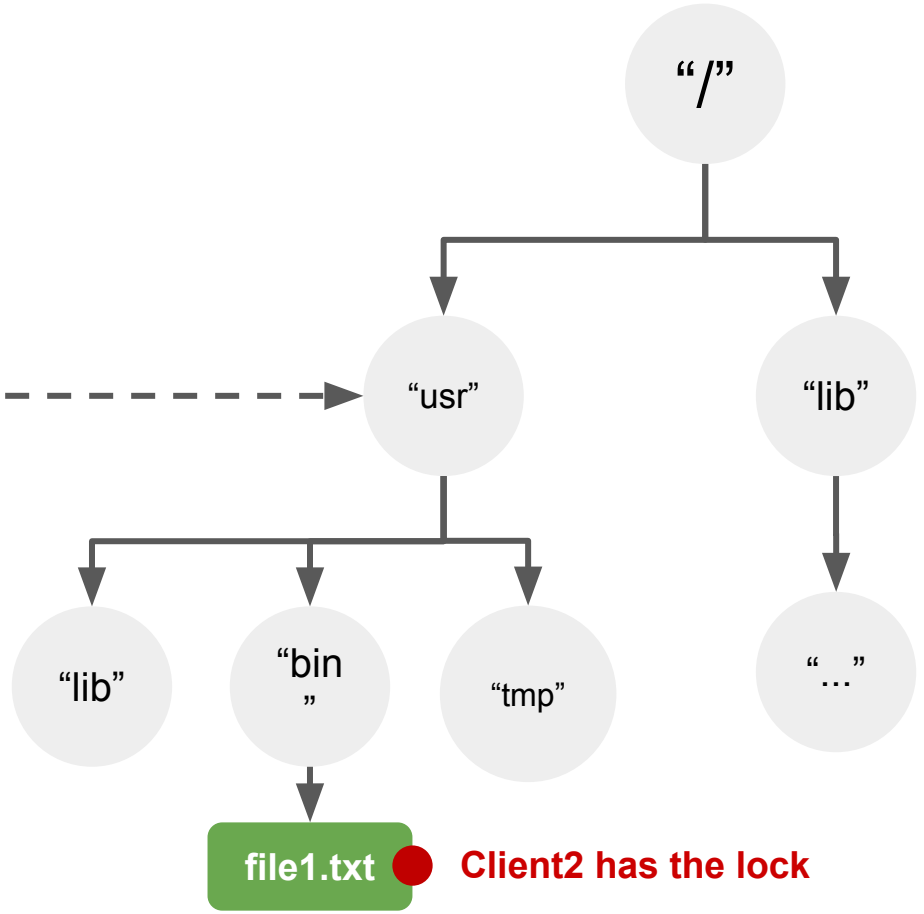


Synchronization



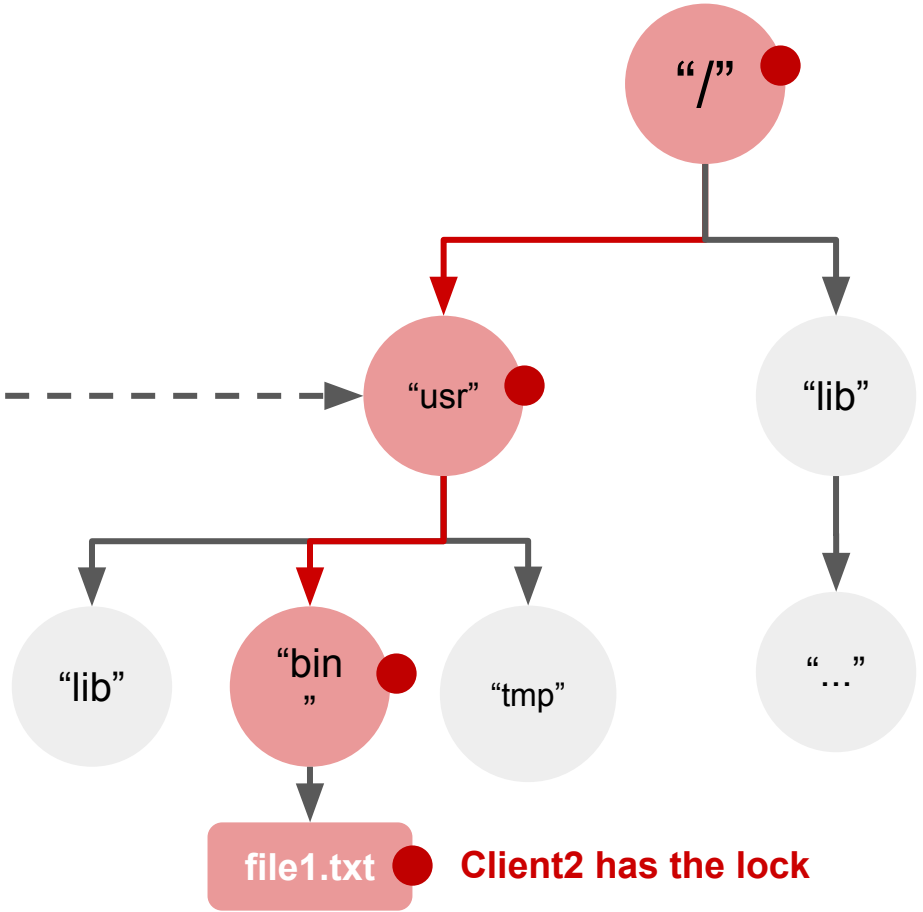
Synchronization

Client3 is Renaming?
SS is registering?

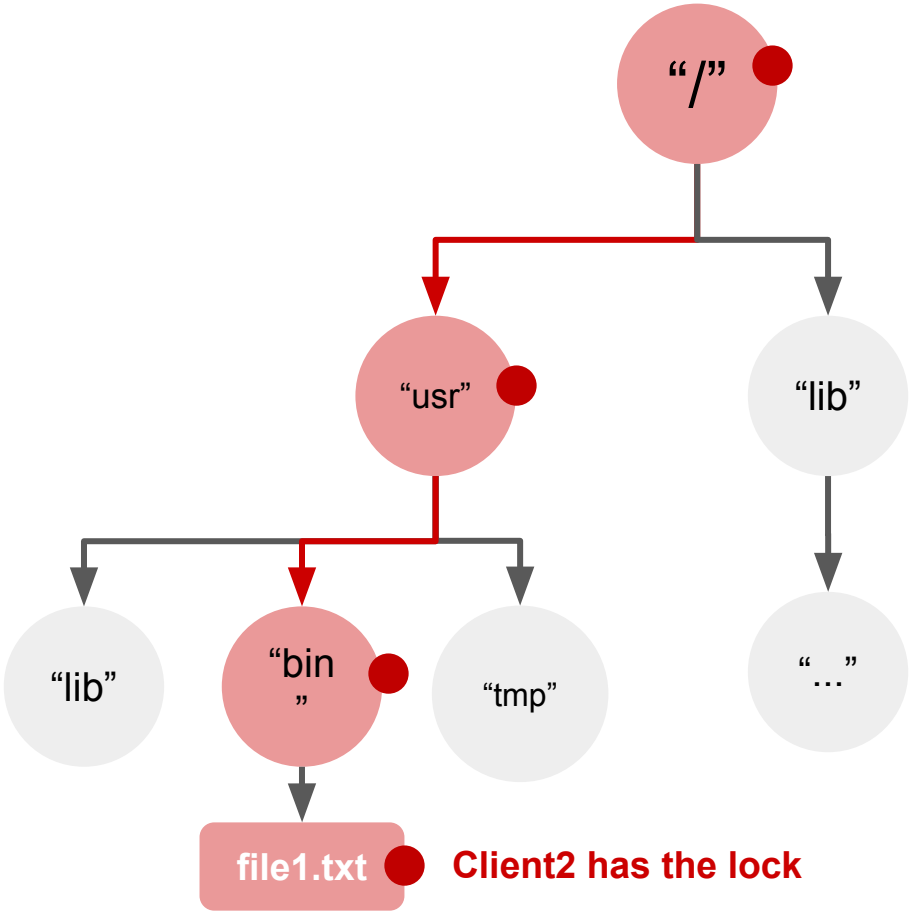


Synchronization

Client3 is Renaming?
SS is registering?

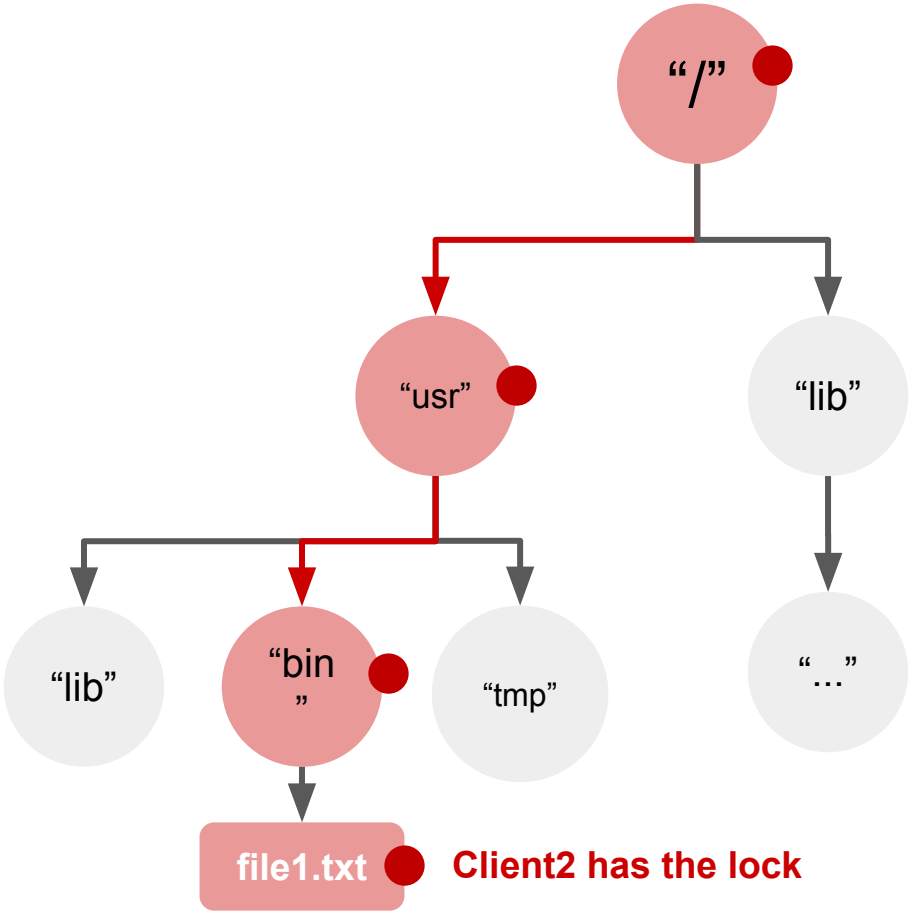


Synchronization



Synchronization

How are locks implemented?

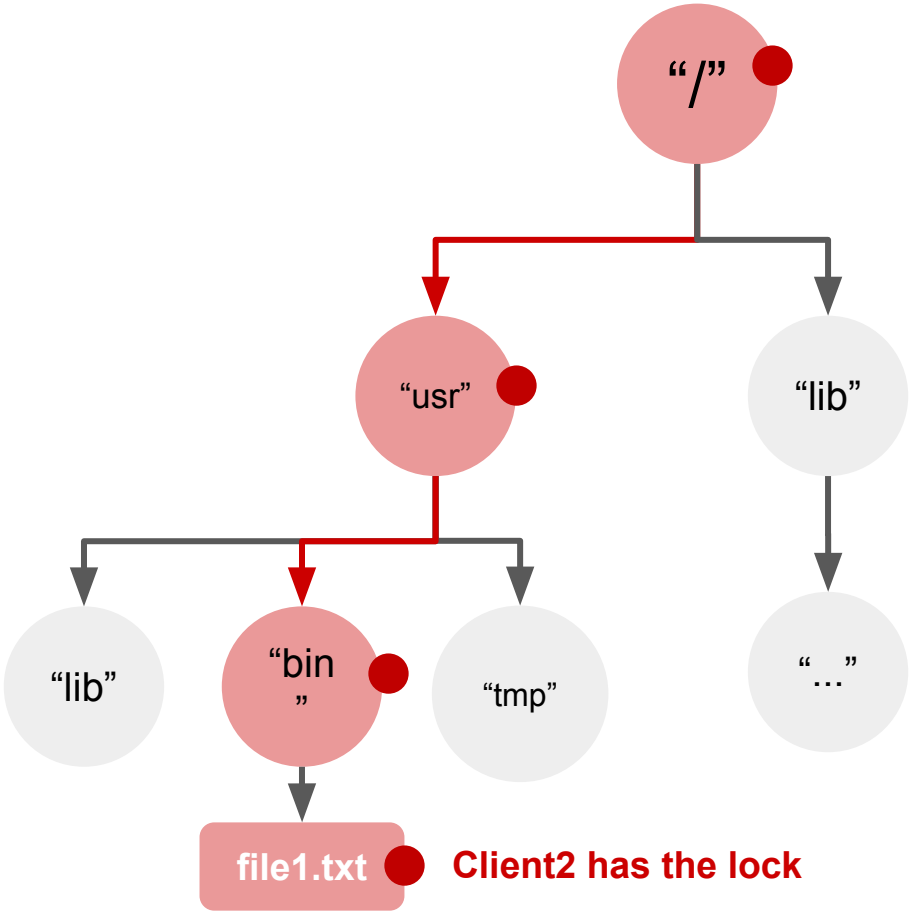


Synchronization

How are locks implemented?

```
Node {
        synchronized obtain_lock() {
                ...
                wait();
        }

        synchronized release_lock() {
                ...
                notifyAll();
        }
}
```



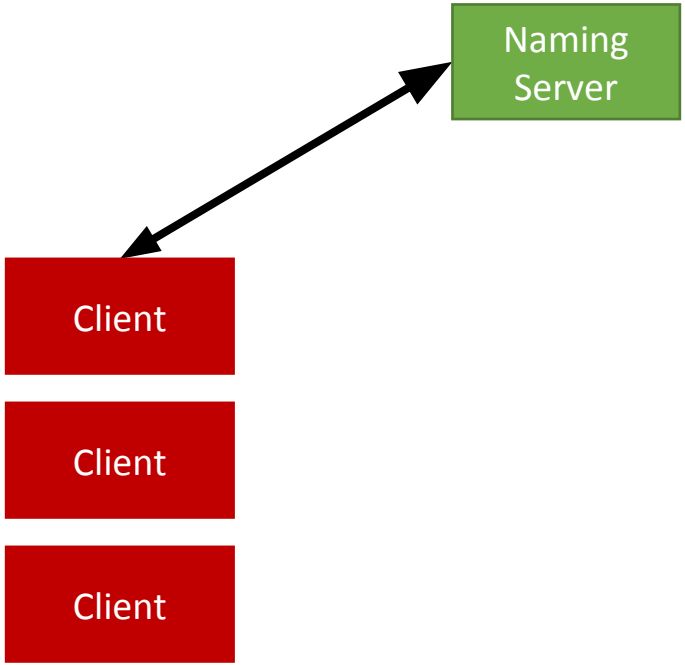
Synchronization

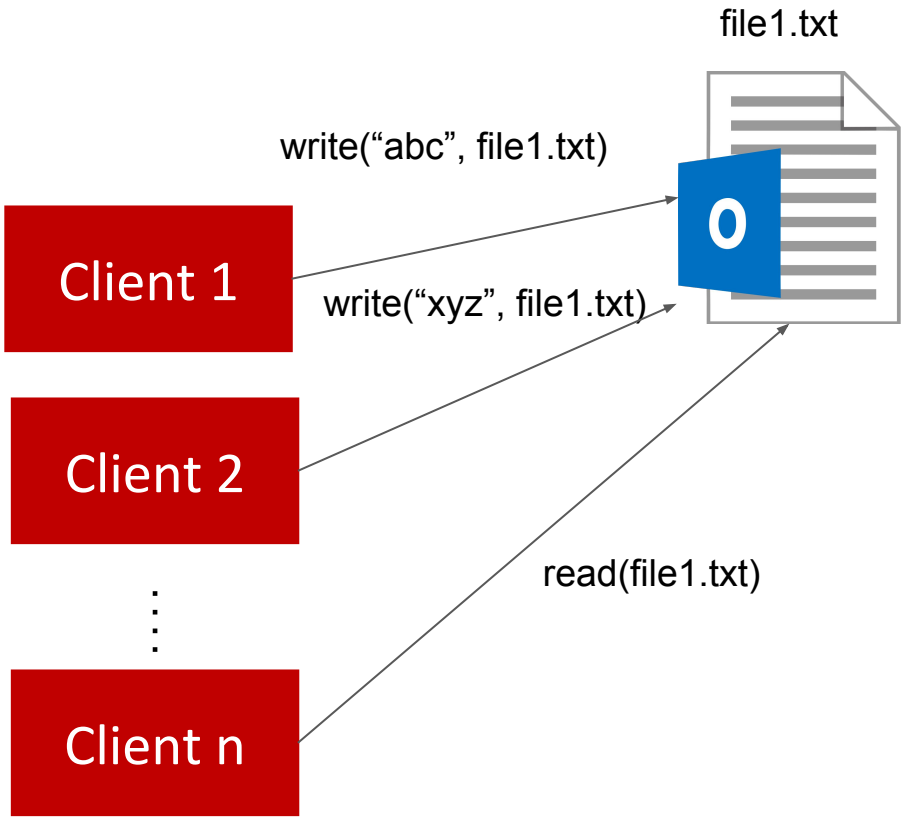
Service Interface

Implements two new methods

:

LOCK(path, lock_type);
UNLOCK(path, lock_type);





What might go wrong?

- Synchronization
- Load-balancing
- Consistency

Project 2 Objectives

Load Balancing

HOT FILES

Frequently Accessed

How do we identify them?

COLD FILES

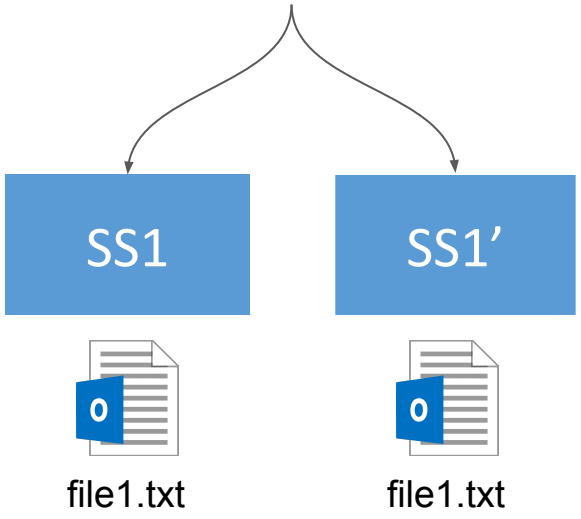
Rarely Accessed

Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



num_replicas = ALPHA * num_requesters

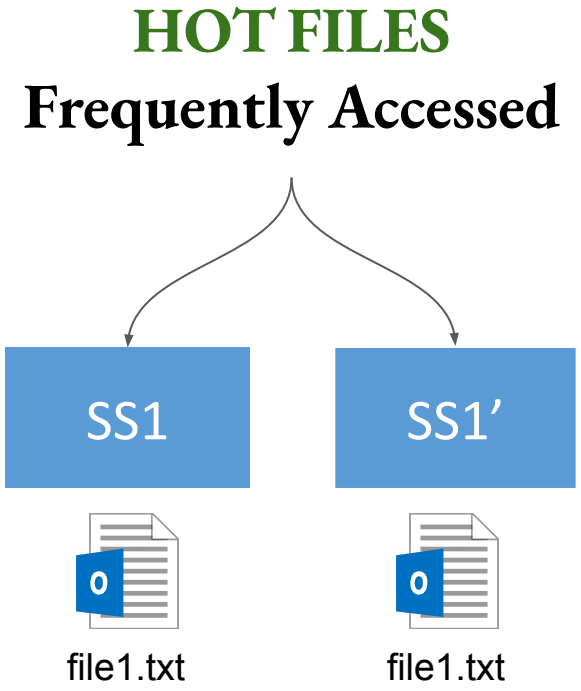
num_replicas = min (ALPHA * num_requesters , REPLICA_UPPER_BOUND)

num_requesters_coarse = {N | N >= num_requesters & a m ul ti pl e o f 20}

num_replicas = min (ALPHA * num_requesters_coarse , REPLICA_UPPER_BOUND)

Load Balancing

How are we scaling?

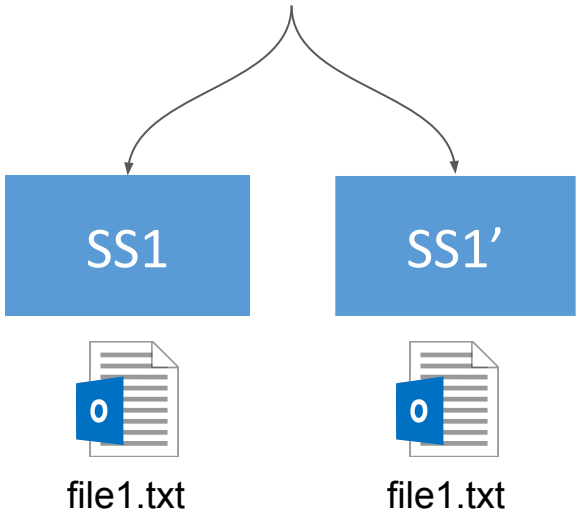


Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



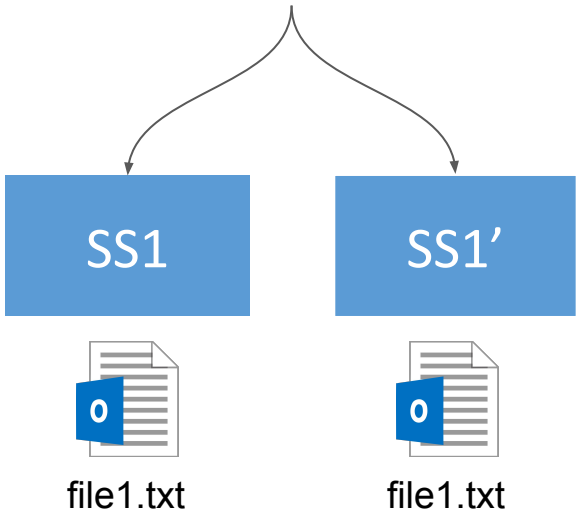
num_replicas = ALPHA * num_requesters

Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



num_replicas = ALPHA * num_requesters

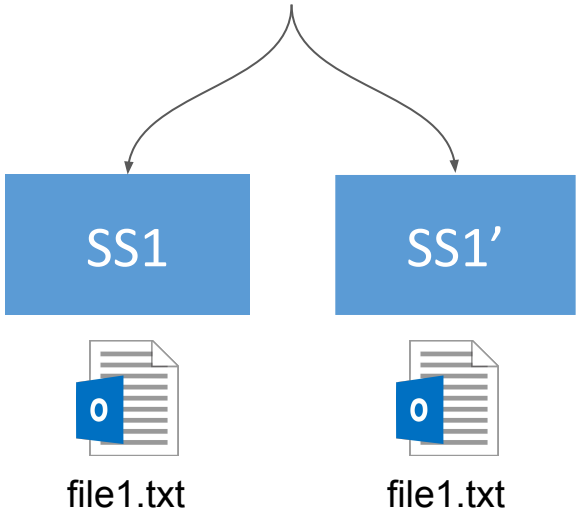
num_replicas = min (ALPHA * num_requesters , REPLICA_UPPER_BOUND)

Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



num_replicas = ALPHA * num_requesters

num_replicas = min (ALPHA * num_requesters , REPLICA_UPPER_BOUND)

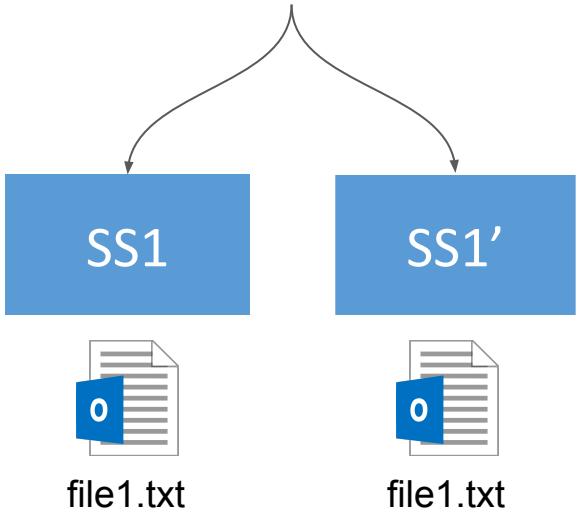
num_requesters_coarse = {N | N >= num_requesters & a m ul ti pl e o f 20}

Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



num_replicas = ALPHA * num_requesters

num_replicas = min (ALPHA * num_requesters , REPLICA_UPPER_BOUND)

num_requesters_coarse = {N | N >= num_requesters & a m ul ti pl e o f 20}

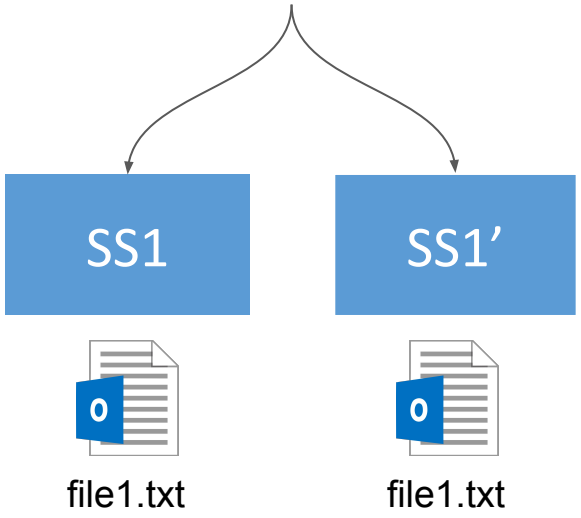
num_replicas = min (ALPHA * num_requesters_coarse , REPLICA_UPPER_BOUND)

Load Balancing

How are we scaling?

HOT FILES

Frequently Accessed



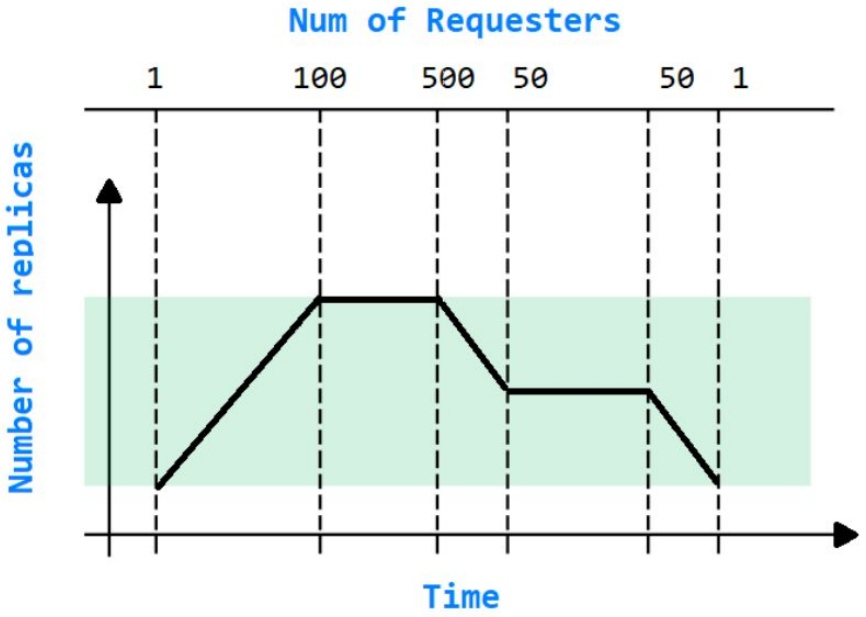
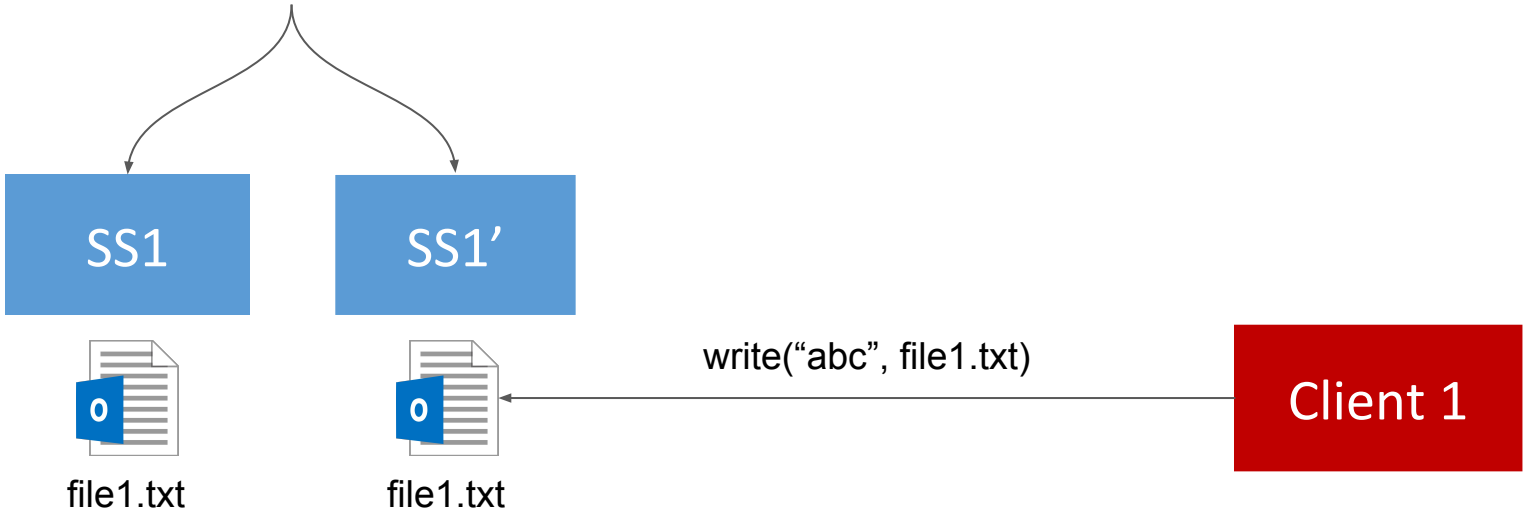


Figure 2: Linear Replication Policy with an upper-bound

Load Balancing

HOT FILES

Frequently Accessed

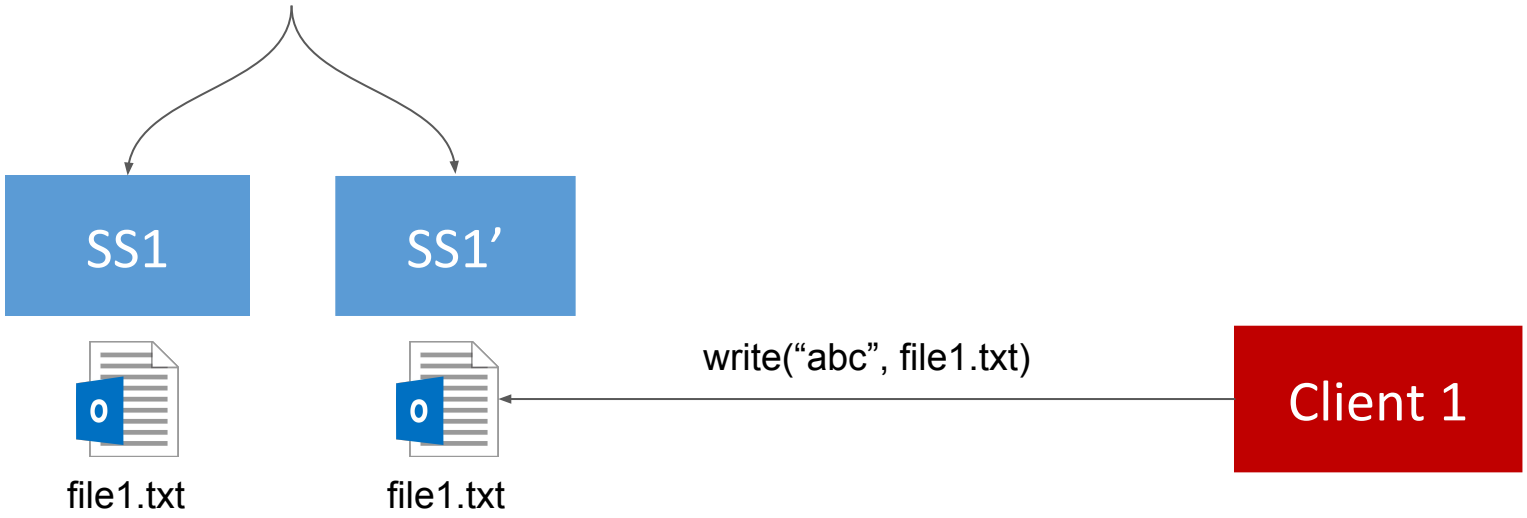


Load Balancing

CONSISTENCY

HOT FILES

Frequently Accessed



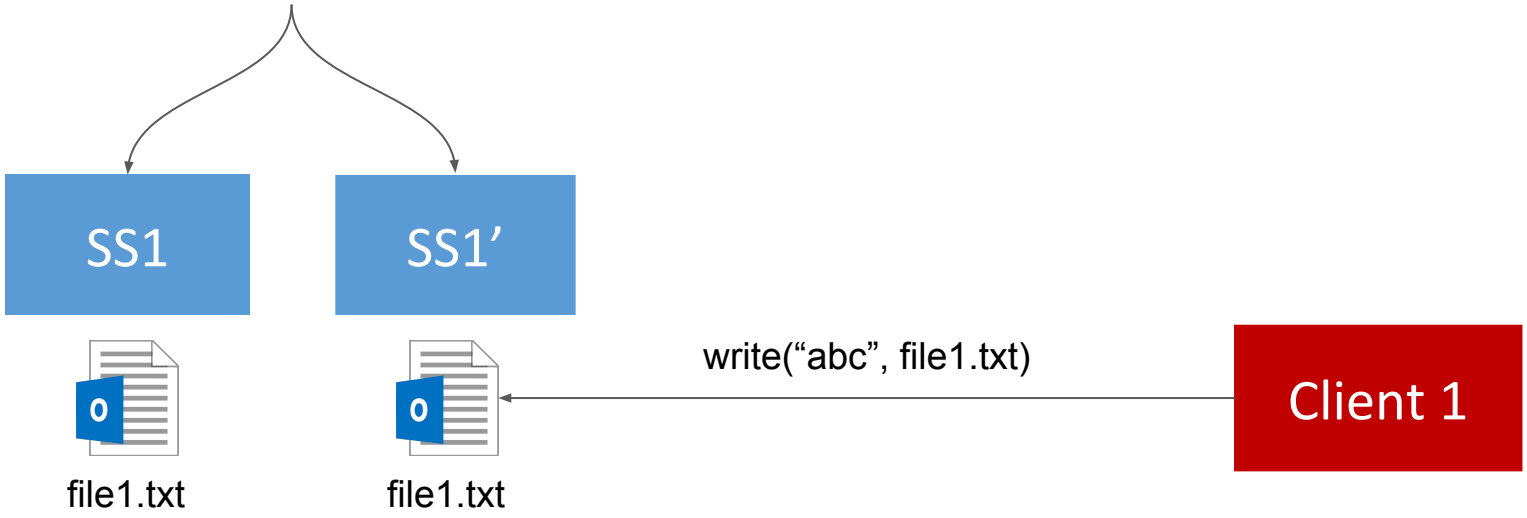
Load Balancing

CONSISTENCY

REDIRECTION

HOT FILES

Frequently Accessed



Load Balancing

CONSISTENCY

REDIRECTION

WRITE REQUESTS INVALIDATION

HOT FILES

Frequently Accessed