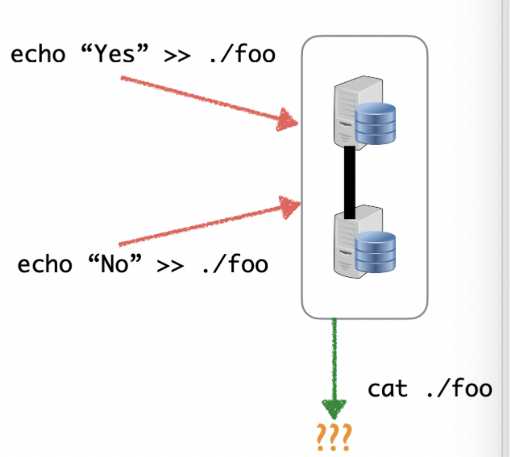
CAS CS 350
Lec 7

Google File System (GFS)

1. Why does Google need it?
    a. Handle huge amount of data and failures
    b. GFS → one of first design to exchange performance and consistency
2. MapReduce
    a. Storage is provided by inexpensive IDE disks attached directly to individual machines. A distributed file system developed in-house is used to manage the data stored on these disks. The file system uses replication to provide availability and reliability on top of unreliable hardware
    b. The more web page point to other webpages, the more popular (collect link among webpages to create a graph where nodes are webpages)
3. Topic: scalability, fault-tolerance, and data consistency
4. Consistency
    a. A correctness condition
    b. Inconsistent → read/write the data at the same time
    c. Important when data is replicated and concurrently accessed by applications
        i. If a write is performed, what value then will later read observe?
        ii. What if two writes are performed concurrently?
        iii. What if the reader is different from the writers?
    d. Data's consistency model defines what is correct
    e. 
```
echo “Yes” >> ./foo
echo “No” >> ./foo
cat ./foo
```

    f. Reason why you store multiple replication of same data → data management (split the work), ensure that server is available even after one replication is down (fault tolerance)

g. Strong consistency
   i. Read always returns the value of the most recent write

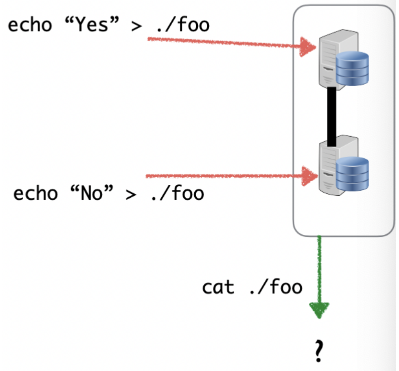
h. Weak consistency
   i. Reads may return “stale” data (not the value of the most recent write)
   ii. Implement weak consistency due to performance
   iii. The benefits we get from using weak consistency is great

i. In a weak consistency model, inconsistency is … OK!

5. Long History of Consistency
   a. Decades of development from architecture, operating systems, database, and distributed communities:
      i. Concurrent processors with private caches accessing a shared memory
      ii. Concurrent clients accessing a file system
      iii. Concurrent transactions on distributed database

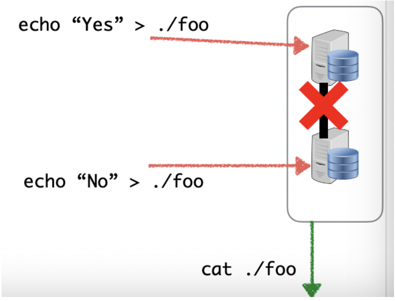
6. Consistency Today?
   a. Important data maybe replicated to prevent it from being lost due to a failure (durability)
   b. Important services maybe replicated to handle increasing demand, or guarantee availability in the case of a failure
   c. Any mutations of a replicated value has the potential to introducing inconsistencies

7. Sources of inconsistency
   a. Multiple threads, multiple clients



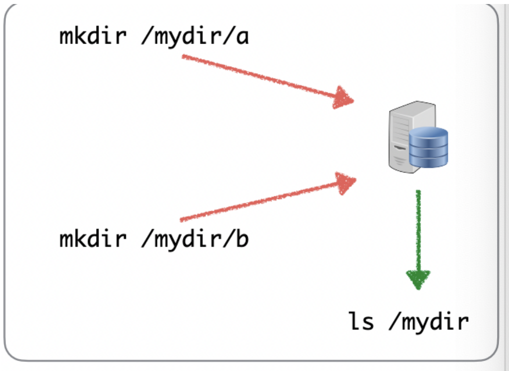
```
echo “Yes” > ./foo
echo “No” > ./foo
cat ./foo
```

   b.
   c. Concurrency!



```
echo “Yes” > ./foo
echo “No” > ./foo
cat ./foo
```

   d.
   e. Network failure or machine failure occurs

8. “Ideal” correctness guarantees
    a. In ideal world, replicated system behaves like a non-replicated system
    b. 

        i. “Single copy” serialized access
        ii. After a write, all reads will always see update value
        iii. Concurrent writes to same file independently done in a well defined order