

# Concurrent events and consistency control

## 1 Concurrency

- **Shared Object** - May be simultaneously accessed by multiple events
- A server which manages shared objects is responsible for ensuring the objects remain consistent when accessed by concurrent events
- Such a control is called concurrency control

Extension to replication - If event T happens before event U in their conflicting access to objects at one of the servers then they must be in that order at all of the servers whose objects are accessed in a conflicting manner by both T and U

## 2 Locking

Grant a lock

- Locks on an object are held (in a server) locally
- Local lock manager can decide whether to grant a lock or make the requesting transaction wait

Issues with Distributed Transaction (DT)

- A DT acquires resource located at different servers
- Can't release any locks until the transaction has been committed or aborted at all servers involved in the transaction
- Objects remain locked and are unavailable for all other transactions during the atomic commit protocol

## 3 Timestamp ordering

Timestamp:

- Assign a timestamp to each transaction when it starts
- Serial equivalence is enforced by committing the versions of objects in the order of the timestamps of transactions that accessed them
- Requirement: Globally unique timestamps

Distributed Transaction:

- Servers of distributed transactions are jointly responsible for ensuring that they are performed in a serially equivalent manner
- To achieve the same ordering at all the servers, the coordinators must agree as to the ordering of their timestamps
- Issue - In practice, a timestamp is usually assigned by a local server, generating a timestamp, server id pair

## 4 Concurrent Operations

In a single machine:

- Concurrent operations (events), no matter originated from different or the same machine, are handled by the time sharing feature of an operating system
- Operations are implicitly executed one by one in a series under a single clock, i.e. order of operations is well-defined

In a distributed system:

- Concurrent operations may run on different machines, i.e. distributed transactions
- Order of operations can't be easily sorted out due to overlapping operations, clock synchronisation problems, network latency and message loss etc

## 5 Deadlock Detection