Light-weight Leases for Large Scale Coordination

Dahlia Malkhi
The Hebrew University of Jerusalem

Joint work with Grisha Chockler
Formerly at Hebrew U and IBM research
Presently at MIT
A storage system

Agreement

Exclusive access
Adding a SAN manager

consensus manager

lease manager
A distributed SAN manager

consensus manager

lease manager

IBM Storage Seminar
03
Dahlia Malkhi
Our approach: Storage-centric
Advantages

- Data management is exactly where the data itself is placed
  - As available as the data
  - No bottleneck servers
  - Autonomous management of each data object
- Minimal requirements on storage servers
- Suitable for large scale internet services
A storage system

lease manager

web
The **storage centric** approach to high-availability internet services

- Shared object, manipulated by clients
- Replicas of the object reside in each server
- Client coordination is done using three phase commit
- Several important features:
  - No server monitoring
  - Freedom of choice per object of:
    - replication group
    - failure threshold
    - quorum system
    - and more
  - Replicas are not aware of each other
  - Messages are sent only to the relevant servers
Shared memory mutual exclusion

- Possible from a single **strong** object (e.g., compare&swap)
  - Not suitable for generic storage servers
  - Impossible for distributed/replicated storage with faults
- Possible from read/write registers for \( n \) clients (e.g., Bakery)
  - Does not scale with \( n \)
- Possible from a single read/write register using **timing**
Fischer’s mutual exclusion

1: do
2: read x into temp
3: until temp == no-proc
4: x := p
5: wait delta
6: if (x != p) goto 1
--- Critical Section ---
7: x := no-proc

No fault tolerance
Adding expiration

1: do
2: read x into temp
3: until temp == <no-proc, 0> or temp has not changed for Omega
4: x := <p, counter++>
5: wait delta
6: if (x != <p, *>) goto 1
--- Critical Section ---
7: x := <no-proc, 0>
Renewal

1: do
2: read x into temp
3: until temp == <no-proc, 0> or temp has not changed for Omega

3: if (x != <p, counter> return false
4: x := <p, counter++>
5: wait delta
6: if (x != <p, *> ) goto 1 return false
--- Critical Section ---
7: x := <no-proc, 0>
Lease properties

- Recoverable
- Renewable
- Uniform

Preliminary performance analysis:

- No contention: 1 read + 1 write + 1 delta
- $N$ contending processes: the above + log $N$
- Probability of safety violation: $\ln \left( \frac{1}{1 - e^{-\text{delay}}} \right)$
The ΠΑΞΟΣ Approach [Lamport]

- Assume a weak leader election primitive
  - Eventually there is a unique leader
    - $\Omega$ failure detector, partially synchronous/timed asynchronous systems, etc.

- To order operations, the leader invokes an instance of the agreement protocol
  - Never disagree on the operation order
  - Might fail to make progress if there is no unique leader
Leases: a fundamental building-block in Agreement

- Reliable shared object
- Coordination
- Fail-prone storage units
The end.