DISTRIBUTED CONSENSUS-
PART 4: (DYNAMIC)
CENTRALIZATION

Instructor: Prasun Dewan (FB 150, dewan@unc.edu)
(Semi) Synchronous Consistency Scenarios (Last Slide of Part 3)

- Multiple client UIs commit to single server
  - Browser-Sakai ✓
- Nested transaction involving multiple logical servers
  - Travelocity (non replicated) ✗
- Physical replication with multiple changers
  - Diff-based with divergence (Git) ✗
  - Snapshot-based (Google Drive, OneDrive) ✗
  - Command-based: replicated state machines (Google Docs, LiveMeeting) ✗
- Lock and other meta/configuration state
  - Live Meeting ✗
- Physical mirroring
  - Akamai ✓
- Master (primary)-slave (backup) replication ✓
- Master-master replication
  - Disjoint writes
  - Overlapping writes ✗
MULTI PROPOSING GENERAL

Dawn

Dusk
COORDINATING: MULTI PROPOSING GENERAL
SYNCHRONOUS

Acceptors

Proposers

Learners
Centralized-Synchronous

Proposers

Acceptors

Learners
Centralized-Asynchronous
ASYNCHRONOUS CENTRALIZED ALGORITHM

<table>
<thead>
<tr>
<th>Value</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td>29</td>
<td>Consensus</td>
</tr>
</tbody>
</table>

1. **Meaning of Life**
2. **Propose (1, 42)**
3. **rPropose (1, 42)**
4. **rPropose (1, 29)**
5. **Learn (1, 42)**
6. **Learn (1, 29)**

**Propose (2, 29)**
SYNCHRONOUS CENTRALIZED ALGORITHM

Meaning of Life

<table>
<thead>
<tr>
<th>Value</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Pending</td>
</tr>
<tr>
<td>0</td>
<td>Pending</td>
</tr>
<tr>
<td>0</td>
<td>Pending</td>
</tr>
<tr>
<td>0</td>
<td>Pending</td>
</tr>
</tbody>
</table>

1. propose(1,42)
2. rPropose(1,42)
3. rPropose(1,29)
4. accept(1,42)
5. accept(1,29)
6. propose(2,29)
Concurrency Control

- **Non atomic**
  - put \((k^1, v^1)\), put \((k^2, v^2)\)
  - put \((k^1, v^1)\), put \((k^1, v^2)\), put \((k^1, v^1)\), del \((k^1)\)

- **Atomic**
  - put \((k^1, v^1)\), put \((k^2, v^2)\)

- **Serializable**
  - put \((k^1, v^1)\), put \((k^2, v^2)\), put \((k^1, v^1)\), del \((k^1)\)

Type-independent Concurrency Control
**Synchronous Centralized Algorithm**

(Review)

![Diagram showing the algorithm's state transitions and decision-making processes.](image-url)

- **Meaning of Life**
  - **Value**
    - 0
  - **State**
    - Pending

- **States and Transitions**
  - **1**
    - Value: 0
    - State: Pending
    - Transition: propose(1,42) → accept(1,42)
    - Decision: Should second pending proposal be accepted?
    - Outcome: Atomic!
  - **2**
    - Value: 0
    - State: Pending
    - Transition: propose(2,29) → accept(1,29) → rPropose(1,29) → accept(1,29)
    - Decision: Should second pending proposal be accepted?
    - Outcome: Not serializable
**Concurrency Control**

- **Non atomic**
  - put \((k^1, v^1)\)  put \((k^2, v^2)\)
  - \(C^1\) \(\rightarrow\) \(S\) \(\leftarrow\) \(C^2\)
  - \(C^1\) \(\rightarrow\) \(S^1\) \(\rightarrow\) \(S^1^{12}\)
  - \(C^2\) \(\rightarrow\) \(S^2\) \(\rightarrow\) \(S^2^{21}\)

- **Atomic**
  - put \((k^1, v^1)\)  put \((k^1, v^2)\)  put \((k^1, v^1)\)  del \((k^1)\)
  - \(C^1\) \(\rightarrow\) \(S\) \(\leftarrow\) \(C^2\)
  - \(C^1\) \(\rightarrow\) \(S^1\) \(\rightarrow\) \(S^{12}\)

- **Serializable**
  - \(C^1\) \(\rightarrow\) \(S^1\)
public enum ConcurrencyKind {
    NON_ATOMIC,
    ATOMIC,
    SERIALIZABLE
}
protected void dispatchPropose(float aProposalNumber, StateType aProposal) {
    if (!isCentralized()) {
        localPropose(aProposalNumber, aProposal);
    } else {
        sendProposeRequest(aProposalNumber, aProposal);
    }
}

public void remotePropose(float aProposalNumber, StateType aProposal) {
    recordProposalState(aProposalNumber, aProposal);
    if (someOtherProposalIsPending(aProposalNumber) && isSerializable()) {
        sendLearnNotification(aProposalNumber, aProposal, ProposalFeedbackKind.CONCURRENCY_CONFLICT);
        return;
    }
    localPropose(aProposalNumber, aProposal);
}
**Server Proposer 1**

---

**I*** (ProposalMade) Meaning, 1.0001=42
**I*** (RemoteProposeRequestSent) Meaning, 1.0001=42
**I*** (RemoteProposeRequestReceived) Meaning, 1.0001=42
**I*** (ProposalAcceptRequestSent) Meaning, 1.0001=42
**I*** (ProposalWaitStarted) Meaning, 1.0001=42
**I*** (ProposalAcceptRequestReceived) Meaning, 1.0001=42
**I*** (ProposalAcceptedNotificationSent) Meaning, 1.0001=42
**I*** (RemoteProposeRequestReceived) Meaning, 1.0002=29
**I*** (ProposalLearnNotificationSent) Meaning, 1.0002=29
**I*** (ProposalLearnNotificationReceived) Meaning, 1.0002=29
**I*** (ProposalStateChanged) Meaning, 1.0002=29
**I*** (ProposalAcceptResponseReceived) Meaning, 1.0001=42
**I*** (SufficientAgreementsChecked) Meaning, 1.0001, 42:1|1|4|4?4.0--null
**I*** (ProposalAcceptResponseReceived) Meaning, 1.0001=42
**I*** (SufficientAgreementsChecked) Meaning, 1.0001, 42:2|2|4|4?4.0--null
**I*** (ProposalAcceptResponseReceived) Meaning, 1.0001=42
**I*** (SufficientAgreementsChecked) Meaning, 1.0001, 42:3|3|4|4?4.0--null
**I*** (ProposalAcceptResponseReceived) Meaning, 1.0001=42
**I*** (SufficientAgreementsChecked) Meaning, 1.0001, 42:4|4|4?4.0--true
**I*** (ProposalLearnNotificationSent) Meaning, 1.0001=42
**I*** (ProposalLearnNotificationReceived) Meaning, 1.0001=42
**I*** (ProposalWaitEnded) Meaning, 1.0001=42
**I*** (RemoteProposeRequestReceived) Meaning, 2.0002=29
**I*** (ProposalAcceptRequestSent) Meaning, 2.0002=29
**I*** (ProposalAcceptResponseReceived) Meaning, 2.0002=29
**I*** (ProposalLearnNotificationSent) Meaning, 2.0002=29
**I*** (ProposalLearnNotificationReceived) Meaning, 2.0002=29
**I*** (ProposalStateChanged) Meaning, 2.0002=29
**I*** (ProposalAcceptResponseReceived) Meaning, 2.0002=29
**I*** (ProposalAcceptedNotificationSent) Meaning, 2.0002=29
**I*** (ProposalStateReleased) Meaning, 2.0002=29

Meaning of Life: 42
REJECTED PROPOSER 2

I**(ProposalMade) Meaning, 1.0002=29
I**(RemoteProposeRequestSent) Meaning, 1.0002=29
I**(ProposalWaitStarted) Meaning, 1.0002=29
I**(ProposalAcceptRequestReceived) Meaning, 1.0001=42
I**(ProposalAcceptedNotificationSent) Meaning, 1.0001=42->SUCCESS
I**(ProposalLearnNotificationReceived) Meaning, 1.0002=29->CONCURRENCY_CONFLICT
I**(ProposalStateChanged) Meaning, 1.0002=29->PROPOSAL_CONCURRENT_OPERATION
I**(ProposalWaitEnded) Meaning, 1.0002=29->PROPOSAL_CONCURRENT_OPERATION

I**(ProposalMade) Meaning, 2.0002=29
I**(RemoteProposeRequestSent) Meaning, 2.0002=29
I**(ProposalWaitStarted) Meaning, 2.0002=29
I**(ProposalLearnNotificationReceived) Meaning, 1.0001=42->SUCCESS
I**(ProposalStateChanged) Meaning, 1.0001=42->PROPOSAL_CONSENSUS
Meaning of Life: 42
I**(ProposalAcceptRequestReceived) Meaning, 2.0002=29
I**(ProposalAcceptedNotificationSent) Meaning, 2.0002=29->SUCCESS
I**(ProposalLearnNotificationReceived) Meaning, 2.0002=29->SERVICE_DENIAL
I**(ProposalStateChanged) Meaning, 2.0002=29->PROPOSAL_SERVICE_DENIAL
I**(ProposalWaitEnded) Meaning, 2.0002=29->PROPOSAL_SERVICE_DENIAL
Accepting Acceptor 3 (Synchronous)

I*** (ProposalAcceptRequestReceived) Meaning, 1.0001=42
I*** (ProposalAcceptedNotificationSent) Meaning, 1.0001=42 --> SUCCESS
I*** (ProposalLearnNotificationReceived) Meaning, 1.0002=29 --> CONCURRENCY_CONFLICT
I*** (ProposalStateChanged) Meaning, 1.0002=29 --> PROPOSAL_CONCURRENT_OPERATION
I*** (ProposalLearnNotificationReceived) Meaning, 1.0001=42 --> SUCCESS
I*** (ProposalStateChanged) Meaning, 1.0001=42 --> PROPOSAL_CONSENSUS
Meaning of Life: 42
I*** (ProposalAcceptRequestReceived) Meaning, 2.0002=29
I*** (ProposalAcceptedNotificationSent) Meaning, 2.0002=29 --> SUCCESS
I*** (ProposalLearnNotificationReceived) Meaning, 2.0002=29 --> SERVICE_DENIAL
I*** (ProposalStateChanged) Meaning, 2.0002=29 --> PROPOSAL_SERVICE_DENIAL
REJECTING ACCEPTOR 4 (SYNCHRONOUS)

```plaintext
I*** (ProposalAcceptRequestReceived) Meaning, 1.0001=42
I*** (ProposalAcceptedNotificationSent) Meaning, 1.0001=42--->SUCCESS
I*** (ProposalLearnNotificationReceived) Meaning, 1.0002=29--->CONCURRENCY_CONFLICT
I*** (ProposalStateChanged) Meaning, 1.0002=29--->PROPOSAL_CONCURRENT_OPERATION
I*** (ProposalLearnNotificationReceived) Meaning, 1.0001=42--->SUCCESS
I*** (ProposalStateChanged) Meaning, 1.0001=42--->PROPOSAL_CONSENSUS

Meaning of Life: 42
I*** (ProposalAcceptRequestReceived) Meaning, 2.0002=29
I*** (ProposalAcceptedNotificationSent) Meaning, 2.0002=29--->SERVICE_DENIAL
I*** (ProposalLearnNotificationReceived) Meaning, 2.0002=29--->SERVICE_DENIAL
I*** (ProposalStateChanged) Meaning, 2.0002=29--->PROPOSAL_SERVICE_DENIAL
```
COORDINATING: MULTI PROPOSING GENERAL

Problems?
## Failure in Centralized Algorithm

<table>
<thead>
<tr>
<th>Meaning of Life</th>
<th>Value</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>Consensus</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>Consensus</td>
</tr>
</tbody>
</table>

### Diagram

- **Node 1**: propose(1, 42) → rPropose(1, 42) → rPropose(1, 29) → learn(1, 42) → learn(1, 29)

### Inconsistency!
- New sequencer may be arbitrarily behind
- Correct state is with at least one alive member
- That member may die in middle of switchover
- We could use the voice of more than one member

### Notes
- The meaning of life value is 29 and the state is Consensus.
SWITCHING CENTRAL REPLICAS

Consensus Problem
If A sends to B via C and B sends A via D, then C == D as central replica replica is unique

Need non centralized algorithm during switchover as no common central replica to solve consensus problem

Can use 0 phases.
Order the replicas by id and pick the next one in the list
Non centralized algorithm with replica proposing itself necessary if sorted list not sufficient
Killing Central Process

```java
public void remotePropose(float aProposalNumber, StateType aProposal) {
    RemoteProposeRequestReceived.newCase(this, getObjectName(), aProposalNumber, aProposal);
    recordProposalState(aProposalNumber, aProposal);
    if (someOtherProposalIsPending(aProposalNumber) && isSerializable()) {
        sendLearnNotification(aProposalNumber, aProposal, ProposalFeedbackKind.CONCURRENCY_CONFLICT);
        return;
    }
    localPropose(aProposalNumber, aProposal);
}
```
I*** (ProposalMade)  Meaning, 1.0001=42
I*** (RemoteProposeRequestSent)  Meaning, 1.0001=42
I*** (RemoteProposeRequestReceived)  Meaning, 1.0001=42
I*** (ProposalAcceptRequestSent)  Meaning, 1.0001=42
I*** (ProposalWaitStarted)  Meaning, 1.0001=42
I*** (ProposalAcceptRequestReceived)  Meaning, 1.0001=42
I*** (ProposalAcceptedNotificationSent)  Meaning, 1.0001=42-->SUCCESS
I*** (RemoteProposeRequestReceived)  Meaning, 1.0002=29
REJECTED PROPOSER 2

I**(ProposalMade)  Meaning, 1.0002=29
I**(RemoteProposeRequestSent)  Meaning, 1.0002=29

I**(ProposalWaitStarted)  Meaning, 1.0002=29
I**(ProposalAcceptRequestReceived)  Meaning, 1.0001=42
I**(ProposalAcceptedNotificationSent)  Meaning, 1.0001=42--->SUCCESS
AReadCommand: An existing connection was forcibly closed by the remote host
I**(ProposalStateChanged)  Meaning, 1.0002=29--->CENTRAL_SERVER_DIED
AReadCommand: An existing connection was forcibly closed by the remote host
I**(ProposalWaitEnded)  Meaning, 1.0002=29--->CENTRAL_SERVER_DIED

I**(Received left message : Host: dewan-t431s Name: 1 ID: 7001)
I**(ProposalMade)  Meaning, 2.0002=29
I**(RemoteProposeRequestSent)  Meaning, 2.0002=29
I**(RemoteProposeRequestReceived)  Meaning, 2.0002=29

I**(ProposalAcceptRequestSent)  Meaning, 2.0002=29
I**(ProposalWaitStarted)  Meaning, 2.0002=29
I**(ProposalAcceptRequestReceived)  Meaning, 2.0002=29
I**(ProposalAcceptedNotificationSent)  Meaning, 2.0002=29--->SUCCESS
I**(ProposalAcceptResponseReceived)  Meaning, 2.0002=29--->SERVICE_DENIAL

I**(ProposalLearnNotificationSent)  Meaning, 2.0002=29--->SERVICE_DENIAL
I**(ProposalLearnNotificationReceived)  Meaning, 2.0002=29--->SERVICE_DENIAL
I**(ProposalStateChanged)  Meaning, 2.0002=29--->PROPOSAL_SERVICE_DENIAL
I**(ProposalWaitEnded)  Meaning, 2.0002=29--->PROPOSAL_SERVICE_DENIAL
I**(ProposalAcceptResponseReceived)  Meaning, 2.0002=29--->SUCCESS
I**(ProposalAcceptResponseReceived)  Meaning, 2.0002=29--->SUCCESS
ACCEPTING ACCEPTOR 3 (SYNCHRONOUS)

I*** (ProposalAcceptRequestReceived) Meaning, 1.0001=42
I*** (ProposalAcceptedNotificationSent) Meaning, 1.0001=42-->SUCCESS

AReadCommand: An existing connection was forcibly closed by the remote host
AReadCommand: An existing connection was forcibly closed by the remote host
I*** Received left message : Host: dewan-t431s Name: 1 ID: 7001
I*** (ProposalAcceptRequestReceived) Meaning, 2.0002=29

I*** (ProposalAcceptedNotificationSent) Meaning, 2.0002=29-->SUCCESS
I*** (ProposalLearnNotificationReceived) Meaning, 2.0002=29-->SERVICE_DENIAL
I*** (ProposalStateChanged) Meaning, 2.0002=29-->PROPOSAL_SERVICE_DENIAL
REJECTING ACCEPTOR 4 (SYNCHRONOUS)

I*** (ProposalAcceptRequestReceived) Meaning, 1.0001=42
I*** (ProposalAcceptedNotificationSent) Meaning, 1.0001=42 --> SUCCESS

AReadCommand: An existing connection was forcibly closed by the remote host

I***Received left message : Host: dewan-t431s Name: 1 ID: 7001
I*** (ProposalAcceptRequestReceived) Meaning, 2.0002=29
I*** (ProposalAcceptedNotificationSent) Meaning, 2.0002=29 --> SERVICE_DENIAL
I*** (ProposalLearnNotificationReceived) Meaning, 2.0002=29 --> SERVICE_DENIAL
I*** (ProposalStateChanged) Meaning, 2.0002=29 --> PROPOSAL_SERVICE_DENIAL
Non centralized algorithm with replica proposing itself necessary if sorted list not sufficient

Different consensus states possible even if ids used for switchover

We could use the voice of more than one member