

Chapter 9: Distributed Transactions (First Part)

Distributed Transactions (1)



Introduction

Terminology

Atomic
Commitment

System
Architecture

Discussion

- „Transfer USD 500,--
from Klemens' account to Jim's account.“
- Two operations:
 - ◆ Increment Jim's balance.
 - ◆ Decrement Klemens' balance.

Distributed Transactions (2)

- New problems in presence of distribution: additional ,opportunities‘ for failures
 - ◆ site failure (one or several sites; partial, total),
 - ◆ loss of messages,
 - ◆ connections may fail, in particular, network falls apart into partitions, disjoint subnets.
- Important objective: ensure atomicity
→ commit protocols.

Introduction

Terminology

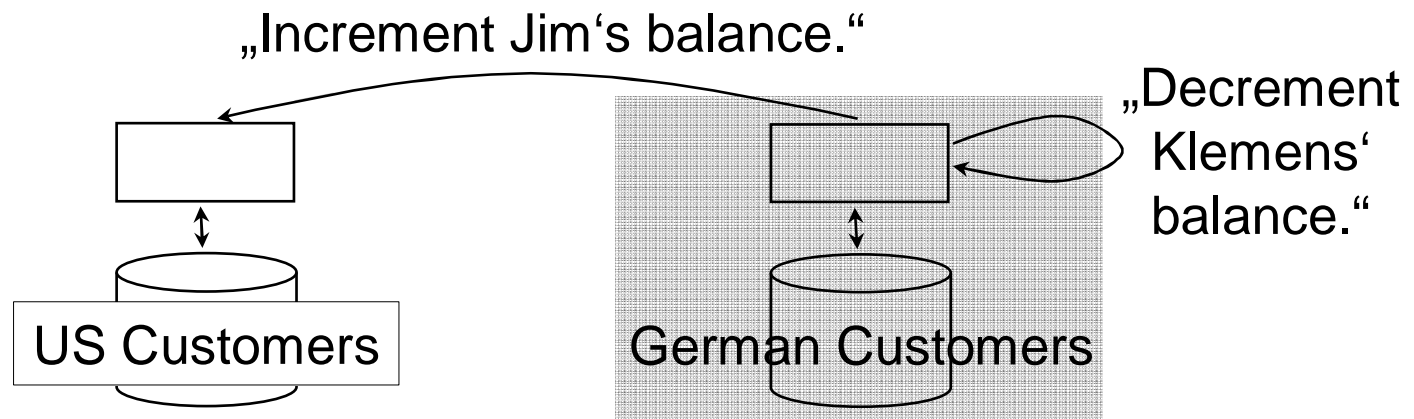
Atomic
Commitment

System
Architecture

Discussion

Structure of Distributed Transactions (1)

- In general, several nodes take part in execution,
- each transaction has **home node/coordinator node**: start of transaction, i.e., execution of BOT.



- Further DB operations and commit issued by home node as well, they may comprise other nodes as well.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Illustration



Introduction

Terminology

Atomic
Commitment

System
Architecture

Discussion

- „Transfer USD 500,-- from Klemens' account to Jim's account.“
- Four operations (slight extension of example):
 - ◆ Increment Jim's balance.
 - ◆ Record receipt of money from Klemens.
 - ◆ Decrement Klemens' balance.
 - ◆ Record transfer of money to Jim.

Structure of Distributed Transactions (2)

- **Local transaction:**
executed exclusively at home node.
- **Distributed or global transaction:**
other nodes as well.
- Nodes that participate execute **subtransaction**.
Subtransaction: all operations of a transaction that execute at the particular node.

- In previous slide: Operations
 - ◆ Increment Jim's balance.
 - ◆ Record receipt of money from Klemens.form a subtransaction.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

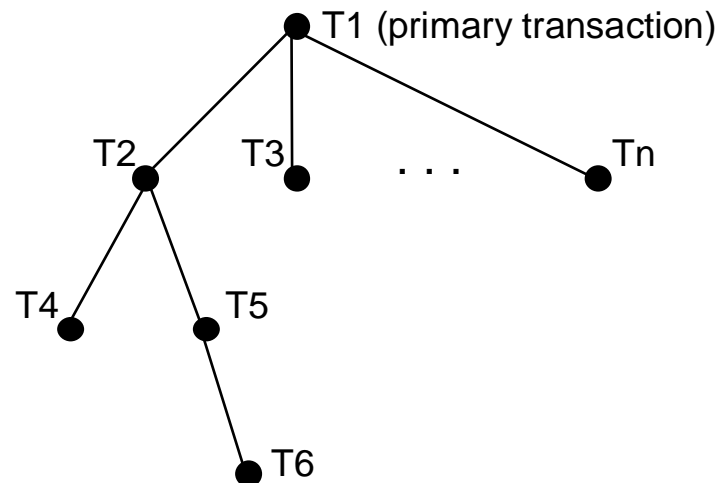
Structure of Distributed Transactions (3)

- **Primary transaction** (root transaction) – subtransaction executed at home node.
- Structure of invocation – directed graph, nodes = participating nodes/subtransactions.
- Invocation structure may contain cycles: DB operation needs data from nodes that have executed operations belonging to same TA before.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Structure of Distributed Transactions (4)

- For transaction management (not for synchronization) simplified representation is sufficient: hierarchically, **transaction tree**.
- Transaction tree
 - ◆ not balanced,
height only limited by number of nodes,
 - ◆ subtransactions may execute in parallel.



Introduction

Terminology

Atomic

Commitment

System

Architecture

Discussion

Structure of Distributed Transactions (5)

- **Abort** of a subtransaction resets entire transaction.

Introduction

Terminology

Atomic

Commitment

System

Architecture

Discussion

Z

Distributed Commit (1)

- All nodes taking part in global TA must agree regarding the **outcome of the commit**: either abort in *all* nodes or commit in *all* nodes.
- Main concern: **correctness**
all-or-nothing also in case of failures.

Introduction

Terminology

Atomic
Commitment

System

Architecture

Discussion

Distributed Commit (2)

- *Atomic Commitment Protocol*, overview:
 1. Coordinator inquires if all nodes are ready and willing to commit (*voting phase*),
 2. if yes: coordinator asks all nodes to commit (*decision phase*).

Problem: failures.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

2PC Basic Protocol

- Initially, each agent is entitled to „unilateral abort“;
- is given up by sending READY,
- agent is then obliged to accept global commit result and to realize it.

Introduction

Terminology

Atomic
Commitment

System

Architecture

Discussion

Overview

- 2PC (Two-Phase Commit Protocol)
- Terminology,
Variants of Atomic Commit Protocols – overview,
requirements,
- Commit structures.
- Next package of slides:
optimizations, 3PC.

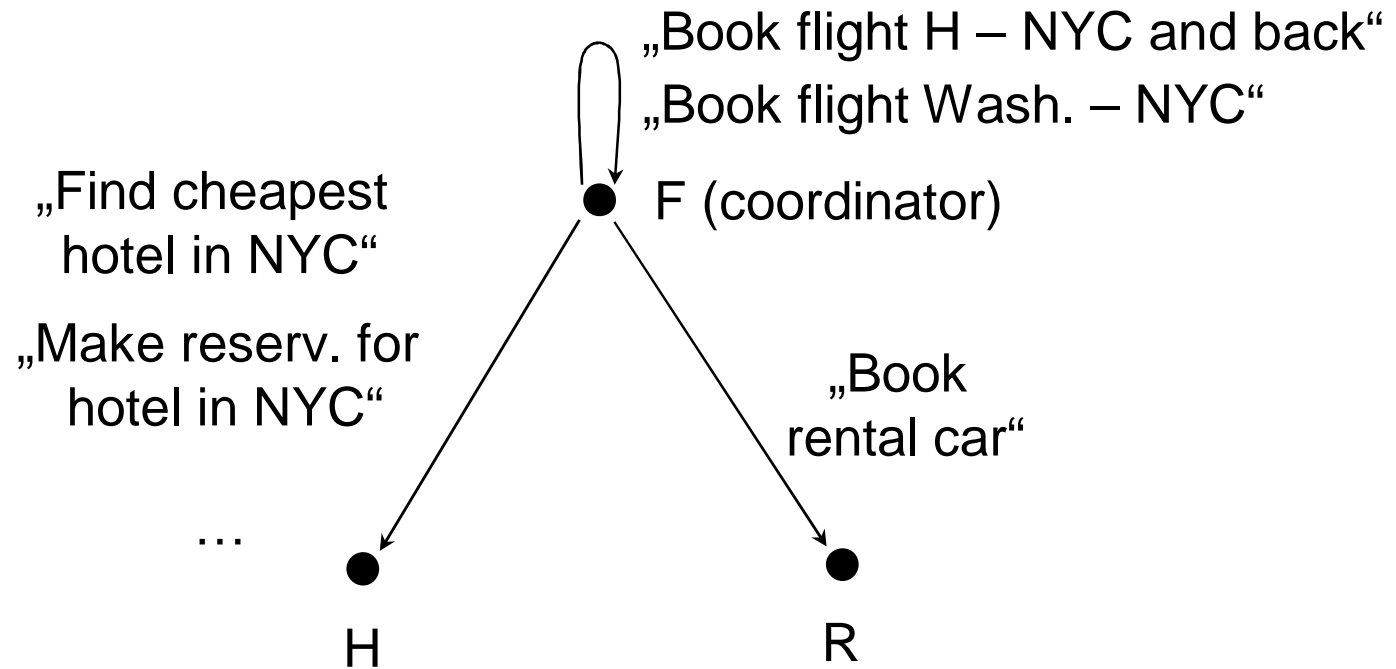
Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Scenario

- Transaction T that consists of several operations:
 - ◆ “Book flight from Hanover to New York City and back.”
 - ◆ “Book flight from Washington DC to NYC.”
 - ◆ “Find cheapest hotel in NYC.”
 - ◆ “Make reservation for it.”
 - ◆ “Make reservation for any hotel in Philadelphia.”
 - ◆ dto. Washington DC
 - ◆ “Book rental car NYC – Washington DC.”
- Flight DB (Node F)
- Hotel DB (Node H)
- Rental Car DB (Node R)

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

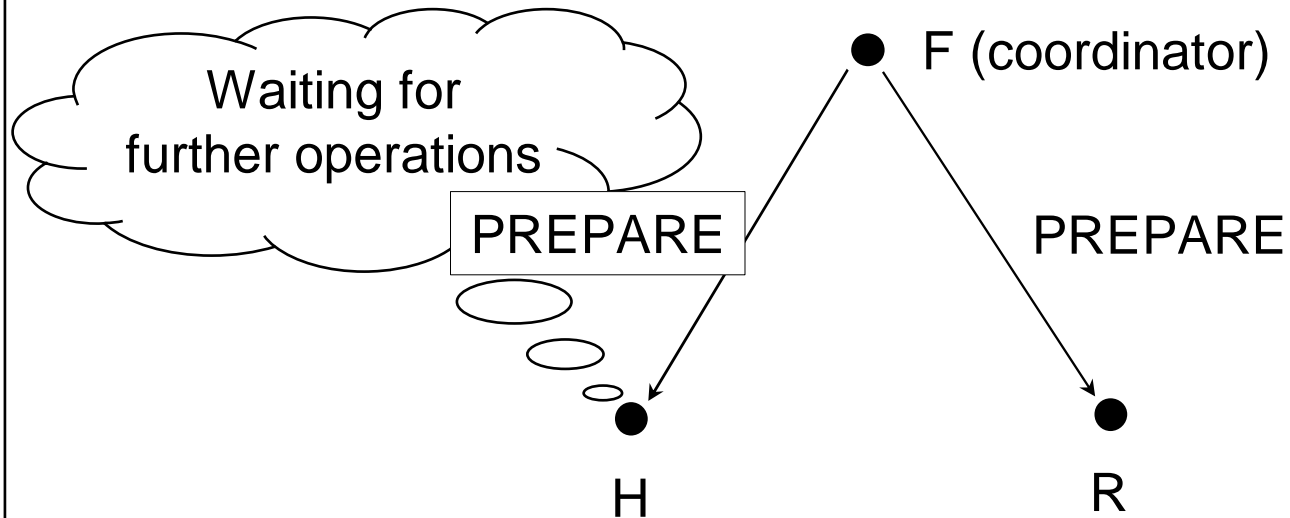
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – no Failures (1)

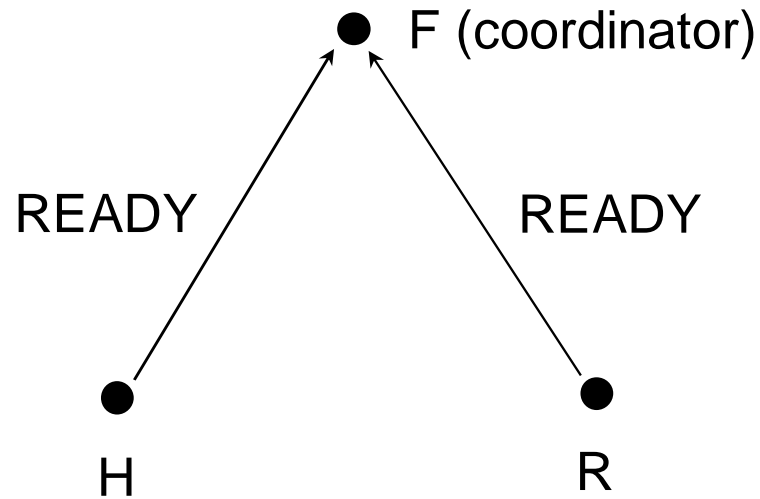
- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	
Database			Log	

2PC Execution – no Failures (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

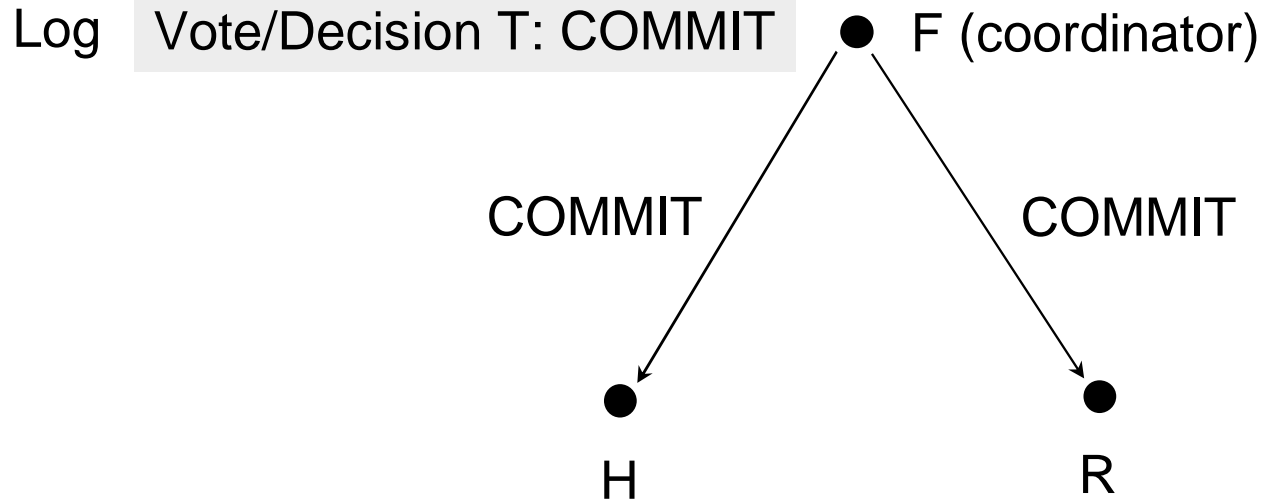
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – no Failures (3)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

Database

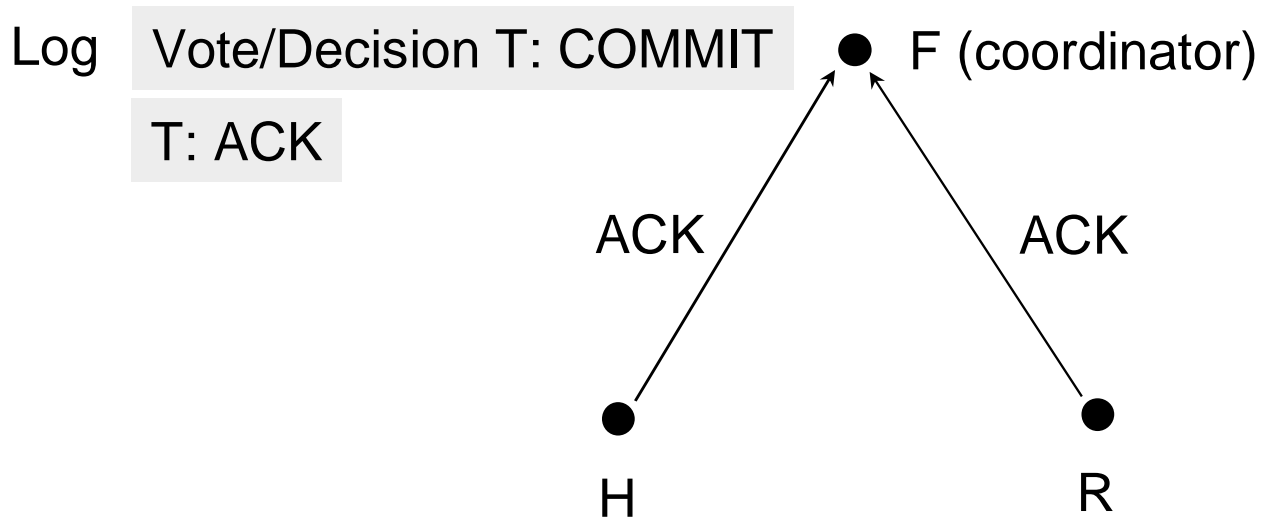
<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – no Failures (4)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	KB <input type="checkbox"/>
Phil H.	KB <input type="checkbox"/>
...	...

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil H.	<input type="checkbox"/>
...	

Vote T: COMMIT

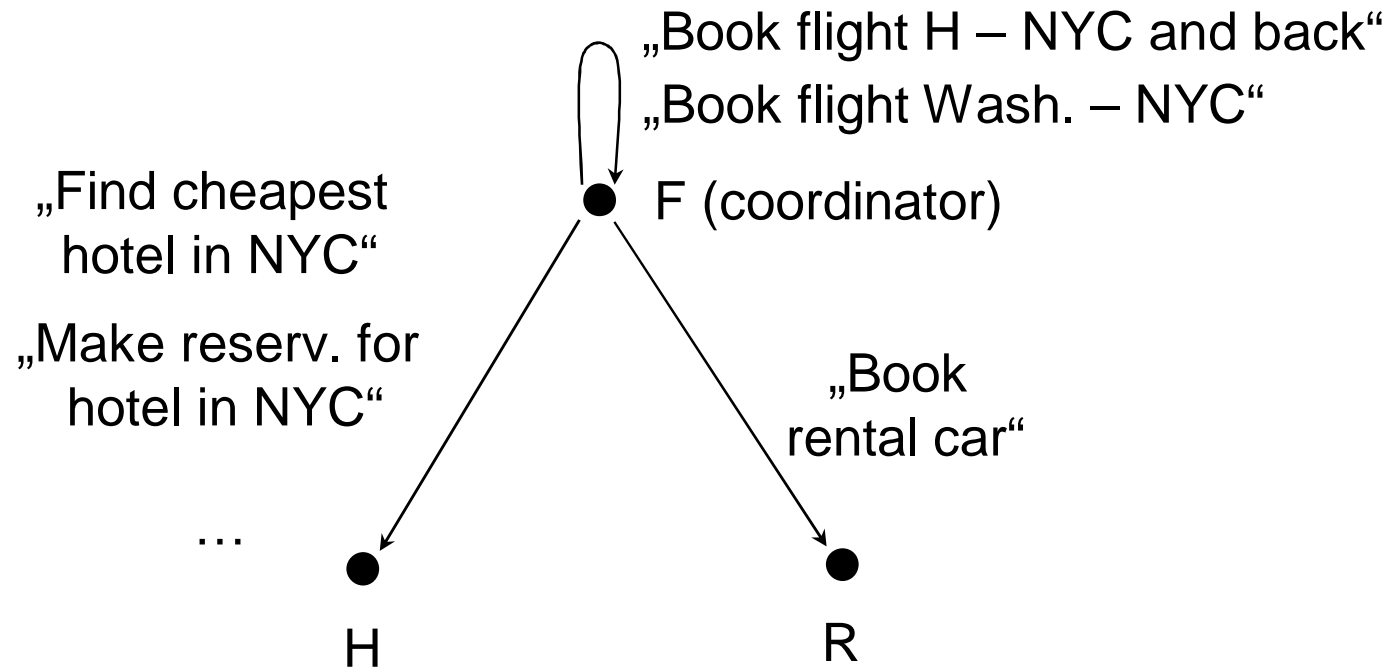
Decision T: COMMIT

2PC Execution – Agent Failures Overview

- Before READY,
- after READY, before decision,
- after decision.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

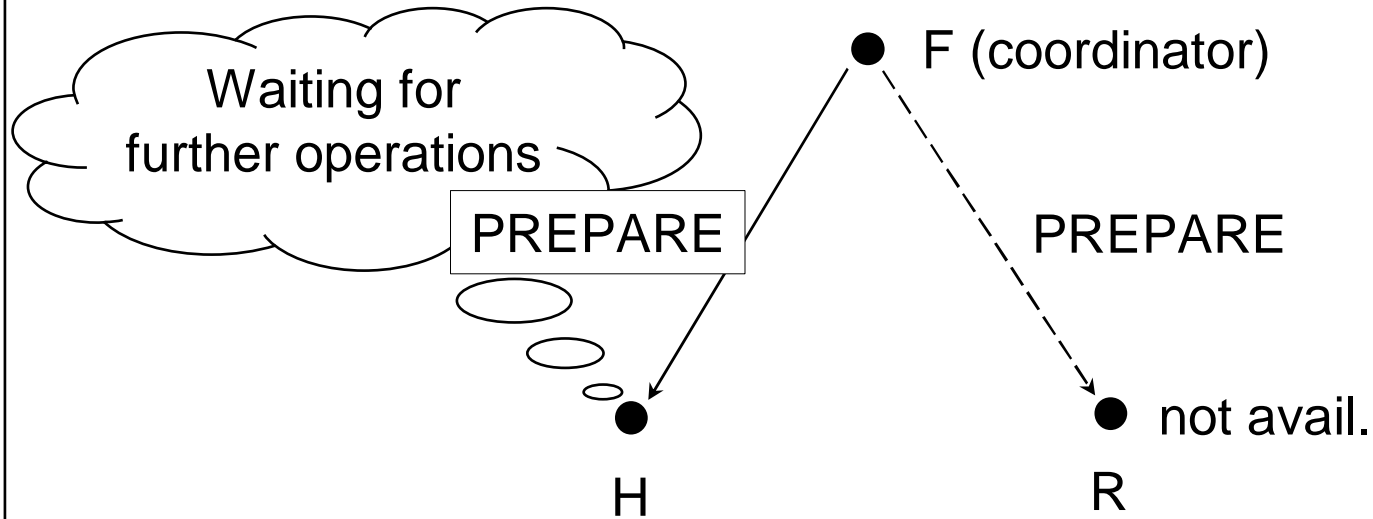
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure before READY (1)

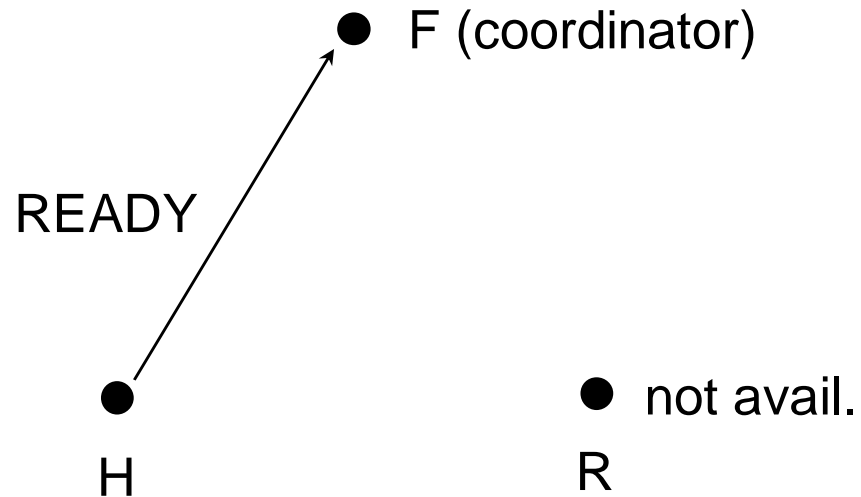
- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	
Database			Log	

2PC Execution – Failure before READY (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

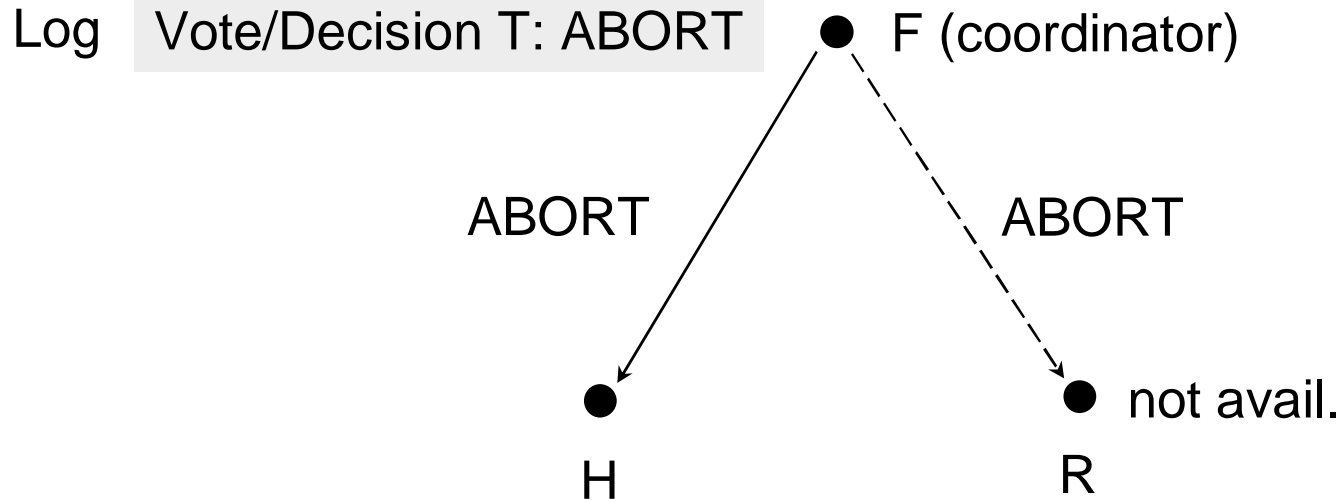
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure before READY (3)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

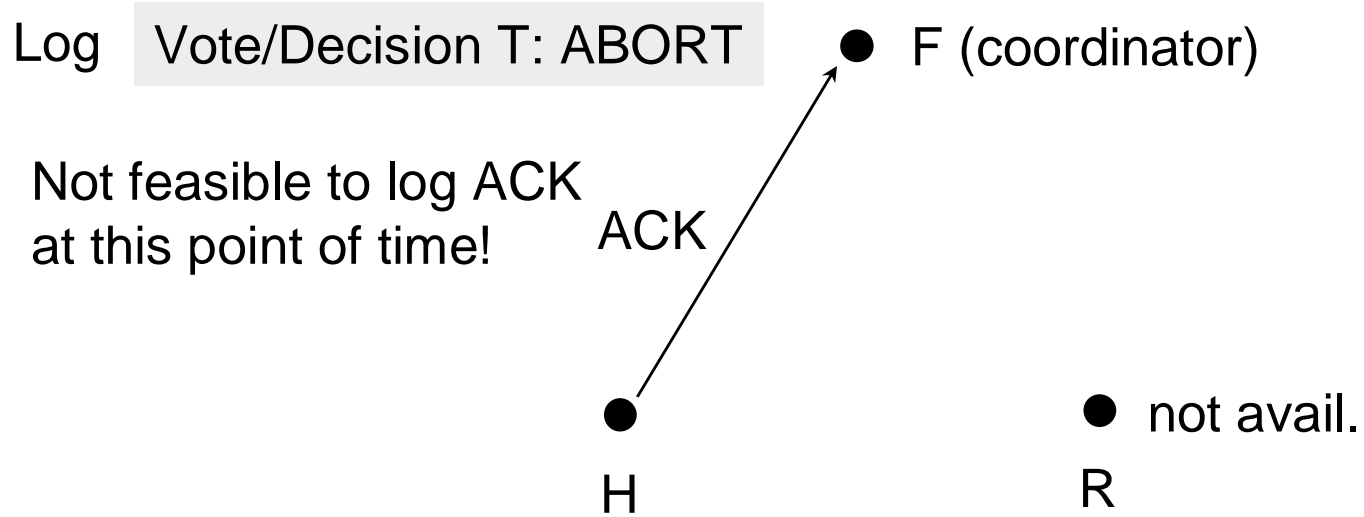
Database

Vote T: COMMIT

Log

2PC Execution – Failure before READY (4)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	NULL <input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>
...	...

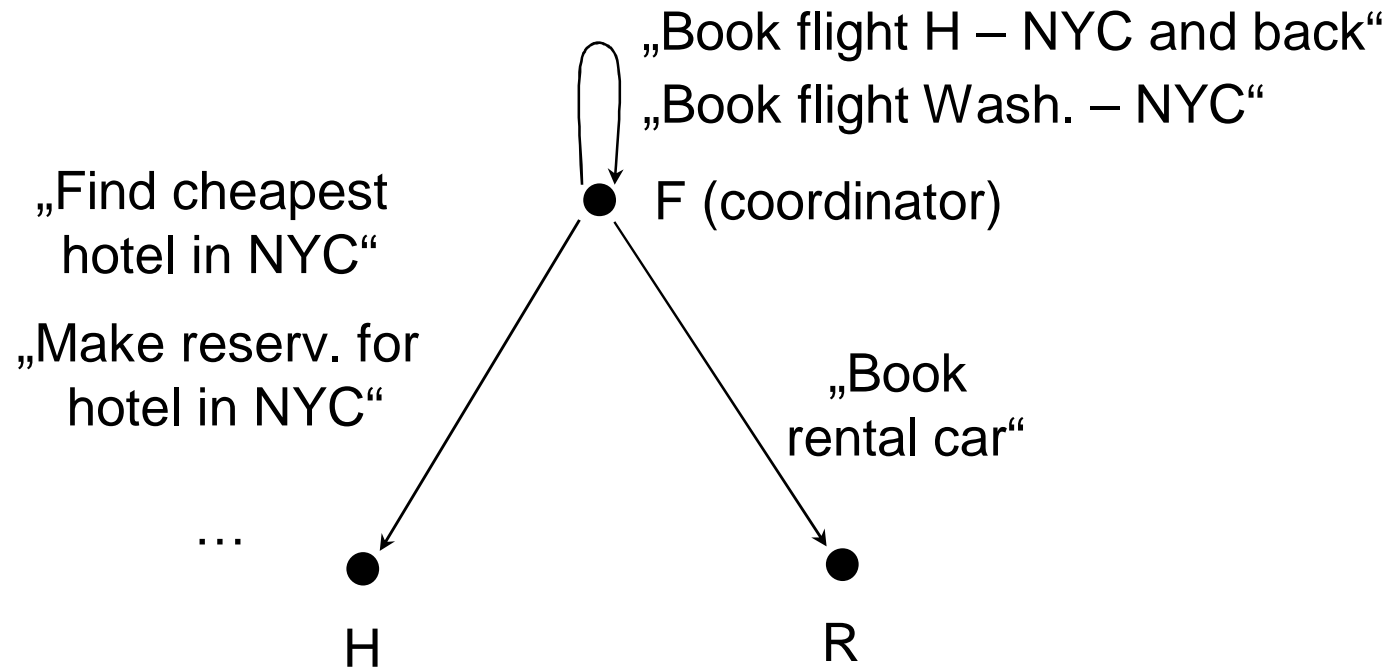
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil. H.	<input type="checkbox"/>
...	

Vote T: COMMIT

Decision T: ABORT

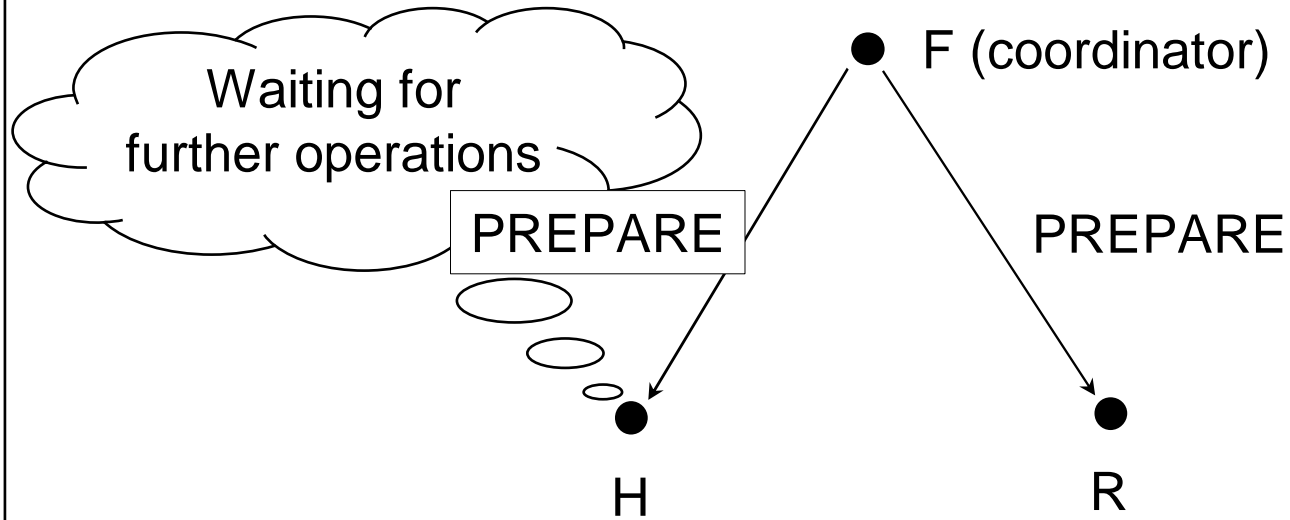
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure before Decision (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

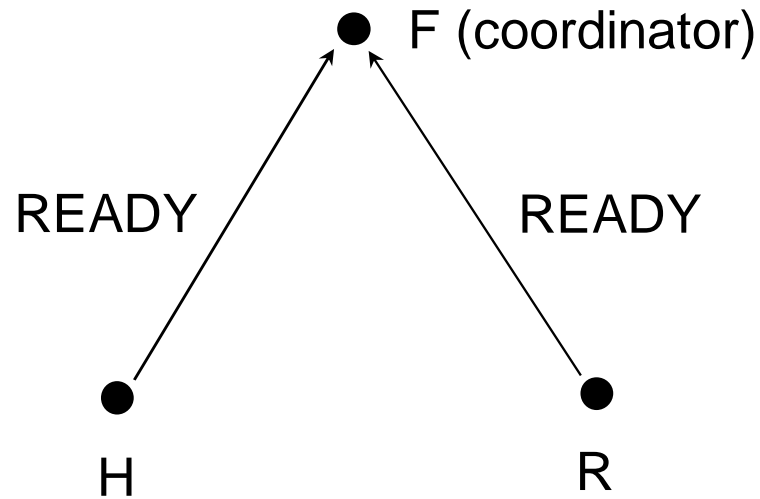


<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database Log

2PC Execution – Failure before Decision (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

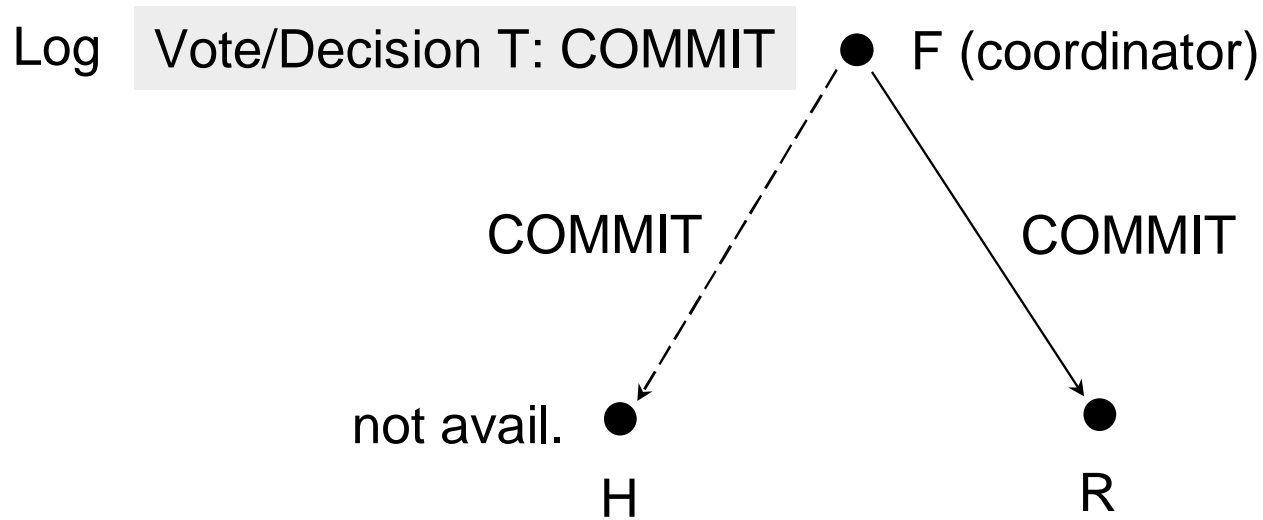
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure before Decision (3)



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

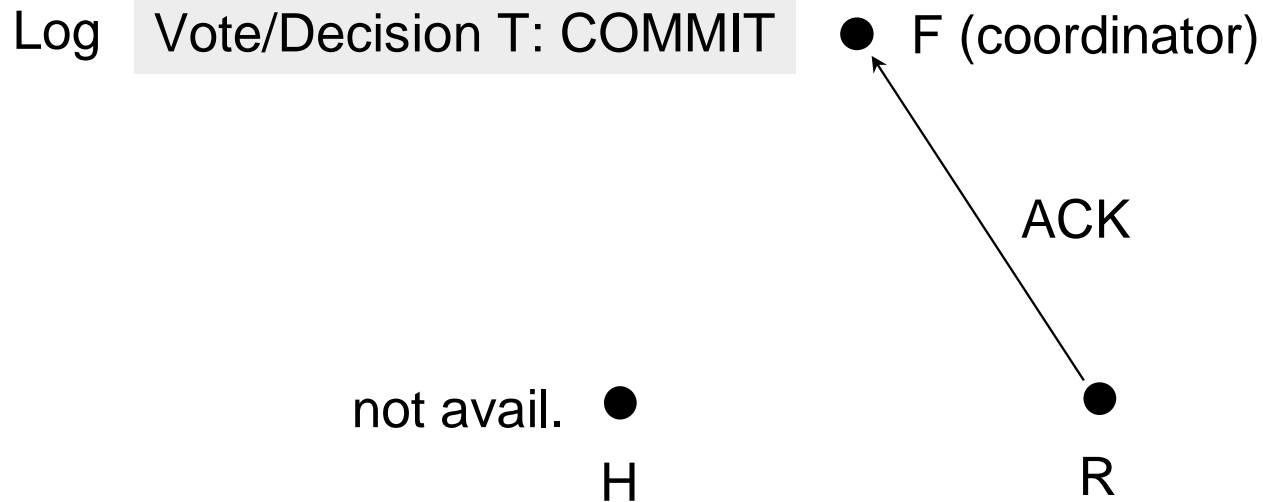
Database

Vote T: COMMIT

Log

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure before Decision (4)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

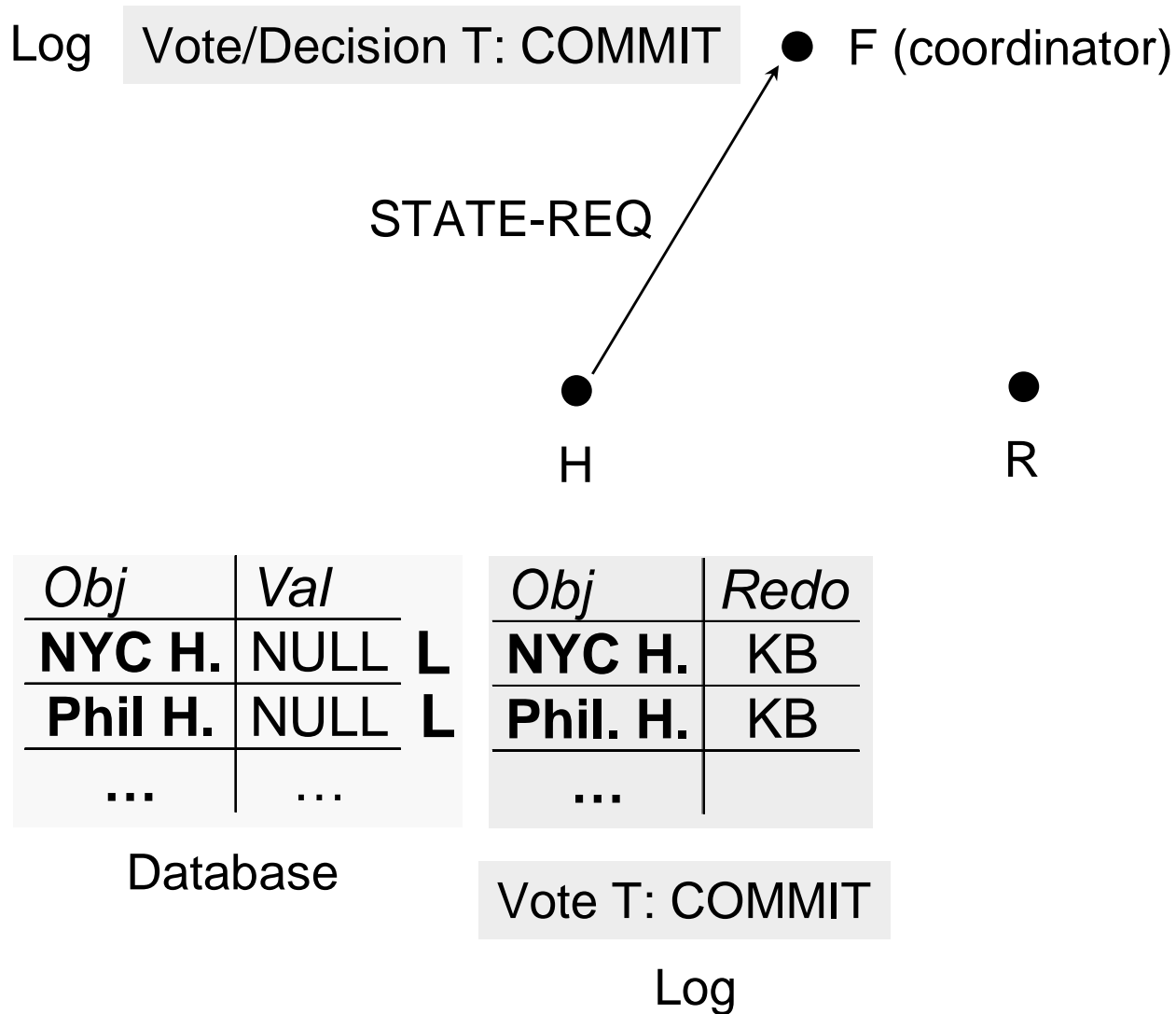
Database

Vote T: COMMIT

Log

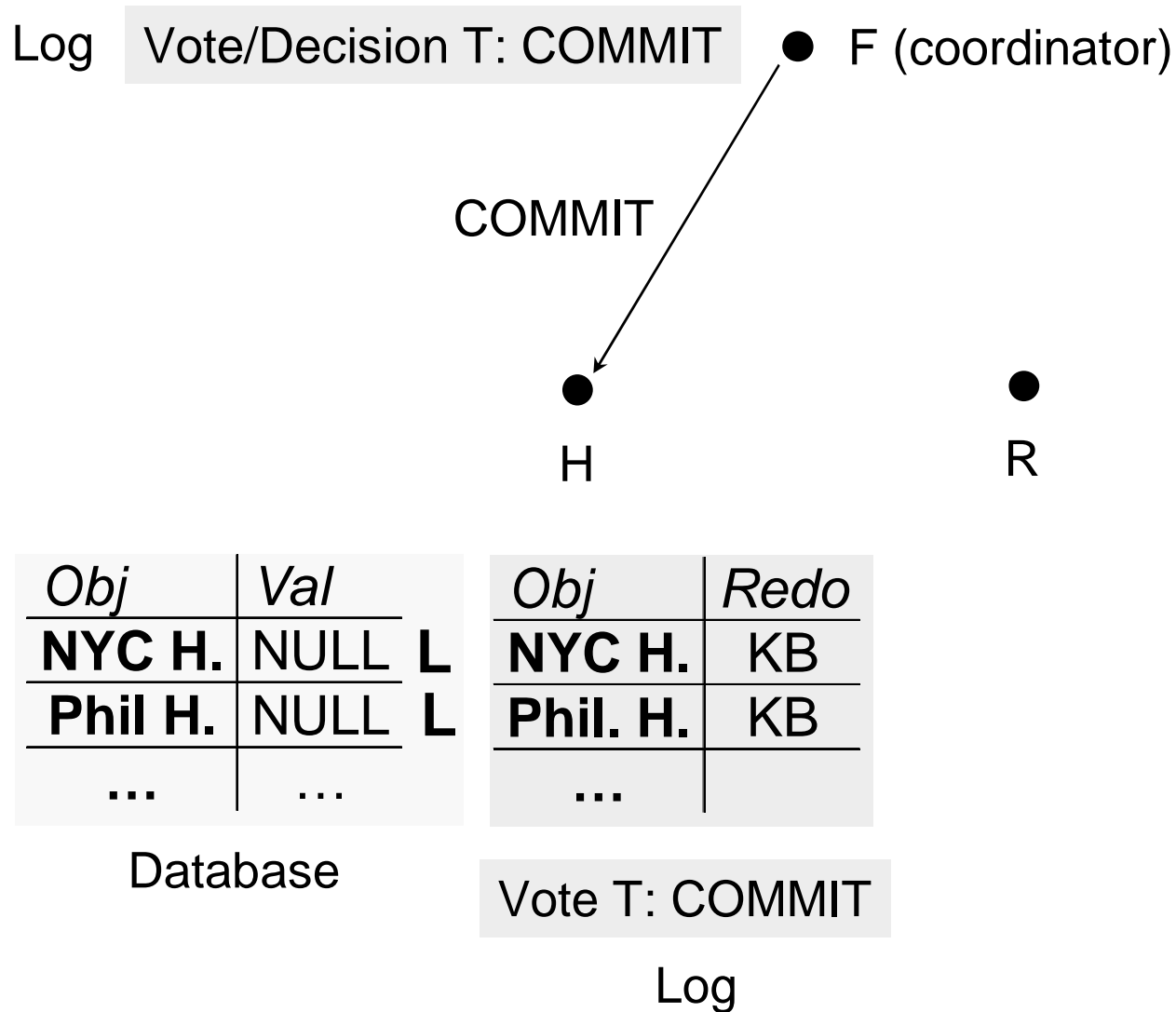
2PC Execution – Failure before Decision (5)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



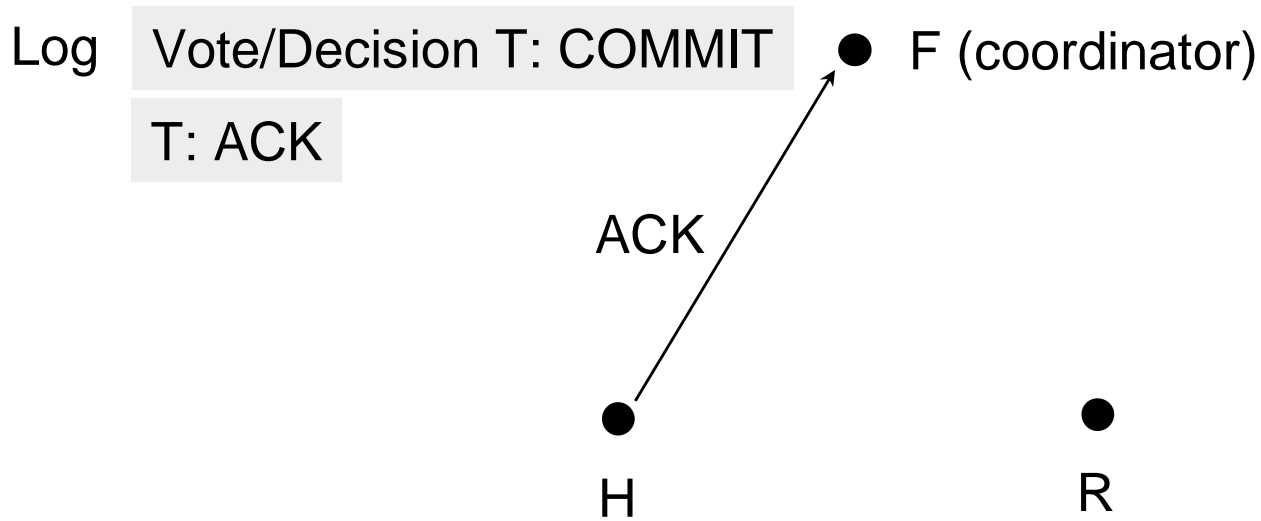
2PC Execution – Failure before Decision (6)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



2PC Execution – Failure before Decision (7)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	KB <input type="checkbox"/>
Phil H.	KB <input type="checkbox"/>
...	...

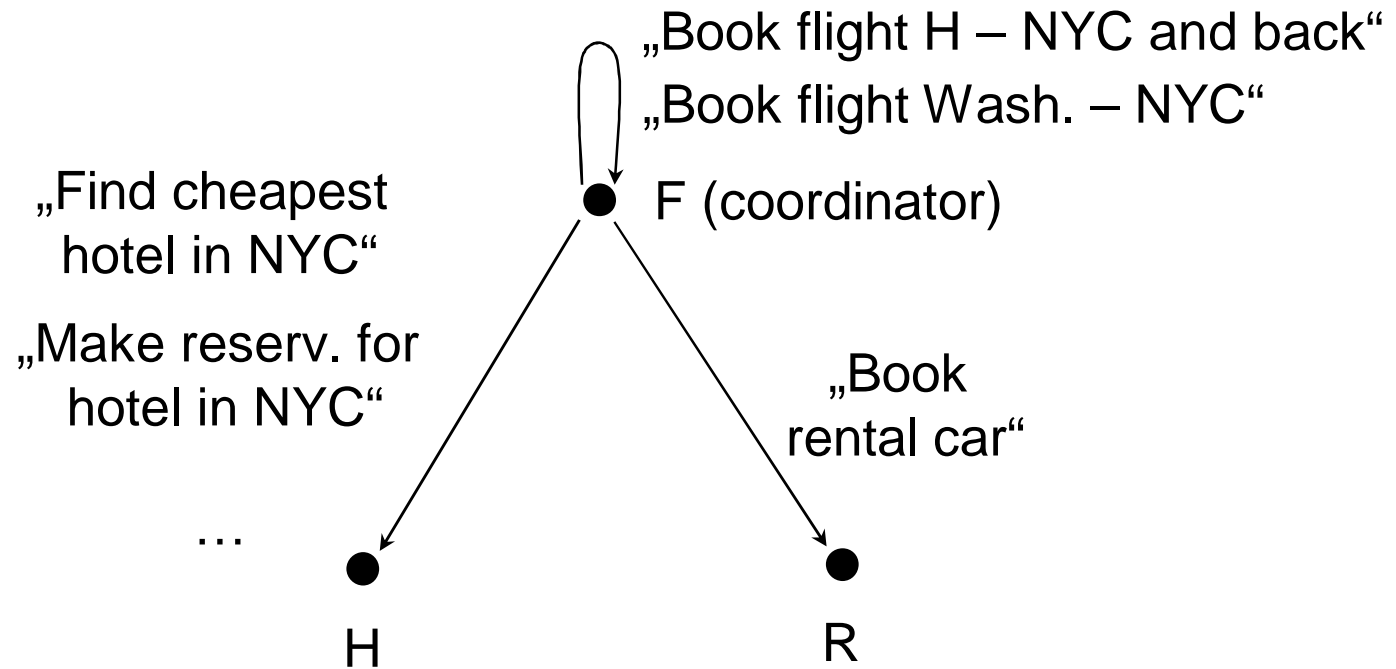
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil H.	<input type="checkbox"/>
...	

Vote T: COMMIT

Decision T: COMMIT

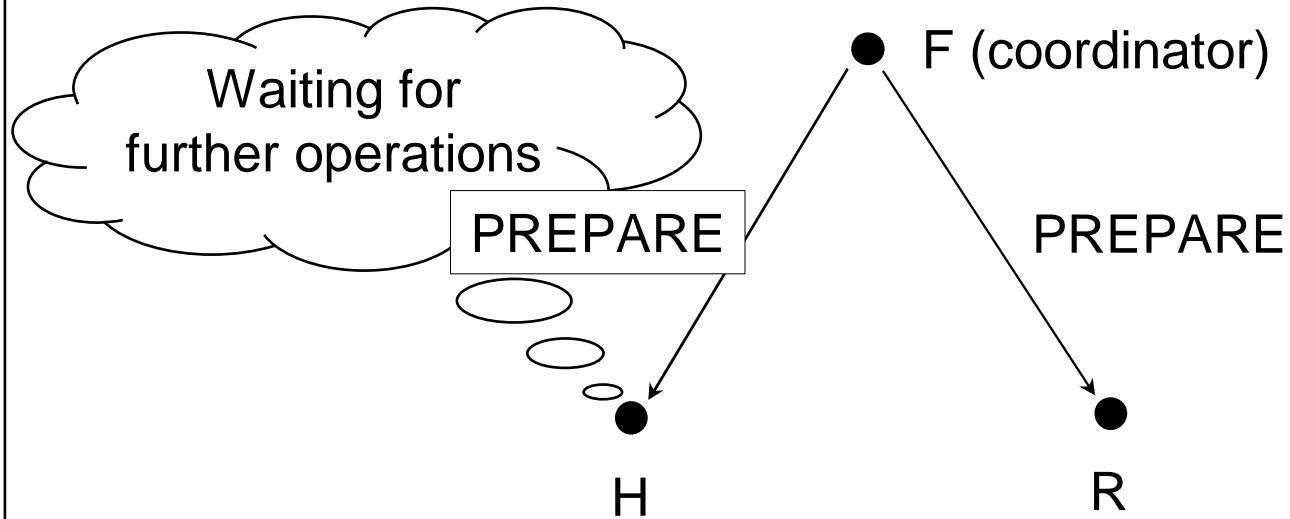
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure after Decision (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

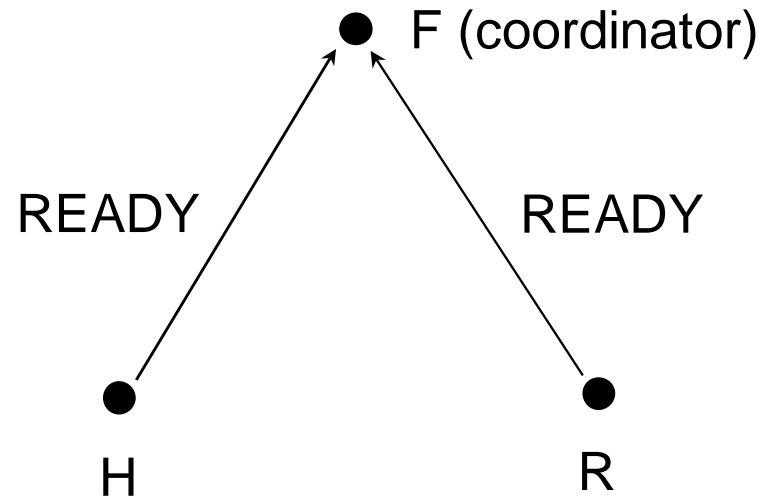


<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database Log

2PC Execution – Failure after Decision (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

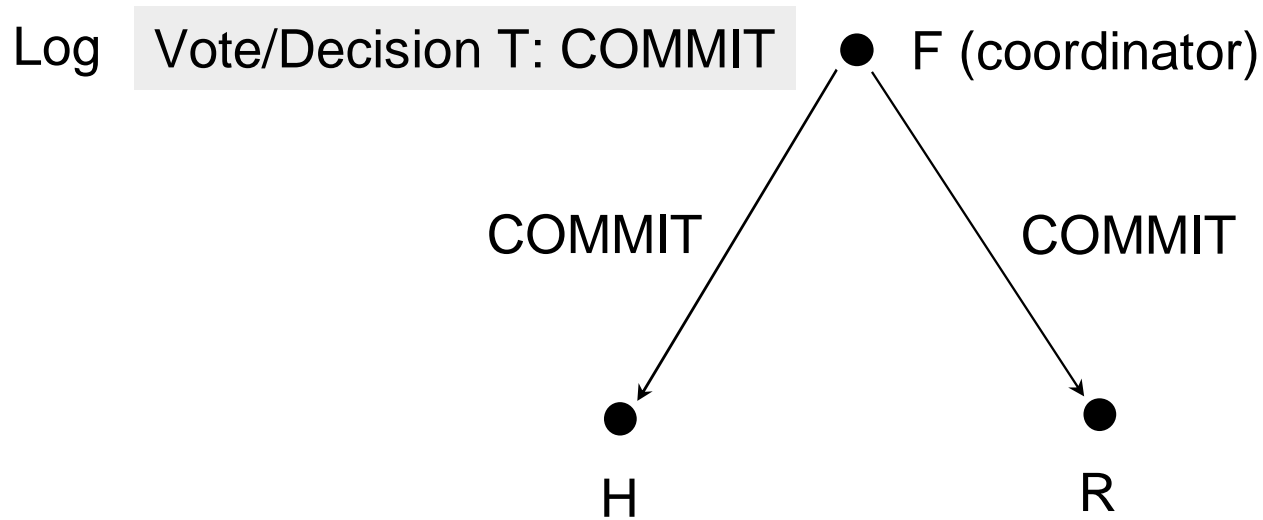
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure after Decision (3)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

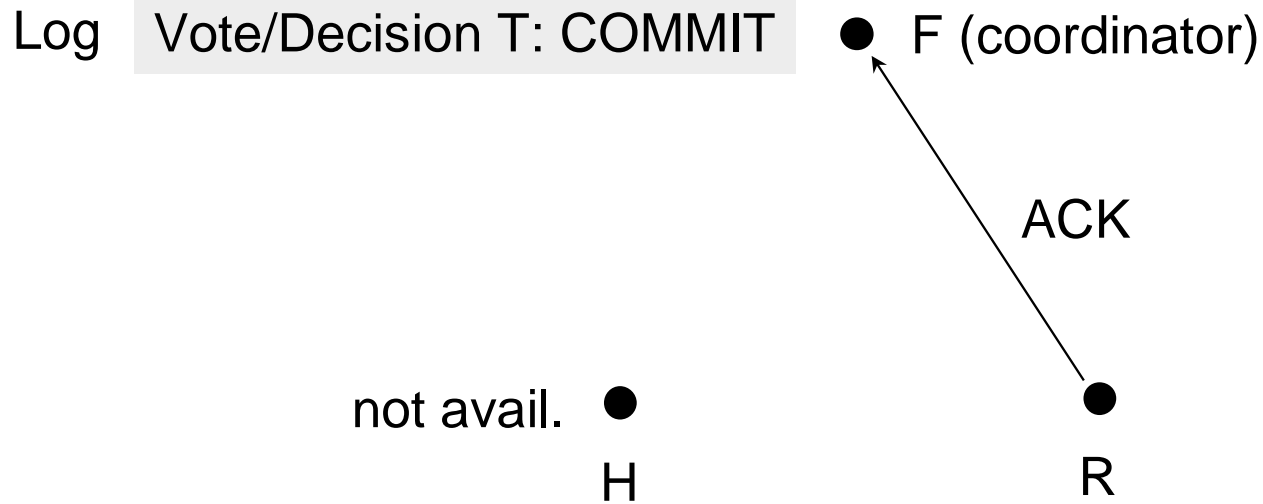
<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database

Vote T: COMMIT

Log

2PC Execution – Failure after Decision (4)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	KB	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

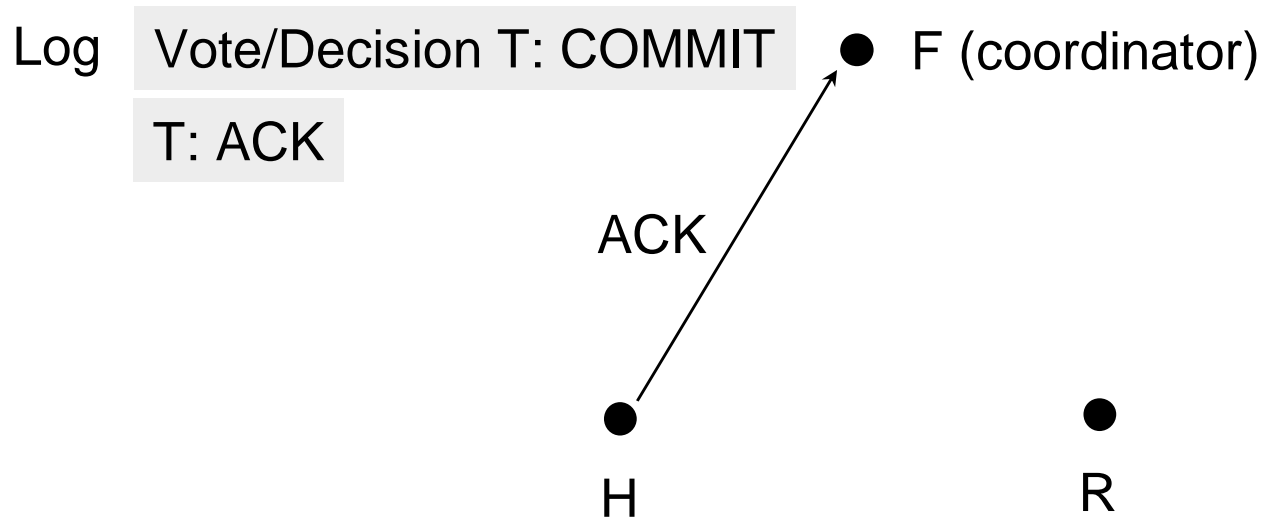
Database

Vote T: COMMIT

Decision T: COMMIT

2PC Execution – Failure after Decision (5)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	KB <input type="checkbox"/>
Phil H.	KB <input type="checkbox"/>
...	...

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil H.	<input type="checkbox"/>
...	

Vote T: COMMIT

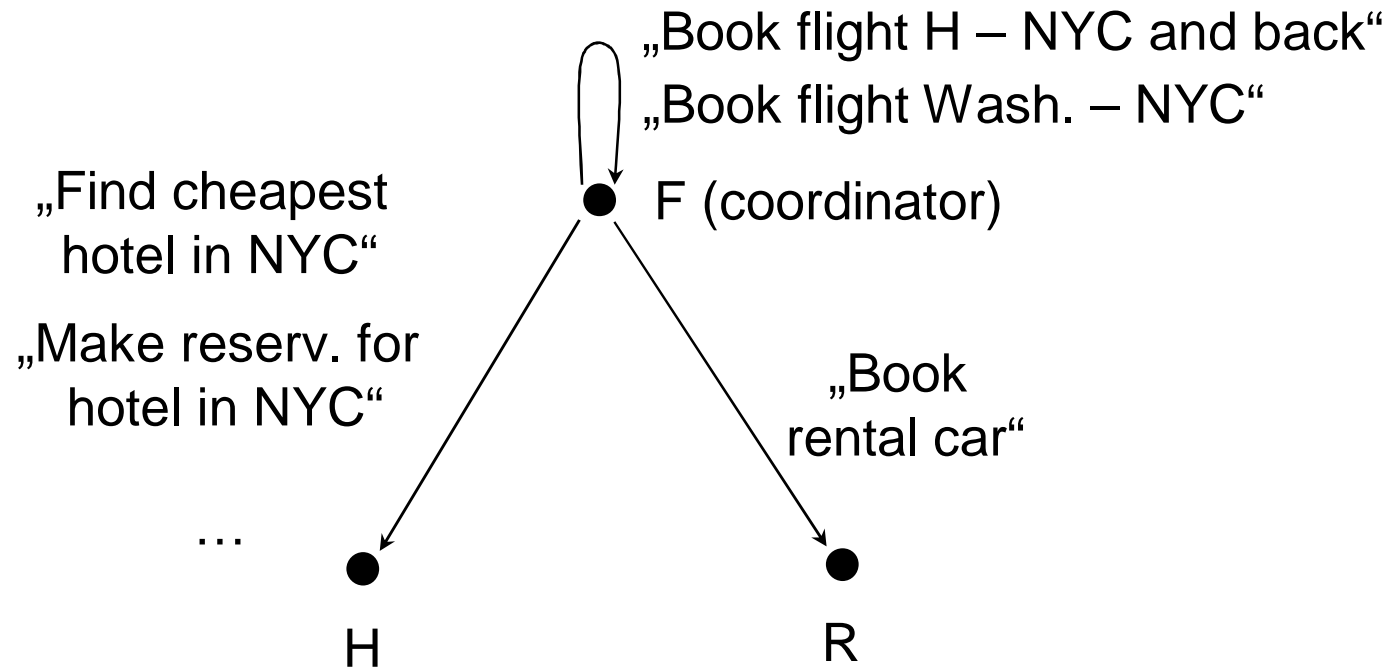
Decision T: COMMIT

2PC Execution – Coordinator Failures Overview

- Before PREPARE,
- after PREPARE, before COMMIT,
- after COMMIT,
before any message has been sent out,
- after COMMIT,
after messages have been sent out.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

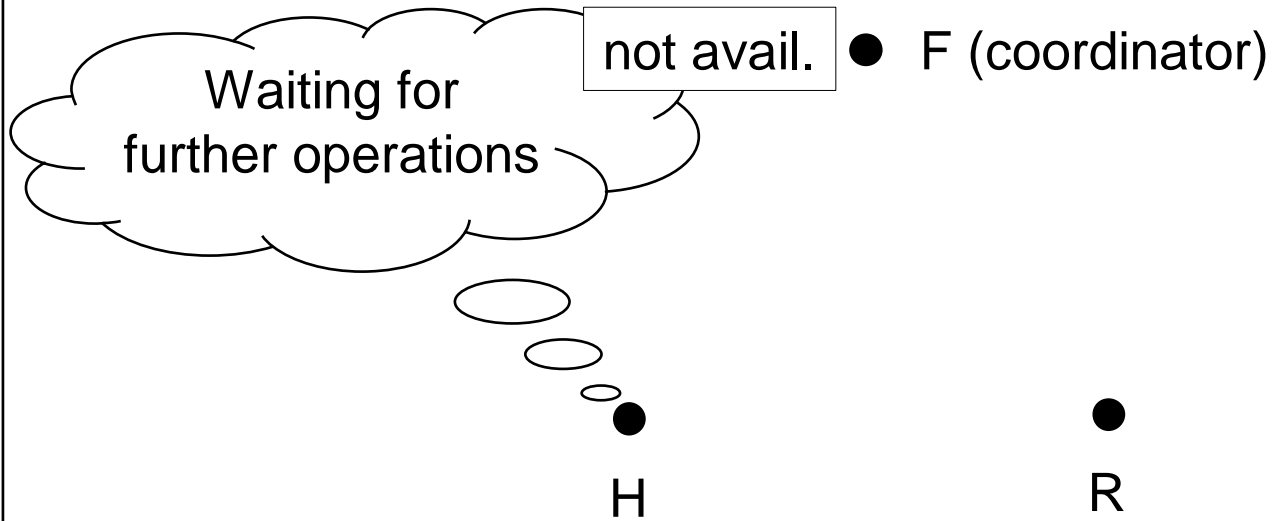
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure before PREPARE (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	
Database			Log	

2PC Execution – Failure before PREPARE (2)

not avail. ● F (coordinator)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

?

●

H

●

R

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Unilateral Abort

Database

Decision T: ABORT

Log

2PC Execution – Failure before PREPARE (3)

not avail. ● F (coordinator)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

●
H

●
R

<i>Obj</i>	<i>Val</i>
NYC H.	NULL <input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>
...	...

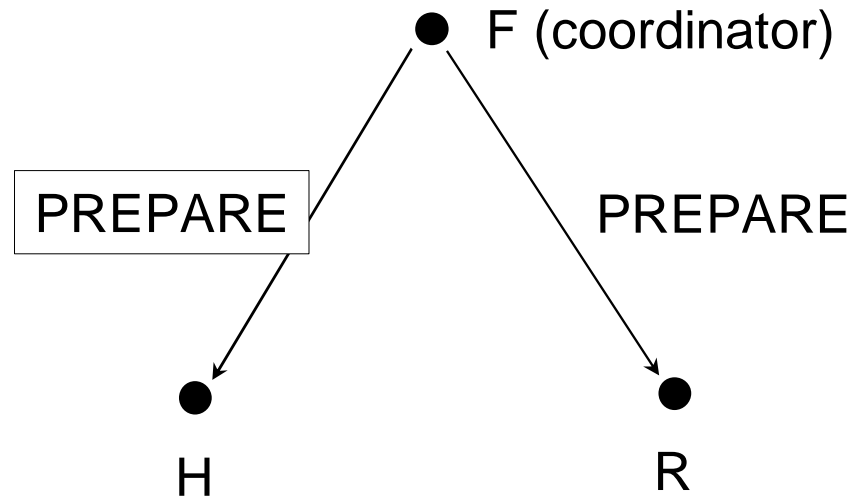
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil. H.	<input type="checkbox"/>
...	

Decision T: ABORT

2PC Execution – Failure before PREPARE (4)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	NULL <input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>
...	...

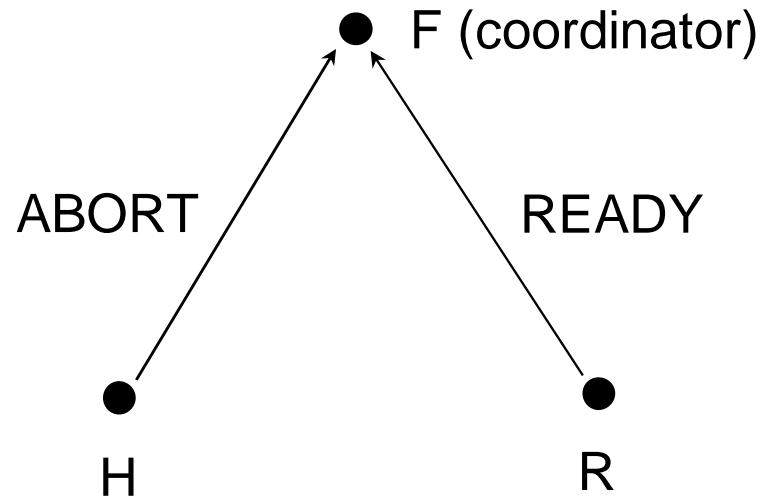
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil. H.	<input type="checkbox"/>
...	

Decision T: ABORT

2PC Execution – Failure before PREPARE (5)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



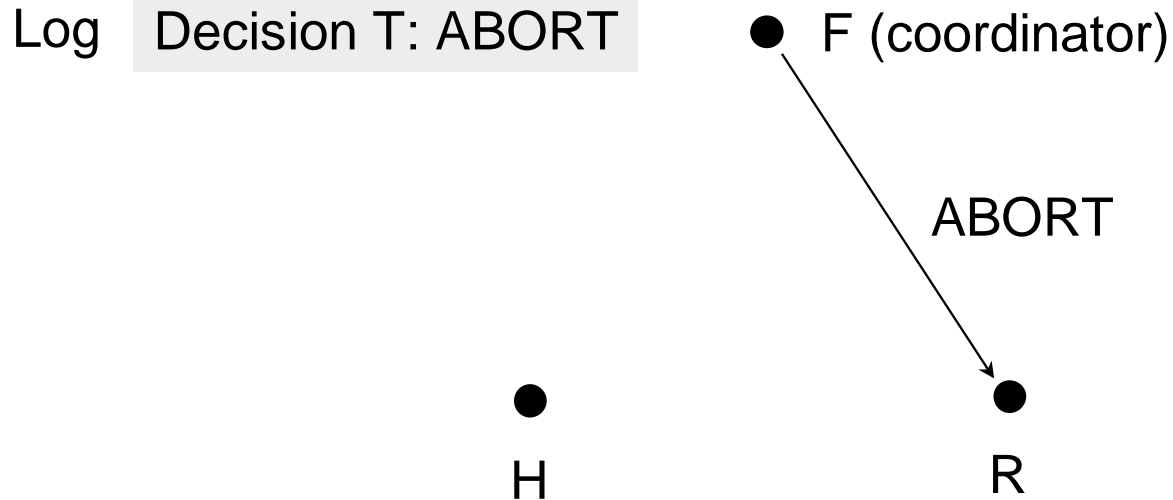
<i>Obj</i>	<i>Val</i>
NYC H.	NULL <input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>
...	...

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil. H.	<input type="checkbox"/>
...	

Decision T: ABORT

2PC Execution – Failure before PREPARE (6)



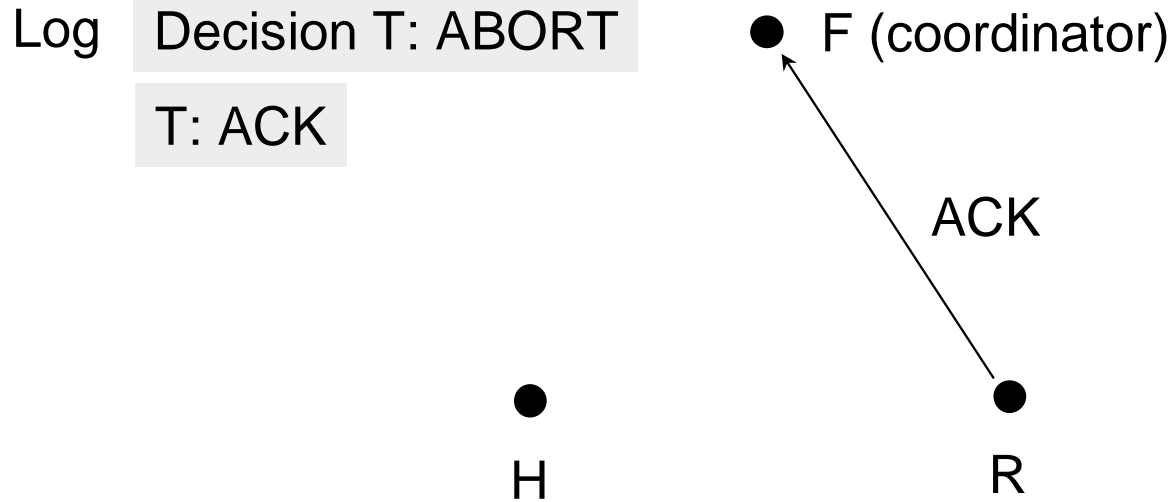
- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>	<i>Obj</i>	<i>Redo</i>
NYC H.	NULL <input type="checkbox"/>	NYC H.	<input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>	Phil. H.	<input type="checkbox"/>
...	

Database

Decision T: ABORT

2PC Execution – Failure before PREPARE (7)

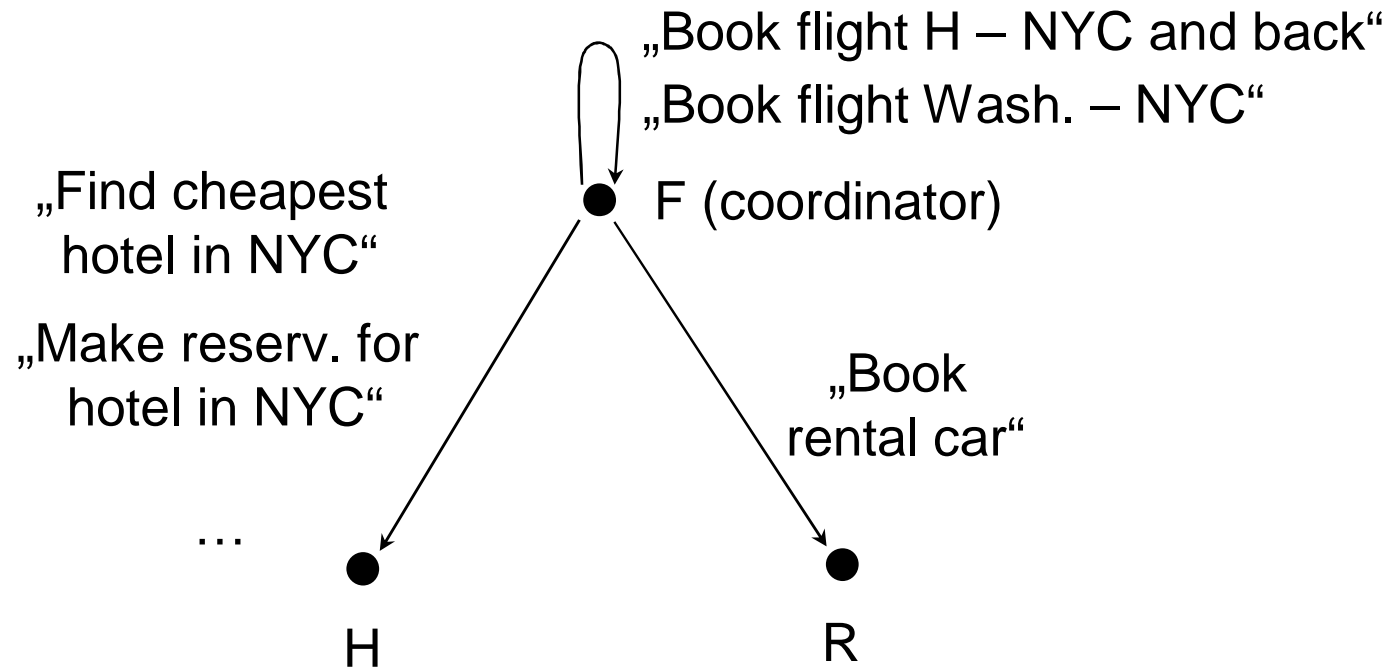


- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>	<i>Obj</i>	<i>Redo</i>
NYC H.	NULL <input type="checkbox"/>	NYC H.	<input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>	Phil. H.	<input type="checkbox"/>
...	

Database Decision T: ABORT

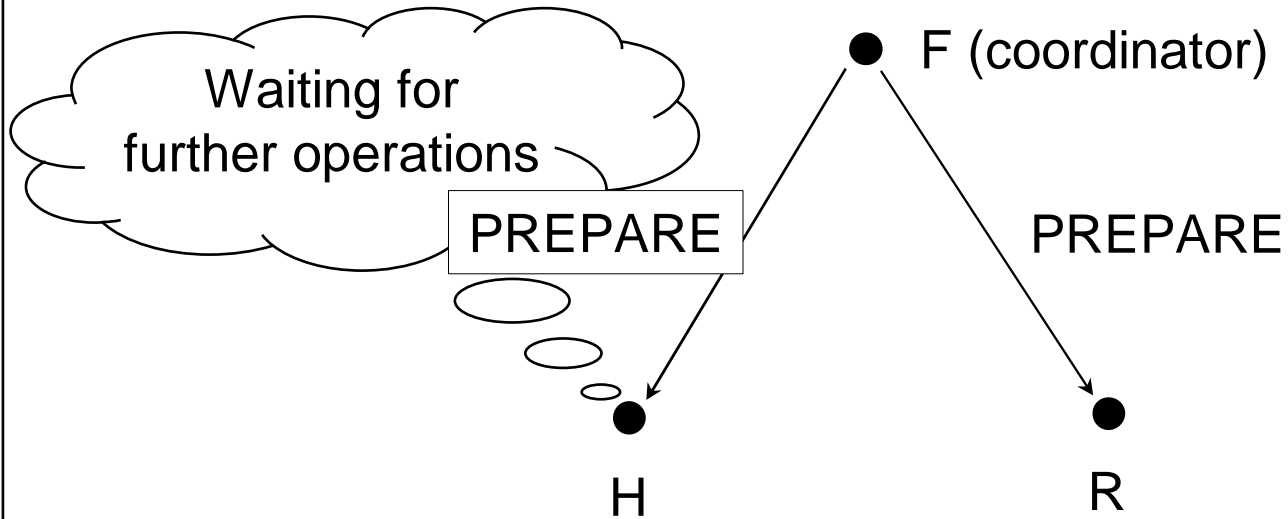
Execution of Subtransactions



Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

2PC Execution – Failure before COMMIT (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

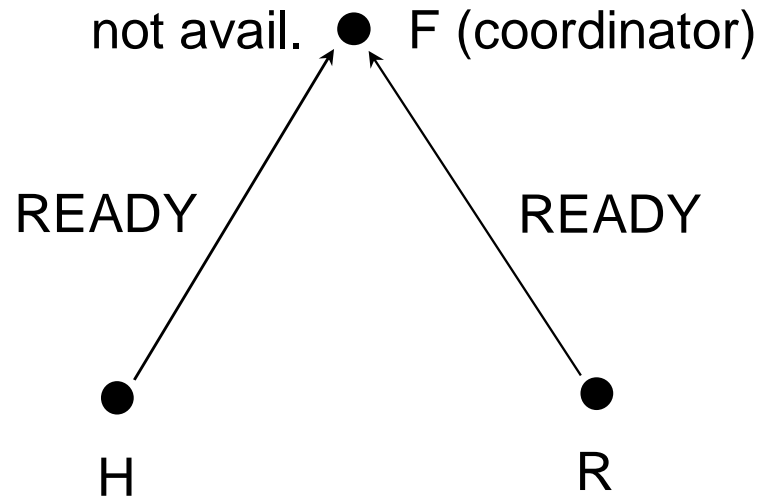


<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database Log

2PC Execution – Failure before COMMIT (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure before COMMIT (3)

not avail. ● F (coordinator)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

Blocking



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

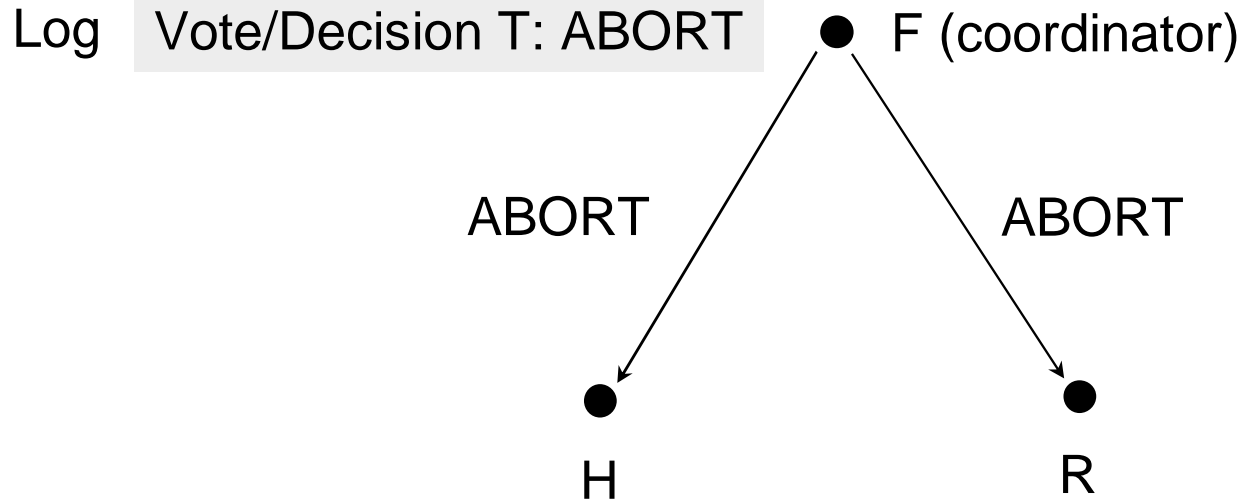
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure before COMMIT (4)



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database

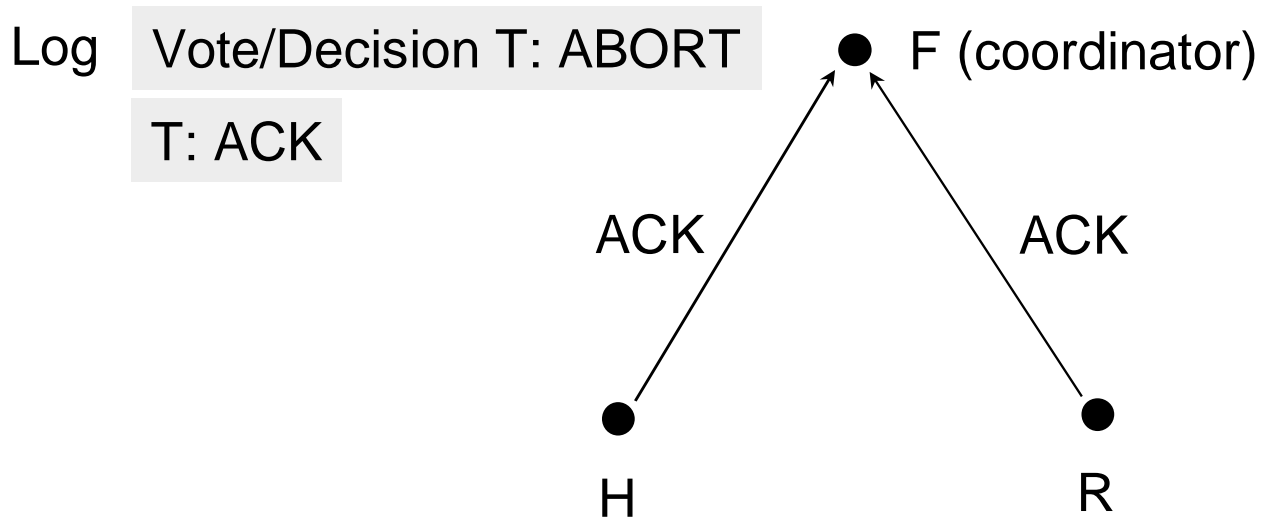
Vote T: COMMIT

Log

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure before COMMIT (5)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	NULL <input type="checkbox"/>
Phil H.	NULL <input type="checkbox"/>
...	...

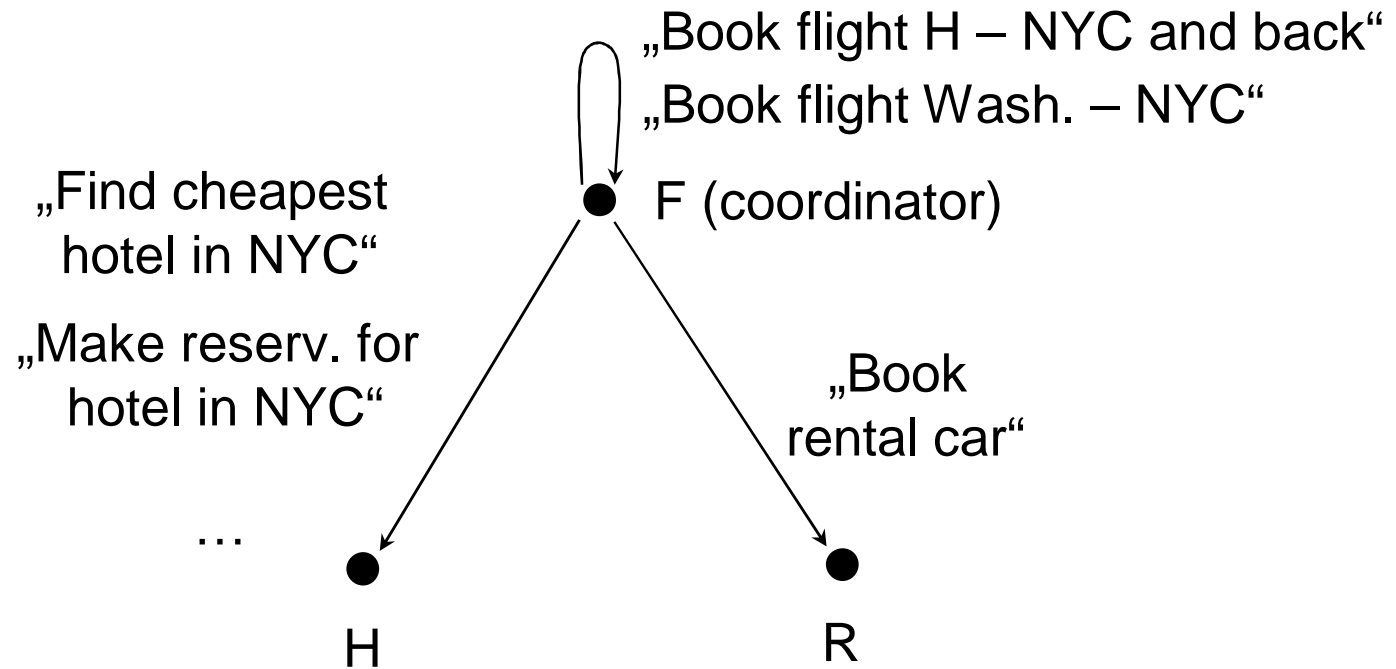
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil. H.	<input type="checkbox"/>
...	...

Vote T: COMMIT

Decision T: ABORT

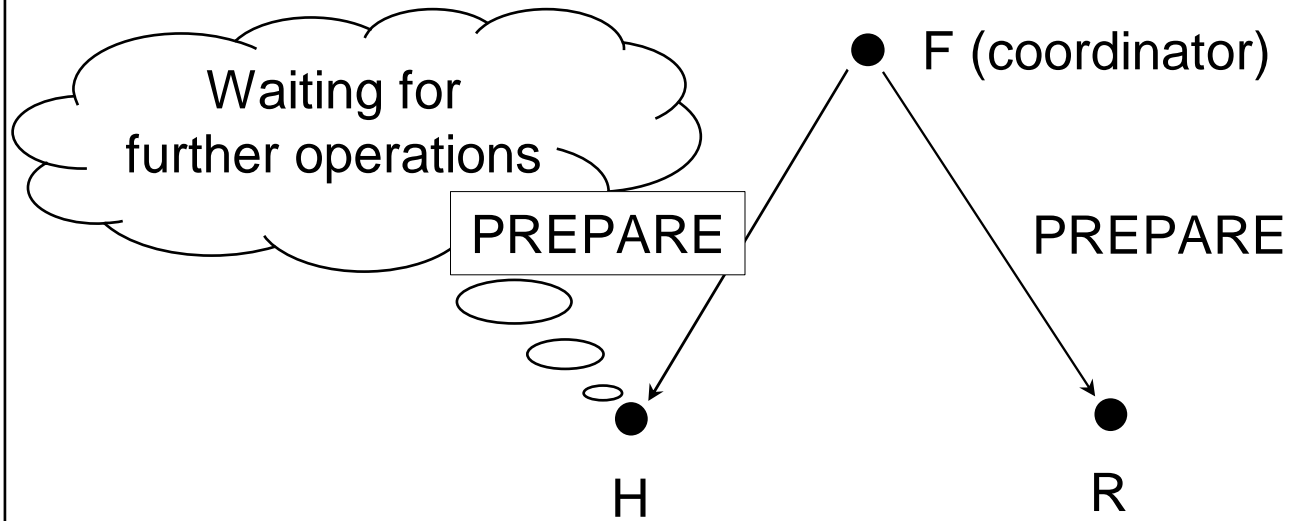
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure after COMMIT, before Messages Sent (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

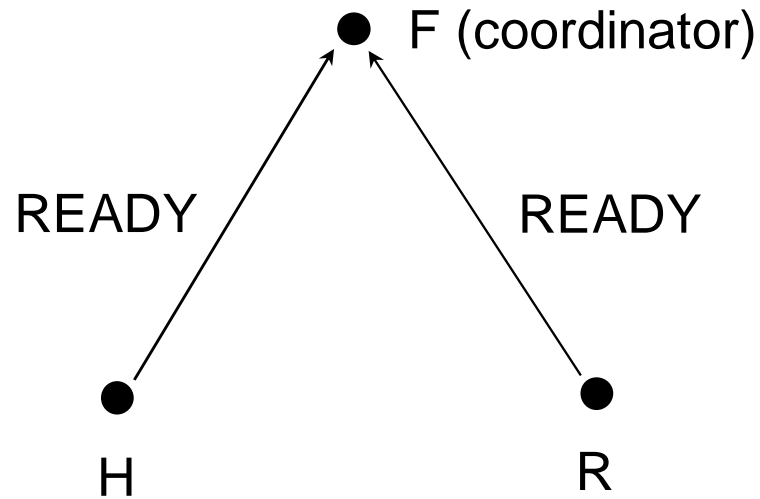


<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database Log

2PC Execution – Failure after COMMIT, before Messages Sent (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, before Messages Sent (3)

Log **Vote/Decision T: COMMIT** ● F (coordinator)
not avail.

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

●
H

●
R

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, before Messages Sent (4)

Log **Vote/Decision T: COMMIT** ● F (coordinator)
not avail.

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

Blocking



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

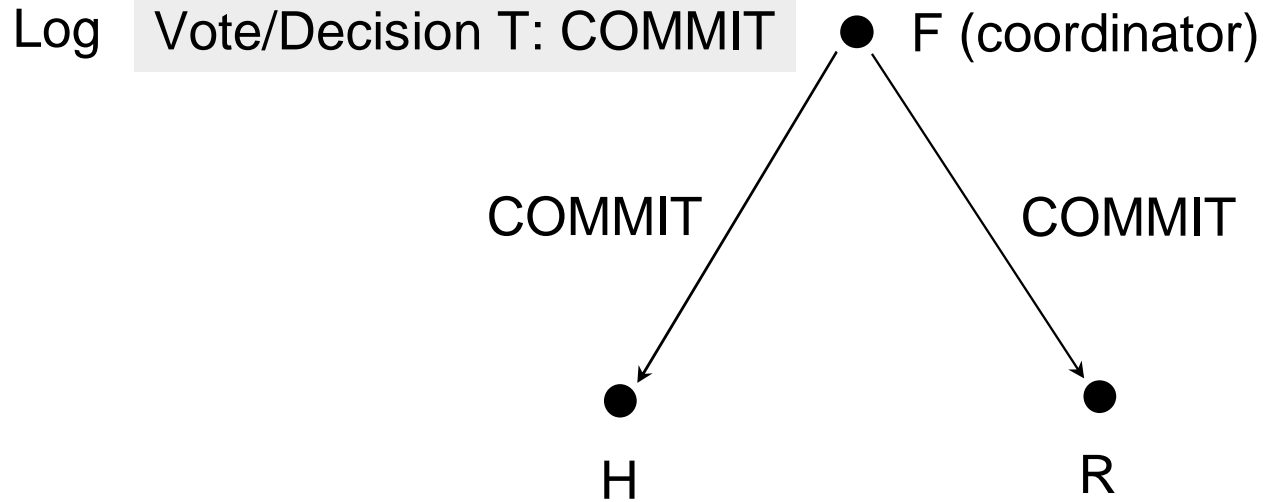
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, before Messages Sent (5)



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

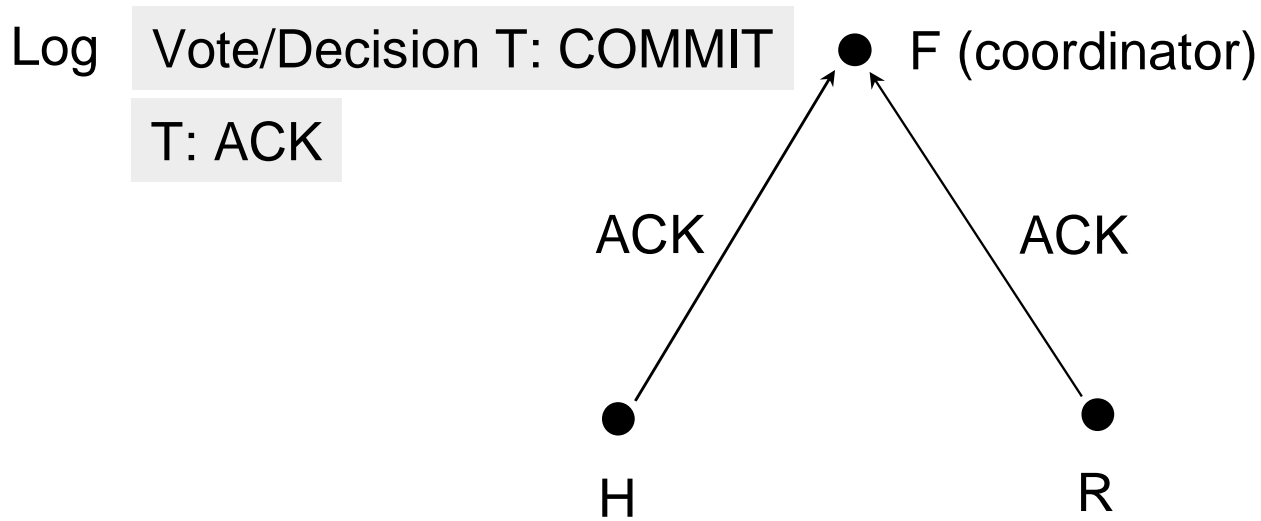
Database

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, before Messages Sent (6)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	KB <input type="checkbox"/>
Phil H.	KB <input type="checkbox"/>
...	...

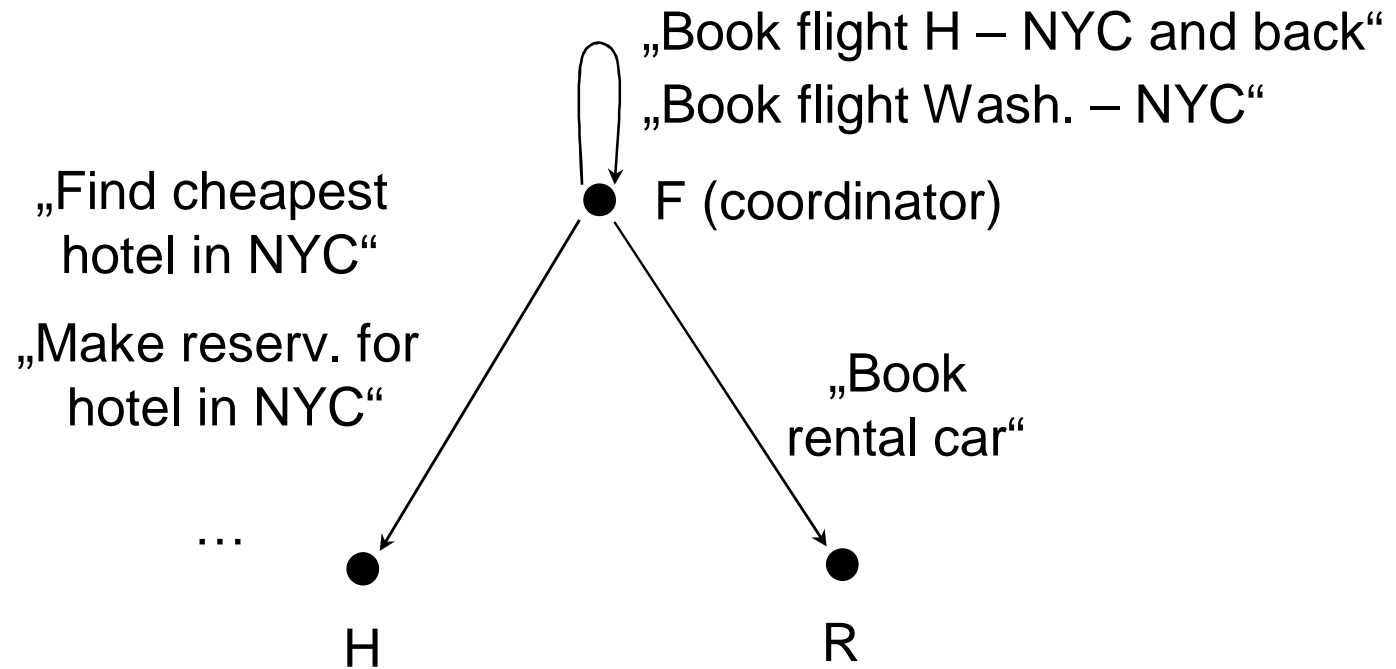
Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil H.	<input type="checkbox"/>
...	

Vote T: COMMIT

Decision T: COMMIT

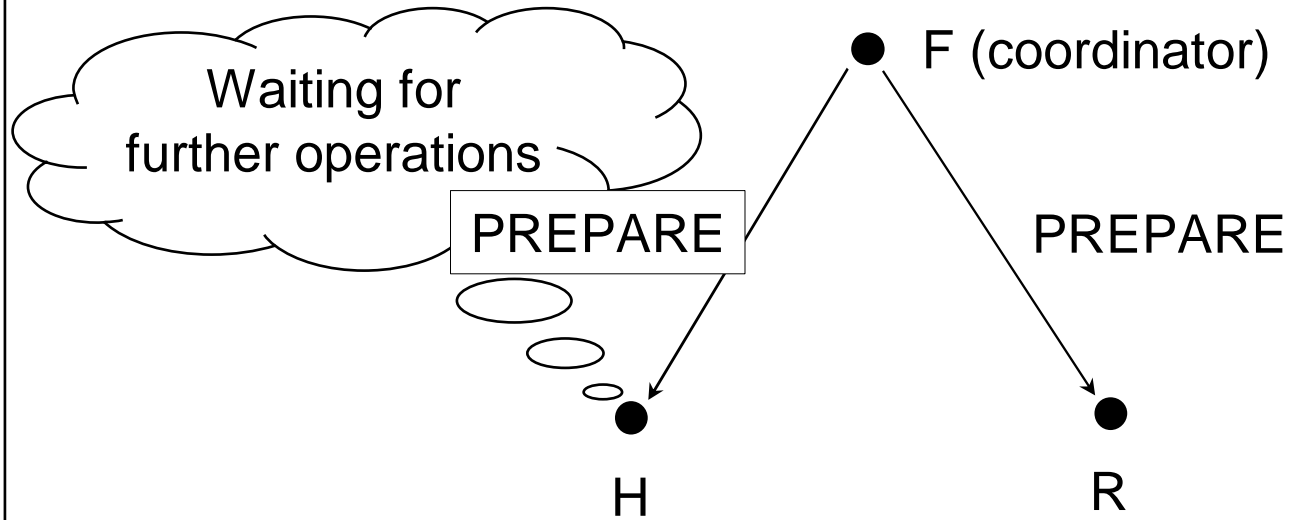
Execution of Subtransactions



- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

2PC Execution – Failure after COMMIT, after Messages Sent (1)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion

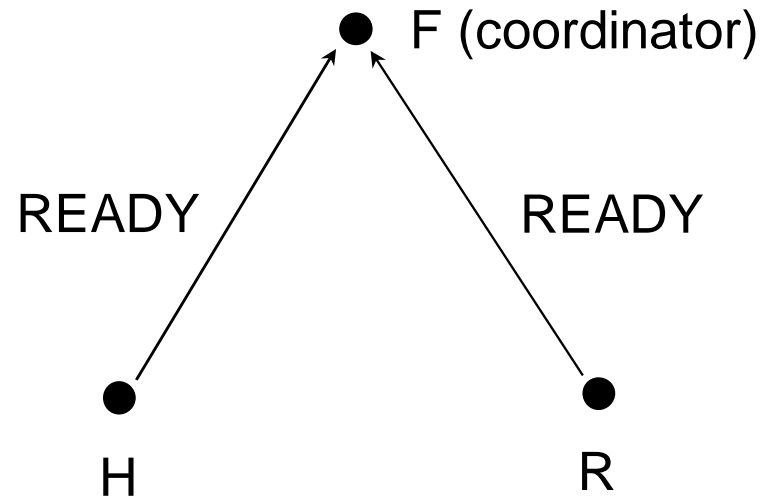


<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database Log

2PC Execution – Failure after COMMIT, after Messages Sent (2)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>	
NYC H.	NULL	L
Phil H.	NULL	L
...	...	

Database

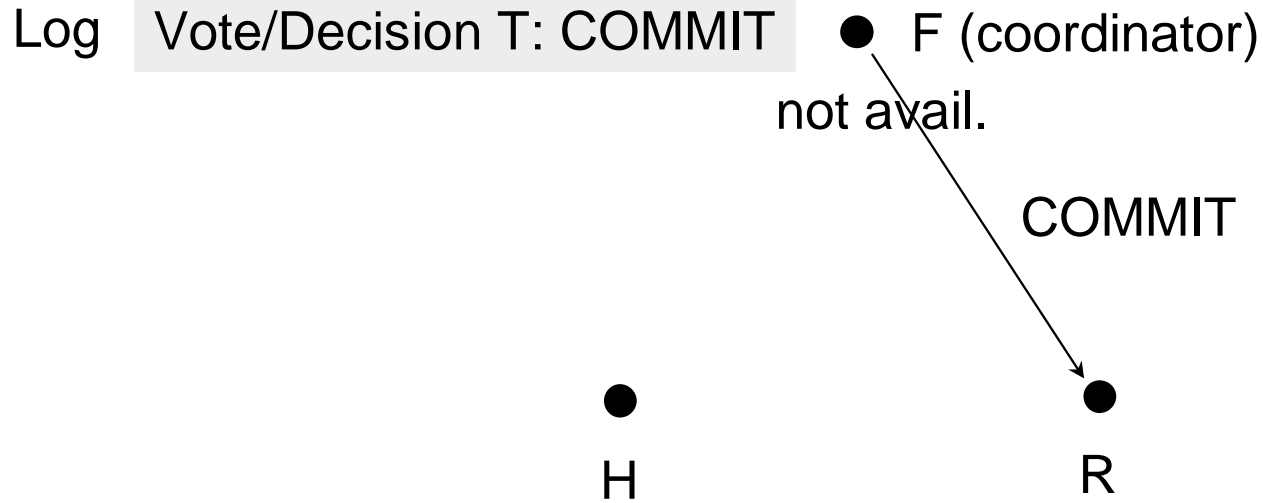
<i>Obj</i>	<i>Redo</i>
NYC H.	KB
Phil. H.	KB
...	

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, after Messages Sent (3)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

Database

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, after Messages Sent (4)

Log **Vote/Decision T: COMMIT** ● F (coordinator)
not avail.

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>		<i>Obj</i>	<i>Redo</i>
NYC H.	NULL	L	NYC H.	KB
Phil H.	NULL	L	Phil. H.	KB
...	

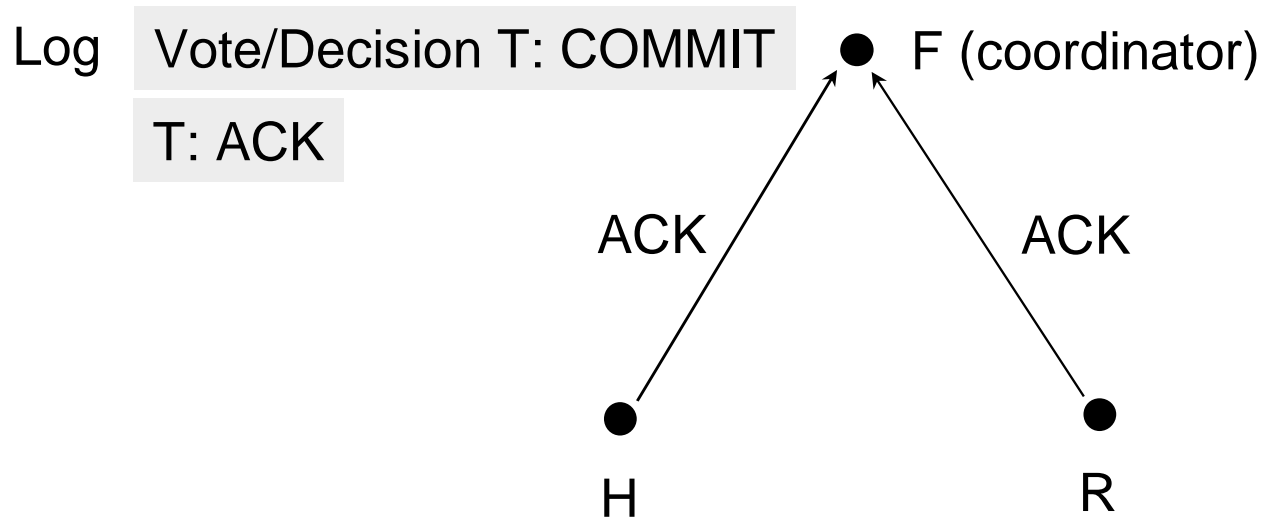
Database

Vote T: COMMIT

Log

2PC Execution – Failure after COMMIT, after Messages Sent (7)

- Introduction
- Terminology
- Atomic Commitment
- System Architecture
- Discussion



<i>Obj</i>	<i>Val</i>
NYC H.	KB <input type="checkbox"/>
Phil H.	KB <input type="checkbox"/>
...	...

Database

<i>Obj</i>	<i>Redo</i>
NYC H.	<input type="checkbox"/>
Phil H.	<input type="checkbox"/>
...	

Vote T: COMMIT

Decision T: COMMIT

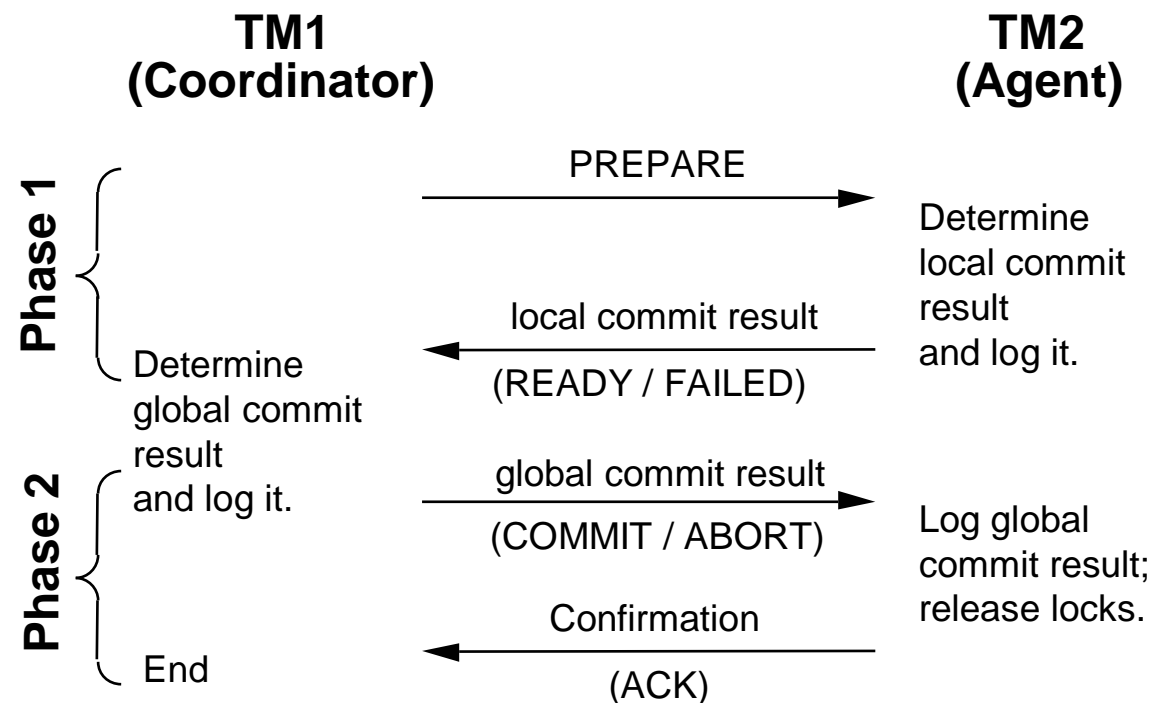
Timeout for ACK Messages

- Not shown in 2PC animations:
timeout in coordinator for ACK messages;
generate log entry which agents
have not responded yet.
Why worth mentioning?

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Basic Protocol

- Two-phase commit (2PC),
- centralized communication structure.
- **Flow of messages:**



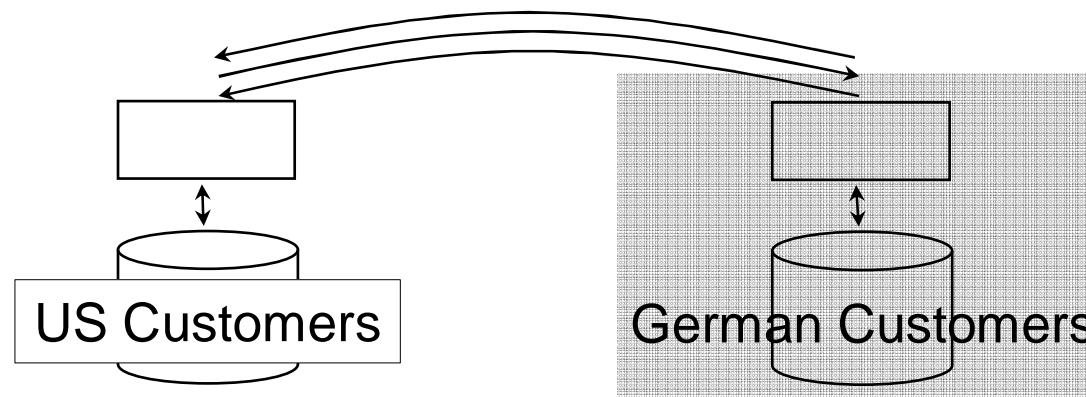
Messages shown occur for each agent.

Z

Atomic Commitment Protocol – Terminology (1)

- *Uncertainty Period:*
time period between vote of a node and notification regarding global decision.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion



Decision
Vote
Prepare

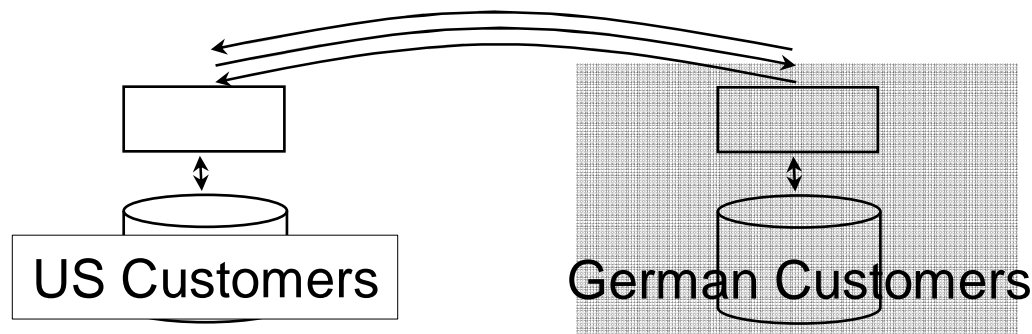
Atomic Commitment Protocol – Terminology (2)

- Terminology might be somewhat confusing:
 - ◆ Vote – nothing democratic; commit only if all participants want it.
 - ◆ Decision – not related to will, discretion, etc. fancy term for schematic step.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Atomic Commitment Protocol – Terminology (3)

- *Process is blocked*, if fixing an error/failure is necessary s.t. it can proceed.
 - ◆ Example: Failure of TM between first and second phase.



Decision
Vote
Prepare

- ◆ By definition, process waiting for timeout is not blocked.
- ◆ Undesired effect; motivation for asking:
Under which circumstances can we avoid blocking?

Introduction
Terminology
Atomic Commitment
System Architecture
Discussion

Atomic Commitment Protocol – Terminology (3)

- *Global state.*
What is current status of protocol?
- *Independent recovery:*
process that is recovering determines global state without any communication – highly desirable.
- Does 2PC feature independent recovery?

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Atomic Commitment Protocol – Characteristics (1)

Atomic Commitment Protocol (ACP):

algorithm to determine decision
(‘Commit or Abort’) such that:

1. All processes coming to a decision come to the same decision,
2. process cannot alter decision once it has been taken.
3. Decision for commit is feasible only if all processes have voted for *Yes*.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Atomic Commitment Protocol – Characteristics (2)

Atomic Commitment Protocol (ACP):

algorithm to determine decision

(,Commit or Abort‘) such that (cont.):

4. No failures, + all processes vote ,yes‘. \Rightarrow Commit.
(I.e., rule out useless protocols
that always decide Abort.)
5. Execution contains a failure, but algorithm
can handle it. After all failures have been fixed
and no new failures for a certain period of time.
 \Rightarrow Decision will eventually be taken.

Introduction

Terminology

Atomic
Commitment

System
Architecture

Discussion

Characteristics of Atomic Commitment Protocols

- Theorem 1: if communication failures and total site failures are possible, each ACP may lead to blocking.
(total site failures – all nodes are down)
- Theorem 2: no ACP can guarantee independent recovery of failed processes.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Variants of Atomic Commitment Protocols

- 2PC: blocking feasible even with site failures only.
Why?
- 3PC – two variants:
 1. Tolerates site failures.
Non-blocking, except for total failures.
Communication failures
→ may result in inconsistencies.
 2. Tolerates both communication and site failures, but blocking.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Atomic Commitment Protocol – Further Requirements (1)

- Further requirements:
 - ◆ *As few messages as possible*, and minimal number of writes to the log file.
 - ◆ High robustness of protocol wrt failures: Keep probability of „blocking“ low.
 - ◆ Each participating node should be entitled to abort global transaction as long as possible (*unilateral abort*).
Why is this desirable?

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Atomic Commitment Protocol – Further Requirements (2)

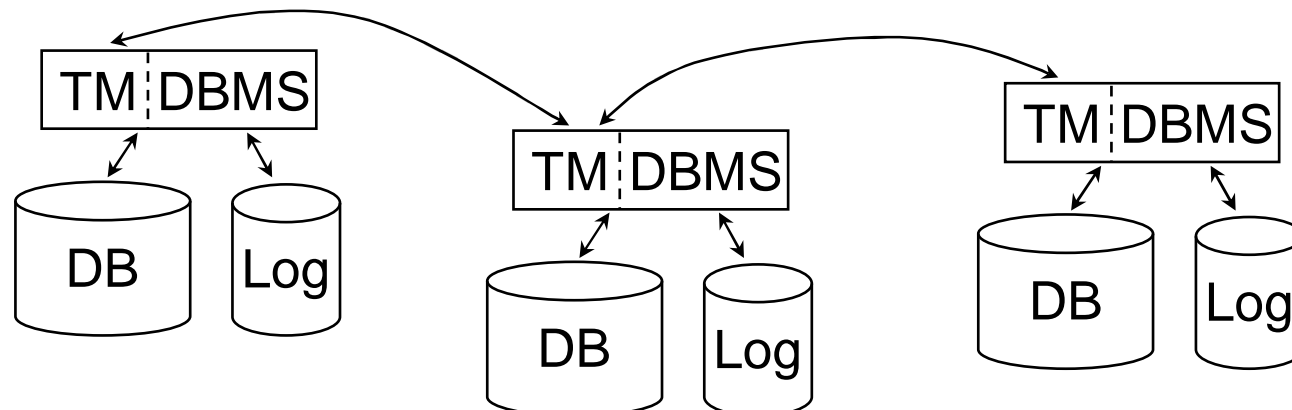
- Conflicting requirements: high robustness typically requires additional messages and I/O operations.
- Assumption: failures are rare
→ optimize design of system for normal mode.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Z

Architecture – Model (1)

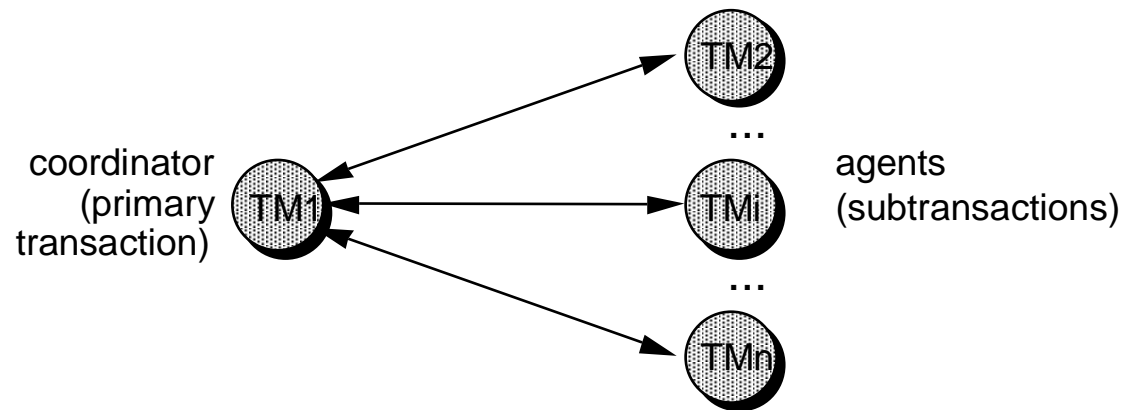
- Transaction Manager (TM) at each node.
- Responsible to administer subtransactions executed locally.
- Cooperates with TMs of the other nodes as part of commit protocol.
- Administers its own log file:
 - ◆ all commit decisions,
 - ◆ database modifications.



Commit Structures (1)

- Different invocation structures (communication between coordinator and agents) conceivable:

1. **centralized commit structure**

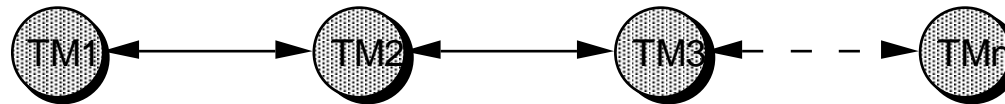


- ◆ Typically no communication between agents.
- ◆ Advantage: processing of commit in parallel at all agents.

Commit Structures (2)

2. linear commit structure

commit is processed sequentially,
but number of messages is reduced.



Two cases:

- ◆ All nodes vote COMMIT.
- ◆ One or more nodes vote ABORT.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

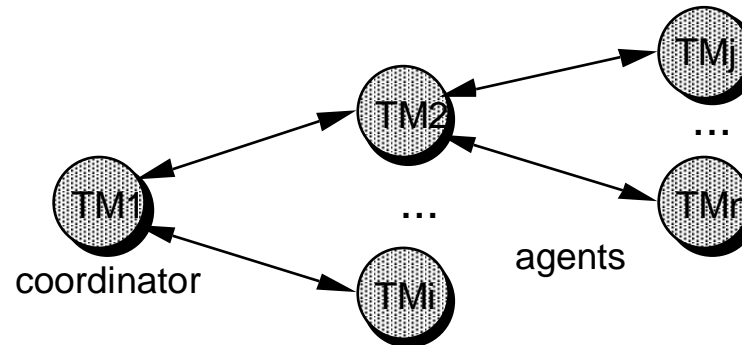
Commit Structures (3)

- Main advantage: number of messages almost halved, as compared to basic scheme; PREPARE messages and $n-2$ ACK messages are avoided. Only $2n-1$ messages remain.
(Basic scheme: $4 \cdot (n-1)$ messages)
One ACK message suffices.
Not one for each node.
- However: commit processing is sequential, if n is large, response time increases significantly; but common case $n=2$ is efficient (3 messages).

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

Commit Structures (4)

3. Hierarchical commit structure



- ◆ takes hierarchical invocation structure within global transaction into account, analogous to transaction tree.
- ◆ Most general structure – previous ones are special case of this one.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

2PC Basic Protocol – Costs

- Four **messages** per agent
→ $4 \cdot (n-1)$ messages
if transaction is distributed over n nodes.
Why $,n-1'$ and not $,n'$?
- For coordinator and each agent two **log writes**
(commit and end as well as prepared and commit);
all log entries take place synchronously,
except for end.
I.e., further processing delayed by duration of I/O.
→ $2 \cdot n$ log operations.
Which ones?

Why can we treat prepared and commit
as one log write in the coordinator?

2PC Basic Protocol – Discussion

- Dependency on coordinator.

Main problem of 2PC:

Coordinator fails → other nodes must wait longer for commit result; still have locks; blocking.

Introduction

Terminology

Atomic

Commitment

System

Architecture

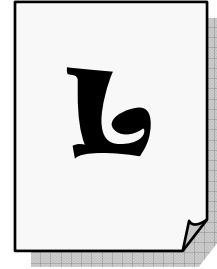
Discussion

Distributed Locking

- Data is partitioned among nodes.
- Each node synchronizes operations that access data on its partition.
- No further communication (except for replicas): distributed execution of transactions and operations already adapted to distribution of data.
- Release of locks with strict 2PL as part of ACP.
- Biggest problem: global deadlocks.

Introduction
Terminology
Atomic
Commitment
System
Architecture
Discussion

2PC Basic Protocol (1)



Steps:

1. EOT: **coordinator** sends PREPARE to all agents at same time.

2PC Basic Protocol (2)



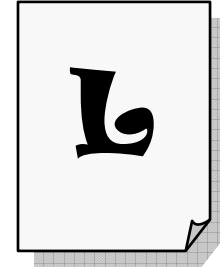
L

2. **Agent** receives PREPARE;
 - ◆ Write log data that has not yet been saved to disk (REDO), together with a ,prepared' log entry; READY message to coordinator, locks are not released at this time.
 - ◆ If subtransaction fails:
write ,abort' log entry to local log file;
FAILED message to coordinator;
reset subtransaction,
release locks and end subtransaction.



Warum?

2PC Basic Protocol (3)



3. Coordinator

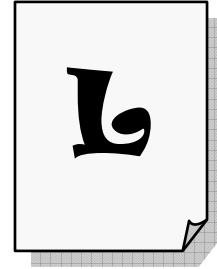
- ◆ writes commit log to local log file, if all agents have replied with READY, Phase 1 ends.

COMMIT message to all agents.

- ◆ If at least one agent sends FAILED, write abort log to local log file; ABORT message to all agents that have sent READY.

Order does not matter.

2PC Basic Protocol (4)



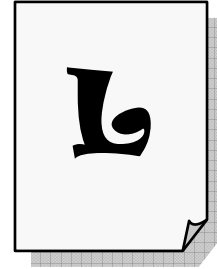
4. Agent

- ◆ receives COMMIT;
writes commit log entry to log file;
releases logs of subtransaction.
- ◆ When receiving ABORT:
Reset subtransaction.
(I.e., must keep UNDO information
until this point of time,
despite EOT and READY.)
Write abort log entry; release locks.

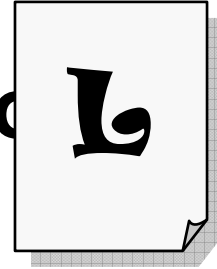
Agent sends confirmation (ACK message)
to coordinator.

2PC Basic Protocol (5)

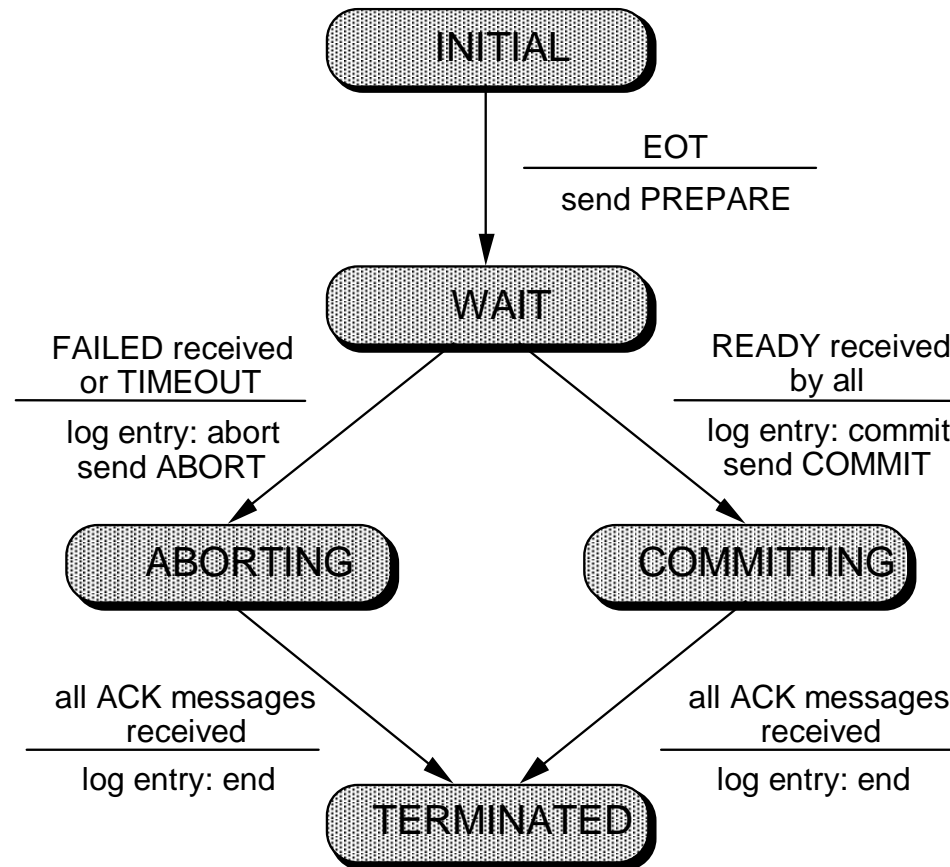
5. **Coordinator** waits for last ACK message; finishes global transaction, writes end log entry to log file.



2PC – State Transitions at Coordinator

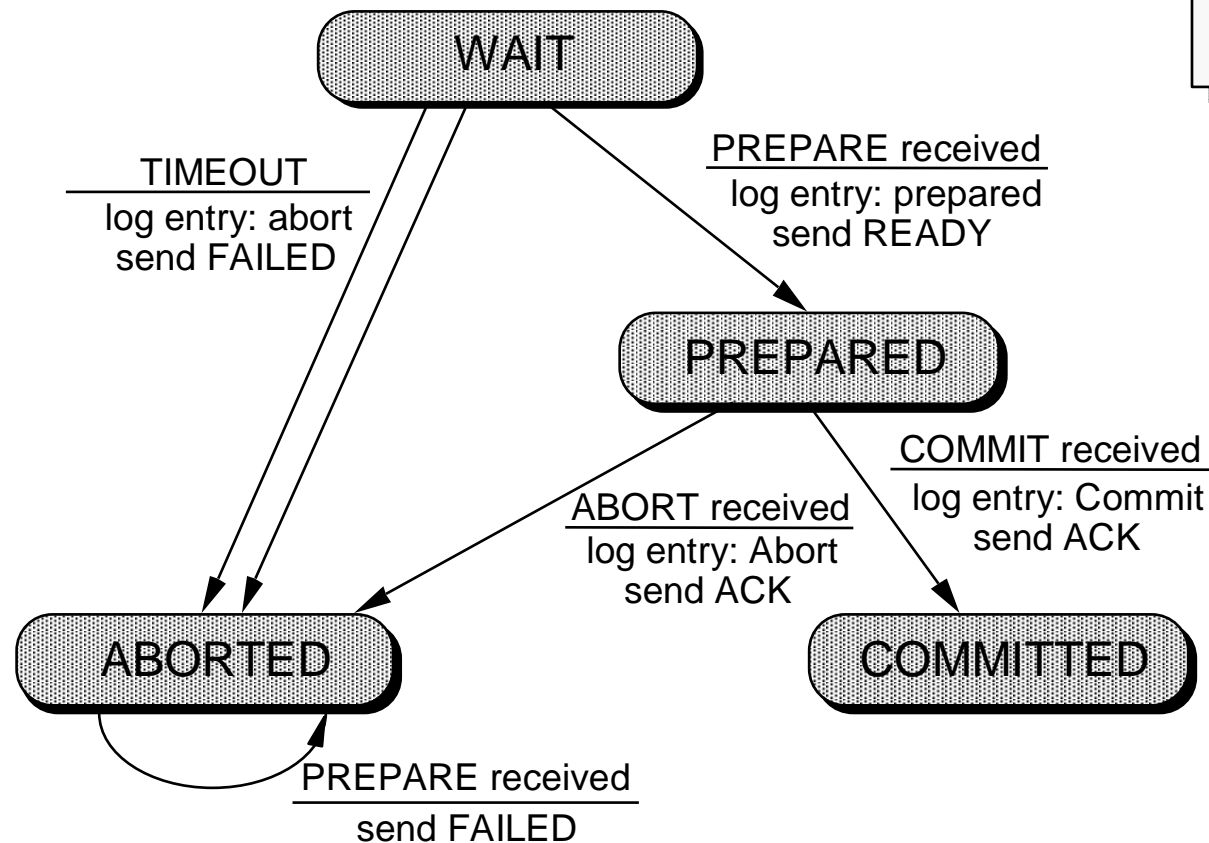
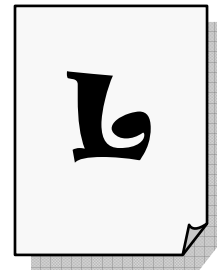


- Correctness in normal mode is obvious;
- in case of failures let us look at state transitions during execution of commit.



Above: action causing transition,
below: actions invoked.

State Transitions – Agent



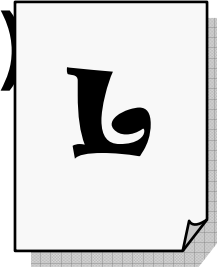
- No TIMEOUT in PREPARED state.
- WAIT – subtransaction finished, waits for PREPARE.

State Transitions – Timeouts



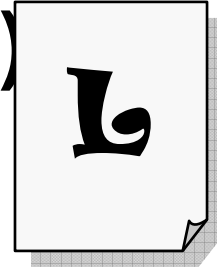
- **Timeout** events, intercept site failures and communication failures.

Recovery after Site Failure (1)



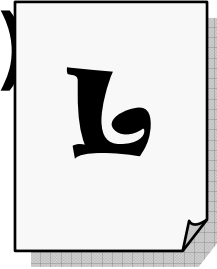
- Use log file to determine commit state that has been reached before failure.
- **Agent nodes:**
 - ◆ Commit entry contained in log file (state: COMMITTED):
global transaction has terminated successfully,
→ local REDO.
 - ◆ Abort entry contained in log file (state: ABORTED):
local transaction has failed,
→ local UNDO.

Recovery after Site Failure (2)



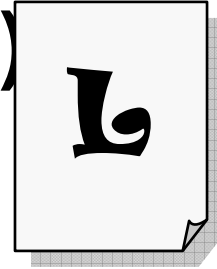
- **Agent nodes** (continuation):
 - ◆ only prepared entry contained in log file (state: PREPARED):
not known how global transaction has ended,
termination protocol
(one of the following transparencies).
 - ◆ None of these three entries contained in log file:
abort transaction –
commit protocol has not yet begun.

Recovery after Site Failure (3)



- **Coordinator node:**
 - ◆ end entry contained in log (state: TERMINATED):
there cannot be any open subtransactions.
→ Local UNDO or REDO,
depending on commit result.
 - ◆ only abort entry contained in log (state: ABORTING): transaction has failed.
→ Local UNDO,
ABORT message to all agents.
(If log tells us which ones
have already received it, do not repeat.)

Recovery after Site Failure (4)



- **Coordinator node** (continuation):
 - ◆ only commit entry contained in log (state: COMMITTING):
transaction has terminated successfully
→ local REDO,
COMMIT message to all agents (i.e., those that are still waiting).
 - ◆ None of these log entries (state: INITIAL or WAIT):
local transaction has failed
→ proceed as in State ABORTING.

Potential Exam Questions

(List is not exhaustive.)

- Give the definitions of the following terms: uncertainty period, blocking, ACP
- Which criteria do you know to evaluate the various ACP?
- Explain the role of timeouts in 2PC?
- How does recovery work with 2PC?