

Add WeChat powcoder



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

8

Distributed Systems

Uwe R. Zimmer - The Australian National University



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Transactions

☞ Assignment Project Exam Help
Concurrency and distribution in systems
with multiple, interdependent interactions?
<https://powcoder.com>

☞ Add WeChat powcoder
Concurrent and distributed
client/server interactions
beyond single remote procedure calls?



Definition (ACID properties):

- Assignment Project Exam Help**
- **Atomicity:** All or none of the sub-operations are performed.
Atomicity helps achieve crash resilience. If a crash occurs, then it is possible to roll back the system to the state before the transaction was invoked.
 - **Consistency:** Transforms the system from one consistent state to another consistent state.
 - **Isolation:** Results (including partial results) are not revealed unless and until the transaction commits. If the operation accesses a shared data object, invocation does not interfere with other operations on the same object.
 - **Durability:** After a commit, results are guaranteed to persist, even after a subsequent system failure.



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Transactions

Definition (ACID properties):

Atomic operations
spanning multiple processes?

How to ensure consistency
in a distributed system?

- **Atomicity:** All or none of the sub-operations are performed.

Atomicity helps achieve crash resilience. If a crash occurs, then it is possible to roll back the system to the state before the transaction was invoked.

- **Consistency:** Transforms the system from one consistent state to another consistent state.

Add WeChat powcoder

- **Isolation:** Results (including partial results) are not revealed unless and until the transaction commits. If the operation accesses a shared data object, invocation does not interfere with other operations on the same object.

Shadow copies?

- **Durability:** After a commit, results are guaranteed to persist, even after a subsequent system failure.

What hardware do we
need to assume?

Actual isolation and
efficient concurrency?

Actual isolation or the
appearance of isolation?



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Transactions

Assignment Project Exam Help

A closer look *inside* transactions:

- **Transactions** consist of a sequence of operations.
- If two operations out of two transactions can be performed *in any order with the same final effect*, they are **commutative** and *not critical* for our purposes.
- **Idempotent** and **side-effect free** operations are by definition *commutative*.
- *All non-commutative operations* are considered **critical operations**.
- Two *critical operations* as part of two different transactions while affecting the same object are called a **conflicting pair of operations**.



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Transactions

Assignment Project Exam Help

A closer look at *multiple* transactions:

- Any *sequential* execution of multiple transactions [will fulfil](https://powcoder.com) the ACID-properties, by definition of a single transaction.
 - A *concurrent* execution (or ‘interleavings’) of multiple transactions [might fulfil](#) the ACID-properties.
- ☞ If a specific *concurrent* execution can be shown to be *equivalent* to a specific sequential execution of the involved transactions then this specific interleaving is called ‘**serializable**’.
- ☞ If a concurrent execution (‘interleaving’) ensures that no transaction ever encounters an inconsistent state then it is said to ensure the **appearance of isolation**.



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Achieving serializability

Assignment Project Exam Help

- ☞ For the **serializability** of two transactions it is necessary and sufficient for the *order* of their invocations of all conflicting pairs of operations to be *the same for all* the objects which are invoked by both transactions.

(Determining order in distributed systems requires logical clocks.)

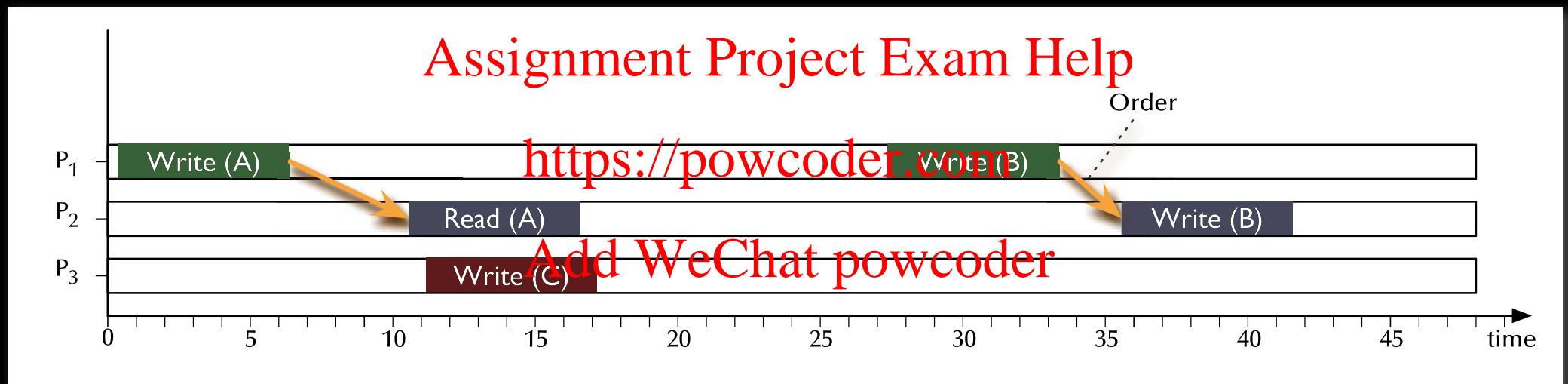


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Two conflicting pairs of operations with the same order of execution.

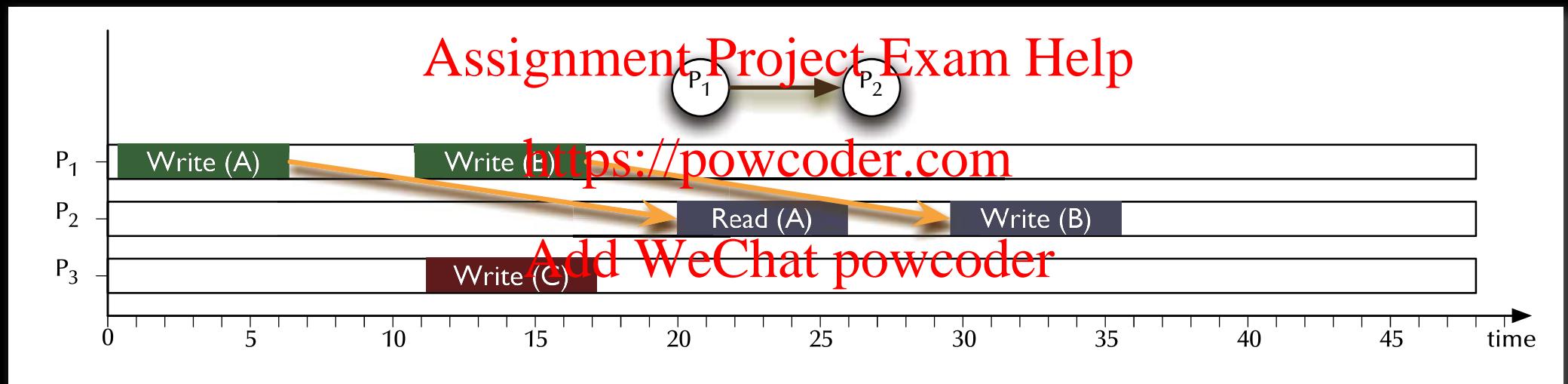


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



☞ Serializable

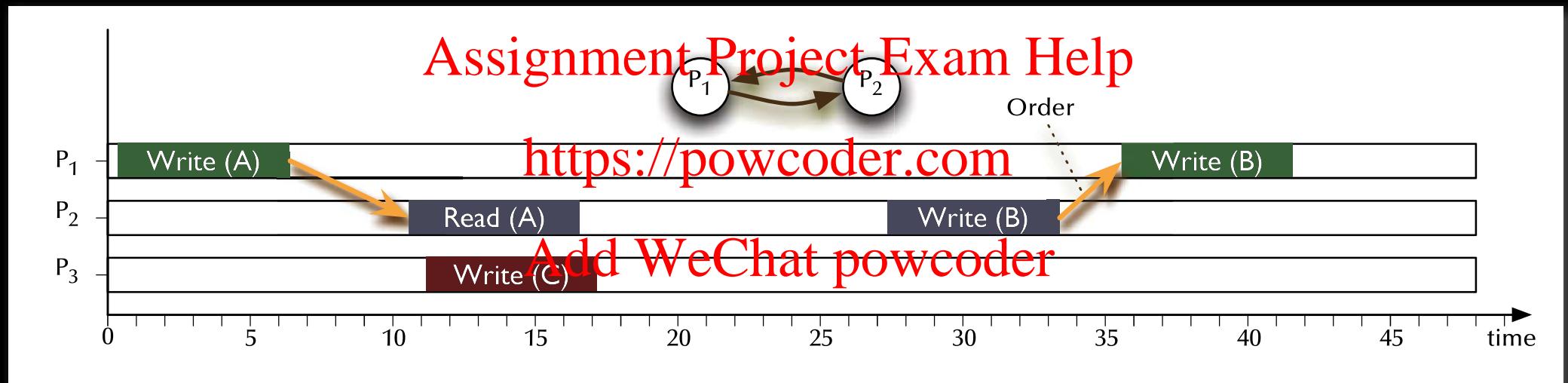


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Two conflicting pairs of operations with different orders of executions.
- ☞ Not serializable.

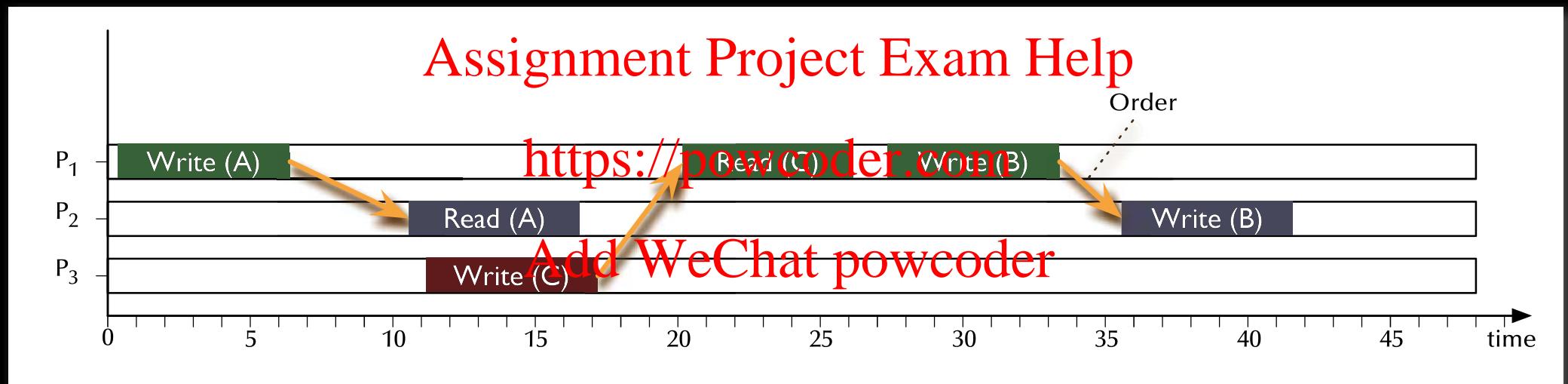


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- The order between processes also leads to a global order of processes.

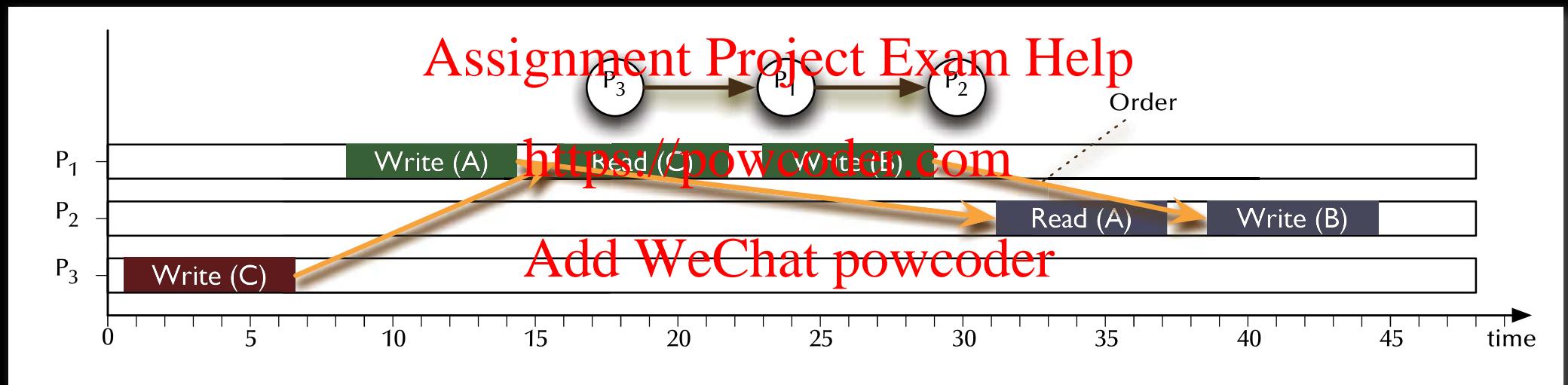


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- The order between processes also leads to a global order of processes.

👉 Serializable

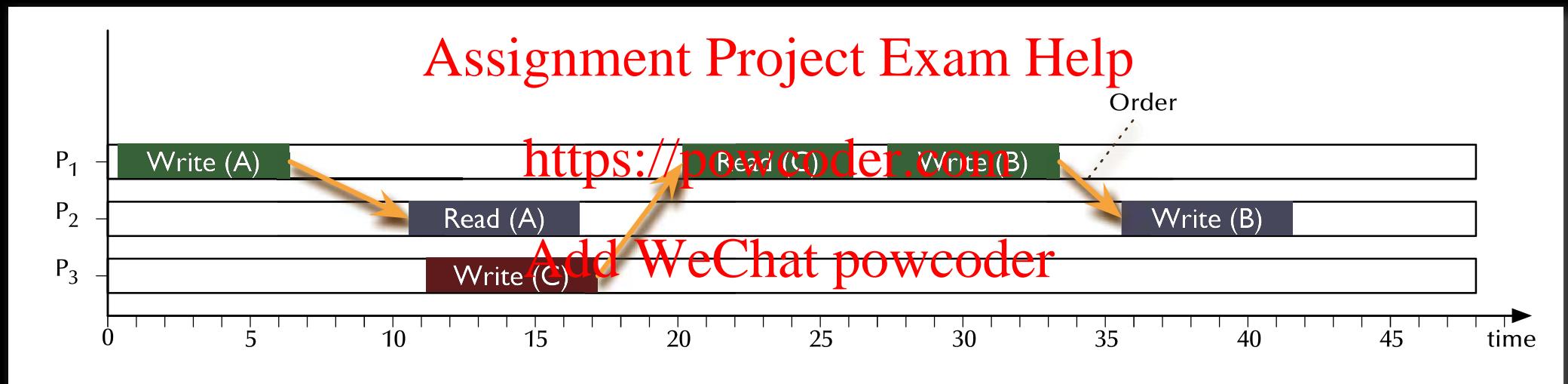


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- The order between processes also leads to a global order of processes.

👉 Serializable

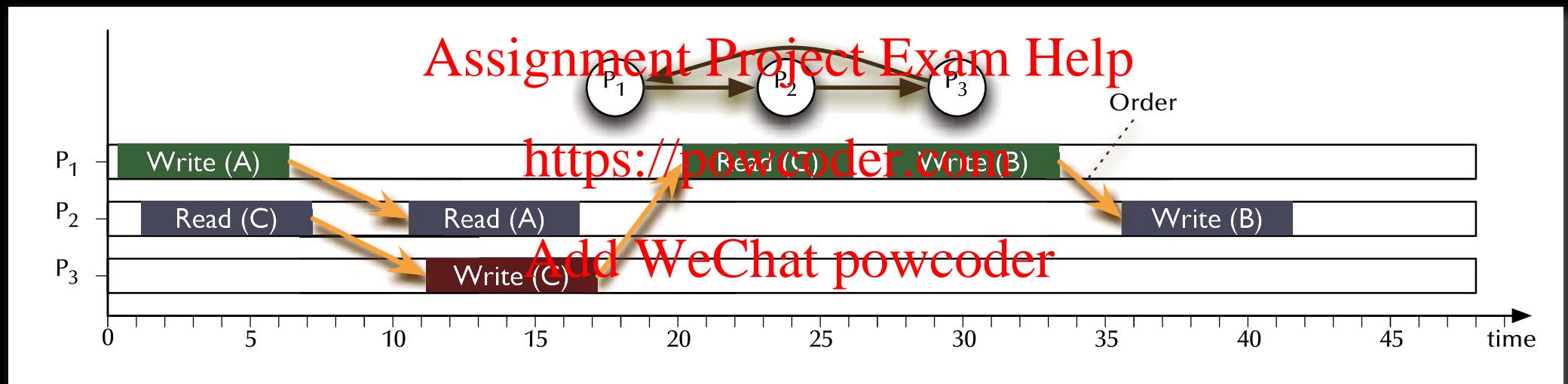


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- The order between processes *does no longer lead to a global order of processes*.

☞ Not serializable



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Achieving serializability

- ☞ For the **serializability** of two transactions it is **necessary and sufficient**

Assignment Project Exam Help

of all conflicting pairs of operations to be *the same*
for all the objects which are invoked by both transactions.

- Define: **Serialization graph**: A directed graph,
Vertices i represent transactions T_i ;
Edges $T_i \rightarrow T_j$ represent an established global order dependency
between all conflicting pairs of operations of those two transactions.

- ☞ For the **serializability** of multiple transactions it is
necessary and sufficient
that the serialization graph is *acyclic*.

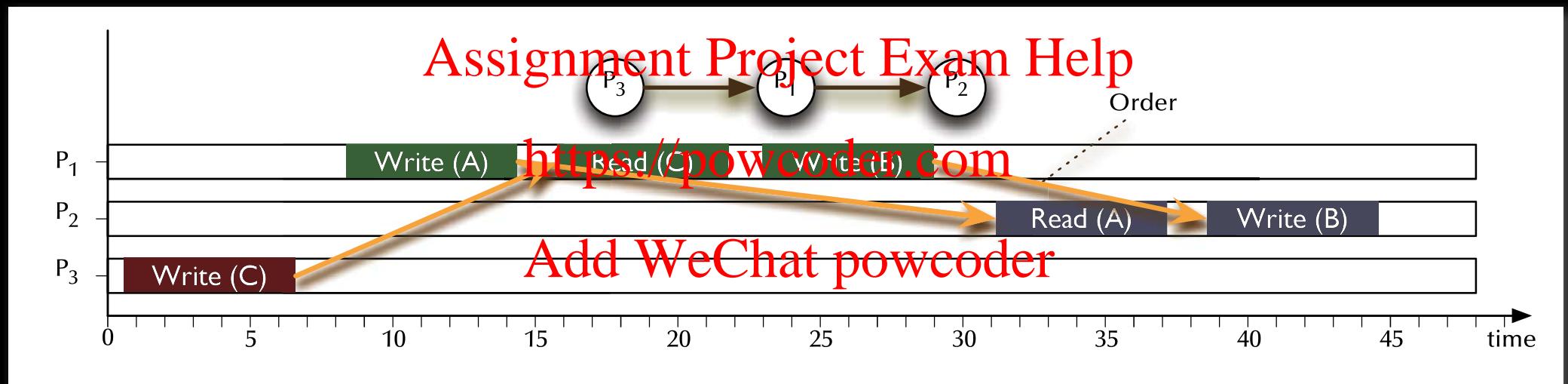


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- ☞ Serialization graph is acyclic.
- ☞ Serializable

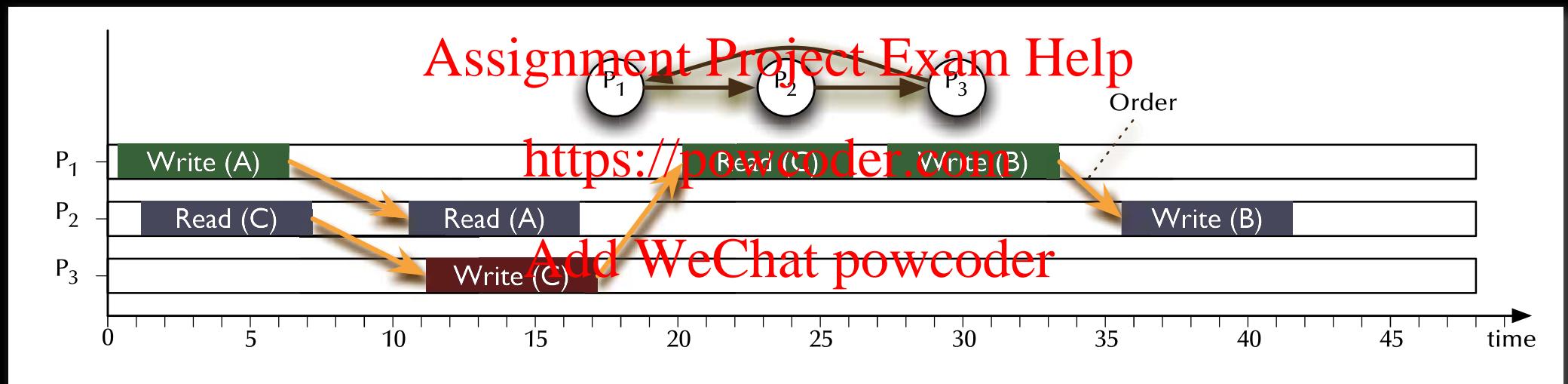


<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder
Serializability



- Three conflicting pairs of operations with the same order of execution (pair-wise between processes).
- ☞ Serialization graph is cyclic.
- ☞ Not serializable



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Transaction schedulers

Three major designs:

[Assignment Project Exam Help](https://powcoder.com)

- **Locking methods:** <https://powcoder.com>
Impose strict mutual exclusion on all critical sections.
- **Time-stamp ordering:** [Add WeChat powcoder](#)
Note relative starting times and keep order dependencies consistent.
- **“Optimistic” methods:**
Go ahead until a conflict is observed – then roll back.



Transaction schedulers – Locking methods

Locking methods include the possibility of deadlocks ↗ careful from here on out ...

- **Complete resource allocation** before the start, and release at the end of every transaction:
 - ↗ This will impose a *strict sequential execution* of all critical transactions.
- **(Strict) two-phase locking:** <https://powcoder.com>
Each transaction follows the following two phase pattern during its operation:
 - *Growing phase*: locks can be acquired, but not released.
 - *Shrinking phase*: locks can be *released anytime*, but not acquired (*two phase locking*) or locks are released *on commit only* (*strict two phase locking*).
- Possible deadlocks
- Serializable interleavings
- Strict isolation (in case of strict two-phase locking)
- **Semantic locking**: Allow for separate read-only and write-locks
 - ↗ Higher level of concurrency (see also: use of functions in protected objects)



<https://powcoder.com>

Distributed Systems Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Transaction schedulers – Time stamp ordering

Add a unique time-stamp (any global order criterion) on every transaction upon start.
Each involved object can inspect the time-stamps of all requesting transactions.

Assignment Project Exam Help

- Case 1: A transaction with a time-stamp *later* than all currently active transactions applies:
 - ☞ the request is accepted and the transaction can go ahead.
 - Alternative case 1 (strict time-stamp ordering):
 - ☞ the request is **delayed** until the currently active earlier transaction has committed.
 - Case 2: A transaction with a time-stamp *earlier* than all currently active transactions applies:
 - ☞ the request is not accepted and the applying transaction is to be **aborted**.
-
- ☞ Collision detection rather than collision avoidance
 - ☞ No isolation ☞ Cascading aborts possible.
 - ☞ Simple implementation, high degree of concurrency
 - also in a distributed environment, as long as a global event order (time) can be supplied.



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Transaction schedulers – Optimistic control

Three sequential phases: Assignment Project Exam Help

1. **Read & execute:**

Create a shadow copy of all involved objects and perform all required operations *on the shadow copy and locally* (i.e. in isolation).

Add WeChat powcoder

2. **Validate:**

After local commit, check all occurred interleavings for serializability.

3. **Update or abort:**

3a. If serializability could be ensured in step 2 then all results of involved transactions are written to all involved objects – *in dependency order of the transactions*.

3b. Otherwise: destroy shadow copies and start over with the failed transactions.



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Transaction schedulers – Optimistic control

Three sequential phases:

Assignment Project Exam Help

How to create a consistent copy?

1. Read & execute:

Create a shadow copy of all involved objects and

perform all required operations *on the shadow copy and locally* (i.e. in isolation).

2. Validate:

After local commit, check all occurred interleavings **for serializability**.

3. Update or abort:

3a. If serializability could be ensured in step 2 then all results of involved transactions
are **written** to all involved objects – *in dependency order of the transactions*.

3b. Otherwise: **destroy** shadow copies and **start over** with the failed transactions.

Aborts happen after everything
has been committed locally.

Full isolation and
maximal concurrency!



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Distributed transaction schedulers

Three major designs:

Assignment Project Exam Help

- **Locking methods:** no aborts
Impose strict mutual exclusion on all critical sections.
<https://powcoder.com>
- **Time-stamp ordering:** Add WeChat powcoder the way
Note relative starting times and keep order dependencies consistent.
- **“Optimistic” methods:** aborts or commits at the very end
Go ahead until a conflict is observed – then roll back.

 How to implement “commit” and “abort” operations
in a distributed environment?



<https://powcoder.com>

Distributed Systems

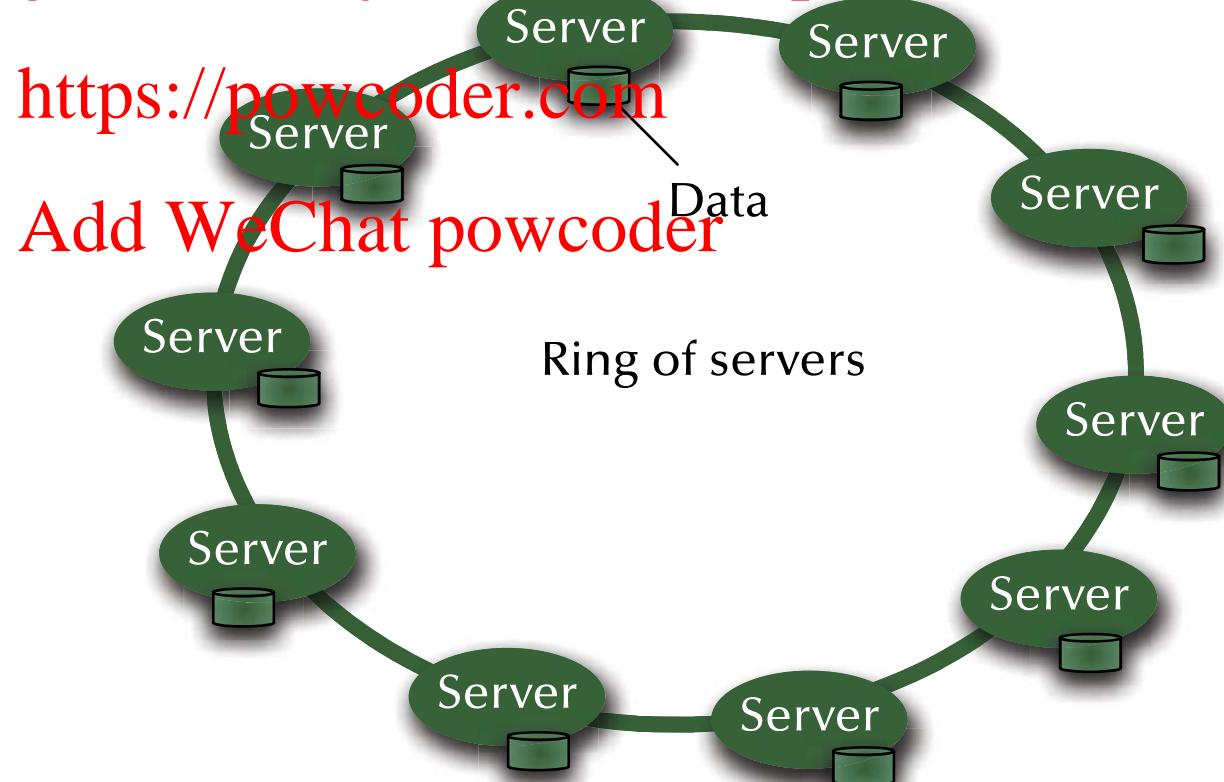
Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Start up (initialization) phase

Assignment Project Exam Help



<https://powcoder.com>

Add WeChat powcoder

Ring of servers



<https://powcoder.com>

Distributed Systems

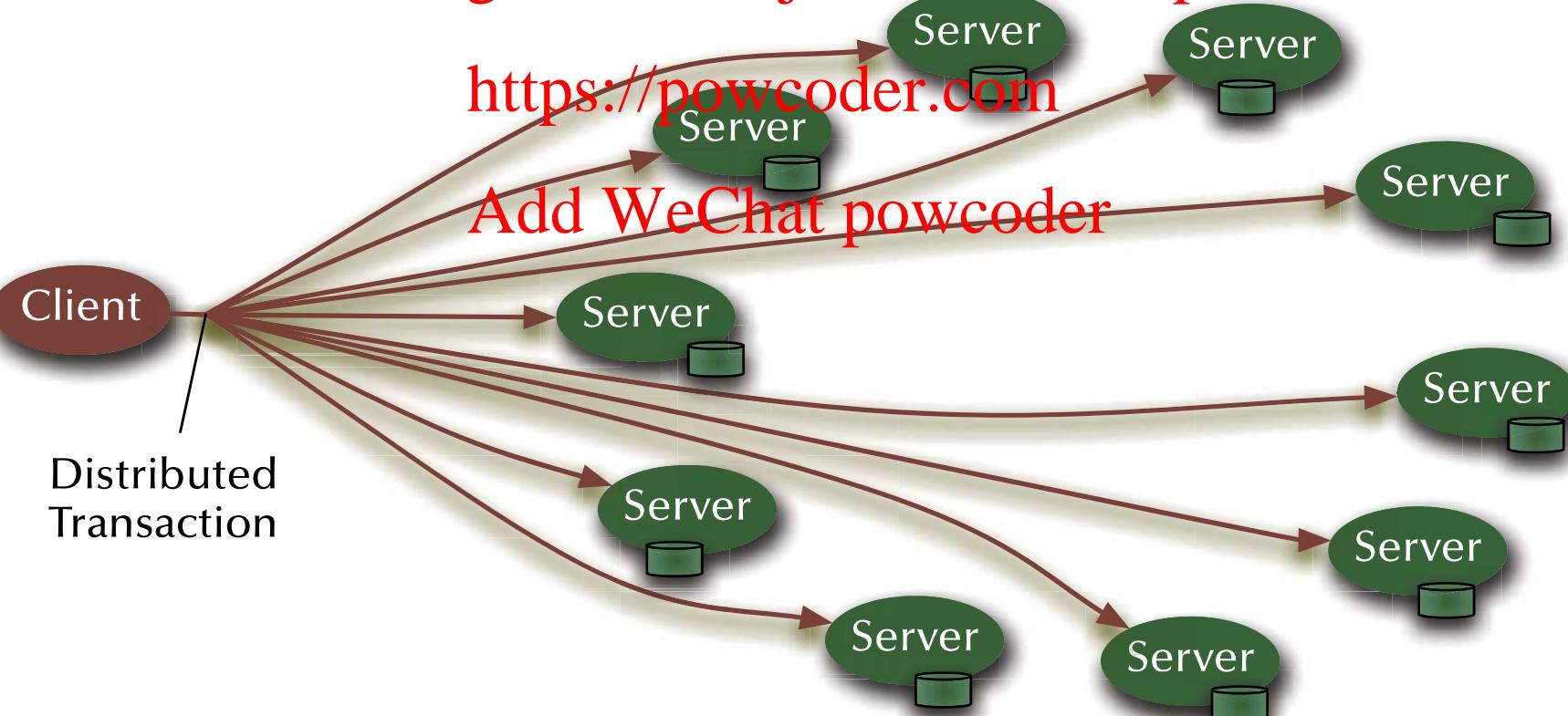
Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Start up (initialization) phase

Assignment Project Exam Help





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Two phase commit protocol

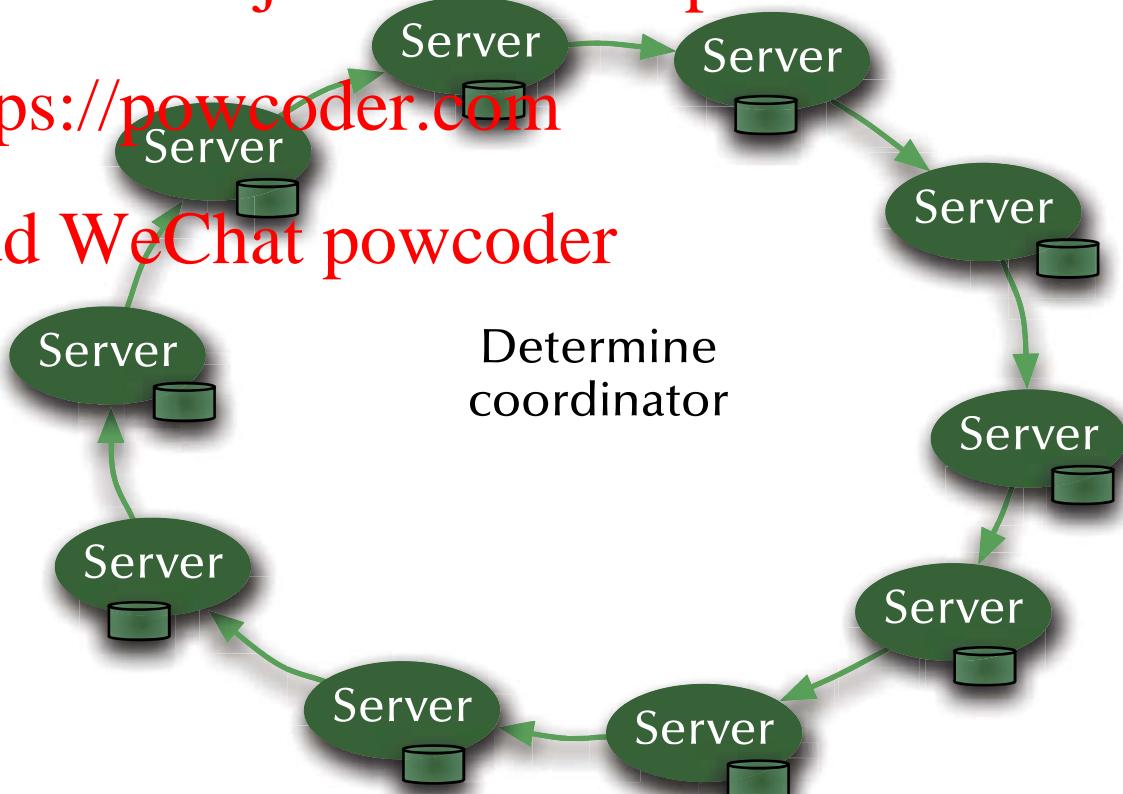
Start up (initialization) phase

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client





<https://powcoder.com>

Distributed Systems

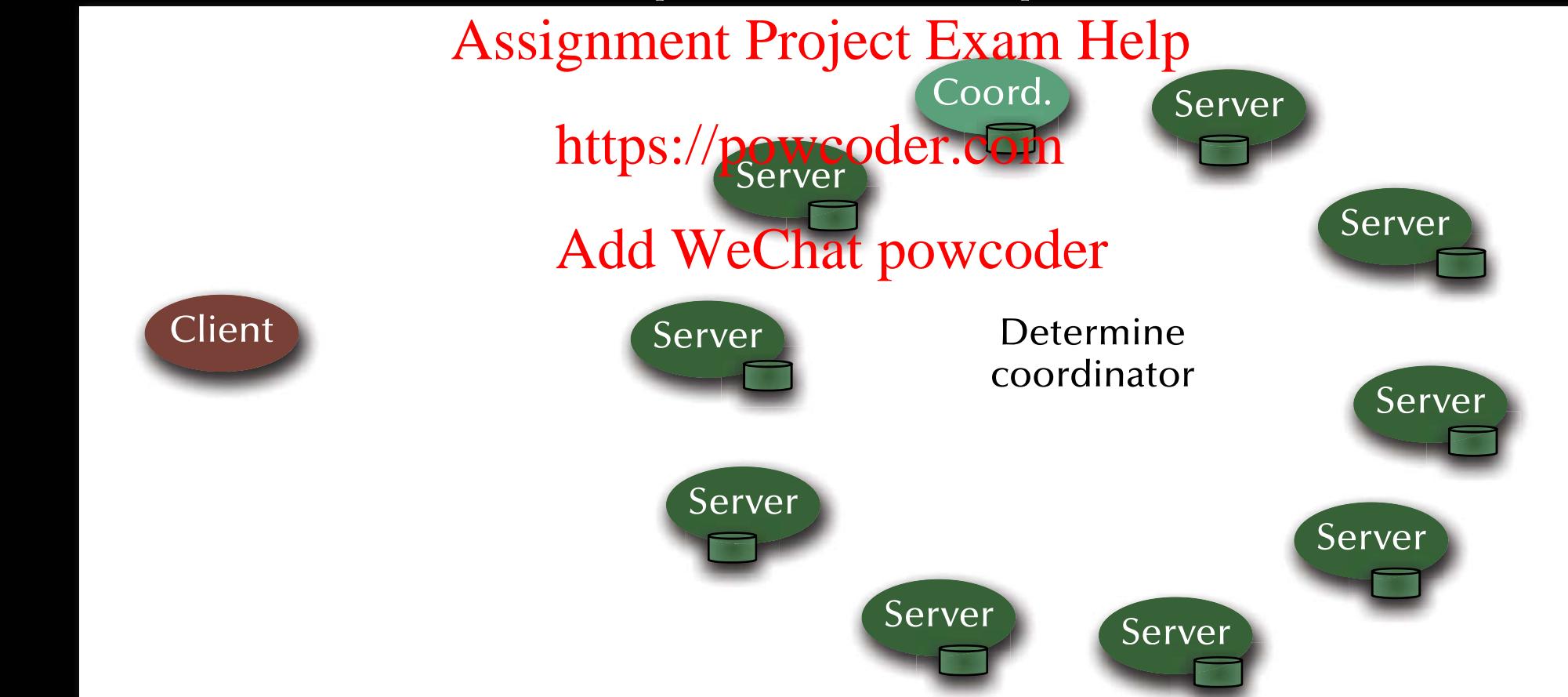
Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Start up (initialization) phase

Assignment Project Exam Help





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

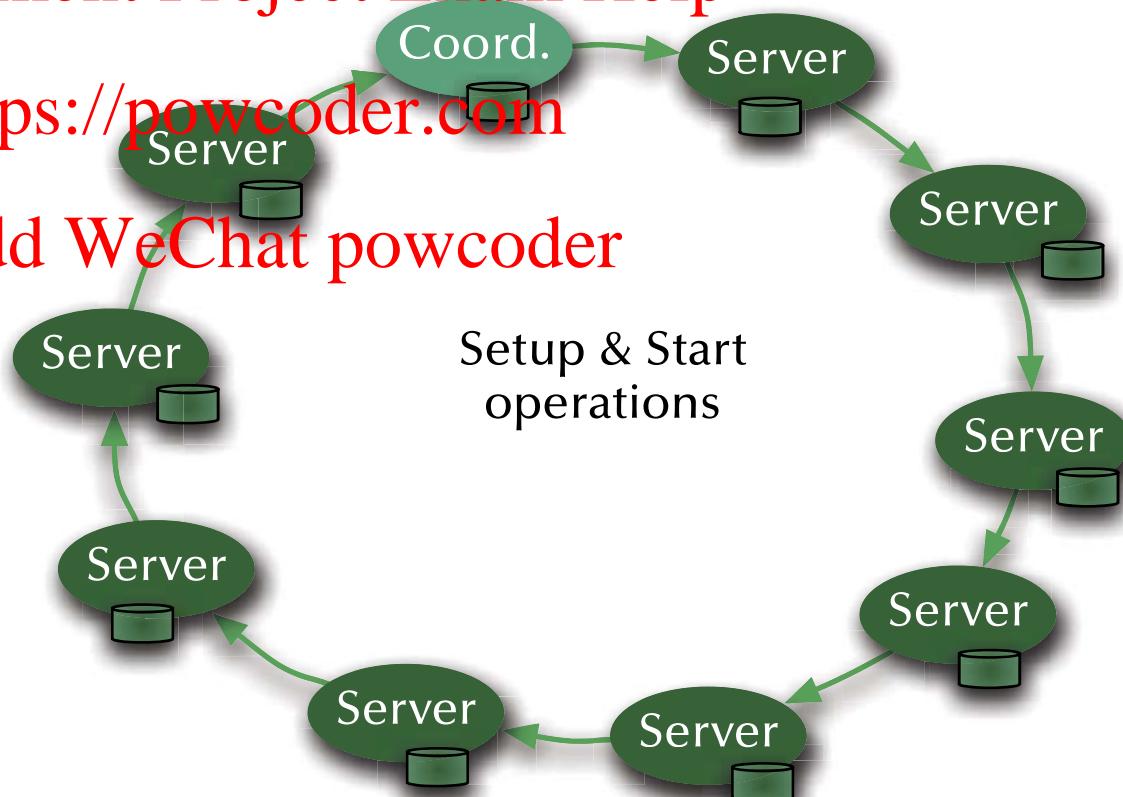
Start up (initialization) phase

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Start up (initialization) phase

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Server

Setup & Start
operations

Server

Server





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

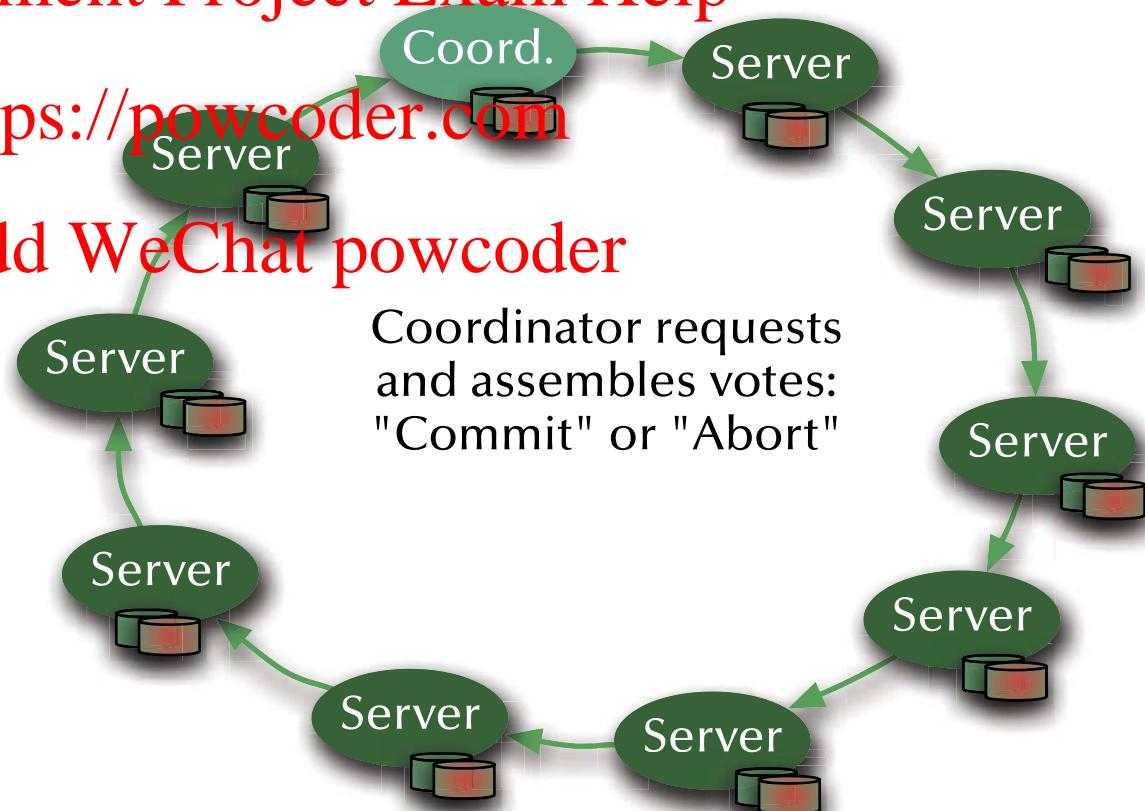
Phase 1: Determine result state

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

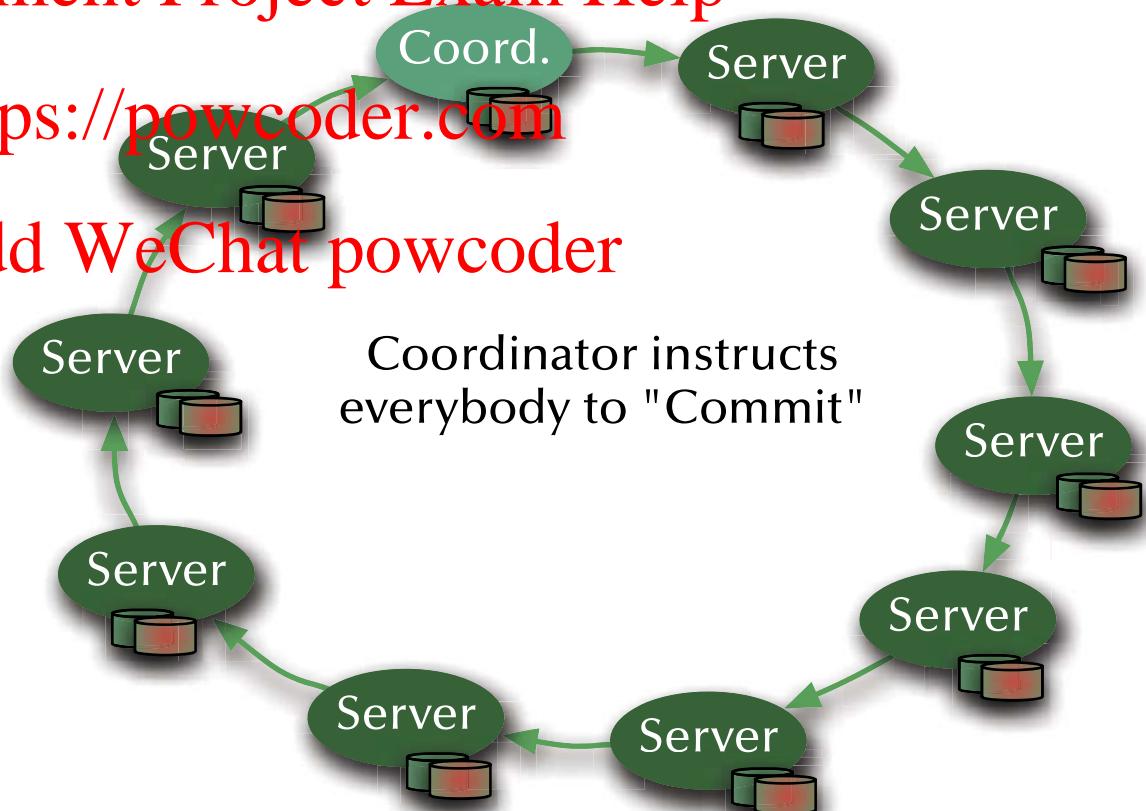
Phase 2: Implement results

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client



Coordinator instructs everybody to "Commit"



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Phase 2: Implement results

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Server

Everybody commits

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Phase 2: Implement results

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Server

Everybody destroys
shadows

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Phase 2: Implement results

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client



Everybody reports
"Committed"



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

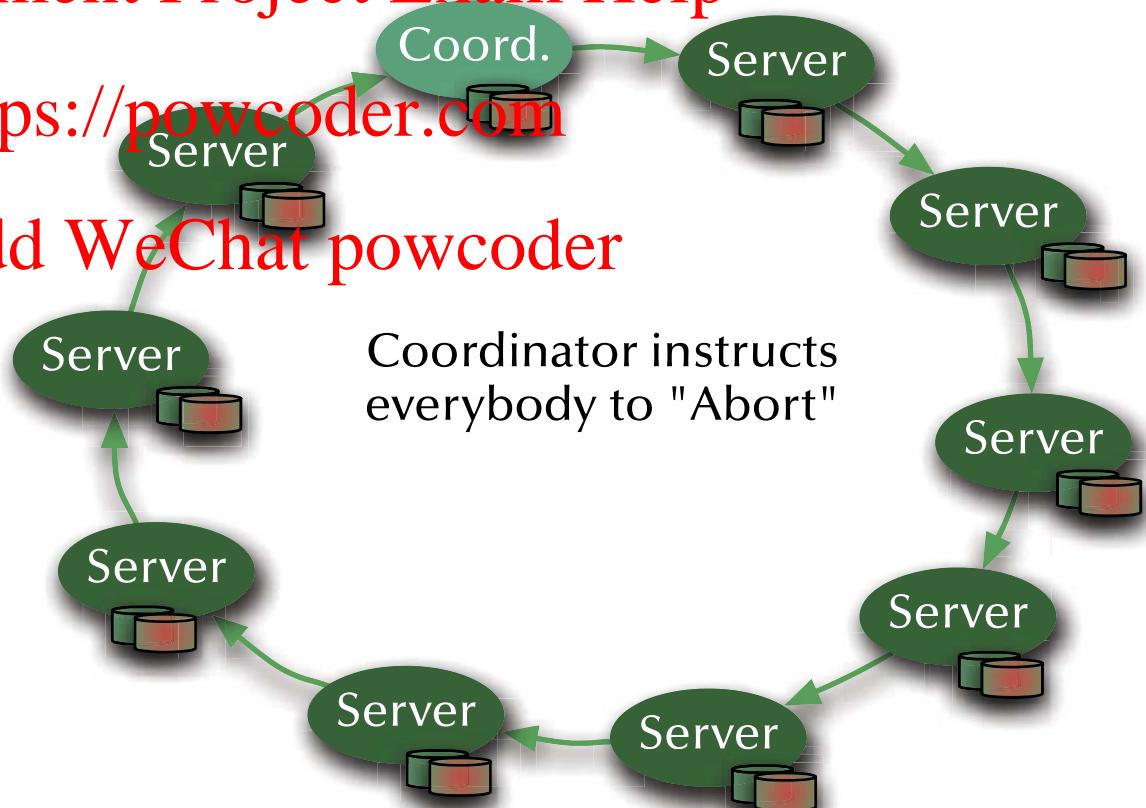
or Phase 2: Global roll back

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client



Coordinator instructs
everybody to "Abort"



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

or Phase 2: Global roll back

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Server

Everybody destroys
shadows

Server

Server

Server

Server



<https://powcoder.com>

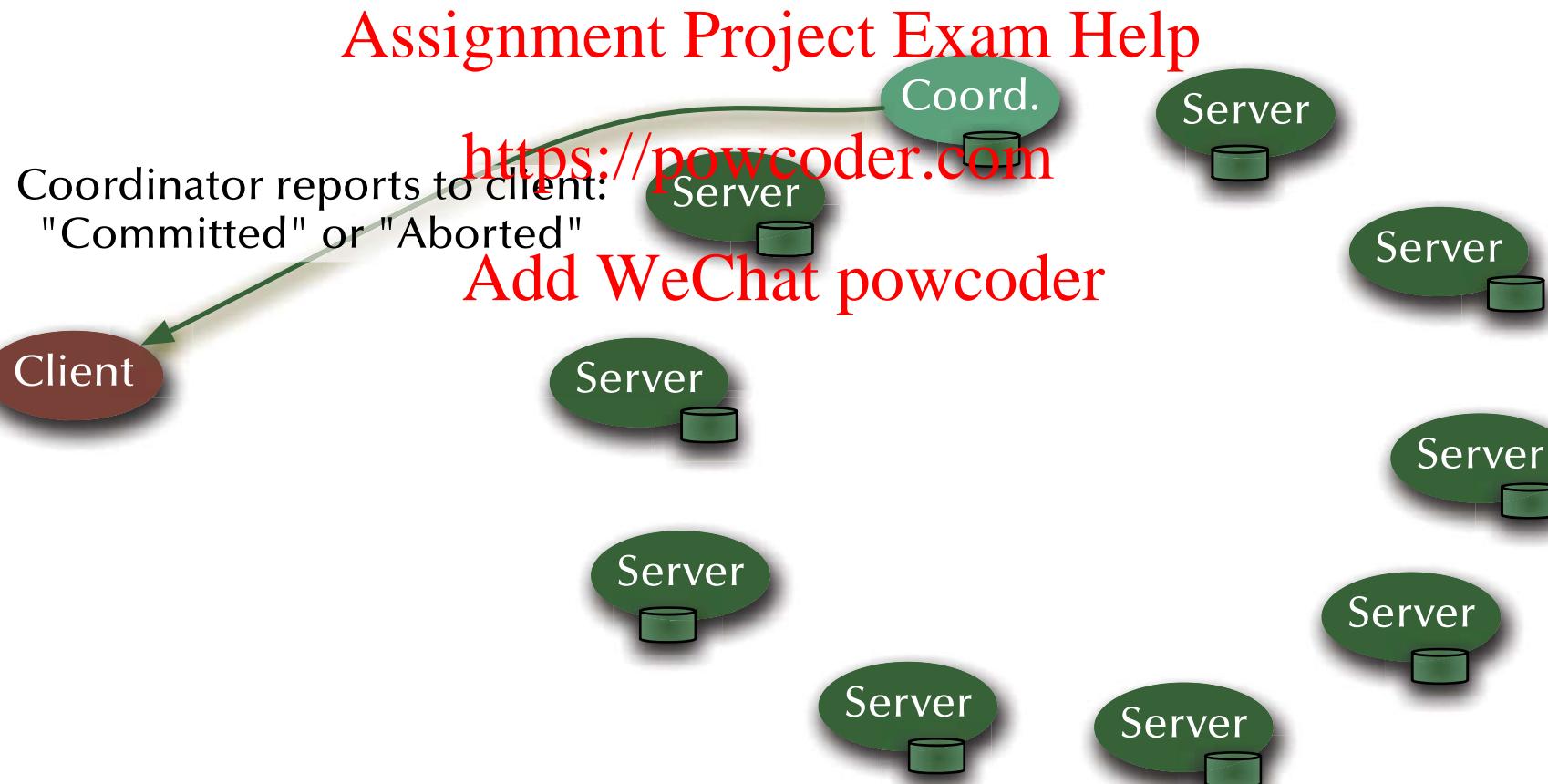
Distributed Systems

Assignment Project Exam Help

Distributed Systems
Add WeChat powcoder

Two phase commit protocol

Phase 2: Report result of distributed transaction





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Distributed transaction schedulers

Evaluating the three major design methods in a distributed environment:

Assignment Project Exam Help

- **Locking methods:** ↗ No aborts.
Large overheads; Deadlock detection/prevention required.
- **Time-stamp ordering:** ↗ Potential aborts along the way.
Recommends itself for distributed applications, since decisions are taken locally and communication overhead is relatively small.
- **“Optimistic” methods:** ↗ Aborts or commits at the very end.
Maximizes concurrency, but also data replication.
 - ↗ Side-aspect “data replication”: large body of literature on this topic
(see: distributed data-bases / operating systems / shared memory / cache management, ...)



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Premise:

A crashing server computer should not compromise the functionality of the system
(full fault tolerance)

Assumptions & Means:

<https://powcoder.com>

- k computers inside the server cluster might crash without losing functionality.
 - ☞ Replication: at least $k + 1$ servers.
- The server cluster can reorganize any time (and specifically after the loss of a computer).
 - ☞ Hot stand-by components, dynamic server group management.
- The server is described fully by the current state and the sequence of messages received.
 - ☞ State machines: we have to implement consistent state adjustments (re-organization) and consistent message passing (order needs to be preserved).

[Schneider1990]



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

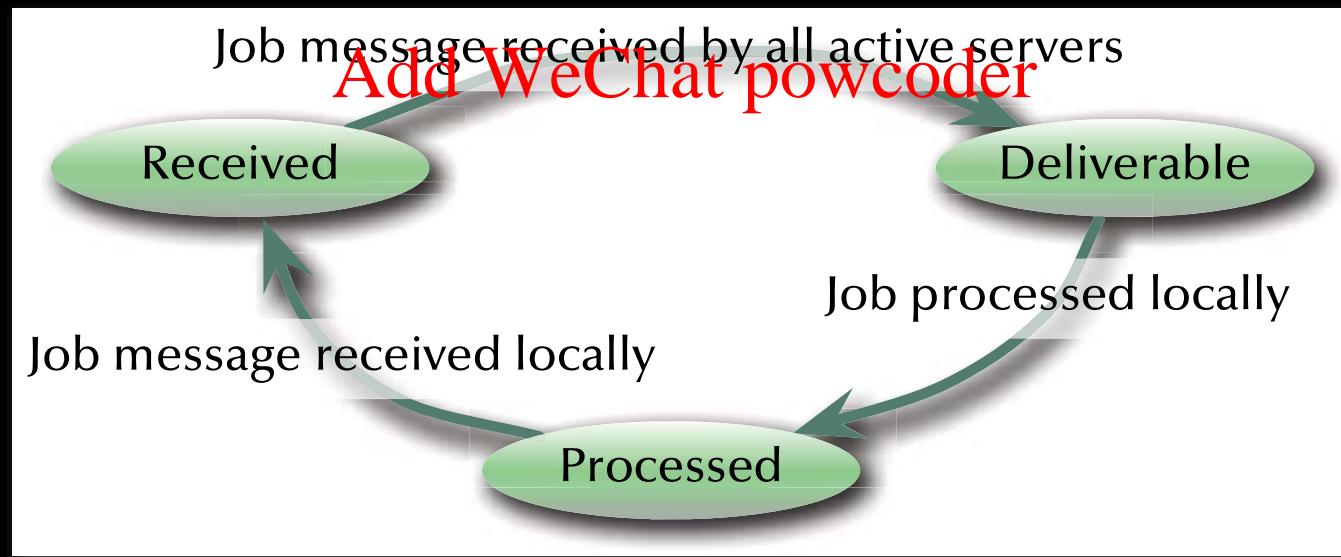
Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Assignment Project Exam Help

<https://powcoder.com>





<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

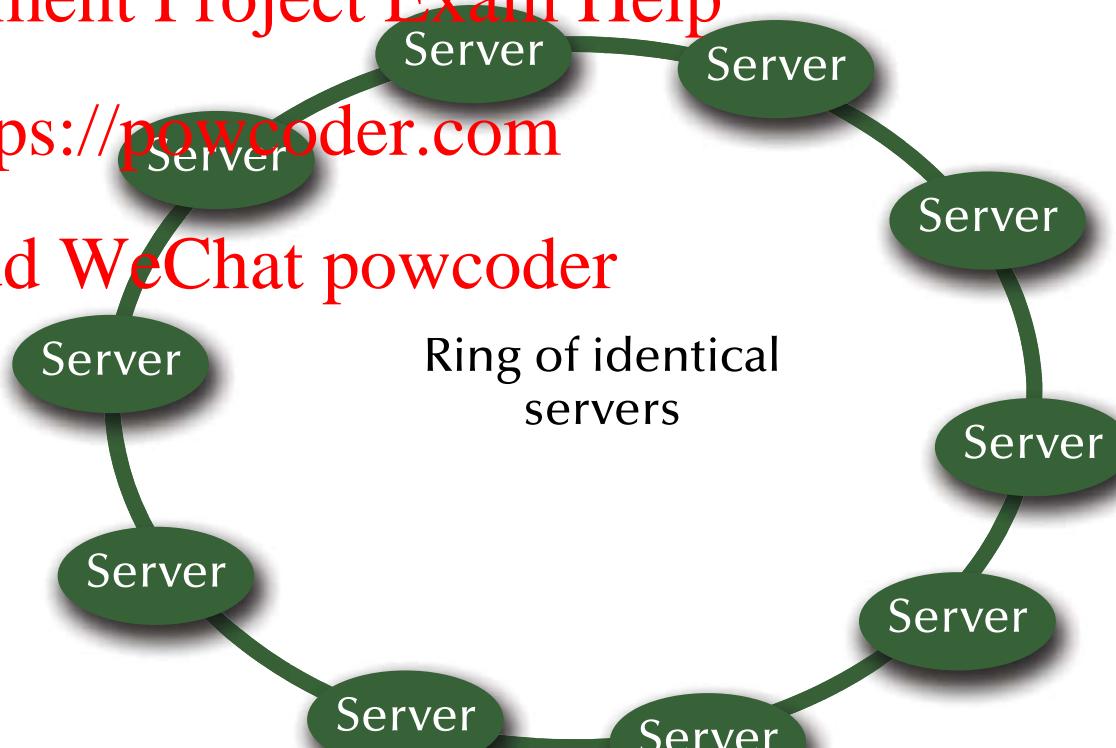
Redundancy (replicated servers)

Start-up (initialization) phase

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



Ring of identical
servers



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Start-up (initialization) phase

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Determine
coordinator



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Start-up (initialization) phase

Assignment Project Exam Help

Coord.

Server

Server

Server

Add WeChat powcoder

Client

Server

Coordinator
determined

Server

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Coordinator receives job message

Assignment Project Exam Help

Send Job

Client

Coord.
Server

Server

Server

Server

Server

Server

Server

Add WeChat powcoder

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Distribute job

Assignment Project Exam Help

Coord.

Server

<https://powcoder.com>

Server

Server

Client

Server

Coordinator sends
job both ways

Server

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Distribute job

Assignment Project Exam Help

Coord.

Server

Server

Server

Server

Server

Server

Server

Add WeChat powcoder

Everybody received job
(but nobody
knows that)

Client

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Processing starts

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Client

Server

Server

First server detects
two job-messages
☞ processes job

Server

Server

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Everybody (besides coordinator) processes

Assignment Project Exam Help

Coord.

Server

<https://powcoder.com>

Server

Server

Server

Client

Add WeChat powcoder

All server detect
two job-messages
☞ everybody
processes job

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Coordinator processes

Assignment Project Exam Help

Coord.

Server

<https://powcoder.com>

Server

Server

Server

Add WeChat powcoder

Client

Coordinator also
received two messages
and processes job

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Result delivery

Assignment Project Exam Help

Coord.

Coordinator delivers
his local result

Add WeChat powcoder

Client

Server

Server

Server

Server

Server

Server

Server

Server



<https://powcoder.com>

Distributed Systems Assignment Project Exam Help

Distributed Systems

Add WeChat powcoder

Redundancy (replicated servers)

Event: Server crash, new servers joining, or current servers leaving.
[Assignment Project Exam Help](https://powcoder.com)

- ☞ Server re-configuration triggered by a message to all
(this is assumed to be supported by the distributed operating system).

Add WeChat powcoder

Each server on reception of a re-configuration message:

1. Wait for local job to complete or time-out.
2. Store local consistent state S_i .
3. Re-organize server ring, send local state around the ring.
4. If a state S_j with $j > i$ is received then $S_i \Leftarrow S_j$
5. Elect coordinator
6. Enter 'Coordinator-' or 'Replicate-mode'



<https://powcoder.com>

Distributed Systems

Assignment Project Exam Help

Add WeChat powcoder
Summary

Distributed Systems

- **Networks**

- OSI, topologies
- Practical network standards

Assignment Project Exam Help

<https://powcoder.com>

- **Time**

- Synchronized clocks, virtual (logical) times
- Distributed critical regions (synchronized, logical, token ring)

Add WeChat powcoder

- **Distributed systems**

- Elections
- Distributed states, consistent snapshots
- Distributed servers (replicates, distributed processing, distributed commits)
- Transactions (ACID properties, serializable interleavings, transaction schedulers)