Easy Commit: A Non-blocking Two-phase Commit Protocol

Suyash Gupta, Mohammad Sadoghi
Dept. of Computer Science
University of California Davis

March 28, 2018
WE ARE CHANGING THE WORLD!!!
JUST KIDDING!
Motivation

Concerns of a Distributed Database Designer:
- Availability
- Consistency

Lots of hardware, resources \(\Rightarrow\) Availability Solved.

If Not Consistent \(\Rightarrow\) Correctness Compromised!
Motivation

- Concerns of a Distributed Database Designer:
  - Availability
  - Consistency
- Lots of hardware, resources $\Rightarrow$ Availability Solved.
- If Not Consistent $\Rightarrow$ Correctness Compromised!
- Need for Agreement!
  - Two Phase Commit Protocol (2PC)? $\Rightarrow$ Blocked!
  - Three Phase Commit Protocol (3PC)? $\Rightarrow$ Heavy!
Concerns of a Distributed Database Designer:
- Availability
- Consistency

Lots of hardware, resources ⇒ Availability Solved.

If Not Consistent ⇒ Correctness Compromised!

Need for Agreement!
- Two Phase Commit Protocol (2PC)? ⇒ Blocked!
- Three Phase Commit Protocol (3PC)? ⇒ Heavy!

Easy Commit ⇒ Two Phases! Non-Blocking!
Two Phase Commit Protocol

Coordinator
Start Commit

Participant
Prepare

INITIAL
Two Phase Commit Protocol

Coordinator

Start Commit

INITIAL

WAIT

Participant

Prepare

INITIAL

READY

ABORT

Votes

Vote-Commit

Prepare

Start Commit

INITIAL

WAIT
**Two Phase Commit Protocol**

- **Coordinator**
  - Start Commit
  - Initial
  - Wait
  - Prepare
  - Global decision
  - Any abort?
  - All Commit?
  - Acks
  - ABORT
  - COMMIT

- **Participant**
  - Prepare
  - Initial
  - Ready
  - Vote-Commit
  - Vote-Abort
  - Global decision
  - Acks
  - ABORT
  - COMMIT
  - Global-abort Ack
  - Global-commit Ack
2PC is Blocking

Coordinator

Participant

Participant

Participant
2PC is Blocking
2PC is Blocking

Coordinator

Participant

GLOBAL ABORT

Participant

Participant

Participant
2PC is Blocking
2PC is Blocking

Diagram:
- Coordinator
- Participant (x3)
- ABORT (message from Participant to Coordinator)
- GLOBAL ABORT (message from Coordinator to Participants)
2PC is Blocking

Coordinator

Participant

Participant

Participant

GLOBAL ABORT

ABORT
2PC is Blocking

Diagram:
- Coordinator
- Participant
- Participant
- Participant

Abort
GLOBAL ABORT
2PC **is Blocking**

System is in an Unstable State.
THREE PHASE COMMIT PROTOCOL

Coordinator
Start Commit

Participant
Prepare

INITIAL

INITIAL
**Three Phase Commit Protocol**

- **Coordinator**
  - Start Commit
  - INITIAL
  - WAIT

- **Participant**
  - Prepare
  - INITIAL
  - READY
  - ABORT
  - Vote-Commit
  - Vote-Abort
  - Votes
THREE PHASE COMMIT PROTOCOL

Coordinator

Start Commit

INITIAL

Prepare

WAIT

Votes

Any abort? All Commit?

ABORT PRE-COMMIT

Participant

Prepare

INITIAL

Votes

Vote-Abort

Vote-Commit

READY

Global message

ABORT PRE-COMMIT
THREE PHASE COMMIT PROTOCOL
3PC is Non-Blocking

- Dale Skeen’s Requirements [SIGMOD 1981]:
  - No state adjacent to both Abort and Commit states.
  - No non-committable state adjacent to the Commit state.
3PC Works

Diagram:
- Coordinator
- Participant
- Participant
- Participant

Three arrows labeled "PREPARE" connect the Coordinator to each Participant.

Suyash Gupta
EDBT 2018
March 28, 2018
**3PC Works**

The 3PC protocol works by involving three parties: Coordinator and two participants. The Coordinator initiates the process by sending a vote to COMMIT to one participant. If the vote COMMIT is received from two participants, the Coordinator commits the transaction. If any participant votes to ABORT, the Coordinator aborts the transaction. This ensures atomicity and consistency in distributed systems.
3PC Works

Coordinator

Participant

GLOBAL
ABORT

Participant

Participant
3PC Works

Coordinator

Participant

Participant

Participant

GLOBAL ABORT
3PC Works

Coordinator

Participant

Participant

Participant

ABORT

GLOBAL ABORT
3PC WORKS

- Coordinator
- Participant
- Participant
- Participant

GLOBAL ABORT
ABORT
3PC Works

Coordinator

Participant

Participant

Participant

GLOBAL ABORT

ABORT
**3PC Works**

- Coordinator
- Participant
- Participant
- Participant
- WHAT?
- WHAT?
-GLOBAL ABORT
- ABORT

Suyash Gupta
EDBT 2018
March 28, 2018
**3PC Works**

- Coordinator
- Participant
- Participant
- Participant

- ABORT
- GLOBAL ABORT
- LET US DISCUSS
3PC Works

Coordinator

Participant

Participant

Participant

System is in a Stable State.
Easy Commit Principle

- First Transmit and then Commit
- Message Redundancy
Easy Commit Protocol

Coordinator:
- Start Commit
  - INITIAL

Participant:
- Prepare
  - INITIAL
Easy Commit Protocol
**Easy Commit Protocol**

**Coordinator**
- Start Commit
- **INITIAL**
  - Prepare
- **WAIT**
  - Global decision
  - Any abort?
    - ABORT
  - All Commit?
    - COMMIT

**Participant**
- Prepare
- **INITIAL**
  - Votes
- Vote
  - Global decision
  - ABORT
  - COMMIT

---

**Suyash Gupta**

EDBT 2018

March 28, 2018
Easy Commit Observations

- Participant cannot directly transition from **INITIAL** to **ABORT**.

- Each participant forwards the global decision to every other node.

- Participant need not wait for global decision from coordinator.

- Existence of **hidden** states: **TRANSMIT-A** and **TRANSMIT-C**.
Easy Commit Protocol – Logical Expansion

Coordinator
Start Commit
INITIAL

Participant
Prepare
INITIAL
Easy Commit Protocol – Logical Expansion

Coordinator

Start Commit

INITIAL

Prepare

WAIT

Participant

Prepare

Votes

INITIAL

Vote

READY
Easy Commit Protocol – Logical Expansion

Coordinator

Start Commit

INITIAL

Prepare

WAIT

Votes

Any abort?

All Commit?

TRANSMIT-A

TRANSMIT-C

Participant

Prepare

INITIAL

Votes

Vote

Global decision

TRANSMIT-A

TRANSMIT-C
Easy Commit Protocol – Logical Expansion

Coordinator

Start Commit

INITIAL

WAIT

Any abort?

All Commit?

TRANSMIT-A

TRANSMIT-C

Global Abort

Global Commit

ABORT

COMMIT

Participant

Prepare

INITIAL

Vote

READY

Global decision

TRANSMIT-A

TRANSMIT-C

Global Abort

Global Commit

ABORT

COMMIT
**Easy Commit Termination Protocol**

- **Coordinator Timeout in WAIT state:**
  - Coordinator didn’t receive Votes.
  - Adds a log entry, Transmits Global-Abort, Aborts Transaction.

- **Participant Timeout in INITIAL state:**
  - Participant didn’t receive PREPARE message.
  - Communicates with other participants.

- **Participant Timeout in READY state:**
  - Participant didn’t receive Global Decision.
  - Communicates with other participants.
Easy Commit Works
Easy Commit Works
Easy Commit Works

Coordinator

GLOBAL ABORT

Participant

Participant

Participant
Easy Commit Works
Easy Commit Works
Easy Commit Works

Coordinator

Participant

Participant

Participant

GLOBAL ABORT

ABORT
EASY COMMIT WORKS

Coordinator

Participant

Participant

Participant

GLOBAL ABORT

ABORT
Easy Commit Works
Easy Commit Works

Coordinator

Participant

Participant

Participant

ABORT

GLOBAL ABORT

LET US DISCUSS

Suyash Gupta

EDBT 2018

March 28, 2018
Easy Commit Works

Coordinator

Participant

GLOBAL ABORT

ABORT

LET US ABORT

LET US ABORT

ABORT

Participant

Participant

Participant
Easy Commit Works

System is in a Stable State.
IMPLEMENTATION: ExpoDB

Application Layer / Testbed (YCSB and TPCC benchmarks)

To Global Append Log

Enable/ Disable Secure Transactions

Consensus Protocols: PBFT, Proof of Work, RBFT.

Concurrency Control Protocols: No Wait, Wait-Die, Timestamp.

Execution Threads

Commit Protocols: 2PC, 3PC and EC.

Message / IO Queues

Block Creator

Transaction Manager

Logging

Storage Layer

Indexes

Data

Commit Protocols: 2PC, 3PC and EC.
**Evaluation**

- 64 Standard_D8S_V3 Azure instances, deployed in US East.
  - Each machine has 8 cores and 32GB memory.

- 4 Worker threads attached to a dedicated core.

- Load of 10000 open client connections per node.

- First 60s warmup and next 60s execution.

- Results averaged over three runs.

- NO-WAIT concurrency control algorithm used.

- Two benchmark suites: YCSB and TPC-C.
Varying Skew Factor – YCSB Zipfian Theta

Number of server nodes = 16 and partitions per transaction = 2.

On varying the YCSB skew factor, EasyCommit throughput is equivalent to 2PC.
Varying Server Nodes

Partitions per transaction = 2 and Skew factor = 0.6.

EasyCommit is scalable as 2PC, on increasing the number of server nodes.
EasyCommit is scalable as 2PC, on increasing the number of server nodes.
TPC-C benchmarking for 16 servers, 128 warehouses per server and 3 CC algorithms: WDIE (WAIT-DIE) and TST (TIMESTAMP).

EasyCommit design is orthogonal to underlying Concurrency Control.
We present novel commit protocol – **Easy Commit**.

Leverages best of twin worlds (2PC and 3PC).

Two key observations:
- First transmit and then commit.
- Message Redundancy.

Easy Commit guarantees both **safety** and **liveness**.
We present novel commit protocol – **Easy Commit**.

Leverages best of twin worlds (2PC and 3PC).

Two key observations:
- First transmit and then commit.
- Message Redundancy.

Easy Commit guarantees both safety and liveness.