ECS 165A
Milestone 3
Team Waifus Forever
Our Team

Caroline Yau  
Team Coordinator, Developer

Alvin Ho  
System Architect, Developer

Peter Lo  
System Architect, Developer

Alejandro Armas  
Manager, System Architect, Developer, Tester

Roberto Lozano  
System Architect, Developer
01
Overview
Design and Solution
Implementation

01 Transaction Semantics Overview

02 QueCC

03 MultiThreading
Bug Fixing

- lazy merge implemented
- restarting database now works
  - we previously could not perform certain queries
- multi-indexing
Transaction Semantics

```
1: BEGIN tran;
2: SELECT * FROM my_table WHERE id > 4760 AND id <= 4780;
3: INSERT INTO my_table
4: VALUES (92106429, 15, 2, 11, 13);
5: COMMIT tran;
```

- set of operations over shared data that transforms the data from one consistent state to another.
Atomicity

if ALL transaction operations successful:
database is transitioned into a new consistent state
else:
    NONE is executed and the database remains in the original state.
Consistency

- integrity constraints set by users
Isolation

- We need to avoid conflicting operations when we interleave concurrent transactions.
- CC protocols facilitate coordination among transactions to ensure correct ordering of operations.
Durability

- Achieved maintaining an ordered undo and/or redo actions
- Necessary for rolling back aborted transactions when dealing with weak isolation
Queue Oriented Control Free Concurrency

Goal: Abandon complex concurrency:

- Hardware trends point to opportunities in leveraging parallelism
  - more contention
- simply execute transactions serially on disjoint partitions of data
  - H-Store introduced this idea
- Exploit determinism through planning
- Deterministic schemes eliminate all execution induced aborts
  - e.g. deadlocks

QueCC: A Queue-oriented, Control-free Concurrency Architecture
Thamir M. Qadah, Mohammad Sadogh, 2018

H-store: A high performance, distributed main memory transaction processing system

Calvin: Fast distributed transactions for partitioned database systems
QueCC
Planning Stage

QueCC

transactions

planning thread 1

... or

planning thread n
Execution Stage

QueCC

Priority Level 3

Priority Level 2

Priority Level 1

Execution Threads
Concurrent Batch Planning
Code Performance: Score

SCHEMA_STRING 01111
Selecting key: 92107428
base [92107428, 155, 159, 154, 144]
SCHEMA_STRING 01111
Score 1000 / 1000
writing to page 0 at offset: 0
02
Q/A
Questions about various aspects of the project
03 Demo
A live demonstration of the code