ECS165A
Milestone 3 Overview

Haley Raizes, Brittany Bates,
Jim McKerney, Nicholas Chen, Chris Bried
Objectives

- Overview of QueCC
- Transaction & Query Ops
- Planner Worker
- Execution Worker
- Intern Worker
- Database UI
- Demo
Milestone 1

The Path to Durability

Milestone 2

The Path to Concurrency

Milestone 3

Volatile Memory

Non-Volatile Memory

Database

Tables

Page Range

Root

Table 1

Table 2

Table 3
Overview of QueCC

- Deterministic
- No aborts
- No locks
Transaction & Query Operations

- Queries return query objects
- Query objects are wrapped in transactions
- Add transactions to the transaction queue to be planned and executed at a later time
Execution Worker

High Priority Group Queue Index 1
- insert(1, 4, 2, 3)
- insert(11, 12, 13, 14)
- Select(6, 7, 8, 9)

Low Priority Group Queue Index 1
- Select(1, 4, 2, 3)
- Select(11, 12, 13, 14)

High Priority Group Queue Index 2
- Select(6, 7, 8, 9)

Low Priority Group Queue Index 2
- Select(2, 3, 4, 5)
- Select(7, 8, 9, 10)

High Priority Group Queue Index 3
- select(3, 4, 5, 6)
- select(8, 9, 10, 11)

Low Priority Group Queue Index 3
- select(8, 9, 10, 11)
- select(9, 10, 11, 12)

High Priority Group Queue Index 4
- select(4, 5, 6, 7)

Low Priority Group Queue Index 4
- select(4, 5, 6, 7)
- select(4, 5, 6, 7)

Execution Worker 1
Assigned Index 1
- H1 Select(6, 7, 8, 9)
  key % queue_count == Queue Index 1

Execution Worker 2
Assigned Index 2
- H1 Select(7, 8, 9, 10)
  key % queue_count == Queue Index 2

Execution Worker 3
Assigned Index 3
- H1 Select(8, 9, 10, 11)
  key % queue_count == Queue Index 3

Execution Worker 4
Assigned Index 4
- H1 Select(4, 5, 6, 7)
  key % queue_count == Queue Index 4
Intern Worker

*Batches transactions

Transaction Queue
+1 +2 +3 +4 +5

Batch of Transactions
Transaction 1
Transaction 2
Transaction 3
Transaction 4
Transaction 5

Execution Threads

Transaction Meta Data

<table>
<thead>
<tr>
<th>query</th>
<th>count</th>
<th>ret_list</th>
<th>is_done</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
<td>5</td>
<td>True, True, True, True</td>
<td>False</td>
</tr>
<tr>
<td>+2</td>
<td>5</td>
<td>True, True, True, True</td>
<td>False</td>
</tr>
<tr>
<td>+3</td>
<td>4</td>
<td>True, True, True</td>
<td>True</td>
</tr>
<tr>
<td>+4</td>
<td>3</td>
<td>[3, 4, 5]</td>
<td>False</td>
</tr>
<tr>
<td>+5</td>
<td>2</td>
<td>[0, 1, 2], [0, 2, 3]</td>
<td>True</td>
</tr>
</tbody>
</table>
Database UI

Front end interface running React.js

- Accessed through a browser

Front end interface talks to a backend running Flask

- Flask is a Python web framework, so it can talk to our database
Demo