COMP 512 Project

Group 9:
Luke Emery-Fertitta
Jonathan Campbell

System Architecture

- Middleware between clients & servers.
- Lock Manager / Transaction Manager on middleware
- 2PC with fault tolerance

Architecture: Middleware

- Uses Web Services.
- Implements ResourceManager.
- Delegates data operations to respective RM servers.
- Uses WSClient class for communication with RMs.
- Handles all customer-related operations, talking to RMs if needed (e.g. reserveItinerary).

Architecture: Lock Manager

- Strict two-phase locking
- Conversion of shared locks

Architecture: Txn. Manager

- Table maintains transactions and their IDs
  - Entry added with start(), deleted with abort/commit.
- List of undo operations maintained per transaction.
  - Call to write preceded by indication of undo op (lambda function).
  - Undo ops execute in reverse order on call to abort().
- Operations request read/write locks on data items.
- Abort called by client, or after deadlock, TTL timeout, or vote failure.

Architecture: 2PC

- RMs maintain two copies of data
- Txn. manager coordinates sending of vote requests and decisions.
- Crash point checking.
- Recovery.

Problems encountered

- Middleware knowledge of RM operations
- Synchronization of start() and shutdown()
- Issues with recovery

Tests

- Middleware: JUnit integration tests
  - Sequence of commands passed to client.
  - Compares client output with expected response.
- Lock Manager
- Transaction Manager
- 2PC

Performance Evaluation

- Graph of TPS/avg. time.
- Single and multi-client.
- 2 transaction types.