Assignment 4: Performance Evaluation Report

Team Members: Tanzim Mokammel (Team Leader) (996155208)
Wina Ng (996066821)
Xiangkun Li (995894340)
Project Manager: Robert Price
Date: April 14, 2009

Introduction

This report outlines the concurrency implementation, its testing conditions, results and analysis. On average, changing the number of clients do not affect the GET calls and SET calls significantly. However, the number of transaction abortions increase linearly by increasing the number of clients. Therefore, we advise using the server with this implementation for retrieving data with individual input (one call at a time, rather than using a program to loop through function calls and display a large number of records).

Methodology

The following sections describe the concurrency implementation and the test conditions to evaluate the server's performance with respect to multiple client connections.

The team chose to use select system calls for concurrent client applications. It accepts commands from various clients simultaneously, and multiplexes them to run one command at a time.

Metrics

There are two categories that group the testing conditions:

- End-to-end tests
- Transaction abort rate tests

End-to-End Time Tests

These variables apply for accessing records in random order:

- Cache size: 100
  - Cache policy: ON/OFF
- Random GET, SET and query calls
  - Number of calls for each GET and SET: 1,2,50,100,150,200,300
  - Query calls: find records with ranks 10 to 100, 120 to 300

Transaction Abort Rate Tests

This test varies based on the number of clients that perform get/set pairs to observe the rate of transaction abortions.

- Client 1 continuously performs set within the entire table with transaction abortion disabled
- Clients 2-6 perform get/set pairs within 10 random records on the table
- There is a random delay between each get and set pair for clients 2 to 6 to increase the number of transaction aborts

- Cache is on with a cachesize for 40
- Each test ran for 30 seconds

Workload

We use the given workload (with 676 entries) containing 2006 Canadian Census from Statistics Canada as provided in the Assignment 3 Handout. Here are the workload operations:

- Call GET and SET using random keys
- Query calls do not involve the cache, therefore it will not be tested.

Results

The numerical results are organized below according to the testing categories and conditions.

End-to-End Tests using Cache OFF

End-to-End Time Tests with 1 Client

Operation calls	Cache Size	Average Time Taken (seconds)
300 GETs	0	15.491998
300 SETs	0	20.809450
QUERY calls for "Rank" equal to 120 to 300	0	6.447291

End-to-End Time Tests with 2 Clients

Operation calls	Cache Size	Average Time Taken (seconds)
300 GETs	0	17.38377
300 SETs	0	36.56312
QUERY calls for "Rank" equal to 120 to 300	0	12.71349

End-to-End Time Tests with 3 Clients

Operation calls	Cache Size	Average Time Taken (seconds)
300 GETs	0	33.36075
300 SETs	0	36.57646
QUERY calls for "Rank" equal to 120 to 300	0	13.55764

End-to-End Time Tests with 4 Clients

Operation calls	Cache Size	Average Time Taken (seconds)
300 GETs	0	48.01062

300 SETs	0	34.2237
QUERY calls for "Rank" equal to 120 to 300	0	15.32416

End-to-End Time Tests with 5 Clients

Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	0	64.28129
300 SETs	0	40.34032
QUERY calls for "Rank" equal to 120 to 300	0	19.78945

End-to-End Tests using Cache ON

End-to-End Time Tests with 1 Client

Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	100	15.663990
300 SETs	100	19.732015
QUERY calls for "Rank" equal to 120 to 300	100	6.460666

End-to-End Time Tests with 2 Clients

Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	100	16.94999
300 SETs	100	39.64965
QUERY calls for "Rank" equal to 120 to 300	100	12.78832

df

End-to-End Time Tests with 3 Clients

Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	100	23.49316
300 SETs	100	39.3249
QUERY calls for "Rank" equal to 120 to 300	100	14.71034

End-to-End Time Tests with 4 Clients

Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	100	52.93173

300 SETs	100	41.13303
QUERY calls for "Rank" equal to 120 to 300	100	17.89585

End-to-End Time Tests with 5 Clients

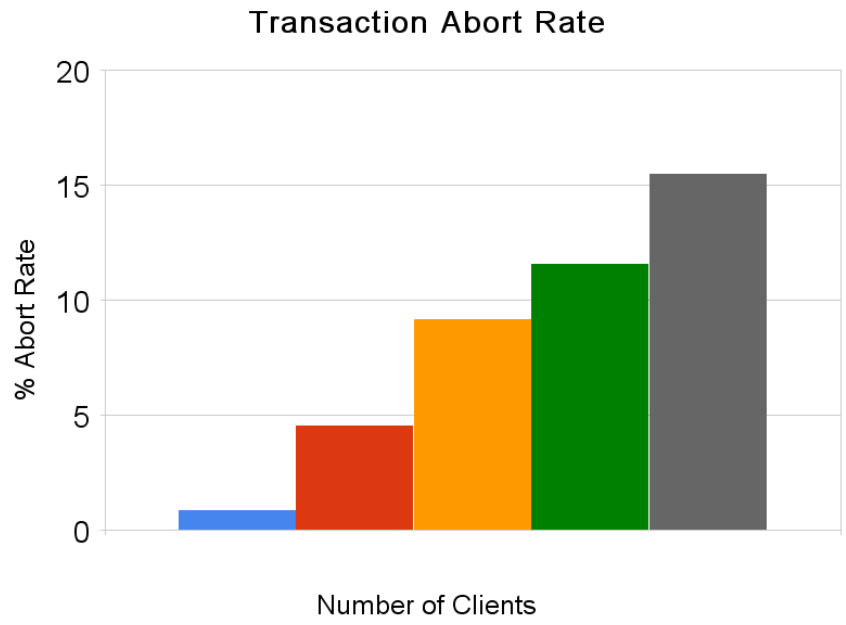
Operation calls	**Cache Size**	**Average Time Taken (seconds)**
300 GETs	100	62.404
300 SETs	100	40.06968
QUERY calls for "Rank" equal to 120 to 300	100	18.3703

Transaction Abort Rate Tests

# of simultaneous clients	Average Abort Rate
2	0.86%
3	4.51%
4	9.16%

5	11.58%
6	15.46%

Note: Client #1 continuously call SET operations.



Analysis

End-to-End Time Tests

The SET calls seem to have slightly more delay than the GET calls as the number of clients increases.

Transaction Abort Rate Tests

The average abortion rate increases linearly (on average) with increasing the number of clients connected to the server.

Based on the results, when the number of connections increases to three or four clients, different operations are affected differently on changing input parameters. The amount of time delays increase between 16-30 seconds for both GET and SET calls. However, the end-to-end time tests seem to affect the query calls only logarithmically. The GET calls appear to have similar results, therefore we advise this application of the server for displaying and searching information more than modifying data, but not for a very busy working environment with a small number of clients.