

Alexandria University - Faculty of Engineering
Electrical Engineering Department

Summer Research Project - 2015

Weekly Report Num. 4

Supervision:

Dr. Bassem Mahmoud Mokhtar - EED

Researcher:

Abanoub Milad Nassief - CSED

Research Fields:

databases, android/web development and machine learning.

Research Goals:

Developing an intelligent application for efficient database management.



Development of a front-end (android/web app) and back-end prototype, and the design of a database to store various data with a large set of attributes.

Capability of applying intelligent data management techniques on the stored data.



Week-3 approach

- Implementing custom search
- Implementing watchlist
- Displaying item, promotion and store details.
- Support shopping by stores.
- User interface modifications.
- Database design modifications to support and optimize smart search techniques.
- Implementing parser to parse user's input for smart search.
- Illustrate system overall interaction.
- Conduct performance study.



Database design modifications


  shop_now_db.key_computer

- # fk_item_id : int(11)
- # keyword : int(11)

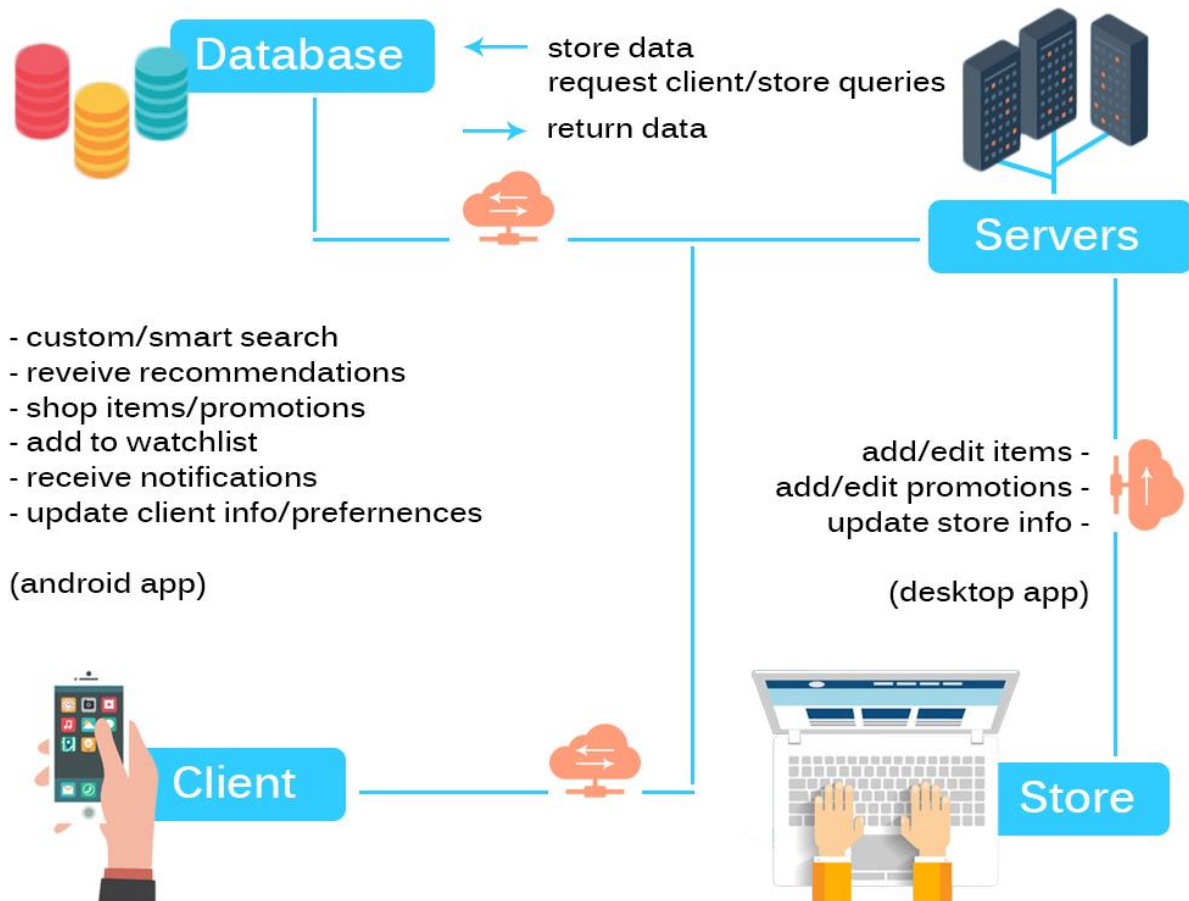
  shop_now_db.items_keys

- # fk_info_id : int(11)
- # keyword : varchar(25)

  shop_now_db.db_info

-  id : tinyint(4) unsigned
- # sub_category_name : varchar(20)
- # category_name : varchar(20)
- # item_tb_name : varchar(20)
- # stock_tb_name : varchar(20)
- # promotion_tb_name : varchar(20)
- # feature_tb_name : varchar(20)
- # key_tb_name : varchar(20)
- # items_count : mediumint(9) unsigned
- # promotions_count : mediumint(9) unsigned

System overall interaction



Performance Study


Tested with

- **1.37 GB RAM**
- **Intel i5 2.4 GHz**
- **Apache Server**
- **MySQL engine**

Database server

- Server: localhost via TCP/IP
- Server type: MySQL
- Server version: 5.6.12-log - MySQL Community Server (GPL)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.4 (Win64) PHP/5.4.12
- Database client version: libmysql - mysqlnd 5.0.10 - 20111026 - \$Id: e707c415db32080b3752b232487a435ee0372157 \$
- PHP extension: mysqli 

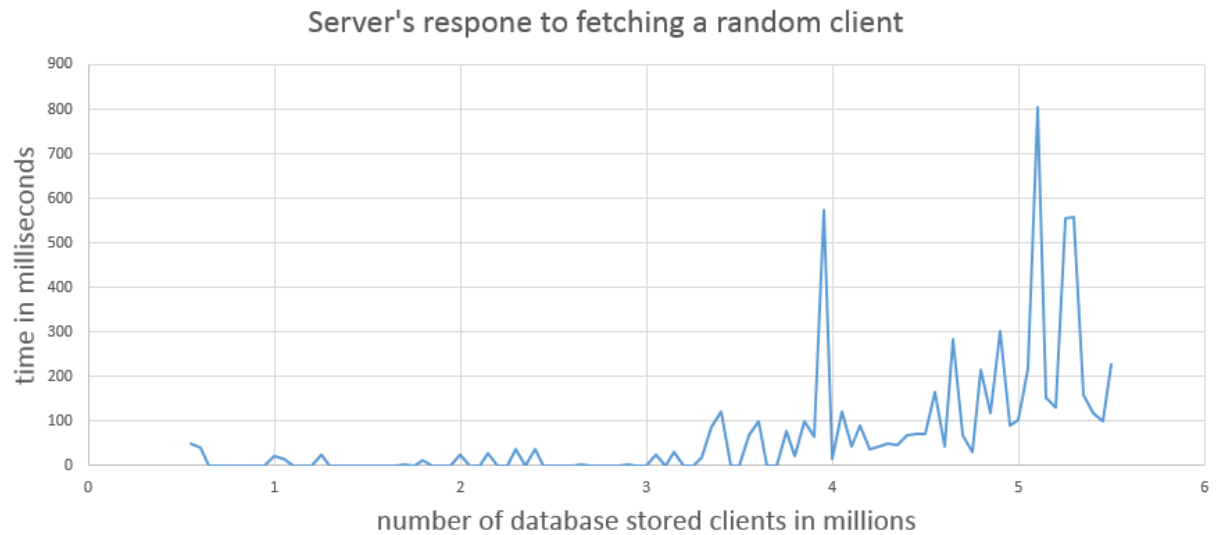


Figure 1

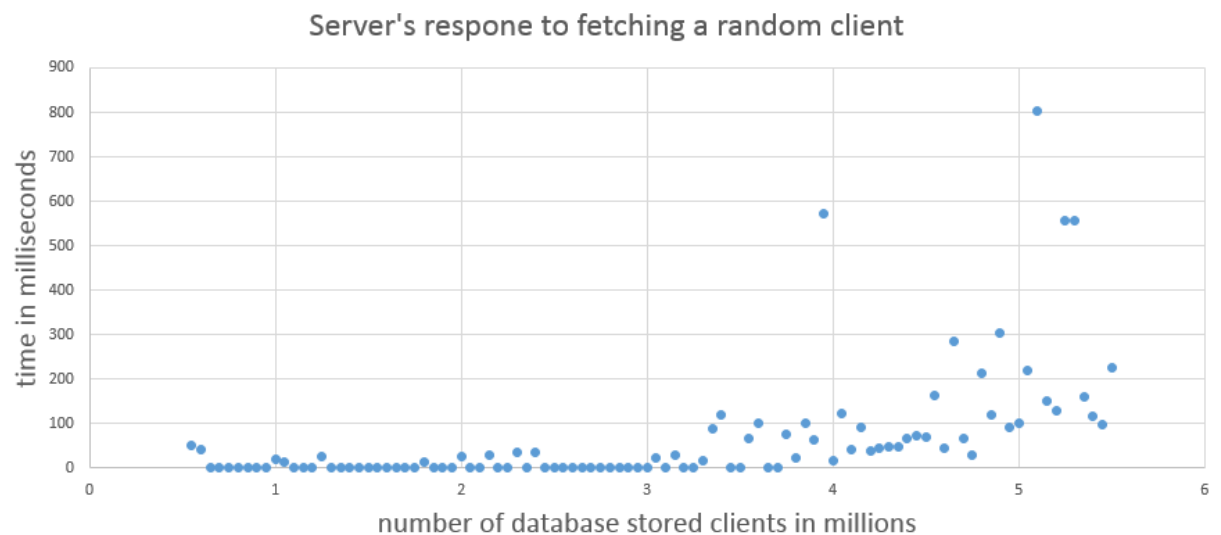


Figure 2

Figure 1 and 2 illustrate server's response time to fetching a client chosen at random from clients table versus the total number of stored clients. The total number of clients ranges 0.5 - 5.5 millions. Server's response tends to cluster around 0 milliseconds with 3 million clients or less. Reaching 3 millions, response clusters around 100 milliseconds. Reaching 4.5 millions, response clusters around 200 milliseconds.

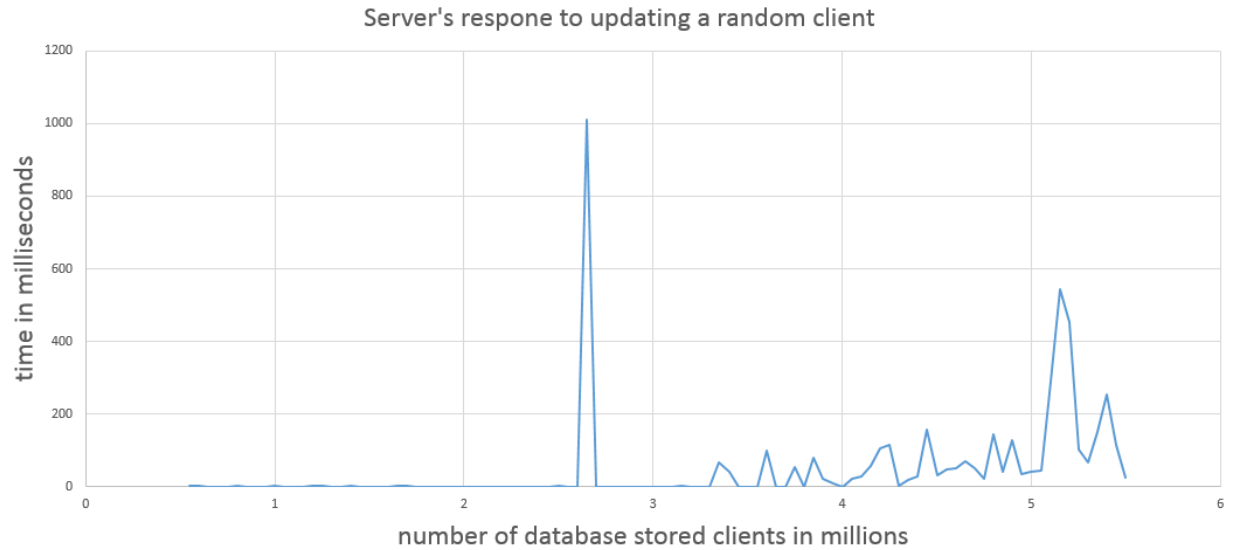


Figure 3

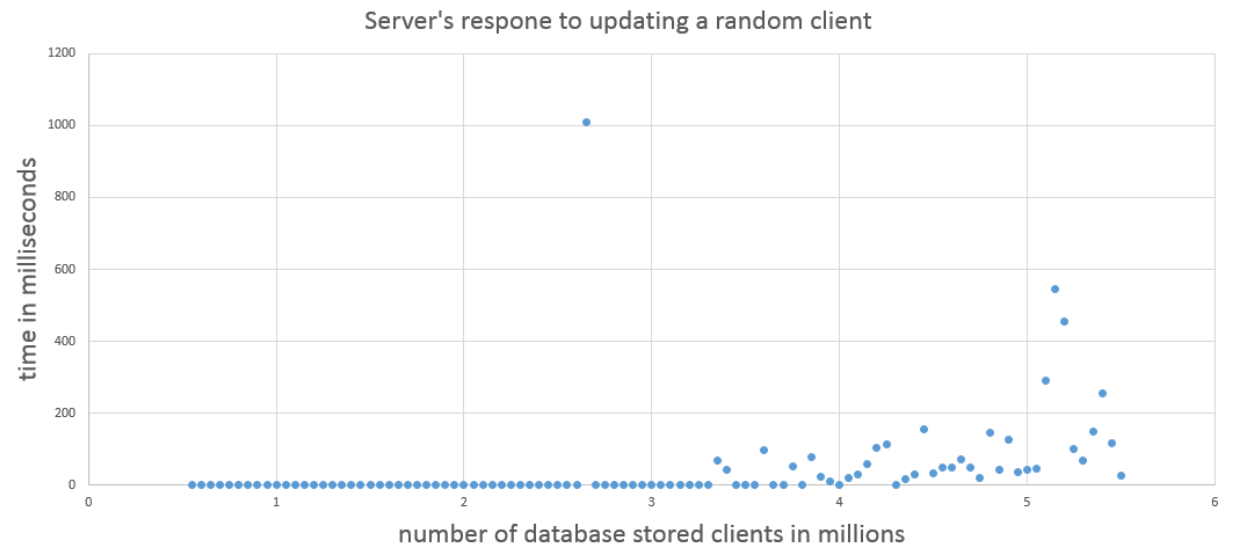


Figure 4

Figure 3 and 4 illustrate server's response time to updating client information chosen at random from clients table versus the total number of stored clients. The total number of clients ranges 0.5 – 5.5 millions. Server's response tends to cluster around 0 milliseconds with 3 million clients or less. Reaching 3 millions, response clusters around 100 milliseconds.

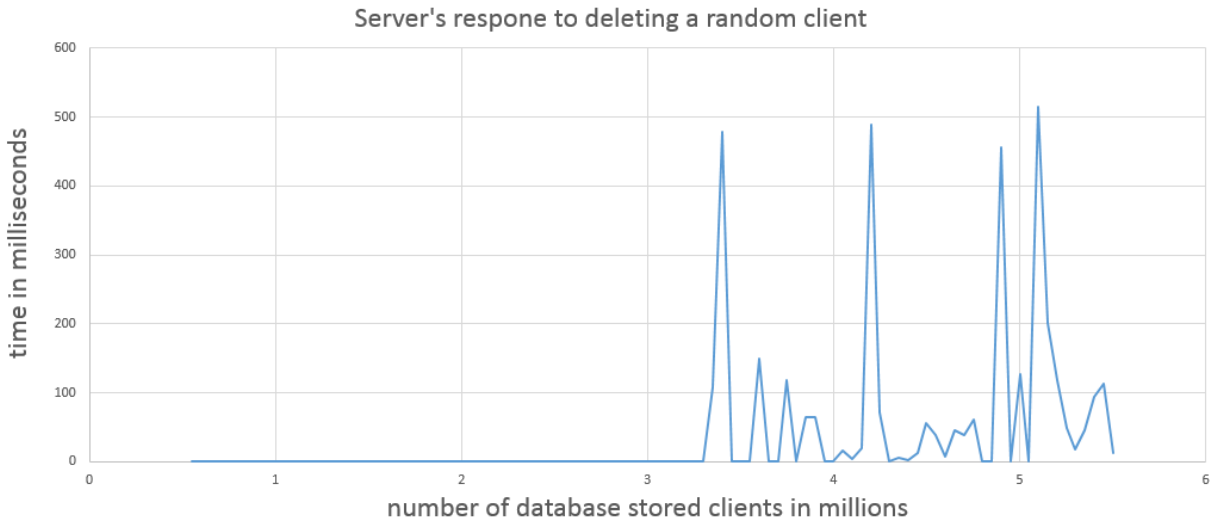


Figure 5

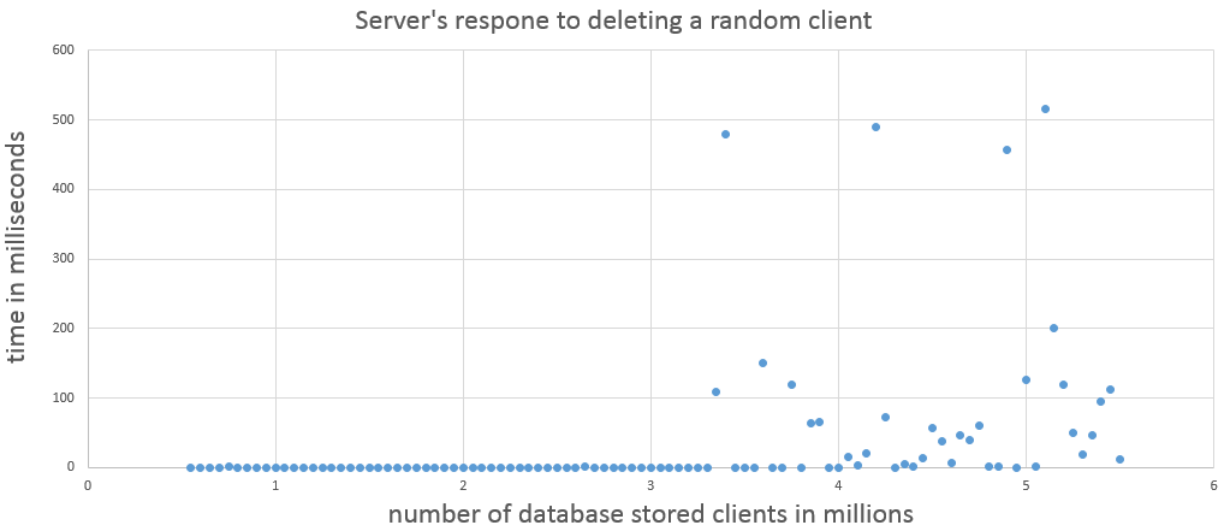


Figure 6

Figure 5 and 6 illustrate server's response time to deleting a client chosen at random from clients table versus the total number of stored clients.

The total number of clients ranges 0.5 – 5.5 millions. Server's response tends to cluster around 0 milliseconds with 3 million clients or less.

Reaching 3 millions, response clusters around 100 milliseconds.

Deleting a client requires more time than fetching as deletion affects the primary key and thus affects table indexing.

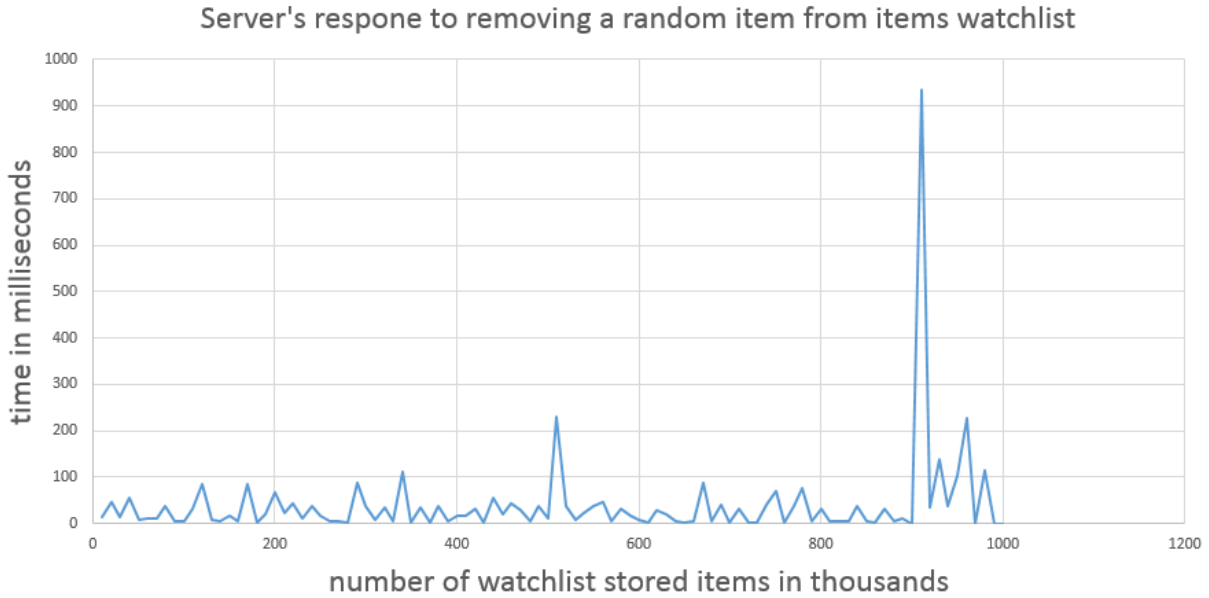


Figure 7

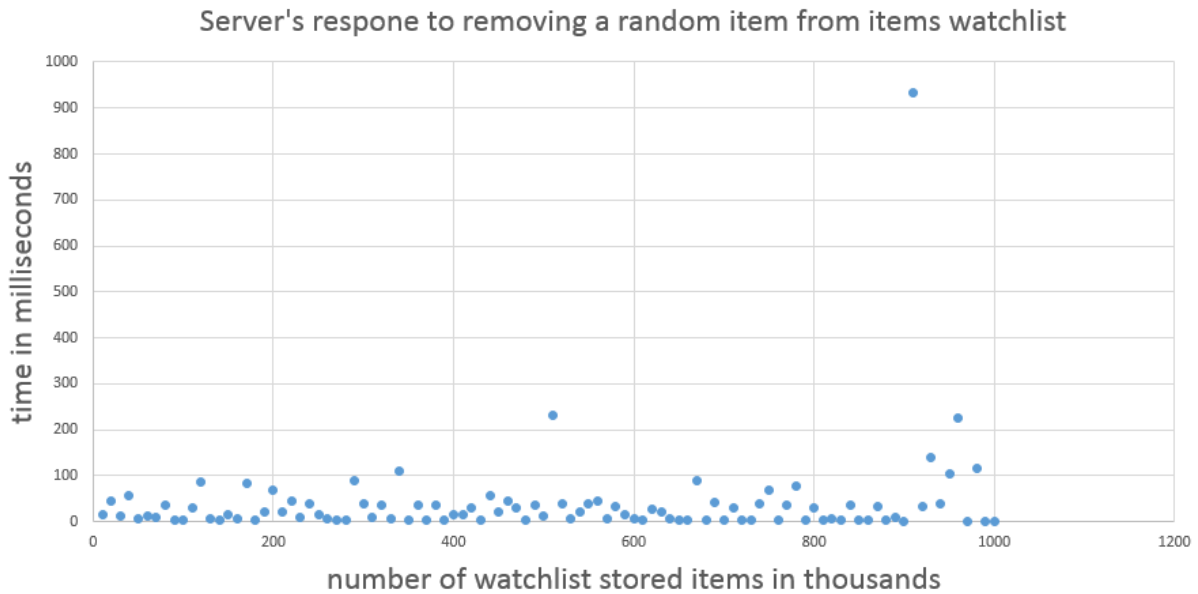


Figure 8

Figure 7 and 8 illustrate server's response time to removing an item chosen at random from items watch list table versus the total number of watch list items. The total number of watch list items ranges 0 – 1 million. Server's response tends to cluster around 50 milliseconds.

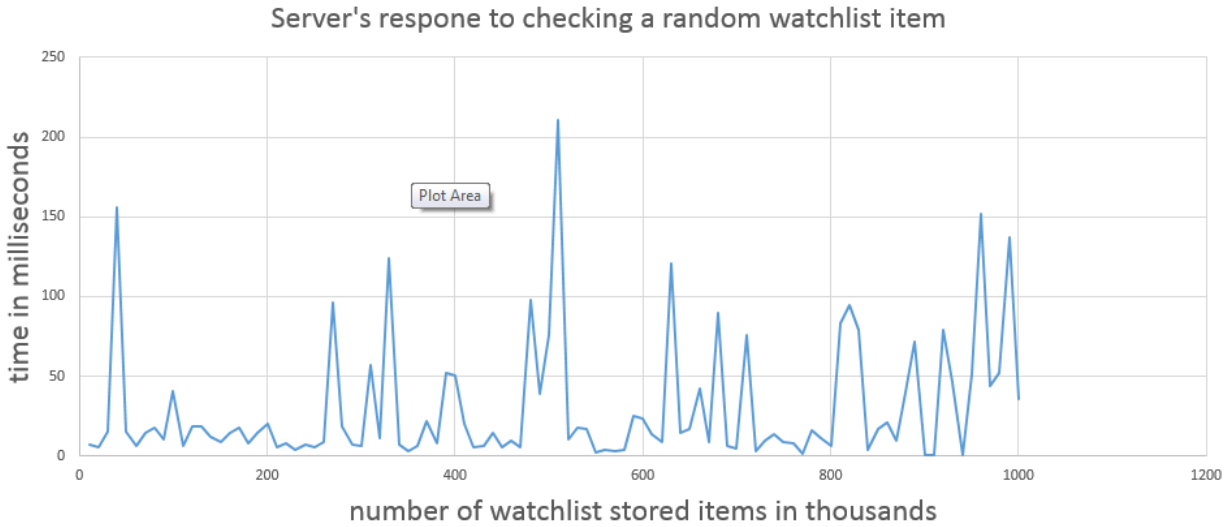


Figure 9

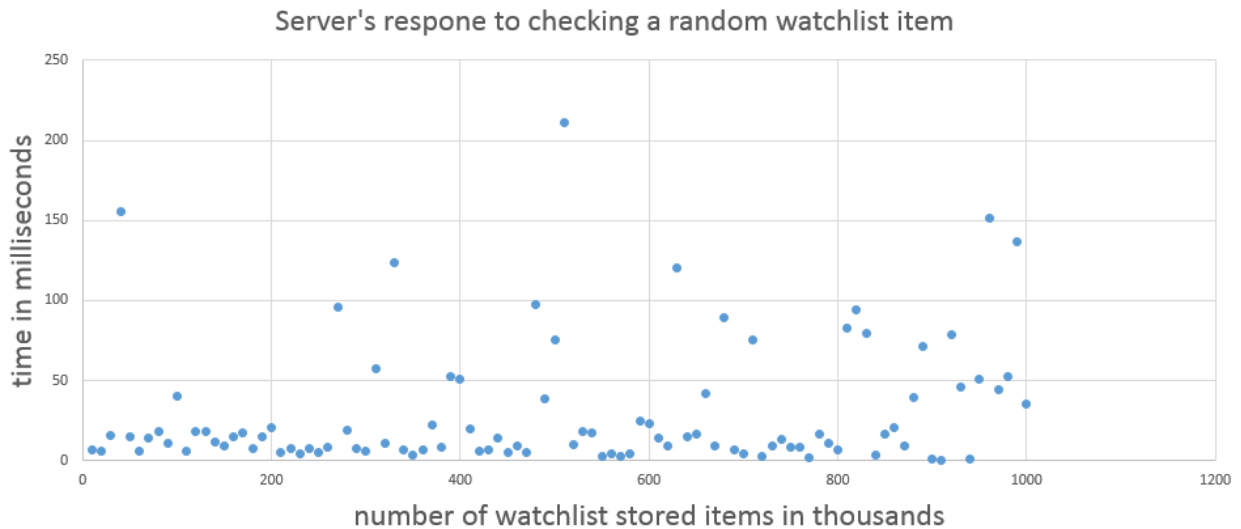


Figure 10

Figure 9 and 10 illustrate server's response time to checking the status of an item chosen at random from watch list table versus the total number of watch list items. The total number of watch list items ranges 0 – 1 million. Server's response tends to cluster under 25 milliseconds and around 75 milliseconds with some glitches in between.



Figure 11

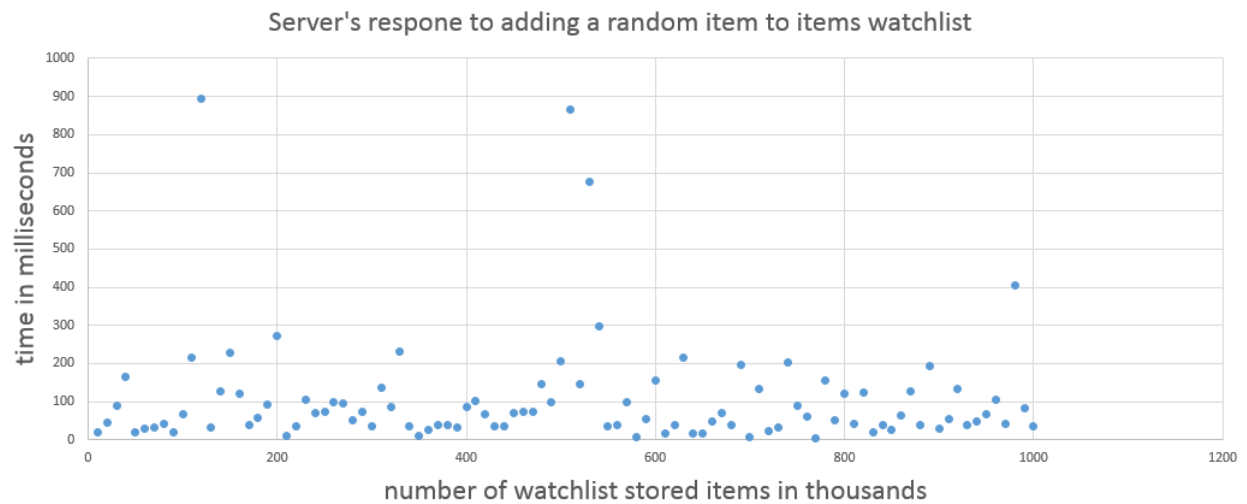


Figure 12

Figure 11 and 12 illustrate server's response time to adding a random item to watch list items table versus the total number of watch list items. The total number of existing watch list items ranges 0 – 1 million. Server's response tends to cluster around 100 milliseconds.

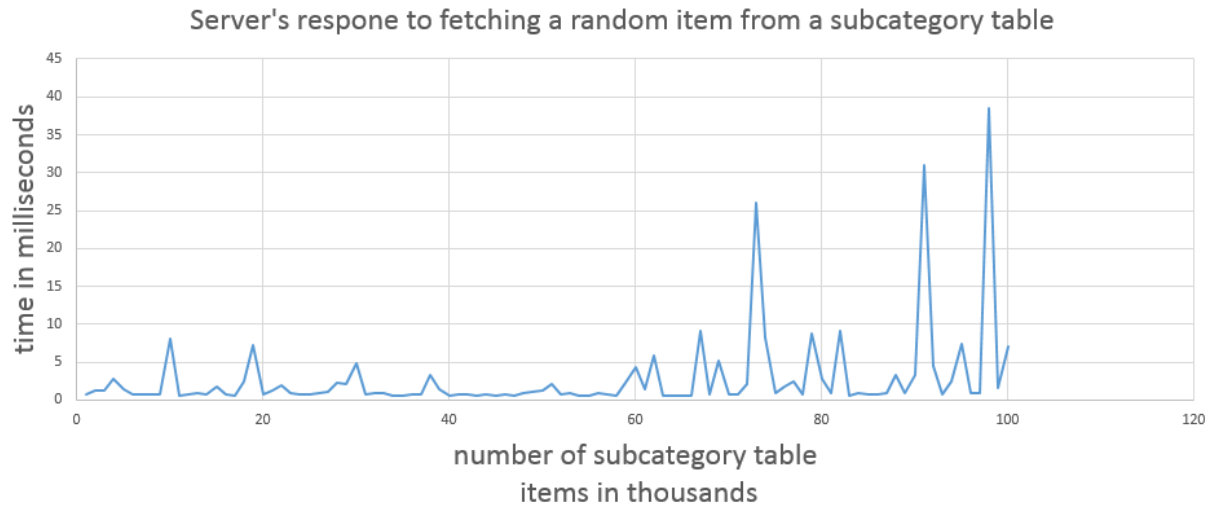


Figure 13

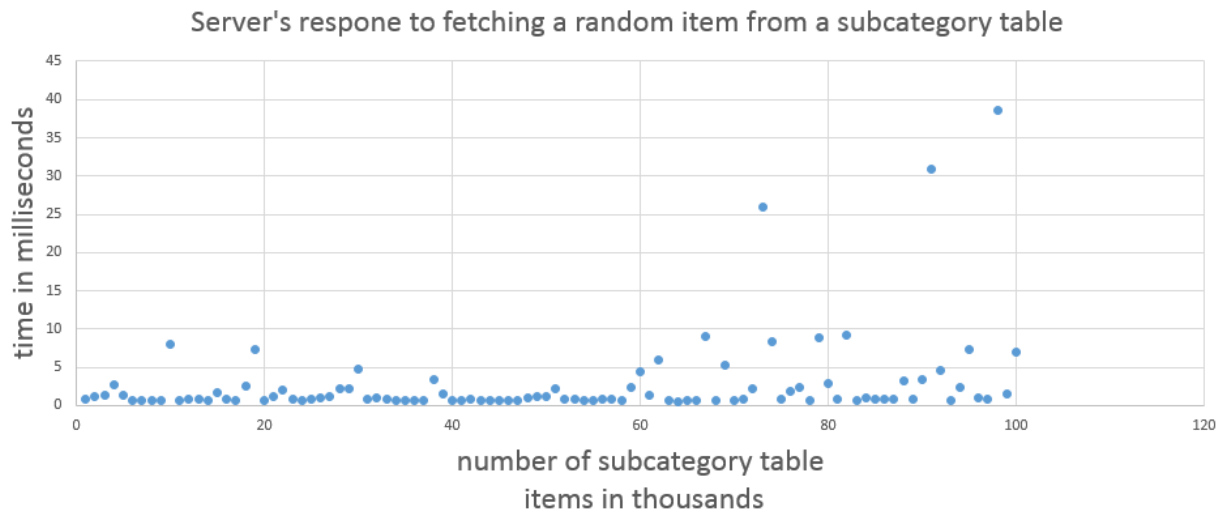


Figure 14

Figure 13 and 14 illustrate server's response time to fetching an item chosen at random from items table versus the total number of stored items in this subcategory table.

The total number of items in this subcategory ranges 0 – 100k. Server's response tends to cluster around 2.5 milliseconds with 60k items or less. Reaching 60k, response clusters around 5 milliseconds.

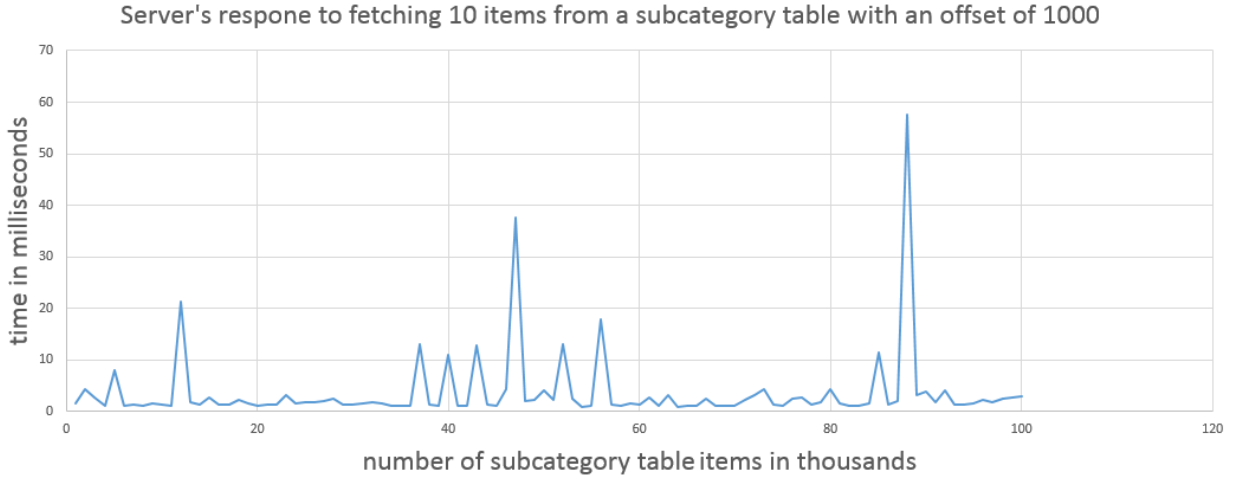


Figure 15

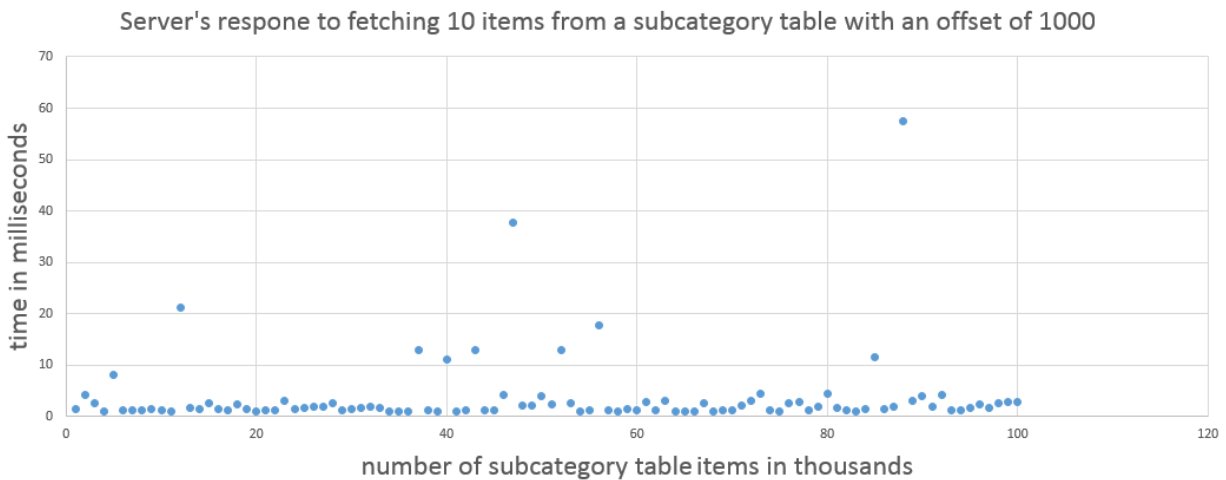


Figure 16

Figure 15 and 16 illustrate server's response time to fetching 10 items chosen from items table after skipping 1000 items (**simulating the situation where a client has viewed 1000 items and asks for 10 items more by pagination**) versus the total number of stored items in this subcategory table.

The total number of items in this subcategory ranges 0 – 100k. Server's response tends to cluster under 5 milliseconds with some glitches in between.

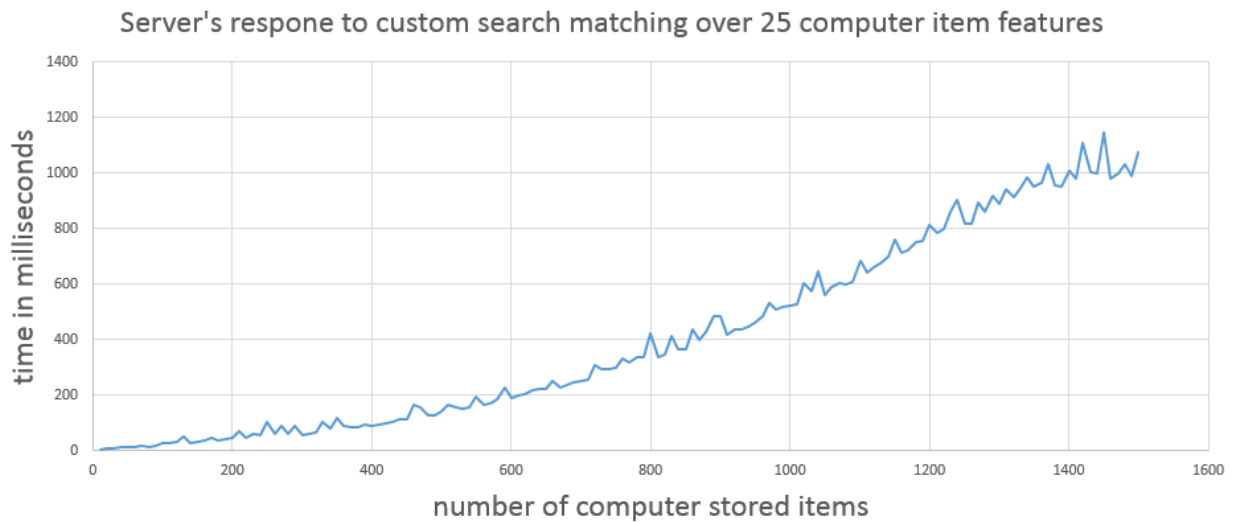


Figure 17

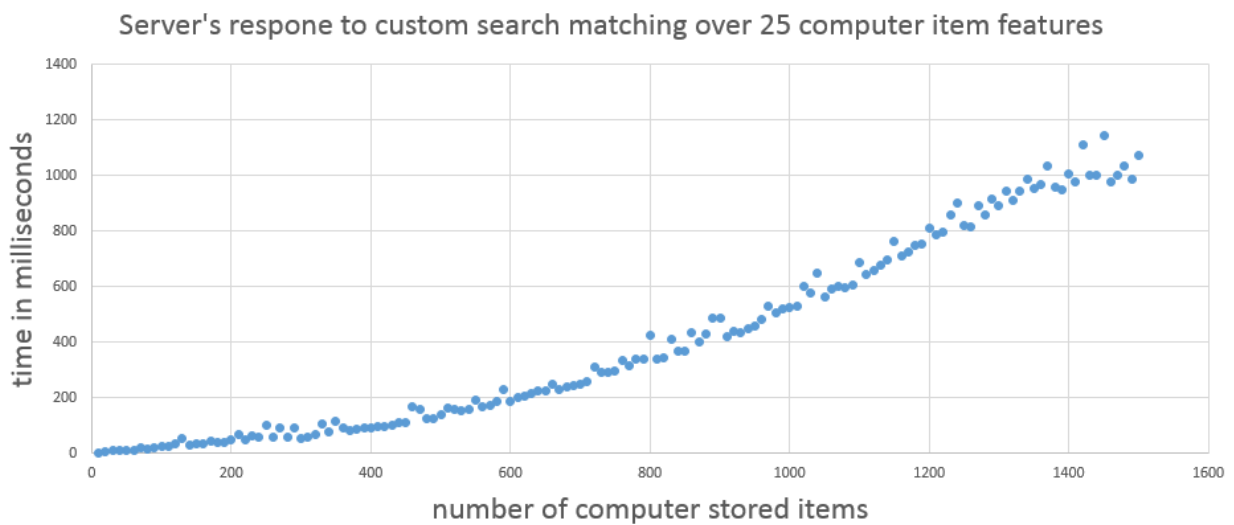


Figure 18

Figure 17 and 18 illustrate server's response time to match required features (25 features) with items table versus the total number of stored items in this subcategory table.

The total number of items in this subcategory ranges 0 – 1600. Server's response increases exponentially.

Performance Notes

- **Table Indexing** by primary and unique keys improves the operations of search and retrieval with large scale tables but slightly affects the operation of deletion as deleting keys changes requires more time to update and balance the indexing files used by MySQL.
- **Custom Search** response time decreases as the number of items increases. Improvements are required to enhance the custom search performance. We can decrease the number of features used in search matching and keep track of the user's selected features only not all features.