

Technology Proposal: MySql Connector MotoMoto

The New Panelists
12/15/2021

Blake Del Rey
Isabel Guzman
Jacob Sunia
James Austin Jr.
Naeun Yu

Team Leader: James Austin Jr.

1. MySqlConnection - Version 0.44.0	3
1.1 Overview	3
1.1.1 Metric Scale	3
1.2 Purpose	3
1.3 Competition Analysis	3
1.4 Conclusion	5

1. MySqlConnection - Version 0.44.0

1.1 Overview

MySqlConnection is an open-source software used to retrieve, view, and insert information into our datastores. With the use of MySqlConnection, we will be able to process asynchronous queries with low memory consumption.

1.1.1 Metric Scale

The metrics are scored between 0 and 1 according to how well the competitor aligns with our purpose stated above.

0	Poor
0.25	Some value
0.5	Average among competitors
0.75	Good but has disadvantages
1	Best among all competitors, no disadvantages

1.2 Purpose

We would like to request this technology for use in the development and maintenance of MotoMoto because it will allow our development team to automate queries for every service that we offer. By using this technology we will be able to ensure that we can query information to our users at a faster pace than the approved technology, ODBC. We will deploy this API to our software to impact the speed of information retrieval.

1.3 Competition Analysis

The decision to request MySql connector as our database connector was made after we compared various different APIs to discover what was most comfortable with our team and most useful for the scope of backend development. The two technologies that were tested after consideration were ODBC 17.8.1.1 and MySql Connector 0.44.0:

The decision to include only two technologies is because other competitors are either significantly non-competitive versus MySql Connector (being under the umbrella of ADO.NET) or are paid services which are completely out of the spectrum for MotoMoto's budget plan.

Metrics:

Load Performance:

We judged read performance based on analyzing information from studies and tests of queries.

- Overall, we can conclude that the performance is solely based off of the number of queries that are being computed.
- ODBC performed faster with a lower number of queries to load to the data store as it took roughly 4 seconds to query 250,000 queries and it took MySql Connector 2 seconds to query 250,000 queries
- On the contrary, MySql Connector took 5 seconds to query 2 million queries while ODBC took roughly 25 seconds to query 2 million, exemplifying that the load speed is ultimately much faster.

Write Performance:

With the connection being a slower interface to larger amounts of data, ODBC continued to struggle in write performance compared to MySql Connector based on the latest ODBC driver 17.8.1.1.

- Overall, we can conclude that the performance is solely based on the number of queries that are being computed.
- ODBC performed faster with a lower number of queries to write to the data store as it took roughly 2 seconds to query 250,000 queries and it took MySqlConnection 3 seconds to query 250,000 queries but MySqlConnection took 5 seconds to query 2 million queries while ODBC took roughly 25 seconds to query the 2 million.

Connection Performance:

As a team we judged connection performance based on analyzing information from studies and tests of queries.

- ODBC takes a longer time to load than third party APIs because ODBC is based on a layer over native Microsoft .NET DBMS APIs. Through research we gathered, it takes around 56 seconds for ODBC to connect to 2.5 million records which is very slow compared to MySql Connector which can query the same amount in roughly 10 seconds.
- MySql Connector can retrieve roughly 10 million requests in 37 seconds which also is much faster than ODBC.

Analysis Table:

	Multiplier	ODBC	MySQL Connector
Load Performance (>250,000 queries)	1.2	0.25	1.0
Load Performance (<250,000 queries)	1.15	0.25	0.5
Write Performance (>250,000 queries)	1.1	0.25	0.75
Write Performance (<250,000 queries)	1.05	0.75	0.5
Connection Performance	1	0.0	1.0
Score		2.76	4.125

1.4 Conclusion

Overall, MySQL Connector was much faster in all categories than ODBC which is the reason why we want to implement a better data access API. Since our team does not have any plans of using OR/MS, having an optimized data API will increase in speed which will be more efficient for storing user information. Since our application will be focused on Load Speed and the performance of the information that a user can retrieve at a time will be important to the success of our application. Clearly ODBC is a lot slower than MySQL Connector as shown above in the metrics explanation and was most taken into consideration when selecting a connector for our datastore.