European Journal of Research Development and Sustainability (EJRDS)
Available Online at: https://www.scholarzest.com
Vol. 3 No. 8, August 2022
ISSN: 2660-5570

FAST WAY TO RETRIEVE DATA FROM SQL DATABASE USING DAPPER COMPARE TO LINQ ENTITY FRAMEWORK

SARMAD HAMZA ALI [1], GESOON J.K.AL-ABBAS [2] AND ALAA ABD ALI HADI [3]
[1]Al-Muthanna University, Iraq
sarmad@mu.edu.iq
[2]Al-Muthanna University, Iraq
ghusoonjawad@mu.edu.iq
[3] AL-Furat AL-Awsat technical university , Iraq
alaa@atu.edu.iq

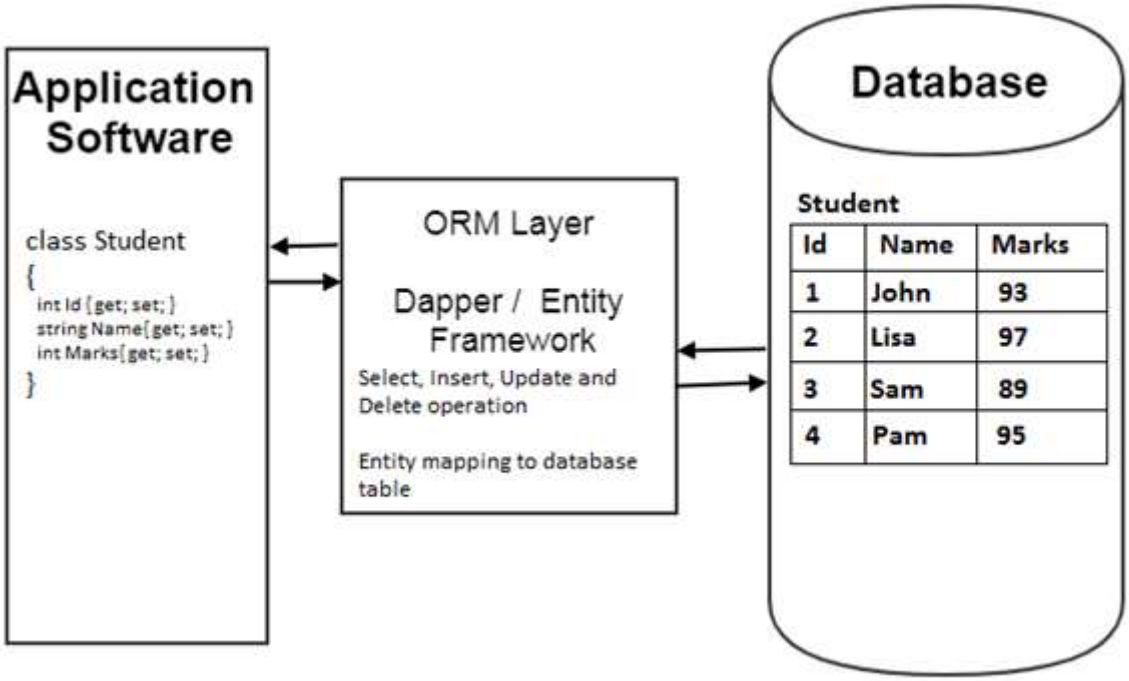
Article history:
Received: 26th May 2022
Accepted: 26th June 2022
Published: 4th August 2022

Abstract:
In this research we did a comparison between using Dapper and LINQ to access Databases, the speed of Dapper is growing, which makes us think why such a simple object mapper framework that used to retrieve data and map it to a class takes that much of attention over time, in the very next line is shown in number how Dapper is faster a saver than any other DB connections ways.

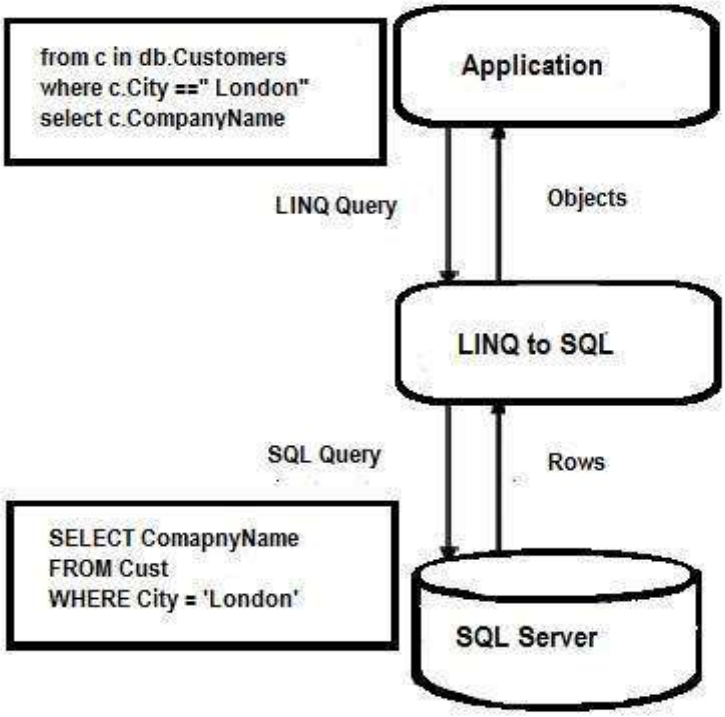
Keywords: Dapper, Data , Sql Server, Retrieve Date, Entity Framework , LINQ

INTRODUCTION

A Data Mapper is an object- relational mapping that maps data from a SQL Database to Programming language, Dapper is considered to be the fastest way to get data , it is as fast as using raw ADO.NET
On the other hand



LINQ to SQL is an ingredient of .NET Framework ver. (3.5) that provides a run-time infrastructure for handling a



relational data as objects.

EXTRACTING DATA

To test Dapper speed for extracting data from Database, four Classes must be built first, C# is the programming Language of choice, Dapper falls into a family of tools known as *micro-ORMs*. These tools perform only a subset of the functionality of full-blown Object Relations Mappers, such as **Entity Framework Core**. Features vary by product

	Micro ORM	ORM
Map queries to objects	✔	✔
Caching results	✘	✔
Change tracking	✘[1]	✔
SQL generation	✘[2]	✔
Identity management	✘	✔
Association management	✘	✔
Lazy loading	✘	✔
Unit of work support	✘	✔
Database migrations	✘	✔

.

DATA MODEL

[sourcecode language="plain"]
public class Students
{
public long Id { get; set;}
public string FirstName { get; set; }
public string LastName { get; set; }
public string College { get; set; }
}
[/sourcecode]
Student must assigned to a College, which leads to the next class.
[sourcecode language="plain"]
public class College

```
{
public long Id { get; set; }
public string CollegeName { get; set; }
}
[/sourcecode]
```

A Student can assigned to multiple groups .

```
[sourcecode language="plain"]
public class Groups
{
public long StudentId { get; set; }
public long CollegeId { get; set; }
}
[/sourcecode]
```

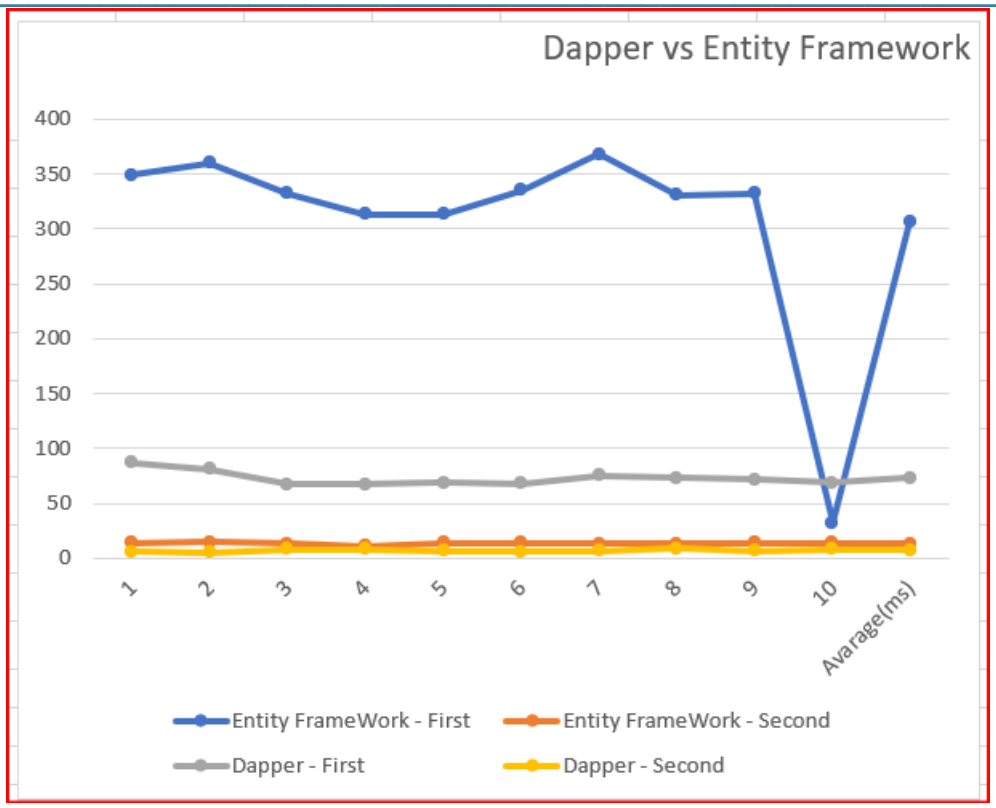
EXAMINING

Out test starts with three colleges , each of which contain 1,000 Item (student) . as the first round of test we found that both Entity Framework and Dapper take longer with the first query than with subsequent queries. For Entity Framework, the query must be complied on the at first of execution, in the other hand, Dapper execute it directly. For all tests, each query executed twice and ran each test 10 times before analyzing the results. All times are in milliseconds.

Loading 1 Students

In the round , a single Student by Id from the database.

Test Rounds	Entity FrameWork - First	Entity FrameWork - Second	Dapper - First	Dapper - Second
1	349	14	87	6
2	360	15	81	5
3	332	13	67	8
4	313	11	67	8
5	313	14	69	7
6	335	14	68	6
7	368	13	75	7
8	331	13	73	9
9	332	14	72	7
10	32	14	69	8
Avarage(ms)	306.5	13.5	72.8	7.1



Clearly the test results shows that Dapper is faster than Entity Framework, especially on the first round of execution.

LOADING MANY STUDENTS BY COLLEGES

In the second round, all the Students have been loaded to see how each framework fared loading multiple records and searching on a string field as alternative to a primary key.

The differences between the first and second round of queries were not severely different .

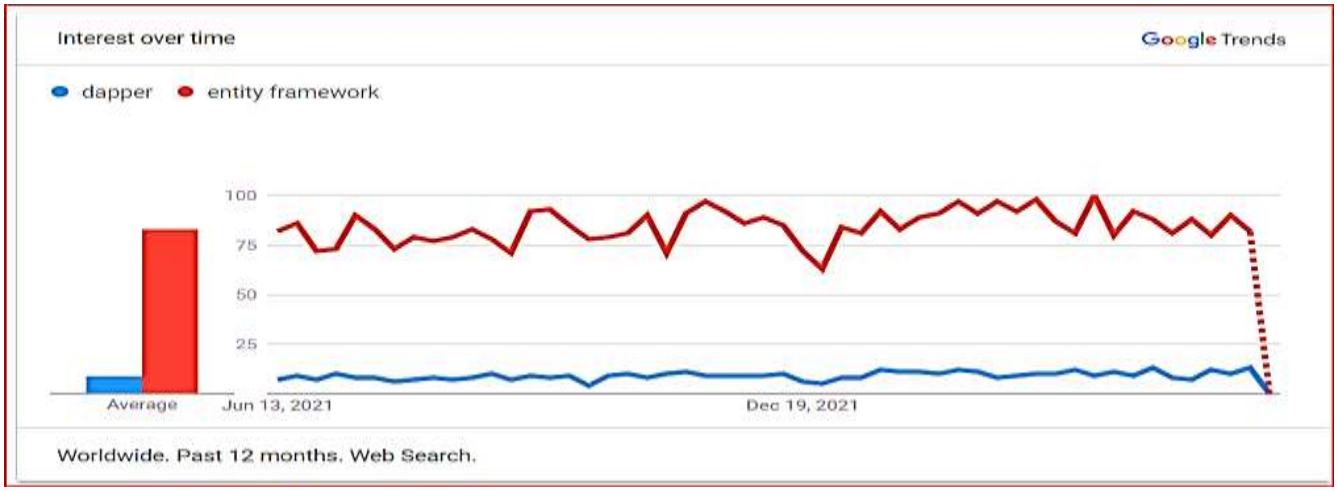
Test Iteration	Entity Framework First	Entity Framework Second	Dapper First	Dapper Second
1	80	35	13	7
2	83	33	19	10
3	81	33	12	8
4	75	33	11	7
5	69	33	11	7
6	79	33	11	7
7	78	38	12	7
8	72	32	12	8
9	76	32	12	8
10	65	34	14	6
Average (ms)	75.8	33.8	12.7	7.5

LOADING GROUPS WITH STUDENTS

Test Iteration	Entity Framework First	Entity Framework Second	Dapper First	Dapper Second
1	143	38	23	16
2	144	42	27	20
3	127	39	21	16
4	132	29	21	26
5	136	38	28	22
6	142	37	24	20
7	132	35	22	16
8	133	39	22	16
9	127	36	24	18
10	121	37	21	21
Average (ms)	133.7	37	23.3	19.1

The results are uniform with the other rounds. Entity Framework is remains slower than Dapper, but if we take a look at average, second Dapper query still 4.2 milliseconds faster.

INTEREST OVER TIME OF DAPPER VS ENTITY FRAMEWORK



CONCLUSION

Dapper Designed to be Faster that the other SQL Server Retrieving Data Methos and the test approved it, a known fact that designers pay attention to every millisecond, in some cases like a small business , saving few millisecond is not matter, we recommend using LINQ integration , but when it come to ORM operation , as we mentioned , Dapper is the solution for that, Dapper will be a very powerful tool to think about before using a raw SQL.

REFERENCES

1- Rosen K. *Elementary Number Theory and Its Applications*. Reading MA: Addision Weley 2010
2- L.E. Dubins and D.Freedman . Machiavelli and Gale-Shapley
3- P.C. Jones . A Polynomial time Market Mechanism . Journal of Information 1983.
4- Wiphusitphunpol, W., & Lertrusdachakul, T. (2017, June). Fetch performance comparison of object relational mapper in. NET platform. In 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON) (pp. 423-426). IEEE.

5- Zmaranda, D., Pop-Fele, L. L., Gyorödi, C., Gyorödi, R., & Pecherle, G. (2020). Performance comparison of crud methods using net object relational mappers: A case study. International Journal of Advanced Computer Science and Applications, 11(1).
6- Nilsson, M. (2022). An evaluation of Language Integrated Queries (LINQ).
7- Sharma, T., Fragkoulis, M., Rizou, S., Bruntink, M., & Spinellis, D. (2018, May). Smelly relations: measuring and understanding database schema quality. In Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice (pp. 55-64).
8- Smith, J. (2021). Entity Framework core in action. Simon and Schuster.
9- Schwichtenberg, H. (2018). Introducing entity framework core. In Modern Data Access With Entity Framework Core (pp. 1-14). Apress, Berkeley, CA.
10- Mace, J., Roelke, R., & Fonseca, R. (2015, October). Pivot tracing: Dynamic causal monitoring for distributed systems. In Proceedings of the 25th Symposium on Operating Systems Principles (pp. 378-393).
11- Naspinski, S. (2011). Selection and Implementation of Technologies for the Re-Engineering of an Existing Software System. University of South Florida.
12- Guo, Z., Zhou, D., Lin, H., Yang, M., Long, F., Deng, C., ... & Zhou, L. (2011). G2: A graph processing system for diagnosing distributed systems. In 2011 USENIX Annual Technical Conference (USENIX ATC 11).