Decision Analysis Report - Object Relational Mapper

Project Name: Tresearch
Application Type: Web Application

Trial By Fire
Jessie Lazo
Matthew Chen (Team Lead)
Pammy Poor
Viet Nguyen

Instructor: Vatanak Vong

Submission Date
12/13/2021

California State University, Long Beach
College of Engineering
CECS491 Section 6 11950, Fall 2021

Purpose

The following is the Decision Analysis and Resolution to analyze which technologies would be the most applicable towards the development of the product.

Technologies

Microsoft Entity Framework Core v.6.0+

i. Open source object-relational mapping framework integrated within the .NET Framework

Dapper v.2.0.123+

ii. Object-relational mapper developed for the Microsoft .NET platform, often referred to as a Micro ORM due to its lightweight nature

NHibernate v.5.0+

iii. Object-relational mapper developed for the Microsoft .NET platform

Preface

In this comparison, Dapper, NHibernate, and Microsoft EntityFramework.Core are being tested to see which Object-Relational Mapper performs better in the given situations. All technologies are being tested on the .NET 6.0 Framework within Visual Studio 2022.

Metrics

ORM Type

a. Compares the capabilities and scale associated with the type of Object-Relational Mapper

Query Type

a. Determines the methods in which the ORM is able to execute queries. SQL denotes that the ORM is capable of accepting direct SQL code, LINQ denotes that the ORM can make use of the language-integrated queries.

Design Capabilities

a. Database first approach requires the set up and connection of a database prior to connection, code first approach is the ability to generate the database, tables, and models through code from the ORM

Start Up Time

a. Denotes how quickly one is able to install the packages and implement into their code so they can begin working with the database, The less configuration needed the better.

Insert Row Speed

a. The duration needed for the ORM to add data to the local database. This metric is important due to the amount of data that our product will be receiving and need to be stored.

Delete Row Speed

a. The duration needed for the ORM to delete the specified amount of rows in the local database. This metric is important as users of the product will be deleting data relatively often.

Benchmark Data

1. For the methods adding and deleting data, a database table “People” with attributes {ID, FirstName, LastName, Company, Title} is used. Random values for the data attributes are generated and then used during the method call. System.Diagnostics.Stopwatch is used to record the duration of each test. Results are the average time of ten runs for each respective method.
   Table. 1 Insert Row Duration

Insert Rows Duration (s)	Dapper	NHibernate	Microsoft EFCore
500 Entries	1.33s	1.84s	2.48s
10000 Entries	3.8s	8.3s	71.94s

Table. 2 Delete Row Duration

Delete Rows Duration (s)	Dapper	NHibernate	Microsoft EFCore
500 Entries	1.19s	1.9s	2.12s
10000 Entries	1.20s	2.12s	2.43s

Metric	Dapper	NHIbernate	Microsoft EFCore
ORM Type [0.6]	Micro-ORM [0.75]	ORM [1]	ORM [1]
Query Type [0.7]	SQL [0.75]	LINQ, Criteria API, QueryOver, SQL [1]	LINQ & SQL [.8]
Design Capabilities	Database First	Code First &	Code First &

[0.7]	[0.75]	Database First [1]	Database First [1]
Start Up Time [1]	Only requires import and and the use of statements [1]	Lots of metadata preparation slows down the initial startup time [.1]	Establish DB Context Class [0.75]
Insert Row Speed [0.9]	Executed within 3.8s [1]	Executed within 10.3s [0.4]	Executed within 71.94s [0.1]
Delete Row Speed [0.9]	Executed within 1.20s [1]	Executed with 2.12s [0.5]	Executed within 2.43s [0.5]
Total Weight	5.25	5	5.05

Recommendation

Based on the analysis between Dapper, NHibernate, and Microsoft EntityFramework.Core, Dapper is the more suitable technology for the specified metrics despite being a micro-ORM. Asides from features such as the query type, ORM type, and design capabilities, Dapper drastically outperforms EntityFramework Core and even NHivernate in operations that will be essential to the performance of the product. As Dapper resulted in the higher score, we are recommending Dapper as an ORM technology.