Research Article

Object Relational Mapping Framework Performance Impact

Dr V.Sivakumar[1], T.Balachander[2], Logu[3], Ramu Jannali[4]
[1]Associate Professor, Department of Computer Science and Engineering, Vel Tech Rangarajan Dr Sagunthala R&D Institute of Science and Technology, Chennai, India
[2]Assistant Professor, Department of Computer Science and Engineering, School of Computing, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Chennai, India
[3]Assistant Professor, Department of Computer Science and Engineering,
Saveetha School of Engineering, Saveetha Institute of Medical And Technical Science, Saveetha University, Chennai, India
[4]Sr. Systems Analyst, One Main Financial, 601 NW 2nd Street, Evansville, IN 47630

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 16 April 2021

Abstract: In a world where frameworks play a key role in software applications, people tend to forget about the impact different frameworks can have on different software applications. This paper will look into the impact of different object-relational mapping frameworks on relational query performance. The objective is to give an average performance impact based on a wide variety of object-relational mapping frameworks. Results are gathered from experiments using 8 different frameworks for 4 different programming languages. The languages considered in this paper are Java, C#, Python and PHP. First, self-defined research questions are answered using literature research. These questions provide background information about the topic of object-relational object-relational frameworks. Secondly, the setup used for conducting the experiments covered in this paper is discussed. Finally, it concludes by showing the performance impact of the discussed object-relational mapping frameworks.

Keywords: Object, ORM, SQL, relations, query, relational mapping

1.Introduction

In modern-day software development object-relational mapping frameworks are often used because of their easy learning curve, independence of a specific database type and easy to use. However, these pro's do come with their own cons. Since an object-relational mapping frame work generates database queries, the developer has less control over them and their efficiency. This may not have a significant impact when building, for instance, a small web application, but can have a big impact on software projects processing gigabytes of data. To find out what the impact of using object-relational mapping has on the performance of some projects, different research methodologies are needed. Those research methodologies will answer the sub research questions and the main research question. The first methodology that is used is literature research to compare different papers with each other and find out what those papers have in common with this research. Secondly, there are tests in 4 different programming languages to compare the object-relational mapping in different programming languages to find out if there are any differences in what programming language is used. In this research article, it concludes that the use of relational mapping frameworks will only significantly impact complex queries and not simple queries. This subsequently means that the use of relational mapping frameworks could be easily justified for most software project and should only be avoided in specific use cases where the database performance is of critical importance.

2. Proposed Work

- Answer self-defined research questions using literature research
- Find other research on this topic andsummaries for later use
- Setup the different ORM frameworks
- Measure the impact of different ORMframeworks
- Write a conclusion based onexperiment results

Main research question

What is the impact of different Object-Relational Mapping Frameworks on Relational Query Performance?

Sub research questions

I. What is an object relational mapping(ORM) framework?
II. How does an ORM framework work?
III. What are the downsides of ORM's?
IV. What are the alternatives to using anORM framework?

I. What is an object relationalmapping framework?

An object relational mapping framework (in short: ORM) is a solution for the problem that arises when trying to map objects inprogramming languages to database tables. The difference in approach is wide between different existing frameworks, but they all boildown to the same idea. They provide a mapping from the object layer (object classes) to the relational (database tables). These differ greatly. Objects are temporary, but data in the relational model is persistent. Objects do not abide a strict schema, whereas the as relationalsql database follows a strict schema.

An object relational mapping framework crosses the bridge in difference by providing an interface to the programming language. This interface can be used to let the object relational mapping framework generate SQL queries. Therefore the application itself does not need to contain any SQL code or have any SQL capabilities.

II. How does an ORMframework work?

An object relational mapping framework provides an interface that can be used to let theORM generate SQL queries. This means the application does not need to have any SQL capabilities. Therefore retrieving the name of the person with id 10 would not look like this:

```
var sql = "SELECT id, first_name, last_name, phone,
        birth_date, sex, age FROM persons WHERE id = 10";
var result = context.Persons.FromSqlRaw(sql).ToList();
var firstName = result[0]["first_name"];
```

But more like this:

From that it is observed it reduces the lines ofcode necessary for using any database functionality. It also greatly

```
var firstName = respository.GetPerson(10).GetFirstName();
```

decreases code maintenance, since the non ORM code needsto be updated every time the database table is changed, but the ORM code only needs to be updated when the first_name property is removed.

III. What are the downsides ofORM's?

Object relational mapping frameworks do suffer from some issues because of their design. As described by Karwin [6] call these issues anti-patterns.

ORM tools are based on the Active Record design pattern according to Karwin. This pattern maps an object to a row in a database table. This produces the issue that models are coupled closely with database schemas. This means that if a model changes the schema becomes invalid and vice versa.

On a more technical level, there arise even more problems. Chen et al[4] points out that object relational mapping frameworks include row by row implementations as loops. They dothis especially when such structures are memory efficient. However set base queries are more preferred in relational theory since they have better efficiency and lower resource costs. There is also the eager fetching problem.The eager fetching problem means that complete row data is fetched and the filtered within the application. This can cause up to 71% decrease in performance [3].

IV. What are the alternatives tousing an ORM framework?

The only widely used alternative for using an ORM framework seems to be using the buildin database connector library of the specified programming language [7]. The connector libraries is most programming languages are structured similarly and all contain functionality to retrieve, store, edit and delete data from a database. Using this alternative method does mean that all SQL has to be written by hand, thus none of it will be generated. This has multiple disadvantages. This first disadvantage is that this results in more lines of code. This means that the is more chance of bugs and also more code to maintain. The second disadvantage is that all SQL has to be rewritten when the application switches its database from for example PostgreSQL to MySQL. The third disadvantage is that when the database schemachanges, this application will not detect outdated SQL queries still in the application. The application will then only throw an exception when the outdated query is executed. This means that a bug can be presentwithin an application for a long time before being discovered.

The upside of writing the SQL queries yourself it that you gain full control over the database performance and behavior. This can be of great importance when the database performance is critical to the applicability of your application.

3. Research Method

For this research have selected 4 different programming languages to base our test resultson. We have chosen

to make tests for object relational mapping frameworks in Java, Python, Php and C#, since these are the most widely used languages for backend applications according to the TIOBE index of May 2020[1].
Within these 4 programming languages we have chosen 8 object relational mapping framework, 2 out of each language and the frameworks have been chosen at random, so asto maximize the best representation of the reality.
The chosen frameworks are Hibernate, JOOQ, Entity Framework, NHibernate, CakePHP, Laravel, Django and SQLAlchemy for Java, C#, PHP and Python respectively.

For each framework the required software was installed, all on the same system. Also the same database was used in combination with the different frameworks as to minimize the changing factors between all tests. The database used for all test is a PostgreSQL database.

After setup of each framework a few different queries were executed. The first query is astraightforward select row by id. This is the most simple query. The second query asks for list of relations of the base entity. The thirdand last query goes one step further and asks for a list of relations containing all subrelations of a given entity. The execution time of each query was recorded and noted in ms. For our results we have taken the average of the test results per framework since a real world application would also use queries of different complexity.

```
SELECT name FROM teacher WHERE id = 1;

SELECT s.name FROM subject s, teacher t
WHERE t.id = 1 AND s.teacher_id = 1;

SELECT st.name FROM student st, teacher t, subject su
WHERE t.id = 1 and s.teacher = 1 AND st.subject_id = su.id
```

4. Results

The results are as expected beforehand and seen in literature about the pro's and con's of using an object relational mapping framework.

Framework	Execution Time (ms)
Django	110
SQLAlchemy	112,5
No Framework	44,1

Table 1: Average Python execution times

Framework	Execution Time (ms)
Hibernate	91,4
JOOQ	71,6
No Framework	34,2

Table 2: Average Java execution times

Framework	Execution Time (ms)
Entity Framework	120,3
NHibernate	119,5
No Framework	44,3

Table 3: Average C# execution times

Framework	Execution Time (ms)
CakePHP	99,5
Laravel	90,1
No Framework	40,9

Table 4: Average PHP execution times

Clearly in all programming languages the object relational mapping framework gives a significant increase in average execution timeof the queries.

5. Conclusion

As clearly visible in Table 1 to 4 we conclude that the impact of an object relational mappingframework on relational query performance is a significant decrease in execution time when using an object relational mapping frameworkrelative to not using an object relational mapping framework.

Though in most applications the difference inexecution times would barely be noticeable, itshould be clear that when database performance is of paramount importance one should not use an object relational mapping framework.

6. Discussion

This research only used select queries to measure the execution times. Ideally we should have also used create, update and deletequeries here also, since that reflects more on reality.

This research only used one database, the PostgreSQL database. This choice was made to provide an extra constant factor between thedifferent tests. In other research multiple databases could also be used to see if the different object relational mapping frameworks work better with some than others.

In the results it is visible that there is a lesser time difference between using an object relational mapping framework and using plainPHP. This could be a topic for another paper since it is out of the scope of this paper

References

1. Index | TIOBE - The Software Quality Company. (n.d.). Retrieved 7 May 2020, https://www.tiobe.com/tiobe-index/
2. Halpin, T. Object-Role Modeling (ORM/NIAM). Handbook on Architectures ofInformation Systems, 1–23. 1998.https://doi.org/10.1007/3-540-26661-5_4
3. D. Colley, C. Stanier and M. Asaduzzaman, "The Impact of Object-Relational Mapping Frameworks on Relational Query Performance," 2018 International Conference on Computing, Electronics & Communications Engineering (iccece), Southend, UK, 2018, pp. 47-52, doi: 10.1109/iccecome.2018.8659222.
4. Chen, T. P., Shang, W., Jiang, Z. M., Hassan, A. E., Nasser, M., & Flora, P, " Detecting Performance Anti-patterns for Applications Developed using Object-Relational Mapping", ICSE 2014: Proceedings of the 36th International Conference on Software Engineering , May 2014 Pages 1001–1012https://doi.org/10.1145/2568225.2568259.
5. [G. Vial, "Lessons in Persisting Object Data Using Object-Relational Mapping" in IEEE Software, vol. 36, no. 06, pp. 3-52, 2019. Doi: 10.1109/MS.2018.227105428
6. B. Karwin, "SQL anti patterns: avoiding the pitfalls of database programming". Pragmatic Bookshelf, 2010
7. J. Armas, P. Navas, T. Mayorga, P. Rengifo and B. Arévalo, "Optimization of code lines and time of access to information through object-relational mapping (ORM) using alternative tools of connection to database management systems (DBMS)," 2017 2nd International Conference on System Reliability and Safety (ICSRS), Milan, Italy, 2017, pp. 500-504, doi: 10.1109/ICSRS.2017.8272872.