
Performance Evaluation of Java Based Object Relational Mapping Tools

SHOAIB MAHMOOD BHATTI*, ZAHID HUSSAIN ABRO**, AND FARZANA RAUF ABRO****

RECEIVED ON 13.03.2012 ACCEPTED ON 21.06.2012

ABSTRACT

Object persistency is the hot issue in the form of ORM (Object Relational Mapping) tools in industry as developers use these tools during software development. This paper presents the performance evaluation of Java based ORM tools. For this purpose, Hibernate, Ebean and TopLink have been selected as the ORM tools which are popular and open source. Their performance has been measured from execution point of view. The results show that ORM tools are the good option for the developers considering the system throughput in shorter setbacks and they can be used efficiently and effectively for performing mapping of the objects into the relational dominated world of database, thus creating a hope for a better and well dominated future of this technology.

Key Words: Mapping, Database Persistency, Object Relational Mapping.

1. INTRODUCTION

Every business application in world needs the long term persistency of the data for future utilization. Obviously all systems rely on three layered logical architecture of the software i.e. presentation/interface layer, business layer, and third is data layer and this third one requires lot of time for the implementation in software development life cycle [1]. During the mapping time development teams have to handle carefully the object relational impedance mismatch that mostly cost the data layer to take the back seat.

In real sense ORM provides the more efficient way by expelling the data layer from the scenario of the system.

Such type of mapping tools have been introduced by many open source communities like JBoss, Avaje, Oracle Fusion Middleware family, etc. also in this regards the business cooperation such as Microsoft is not behind by introducing Microsoft Entity Framework, etc.

ORM technique can easily overcome the paradigm mismatch problem that is encountered by the developer during handling of data layer in application development [1]. ORM technique can automatically create the front end and back end, therefore it is the intermediary bridge connecting the object oriented code with relational schema. ORM technique drives the thinking of developers from data layer and object to the data layer object i.e. persisting

* Post-Graduate Student, Department of Information Technology, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah.

** Professor, Department of Information Technology, Quaid-e-Awam University of Engineering, Science & Technology, Nawabshah.

*** Assistant Professor, Department of Electronic Engineering, Mehran University of Engineering & Technology, Jamshoro.

object. Hereinafter system itself handles mapping at the runtime through the SQL statements, like create, insert, delete, update, select and so on while keeping track of object update as well.

There is very little work done over the performance of the ORM tools. Only Cvetkovic and Jankovic [2] have performed the comparative study of two object mapper tools NHibernate and Microsoft Entity Framework based on .NET platform, however no work has been done on the performance of Java based ORM tools. So, there was need to recommend the best Java based ORM tool to a developer based on its performance. So the research study presented in this paper was conducted to measure the performance of three Java based ORM tools: Hibernate, Ebean and oracle TopLink.

The paper is divided as follows: next section presents the related work. Section 3 presents the results in detail along with the testing environment and then last section concludes the paper.

2. LITERATURE REVIEW

With regards to ORM tools, Cvetkovic and Jankovic [2] have performed the comparative study of two object mapper tools that are NHibernate and Microsoft Entity Framework which are based on .NET platform. Zyl, et. al. [3] have explained the performance misconception that surrounded to ORM tools by presenting the comprehensive studies of one object relational tool Hibernate with the object oriented database that is db4^o which persists the objects into the databases by using benchmark 007 to evaluate the execution speed of a set of classical operations of persistency in them. Zhang and Ritter [4] have measured the role of object-relational database mechanism into the development process of object-oriented software. The empirical theory test conducted by them led to the conclusion that there is

still a substantial breach connecting the object(s) with the object-oriented paradigm. Kopteff [5] conducted the research to determine the possibility of replacing object(s) mapping tools and relational databases with object databases, for that purpose he selected the Versant ODBMS and Hibernate ORM, de-facto object database benchmark 007 is used to measure the features of both.

Recently, Shiva, et. al. [6] have studied mapping annotation and their categories. They elaborated them in detail by mentioning ways the annotations maintain injecting dependencies into property like mail sources, data sources, EJBs, web services, environment entries, etc. Jordan [7] has prescribed the set of criteria by comparing persistency mechanism in order to determine the better mechanism of persistency in perspective of widely used benchmark. Chuanlong, [8] has performed a comprehensive research study on open source ORM tool Hibernate only. Zyl, et. al. [9] have researched that how vendor recommended optimization technique influences the Hibernate performance. The benchmark 007, customized to explore the Java based persistence frameworks, was used to generate a database for traversing, querying, updating and deleting purpose. Takasaka [10] in his master thesis goes to determine that most modern object oriented programming languages do not support directly object persistency by further claiming that transparent object persistency would go to increase output by permitting programmers to pay more attention on the implementation of trade logic. Therefore, according to the best of our knowledge, no work has been conducted to compare the performance of various Java based ORM tools such as Hibernate, Ebean and oracle Toplink. In this regard, this paper presents the performance measurement of these three ORM tools: Hibernate, Ebean and oracle TopLink. A comparative list of ORM tools is available at <http://c2.com/cgi/wiki/ObjectRelationalToolComparison>.

2.1 Hibernate

Being the most popular tool in the industry mapping object, it is an implementation of JPA (Java Persistence API) specification developed by JBoss community under the leadership of Gavin King. In Java sometimes we want our object(s) to live beyond the scope of the Java interpreter (JVM) so that the same state is available later, Hibernate further assists the programmer to exploit POJO-style realm representation in its function of applications that are well beyond ORM.

Hibernate was selected for this research because it is:

- (a) An open source tool
- (b) Most widely used in the area of mapping
- (c) Easy to understand and learn because of all sort of resources are available such as documentation, online forum, books, video tutorials at single glance, etc.

2.2 Ebean

It is based upon the newly design architecture of mapping which does not follow the specification of the JPA, like JPA entity manager. Ebean has a persistence context and by default it is transaction scoped and automatically managed. Ebean was selected for this research because it is:

- (a) An open source ORM tool
- (b) Ebean has sessionless API
- (c) Ebean has simpler query language

2.3 Oracle TopLink

TOP, where it all begins in early 1990's, stands for "The Object People". Initially for Smalltalk, then in the period

between 1996-1998 its Java version was introduced in the market. Suddenly after that, WebGain got its all the copyrights. In 2002 it was owned by the Oracle Inc, hereinafter it is the member of Oracle Fusion Middleware family. Oracle TopLink was selected for this research because it is:

- (a) The first ever and the oldest tool in ORM area
- (b) An open source tool
- (c) A member of Oracle Fusion Middleware family

3. TESTING ENVIRONMENT WITH QUERY DESCRIPTION

The testing environment is first presented.

3.1 Testing Environment

Johnson, [11] has emphasized on experimental work and experimental analysis of algorithms. For queries to be executed and results to be generated, the complete compilations of the queries were done on the single machine which consists upon hardware configuration as:

System Model: HP Compaq dc5100 MT(PW099ET)

System Manufacturer: Hewlett-Packard

Processor: Intel(R) Pentium(R) 4 CPU 3.40GHz

BIOS: Default System BIOS

Page File: 497MB used, 1948MB available

Memory: 1016MB RAM

OS: Service Pack 2 on Windows XP Professional (5.1, Build 2600)

DirectX Version: DirectX 9.0c (4.09.0000.0904).

On the contrary to the hardware the software configurations which were used are:

- hibernate-annotations 3.4.0GA
- ebean-2.7.4.
- TopLink
- hibernate-distribution3.3.2GA
- JDK
- Eclipse Galileo
- MySQL Server 5.5
- MySQLAdministrator 1.1

Considering the schema in the database server was a complex task for this research because as MySQL server does not come with any default schema like MS SQL and Oracle SQL server do, therefore a famous schema may be embedded within the database server. Therefore it was decided that the default schema of oracle server will be recreated in the MySQL server. The list of all database relations which was used throughout the testing, together with all number of records is: employee (emp) ≈ 5200 records and department (dept) = 40 records, there was one-to-many relationship between them.

3.2 Query Description

All three selected tools support multiple query languages, therefore, deciding and working on one particular query language depends upon the developer's choice and experience. Every tool has different query language settings as described:

3.2.1 Hibernate: CRITERIA API

It is a simpler API for persisting the entities, extracting, and modifying all its objects by constructing query-defining objects.

Criteria queries are structured by means of Java programming language APIs. Criteria query API is an

insightful and extensible technique to mine the object from the relational schema of database. The feature of Criteria Query API is to check syntax errors at compile time which is not the part of SQL language; Criteria API even enshrines the functionality of QBE (Query By Example) through that an example object(s) can supply the properties which a developer would like to replace as a substitute of having to spell the components of query step by step. API criteria also embed projections and aggregation methods including counts.

3.2.2 Ebean: Ebean Query Language

It is an expressly design query language for Ebean, that is thought to be more pretended to spawn graph of object. Ebean Query Language allows Ebean to simply maintain partial object(s) to return typed lists, set and maps using generic queries. With embedding of the partial object query support, a developer can maintain an optimal query for each use case. In case a developer has predetermined annotations to state whether a property is eager or lazy loaded then he/she will lost optimal queries for a particular use case. So developer will frequently encounter the queries which are suboptimal.

3.2.3 TopLink: TopLink Expressions

It is a dedicated querying language for the TopLink, where all expressions illustrate the use of methods such as to Upper Case mathematical methods, etc. where their prototypes contain Expression class, ExpressionMath class, respectively. Mostly expressions use standard boolean operators, such as AND, OR, and NOT, and a developer can combine multiple expressions to form more complex expressions. A simple expression usually consists of three parts. The first is attribute which represents a mapped attribute, second is the operator, which is an expression method such as Greater Than, Equal, or Like and the third is comparison or constant, which passes on to the certain value used to retrieve the object.

3.3 Performance Testing

To see the performance of ORM tools, this research work has measured the time required to execute the queries by ORM tools. There were total fourteen different queries that have been executed via these ORM tools. Each query was executed ten times and their results are then averaged and are presented in seconds. Among them, eleven are Select queries with different comparison and logical operators, one affects the join criteria and one each for insert, delete and update.

4. RESULTS

This section completely describes the queries and their behavior by simply screening queries and followed by their performance results.

Q1 (Query One) shows the insert query, this query was run ten times on all three ORM tools. Fig. 1 shows that insert task is completed by Hibernate on average in 1.5 seconds, by Ebean on average in 1.8 seconds while TopLink also took on average 1.8 seconds. From this statistics it is obvious that Hibernate is executed in quicker time averagely as compared to the Ebean and TopLink.

Q2 is a query of select with equal to operator alongside order by, Fig. 2 shines the stars for the Ebean which just took 1.4 seconds on average to execute the query in the shortest time.

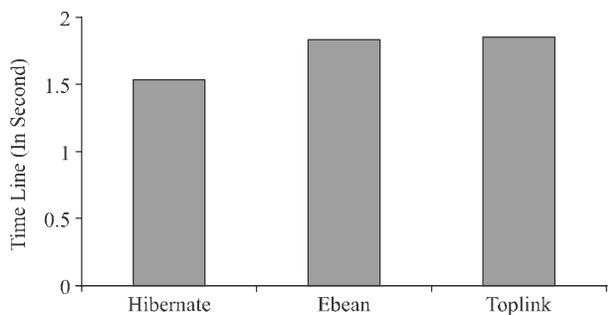


FIG. 1. COMPARISON OF INSERT QUERY

Q3 is a query of select with greater than operator. Fig. 3 shows the time of the execution where Ebean took only 1.7 seconds on average to execute in shortest setback.

Q4 is the query with greater than equal to along with 2 order by clause. Fig. 4 shows that the Ebean executed faster i.e. in 1.7 seconds.

Q5 is the query with less than operator. Fig. 5 shows the Ebean executed faster and just took 1.4 seconds.

Q6 is the query with less than equal to operator. Fig. 6 shows the Ebean as the winner by 1.4 seconds.

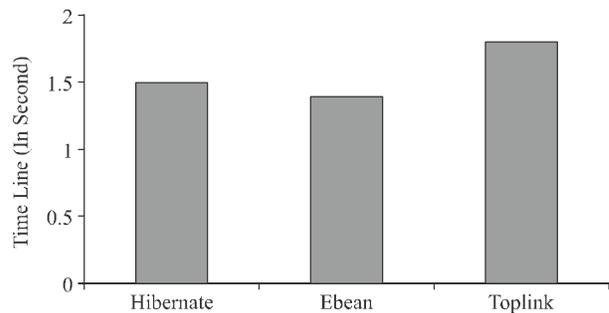


FIG. 2. COMPARISON OF SELECT WITH EQUAL TO AND ORDER

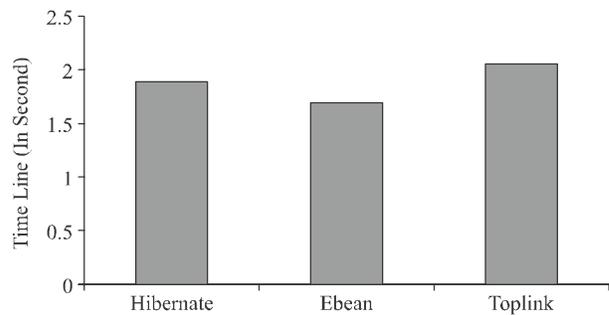


FIG. 3. COMPARISON OF SELECT WITH GREATER THAN OPERATOR

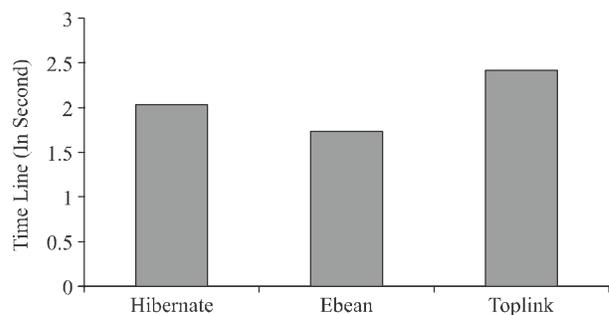


FIG. 4. COMPARISON OF SELECT WITH GREATER THAN EQUAL TO ALONG WITH 2 ORDER

Q7 is the query with not equal operator. Fig. 7 shows again that the Ebean won by running in shortest time, i.e. in 1.7 seconds.

Q8 is the query with logical operator and. Fig. 8 shows again the Ebean as winner (1.4 seconds).

Q9 is the query with so called other comparison operator which is between and operator. Fig. 9 shows the Ebean executes this comparison operator faster just in 1.5 seconds.

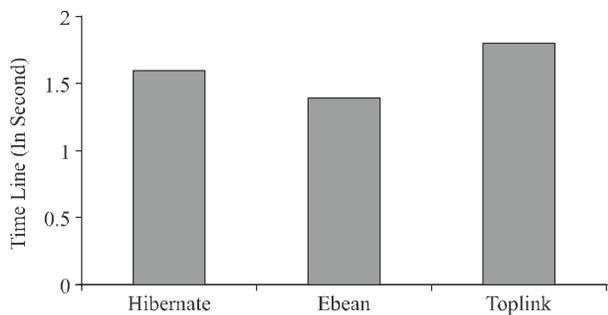


FIG. 5. COMPARISON OF SELECT WITH LESS THAN OPERATOR

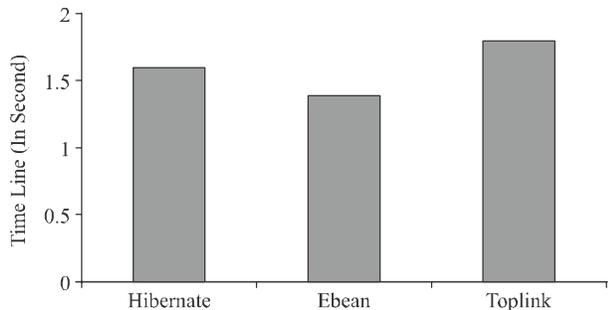


FIG. 6. COMPARISON OF SELECT WITH LESS THAN EQUAL TO OPERATOR

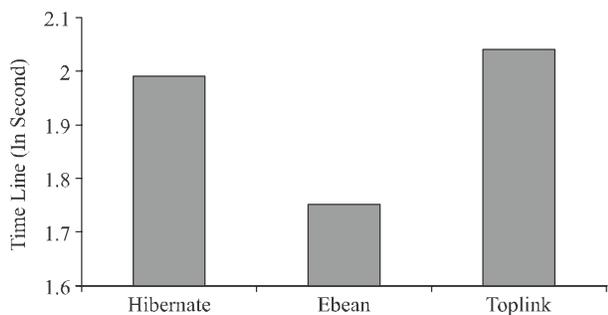


FIG. 7. COMPARISON OF SELECT WITH NOT EQUAL OPERATOR

Q10 is the query with another comparison operator which is in(set) operator. Fig. 10 shows that Ebean executed this query faster just in 1.4 seconds.

Q11 is the query with comparison operator which is like operator. Fig. 11 shows the Ebean again faster (1.4 seconds).

Q12 is one of the advanced queries with an inner join amongst emp and dept tables. Fig. 12 shows the Ebean executed this join condition in 1.3 seconds only.

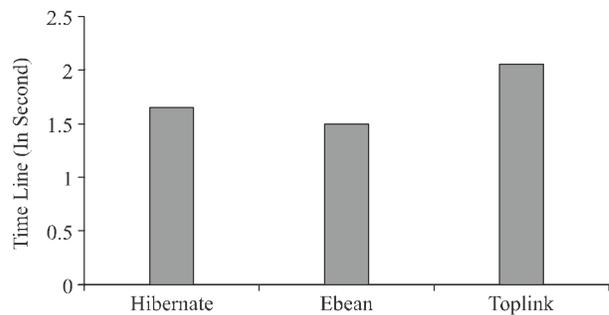


FIG. 8. COMPARISON OF SELECT WITH AND OPERATOR

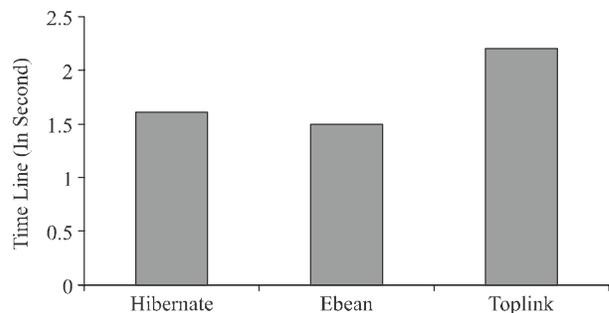


FIG. 9. COMPARISON OF SELECT WITH BETWEEN AND OPERATOR

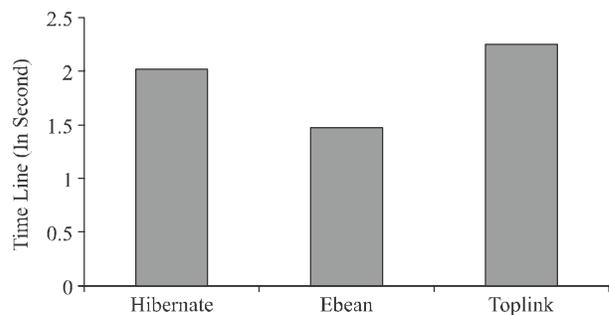


FIG. 10. COMPARISON OF SELECT WITH IN OPERATOR

Q13 is of updating the schema emp. Fig. 13 shows the Ebean performed updating just in 1.4 seconds means in shortest setback than Hibernate and TopLink.

Q14 is of deleting the records from the schema emp. Fig. 14 shows the Ebean performed deleting just in 1.4 seconds hence in shortest setback than Hibernate and TopLink.

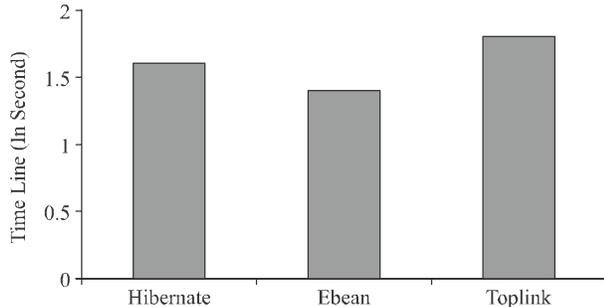


FIG. 11. COMPARISON OF SELECT WITH LIKE OPERATOR

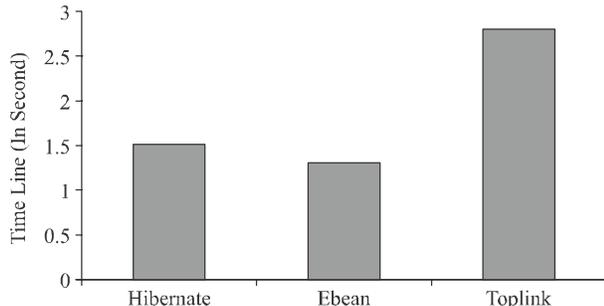


FIG. 12. COMPARISON OF SELECT WITH EQUIJOIN

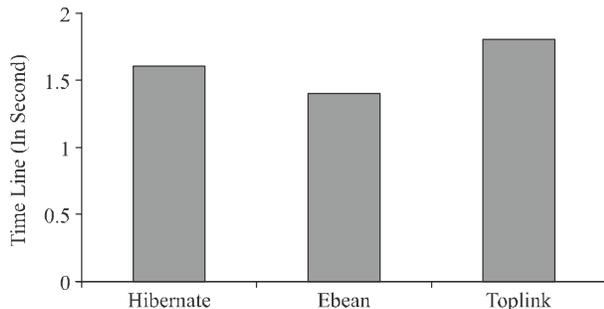


FIG. 13. COMPARISON OF UPDATE QUERY

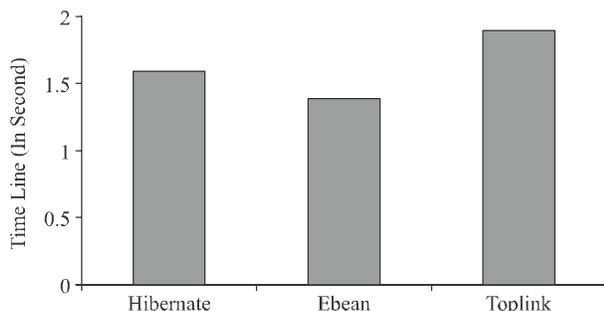


FIG. 14. COMPARISON OF DELETE QUERY

4. CONCLUSION AND FUTURE WORK

ORM tools are the handy tools for the developers. Therefore, it was necessary to measure their performance in order to recommend the better and faster ORM tool for developers. According to our results it is lucid from the Figs. 1-14 that Ebean is the fastest tool amongst three selected tools. This is due to its sessionless API. Even it provides the simpler query language for retrieving the records. As far as the insert query is concerned where Hibernate succeeds against the Ebean, it is critically examined that while inserting the data Hibernate does not need to create the session with database server, therefore Hibernate stood first in the insertion matter. So the best tool recommended for developers is the Ebean followed by Hibernate.

As the future work, this work will be enhanced to the more advanced queries evaluation on the same tools in near futures and as well as for the other ORM tools and on latest machines (i.e. Core i7). Regarding the platform independency the ORM technology will be measured on the different operating systems along with different parameters.

ACKNOWLEDGEMENTS

The authors are highly thankful to Quaide-Awam University of Engineering, Science & Technology, Nawabshah, and Mehran University of Engineering & Technology, Jamshoro, Pakistan, for providing necessary funding and research facilities during their research studies. This work has been carried out during the MS thesis of the first author.

REFERENCES

- [1] Keene, C., "Data Services for Next Generation SOAs", SOA World Magazine, 2004. <http://soa.sys-con.com/node/47283>
- [2] Cvetkovic, S., and Jankovic, D., "A Comparative Study of the Features and Performance of ORM Tools in a .NET Environment", ICODDB 2010, LNCS 6348, Springer-Verlag, Faculty of Electronic Engineering, Aleksandramedvedeva 14, 18000 niš, Serbia, 2010.
- [3] Zyl, P.V., Kourie, D.G., and Boake, A., "Comparing the Performance of Object Databases and ORM Tools", Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries, pp. 1-11, Somerset West, South Africa, 2006.
- [4] Zhang, W., and Ritter, N., "The Real Benefits of Object-Relational DB-Technology for Object-Oriented Software Development", Proceedings of 18th British National Conference on Databases, Oxford, Advances in Databases, LNCS 2097, pp. 89-104, Springer, 2001.
- [5] Kopteff, M., "The Usage and Performance of Object Databases compared with ORM Tools in a Java Environment", 1st International Conference on Objects and Databases, Berlin, Germany, 2008.
- [6] Prakash, S., Saini, J.P., Singh, R.K., and Singh, K., "Features of Annotations and their Applications", International Journal of Computer Applications, Volume 16, February 2011.
- [7] Jordan, M., "A Comparative Study of Persistence Mechanisms for the Java™ Platform", Sunmicrosystems, USA, September, 2004.
- [8] Xia, C., "Efficient implement of ORM (Object/Relational Mapping) use in J2EE Framework: Hibernate", IEEE International Conference on Computational Intelligence and Software Engineering, pp. 1-3, 2009.
- [9] Zyl, P.V., Kourie, D.G., Coetzee, L., Boake, A., "The Influence of Optimizations on the Performance of an Object Relational Mapping Tool", SAICSIT, Vanderbijl Park, South Africa, ACM, 2009.
- [10] Takasaka, S., "Survey of Persistence Approaches", Department of Computer and Systems Sciences, Master Thesis, Royal Institute of Technology/Stockholm University in Collaboration with Swiss Federal Institute of Technology - ETH, Zurich, December, 2005.
- [11] Johnson, D.S., "A Theoretician's Guide to the Experimental Analysis of Algorithms", AT&T Labs, Research, USA, November 2001, www2.research.att.com/~dsj/papers/experguide.pdf