Hibernate Application for Cost Effective and Efficient Project Tracking System

SHARAYU LOKHANDE, ANUP KADAM

Army institute of Technology, Pune, India

ABSTRACT

This article describes object oriented applications with relational databases. Currently there exist many technologies supporting object-relational mapping (ORM) for J2EE technology. There is a comparison of Hibernate over JDBC in case of their performance. Project tracking application designed to satisfy both project management and finance at the same time, making the organization more efficient and reliable. Performance in case of project execution and control improves with the help of ORM tool Hibernate.

KEYWORDS: *Java, J2EE, ORM, JDBC, Hibernate, Project execution and control.*

I.   Introduction

Successful businesses worldwide use computer-aided time management processes to keep things intact, predictable and under control. Enterprises utilize very complex solutions to track work time load and distribution of work time between various tasks an employee has to accomplish. Time is the key factor for determining work efficiency as well as business and personal performance. The highly competitive business environment demands that managers run their organizations leaner, faster and smarter - with fewer people and lower costs.

Using an intuitive web based timesheet entry interface to the employees will be able to capture their time in a quick and efficient manner, reducing data entry times and speeding up the cost collection and reconciliation processes.

All companies have to track and report on employee work. Collecting timesheet information from employees can be tedious, challenging and costly. To streamline the collection, Timesheet Software allows anyone with an internet connection to enter time and submit it for approval. Approval requests can be routed to managers, project managers, clients and HR members electronically. Managers can review resulting data to proactively make adjustments in employee schedules and client deliverables.

II.   The Role of Open Source Software

Open source software plays an important role in applications by allowing its basic software elements - virtual machine images and appliances — to be created from easily accessible components. This has an amplifying effect:

- Developers, for example, can create a database appliance by layering MySQL software onto an instance of the Open Solaris Operating System and performing customizations. Appliances such as these enable applications to be created, deployed, and dynamically scaled on demand.

III.   Bottleneck with using JDBC

The recent emergence of object/relational mapping frameworks (such as JDO and Hibernate) and SQL mapping frameworks (such as iBATIS) did not occur in a vacuum. Instead, they emerged from the Java community's repeated frustrations with JDBC. Let's review the problems with using JDBC

directly in order to understand the motivations behind the newer frameworks. There are three main reasons why using JDBC directly is not a good choice for many applications:

- Developing and maintaining SQL is difficult and time consuming: some developers find writing large, complex SQL statements quite difficult. It can also be time consuming to update the SQL statements to reflect changes in the database schema.
- There is a lack of portability with SQL: because you often need to use database-specific SQL, an application that works with multiple databases must have multiple versions of some SQL statements, which can be a maintenance nightmare. Even if your application only works with one database in production, SQL's lack of portability can be an obstacle to using a simpler and faster in-memory database such as Hypersonic Structured Query Language Database Engine (HSQLDB) for testing.
- Writing JDBC code is time consuming and error-prone: you must write lots of boilerplate code to obtain connections, create and initialize prepared statements, and clean up by closing connections and prepared statements. You also have to write the code to map between Java objects and SQL statements. As well as being tedious to write, JDBC code is also error-prone.

The first two problems are unavoidable if your application must execute SQL directly. Sometimes, you must use the full power of SQL, including vendor-specific features, in order to get good performance. Or, for a variety of business-related reasons, your DBA might demand complete control over the SQL statements executed by your application, which can prevent you from using persistence frameworks that generate the SQL on the fly. Often, the corporate investment in its relational databases is so massive that the applications working with the databases can appear relatively unimportant.
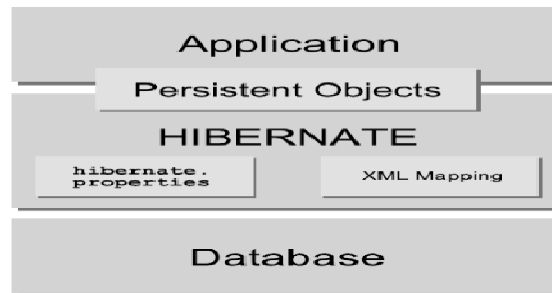
IV.   Hibernate Architecture

Figure 1 : Hibernate architecture

The figure 1 shows Hibernate using the database and configuration data to provide persistence services (and persistent objects) to the application.

An application that uses Hibernate API must use a single configuration file. A configuration file for a Hibernate application will tell some of the information like the URL of the Database Server to which the application wants to connect, the username/password for the Database, class name of the Driver file and with other set of preferences.

A Hibernate configuration file can be an XML-based file (hibernate.cfg.xml) or it can be ordinary Java properties file (hibernate.properties) with key-value combination. This configuration file should be placed in the run-time classpath of an application. If both the XML-based configuration file and the properties-based configuration file are found in the classpath, then the XML-based configuration file will take preference over the other.

Hibernate has two-level cache architecture:
The first-level cache is the persistence context cache. A Hibernate session lifespan corresponds to either a single request or a conversation. This is mandatory first-level cache that also guarantees the scope of object and database identity.

The second-level cache in Hibernate is pluggable and may be scoped to the process or cluster. This is a cache of state (return by value), not of actual persistence instances. A cache concurrency strategy defines the transaction isolation details for a particular item of data, whereas the cache provider represents the physical cache implementation.
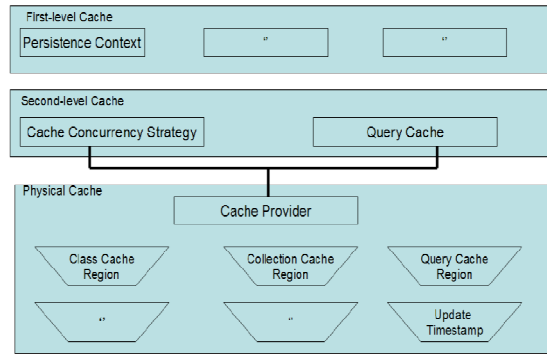
Figure 2 : Two-level cache architecture

Use of the second-level cache is optional and can be configured on a per-class and per-collection basis—each such cache utilizes its own physical cache region.
Hibernate also implements a cache for query result sets that integrates closely with the second-level cache.

V.   Will an Application Benefit from Caching?

The short answer is that it often depends on whether it is CPU bound or I/O bound. If an application is I/O bound then the time taken to complete a computation depends principally on the rate at which data can be obtained. If it is CPU bound, then the time taken principally depends on the speed of the CPU and main memory. While the focus for caching is on improving performance, it also worth realizing that it reduces load. The time it takes something to complete is usually related to the expense of it. So, caching often reduces load on scarce resources.

A. Speeding up CPU bound Applications

CPU bound applications are often sped up by:
• improving algorithm performance
• parallelizing the computations across multiple CPUs (SMP) or multiple machines (Clusters).
• upgrading the CPU speed.
The role of caching, if there is one, is to temporarily store computations that may be reused again. An example from Ehcache would be large web pages that have a high rendering cost. Another caching of authentication status, where authentication requires cryptographic transforms.

B. Speeding up I/O bound Applications

Many applications are I/O bound, either by disk or network operations. In the case of databases they can be limited by both. There is no Moore's law for hard disks. A 10,000 RPM disk was fast 10 years ago and is still fast. Hard disks are speeding up by using their own caching of blocks into memory. Network operations can be bound by a number of factors:
• Time to set up and tear down connections
• Latency, or the minimum round trip time
• Throughput limits
• Marshalling and unmarhshalling overhead
The caching of data can often help a lot with I/O bound applications. Some examples of Ehcache uses are:
• Data Access Object caching for Hibernate
• Web page caching, for pages generated from databases.

VI.   Project Execution and Control

Project Execution and Control is where most of the resources are applied/expended on the project. A significant number of team members will join the project at the beginning of this phase. The primary task of the Project Manager during Project Execution and Control is to enable the Project Team to execute the tasks on the defined Project Schedule and develop the product or service the project is

expected to deliver. The Project Manager uses the processes and plans prepared during Project Initiation and Project Planning to manage the project, while preparing the organization for the implementation of the product/service and for transitioning the product/service responsibility from the Project Team to the Performing Organization.

There are many groups of people involved in the project lifecycle.

The Project Team is a group that is responsible for planning and executing the project. It consists of a Project Manager and a variable number of Project Team members, who are brought in to deliver their tasks according to the Project Schedule.

The Project Manager is the person who is responsible for ensuring that the Project Team completes the project. The project Manager develops the Project Plan with the team and manages the team's performance of project tasks. It is also the responsibility of the Project Manager to secure acceptance and approval of deliverables from the Project Sponsor and Stakeholders.

The Project Team Members are responsible for executing tasks and producing deliverables as outlined in the Project Plan and directed by the Project Manager, at whatever level of effort or participation has been defined for them. On larger projects, some Project Team members may serve as Team Leaders, providing task and technical leadership.

VII.   Case Study

The e-Job Card is a full-featured web based time tracking, timesheet software. The e-Job Card allows to easily manage employee timesheets, approve timesheets and calculates payrolls, tracks project time and prepares invoices based on tasks and/or employees' and contractors' rates. The e-Job Card is software designed for those who need a customizable timesheet tracking and it is particularly convenient for accountants, consulting companies, contractors, IT specialists, temporary agencies and lawyers. This software reports flexible and customizable time and tasks. The e-Job Card is not just another time and attendance system, but it's a web based timesheet and project tracking application designed to satisfy both project management and finance at the same time, making the organization more efficient and reliable.

Most of the business applications work with relational databases. Moreover applications are being made faster and faster. It takes almost one third the time of creating application based on JDBC/SQL to make persistent tier. To work efficiently and faster developers reach for ORM technologies. ORM technologies mediates between object oriented architecture system and relational environment.

1.   Bottlenecks in existing time system

The DailyTimesheet is the existing in-house legacy software, which has been used over several years to track the individual and project timesheets. There are several issues with existing timesheet system. Some of the major issues associated with the existing timesheet system are as follows,
- Poor project tracking system.
- The system doesn't track the time history for the projects over the years. This is a major disadvantage in case there is a need to quote again in a similar project.
- It has become very difficult to calculate the efficiency of the individuals. Because there is no mechanism to compare the time required to complete the task and time taken by the individual.
- The time database is not organized in a structured way nor is it convenient for accounting.
- The project manager needs to compile the complete project-hours manually for the entire team members who worked on the project to compute the team efficiency.

The new web based project tracking system, e-Job Card has been designed to remove the above issues associated with the existing in-house legacy software (DailyTimesheet). The Java developers are facing some issues e.g. they need to store and retrieve persistent data. To do this operation there are several options available to them such as JDBC, JDO, object-relational mapping tools, etc.

The JDBC has certain disadvantages as follows:
- Lack of portability with SQL.
- Developing and maintaining SQL becomes difficult and it also increases time.
- JDBC code is time consuming and error prone.

Because of above disadvantages the time taken for creating applications based on JDBC/SQL is more compared to that for persistent tier. The solution for managing persistent data is one of the few challenges that modern technologies/ products are facing. The developers need not spend significant development time on designing the data access layer of an application, instead of this they can directly utilize the most standards-compliant open source Java persistence implementation. Our design makes use of ORM tool and gives a better performance. To prove this, in the new application we are going to analyse the performance of data access on database for different types of CRUD operations by using the open source Java persistence framework "Hibernate", in comparison with plain JDBC.

2. Contribution

The e-Job Card project tracking system is developed to resolve the issues related to the existing in-house legacy software (DailyTimesheet). The e-Job Card is an application for the employee to define, view and track their project tasks and generate task timesheets based on the effort spent on these through a web enabled application. Following are the solutions associated with the above defined issues with existing in-house legacy software:

- The new web based project tracking system e-Job Card provides the user to fill their efforts on the task basis, which helps the project manager to easily track the project efforts.
- The e-Job Card project tracking system provides the database time history for the projects tasks. This helps to quote similar tasks of the projects in future.
- The e-Job Card project tracking system allows the project manager to track the estimated time required to complete the task verses time taken by individual to complete the given task. The Project manager accesses the e-Job card to track the project based on each task level. This also helps to identify the bottleneck in the project.
- The e-Job Card project tracking system captures the invoice format of accounting after the project completion.
- The e-Job Card project tracking system captures the total summary of the effort taken on the projects.

The e-Job Card system allows the project manager to assign the tasks to the individual team members. The team members will prioritize the tasks, align with project plan and select what projects/tasks to work on. By the developer's perspective this application proves that the persistence framework supports all major O/R mapping designs allowing specifications of persistence needs using XML or Java5 annotations enabling the development team to spend their valuable time in providing value added business logic rather than the routine task of storing/retrieving objects. Application depends on the ORM for all the database-related services like persisting service (for saving the data) and query service (for retrieving existing data from the database). The ORM takes care of communicating with the appropriate database. To achieve better performance over plain JDBC we are using ORM tool in our application for report generation task. In particular, our design choice has added to the options available for connectivity to both relational and object-oriented databases.

3. Architecturally-Significant Model Elements

Business Components
The business components are divided in to five layers as
- *Communication layer*: The user interface logic is topmost. Code responsible for the presentation and control of page and screen navigation forms the presentation layer.
- *Business layer*: The exact form of the next layer varies widely between applications. It's generally agreed, however, that this business layer is responsible for implementing any business rules or system requirements that would be understood by users as part of the problem domain. In some systems, this layer has its own internal representation of the business domain entities. In others, it reuses the model defined by the persistence layer.
- *Persistence layer*: The persistence layer is a group of classes and components responsible for data storage to, and retrieval from, one or more data stores. This layer necessarily includes a model of the business domain entities (even if it's only a metadata model).

➤ *Database*: The database exists outside the Java application. It's the actual, persistent representation of the system state. If an SQL database is used, the database includes the relational schema and possibly stored procedures.

➤ *Helper/utility classes*: Every application has a set of infrastructural helper or utility classes that are used in every layer of the application (for example, Exception classes for error handling). These infrastructural elements don't form a layer, since they don't obey the rules for interlayer dependency in a layered architecture.

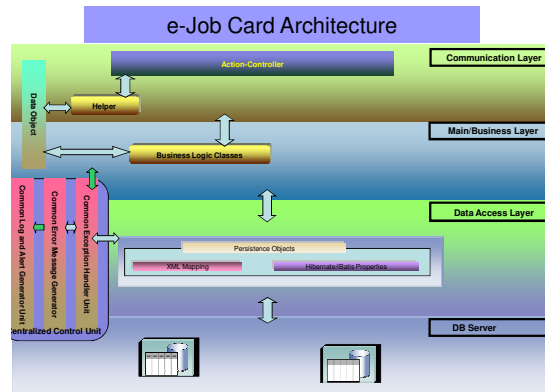The application business components and the layers of the e-Job Card are shown in the diagram below.

Figure 3 : application business components and the layers of the e-Job Card

4. Key benefits of e-Job Card

• **Improved Resource Utilization**: All aspects of resource and project management processes have been integrated into one standard web based solution. For example, staffs are now able to log time expended against project plans and e-Job Card has provided accurate and detailed resource information to IT Managers in order to improve utilization.

• Provided an automated and centralized method of tracking and monitoring based working.

• Delivered a quick and easily unified method of identifying under-utilized staff with obsolete skills, which could be cross trained and used to support overworked staff or replace contractors.

• **Improved Planning and Recharging**: The Commercial IT Division is now able to plan and manage projects more efficiently. The standardization of processes into one integrated business solution has radically improved project cost accuracy enabling the timely recharging and billing to internal customers. The result has been a dramatic increase in the number of chargeable hours, leading to accurate project costs.

• **Improved cash flow:** Many businesses are required to track the time spent on various projects/tasks. These are consulting companies or companies where there is a consulting component to the business. Timely billing to the client can significantly improve cash flow. A lot of time is spent collecting the billable time information from the consultants / employees. They could come in the form of emails, faxes, spreadsheet files, etc. By using web timesheet, project manager could eliminate the process of collecting and consolidating timesheet information.

• **Better project management:** There are businesses that collect time spent on various projects for better project management. Using e-Job Card will allow these businesses to collect timecard information and improve project management. Apart from collecting just timecard information, employees can also add project comments. These comments can form an ongoing project 'diary' detailing the project progress and status. Managers can consolidate these notes by project, reporting period, employee, etc. This becomes a very powerful system in analysing the project on an ongoing basis and provides historical information.

- **Minimal roll out cost:** Apart from the business benefits of e-Job Card, there are many operational benefits. Some of them are:
  - o Low or virtually no training required.
  - o Seamless integration into the Intranet.
  - o The application is accessible worldwide (As long as the Intranet is securely connected to the Internet).
- **Real-Time Management Reporting:** e-Job Card web based reporting technology has helped reduce administrative time spent on management reporting.

5. Performance evaluation

Figure 4 shows the time required for Hibernate as well as JDBC with the same input. It can be concluded that the time required for hibernate is much more less as compared to the hibernate.
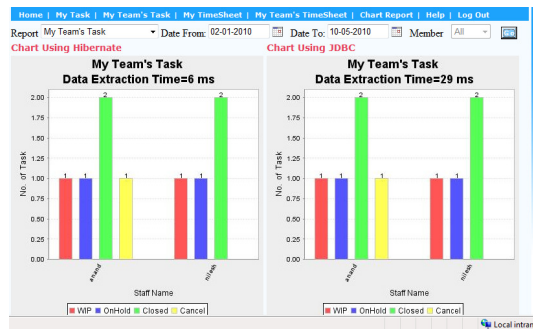

Figure 4 : Comparitive study

VIII.   Conclusions

Realization of example application show way of defining persistence of object graph, using API and how to write object oriented query. Based on results using example application the most performance technology is Hibernate. The e-Job Card time tracking system improves performance and the communication between the distributed and remote teams. The e-Job Card heightens the visibility and control of application development lifecycle-giving a better control over the entire development process, from the management stage through development. Besides reporting the time spent on project activities, users can now report against non-project activities and administrative time through the use of this application. These processes make possible the capturing of time spent on task-oriented work and non-project time commitments that historically have not been tracked in Project plans.

References

1. IEEE Recommended Practice for Software Design Specification, ANSI/IEEE Std 830-1998.
2. IEEE Guide for Developing System Design Specifications, ANSI/IEEE Std 1233-1996.
3. Chidamber and Kemerer: A metric suite for object oriented design IEEE Transactions on software Engineering, VOL. 20, NO. 6. JUNE 1994
4. Rajib Mall,"Fundamentals of Software Engineering", (Prentice-Hall India).
5. Herbert Schildt,"Java 2: The Complete Reference", Fifth Edition (Paperback).
6. Inderjeet Singh, Beth Stearns, Mark Johnson, and the Enterprise Team,"Designing Enterprise Applications with the J2EETM Platform", Second Edition.
7. Piotr Ziemniak, Bartosz Sakowicz, Andrzej Napieralski "Object Oriented Application Cooperation Methods with Relational Database (ORM) based on J2EE Technology"CADSM'2007, February 20-23, 2007, Polyana, UKRAINE.
8. Rod Johnson ,"J2EE Development Frameworks " , Interface21, January 2005
9. Hibernate Reference Documentation ,Version: 3.0.3
10. Christian Bauer, Gavin King, "Hibernate in action" (Manning publications Co.)
11. J. Wojciechowski, J. Murlewski, B. Sakowicz, A. Napieralski, "Object-relational mapping application in web-based tutor-supporting system", CADSM, Lviv- Polyana, Ukraine, Feb. 23-26, 2005,pp. 307-310, ISBN 966-553-431-9.

Authors Biographies

Sharayu Lokhande received her Master degree in Computer Engineering in year 2008 presently she is working as Assistance professor in Computer Science and Engineering Department, Army Institute of technology, Pune, Maharashtra. She has 9 years of teaching and professional experience. My area of interest includes database, algorithms. She has published in 4 National & 3 International conferences.

Anup Kadam is perusing his M.Tech in Computer Networking in Pune university from Sinhagad college of engineering, Pune, Maharashtra. He is working as Assistance professor in Computer Science and Engineering Department, Army Institute of technology, Pune, Maharashtra. He has 7 years of teaching and 1.6 years of industrial experience. His area of interest includes Operating system and computer networking, he has published in 1 International conferences.