DEVELOPING HIGH PERFORMANCE IN-HOUSE MESSAGE LOADERS

A PROJECT REPORT

*Submitted by*

**Resika .S.B (111712104083)**

**Yamini.V (111712104119)**

***In partial fulfilment for the award of the degree***

*Of*

BACHELOR OF ENGINEERING

IN

**COMPUTER SCIENCE AND ENGINEERING**

 

RMK ENGINEERING COLLEGE, KAVARAIPETTAI

ANNA UNIVERSITY :: CHENNAI 600 025

APRIL 2016

**ANNA UNIVERSITY : CHENNAI 600 025**

**BONAFIDE CERTIFICATE**

Certified that this project report “**DEVELOPING HIGH PERFORMANCE IN-HOUSE MESSAGE LOADERS**” in the bonafide work of “**Resika.S.B** (111712104083), **Yamini.V** (111712104119)” who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE**  **Dr. K.L.Shanmuganathan, M.E.,M.S.,Ph.D**  HEAD OF THE DEPARTMENT  PROFESSOR  Department of Computer Science and  Engineering,  R.M.K Engineering College,  Kavaraipettai-601206. | **SIGNATURE**  **Dr. B.Jaison, M.E., Ph.D**  SUPERVISOR  PROFESSOR  Department of Computer Science and Engineering,  R.M.K Engineering College,  Kavaraipettai-601206. |

Submitted to Project Viva Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| **EXTERNAL EXAMINER** | **INTERNAL EXAMINER** |

**ABSTRACT**

This project is to enhance the performance of the existing message loaders used in financial messaging. The traditional concept of querying data from a database will face time constraints. Hence we use in memory cache to enhance the performance. The XML files are feed as input from the queue. The messages are read and converted to java objects using jaxb parser. All these java objects are then put into a HashMap. These java objects are then fed into the in-house memory using hazelcast. The loading of information into the database will then be done by the hazelcast.  Every node adds their CPU to the cluster. Nodes can fail randomly without data loss or significant performance impact to running applications. This is a way for developers to easily program the cluster of machines as if it were a single machine. It enables very large data sets to be manipulated in main memory. Every node adds their RAM to the cluster’s memory pool. In-Memory Data Grids are often used with Databases in order to improve performance of applications, to distribute data across servers, clusters and geographies and to manage very large data sets or very high data ingest rates. Thus loading data into the database is fast and efficient.

**ACKNOWLEDGEMENT**

Support  on  demand, encouragement  at  the  needed  moment  and  guidance  in the  right  direction  are  in  dispensable  for  the  success  of  any  project. We  have received  these  in  excess  from  all  corners  from  various  people,  we  are glad  to  submit  our  gratitude  to  them.

We thank **Shri.R.S.Munirathinam, Chairman,** and **Shri.R.M.Kishore, Vice Chairman** of   RMK  group  of  Institutions  for  extending  a  generous hand  in providing  the  best  of  resources  to  the  college. Dr**. Elwin Chandra Monnie, the esteemed  Head  of  our  Institution**  has  been  a  source  of

motivation to  all  the staffs  and  students  of  our  college. We are so much thankful to him. Our sincere thanks to **Dr. Mohammed Junaid, the Vice Principal** for giving the continuous support for the completion of our project.

Our sincere thanks to **Dr. K.L.Shanmuganathan, the  Head  of  the  Department** for  his continuous  support  and  motivation  throughout  our  project.

We  extend  our  profound  gratitude to **Dr. K.Sethukarasi our Project Coordinator**  and **Dr.B.Jaison our Guide** for his guidance, who  has indeed  been  a  polestar throughout  the  course  of  the  project, we thank  him  for  giving   us  full  support to  complete  the  project  successfully.

Last,but not the least,wetake this opportunity tothank  all  the  **staff  members**  of  the  Department  of  Computer Science and Engineering. Regards  to  **our  family, classmates  and  friends**  who  offered  an  unflinching  moral  support  for  completion  of  this  project.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | iii |
|  | **LIST OF FIGURES** | viii |
|  | **LIST OF ABBREVATIONS** | ix |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 OVERVIEW | 1 |
|  | 1.2 OBJECTIVE OF THE PROJECT | 1 |
|  | 1.3 NEED FOR THE STUDY | 2 |
|  | 1.4 SCOPE FOR THE PROJECT | 2 |
|  |  |  |
| **2.** | **LITERATURE SURVEY** | 3 |
|  |  |  |
| **3.** | **SYSTEM ANALYSIS** | 7 |
|  | 3.1 PROBLEM DEFINITION | 7 |
|  | 3.2 EXISTING SYSTEM | 7 |
|  | 3.3 PROPOSED SYSTEM | 7 |
|  | 3.4 FEASIBILITY STUDY | 8 |
|  | 3.4.1 TECHNICAL FEASIBILITY | 8 |
|  | 3.4.2 ECONOMICAL FEASIBILITY | 9 |
|  | 3.4.3 OPERATIONAL FEASIBILITY | 9 |
|  | 3.5 SYSTEM CONFIGURATION | 10 |
|  | 3.5.1 INTRODUCTION | 10 |
|  | 3.5.2 HARDWARE AND SOFTWARE REQUIREMENTS | 10 |
|  | 3.5.2.1 HARDWARE REQUIREMENTS | 10 |
|  | 3.5.2.2 SOFTWARE REQUIREMENTS | 11 |
|  |  | 7 |
|  |  |  |
| **4.** | **TECHNOLOGIES USED** | 12 |
|  | 4.1 JAVA | 12 |
|  | 4.2 ECLIPSE | 12 |
|  | 4.3 ANALYSED TECHNOLOGIES | 12 |
|  | 4.3.1 XML | 12 |
|  | 4.3.2 SPRING | 13 |
|  | 4.3.3 JAXB | 13 |
|  | 4.3.4 HIBERNATE | 14 |
|  | 4.3.5 MAVEN | 14 |
|  | 4.3.6 HAZELCAST | 15 |
|  |  |  |
| **5.** | **SYSTEM DEVELOPMENT** | 16 |
|  | 5.1 MODULE DESCRIPTION | 16 |
|  | 5.1.1 QUEUES | 16 |
|  | 5.1.2 PARSER PROGRAM/JAXB PARSING | 16 |
|  | 5.1.3 HIBERNATE HAZELCAST CACHE | 19 |
|  | 5.1.4 DATABASE | 22 |
|  |  |  |
| **6.** | **SYSTEM DESIGN** | 23 |
|  | 6.1 ARCHITECTURE DIAGRAM | 23 |
|  | 6.2 GENERAL ARCHITECTURE | 25 |
|  | 6.3 UML DIAGRAMS | 26 |
|  | 6.3.1 ACTIVITY DIAGRAM | 26 |
|  | 6.3.2 COLLABORATION DIAGRAM | 28 |
|  |  |  |
| **7.** | **IMPLEMENTATION AND MAINTAINENECE** | 29 |
|  | 7.1 IMPLEMENTATION | 29 |
|  | 7.2 MAINTAINENECE | 29 |
| **8.** | **TESTING** | 30 |
|  | 8.1 INTRODUCTION | 30 |
|  | 8.2 CODE GENERATION TEST | 30 |
|  | 8.3 RUN TIME TEST | 30 |
|  | 8.4 SOURCE CODE TESTING | 31 |
|  | 8.5 SPECIFICATION TESTING | 31 |
|  | 8.6 MODULE LEVEL TESTING | 31 |
|  | 8.7 UNIT TESTING | 31 |
|  | 8.8 INTEGRATION TESTING | 31 |
|  | 8.9 VALIDATION TESTING | 32 |
|  | 8.10 RECOVERY TESTING | 32 |
|  | 8.11 SECURITY TESTING | 32 |
|  | 8.12 PERFORMANCE TESTING | 32 |
|  | 8.13 BLACKBOX TESTING | 33 |
|  | 8.14 OUTPUT TESTING | 33 |
|  | 8.15 USER ACCEPTANCE TESTING | 33 |
| **9.** | **PERFORMANCE MEASURES** | 34 |
| **10.** | **CONCLUSION** | 35 |
|  | **APPENDIX 1- SCREENSHOTS** |  |
|  | **REFERENCES** |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 5.1 | Module Description | 16 |
| 5.1.2 | Jaxb Architecture | 18 |
| 5.1.4 | Database Model | 22 |
| 5.1.5 | Technical Loading Process | 22 |
| 6.1 | Detailed Architecture | 24 |
| 6.2 | Process Flow | 25 |
| 6.3.1 | Activity Diagram | 27 |
| 6.3.2 | Collaboration Diagram | 28 |
|  |  |  |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ACRONYM** | **DESCRIPTION** |
| 1. | EFM | Enterprise Financial Messaging |
| 2. | OBI | Operational and Business Intelligence |
| 3. | DM | Data Marts |
| 4. | ETL | Extract, Transform, Load |
| 5. | MQ | Message Queueing |
| 6. | RT | Real Time |
| 7. | UML | Unified Modelling Language |
| 8. | OLAP | Online Analytical Processing System |
| 9. | US | United States |
| 10. | OFAC | Office of Foreign Assets Control |
| 11. | DNBW | Do Not do Business With |
| 12. | SWIFT | Society for Worldwide Interbank Financial Telecommunication |
| 13. | FED | Federal Department of United States |
| 14. | CHIPS | Clearing House Interbank Payment System |
| 15. | XML | eXtensible Markup Language |
| 16. | J2EE | Java Enterprise Edition |
| 17. | MVC | Model View Controller |
| 18. | JAXB | Java Architecture for XML Binding |
| 19. | DOM | Document Object Model |
| 20. | SAX | Simple API for XML |
| 21. | API | Application Programming Interface |
| 22. | XSLT | Extensible Stylesheet Language Transformation |
| 23. | XHTML | Extensible HyperText Markup Language |
| 24. | DTD | Document Type Definition |
| 25. | WORA | Write Once Run Anywhere |
| 26. | IDE | Integration Development Environment |
| 27. | JDT | Java Development Tools |
| 28. | JWSDP | Java Web Services Development Pack |
| 29. | FIFO | First-In-First-Out |
| 30. | POJO | Plain Old Java Objects |