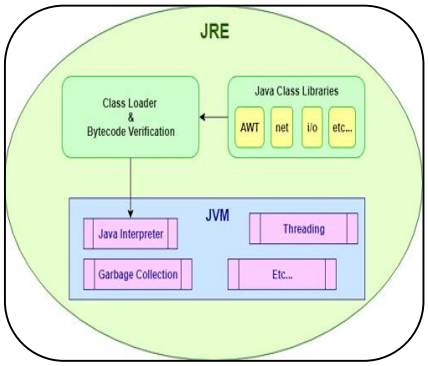
REVIEW OF RESEARCH
IMPACT FACTOR : 5.2331(UIF) UGC APPROVED JOURNAL NO. 48514 ISSN: 2249-894X
VOLUME - 7 | ISSUE - 9 | JUNE - 2018

RUNTIME ENVIRONMENT OF JAVA APPLICATIONS



Prof. Santosh Kulkarni
Assistant Professor,
Prin. K.P. Mangalvedhekar Institute of Management,
Career Development & Research , Solapur.

ABSTRACT

This paper presents an overseeing climate for the java virtual machine (jvm), which is beginning to be utilized in mission and wellbeing basic application, frequently with continuous prerequisites. The environment known as jvm monitor gathers information about the monitored virtual machine's state as well as its failures. In a jvm, the model can identify a low memory condition. Application memory management and application recovery tasks should make use of this low memory detection.

KEYWORDS: *Application Management, Java Virtual Machine, Application Profiling.*

INTRODUCTION

Each task aimed at determining a system's dependability requires a managing environment. Profiling and managing applications are essential components of any software system. Architecture applications must be monitored and profiled during the testing phase and the implementation phase, even if careful planning is done during the design phase, to ensure optimal application performance. The following are any management model's most essential requirements: It should use less memory, be safer, and have a low data rate. There should be no more than 1% performance overhead in the model. The monitor ought to have the capability of performing on-demand management.

Multiple clients should be able to use the managing model at the same time, according to the model. JConsole is utilized for managing the server-installed java virtual machine. The absence of real-time management, remote management, report generation, and many other benefits of JConsole are disadvantages. The inability to analyze the historical data is one of the main drawbacks. so that optimization of the server is impossible. The data are unsorted, making them difficult to retrieve as well as challenging to retrieve using remote applications.

The inconsistent management of the data and the delayed access to the data from the Java virtual machine reveal this issue. The data's consistency cannot be controlled by the manual system. As a result, the data are unreliable for server optimization research. Due to the data's unordered nature, the system administrator found it extremely challenging to analyze the data. Additionally, the data cannot be accessed quickly. The Java virtual managing system does not have a centralized data access system for managing historical data.

A common interface for managing and controlling Java applications is the goal of JMX. By letting you see and change what is happening behind the scenes, the JMX framework can make your applications easier to manage. Any JVM's data for the number of threads and heap memory can be captured and analyzed.

It is possible to set up an alerting mechanism through Yahoo Messenger, which enables the managing team to address issues early. There are three parts to the JVM monitor: a Data Collector that stores events and state snapshots in a database.es, a Local Monitor Daemon that receives data from such an agent and updates the state of the JVM, and a Managing Agent that collects data from the Monitored VM.

System Overview

A web application called "JVM Managing system using JMX" was developed to carry out the task of online data retrieval while incorporating security features. SQL queries are used to retrieve JVM data stored in a database in this JMX system. The JVM information are transferred from different clients subsequent to breaking down. For the purpose of analysis, these data are kept in a centralized database. A web application can be used to analyze the data and connect the same database to the remote system.

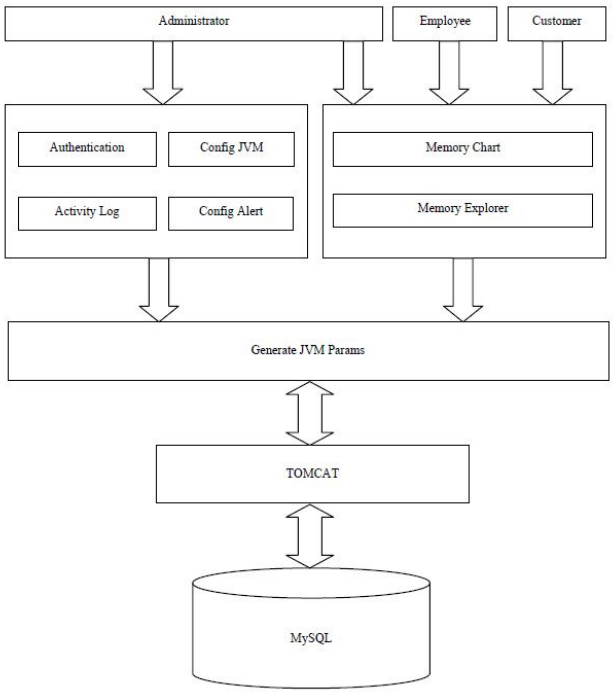


Figure 1. Architectural design of JVM monitoring system using JMX

System Characteristics

Online JVM data Management is one of the functions that this software product handles. so that data from the JVM Server can be obtained with just a few clicks.

The term "product function" asserts that a mathematical function of the product's input factors determines the maximum output of a technologically determined production process. The product function would only be comprised of the combinations that encompass a maximum output for a particular set of inputs if the set of all technically feasible combinations of output and inputs were taken into consideration. An alternative definition of a product function is the specification of the minimum input requirements required to produce specified output quantities using the technology that is available. Typically, it is assumed that distinct product functions can be developed for each product technology.

This product is planned in a manner with the end goal that a client can involve this web application in a simple way. A distributed database system makes it possible to work with this web application over a network. Thus, the software can be used simultaneously by multiple users.

It's a web-based application that works over the Internet. It can be made to work in more than one environment (network) and assigned to use the software simultaneously, provided that each module is assigned to a specific user. In addition, the user interface is designed with the use of MDI (multiple document interface), which makes it easier and simpler for a single person to manipulate all of the modules.

Modules

Six modules are utilized in the JVM Managing System Implementation Using JMX. The following are examples:

JVM Configuration

The entire system can be configured in this module, "JVM Configuration." The entire system will function in accordance with the configuration settings. Configuration options for the java virtual machine include register, delete, and view. It acquires the data and saves it to a database. The administrator has the ability to modify these configuration settings at any time.

Alert Configuration

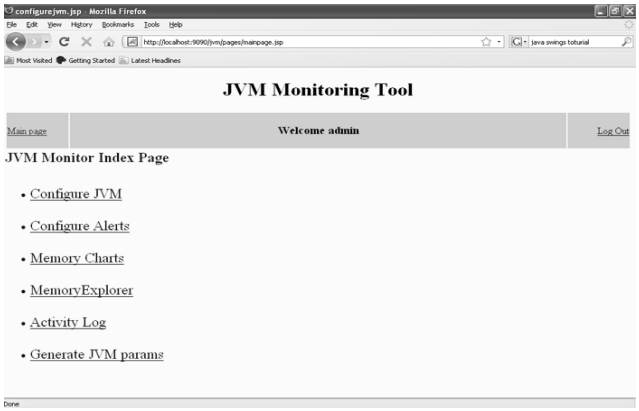
"Ready Setup" manages the different arrangements setting for cautioning the client as well as the manager with respect to the JVM cycle. Heap memory alert is one of the primary processes. The notification will be sent via email. Set alert and view alert are the submodules found here.

Memory Chart

Using a JFree chart, the "Memory Chart" module presents the memory consumption in a graphical format. Two kinds of management can be done here. Real-time management and history management are two examples. History overseeing is finished through the information accessible in the data set. This is used to investigate the various JVM behaviors.

Memory Explorer

Using a numerical data format, the "Memory Explorer" module presents the memory usage in a graphical representation. Two kinds of management can be done here. Real-time management and history management are two examples. History overseeing is finished through the information accessible in the data set. This is utilized to concentrate on the different way of behaving of JVM in the hypothetical way.



JVM Params

This is the core module of the JVM-based JMX-managed system. The background process known as JVM params retrieves the JMX-configured data from the client system. These data will be uploaded to the database by it.

Activity Log

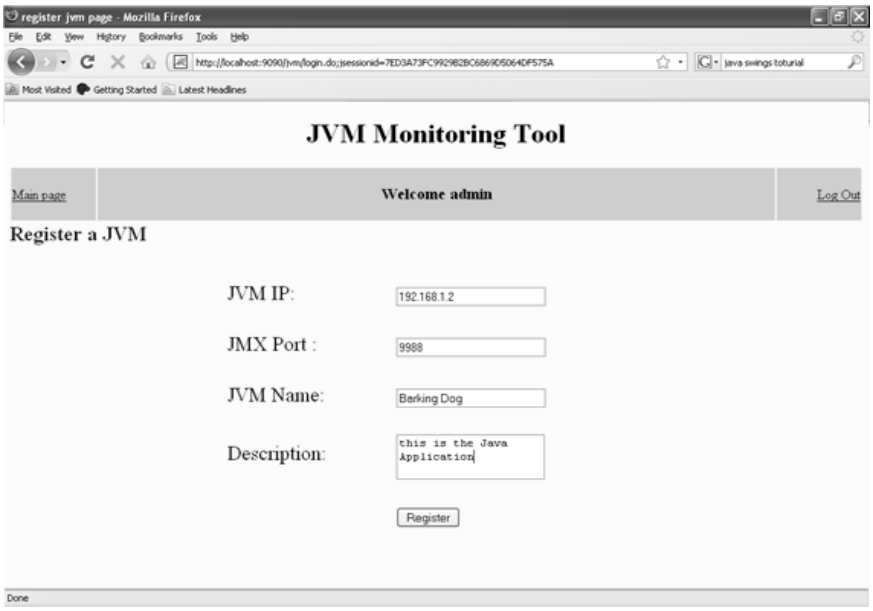
The entire system's log file can be found here. The timestamp for each activity is recorded in the activity database. Additionally, the error detail and unauthorized client intrusion can be analyzed using this activity log.

Authentication is required for all of the aforementioned modules. As a result, the login and password system is activated for security reasons.

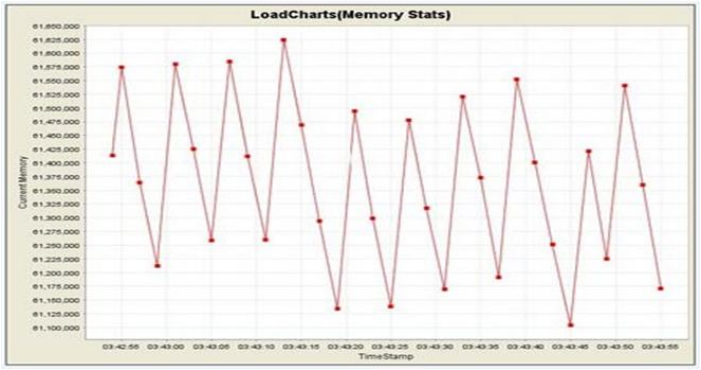
System Purpose and Scope

The creation of an automated system for managing the data of Java virtual machines is the primary objective of this project. Since the JVM load memory is of immense sum, data sets must be utilized for putting away the information and can be utilized for tutoring reason. Due to this execution, there will be a simple and reasonable admittance to the information for keeping up with the information also as the information can be recovered in a deliberate way which will accommodating for the framework directors for dealing with the server. Recent updated data are immediately accessible to end users because the data are stored in a centralized database.

This scope explains the software's limitations, like the requirement that the user meet the software's environment-based requirements. Therefore, the scope of this software is that the user must use it for the intended purpose. Date must be given in a substantial configuration to distinguish the specific clump. Because this software is based on the more recent user interface provided by the.net framework, it gives the user the best "look and feel" and allows them to understand the software even if they don't have much technical knowledge. They can also use the software without a manual.



JVM Configuration Page - Register



Memory Chart

CONCLUSION

Using Java Virtual Machine Managing, Distributed Managing and Reporting of Multiple Java Virtual Machines is built to assess the system's dependability. Profiling and managing applications are essential components of any software system. Architecture applications must be monitored and profiled during QA testing and production to ensure optimal performance, even if careful planning is carried out during design. A sophisticated managing system that provides insight into the operational details of any JVM application is provided by this project. It looks at how well the JVM works. It displays a memory chart with timestamps, used memory, and threshold values as well as records JVM failures and state changes.

This project can be utilized effectively in a variety of fields when packaged in specialized kits with advanced features. It is usable in J2ME-based applications. It can be utilized in JSR-3-compliant embedded devices. As the application runs, the level of debugging and logging can get better still. It can be used to connect JMX to your hardware, database, and application servers in order to track the environment's performance. We demonstrated that it can monitor a wide range of operational systems with a manageable overhead. This managing environment is currently being used in a campaign to evaluate the Java Virtual Machine's dependability in two ways: I) an estimation based steadfastness examination, focusing on the Java Virtual Machine with high Memory Bound, I/O Bound or central processor Bound jobs and gathering field information about its disappointments ii) a constancy benchmark, infusing programming deficiencies in JDK center classes, which cooperate with the virtual machine.

REFERENCES

1. Mukherjee A. and Siewiorek D. P., (1997), “Measuring software dependability by robustness benchmarking”, in the proceedings of IEEE Transactions on Software Engineering, vol. 23(6), pp. 366–378.
2. Smith J. E. and Nair R., (2005), “The architecture of virtual machines” ,in the proceedings of IEEE Computer, vol. 38(5),pp. 32–38.
3. Bressoud T. and Schneider F., (1995), “Hyper visor-Based Fault Tolerance”, in the proceedings of the 15th ACMSymposium on Operating System Principles, pp. 1–11.
4. M. Poel, R. den Akker, A. Nijholt, and A. J. van Kesteren, “Learning emotions in virtual environments,” in Proc. 16th Eur.Meeting Cybern.Syst. Res., R. Trappl, Ed., 2002, vol. 2, pp. 751–756.
5. K. Doya, “Meta-learning and neuro-modulation,” Neural Netw., vol. 15, no. 4–6, pp. 495–506, 2002.
6. Deivamani et al, “Agent-Based Learning Approach for Analyzing the Human Emotions” National Conference and FirstPhase of International Journal of Machine Intelligence & its Applications, 2010.
7. K. Abu Maria and R. Abu Zitar, “Emotional agents: A modeling and an application,” Inf. Software Technol., vol. 49, pp.695–716, 2007.
8. M. I. Gobbini and J. V. Haxby, “Neural systems for recognition of familiar faces,” Neuropsychologia,

vol. 45, pp. 32–41,2007.
9. T. Baumgartner, M. Esslen, and L. Jancke, “From emotion perception to emotion experience: Emotions evoked by picturesand classical music,” Int. J. Psychophysiol., vol. 60, pp. 34–43, 2006.