A FRAMEWORK FOR MANAGING RUNTIME ENVIRONMENT OF JAVA APPLICATIONS

T.VENGATTARAMAN[*]

Department of Computer Science, Pondicherry University,
Puducherry, India.

A.RAMALINGAM

Department of MCA, Sri Manakula Vinayagar Engineering College,
Puducherry, India.

P. DHAVACHELVAN

Department of Computer Science, Pondicherry University,
Puducherry, India.

Abstract

This paper presents a managing environment for the java virtual machine (jvm), which is starting to be employed in mission and safety critical application, often with real-time requirements. This environment, named jvm monitor, collects data about both the state and the failures of the monitored virtual machine. The model has the ability to detect a low memory condition in a jvm. This low memory detection should be used to do application recovery tasks and application memory management tasks.

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

1. Introduction

A managing environment is a key component in each task, aimed at evaluating the dependability of a system. Application managing and profiling are crucial parts of any software system. Even if careful planning is done during design phase, architecture applications need to be monitored and profiled during testing phase and Implementation phase, to make sure that the performance of the application is optimal. The most important requirements of any managing model are; it needs to be less in memory and more secure and have a low rate of data. The model should not have more than 1% performance overhead. The monitor should provide the facility to do on demand managing. The model should allow for multiple clients to concurrently use the managing model.

For managing the java virtual machine which is installed in the server, JConsole is used. The drawbacks of using JConsole are no real time managing, no remote managing, no report generation and much more. One of the main drawbacks is the historical data cannot be analyzed. So that server optimization cannot be done. The data are unordered and difficult for retrieving as well as difficulties for retrieving through remote applications.

This problem is recognized in the delayed access of the java virtual machine data and inconsistency in managing the data. The manual system has no control over the data for its consistency. So the data are not reliable for carrying out a research work for server optimization. The system administrator found very difficult in analyzing the data since the data are unordered. And also there is no quick access to the data. There is no centralized data access system for managing the historical data of java virtual managing system.

The purpose of JMX is to provide a standard interface for managing and controlling Java applications. The JMX framework can improve the manageability of your applications by letting you see and adjust what is going on in the background. The data for heap memory and thread count for any JVM can be captured and can be used for analysis. Alerting mechanism through Yahoo messenger can be set up, which helps the managing team to work on the issues proactively. JVM monitor is constituted by three components: a Managing Agent which collects data from the Monitored VM, a Local Monitor Daemon that receives data from such agent and updates the state of the JVM and a Data Collector, which stores events and state snapshots in a database.es.

2. System Overview

"JVM Managing system using JMX" is a web application derived to perform the task of online data retrieval with the implementation of security. Here in JMX system, JVM data are stored in a database and retrieved using SQL queries. The JVM data are uploaded from various clients after analyzing. These data are stored in a centralized database for analyzing. The same database can be connected to the remote system and the data can be analyzed using a web application.

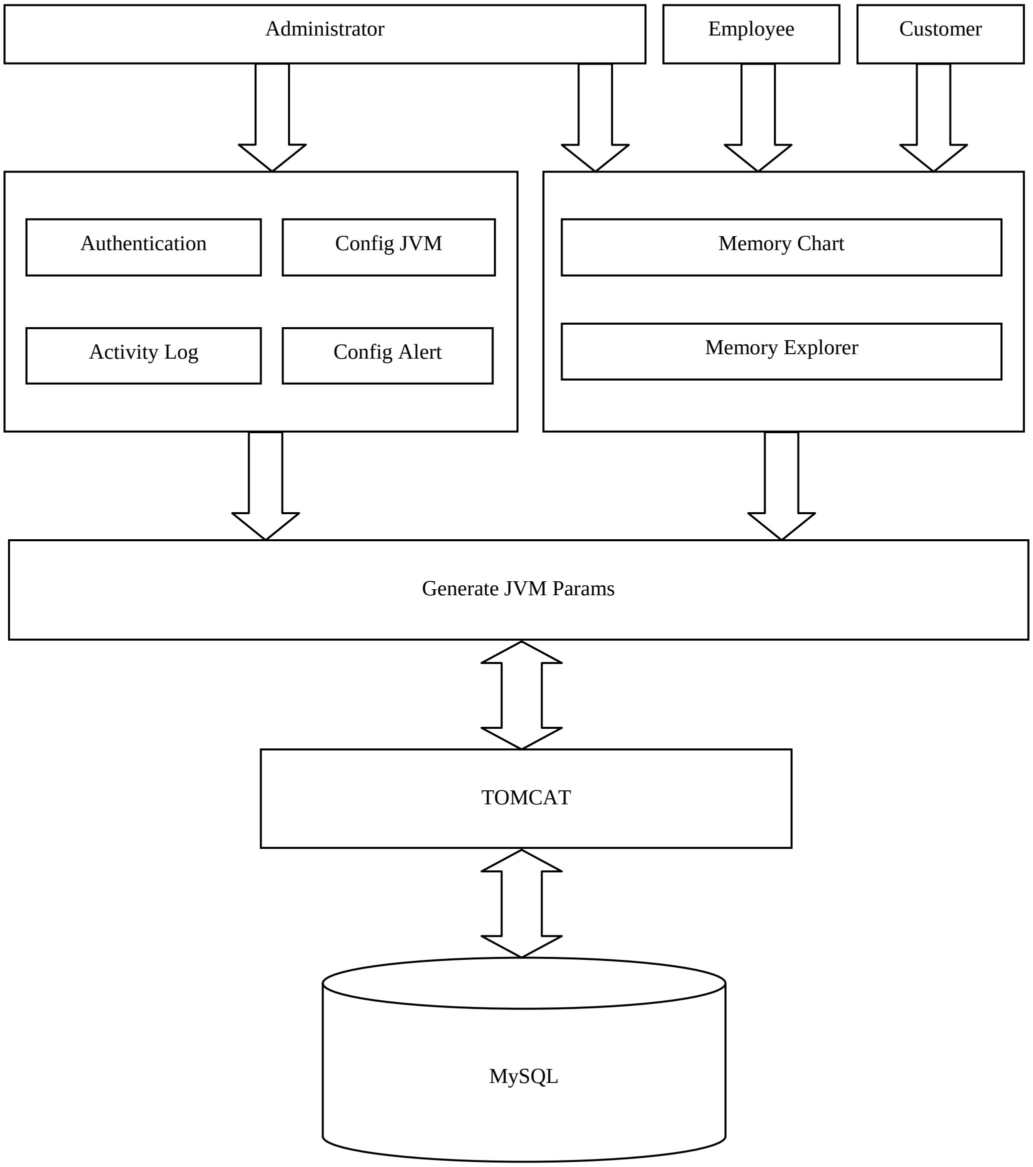


Figure 1. Architectural design of JVM monitoring system using JMX

2.1 System Characteristics

This software product deals with the functions such as online JVM data Management. So that JVM Server data can be obtained with simple c01licks.

+ monitoringmo
+.om

Product function asserts that the maximum output of a technologically-determined product process is a mathematical function of input factors of product. Considering the set of all technically feasible combinations of output and inputs, only the combinations encompassing a maximum output for a specified set of inputs would constitute the product function. Alternatively, a product function can be defined as the specification of the minimum input requirements needed to produce designated quantities of output, given available technology. It is usually presumed that unique product functions can be constructed for every product technology.

This software is designed in a way such that a user can use this web application in an easy manner. And also this web application can be manipulated in a network environment as this uses a distributed database system. So that, multiple users can also use the software simultaneously.

It is a web application that can be operated with a Internet. It can be extended to work in a multiple environment (network) and can be allotted for using the software simultaneously provided every module must be allotted for the particular user apart from this the user interface as be designed with the use of MDI (multiple document interface) which makes it easier and simple for a single person to manipulate all the module.

2.2 Modules

In Implementing JVM Managing System using JMX, 6 modules are used. They are given bellow:

2.2.1 JVM Configuration

Here in this module, “JVM Configuration”, the entire system can be configured. Depending upon the configuration settings the whole system will work. Settings such as register, delete, and view java virtual machine can be configured. It get the data and stored in database. These configuration settings can be changed by the administrator at any time.

2.2.2 Alert Configuration

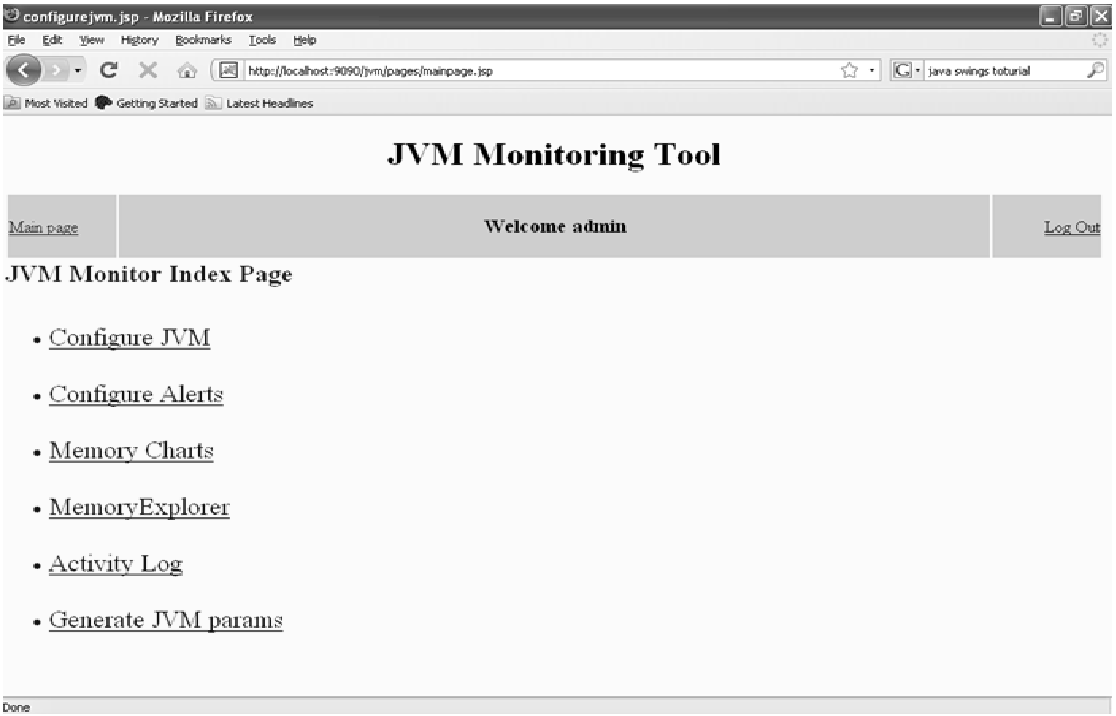
“Alert Configuration” deals with the various configurations setting for alerting the client as well as the administrator regarding the JVM process. One of the main processes is heap memory alert. The alert will be send as a mail. Here the sub modules are set alert and view alert

2.2.3 Memory Chart

The “Memory Chart” module displays the memory usage in a graphical representation format using JFree chart. Here two types of managing can be done. One is the real time managing and another is history managing. History managing is done through the data available in the database. This is used to study the various behavior of JVM

2.2.4 Memory Explorer

The “Memory Explorer” module displays the memory usage in a graphical representation using numerical data format. Here two types of managing can be done. One is the real time managing and another is history managing. History managing is done through the data available in the database. This is used to study the various behavior of JVM in the theoretical manner.


configurejvm.jsp - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9090/jvm/pages/mainpage.jsp
java swings toturial
Most Visited Getting Started Latest Headlines
Done

JVM Monitoring Tool

Main page | Welcome admin | Log Out

JVM Monitor Index Page

- Configure JVM
- Configure Alerts
- Memory Charts
- MemoryExplorer
- Activity Log
- Generate JVM params

Fig. 2. JVM Monitor Index Page

2.2.5 JVM Params

This is the core module of JVM managing system using JMX. JVM params is the background process that gets the data from client system that are configured in the JMX configuration. It will upload these data to the database.

2.2.6. Activity Log

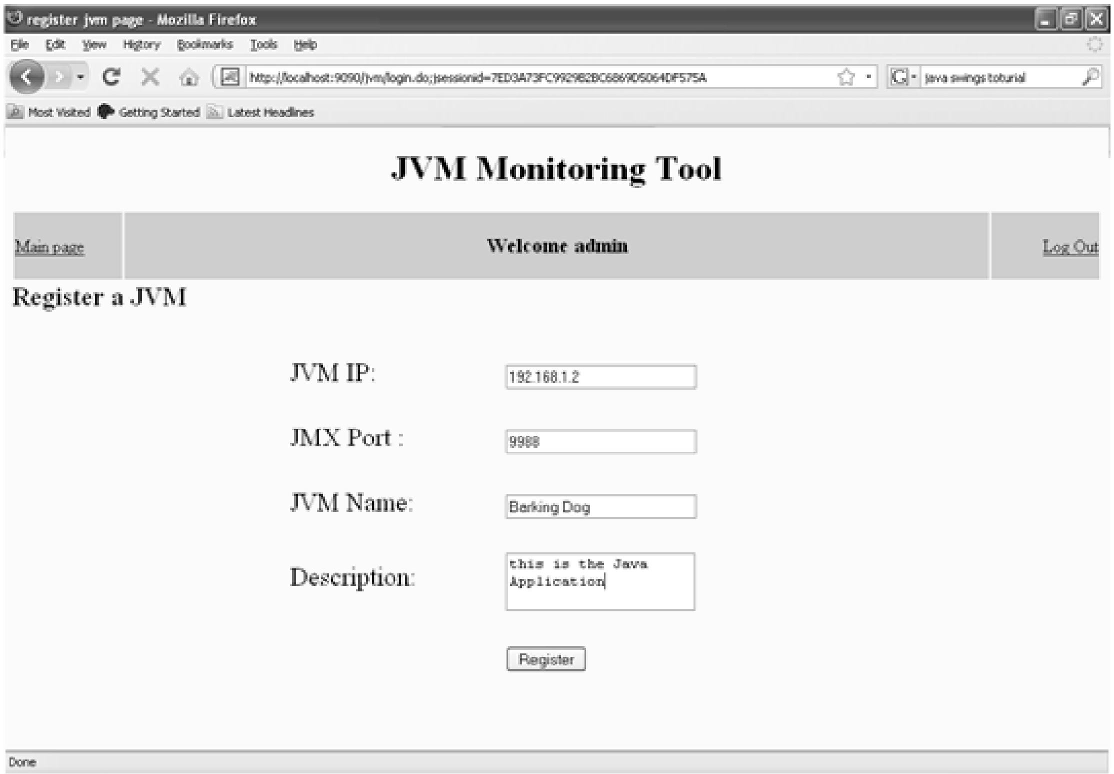
This is the log file of the entire system. Each and every activity are logged into the activity database with timestamp. This activity log can be used for analyzing the error detail and unauthorized client intrusion also.

For all the above modules, authentication is a mandatory. So login and password system is enabled for a security purpose.

3. System Purpose and Scope

The primary purpose of this project is to provide an automated system for managing the java virtual machine data. Since the JVM heap memory is of huge amount, databases have to be used for storing the data and can be used for mentoring purpose. Because of this implementation, there will be an easy and manageable access to the data for maintaining the data as well as the data can be retrieved in an orderly manner which will helpful for the system administrators for managing the server. Since the data are stored in a centralized database, recent updated data are immediately available to the end users.

This scope describes the limitations of this software such as the requirement that the user have to fulfill the requirement of the software based on the environment. So the scope of this software is that the user has to use this software based on the purpose of it. Date have to be provided in a valid format to identify the particular batch. As this software is based on the newer interface provided by the .net framework to make the user to best feel the user interface design, so that user feels the best “look and feel” to make the user to understand the software with very little technical knowledge and also one can understand the use of the software without software manual.


register jvm page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:9090/jvm/login.do;jsessionid=7ED3A73FC9929B2BC6869D5064DF575A
java swings toturial
Most Visited Getting Started Latest Headlines
Main page
Log Out
Register
Done

JVM Monitoring Tool

Welcome admin

Register a JVM

JVM IP: 192.168.1.2

JMX Port : 9988

JVM Name: Barking Dog

Description: this is the Java Application

Fig 3. JVM Configuration Page - Register

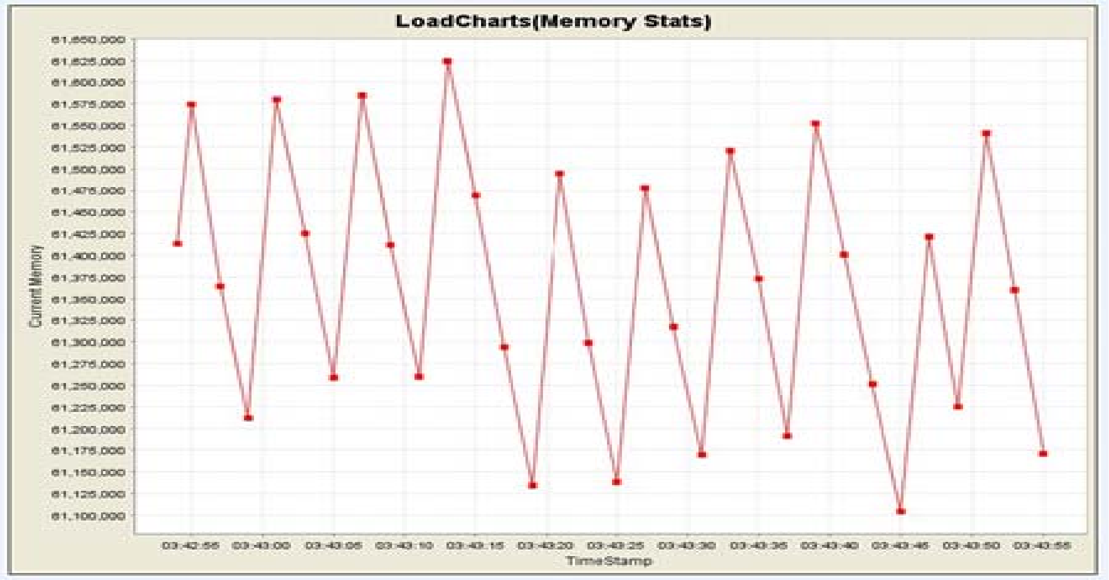


Fig. 4 Memory Chart

4. Conclusion

Distributed Managing and Reporting of Multiple Java Virtual Machines is constructed with the Java Virtual Machine Managing which is aimed to evaluate the dependability of the system. Application managing and profiling are a crucial part of any software system. Even if careful planning is done during design, architecture applications need to be monitored and profiled during QA testing and production to make sure that the performance of the application is optimal. This project provides a sophisticated managing system which gives insight into the operational details of any JVM application. It evaluates the performance of the JVM. It monitors and captures JVM failures and the evolutions of the state and displays memory chart with timestamp and used memory and also for threshold values.

In specialized Kits with advanced features then this project can be effectively used in the various fields. It can be employed in J2ME applications. It can be used in embedded devices compliant to JSR-3. The level of debugging and logging can be improved further as the application runs. It can be used for interfacing JMX with your hardware, database, and application servers to monitor the performance of the environment. We showed that it is capable of monitor a wide class of operational systems keeping an acceptable overhead. Currently this managing environment is being employed in a dependability assessment campaign for the Java Virtual Machine which spans into two directions: i) a measurement based dependability analysis, stressing the Java Virtual Machine with high Memory Bound, I/O Bound or CPU Bound workloads and collecting field data about its failures ii) a dependability benchmark, injecting software faults in JDK core classes, which interact 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 ACM Symposium 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 First Phase 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 pictures and classical music,” Int. J. Psychophysiol., vol. 60, pp. 34–43, 2006.