Intelligent Asset Management Achieving Autonomy Using Alchemi Middleware

Ms. Sanam Kagde
Assistant Professor, Information Technology,
MHSSCOE
Mumbai

Mr. Shrinidhi Gindi
Assistant Professor, Information Technology,.
MHSSCOE
Mumbai

Mr. Shaikh Mohd. Faiz
Student, Information Technology,
MHSSCOE
Mumbai

Ms. Nazmeen Shaikh
Student, Information Technology,
MHSSCOE
Mumbai

Abstract **— While the IT industry is moving forward with the advent of recent technologies, IT assets management is semi automated and unmanaged. Though lots of research is carried on and existing asset management softwares have become intelligent by integrating it with mining and Artificial Intelligence but as manual operation is required to manage the system it leads to poor efficiency and security. Self-management of resources can make the system up and running at full pace most of time. Autonomic computing provides such an approach by enabling the design and development of Systems that can adapt themselves to meet the requirements without manual intervention.**

In this paper we address technical challenges by making the asset management software self-protecting, self-configuration, self-healing and self-optimization. We have proposed design and implementation of a system that manages assets of an organization without the human intervention with the use of middleware Alchemi. Agent environment of Alchemi is responsible for resources location and allocation, authentication, unified information access, communication, task assignment, agent library and others thereby providing intelligence to Alchemi middleware thus helping it to exhibit autonomic behavior which is becoming necessity of assest management systems day by day.

Keywords— Alchemi; asset management; autonomic computing; security

I. INTRODUCTION

An IT Asset is any data, device, or other component of the environment that supports information-related activities. Assets generally include hardware (e.g. servers and switches),
software (e.g. mission critical applications and support systems). The traditional lifecycle approach of asset is to submit a new request for equipment, get management approvals, submit to procurement where they do the price negotiations and normally from a pre-selected vendor list and then the PO is sent to the vendor. Like this the
whole life cycle is handled. The system is very time consuming and lazy. Again all the hardware,
software and their related documents e.g. invoices, manuals, licenses are scattered everywhere which is difficult to manage. This system is more prone to errors and sometimes the approach to various problems is unstructured. IT asset involves gathering detailed hardware and software inventory information which is then used to make decisions about hardware and software purchases and redistribution. IT inventory management helps organizations manage their systems more effectively and saves time and money by avoiding unnecessary asset purchases and promoting the harvesting of existing resources. IT asset management (ITAM) is the set of business practices that join financial, contractual and inventory functions to support life cycle management and strategic decision making for the IT environment. Organizations that develop and maintain an effective IT asset management program further minimize the incremental risks and related costs of advancing IT portfolio infrastructure projects based on old, incomplete and/or less accurate information. Alchemi is a .NET grid computing framework that allows you to painlessly aggregate the computing power of intranet and Internet-connected machines into a virtual supercomputer (computational grid) and to develop applications to run on the grid. It has been designed with the primary goal of being easy to use without sacrificing power and flexibility. The research enables IT asset management software to be self-protecting, self-configuration, self-healing and self-optimization.

II. IDENTIFYING THE VARIOUS PROBLEMS NECESSITATING A COMPREHENSIVE IT ASSET MANAGEMENT STRATEGY

Fortunately, the problems necessitating the implementation of a comprehensive IT asset management strategy are fairly easy to identify. Unfortunately, however, far too many

organizations cannot answer the following questions with an acceptable degree of certainty:

- How many PC‟s is my organization responsible for?
- Where are they currently located?
- What software applications currently reside on each PC?
- Are any of these unauthorized, or illegal copies?
- How is each PC configured (processor, memory, etc.)?
- Which IT assets are leased?
- Which IT assets have been purchased?
- Which departments are responsible for which assets?
- Can IT assets (HW) currently not being used, be redeployed? If so, who needs them?
- Can software licenses residing on unused PC‟s be reclaimed?
- Are we purchasing upgrades for IT assets currently not in use?
- Are we purchasing annual maintenance and support for these unused assets?

It is usually the case that organizations that cannot answer any or all of the above questions are experiencing problems in both the IT department, as well as in other departments responsible for compiling and reporting accounting and financial data regarding the IT assets.
Typically, the multi-departmental problems resulting from not having the up to date information outlined above will materialize in several areas. These usually will include:

- Forecasting and budgeting problems.
- Inability to pass required audits.
- Inability to provide accurate and meaningful management reports.
- Purchasing problems.
- Upgrade problems.
- Problems tracking specific assets linked to specific contracts and agreements.
- “MAC” (moves/adds/changes) problems.
- Reutilization problems.
- Inability to develop an accurate migration strategy.

III. RISKS TO EFFECTIVE ASSET MANAGEMENT

Security risk is often, quantitatively, represented as any event that compromises the assets, operations and objectives of an organisation. There are at least five such risks that primarily contribute to an organization’s failure to optimally manage their assets:
1) Lack of Intelligence to the system
2) Improper maintenance;
3) Improper manual operation;
4) Improper risk management;
5)Lack of Self Healing and Self-Configuration

IV. AGENT FRAMEWORK FOR ALCHEMI

Agent environment is the kernel of Grid computing which is responsible for resources location and allocation, authentication, unified information access, communication, task assignment, agent library and others. Agent based technologies have become more and more widespread during the last few years and the number of practical applications has noticeable increased, especially in the new Grid Computing paradigm. Agents and Grid both need each other. Grid computing can be seen as a multi-agent system facilitating the sharing of compute resources, allowing users to discover and use remote resources. Users are able to submit jobs to remote resources and typically have noexplicit control over the resources themselves. [6] Thus, both users and resources can be viewed as autonomous agents, having control of their own behavior. This autonomy gives rise to inherent uncertainty, since an individual cannot predict how another will respond to changing situations.

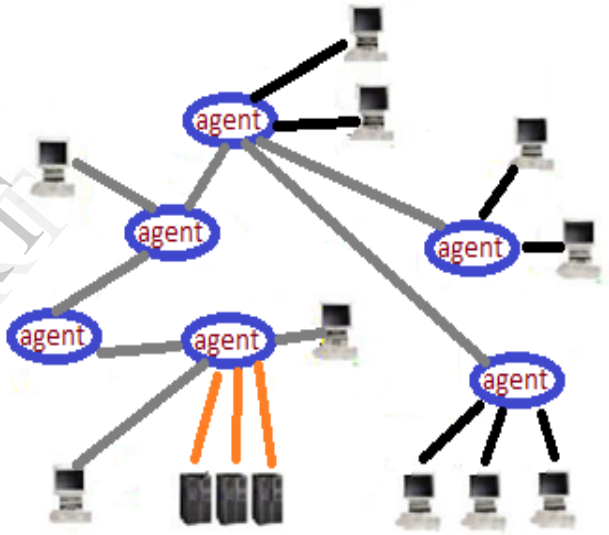
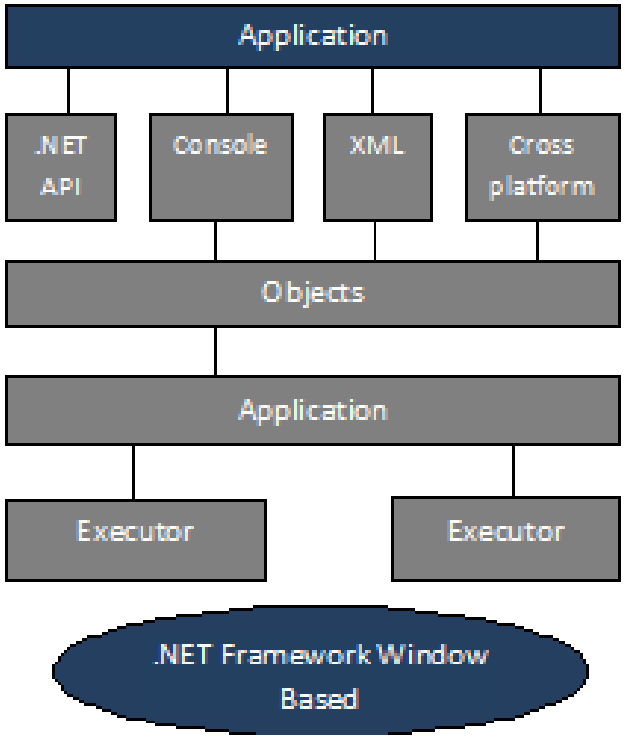


Figure: Agent framework for Alchemi.

V. PROPOSED SYSTEM

In this paper we implement an intranet using Alchemi middleware and execute the application using .Net framework. An intranet is created by installing Executors each machine that is to be part of the intranet and linking them to a central Manager component. We execute the application developed in a .Net framework when the intranet is established .Thus by implementing the architecture of Alchemi we can observe and study its autonomic behaviour. Also the grid implementation using Alchemi helps us to implement agent framework and map the different agents with Alchemi middleware. The run time components of Alchemi middleware uses .Net framework for execution of grid applications. The different nodes are created as Executors which are connected to a central Manager node. The user application created has API, console interface, Xml representation of data which interact

with .Net objects created for the application.(Request object, Response object etc) as shown in figure below.



VI. IMPLEMENTATION

A. *Configuration of Manager and Executor nodes*

We configure the manager and executor set up through installer interface. In this step we have to provide the installer of manager and executor node with port numbers of each, password for authentication, mode information (Dedicated or non-dedicated).

B. *Start the Manager node*

After providing the details we run the installer for Manager on port no 900.The Manager node is started and the master slave architecture is deployed for the asset.

C. *Connecting Executor nodes to Manager nodes*

After starting the manager node we connect the Executor node to the Manager node using the
Connect option. Now our Grid nodes are connected to each other and Grid environment is enabled. The different assets are available on executor node and are under the control of manager since the system is deployed using master slave configuration.

D. *Successful execution of the code or application*

After the system is deployed, the user can start viewing all the available assets. If the user requires a particular asset he will make a request and then the manager will assign an executor to the request to execute the request. The main benefit here is that we will achieve the benefits of Alchemi like self-protecting, self-configuration, self-healing and self-optimization. The user need not install the software assets on their system as our application will provide it to them on demand. It will save the hardware and software requirements on the clients system. The user can perform a variety of operations like adding a new asset, deleting them, as well as viewing the available assets. Also our application will give out alerts to the user if any asset is due to renew.

VII. RESULT ANALYSIS

The benefit of using Alchemi is that it provides us with the following four important benefits in our project which are responsible for making our project autonomic in nature:

1. Self-Protection: Self-protecting capabilities that can detect, identify, and defend itself against viruses, unauthorized access, and denial-of-service-attacks. If a node is about to fail then it will try it's best to protect itself from various errors and security breaches. It will be autonomic in nature and won't require human intervention.

2. Self-Configuration: Self-configuring capabilities that enable our system to adapt to unpredictable conditions by automatically changing its configuration, such as adding or removing new components, or installing software changes without disrupting service. Whenever a new asset is added in our system it will configure itself automatically by communicating with the manager. It will be very useful since we won't have to manually configure each and every asset or node.

3. Self-Healing: Self-healing capabilities that can prevent and recover from failure by automatically discovering, diagnosing, and recovering from issues that might cause service disruptions. A self-healing system automatically detects, diagnoses and repairs localized hardware and software problems. Thus, we expect our system to perform runtime reconfigurations and repairs of its components.

4. Self-Optimizing: Self-optimizing capabilities that enable the system to continuously tune itself-proactively to improve on existing processes and reactively in response to environmental conditions.

VIII. CONCLUSION

The Intelligent Asset Management achieving Autonomy using Alchemi Middleware software is very useful as it provides automation since we make use of Alchemi Middleware. It overcomes the various disadvantages of manual and semi-automated systems. It also helps in managing the various risks associated with asset

management. Agent environment of Alchemi is responsible for resources location and allocation, authentication, unified information access, communication, task assignment, agent library and others thereby providing intelligence to Alchemi middleware thus helping it to exhibit autonomic behavior which is becoming necessity of assest management systems day by day.

REFERENCES

[1] Akshay Luther, Krishna Nadiminti, Rajkumar Buyya, Alchemi: A .NET-based Enterprise Grid System and Framework by July 2005. Url: http://www.gridbus.org/~alchemi/files/1.0.beta/docs/AlchemiManual v.1.0.htm
[2] http://www.flashmagazine.com/news/detail/alchemy_toolkit_-_preview_version_out_on_adobe_labs/
[3] The University of Melbourne,Australia, Dept. of Computer Science and Software Engineering, Alchemi: A .NET-based Enterprise Grid Framework:,, url: www.gridbus.orwww.alchemi.netg /
[4] http://msdn.microsoft.com/us/library/7sxk9h.aspx
[5] John E. Provo, IT Asset Management - an Integral Part of an Overall Asset Management Strategy.
[6] Ismail Ari, Jun Li, Jhilmil Jain, Alex Kozlov, Management of Data Mining Model Lifecycle to Support Intelligent Business Services,, April 24, 2008.
[7] H. Li, X. Liu, W. Sun, Analytics-Driven Asset Management, IBM Journal of Research and Development, January/March 2011
[8] Grahame Fogel and Stefan Terblanche, Why Asset Management Doesn't Work in the Mining Industry and How to Fix It.
[9] ISO 55000 (Draft): Asset Management – Overview, principles and terminology.