Software Installation on a Network Using Mobile Agents

Sandeep D Tukkoji
Department of Computer Science
B.V.B College of Engineering & Technology
Hubli, India .
Email: sandeeptukkoji@gmail.com

Chetana D. Tukkoji
Department of Information Science
Atria Institute of Technology
Bangalore, India.
Email: chetana20.kle@gmail.com

ABSTRACT:

Software Installation (or software setup) is the process of restoring software from external media and putting the program onto a computer system that writes the necessary data on to local file system (hard drive) so that it can be executed. Typically, software installation is done at each node manually either by running the setup from a removable media (CD-ROM, flash drive etc.) or through a client–server model in a network wherein the client requests and downloads the setup file from the server (or by simply sharing the files in a network and directly executing at remote node). Consequently, traditional approach of software installation cannot scale well because of its time-consumption properties. In order to solve this problem and provide an efficient solution, this work proposes a system that makes use of Mobile Agent paradigm to perform software installation automatically thus eliminating human intervention at node's on the network.

Keywords: Software Deployment, Software Installation, Mobile Agents, Security.

I. INTRODUCTION

Personal computers now run a wide range of applications each serving one or more users. Each application to get executed on the system requires it to be installed; that involves writing the necessary data on system drive for running the program. Installing the required application using traditional software installation on a single node or on few is not a difficult task but when required to install on several node"s (e.g. Networked computers in Educational Institutions, Organizations etc) is boring and time consuming job.

The following are the disadvantages of traditional approaches

- It requires lot of human intervention.
- It is time consuming process.
- It requires additional physical media resources like CD-ROM, Flash drive etc and
- It is difficult to manage Applications over PC"s.

This paper is organized as follows: Section 2 discusses the related works, and technologies in software deployment process. Section 3 provides an introduction to mobile agents and methodology for the proposed system. Section 4 presents the system requirements. Section 5 concludes and presents the future work.

II. LITERATURE SURVEY

The software deployment is a complex procedure that involves the release, installation, adaptation, reconfiguration, update, activation, deactivation, retirement and removal of software.

The following are some of the existing technologies [1] that support various aspects and activities of the software deployment process.

A. Installation Tools

Generate a set of executable and artifact files that must be downloaded or provided via a distribution media like a CD, or via Internet. These files can be compressed in a single installation executable file. In order to install the system, the installation software needs to be executed in the customer site.

One example of these tools is the InstallShield installation tool that generates a self-extractor file that manages the installation and de-installation procedures of an application. This Install Shield is for software developer to build reliable Windows Installer (MSI) and Install Script installations for desktop, server, Web, mobile applications [2] and not to install a software program on computers.

Disadvantages: These tools are usually platform specific, and allow a minimal degree of configuration on what components to install.

B. Package Managers

These installation tools use the concept of packages. A package is an archive that contains the files that constitute a system together with some meta-data describing the system. The packages have to be copied to the customer site in order to be installed. Some package managers provide file transfer capabilities. Once in the system, the package is installed by the package manager. An example of a package tool is the Red Hat RPM.

Disadvantages: These tools usually do not provide activation and deactivation capabilities and some of them allow a primitive policy specification.

C. Application Management Systems

These kinds of systems generally support all of the life cycle activities except the producer-side release activity. Their architecture is generally centralized. These systems were designed to manage the software deployment in large or medium organizations that both produce and consume software. In these systems, a central server typically controls all management and deployment activities. An example of such application is the System View from IBM.

Disadvantages: These technologies don"t allow both the transportation and execution of the installation software in a more customizable way.

In reference [1], although no implementation is provided in this paper some ideas concerning the use of mobile agents in software deployment process, issues and requirements, benefits and drawbacks are discussed. Programming Mobile agents to request the installation files according to current configuration of system instead of getting all installation files at once is one idea and the workflow management using mobile agent paradigm is perhaps basic idea to our work. The paper also advocates the use of interpreted languages like java to achieve platform independence. The author suggests a Hybrid policy to overcome lower internet speeds. In our system node's are over LAN and hence one time transfer of payload is required.

The reference [3] discusses some previous works that focus to deal software deployment problems in clusters.

I. Disk image-based deployment

Disk image-based deployment approach was designed for deploying operating system. System Imager provide tools for acquirement, management and deployment of system images. The deployment of image is centralized and all node's obtain image from the image server by using the remote replication tool rsync. OSCAR (Open Source Cluster Application Resource) composed of Software Suite which can be easily integrated and installed, provides a set of construction, management, maintenance and high-performance cluster tools.

Disadvantages: Poor Flexibility

II. ehavior-based deployment

The basic idea is to record all of the disk operations during the software deployment and the software installation process can be replayed. It is required to trace the kernel operation of installation process. ZSDT (Zju Software Delivery Toolkit) can automate software deployment in all aspects, including planning, implementation, reporting.

Disadvantages: Tracing the kernel operation of installation Process is difficult to realize.

III. ***Package-based deployments***

XcelleNety use Prism Deploy for Software distribution. Prism Deploy create packages which contain State information including file system, registry settings, and other information. Prism Deploy also supports the distribution of Msi package format. This uses the idea of batch process. It writes all the commands that are implemented during the installation process in a batch file.

When deploying software on the target node, we should send the installation package and batch file to the target, and then execute the batch commands. This method is flexible.

Disadvantages: Hard to solve the interactive installation, because some software needs to interact with the administrator/user during installation process.

In paper [3], the method "NOVEL SOFTWARE DEPLOYMENT SYSTEM" focuses on how to deal with deploying software automatically in large scale Linux based clusters. The author feels that his method improves Package Based Deployment methods which are highly flexible. Phases included in deployment process are clearly discussed. The Linux installation processes are identified and handled accordingly. Interactive installation procedure which is a complex task is achieved through Script and Expect tools. This proposed method for Linux based systems is closest to our work. The author also mentions his method lacks stability and there are some random factors, so when deploying software by using the automatic deployment system, there may be individual failure.

The reference [4], describes a system that performs software installation on a heterogeneous network using mobile agents which is closest to our own work. This system is suited only for installation of mandatory software: which means only similar software can be installed on entire network. Individual node's over LAN cannot be handled. Two separate Mobile agents Agent Controller and Agent are used for software installation. Creating N clones of Agent Controller for N entries in Inter tablets and M Agents for M entries in Intra table increases network traffic and gets worse if the installation software or node's in network is large. We believe our proposed system will be feasible approach for installing required software on windows based node or network.

III. PROPOSED ARCHITECTURE

A. Motivation

The disadvantages of existing systems and the advantages of Mobile Agents [5] motivated us to carry on with this work. The proposed architecture develops a system that performs the required Software installation on a windows based node or network which is an activity of Software Deployment that makes software systems available for use.

B. Mobile Agent

Mobile agents refer to programs that perform certain tasks on behalf of the user and migrate from one computer to another in the network and execute on several machines. A mobile agent is a specific form of mobile code that consists of the program code and the program execution state (the current values of variables, next instruction to be executed, etc.)

Initially a mobile agent resides on a computer called the home machine. The agent is then dispatched to execute on a remote computer called a mobile agent host (a mobile agent host is also called mobile agent platform or mobile agent server). When a mobile agent is dispatched the entire code of the mobile agent and the execution state of the mobile agent is transferred to the host. The host provides a suitable execution environment for the mobile agent to execute. The mobile agent uses resources (CPU, memory, etc.) of the host to perform its task. After completing its task on the host, the mobile agent migrates to another computer. Since the state information is also transferred to the host, mobile agents can resume the execution of the code from where they left off in the previous host instead of having to restart execution from the beginning. This continues until the mobile agent returns to its home machine after completing execution on the last machine in its itinerary.

The life cycle of a mobile agent:-

1. The mobile agent is *created* in the Home Machine.
2. The mobile agent is *dispatched* to the Host Machine A for execution.
3. The agent executes on Host Machine A.
4. After execution the agent is *cloned* to create two copies. One copy is dispatched to Host Machine B and the other is dispatched to Host Machine C.
5. The cloned copies execute on their respective hosts.
6. After execution, Host Machine B and C send the mobile agent received by them back to the Home Machine.

7. The Home Machine *retracts* the agents and the data brought by the agents is analyzed. The agents are then *disposed*.

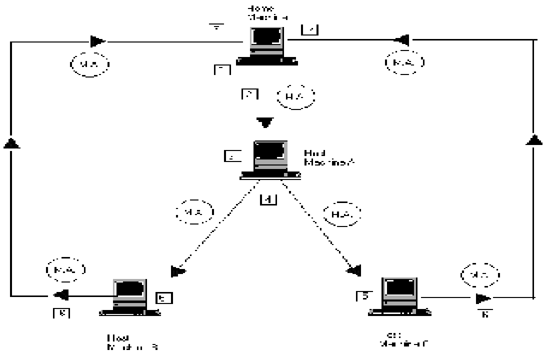


Figure 1.The life cycle of a mobile agent.

From the Figure 1 .We observe that a mobile agent experiences the following events in its life cycle:

Creation: a brand new agent is born and its state is initialized.

Dispatch: an agent travels to a new host.

Cloning: a twin agent is born and the current state of the original is duplicated in the clone.

Deactivation: an agent is put to sleep and its state is stored on a disk of the host.

Activation: a deactivated agent is brought back to life and its state is restored from disk.

Retraction: an agent is brought back from a remote host along with its state to the home machine.

Disposal: an agent is terminated and its state is lost forever.

C. Methodology

The proposed architecture makes use of Aglets Software Development Kit (ASDK) 2.0.2 which is a Java based framework and environment for researching and developing mobile agents in JAVA to develop this system. We do require a one-time installation of the ASDK at each node in the network and network connection, at least during the time necessary to the agent migrate to the node and collect all artifacts for the installation. After that, the mobile agent can be configured to perform the installation process without the network connection presence.

The system consists of a Home Machine that contains setups of the softwares to be installed on the network. The mobile agent is *created* in the Home Machine and is dispatched to specific node which on its arrival on to the destination node will start execution. Tools like EMCO Msi Package Builder can be used to create software Images. A log history is maintained by the system which records all the activities right from the start of the process till the end.

Before Installation, the node is checked for the presence of available applications on it. This can be done by creating an agent that dispatches to specified node, collects the necessary information and reports to administrator. In case if the required application is not present, an agent is created by the administrator and dispatched, that performs necessary operations on its arrival at the destination node. The time required by a single agent to install the required software on all the node"s over the network is very high. The alternative is to create multiple agents (N agents for N node"s) to perform the task. The security is provided by configuring java.policy file. The dispatched agent reports to administrator after successful execution and disposes at destination itself

V. RESULTS AND DISCUSSIONS

This chapter presents the results obtained from the developed system and discusses the same. The SIUMA Server is running on a system containing software setup files and remote nodes were selected by their IP Addresses connected to in a network

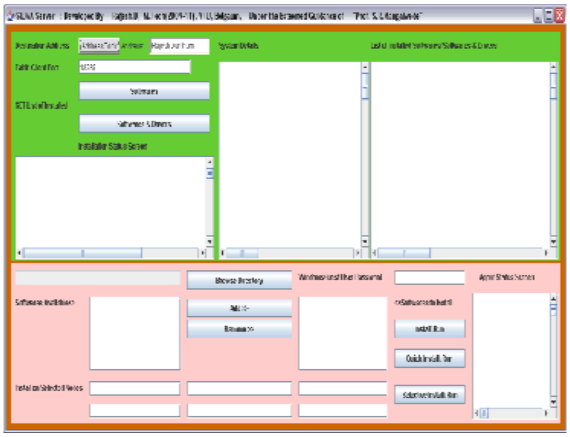


Figure 7.1. Graphical User Interface (GUI) for Administrator.

The snapshot shown in the Figure 7.1 is the GUI the SIUMAServer creates for the Administrator. Administrator can specify the remote node address and SIUMAClient port address to retrieve remote node"s system details, list of softwares installed and installs the selected softwares.

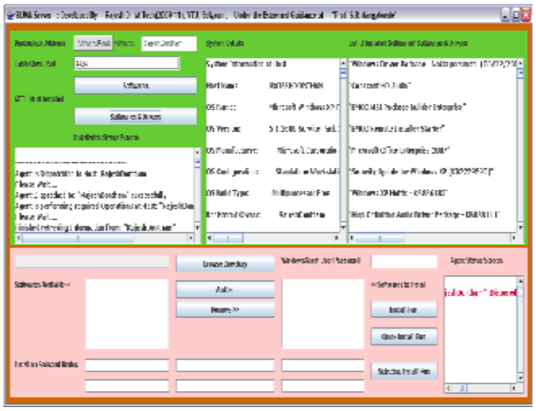


Figure 7.2. Remote node details and list of softwares installed

The snapshot shown in the Figure 7.2 depicts the system details and list of softwares installed on remote node „RajeshDontham" running SIUMAClient at port 4434.

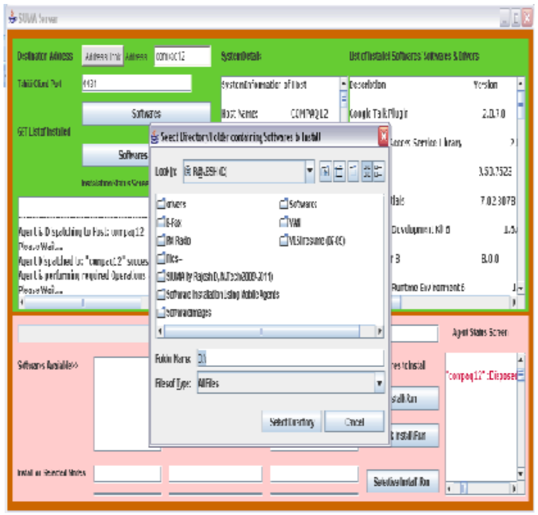


Figure 7.3. Selecting Directory/Folder containing softwares

The snapshot shown in the Figure 7.3 depicts the SIUMAServer Directory/folder selection option. Administrator can choose the directory containing softwares to be installed on remote nodes.

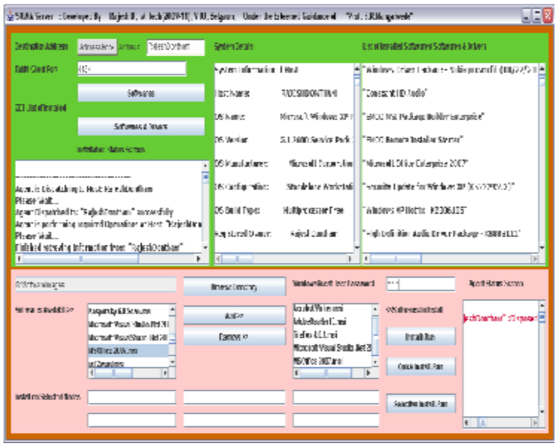
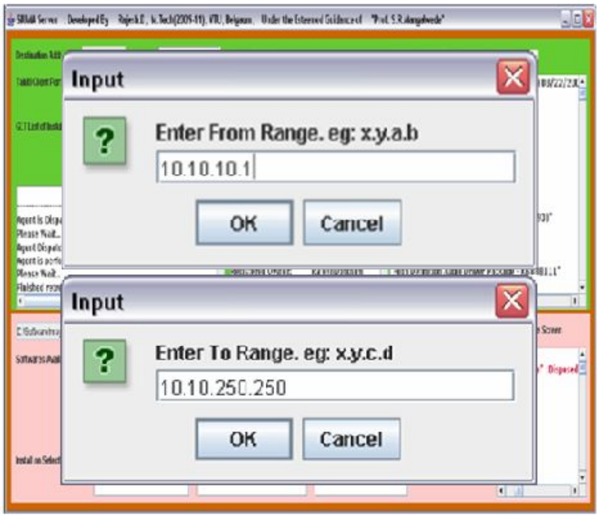


Figure 7.4. Selected sofwares to install.

The snapshot shown in the Figure 7.4 depicts that a list of softwares were added to „Softwares to Install" list from „Softwares Available" list. The Administrator can now install the selected softwares by clicking Install\Run button to install on selected remote system or click on Quick Install\Run button to install on multiple remote systems which prompts the Administrator to specify the IPRange as shown in the Figure 7.5. The Administrator can also install the selected softwares on selected systems (Maximum six) at a time by entering the remote node addresses in „Install on Selected Node"s" fields as shown in the Figure 7.6. Administrator can obtain the installation status at „Installation Status Screen" and Agents status in „Agent Status Screen".



Input

Enter From Range. eg: x.y.a.b

10.10.10.1

OK Cancel

Input

Enter To Range. eg: x.y.c.d

10.10.250.250

OK Cancel

Figure 7.5. Quick Install: Specify IpRange.

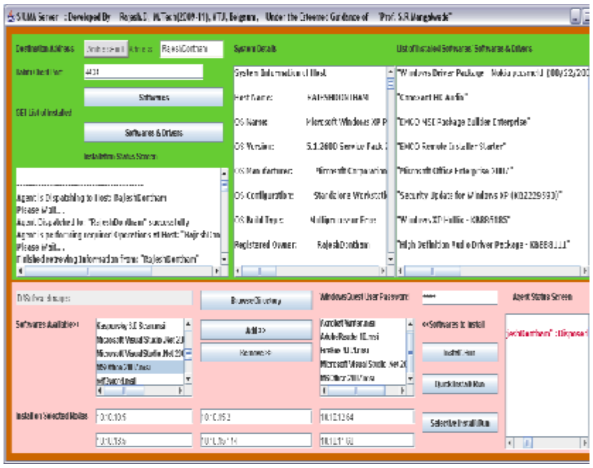


Figure 7.6. Selective Install: Specify addresses of selected remote systems.

CONCLUSION & FUTURE ENHANCEMENT

This paper presented a possible approach for software installation on a windows based node or network. We believe this proposed system is best suited for software management of individual node"s over a network.

As future work, we plan to implement the proposed system using ASDK and verify the efficiency of proposed approach.

REFERENCES

[1] Roberto Silveira Silva Filho, "**Mobile Agents and Software Deployment**", Roberto Silveira Silva Filho, ICS -Information and Computer Science, UCI - University of California Irvine 444 Computer Science Building, University of California Irvine, CA 92697-3425, USA.

[2] InstallShield: http://www.flexerasoftware.com/products/installshield.htm

[3] Hongbo Tian, Xiaoyi Zhao, Zhixing Gao, Taiqiang Lv, Xiaoshe Dong, **"A Novel Software Deployment Method based on Installation Packages"**, The Fifth Annual ChinaGrid Conference.

[4] B.Srikanth, D.Kirubakaran, N.Siddharth, Sudip Sanyal, **"Software Installation on a Huge Heterogeneous Network using Mobile Agents"**, 6th IEEE/ACIS International Conference on Computer and Information Science.

[5] Lange, D., and Oshima, M, **"Seven Good Reasons For Mobile Agents**", Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley Longman, Reading, Mass., 1998.