Virtual Machine in Automation Projects

Master Thesis
Xiaoyuan Xing
Department of Production Engineering and Management
KTH, Sweden
June 2010
Abstract

Virtual machine, as an engineering tool, has recently been introduced into automation projects in Tetra Pak Processing System AB. The goal of this paper is to examine how to better utilize virtual machine for the automation projects.

This paper designs different project scenarios using virtual machine. It analyzes installability, performance and stability of virtual machine from the test results. Technical solutions concerning virtual machine are discussed such as the conversion with physical computers, the configuration of virtual network and the use on different software platforms. It also suggests a workflow for the company to utilize virtual machine and share virtual machine resources across different project phases.

Key words: virtual machine, automation project, performance, network configuration
Table of Contents

Abstract .......................................................................................................................... 1
Chapter 1 Introduction ................................................................................................. 3
  1.1 Background ........................................................................................................ 3
  1.2 Goals ............................................................................................................... 3
  1.3 Scope of the research ....................................................................................... 4
Chapter 2 Theoretical Background for Virtual Machine ......................................... 5
  2.1 Basic concepts and classification .................................................................... 5
  2.2 Shared resources of system virtual machine ................................................... 6
  2.3 General comments on virtual machines .......................................................... 9
  2.4 File structure of virtual machine ..................................................................... 9
  2.5 Virtualization products ................................................................................... 9
Chapter 3 Methodology ............................................................................................. 11
  3.1 Basic idea for the design of the experiments ................................................ 11
  3.2 System configuration ....................................................................................... 12
Chapter 4 Results ...................................................................................................... 17
  4.1 Installability ..................................................................................................... 17
  4.2 Performance .................................................................................................... 18
  4.3 Stability ........................................................................................................... 19
Chapter 5 Discussion of Technical Issues ............................................................... 20
  5.1 Conversion ...................................................................................................... 20
  5.2 Snapshot and different versions of software platforms .................................. 22
  5.3 Network performance ..................................................................................... 24
  5.4 Comparison between hypervisor and hosted structure .................................. 25
  5.5 How to choose network type ......................................................................... 26
  5.6 IP Conflicts on the Lab Intranet .................................................................... 27
Chapter 6 Discussions and Recommendations ......................................................... 29
  6.1 Workflow to use VM in automation projects ............................................... 29
  6.2 Ideas and suggestions ..................................................................................... 32
  6.3 Limitations of the research and future work .................................................. 33
Chapter 7 Conclusion ............................................................................................... 34
Acknowledgements ................................................................................................... 35
Reference .................................................................................................................. 36
Appendix .................................................................................................................... 37
Chapter 1 Introduction

1.1 Background

Tetra Pak Processing Systems AB (TPPS) is a company in Tetra Pak group. TPPS provides process solutions and services for food manufactures to achieve superior production. It collaborates with customers from the whole world in fields of dairy, cheese, ice cream, beverage and prepared food. Global Projects is the main organization within TPPS which handles major projects for customers.

Engineers in TPPS meet problems aroused by software versions when they provide process services for customers. Customers might have different versions of software platform, and engineers must install the compatible software version on hardware for development. The installation often takes long time and delays the project. It becomes worse when some old versions of software are not available anymore. Another problem is that hardware cost for development tests is high and hardware deficiency limits development activities in the projects. Finding solutions for such problems are necessary.

Virtual machine has been very popular in last years. It has been widely used in IT infrastructure in many companies. People in TPPS also start to use virtual machine for different purposes. Automation engineers in Global Projects create virtual machines as copies of the customer’s physical machines and develop automation applications in these virtual machines. In such way they can easily set up the same system environment with the customer’s site and save the installation time. Engineers can view and modify code in virtual machines conveniently. Roll-out and Support team use virtual machines to make demonstration models for the customers. Virtual machines are easy to take and play, saving the bother of additional hardware and long installation time. Plant Automation team use virtual machines to distribute their software packages. However, more usages of virtual machines are to be analyzed within the company. Meanwhile some customers start to use virtual machines, making it necessary for TPPS to get more knowledge of virtual machines to achieve further collaboration with the customers.

1.2 Goals

The goal of this paper is to uncover the usefulness of virtual machine for the automation projects in TPPS. It is expected to give suggestions about how virtual machine can be utilized as an engineering tool in TPPS automation projects.

Several technical analyses are to be done in the research:
a) Make literature studies for virtual machines to find possible ideas how to use them
b) Test the feasibilities for the usage of virtual machines
c) Suggest structure and develop basic templates for using virtual machines

The documentation for templates is expected to help form internal guidelines.

1.3 Scope of the research

Virtual machines have been used broadly across different industries and different sizes of companies. Virtualization products can provide vast solutions such as server consolidation, virtualization management and enterprise desktop control, etc. Many of the services are for IT virtualization structure. The scope of this paper is focusing on the use of virtual machines in automation projects in TPPS. That is to say, that the vast functions of virtual machines for IT structure and data centres are not analyzed. The analysis in this paper is to discuss the possibilities to utilize virtual machines for the automation projects in TPPS in the near future. The focus is about technical aspects such as performance, scalability compatibility, and network configuration of virtual machines. The software products used and discussed in the tests are from the American computer software company VMware Inc.
Chapter 2 Theoretical Background for Virtual Machine

The system architecture of modern computer is composed of hardware, operating system and application software. These three layers are often developed in different companies, and sometimes even during different time periods. The well-defined interfaces among the different layers allow different groups of developers to concentrate in their field, but also lead to problems of incompatibility and inefficiency. Nowadays there are more requirements from the users for server consolidation, desktop consolidation, capacity optimization and better security. These problems might be solved by implementing a layer of software that provides a virtual machine environment.

2.1 Basic concepts and classification

Virtual machine (VM), originally defined in 1960s, is a software abstraction with the looks of a computer system’s hardware. Today the term includes a large range of abstractions in computer science. Virtual machines can be divided into two categories: system virtual machine and process virtual machine, depending on whether at the system or process level that the virtual machine is supported.

System virtual machine provides a complete system environment. It supports complete instructions from both the user and the system [1]. Figure 1(a) is an example of system virtual machine. The physical hardware on which the VMs are running is referred as “host”, the operating system (OS) installed the host hardware is referred as “host OS”. The software that runs on host OS and supports multiple virtual machines is referred as “virtual machine monitor (VMM)”. VMM provides a platform supporting different types of guest operating systems. There are two virtual machines running on the VMM in Figure 1(a), one is with Linux OS and the other is with Windows OS. The VMM is able to support multiple system VMs with access to I/O devices and graphical user interface on the desktop.
Process virtual machine supports an individual process. It is created and terminated along with its guest process. Figure 1(b) is an example of process virtual machine. The host Linux is able to support Linux guest applications but not some Java programs. It is Java VM that can help to execute these Java programs. It exists during the time when these Java programs are executed.

To clarify, all the virtual machines discussed later in this paper are system virtual machines, which are more applicable for the automation projects in TPPS.

2.2 Shared resources of system virtual machine

For a system virtual machine, the central problem is how to virtualize the shared hardware resources and allocate them among multiple guest operating systems. When the host is industry standard x86-architecture, the shared resources are mainly CPU, memory and I/O devices. The guest OS and the application programs compiled for it are controlled by VMM. VMM intercepts the operation when the guest OS performs a system operation involving the shared resources. The resources may be partitioned or time shared, depending on the resource use for VM. Some virtual resources might not have a corresponding physical resource, and in such cases the VMM may emulate the action of the desired resource through a combination of software and other resources on the host.

2.2.1 CPU

x86-based processor offers different levels of privileges for OS and applications to access hardware resource. In order to support the guest OS, the VMM need to change the default privileges in CPU and obtain control of hardware. The control can be obtained through techniques of binary translation or guest OS modification. Binary
translation technique translates the kernel code in guest OS to bring intended effects on the virtual hardware, meanwhile the application code in guest OS are executed directly on the processor. Guest OS modification technique modifies the kernel of the guest OS and enables the VMM to gain higher privileges than the guest OS.

To increase the processor utilization has been a main drive for virtual machines, since a computer might have very low CPU utilization rates when it is supporting only one OS. System virtual machines improve the CPU utilization by time sharing the processors through VMM. Even if every VM is assigned with only one virtual CPU, however, they are time sharing all the processors in the host.

VMM is more and more supported by the hardware suppliers now. The new generation of 64-bit processors is integrated with more features to simplify CPU virtualization. For example, the latest AMD-V processor has privileged instructions with a new CPU execution mode that allows the VMM to run in a root mode with higher privilege, without the need to do binary translation or guest OS modification.

2.2.2 Memory
In most computer systems without VMM, the OS will not release the used memory until other processes call for them. It would be better if the OS can manage the memory dynamically.

As shown in Figure 2, the machine memory in the host are partitioned and assigned as physical memories of the virtual machines. VMM is responsible for mapping the physical memory to machine memory, and the guest OS still has the control to map
the virtual addresses to guest memory physical addresses. Applications in the guest OS see continuous virtual address space, not physical memory. But sometimes VMM can map from virtual memory to machine memory directly. The exceptional dotted line in Figure 2 represents that the VMM maps virtual memory directly to machine memory with the assist of translation look-aside buffer (TLB) in the hardware [2]. When the OS newly maps the virtual memory to physical memory, the VMM will build and update the shadow page tables, to enable a direct look-up and accelerate the mappings next time. The new generation of hardware has more such assisting features and will bring more efficiency gains for memory virtualization.

With an efficient VMM managing multiple OS, the memory can be partitioned, borrowed and shared among different VMs [3]. As shown in Figure 2, each virtual machine is assigned a partition of the host memory. The partition is dynamically managed by VMM. When one virtual machine doesn’t have enough memory for its applications, it can turn to the VMM to borrow physical memory from another virtual machine. This borrowing action is like a request from the applications in the lender. The lender VM swaps out its pages to disks and gives it to VMM who finally lends the memory out. Moreover, when several virtual machines with the same OS execute similar applications and have the same pages in their virtual memory, all the VMs can point to the same single page of physical memory. It avoids duplication of the pages and saves lots of RAM to run multiple VMs with the same OS.

2.2.3 Device

Device virtualization involves routing I/O requests among the virtual machines and physical host. Devices such as keyboard, mouse and display, don’t have to be virtualized. Requests to and from the guest OS go through VMM and the interrupts are queued up by the VMM to be handled. For device such as storage disk, VMM is able to partition it into multiple parts and dedicates the parts to the virtual machines as virtual disks. For devices such as network adapter card, VMM duplicates them into multiple virtual devices and assign them to the VMs.

Virtual devices require virtual device drivers, which mostly can be obtained from VMM during installation. But manual driver installation might be necessary for the very old or very new guest OS.

The VMs have similar hardware devices if they are created by the same VMM, even if they are virtualized from different physical sources. Such similarity helps to achieve standardization and portability among different host platforms.
2.3 General comments on virtual machines

Virtual machine can help to solve the problems aroused by the standardized computer architecture. It can enhance computer performance through taking implementation-specific information into consideration. Virtualization software can adjust to better development of the processor and cores. It can also provide hardware resource duplication, giving appearance of multiple virtual hardware platforms which actually run on single physical hardware platform. Different types of operating systems can be supported on one host. Various types of virtual machines can be composed to form a virtual architecture freed of traditional compatibility constraints.

There are also some challenges when using virtual machines. The most concerning one is about performance. Virtual machines will increase the overhead on the host, which might decrease the performance level of both guest and the host. For example, in some cases system calls in guest VM must be trapped, interpreted and reissued to the host through VMM. As a result, additional overhead and longer queue time is unavoidable in CPU. Moreover, stability of virtual machine requires further investigation in automation projects. Although more and more software platforms support virtualized environment, malfunction or incompatibility might happen.

More usage of virtual machines for the automation projects will be discussed in later chapters.

2.4 File structure of virtual machine

The virtual machine provides an emulated environment with all the hardware resources the guest OS needs, but the nature of virtual machine is just a collection of files on the host’s storage disk. Take virtual machines created by VMware Workstation for example. The VM exists as a folder including vmx, vmdk, vmsn and txt files on the host. Some configurations of the VM can be done by editing these files on the host, and more configurations must be done through VMM. When the virtual machine is viewed in a VMM, however, what one can see is the files of the guest OS on the virtual disk. They look much similar to files of an OS on a physical disk.

2.5 Virtualization products

The market for virtualization products is boosting now, and more and more companies are taking use of virtual machines for server consolidation, enterprise desktop, cloud computing, and virtualized management. Different types of products include data centre platform, desktop products, management products, migration tools, conversion tools, etc. The three main software providers are VMware,
Microsoft and XEN companies. Other providers include Citrix, Novell, Oracle, Red Hat, Sun, etc. VMware has the most diversity of products. It provides desktop products installed on Windows, Linux and Mac OS, and enterprise software installed on bare hardware. XEN products are featured with para-virtualization, which enables some applications in guest OS to call host hardware resource directly. Such feature gives better performance in some particular cases, but XEN has only compatibility with Linux and hasn’t been able to fully support Windows yet. Microsoft provides hypervisor and desktop products, and emphasizes more for virtualization management.

The virtualization software can be divided into different types, depending on where they are installed, and what they can support. Some products are installed as a software in the host OS, such as VMware Workstation and VMware Server. Others products are installed on bare hardware, without the need to install a host OS, such as VMware ESX Server and Microsoft Hyper-V. They are called Hypervisor, to be distinguished from products requiring a host OS.

Some virtualization products emulate bare hardware machine and support guest OS, such as products from VMware, XEN and Microsoft. Additional installation of guest OS is necessary. Such products have better flexibility in migrating VMs between different hosts, and users can choose from different guest operating systems. Other virtualization products are emulating bare hardware machine with OS already installed, such as products from Virtuozzo and OpenVZ. Virtuozzo is able to support more than a thousand VMs on one physical host, but all the VMs must have the same OS.

The selected VMM for the tests in the following chapters are VMware Workstation, VMware Server and VMware ESX Server, which all come from the company VMware. VMware Workstation is a desktop product, and is used to create the virtual machines. VMware Server is a data centre platform supporting multiple virtual servers. Both of VMware Workstation and VMware Server run on a host OS. VMware ESX Server is a data centre platform which runs on a bare hardware. The reason for the choice is that in automation projects the most widely used OS is Windows series, and VMware provides the most proved support for virtual machines with Windows OS now.
Chapter 3 Methodology

3.1 Basic idea for the design of the experiments

Through interviews it is found out, that different groups of people in the company start to use virtual machine because of its portability. But the usage has been limited to be partial; no complete development for the automation projects in the company is done in virtual environment. Three scenarios are designed in this paper to emulate a simplified automation project. The scenarios are used to test and evaluate the performance of virtual machines, to see whether virtual machines can be migrated into our development environment in different contexts.

In many automation projects the most common model for customers’ system is consisted of PLC stations, OPC servers, Human Machine Interface (HMI) stations and Database. In a simplified model with only one OPC server and one HMI station, they can be combined to be one station. The system model in this paper contains three virtual machines; representing the PLC station (VM1), OPC server and HMI station (VM2), and database (VM3). Figure 3 shows how these three VMs communicate data with each other. HMI (VM2) takes control of the automation activities through PLC stations (VM1), and Database (VM3) stores the production data through operator station (VM2). The network configuration is made according to Figure 3, and details for setting up the model can be referred in the appendixes. It expected to observe the network performance in different scenarios.

The virtual machines are supported by VMM which is installed either in the host OS (Scenario 1 and 2) or on a bare hardware machine (Scenario 3). It is expected to observe the performance of different types of VMM. The structures for the three scenarios are shown in Figure 4, 5 and 6 respectively. Through the comparison of
Scenario 1 and 2, it is expected to observe the difference of virtual network and physical network. Through the comparison of Scenario 1, 2 and 3, it is expected to observe the difference of a hypervisor and a host structure. Different types of workload are selected to observe how the workload influences the virtual machine’s performance.

3.2 System configuration

The configuration for the host computer, VMM and the guest virtual machines in different scenarios are stated below.

3.2.1 Scenario 1

![Figure 4 Model structure for Scenario 1.1 and 1.2](image)

Scenario 1.1 and 1.2 have the same model structure shown in Figure 4, the only difference is the host configuration.

**Scenario 1.1 Host Configuration**

Base Hardware: x64 architecture, 4-core AMD Opteron Processor 280 2.40 GHz
Memory: 8.83GB
Disks: 300GB
Operating system: Microsoft Windows Server 2003 Enterprise x64 Edition SP1
Network Adapter Card: two HP NC7782 Gigabit Server Adapter cards
CD/DVD Drive available

**VMM Configuration in Scenario 1.1**

VMware Server Console 1.0.1
Scenario 1.2 Host Configurations
Base Hardware: x86 based architecture, 2-core Intel CPU T9600 2.8GHz
Memory: 3.48GB
Operation System: Microsoft Windows XP Professional SP3
Network Adapter Card: one Intel 82567 LM Gigabits Network Connection card
CD/DVD Drive available

VMM Configuration in Scenario 1.2
VMware Workstation 6.5

Guests Configuration (Virtualized Systems)
All three virtual machines have the same virtual resources as following:
Hard Disk: 16 GB
Memory: 1024MB RAM
Processors: 1
Operating system: Windows XP Professional
Network Card: one virtual network adapter card

The three virtual machines are installed with different software:
VM1: Siemens SEMATIC Manager and WinLC
VM2: WonderWare Intouch, SEMATIC NET PC, OPC Link, Tetrapak Plant Master
VM3: PLC Pump, Microsoft SQL package

Network Configuration
In VMware products there are three types of network that can be used by the virtual machines: bridged network, network using network address translation (NAT) and host-only network. Virtual devices such as virtual adapter cards and virtual switches are available for network configurations. In Scenario 1 host-only network is chosen. Static IP addresses are necessary for the communication among PLC stations, OPC servers and SQL database. The virtual DHCP server is disabled, and static IP addresses are assigned to virtual machines. After the network configuration is done, more configurations are required at application level.
Please refer to Appendix A for guidelines of network configurations.

3.2.2 Scenario 2

![Figure 5 Model structure for Scenario 2](image)

**Host Configuration**

There are three hosts in this scenario, each running one virtual machine. They are of the same type hardware.

Base Hardware: HP xw4300 Workstation, x86, Dual core Intel Pentium D processor, 3.2GHz

Memory: 2GB

Disks: 80GB

CD/DVD Drive available

Operating system: Microsoft Windows XP Professional

**VMM Configuration**

VMware Server Console 1.0.1

**Guests Configuration (Virtualized Systems)**

The guest OS configurations for the three virtual machines are the same with Scenario 1.

**Network Configuration**

Bridged network is chosen for scenario 2. A physical switch is needed to create network for these three virtual machines. In scenario 2 the virtual machines
communicate with each other through the physical network adapter card on their hosts. Please refer to Appendix A for further details of setting up a bridged network.

3.2.3 Scenario 3

![Model Structure for Scenario 3](image)

**Figure 6 Model Structure for Scenario 3**

**Host Configuration**

Base Hardware: X86 Dell OptiPlex, 2-core Intel processor, 2.8GHz  
Memory: 3G RAM  
Disks: 160GB  
CD/DVD Drive available

**VMM Configuration**

VMware ESX Server 3.5 Trial

**Guests Configuration (Virtualized Systems)**

The virtual machines and the software installed on them are the same as in Scenario 1 and 2.

**Network Configuration**

Another PC is required to be the remote control panel for VMware ESX Server, since the interface of ESX Server is much like Linux, which is not convenient for the administrator to organize and control. The PC station is linked with the host machine through physical switch and network cables, installed with controller tools provided by VMware. The structure is shown in Figure 6. All the virtual machines are
configured with host-only network. The PC station communicates with the host as a control panel, and the virtual machines are controlled remotely.

3.2.4 Workloads

The applications selected in this paper come from two sources: application from marketing demo which is relatively simple, and application from a going-on project, which contains much heavier workload on the PLC stations. They have similar functions and both use Tetra Pak Plant Master, they are both one part of food processing solutions. The application is initiated by HMI (VM2) to start a food recipe, and the PLC Station (VM1) will follow the recipe and execute PLC codes in PLC software, which is emulating real PLC in production environment. The Database (VM3) records the production data in VM1 and VM2 for traceability.

In automations projects, the PLC stations stand for the most workload in the network. There is only one PLC Station in Scenario 1 and 3. There are five PLC Stations in Scenario 2, which significantly increase the workload.
Chapter 4 Results

The results of different scenarios are observed from three aspects: installability, performance and stability.

4.1 Installability

Installability here refers to the time that the whole installation process takes for the scenario, including the network configuration. It is taken for emphasis since time can be a critical issue sometimes in automation projects. The precondition is that the tester has enough knowledge of the network model in Figure 3.

4.1.1 Scenario 1.1 and 1.2

The installation includes: install virtual machine monitor on the host OS, copy or create the three virtual machines and configure the network. Installing the VMM is just the same as installing other software, which takes short time. Copying the files of the virtual machines to the host runs at the speed of transferring files between different folders on the same host, the time is quite short, but is also considerable when the virtual machines’ files take large disk space. Network configuration is done with virtual network adapters and switches and takes short time. The installation time for scenario 1.1 and 1.2 takes around 2 hours.

4.1.2 Scenario 2

The installation includes: install virtual machine monitor on all the three hosts, copy or create virtual machines to each of the host, connect the hosts with a physical switch, configure all the virtual machines and hosts to be on one network. The installation is much the same with scenario 1, but more duplicate installations on the hosts. The network is first set up among the physical hosts through a switch. It takes short time since there is no need of DNS on this small-size network. Then the virtual machines are linked to this network through bridged virtual adapter cards. The installation can be done around half day.

4.1.3 Scenario 3

The installation includes: install VMware ESX Server on host, copy or create virtual machines on the host, install VI Client on PC which is acting as control panel, set up network for the host, the virtual machines and the PC control panel. The ESX Server is like an OS itself, and has an interface similar to Linux. Installation of ESX server takes around one hour. But then the installation of right drivers for the hardware takes days, and too tough to be solved under special hardware conditions. The installation
can hardly be done without sufficient knowledge and experience with Linux and VMware ESX Server.

4.2 Performance

Performance is a general word to evaluate the different scenarios. Specifically in this paper, it means the respond time between VM1 and VM2. In automation projects, the communication between VM1 and VM2, which is between PLC station and HMI station, is the most concerning aspect for engineers. After the HMI station initiates an application, the PLC station is supposed to respond and execute corresponding code as soon as possible. The response time is observed through the desktop in the host of VM1 and VM2.

4.2.1 Scenario 1.1 and 1.2

In Scenario 1.1 with the support of 4-core CPU and enough RAM in the host, the three virtual machines are working on the host-only network without any delay. Increasing the workload with another type of application doesn’t affect the responding time.

In Scenario 1.2 with hardware support of normal PC type, the host can support two virtual machines simultaneously without delay in powering on them. When the third virtual machine is powered on there appears delay in loading the guest OS. The delay is apparent when VM2 initiates an application. It takes around 10 minutes for VM1 to respond and execute corresponding code, and it takes even longer time for VM1 to write the data into database in VM3. No additional workload from the application can be loaded.

4.2.2 Scenario 2

Running three VMs on three hosts has similar performance as if the application software is running without virtual machine monitors. The delay can hardly be noticed when the workload is light.

However, the performance will decrease when PLC stations are loaded with heavy applications. Delay is more serious when the host of VM1 runs three virtual machines as PLC stations. Similar to Scenario 1.2, it takes around 10 minutes to power on three virtual machines for PLC stations. The whole network performance is largely decreased because of the increased workload on a single host for PLC stations.
4.2.3 Scenario 3

Because the installation meets problems of the hardware drivers, the performance of scenario 3 is not available. According to the test reports from VMware, with the right hardware support, which is important precondition, one ESX Server is able to successfully support 4-6 virtual machines without performance decrease. Scenario 2 is expected to give better performance than scenario 1.2, but it requires proof that to what extent the performance can be better for the automation projects.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Installability</th>
<th>Performance</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>About 2 hours</td>
<td>high</td>
<td>good</td>
</tr>
<tr>
<td>1.2</td>
<td>About 2 hours</td>
<td>low</td>
<td>good</td>
</tr>
<tr>
<td>2</td>
<td>About half day</td>
<td>High/low</td>
<td>good</td>
</tr>
<tr>
<td>3</td>
<td>One day or more</td>
<td>------------</td>
<td>good</td>
</tr>
</tbody>
</table>

Table 1 Summary of results in different scenarios

4.3 Stability

Stability here refers to the ability of the virtual machines to run without system disaster. The virtual machines rely less on OS in the host, and rely more on VMM. There was no stability problem appeared during the tests for all scenarios, due to the stability of Windows OS and VMware virtualization products.
Chapter 5 Discussion of Technical Issues

5.1 Conversion

The conversion and clone functions of VMM contribute much to virtual machine’s portability. The virtual machines in all scenarios root from the same source, named VM0 here, which is a physical machine installed with software for PLC, HMI and Database. The steps to create VM1, VM2 and VM3 in the tests are:

- a) Convert source physical machine to virtual machine VM0
- b) Clone three copies of VM0, then we get three identical virtual machines as bases for VM1, VM2 and VM3
- c) Uninstall the unnecessary software from each virtual machine, make system and network configurations according to their station roles.

From the test results of Scenario 1 and 2 it can be seen that it saves much installation time to set up scenarios with multiple virtual machines in such steps. It will be very helpful in cases with many identical virtual machines, for example, a production line model with multiple PLC stations.

Three types of conversion can be used for automation projects, which will be explained further as follows.

5.1.1 Virtual to physical conversion

Virtual machine to physical machine (V2P) conversion is needed in several circumstances. For example, although the use of virtualization has allowed failover practices less reliant on physical hardware and recovery can be done through keeping the files on storage disk, still sometimes conversion to physical machine is necessary after the disaster recovery. Conversion can also happen for delivering development files to customers who don’t use virtual machines. Besides, conversion might be needed when restructuring lab system with multiple virtual machine and physical hosts.

Most VMM don’t provide V2P function, additional conversion tool is needed. In order to accomplish V2P conversion, there should be a source virtual machine, a target physical machine and a third-party conversion tool, which should satisfy special requirements and be compatible with each other. For example, conversion tools should be compatible with the guest system in virtual machine. It should be Microsoft Sysprep and SymantecGhost for Microsoft Windows OS, and Kudzu for
Linux OS. And a symmetric multiprocessing (SMP) virtual machine must have target physical machine which is SMP supportive. The conversion tool is installed on the source virtual machine and creates an image of the virtual machine. Then the image is transferred to the target machine, and additional drivers should be installed.

The V2P conversion often meets problems with drivers. Some drivers are installed by imaging tool, but not enough for all the devices on the target machine. Event logs of the target machine are often filled with driver mistakes, especially for the network drivers. The local system on the target physical machine and hardware manufacture should be checked to find appropriate drivers for network card, disk controller, graphics card, keyboard, mouse and other devices.

5.1.2 Physical to virtual conversion

Physical machine to virtual machine (P2V) conversion is needed when creating a virtual machine for different purposes. Most VMM software provides function for P2V conversion, such as VMware Workstation, VMware Server Console and Microsoft Virtual PC. The P2V process is to convert the system image into virtual machine files on disk, and it is easier and quicker to do P2V conversion than V2P conversion. There are fewer problems of drivers in P2V conversion. In most cases the drivers for the virtual devices in the virtual machines will be automatically installed. For example the VMware Tools provided by VMware contains most drivers for virtual machines. The exception is that when the OS in the virtual machine is very old, for example, Windows NT, then some device drivers might require manual installation.

5.1.3 Virtual to virtual conversion

Virtual machine to virtual machine conversion (V2V) is as a matter of fact the action of copying files. V2V conversion contributes much to the portability and mobility of virtual machines. Most VMM software provides function for cloning virtual machines. When the identical virtual machines are running on the network, changes should be made for computer names and network configurations to avoid conflicts.

One issue for V2V conversion is to upgrade or downgrade a virtual machine. Due to different versions of virtualization software, there are different versions of virtual machines. For example, virtual machines created by VMware Workstation can be of Version 4, 5, 6, or 6.5, and each version runs on compatible platform. VMware Server Console 1 can only support virtual machine created by VMware Workstation 5 or lower.
Upgrading or downgrading the versions of virtual machines is to add or delete some features of the virtual machines provided by the VMM tool, the core of the virtual machine doesn’t change. In Scenario 1.1, the virtual machines are downgraded from Version VMware Workstation 6.5 to 5 to be able to run on the physical server with VMware Server Console 1. The amount of virtual devices, such as virtual switches and virtual adapter network card, is reduced after downgrading. When setting up scenarios for automation projects, the engineers should be cautious of the versions and required features of the virtual machines.

5.2 Snapshot and different versions of software platforms

In automation projects it is important to follow the system integrity. The versions of software platforms for PLC, HMI and the operating system used in the lab should be the same with the ones used at customers’ site in order to guarantee the functions and performance. The computers in production environment mostly forbid automatic updates and keep a certain old version, and sometimes they are updated at the customers’ site. Traditionally, the automation engineers keep in accordance with the customers by searching for right versions of OS and software each time. Installation takes long time, and it is annoying when the versions at the customers’ site are too old to find. Taking snapshots of a virtual machine can make it easier to store different versions of software platforms and different application developments.

Taking snapshots of virtual machine is making records of different states of the virtual machine. A snapshot captures the states of memory, system setting and virtual disks in the virtual machine. Each snapshot has unique files in the file folder of the virtual machine. Snapshots files are distinguished from each other and they don’t interact with each other. In a case where engineers need different versions of the software and OS, they can use one virtual machine to take snapshots for different versions of software and OS. Older snapshot is for lower version of software, and newer snapshot for higher version of software. Either linear snapshots or process tree snapshots can be taken, according to the complexity of software and OS version.

Linear snapshots can be taken when the virtual machine has only one kind of software that has different versions. Each snapshot point is a restoration point in the sequence. Take a snapshot before the software is updated to a new version, and restore to the old version when needed. Figure 7 shows an example of linear snapshots for virtual machine. The baseline is a virtual machine installed with Wonderware InTouch 7.1. As the software platform is updated, each version has a corresponding snapshot. The user can restore to version 7.1, 8.0, 9.0 or 10.1 without delay and difficulty. A liner snapshot is especially suitable for training and demonstration. Starting point of each training lesson or demonstration scenario can be saved in a sequence.
In a more complex case where the OS and multiple sorts of software have different versions, a process tree of snapshots can be taken. The relationship between the older snapshot and newer snapshot is like parent and child. One older snapshot can have more than one newer snapshot as its children. A baseline state is selected, usually with a lower version service pack for OS. An example of process snapshot tree is shown in Figure 8, the baseline is a virtual machine installed with Wonderware InTouch 7.1, OPC Link 7.1 and MS Windows XP SP1, when InTouch, OPC Link and the service pack are updated, more snapshots of different versions can be taken.

Snapshot is helpful when engineers would like to try out some applications in different environments. They can go forward or back in different versions of the software platform to make tests on the same virtual machine.

When the engineers need completely different operating systems, it is suggested to create a new virtual machine rather than make snapshots. For example, if the automation project needs to test two types of operating systems, then two virtual machines should be created.
More broadly speaking, a VM template library can be established for the convenience of automation projects in the near future. The categories of virtual machines can be divided according to their operating systems and software platforms. Virtual machines installed with certain operating system and software platform can be seen as a baseline. The operating systems are mostly from Windows OS series, and the software platforms are from ABB, Siemens or Rockwell. Then different snapshots are recorded for each virtual machine, representing different versions of software platform and OS. Such virtual machine library can be used to collect virtual machine templates and set up scenarios for automation projects. It is helpful to save installation time, and can promote to share resources among different project teams.

5.3 Network performance

The virtual machines are equipped with two key virtual network devices: virtual network adapter and virtual switch. They are both Layer 2 Ethernet devices. There are different types of virtual adapters to choose from. For example, in VMware products, a e1000 is a virtual adapter which strictly emulates the Intel E1000 Ethernet adapter, and a vlance is a virtual adapter which strictly emulates AMD Lance PCNet 32 Ethernet adapter. In Vmware VMM there is an adapter vmxnet which is especially designed for higher performance in virtualization. One virtual machine is allowed to have multiple adapter cards in order to be on multiple networks. Each adapter card has its own MAC address which can be assigned by the user. Virtual switch is much the same with a modern Ethernet switch: it looks up each frame’s destination MAC when it arrives and forwards the frames to one or more ports for transmission. There are virtual ports on virtual switches, which are logical connection points among virtual devices and between virtual and physical devices.

The obvious benefits of virtual network are the lower hardware cost and faster network communication. No additional physical adapters are needed to set up a host-only network, as long as the host is powerful enough to support the whole network. It is convenient to set up networks with these virtual network devices. For example, in Scenario 1 the network provides same network service as physical network but requires less duplex settings. This network contains one physical host and three VMs, whose communication all takes place in the host system’s RAM. All the data transfer is nearly instantaneous without collisions or signaling-related errors. The data read-in from VM2 OPC Server to VM3 Database Station takes almost no delay. Scenario 2’s network, where the communication of the virtual machines takes place through the physical adapters and switch, has some delay in data transfer and remote commanding.

The network speed is not a concerning issue in automation projects in TPPS, but the flexibility and migration of network is of great concern. The features of virtual
network might be used to balance network traffic and achieve better integrated performance. Diverse network structures can be set up with combination of physical machines and virtual machines. The stations requiring frequent and high-speed communication can be arranged as virtual machines on one host, and the stations which require frequent changes in installation should be set as virtual machines. Since additional features such as DNS server and remote desktop control can easily be added, the network has enough extendibility for automation engineers to set up complex structures.

5.4 Comparison between hypervisor and hosted structure

There are two types of VMM to choose from to support system virtual machine: a hypervisor VMM which takes control of the bare hardware directly, or a hosted VMM which is installed on the host OS. Scenario 1 and 2 use hosted VMM while Scenario 3 uses hypervisor VMM. A hypervisor has high availability and management features of the hardware. A hypervisor can take better use of the hardware than in Scenario 1 and 2, but it has advanced requirements for CPU and memory in the host hardware. The recently released hypervisors require CPU that supports 64 bit systems and compatible devices. Since most of the machines in the lab are only 32 bit support, there is difficulty to utilize hypervisor.

As its name indicates, a hypervisor is designed to have complete control of underlying hardware, and it is mostly used in datacenters where frequent client services and data communication are needed. It is supposed to be able to run 4 to 6 virtual machines on one host machine with no decreasing performance, while a hosted VMM can normally support 2 to 4 virtual machines. Most hypervisors require additional PC as control panels, since the interfaces of hypervisors are difficult to handle. In contrast, most hosted VMM are desktop products, engineers who are familiar with Windows can learn to use VMM quickly. Hypervisor mostly integrates extra functions such as live migration and memory management, but they are not at the request of automation projects now. Hypervisor is suitable for scenarios running multiple virtual machines for PLC stations, as it has strong availability. It can better optimize network and storage devices since the hypervisor talks directly to the devices and requires less passed certification, but it is not suitable to run scenarios requiring development work. For example, in automation projects it is common to emulate customers’ production line in the lab. It is not convenient for engineers to do coding and testing on hypervisor during the project. But hypervisor might be a good choice for final testing in commissioning at the site if the production environments are using hypervisors at the site in the future.

In summary, the hosted structure has important advantages that it is easy to be installed and implemented, and that it can support more device virtualization since it
can take use of diverse drivers in the host OS, while the hypervisor has the advantages of less overhead and higher optimized I/O subsystem for network and storage devices. The engineers should consider both the available resources of physical hardware and the required functions of the virtual machines when choose from hypervisor and hosted VMM for automation projects.

5.5 How to choose network type

Three types of network can be set up for virtual machine: NAT network, host-only network and bridged network [4]. They provide different communication ways. The commonly used network devices are virtual network interface controller (NIC) and virtual switch provided by VMM. Engineers should choose the network type according to the requirements of the project and the different features of the network.

NAT network configures a private network on the host computer. The virtual machines are assigned with IP addresses by the virtual DHCP server. Virtual machines can communicate with each other, and virtual machines can initiate connection with external machines on the external networks. But the other computers on the external network can not initiate connection with the virtual machines by default. Special configuration about ports and NAT translation table should be done to allow external machines initiate connection with the virtual machines. NAT is helpful when there are not enough IP addresses for the projects, or when there are duplicated computers with IP conflicts.

Host-only network sets up a private network on the host. Communication can take place between the virtual machines and the host. But there is no way for the virtual machines to communicate with machines on the internet. The isolated host-only network is suitable for small automation projects in which all virtual machines can run on one host and no external network communication is needed. It is also helpful for small-sized function test, since it is very easy and fast to set up.

Bridged network configures a virtual machine as a unique identity on the network, separate and unrelated from its host. The virtual machine has its own IP address and MAC address, and it is indistinguishable from other physical machines on the external network. A bridged virtual NIC sends all the packets on the wire with its own unique IP and MAC address, and it is connected to a virtual switch which is driven by the VMnet driver. The virtual switch forwards the packets to the physical NIC through virtual bridge, and then the packets is sent out to external network. The virtual machine needs the physical NIC as a media to connect the external network, but it keeps its dependence of identity. The virtual machine is visible to other computers on external network. Bridged network works with both wired and wireless
physical network cards. Bridged network treats the virtual machines as a physical machine on the network and provides convenient unimpeded communications, but it requires unique IP address for each virtual machine. Bridged network can be chosen when there are enough IP resources and when there is no IP conflict problem on the network.

5.6 IP Conflicts on the Lab Intranet

5.6.1 Problem statement

When automation engineers deliver process solutions to the customers, the IP addresses of the delivered equipments are set during development procedures, which belong to the subnet of the lab intranet of TPPS. There is possibility that the engineers set the same IP address for computers in different projects because of the standardized project structures. When the customer requires after-sale or extension service, the automation engineers take the virtual machine files back to the lab intranet. And in order not to make trouble in configuration files when deliver them back to the customers, the engineers prefer to keep the IP addresses in virtual machines as they were, which results in the problem that there are more than one virtual machine with the same IP address on the lab intranet.

This problem can be solved by setting up network address translation (NAT) for the virtual machines. In NAT the virtual machines form a separate network on the host computer and there is a virtual DHCP provided by the VMM to assign IP addresses for this separate network. The VMs don’t have their own IP addresses on the external network and they appear as the host machine to the other machines on the external network.

5.6.2 Solution using NAT

NAT became popular in mid-1990s to solve the exhaustion problem of IPv4 address. The motivation is that the local network uses just one IP address as far as the outside world concerned [5]. The NAT device in VMM uses similar principle as a physical NAT device. The NAT network is consisted of virtual machines, virtual switch, virtual NAT device, and virtual DHCP. The virtual machines are connected to the virtual switch, so is the NAT device and the virtual DHCP server.

NAT device implements network address translation for the communication between virtual machines and external machines on the internet. It has a reserved IP address assigned by the DHCP server. For outgoing data grams from the virtual machines to the external network, NAT device replace (source IP address, port #) with (NAT IP address, new port#). For incoming data grams from the external network to virtual
machines, NAT device replace (NAT IP address, new port #) with (source IP address, port #). It is able to remember the translation results in a NAT translation table to improve the translation efficiency.

For outgoing data grams, the connection is initiated by the virtual machine. As long as the virtual machine knows the destination IP address, the connection can be accomplished by default. A virtual machine on the NAT network can automatically use any protocol using TCP or UDP, and it works well for client applications such as Web browsing and passive-mode FTP. In VMware products additional protocol support is built into the NAT device to allow FTP and Ping to work transparently through it.

Manual configuration must be done on the NAT device, to make the incoming data gram from external computers destined for a certain port, because NAT device will not forward the incoming data gram to the virtual machine by default. When external machine initiates connection with virtual machines, it must send the datagram to a pre-defined port on the host machine, then the NAT device forwards the datagram from this pre-defined port to the corresponding port on the destined virtual machine, according to the NAT translation table. The configuration line is usually written in the configuration file of VMM in the following form:

port no. on the host: IP address of the destined virtual machine: destined port no. on the virtual machine

Please refer to the appendix for the step by step guide to configure IP and port using NAT.

NAT seems to be a good way to solve IP conflicts of virtual machines, and it can also bring better security for the network. A standard NAT configuration provides basic fire-wall protection and external machines can’t connect the virtual machines unless they grasp the translation table in NAT device. But NAT can cause some performance loss, since all the data grams coming and leaving from virtual machines requires NAT device to translate the addresses and ports.
Chapter 6 Discussions and Recommendations

6.1 Workflow to use VM in automation projects

Virtual machine can be seen as an engineering tool for automation projects, which helps the engineers to better utilize hardware resources and gain more operational flexibility. It can be used across different stages of an automation project.

6.1.1 Quotation Phase

- Feasibility study
- Risk assessment
- Quotation work and review

In quotation phase, virtual machine can help to save some time and cost for the team to gain the contract. The marketing team can show demonstration model to the customers in a virtual machine, which is convenient and efficient. Virtual machine can also help to do preliminary feasibility study after the customer’s inquiry is received. Engineers can partially convert the computers at the customer’s site into virtual machines, bring them back to the lab and explore potential automation improvement. In such way they can set up similar scenarios with the customer, yet don’t have to spend long time on installation. During the negotiation phase of the project, it is practical to discuss and verify primary feasible solutions using virtual machine.

6.1.2 Implementation Phase

- Engineering
- Installation
- Procurement

Virtual machine can play more roles in implementation phase of a project as an engineering tool. After the contract is gained and the customer’s order is identified, system model for process solutions would be discussed. Since virtual machine can provide operating system and application software as a physical computer does, the system model can be designed as a combination of physical computers and virtual machines. Several steps can be followed to set up a system model for development work, as shown in Figure 9.
Step 1 Design a system model for process solutions, where all the computers are designed to be physical ones. The computer stations might include PLC stations, OPC servers, operator controllers, database, etc. The design method should be the same as the traditional system models in the automation projects. It can either be a new delivery or an after-sale service such as extension solution.

Step 2 Based on the system model in Step 1, discuss which of the computers in the system can be set as virtual machines. The decision should be based on both technical feasibility and financial worth.

From technical aspect, several questions about virtual machines should be taken into consideration:
- Whether the virtual machines can provide the required functions as physical computers.
- Whether it is possible to set up the virtual machine.
- Whether there is enough knowledge to configure the virtual machines.
- In what form can the development work in virtual machines be delivered to the customers? Is the migration feasible?

From the financial aspect, the cost analyzer should consider the tradeoff to use virtual machines. A system with virtual machines requires less hardware amount, less electricity energy and less cooling fee for the lab temperature. But it might require more expensive physical hosts to run the virtual machines. Since the project team is not isolated but connected with the whole department, the hardware supporting virtual machines might be recycled to be used by other projects, which might bring long-term benefit for the whole department. The virtualization software cost and man-power training cost should also be taken into consideration.

Decision in Step 2 largely depends on the project type. If it is a completely new delivery, TPPS delivers not only the development work but also all the supporting hardware. The engineers will have sufficient hardware for development work. In that case virtual machine can be used in the initial period when hardware has not been fully settled, scenarios can be set up in virtual machines earlier and promote the project progress. If it is an after-sale service such as extension solution, then virtual machine is likely to be used more in engineering phase and plays greater role.

Step 3 Select the underlying hardware and the virtualization software to support the virtual machines.

There are different requirements for the underlying hardware and virtualization software depending on the amount and the types of virtual machines. For the hardware, the bit of processor, the amount of cores, the size of memory, the storage disk size and the required I/O devices should be considered, it should guarantee support for the virtualization software. The virtualization software can either be a workstation which can create and modify virtual machines, or a player that only runs virtual machines but provides no editing functions.
Step 4 Migrate virtual machines to the physical hosts. The engineers can check the virtual machine library for templates with compatible operating system and software platforms. If there is no template in the library, then the engineers can either create a new virtual machine and install OS and software platform on it, or turn to the customers for help. Whenever a new virtual machine is created, it should be added to the virtual machine library.

Step 5 Configure network for virtual machines. Make them communicate well with each other and with other physical computers. Details about network configuration can be referred in the appendix guidelines.

Step 6 Establish a backup solution for virtual machines. Store the backup files as reference files for further use and disaster recovery.

6.1.3 Commissioning phase
- Delivery
- Commissioning
Virtual machine can be used to improve working efficiency in commissioning phase. When engineers deliver the solutions to the customer site, they can duplicate the virtual machines. When the solution requires modifications, different groups of people can work simultaneously and remotely on it. Since virtual machine can use snapshots to record different stages of the development work, it is convenient for the engineers to go back to a certain state of the development work and check the codes.

6.1.4 Warranty Phase
- Maintenance
- Further extension
All computers used in implementing phase can be converted to virtual machines and kept as warranty files. When the customers come back later for maintenance or extension service, the automation engineers can easily set up the scenarios with virtual machine files. A virtual machine folder should be set up for each project.

6.2 Ideas and suggestions
How to utilize virtual machine as an engineering tool is not only a technical issue. It also requires people’s attention how to better involve virtual machine into the workflow of a project. Virtual machines can play different roles in different project phases, and different groups of people can focus on different aspects. Technical solution designers should know about the similarities and differences between a physical computer and a virtual machine. Automation engineers should be capable to set up, configure and troubleshoot virtual machines. People in the department have scattered knowledge and experience about virtual machine, and it is better to gather their knowledge and experience to make it available as a shared resource among different teams. The shared resources can be, for example, a virtual machine library containing templates with different versions of operating systems and software platforms, or a virtual machine file folder for back up files of each project, or virtual machine configuration instructions. Shared resources can help to promote virtual machine to be better utilized by different groups of people and improve the integrated working efficiency.

6.3 Limitations of the research and future work

All the tests in this paper are using VMware products which all use the same virtualization kernel. Since VMware can well support the virtual machines used in automation projects now, no other virtualization product has been considered so far. This paper only uses Windows OS to take tests, since the OS used in automation projects now are Windows series. But virtual machine might give different performance under Linux OS, and different virtualization products can vary a lot. In the future more types of virtualization products from other companies can be tested and used in the projects.

Network security is not discussed in this paper. In automation projects the development work is done through lab intranet and commissioned through the customer’s industrial network, and network security is not a great problem. But if any of the automation projects requires connection with internet in the future, virtual network security should be further investigated.

Virtual machine has been considered as an engineering tool so far, and it is working as one part of the developing platform, but there is a trend for it to become a whole developing platform. There might be a virtual lab established in the future, where the tests and development work are completely done on virtual machines. Such virtual lab is practical when our customers and software suppliers have more support for virtual machines, and when virtual machines are more common for use in the industrial production. The company should have a forward view yet keep pace with the customers.
Chapter 7 Conclusion

Through computer tests it is found out that virtual machine can be used to improve the work efficiency and reduce the cost in automation projects. Virtual machine can easily be converted, migrated and stored, providing convenience for the engineers to set up and modify the computers. It can better utilize the hardware resource, and has the advantages of portability and flexibility to support different automation software platforms. Virtual network can provide non-decreasing performance under certain hardware conditions, and can be used to set up diverse networks for the automation projects.

Virtual machine can be used in different project phases by different groups of people. Step-by-step workflow concerning virtual machine is suggested in this paper. It is recommended to set up virtual machine library and back up all virtual machine files to share and maintain virtual machine resources among different project teams.

As the virtualization technique of hardware and software is developing fast, it will be more common to use virtual machine in automation projects in the near future. More issues such as diverse guest operating systems and system security are to be discussed.
Acknowledgements

This thesis work is carried out at Global Project, Tetra Pak Processing System AB in Lund, Sweden from February to June, 2010. During the course of interviews and investigations I have received valuable help from the colleagues for knowledge of automation project.

Especially, I wish to thank my supervisor Bert Lindström at Tetra Pak for suggesting the approach developed in Chapter 3 and helping me acquainted with the automation project. I would also like to thank Torbjörn Ahlm and Maria Hansson, the automation engineers in Tetra Pak, for the technical support in the virtual network setting up.

It is only with the help of all the people here that I can finish this thesis work.

Thank you all!

Xiaoyuan Xing

June, 2010
Reference


6 Peter M. Chen and Brian D. Nobel, 2006, When Virtual Is Better Than Real, University of Michigan.


13 Paul Barham, Boris Dragovic, Keir Fraser, Steen Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Lan Prattt, Andrew Warfield, 2003, Xen and the Art of Virtualization, University of Cambridge Computer Laboratory.


Appendix
Network Configuration for Virtual Machine

Contents

1 Target ..............................................................................................................................38
2 Configuration at network layer .......................................................................................38
  2.1 Set up a host-only network 38
  2.2 Set up a bridged network 39
  2.3 Set up a NAT network 40
    2.3.1 Set up a general NAT 40
    2.3.2 Port configuration in NAT 40
3 Configuration at application layer ..................................................................................41
  3.1 VM1 PLC station 41
  3.2 VM2 HMI OP Station 41
  3.3 VM3 SQL Database Station 41
1 Target
The target of this document is to help set up network for virtual machines using
VMware products. Please refer to Chapter 3 Methodology in the thesis paper for
details of the hardware and software used in the configuration. Please refer to Section
5.5 and 5.6 for how to choose network type for virtual machines.

In order to accomplish the communication among the three virtual machines (PLC
station, HMI and OP station and Database), configurations for network layer and
application layer are needed. All configurations are done on VMware Workstation
6.5.

2 Configuration at the network layer
2.1 Set up a host-only network
After we get three virtual machines, we should first set up a network. Host-only
network is useful when all the virtual machines run on the same host. A host-only
network is set up in scenario 1 as following steps.

1 Make sure that the VMware Network Adapter VMnet1 is enabled on the host. You
can check it through the ‘Network Connections’ in Control Panel on the host. This
network adapter card is created automatically when you create a virtual machine, and
is used as default adapter card for host-only network.

2 Configure the adapter card
One adapter card is created automatically for every virtual machine. Configure them
all to be host-only type. A physical PC must have a network adapter or NIC (network
interface controller), for each physical network connection. Similarly, a virtual
machine must be configured with a virtual network adapter for each LAN segment it
interacts with.

To connect a virtual machine to multiple LAN segments simultaneously, you must
configure that virtual machine with multiple network adapters. In this experiment
only one network is needed, and each virtual machine has only one adapter card.

Select a virtual machine, and choose VMware Workstation-> VM -> Settings->
Network Adapter Card, then set it as Host-only.

3 Disable the virtual DHCP server
In order to assign static IP addresses to the three virtual machines, the virtual DHCP
server should be disabled; otherwise it will assign dynamic IP addresses to all the
virtual machines on the host-only network.
Choose VMware Workstation-> Edit-> Virtual Network Editor-> DHCP ->VMnet1,
then Stop it.

4 Assign IP addresses for the three virtual machines
Assigning IP for a virtual machine is nothing different from for a physical machine. Open a virtual machine, in Windows XP, open the control panel, and select Network Connections-> Properties-> Internet Protocols Properties, then type a static IP.

The benefit of host-only network is that it is isolated and will not communicate with other machines on the Internet, so you don’t have to worry about the IP conflicts. VMware Workstation has default subnet for VMnet1, which is 192.168.0.***. 192.168.0.(2-127) are for static addresses, and 192.168.0.(128-253) are for DHCP assigned. The default static addresses are used in this experiment. But certainly other subnets and IP addresses can be set as you like.

2.2 Set up a bridged network
Bridged network is used when virtual machines are running on different hosts. A bridged network is set up for Scenario 2 as following steps.

1 In bridged network VMnet0 is used for virtual machines. The virtual machine is homogeneous to other machines on the network. VMnet0 is mapped to a physical adapter card on the host. The host might be using different adapter cards to be connected to different networks, make sure the card for the network where the VMs use is enabled. You can check it through the ‘Network Connections’ in Control Panel on the host.

2 Configure the adapter card of the virtual machine to be ‘bridged’. VMnet0 enables the virtual machine to be on the same network where its host is.

Select a virtual machine, and choose VMware Workstation-> VM -> Settings-> Network Adapter Card, then set it as Host-only.

3 Disable the virtual DHCP server
There is no physical DHCP on the network, but in order to assign static IP addresses to the three virtual machines, the virtual DHCP server should be disabled; otherwise it will assign dynamic IP addresses to all the virtual machines on the host-only network.

Choose VMware Workstation-> Edit-> Virtual Network Editor-> DHCP -> VMnet 0, then Stop it.

4 Similar to Scenario 1, assign static IP addresses to the three virtual machines and the three physical machines. Make sure the IP addresses are in the same subnet.

There is no DNS server in this scenario. If a network with DNS needs to be set up, after the above steps the IP addresses and computer names of the virtual machines need to be added to the DNS server, just the same configuration as if they are physical machines.
After these steps the virtual machines are connected at network layer and should be able to use PING command for each other. Then the configuration at application layer should be done as follows.

2.3 Set up a NAT network
2.3.1 Set up a general NAT

1 Make sure that the VMware Network Adapter VMnet8 is enabled on the host. VMnet 8 is the default virtual device for NAT. You can check the status through the ‘Network Connections’ in Control Panel on the host.

2 Configure the adapter card of the virtual machine to be ‘NAT’.

Select a virtual machine, and choose VMware Workstation-> VM -> Settings-> Network Adapter Card, then set it as NAT.

3 Disable the virtual DHCP server

Choose VMware Workstation-> Edit-> Virtual Network Editor-> DHCP -> VMnet8, then Stop it.

4 Assign static IP addresses for the virtual machines.

2.3.2 Port configuration in NAT
The four steps in 2.3.1 enable virtual machines to communicate with each other and initiate communication with external computers. If the external computers also need to initiate communicate with the virtual machines, or the there are IP conflicts in the network, then port configuration should be made as follows.

1 Open the configuration file for NAT. The location is C:\Documents and Settings\All Users\Application Data\VMware\vmnetnat.conf

2 In the [incomingtcp] section of vmnetnat.conf, assign a port number on the host to an IP address and port number on a virtual machine. For example, in the following line,

   8887= 192.168.32.4:21

   It creates a map from port 8887 on the host computer to port 21 on the virtual machine with IP address 192.168.32.4. All the tcp incoming data packages at port 8887 on the host will be forwarded to port 21 on the virtual machine. Multiple lines can be configured in this file.

3 In the [incomingudp] section of vmnetnat.conf, assign a port number on the host to an IP address and port number on a virtual machine. This section allows the user to
configure the forwarding of udp data packages. The form of the configuration lines is the same with the lines in [incomingtcp] section.

3 Configuration at the application layer
3.1 VM1 PLC station
Open the software SIMATIC Manager, choose network configuration and edit the station model. In this case only OPC server is communicating with PLC station through Industrial Ethernet. The station model is shown in the following figure. The PLC station and HMI OP Station should be equipped with IE General adapter card.

3.2 VM2 HMI OP Station
Open the software OPC Link and configure the OPC sever with the type of SIMATIC.net, then configure the topic names that will be communicated with PLC station.

3.3 VM3 SQL Database Station
Open the file folder in Tetra Pak Plantmaster PLC Pump, and open the file 1_PEConfig.xml, find Module->Config and set it as 2_PPConfig.xml. By doing this 2_PPConfig.xml is set as the configuration file.

Then open the file 2_PPConfig.xml, set OPC sever-> Node as the computer name of the HMI OP station, for example in this case it is ‘IO01’, and set the OPC server-> Name as ‘OPC. SimaticNet’. By doing this the OPC sever on VM2 has been pointed to the SQL database station.

After these steps an application should be able to run on this network. Application can be initiated in HMI OP Station, and status of the PLC Station can be recorded into SQL database.