



Comparison of Applications Performance between PA and HA: using software engineering parameters

Kirandeep Kaur

Research Scholar, Department of Computer Engineering
Punjabi University, Patiala
India
kirandeepgogna@gmail.com

Gurjit Singh Bhathal

Department of Computer Engineering
Punjabi University, Patiala
India
gurjit.bhathal@gmail.com

Abstract - Operating System itself is very complex software and when application software(s) are installed over it, it become more complex to manage. Therefore, a way out must be used so that the OS remain intact and the work of user also remain running without any conflict. This paper offers a path which can lead to better utilization of OS without increasing the complexity of usage. There is a discussion of other IT problems also which leads to increase in complexity and mesh ups and an answer to reduce the actual working time for the betterment of the society. The solution provided here will help the IT organization as well as the end user for managing their PCs, servers etc, and give a clean environment for all to work.

Keywords - PA- Portable Application, HA- Hosted Application.

I. INTRODUCTION

Today is the time where a child of only one year can operate an iPhone. How deep the world of IT has affected our lives. Today everyone have smart phones and each one of us wants to get the work done quickly and easily. So the researchers are day by day working on the process to make your lives easy and comfortable. And in today's era the life will be easy only when all the resources are working in order. What if, your work hinders because of system crash and the crash is due to some virus. This mars the user experience. You reboot your PC, but it doesn't work. Now you have the choice of switching your work to your friend's computer but there you cannot get every application you need and with the same customized preferences and settings you set on your PC, which become an obstacle for your smooth flow of work. Next choice is you start setting up your own PC again, which is a very time consuming and tedious job.

Further, sometimes the job demands to work on two versions of the same application, but any two versions of the same application cannot be installed on the same computer.

In this fast growing world, the need of high availability of applications is increasing. All of us want to have availability of their preferred applications even when we are offline. This all needs to be resolved. And the answer can be PORTABLE APPLICATIONS. By portability I mean an application which can be taken out of one PC and can be launched on any other PC without any difficulty like we take our mobile phones where ever we go. Earlier only land line

phone were available which had limitations as they hindered our mobility and availability but with mobile phones we are connected and approachable to everyone all times. As it is portable and handy we can easily carry it anywhere. Same concept can be applied to software also. So that it is available to us where ever we go whether we are connected to internet or not, whether we are on our own PC or not, same application with same environment.

I. PORTING, PORTABILITY AND PORTABLE APPLICATIONS

Porting means to create an executable version of a system unit, so that it may run in a new environment based on existing version.

Therefore, portability means the ability of a system unit to be ported. In the software world Mooney described two types of portability: Binary portability, means porting the executable form and Source portability, means porting the source language representation. I have studied the Binary portability.

Thus portable applications can be defined as an executable version of a software unit which can be ported to another environment without making any changes in its code.

II. CONCEPT BEHIND PORTABLE APPLICATIONS

Traditional application installations penetrate the operating system and change its configuration. Eventually, managed or unmanaged systems become completely transformed and unrecognizable. For this reason, many organizations continually reimage their desktops over time to reset them to a known configuration.

Modern operating systems attempt to keep programs isolated from each other. If one program crashes, the remaining programs generally keep running. However, bugs in the operating system or applications can cause the entire system to come to a screeching halt or, impede other operations. This is the major reason virtualization has become desirable.

The basic idea behind the portable applications is application virtualization, which is a concept of virtualization. Virtualization is the creation of a virtual (rather than actual) version of something; therefore in computer world it can be an operating system, a server, and a storage device or network resource. It is basically one physical computer pretending to be many computing environments. Further there are 3 types of



virtualization i.e. hardware virtualization, OS virtualization and Application Virtualization. As application virtualization is used to build the portable applications thus only this technique is discussed in detail.

Application virtualization means to virtualize the software and put it in some virtual container so it may not directly interact with the OS. Thus virtualizing an application means re-packaging an application by some means and redistributing this same application under a different outline/layout. The application is fooled at runtime into believing that it is directly interfacing with the original operating system and all the resources managed by it, however in reality it is not. This new “format” is usually a single big file that gets “copied” on top of an OS and that doesn’t need to be “installed”.

Application virtualization protects the operating system from any modifications and supports completely secure environments. Once an application has been virtualized, it no longer needs to be repackaged each time you need to change the OS.

The end results are that the underlying file system and registry settings are never changed applications no longer create fracas and the base operating system remains intact. This allows running diverse and potentially irreconcilable applications on the same Operating System without each application stepping over each other due to DLL conflicts or registry incompatibilities. This is because these applications are basically shielded and are distributed as single huge file that contain everything like DLL’s, custom registry entries etc. Example of products and technologies that implement this concept are Microsoft SoftGrid, ThinApp etc. This concept have the ability to dynamically stream applications to a user which provides flexibility, faster deployment, and greatly reduced IT labor required to deploy and update applications.

Application Virtualization is definitely going to be very applicable in the personal productivity (i.e. PC) environments where you have heterogeneous GUI applications to run and where the local end-user OS requirements are: ease of use and flexibility. For this reason alone, Application Virtualization is one of the most powerful new technologies IT is fast adopting.

III. ROAD MAP FROM APPLICATION VIRTUALIZATION TO VIRTUALIZED/PORTABLE APPLICATION

With help of application virtualization technique a virtualized application can be made, which is an application which contains all the necessary system software packages (also called just enough OS or “jeos”) and the application software inside a virtual machine container which can launch directly without installation. Fig 1 below shows the concept of portable application. From now onwards virtual application, virtualized application and portable application will be used interchangeably.

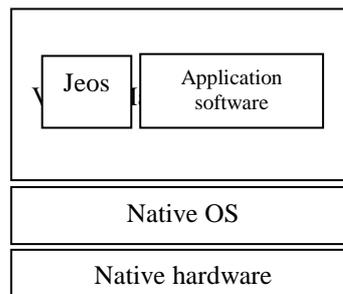


Fig 1: - Architecture of Portable Application

Now it can be stated that Portable application is transportable and adapts every platform where it launches. It do not require anything extras in order to operate. Thus Portable application which can also be called standalone, is a program designed to run on a compatible computer without being installed in a way that modifies the computer's configuration information. This type of application can be stored on any storage device, including internal mass storage and external storage such as USB drives or Cloud storage for that matter for storing its program files and any configuration information and data on the storage medium alone.

The complexity of implementation of portability depends on the operating system. Portable apps are distinct from software portability, as in which different source code is written to be compile-able into different executable programs for different computing platforms. A portable application does not leave its files or settings on the host computer, which may be convenient or required if you have no administrator privileges on a computer. Typically the application does not write to the Windows registry or store its configuration files (such as an INI file) in the user's profile; instead, it stores its configuration files in the program's directory. Also the storage of portable app is in relative format, since file paths will often differ on changing computers due to variation in Windows drive letter assignments. Thus the application virtualization layers replace part of the runtime environment normally provided by the operating system. The layer intercepts all file and Registry operations of virtualized applications and transparently redirects them to a virtualized location, often a single file.

In addition, the deployment process requires less time because the applications do not need to be configured for different types of hardware. Even a mandatory reboot that a software installation may prescribe is no longer needed and thus saves time for administrators and users. As all changes are captured inside the virtual environment, a virtualized application cannot harm other virtualized or natively installed software. Even the operating system with all its core libraries stays untouched.

As the virtualization of applications does not require creating an abstraction for the hardware components, it is very resource-conserving.

The virtual container allows application to read or modify files that were placed into the file system at the time of installation. Because of the isolation, these files are not



located at the same place on the client. Thus, the virtual layer has to make the application believe that all files reside where it expects them. Thereto, it has to merge the files with the local file system or it has to override existing ones that are different to the ones of the virtual application. Also the virtual layer allows the portable application to read and write the registry like if it was installed locally. But instead, the changes have to be saved in a separate place to keep the real one clean. If an application creates a new file, the layer has to decide if it is a part of the virtual application and thus needs to be captured and isolated as well. On the other side, the file could be a document that needs to be saved to the native file system which means that the layer must not redirect this request to the virtual counterpart.

Thus we have seen that the virtual applications are quite different from hosted or installed applications. Thus I have compared the portable application with the hosted applications. The example taken is Portable MS Office 2007 and Hosted MS Office 2007 where ever necessary. This research can help the end users as well as the IT organizations to manage their PC's in a better way.

IV. PARAMETERS FOR COMPARISON

As described in Pressmen's book if Software Engineering there are two types of metrics viz:

1. *Process metrics* measure aspects of development and maintenance processes, and are used to describe resources used such as dollars, equipment, person-days, etc. or production measures such as lines of debugged code, CPU utilization, memory usage, I/O usage.
2. *Product metrics* measure attributes of a software unit or system itself, and is used to characterize the "quality" of that software i.e. it measures maintainability, flexibility, testability, portability, reusability, interoperability, and reliability.

V. COMPAIRING HOSTED AND PORTABLE APPLICATIONS IN PROCESS METRICS

The very first issue is of course the cost incurred for developing the hosted application and the portable application. To develop a hosted application one has to start from scrap again to develop each and every module. Thus a lot of hard work and challenges has to be faced even if some of the modules are reused. The developer than sells the copies of the software developed and one has to build a strong support network which is very cost intensive. However in case of a portable application the developer itself is free from the fear of pirated copies, extensive support system. The developer can very well monitor and analyze its user base and take necessary steps to expand the same. Many tools are available for developing portable applications and deploying the same on various machines with heterogeneous hardware.

Even in case of virtualized application being not on cloud the developer can very well distribute its software through secured storage devices. Clearly, from above the developer saves a lot on piracy cost and support network cost.

The next metrics is the utilization of CPU and memory. As portable software unit is not finely tuned to a particular environment, and may exhibit lower performance or less efficient resource usage than a tuned implementation. Some authors have compared (fig 2) between the native (hosted) and virtualized applications and found that there is slight overhead when the virtual application is launched i.e. due to the virtual layer which is induced between the application and the OS. Also the memory and I/O usage is also complex and more utilized. Formal tests have been conducted to show the resource requirements of virtual application are somewhat more than hosted application.

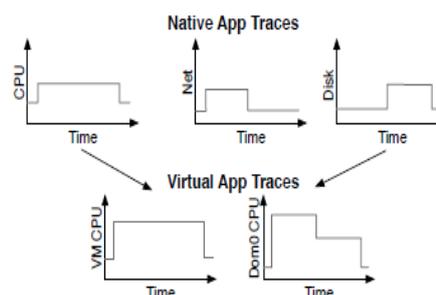
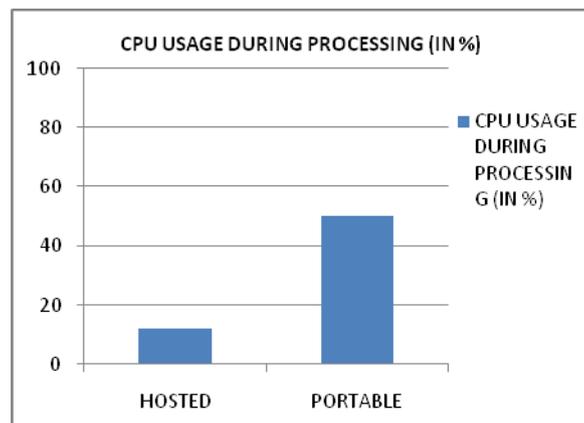


Fig 2: Difference between the resource needs of native (hosted) application and virtual application

This is because of the fact that virtualized application container need to take the control over the resources indirectly, as it is not in direct contact with the OS. Thus a long communication path between the application and the hardware exist, also the hardware drivers are not in direct link with portable application thus performance degrades.

I have also measured the CPU usage and RAM usage and the results are shown in fig 3 & 4 below. As the portable application is not to be or rather never to be installed on the machine it requires higher CPU usage percentage while the app is launched. This performance for hosted application is better however it is to be kept in mind that during installation of the hosted app huge amount of CPU and Ram had been used.



cache and RAM apart from numerous registry entries in the OS which in-turn affects the system performance for all types



of its uses. On the other hand the portable app works in a virtual bubble without affecting the system registry entries and the physical storage.

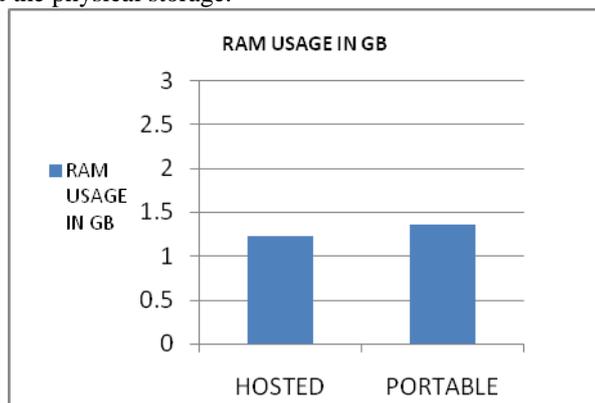


Fig 4: Graph showing RAM usage

The above graph (fig 4) shows that RAM usage of both is nearly the same. Also I have checked it for over around two months using a PC with hosted applications i.e. everything is installed over the OS with number of registries and another with clean PC i.e. nothing is installed over the OS only the virtualized portable applications were used. And it was observed that, the PC with no extra installation over the OS performs better than the 1st PC where there are several entries of registries.

VI. COMPAIRING HOSTED AND PORTABLE APPLICATIONS IN PRODUCT METRICS

Another face of the coin which people usually ignores before but realizes afterwards is the non-functional requirements. The product metrics explains the basic needs of a user in a product except its technical requirements. Product metrics covers the reliability, performance, portability and maintainability of an application. But there is no clear way out to express non-functional requirements. However, if such requirements exist they must (presently) be expressed by less formal means. Here I have tried to measure those non-functional requirements of a portable application and compared the same with hosted applications also, which may aid in making effective portability related decisions.

The results observed are as follows:

1. The biggest flaw in hosted application is failure and recovery time, which is much more in comparison to the slight degradation in performance when portable application is launched.
2. Next good reason to use portable applications is that they are not affected by virus, Trojan and worms, as these are packed in a special container which contains all the administrative specifications and executable files. Thus it is not direct contact with the OS and any other software which is running at the same time. So performance on this side increases.
3. As the portable applications are packed in a container, these don't change any registry of OS, thus when the

work is complete and the portable application is closed, the PC remain intact as if no such application ever launched on that PC. On the other hand, when a hosted application is installed it changes the registry, thus burdening the OS performance; when it is closed it saves its data in the form of registry and cache. And when the number of application installations increases the OS become more complex to handle, which lead to failure or bad performance.

4. Because the portable applications use the virtual container in which the run their executable file thus more than two versions of the same software can be launched, which is not possible in hosted applications.
5. Time required to install a hosted application and to launch a portable application is shown in the fig 5 below.

TIME REQUIRED FOR INSTALLATION IN MINUTES

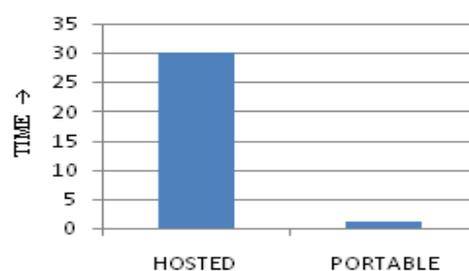


Fig 5: Showing time required to install an application

The chart above clearly shows that if we tend to install a hosted application it will take a long time for the complete process, after that also it might need a reboot to start its working in normal condition. On the other hand a portable application needs less than a minute to launch without any reboot.

6. The table-1 shown below clearly indicates that with different versions of hardware, OS and even with any other version of the application installed, the portable application launches without any problem. However, on the other hand if one tries to install the different version of Application on hosted machine already having some other version of the same application installed, the same won't get installed without replacing the already present version.



CPU	OS	MS OFFICE VERSION INSTALLED	HOSTED MS OFFICE 2007	REMARKS	PORTABLE MS OFFICE 2007	
Intel Core2 Duo	32bit, Windows 7 professional	2007	✓	No need for any new installation.	✓	
Intel Core2 Duo	32bit, Windows 7 Ultimate	2007	✓		✓	
Pentium®4	32 bit, Window XP Professional	2007	✓		✓	
AMD, hp	64 bit, win 7 professional	2013, 64 bit	×		The installation wizard prompts user for replacing the present version of MS Office.	✓
Pentium®4	32 bit, Window XP Professional	2003, professional	×			✓
i5, 4g	64 bit, Window8.1	2010	×			✓
Intel Core2 Duo	32bit, Windows Vista	2003	×			✓
2g, Intel Core, i3	64 bit, Window 8 Enterprise	2013	×		✓	

Table 1: Comparison between hosted and portable application

This practice shows that Portable Applications are flexible, reliable and portable. Which are the major factors required these days. Thus, these apps can be taken anywhere and can be launched on any Window based platform, with any hardware specifications and any other version of the same application may be installed, it will work.

- When the hosted application is installed, and the portable application is also on the same PC or the USB is ready to work, then the launching time of both the applications are same (shown in fig 6).

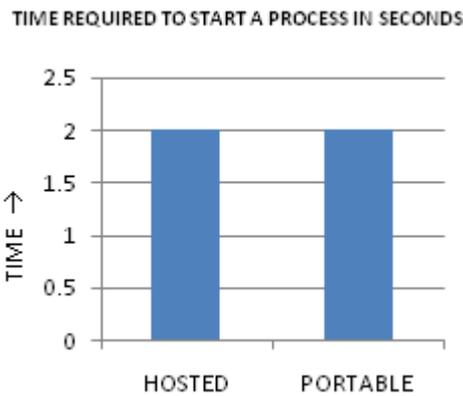


Fig 6: Showing time required to start the process

The result of above chart has been proved by noting the time of launching of both hosted and portable application on different platforms. The table-2 is shown below:

CPU	OS	MS OFFICE VERSION HOSTED	Time to launch hosted application (in seconds)	Time to launch portable MS OFFICE 2007 (in seconds)
Intel Core2 Duo	32bit, Windows 7 professional	2007	2	2
Intel Core2 Duo	32bit, Windows 7 Ultimate	2007	1	1
Pentium®4	32 bit, Window XP Professional	2007	2	2
AMD, hp	64 bit, win 7 professional	2013, 64 bit	2	2
Pentium®4	32 bit, Window XP Professional	2003, Professional	1	1
i5, 4g	64 bit, Window8.1	2010	2	1
Intel Core2 Duo	32bit, Windows Vista	2003	1	2
2g, Intel Core, i3	64 bit, Window 8 Enterprise	2013	2	1

Table 2: Comparison of launching time of hosted and portable applications.

Same will be the case even when the application is present on cloud assuming that there is no limitation of bandwidth and internet speed, the application will work at same speed.

VII. CONCLUSIONS

In these fast paced times of growing world of technology one needs the work to be done accurately, speedily and smartly, but without much effort. So portable application is the right answer as the IT organizations also need not to put much effort once the application is ready. All the updates, improvement patches, bugs etc can be addressed in one go and made available to whole of the user base. However, in hosted applications, the developer has to maintain an extensive support system addressing to the user issues.

The portable application has agent-less architecture i.e. it do not need any agent on the target devices to run and is designed for fast deployment and ease of management. It provides a complete application isolation single executable that runs independently on any endpoint, allowing multiple versions or multiple applications to run on the same device without any conflict. It has built-in security as the application package run in User Mode only, even in locked-down PC environments. Over and above the performance optimizations of Portable Applications is highest as it run directly from a compressed state with block-by-block execution streaming to improve launch speed and runtime performance. These applications follows the concept of package once, deploy to many. Due to the concept of application virtualization, these applications can also run in environments that do not suit the hosted application. Further application virtualization offers an opportunity to deploy and manage applications in ways that are just not possible with traditional Application Delivery. If only portable applications are used they will protect the operating system and itself from other from poorly written or buggy code and provides memory protection too. Also the performance of the system will remain intact with time i.e. it



will not degrade. Portable Applications uses fewer resources than a separate virtual machine. It can be deployed on a virtual cloud and then imported to client computers without need of installing. As the portable apps are in a virtual container thus incompatible applications can run side-by-side, at the same time and with minimal conflicts against one another. Portable applications will reduce system integration and administration costs by maintaining a common software baseline across multiple diverse computers in an organization. These also support multi-tenancy as multiple users can use these applications at the same time. This software is linearly scalable i.e. it can be copied to other PCs and used side by side without conflict, it can be put on cloud and can be provided to n-users at the same time. It offers the SLA driven approach, i.e. at peak loads also it will work at the same acceptable performance. Because it is virtualized it is decoupled from the underlying OS and hardware, more over it uses the deployed mode of execution thus no limitations of network bandwidth or availability of internet services. The concept of virtualization makes portable applications flexible, isolated from other software, reliable. The long wait of program installations have also reduced by portable applications to just a click of mouse button. The deployment of software is simplified because a virtual application consists of one single file that includes everything the program requires. Normally, companies have a complex application lifecycle management that involves much time for regression testing. With application virtualization the management is simplified because the potential risk of conflicts is reduced and thus crashes that may be the result of such problems are avoided. Certainly, with the isolation the removal of old version is not necessary anymore. If products have to be available in different versions on the same machine, application virtualization will make this scenario possible because program files, libraries or unequal dependencies cannot conflict with each other. As all changes to a virtual application are saved in the user's roaming profile, the configuration settings will follow the user from one workstation to another. This is a huge difference to natively installed programs where some of them are not capable of a multiuser environment with network clients and store all settings on the local machine. Through the separation of the application and the user's adjustments, the support becomes easier for help desks and administrators because the application can be reset to its original state if a user experiences problems with his application. Also, the piracy eats into a lot of revenue of the hosted app developers; on the other hand, piracy is completely managed with portable apps.

REFERENCES

- [1] Rabi Prasad Padhy, Manas Ranjan Patra, Suresh Chandra Satapathy (December 2011), Virtualization Techniques & Technologies: State-Of-The-Art, Journal of Global Research in Computer Science.
- [2] Kyong-I Ku, Won-Hyuk Choi, Moonyoung Chung, Kiheon Kim, Won-Young Kim, Sung-Jin Hur (2010), Method for Distribution, Execution and Management of the Customized Application based on Software Virtualization, The 12th International Conference of Advanced Communication Technology.
- [3] Gurjit Singh Bhathal & G N Singh (2010), A Comparative Study of Application Portability with Virtualization Softwares, International Journal of Computer Science & Communication.
- [4] Timothy Wood, Ludmila Cherkasova, Kivanc Ozonat, and Prashant Shenoy (2008), Profiling and Modeling Resource Usage of Virtualized Applications, seelab.ucsd.edu.
- [5] John Casey (2003), A Portable, Cross-Platform Emulation System, Proceedings of the First Australian Undergraduate Students' Computing Conference
- [6] James D. Mooney, Bringing Portability to the Software Process.
- [7] Marcia Kaufman (2011), Creating a Sense of Application Awareness in IT Virtualization Environments, A Hurwitz Research Report
- [8] Gary Chen [January 2010], Virtualizing Tier 1 Applications: A Critical Step on the Journey Toward the Private Cloud, Sponsored by: VMware, IDC Analyse the future.
- [9] VMware Company (2009), VMware ThinApp Deploy and run applications without conflict, White Paper.
- [10] Michelle Bailey (November 2009), The economics of Virtualization: Moving Toward an Application-Based Cost Model, White Paper, sponsored by VMware.
- [11] Michael Rose et al (2008), Agent less Application Virtualization: Enabling the Evolution of the Desktop, White Paper, IDC, and Sponsored by VMware.
- [12] VMware ThinApp Agentless Application Virtualization Overview, white paper by VMware.
- [13] Virtualization Overview, white paper by VMware Company.
- [14] Aaron Black (2011), Streaming Execution Mode (Application Streaming with VMware® ThinApp™), information guide.
- [15] Aaron Black (2010), VMware ThinApp (A Guide for Enterprise VMware ThinApp Deployments), reference architecture.
- [16] Aaron Black (2009), VMware ThinApp Deployment Guide (A guide to Virtualizing Applications using VMware ThinApp), deployment guide.
- [17] Aaron Black (2009), Application Registration with VMware® ThinApp™, information guide.
- [18] Tolly group companies (2009), Application Virtualization Comparison, test report.
- [19] Sven Huisman, Matthijs Haverink (2008), VirtualFuture.Info (Application Virtualization Comparison Chart).
- [20] James D. Mooney, Issues in the Specification and Measurement of Software Portability. James D. Mooney in his paper Portability and Reusability: Common Issues and Differences.