ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)
Vol. 3, Issue 1 (Jan. - Mar. 2015)
International Journal of Advanced Research in Computer Science & Technology (IJARCST 2015)

User-friendly Virtual Operating System

[I]Chinmay Desai, [II]Devdatta Deshpande, [III]Sourabh Honrao, [IV]Amit Kadam
[I,II,III,IV]Dept. of Information Technology, Savitribai Phule Pune University, AISSMS IOIT, Pune

Abstract
First time Linux users face various complexities such as Linux is not quite ready for prime time as it is difficult to install and configure. Various Windows application is not supported by Linux. There is a smaller selection of peripheral hardware drivers in Linux. Linux currently does not support Blue-ray discs. Too many package managers make Linux hard to learn and become master. Interoperability with other platforms. Hence this operating system is difficult to use .The above listed complexities make this OS difficult to use. To overcome these difficulties we are proposing a Virtual Operating System (VOS). The application is very light in terms of memory consumption and resource utilization. It is also very user friendly and easy to perform operations on. The application loads within few seconds and has no problems like RAM sharing associated with it. Also the application is, economically, technically and socially practical.

Keywords
Interoperability, VOS, RAM, Resource Utilization

I. Introduction

Virtual operating system is a concept in which we can use more than one operating system at the same time. All the command being used in windows platform will get executed on Linux platform, hence new user don't need to learn Unix command and he can use Unix as efficiently as he use the windows. So in future any user can use Unix without any fair in mind and can do his work without losing time.

This system will be used in training phase of all the company where all new comers are not aware of Linux if any user feel difficult to use Linux at that place it will applicable as substitution. Also no kind of special training is needed for it. Also the new user who doesn't want to use Linux can start with this software. Migrating operating system instances across distinct physical hosts is a useful tool for administrators of data centers and clusters. It allows a clear partition between hardware and software, and makes easy fault management, load balancing, and low-level system maintenance. By accomplishing the majority of migration while OS's stand to run, we achieve magnificent performance with minimal service downtimes. We establish the migration of entire OS instances on a commodity cluster, reporting service downtimes as low as 60ms.

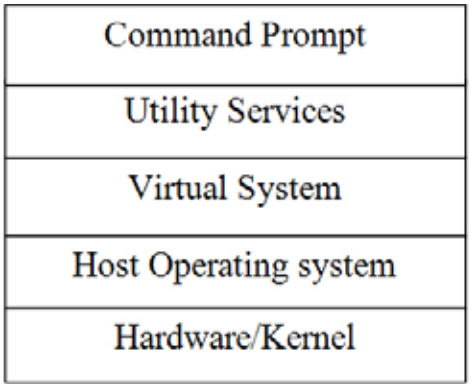

Fig.1: System Architecture Diagram

II. Proposed System

At a high level we can consider a virtual machine to encapsulate access to a set of physical resources. Implementing live migration of these VMs in a clustered server environment leads us to focus on the physical resources used in such environments especially on memory, network and disk. This section summarizes the design decisions that we have made in our approach to live VM migration. We begin by outlining how memory and then device access is moved across a set of physical hosts and then go on to a high-level description of how a migration advance takes place.

When migrating a live operating system, the most compelling impact on service performance is the overhead of coherently transferring the virtual machine's memory image. As remarked previously, a simple stop-and-copy approach will achieve this in time proportional to the amount of memory designated to the VM. Unfortunately, during this time any running services are completely unavailable. A more attractive alternative is pre-copy migration, in which the memory image is relocated while the operating system (and hence all hosted services) continues to run. The drawback however, is the wasted overhead of relocating memory pages that are subsequently modified, and hence must be relocated again. For many workloads there will be a small set of memory pages that are updated very commonly, and which it is not worth attempting to maintain coherently on the destination machine before stopping and copying the remainder of the VM

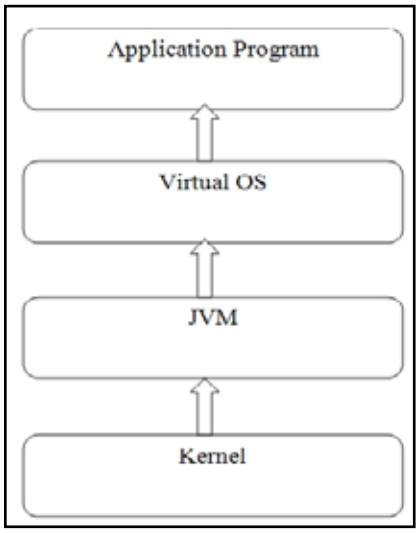

Fig.2: Block Diagram of VOS

III. Functionality

A real operating system presents three principal interfaces to its users. the virtual machine or operating system primitives accessible through programming languages; the utility programs such as compilers, linkers, and editors, and the command language

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

or means by which users access system resources from a terminal. Most system services are available through one or more of these interfaces. The idea of a virtual operating system is to provide standard versions of these interfaces, based on organizational requirements. Possible applications include data management environments, office instruction environments, real-time process control environments, and program development environments, to name some.

Once the three interfaces are specified, implementation consists of:

1. Choosing one or more programming languages.
2. Developing run-time libraries or extending the selected programming languages to support the chosen virtual machine on each target system.
3. Implementing the utilities and command language in one or more of the selected programming languages, relying on the virtual machine to interface to the target operating systems.
4. Writing the necessary documentation.

The virtual machine is a highly idealized set of primitive functions geared to organizational programming requirements. It carries almost no functional similarities to the underlying hardware which actually performs the work.

A virtual operating system becomes a real operating system when the associated virtual machine corresponds to a physical machine. However, the emphasis in building a virtual operating system is on the interface presented to the user.

IV. Conclusion

Using the virtual operating system approach, uniformity can be achieved at the three principal levels of user interface-the virtual machine, the system utilities, and the command language. For at least one realization of the virtual machine interface, the functional equivalence of vendor operating systems has been established. Although the effort to install a virtual operating system is large when compared to the effort required when moving a single program, it is small when compared to the cost of moving an organization's software. Moreover, when personnel retraining costs are considered, installation costs are insignificant. The approach permits accurate estimation of the cost of moving to a new system. The cost of moving people are zero, and the cost of software is equal to the cost of implementing the virtual machine. The question of machine efficiency can also be addressed. By anticipating bottlenecks in resource utilization, critical functions can be isolated and solutions incorporated in the architecture of the virtual machine. This permits the benefits to be shared by all software.

V. Acknowledgment

This acknowledgement is a conscious attempt to be a thanks giving gesture to all those people who have been involved directly or indirectly with our work.

First and foremost, we express our special thanks with gratitude and great respect to our valuable guide Prof. Riyaz A.Jamadar for his keen interest in our case study, fruitful suggestions and valuable guidance with motivation and constant encouragement to complete our project work.

We owe a great debt to our project in-charge Prof. M.K. Pathak, who guided us through the maze of details associated with our project. We are thankful to Principal Dr. P. B. Mane and Head of Department Prof. P.A. Patil for the ready availability of college facilities as and when required. Lastly we are deeply grateful to everyone who has been associated with our project.

References

[1] "Virtual Operating System" Philip H. Carns Parallel Architecture Research Laboratory Clemson University, Clemson, SC 29634, USA

[2] "Secure Virtual Machine Execution under an Untrusted Management OS" By Chunxiao Li (Department of EE), Anand Raghunathan (School of ECE), Niraj K. Jhav(Department of EE).

[3] "A Survey of Virtual Machine System: Current Technology and Future Trends" By Yunfa Li, Wanqing Li, Congfeng Jiang Grid and Service Computing Lab, School of Computer Science and Technology Hangzhou Dianzi University, 310018 Hangzhou, China

[4] "An Introduction and Study to Tiny Operating System (T.O.S.)" By Balram Yadav Reader CSE Dept. MITM Ujjain , Vishwas Karhadkar Lecturer CSE Dept. MITS Ujjain, Yodesh Kakde Lecturer CSE Dept. MITM Ujjain.+

[5] "Design of ARM Based Embedded Operating System Micro Kernel" By: Bo Qu School of Mathematics and Information Technology Nanjing Xiaozhuang University Nanjing, China Zhaozhi Wu School of Mathematics and Information Technology Nanjing Xiaozhuang University Nanjing, China

[6] "Impact of the host operating systems on virtual machine performance," By: Martinovic´, G.; Balen, J.; Rimac-Drlje, S. MIPRO, 2010 Proceedings of the 33rd International Convention, vol., no., pp.613, 618, 24-28 May 2010

[7] Thomas E. Anderson, Michael D. Dahlin, Jeanna M. Neefe, David A. Patterson, Drew S. Roselli, and Randolph Y. Wang. Server less network files systems. In Proceedings of the Fifteenth ACMSymposium on Operating Systems Principles, pages 109–126. ACM Press, December 2010.

[8] Rajesh Bordawekar, Steven Landherr, Don Capps, and Mark Davis. Experimental evaluation of the Hewlett-Packard Exemplar files system. ACMSIGMETRICS Performance Evaluation Review, 25(3):21–28, December 2011.

[9] Peter J. Braam. The Coda distributed file system. Linux Journal, #50, June 2013.

[10] J. Bruno and P. Cappello. Implementing the Beam and Warming Method on the Hypercube. In Proceedings of the Third Conference on Hypercube Concurrent Computers and Applications, January 2013