Enabling Ubiquitous Workplace Through Virtualization Technology

Carlos Roberto de Oliveira Junior
Department of Informatics
Instituto Federal de Educação, Ciência e Tecnologia (IFRJ)
Rio de Janeiro, Brazil
Carlos.roberto@ifrj.edu.br

***Abstract*—In a ubiquitous computing world, users would always have the right information, in the right form, within the right context. In addition, they would like to manipulate their information with their preferred tools. In this paper, an approach based on the advances in virtualization that can change state of art in ubiquitous computing is proposed. Also, how this change can be achieved, showing what has been done and the open issues, is explained. The goal of this paper is not to solve these issues but to give an insight on how the Virtual Machine (VMs) technology can be used towards materializing the ubiquitous computing dream. We concluded that, in a near future, it will be possible to enable ubiquitous computing.**

Keywords-virtualization; ubiquitous; workplace.

I. INTRODUCTION

Computers prices are getting down each year, so a person can have more than one machine (PC, notebook, tablet, etc.) [1]. In addition, nowadays many people use another computer at work. This leads to different information and applications in these many computers. Nevertheless, it is desirable to have the same information in all computers.

According to Oulasvirta and Sumari [2] there are 3 main problems while having multiple machines: 1) the physical effort demanded by various management tasks, 2) anticipating what data or functionality will be needed and shared, and 3) to align these efforts with work, mobility, and social situations.

The ability to have data anytime/anywhere can also be useful for companies as they can have their applications servers (running in VMs) “moved” to anywhere at anytime. Having everything anytime/anywhere is an old dream. The term *ubiquitous computing* has been used first in 1988 by Mark Weiser who published the paper *The Computer for the 21st Century* [3] in 1991. To achieve this dream, many problems needs to be solved. Instead of have isolated computers we need to have them integrated in a transparent way to the user; it can be called as an *user interface* problem.

We need also to have in mind that to achieve gracefully this desirable ubiquitous computing we need to integrate all computers and not only propose a solution to integrate a new kind of computer. It is necessary to regard existing ones.

The concept of *ubiquitous workplace*, introduced in this paper, is a service that allows users to have their data anytime/anywhere in a transparent way. In this service, user's data is stored in the cloud. This data is his preferred OS and configured applications. Users can use any computer to access their data. At the moment, an user logs in a computer, his data would be downloaded to this computer. In our proposal, user's data is represented by a VM and each user has a VM.

Using VMs is possible to carry a whole OS and user's profile from a physical machine to another machine in the same network in a few seconds and even without stopping services running on these VMs [4]. It is also possible to carry VMs across Wide Area Network (WAN) [5]. Moving a VM to anywhere is important because the user can carry all his profile doing it. This is better than moving only files because there are files that need specific applications to be opened. It is also possible to stop tasks in a machine and resume it in a different machine. This is an important feature because it solves the backup problem. Having all profile in a VM, the user needs not to worry about doing backups because all changes are directly applied to the VM. Moreover, the user needs not to worry about reinstall a system when buying a new machine. Running the VM in the new machine, the user will be carrying the whole system to the new computer.

Our purpose is not to solve problems in ubiquitous computing, but we intend to propose a new manner to advance the state of art using VM technology, showing what has been done and the open problems. We introduce the concept of *ubiquitous workplace*. An ubiquitous workplace consists of a VM that can be accessed independent on the physical machine the user is current using. This VM contains any OS and user applications, like it would be if the user was using an OS installed directly on a physical machine. Our goal is to allow the user to access his whole profile with all his data and configured applications anytime/anywhere without worry about the physical machine current being used. The machine is only a way to access the ubiquitous workplace.

The proposed architecture consists of three main agents: clients, client's file system and VM repositories. Clients are ubiquitous workplaces users. A client uses any computer to access his ubiquitous workplace. Clients need not to worry about the underlying OS, installed in the machine currently being used. A prerequisite is that computers need to have a virtual machine monitor (VMM) installed to allow clients to run VMs. Clients VM's are stored in a VM repository. This repository can be a cloud service like Amazon EC2 [6]. Client's VM are downloaded to the machine the client is currently accessing at the time he logins. When a client

 ISBN: 978-1-61208-257-8

finishes his work, his VM is updated to the repository. Later on, his VM can be migrated to another machine.

In this paper, we discuss five requirements that must be addressed and integrated in an unique support to allow a complete implementation of ubiquitous workplace. In the future, when we have these five requirements addressed, it will be possible to implement an autonomous system in which the user workplace is automaticaly moved from a physical machine to the physical machine the user is currently using. In previous works [42] [43], we have already implemented an autonomous system in which VMs (containing web servers) are moved from a physical machine to another according to resource usage constraints. This implementation could be adapted to manage the user workplaces.

The rest of the paper is organized as follows. Section II describes the basic requirements for the implementation of our proposal. Section III points out some scenarios in which our proposal could improve user experience with an ubiquitous workplace. Section IV presents related work and Section V concludes the paper.

II. REQUIREMENTS

Our proposal is based on virtualization that is a consolidated technique. But some essential issues and their related requirements need to be addressed in order to allow a complete implementation of our proposal. In the next Subsections we present these requirements and discuss the solutions proposed by researchers to address each of them.

A. *Creating a VM from an installed system*

In our proposal, users need to have a VM. But we need to take into account those users that do not use virtualization and have already their systems installed in a physical machine and their profiles, files and applications installed and configured on the physical machine. We cannot propose to users to leave their systems and to start a new installation into a VM. Thus, the first requirement is to allow users to create their VMs starting from their installed systems. To do this, we need to have a service that allows users to submit their systems and the service returns a VM containing this system.

Some tools like VMware Converter [7], that is a free product, provide this service. The process to create a VM, using products like this is (a) The user has an installed OS (note that an user can have any OS) containing his applications. (b) User uses an application to create a VM containing his system. (c) The application creates a VM with his whole system. This VM can run in any computer that supports the VMM the VM was created to, even if the VMM are installed over a different OS. For instance, suppose that OS in user's VM is Windows. This VM can run in a computer that has the same VMM installed over Linux.

An issue here is that, sometimes, the OS the VMM is running over does not offer all the capabilities the guest OS offer. Thus, the user can face some troubles when trying to use a capability that is not supported by the host OS.

B. *Migrating VMs in WANs*

As we want an ubiquitous workplace, we need to be able to move VMs to outside of LANs. Migrating a VM in WANs have been studied and some authors have published their work and contributions.

Bradford et al. [5] have studied VMs migration in WANs. Their contribution is the design and evaluation of a system that enables live migration of VMs that (a) uses local storage, and (b) have open network connections, without severely disrupting their live services, even across the Internet.

When the migration is in a LAN, the VM's image (its file system) is not moved from the source host to the target host. The VM's image is stored in a Storage Area Network (SAN) or in a Network File System (NFS). Thus, only the memory state needs to be migrated. However, as we want to access a VM anywhere we cannot be limited to use SAN or NFS. Bradford et al. [5] use Xen as VMM and they use *block tap* [8] to export block devices. Their system pre-copies local persistent state and transfers it from the source to the destination while the VM operates on the source host. During the transfer it is employed an user-level block device to record and forward any write accesses to the destination, to ensure consistency. In their experiments, they have complete a migration in the WAN in 68s. Two related issues in the migration context are discussed below.

1) *Keeping open connections*

There are a few number of scenarios in which should be necessary to keep open connections (e.g., when the user's VM is migrated to the cloud with an ongoing download). As described in the following paragraphs, it is possible to handle these situations in the context of an ubiquitous workplace.

For managing migration in a LAN with respect to network it is generated an unsolicited ARP reply from the migrated host, advertising that the IP has moved to a new location. This will reconfigure peers to send packets to the new physical address and, while a small number of in-flight packets may be lost, the migrated domain will be able to continue using open connections with almost no observable interference. Some routers are configured not to accept broadcast ARP replies (in order to prevent IP spoofing), so an unsolicited ARP may not work in all scenarios [4].

When migration takes place between servers in different networks, the migrated VM has to obtain a new IP address and thus existing network connections break. Bradford et al. [5] implemented a redirection scheme in their framework to overcome this by combining IP tunneling [11] with Dynamic DNS [10]. With the help of iproute2 they set up an IP tunnel between the old IP address at the source and its new IP at the destination. Once the migration has completed and the VM can respond at its new network location they update the Dynamic DNS entry for the services the VM provides. This ensures that future connections are directed to the VM new IP address. In addition, they begin forwarding all packets that arrive at the source for the VM old IP address to the destination, through the tunnel. Packets that arrive during the final migration step have to either be dropped or queued in order to avoid connection resets. After the restart of the VM at the destination the VM has two IP addresses: its old one, used by existing connections through the tunnel, and its new one, used by new connections directly. The tunnel is torn down when no connections remain that use the VM old IP address.

2) *Discovering and mapping devices*

We need not only to keep open connections but we also need to discover available devices where the user currently is. An example that shows the importance of discovering and mapping devices is an user who has a printer configured at home. But when the user moves to the office, that configured printer is not available anymore. Thus, we need to discover available printers at the new location and configure them in a transparent way (when it is possible) to the user. We need also to consider that devices may be added or removed anytime. To discover available devices we can use an approach similar to Gaia [11], which is a framework for smart spaces. Gaia consists of a number of different types of agents performing different tasks. There are agents that perform various core services required for the functioning of the environment like device discovery and selection. There are agents associated with devices that enable them to be a part of a particular environment where they can be readily available [12]. We must take in account that should be difficult to implement a system to discover and map devices in an environment where the user is continuously moving. The system can experiment problems like different passwords and authentication methods.

C. *VM storage repository*

As we are proposing the use of VM also for end-users and not only for enterprises, we should have a way to store VMs in an accessible manner for these users. When a company uses virtualization, it relies on NFS or SAN support to store the VMs. When thinking about end-users it is a bit different. These users will likely not need to have their VMs running all the time but probably only when they are working in front of a computer. And so, they probably do not want to leave their machines at home turned on all the time to make possible to migrate their VMs when finishing their works on the machine they use at the office, for example. Thus, the user has two options: (a) leave his machine in the office turned on and when arriving at home remotely accesses his computer at the office to migrate his VM or (b) have another place to store his VM as he cannot leave his machine turned on. The last option sounds better.

With the advance of cloud computing, our proposal can use a popular service like Amazon EC2 [6] to store the VMs and let it accessible anytime/anywhere. Users can have their VMs updated to the cloud when they need to go away, log off, sleep, etc and have their VMs downloaded when necessary. The idea is that after an user stops working, his VM is migrated to the cloud and still there ready to be used when necessary. Other proposals have been done in this area. For instance, a storage access mechanism that supports live VM migration over WAN is proposed in [13]. This mechanism completely relocates virtual disks between source and remote sites in a transparent manner.

An important issue in this requirement is that a VM (that is, the OS and applications) can have dozens of gigabytes. This large amount of data can turn unfeasible the task to move a whole VM through the network. An approach to address this issue is to transfer only the OS image to the machine the user is currently using, while user`s file system (that is, his files) stays in the cloud. Memory pages from the file system storage are transfered on demand to the machine the user is using. It would work as a NFS and will decrease the amount of data transfered through the Internet. It could also be used a file system like Coda [14]. In Coda, Kistler and Satyanarayanan suggest a disconnected operation mode that enables a client to continue accessing critical data during temporary failures of a shared data repository. Their work can be seen as an extension of the idea of write-back caching. Whereas write-back caching has been used for performance, it can be extended to mask temporary failures too.

D. *Independence of VMM*

To advance in the state of art we cannot propose a solution that uses a specific VMM. We cannot force users to use what we want them to use, they must be free to choose their preferred VMM. Ideally, a VM could be migrated across any node with similar machine configuration and granted resources. However, as there are currently several VMMs (e.g, Denali [15], VMware [16], Xen [17] and kvm [18]), the heterogeneity of the underlying VMMs makes it hard to migrate a VM originally running on one VMM to a system that runs another VMM. To address this problem, authors in [19] have proposed and implemented Vagrant, which is a VM migration scheme aiming to migrate VMs among computing nodes even if they are managed by heterogeneous VMMs. To render it practical, Vagrant does not requires changes to the hosted OS and its applications. It requires only minimal changes to the core VMM abstraction.

To support migration among heterogeneous VMMs, the key issue is the semantic gap among different resource abstractions and migration protocols. First, each VMM provides its own abstraction of the underlying hardware resources. Second, the migration software for different VMMs usually has its own format of migration protocols. Third, there are currently several memory migrating algorithms, including stop-and-copy [8], pre-copy [4], push and pull. To address this issue, Vagrant has a common migration protocol and common virtual machine abstraction. It intercepts the migration control and data issued by the source VMM and transforms them into the Vagrant format. On the destination side, the data in Vagrant format is transformed into the format of the target VMM. For memory migration algorithms, Vagrant provides a pool of common algorithms in both VMMs and dynamically selects the migration algorithms according to the types of the communicating VMMs. Until now, Vagrant supports only migration within LAN. Nevertheless, its authors plan to extend it to support migration in WAN. They also plan to add features such as QoS control to make Vagrant more flexible and robust. Using mechanisms like this, we expect that, in the future, it will be practical to migrate VMs without worrying about the particular available VMM.

E. High speed and reliable Internet access

Nowadays, pen drives can store many gigabytes; thus, an user could carry his VM anywhere using his pen drive. But this option is not convenient or comfortable to the user, and it is against the idea of pervasive computing. Thus, we need to have a high speed and reliable Internet access, allowing us to access our VMs in a transparent way. Many companies and universities have gigabit links to access Internet, but we need to have a high speed access to domestic users too. Moreover, we need to have a reliable Internet access. Users must be able to access their VMs anytime. The American government proposed a National Broadband Plan that is an initiative to bring 100-megabit-per-second broadband to 100 million U.S. households by 2020 [20]. This service is already reality in Japan that has the fastest consumer broadband in the world, 160-megabit-per-second service [21]. In Japan, an user should pay only $20 to have a service like this at home.

As exposed in this Section, all the five essential requirements we have identified were successfully addressed by researchers. Nevertheless, they still need to be integrated in a simple solution to users who are not acquainted with the virtualization technology.

III. ADVANCES EXPECTED IN OUR LIVES

In this Section, we show a scenario that would be possible after integrate these five requirements. Consider the following scenario.

"*An user leaves office and afterwards continue working at home*"

During the day at office one can create or modify documents, receive or modify files, install and uninstall applications and would like to have these changes applied to his computer at home. Moreover, imagine a student who has a work in progress. This student can make some progress in his work at university and may want to continue this work at home. This work may be complex, for instance, requiring a specific system configuration. Using a ubiquitous workplace it is possible to continue his work anywhere.

A further advantage of our proposal is that companies and universities do not need to worry about untrustful user's applications. In a virtualized system, a VM runs independently from others. It leads immediately to a safe environment. For instance, if an user has a virus on his VM, this virus cannot be propagated to other VMs because they are isolated from each other.

Taking into account requirement 1, one may have his VM. By requirements 2, 3 and 5, one can store and access his VM with all his configured environment anytime. We need also to consider that the company or university may have a payed VMM while a employee or student has a free VMM at home. Thus, by addressing requirement 4, an user can move his ubiquitous workplace to anywhere. That is, from a computer to another computer independently of the VMM used on each computer.

IV. RELATED WORK

Remote Desktop [30], included with Windows XP Professional, enables users to connect to their computers across the Internet from any computer. The disadvantage of Remote Desktop is that users need to leave their computers turned on all the time. Moreover, this computer is a single point of failure and the user has no choice about the OS.

Windows 7 [31] brings a new feature termed HomeGroup [32] that allows users to have all computers at home integrated. It allows users to use a machine to access a file stored in another machine. For instance, one can view a picture that is stored in another physical machine on his HomeGroup. One can also use a printer and other devices that are connected to other machine. Is even possible to give permissions to files, specifying who can access which files. The disadvantage of HomeGroup is that if the user goes to the office, for instance, he cannot access his files as he could do at home. HomeGroup can be used only for machines in the same network. So, it does not allow users to access their data anytime/anywhere.

Another proposed solution to allow users to access data anytime/anywhere were the Web OSs. One of these Web OS is Jooce [33]. Jooce is a Flash-based web OS and sharing platform. Google has also its proposal to a Web OS. It is termed ChromeOS [34], that aims to allow users to quickly access Internet and have their applications and data stored on the Internet. They start from the assumption that when users are using a computer, they spend 90% of time on the Internet. In our view, Web OSs were not well succeed mainly because they can run only applications that have been written for them. Simply, users do not want to leave their OSs and applications to change to a completely new OS.

f_Desktop [35] is another proposal that aims to support user mobility in ubiquitous computing environment. f_Desktop allows the user to carry his desktop computing environment defined as a set of follow-me applications from computer to computer. f_Desktop uses MobileSocket to migrate applications. Nevertheless, f_Desktop is based on Java, so it is only possible to use applications based on Java.

Intel has proposed the *Personal Server* [36] as a solution to allow users to have data anywhere. Its proposal aims to allow users to carry their data anywhere. *Personal Server* is a device that has no screen. The Personal Server aims to overcome the fundamental limitation of cell-phones, PDAs, and laptops: if they're small enough to carry, then the displays are too small to easily use. Thus, *Personal Server* needs to be connected to another device that has a screen.

The Internet Suspend/Resume [37] is a project that aims to provide mobile computing by combining two technologies: VM technology and distributed file systems.

As far as we know, this was the first work to propose the use of VM technology to allow mobile computing; it demonstrates that the basic support required by our proposal is feasible. Some issues not considered in [37] are explicitly discussed in our work.

Das et al. [38] implemented a system, termed LiteGreen that aims to save desktop energy by virtualizing the user's desktop computing environment as a virtual machine (VM) and then migrating it between the user's physical desktop machine and a VM server, depending on whether the desktop computing environment is being actively used or is idle. This operation allows LiteGreen to save energy during short idle periods (e.g., coffee breaks) and long idle periods (e.g., weekends). In [38], the ubiquitous computing was not achieved as the system is limited to be used in an enterprise environment. As a future work, we plan to implement a system that, similarly to LiteGreen, moves the VM between a cloud service and user's PC whether the user logs in/out a web site that provides our service. Unlike LiteGreen, our proposal requires no changes to the desktop. In our paradigm, is not desirable or convenient to apply modifications to desktops, as the user should be able to use any desktop.

V. Discussion and Conclusion

System virtualization has become a disrupting force in the computer systems and is likely to be the new foundation of system software [39]. A conspicuous feature enabled by system virtualization is the migration of VMs. In principle, by dynamically relocating an entire environment including the operating system from one machine to another, we can achieve the ubiquitous workplace paradigm. The five requirements to support our proposal have already been addressed by many researchers as presented in Section 2. The next step is to integrate these requirements in a comprehensive environment and make it available to be used by people who do not want to know about the underlying technology.

Many proposals with similar goals in ubiquitous computing and related areas have failed mainly because they tried to impose to users their specific solutions or tools. Now, we know that if we want to change the state of art we must propose a feasible solution to the general users and not only for a specific group. In particular, it is not fair trying to convince them or impose to them a particular choice (as Web OS tried to do). In sum, giving freedom to the user to choose his preferred environment is essential. It is described bellow some benefits of our proposal.

- Access to the ubiquitous workplace anytime/anywhere: an user can do all his work on his ubiquitous workplace using a generic machine and when he needs to go to another place he can easily continue his work on another different machine. As a nice effect users would not need to worry about migrating to a new hardware platform, if by any reason they decide to change the machine they keep at their homes or offices. In the future, when the five identified requirements are fully integrated, users will not need to worry about particular computers because they will be widely available. They would only need a way to connect to their ubiquitous workplace.
- Collaborative work: an user can send a workplace to a colleague in another place to continue or evaluate his work. For instance, a workgroup team can have a developer working in a country and the tester working in another country. Instead of sending only the code to the person who will test it (who can have to reconfigure his system to run the application), the developer can send a copy of his workplace with the whole associated environment.
- Data sharing: taking advantage of the cloud-based storage, users could share data (photos, videos, etc) with friends or familiars just sharing their local repositories. This would allow the partners to use their preferred tools (e.g., viewers). This would avoid users the stress of having to download, install and learn to use a specific viewer to see the shared media.
- Backup problem: all modifications the user does in the environment (e.g., by installing programs or modifying data) are immediately applied to the underlying VM, that at any given moment can be reflected in the cloud-based storage.
- Energy savings: as users tend to create huge amounts of data during their lives, by keeping the currently unused part of this data in the cloud can be useful to save energy, as pointed out in [39].
- User habits preserved: many proposals until now were based on new abstractions forcing the user to change his or hers habits. Our proposal is practical and feasible because virtualization allows one to have only one ubiquitous workplace, containing his preferences, programs and data, accessible anytime/anywhere.

We also expect that in the future, vendors will sell machines with the basic virtualization mechanisms required to implement our proposal. Moreover, ubiquitous workplaces will be easily stored in cloud services and users will have high speed and reliable access to them through the Internet. Our expectations are in accordance with Google [40] and IBM [41] that foresee that desktop concept will be obsolete in a few years.

References

[1] D. Dearman, and J. S. Pierce, “It's on my other Computer!: Computing with Multiple Devices,” Conference on Human Factors in Computing Systems, 2008, pp. 767-776.

[2] A. Oulasvirta, and L. Sumari, “Mobile kits and laptop trays: Managing multiple devices in mobile information work,” ACM, 2007, pp. 1127-1136.

[3] M. Weiser, “The computer for the 21st century,” Scientific American, 1991, pp. 94-104.

[4] C. Clark et al, “Live migration of virtual machines,” IEEE, 2005, pp. 273-286.

[5] R. Bradford et al, “Livewide-area migration of virtual machines including local persistent state,” ACM/Usenix International Conference On Virtual Execution Environments, 2007, pp. 169-179.

[6] Amazon, “Amazon elastic compute cloud”, http://aws.amazon.com/ec2/ [retrieved: 02, 2013]

[7] VMware, “Vmware vcenter converter”, http://www.vmware.com/products/converter/ [retrieved: 02, 2013]

[8] A. Warfield, H. Steven, K. Fraser and T. Deegan, “Facilitating the development of soft devices,” USENIX Annual Technical Conference, 2005, pp. 379-382.

[9] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam and M. Rosenblum, Optimizing the migration of virtual computers,” in ACM SIGOPS Operating Systems Review, 2002, pp. 377 – 390.

[10] B. Wellington, “Secure dns dynamic update,” RFC 3007, 2000.

[11] M. Rom´an and R. H. Campbell, “A middleware-based application framework for active space applications,” in Middleware ’03: Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 433–454.

[12] A. Ranganathan and R. H. Campbell, “A middleware for context-aware agents in ubiquitous computing environments,” in Middleware ’03: Proceedings of the ACM/IFIP/USENIX 2003 International Conference on Middleware. New York, NY, USA: Springer-Verlag New York, Inc., 2003, pp. 143–161.

[13] T. Hirofuchi, H. Nakada, H. Ogawa, S. Itoh, and S. Sekiguchi, “A live storage migration mechanism over wan and its performance evaluation,” in VTDC ’09: Proceedings of the 3rd international workshop on Virtualization technologies in distributed computing. New York, NY, USA: ACM, 2009, pp. 67–74.

[14] J. J. Kistler and M. Satyanarayanan, “Disconnected operation in the coda file system,” ACM Trans. Comput. Syst., vol. 10, no. 1, pp. 3–25, 1992.

[15] A. Whitaker, M. Shaw and S. D. Gribble., “Scale and performance in the denali isolation kernel,” TACM SIGOPS Operating Systems, 2002, pp. 195-209.

[16] J. Sugerman, G. Venkitachalam and B. Lim, “Virtualizing i/o devices on vmware workstation’s hosted virtual machine monitor,” USENIX Annual Technical Conference, 2001, pp. 1-14.

[17] P. Barham et al., “Xen and the art of virtualization,” in SOSP’03. ACM, 2003, pp. 164–177.

[18] A. K. Qumranet, “kvm: the linux virtual machine monitor,” in Proceedings of the Linux Symposium, 2007, pp. 225-230.

[19] P. Liu, Z. Yang, X. Song, Y. Zhou, H. Chen and B. Zang, “Heterogeneous live migration of virtual machines,” International Symposium on Computer Architecture, 2008.

[20] PC World, “Fcc’s 100 megabits to the home: What it means to you,” http://www.pcworld.com/article/189507/. [retrieved: 02, 2013]

[21] The New Yor Times, “Worldest fastest broadband at $20 per home,” http://bits.blogs.nytimes.com/2009/04/03/. [retrieved: 02, 2013]

[22] VMware, “Vmware workstation: The gold standard in desktop virtualization,” http://www.vmware.com/products/workstation/ [retrieved: 02, 2013]

[23] Microsoft, “Windows xp,” http://www.microsoft.com/windows/windows-xp/ [retrieved: 02, 2013]

[24] Microsoft, “Windows xp service pack 3 overview,” http://www.microsoft.com/downloads/details.aspx?FamilyId=68C48DAD-BC34-40BE-8D85-6BB4F56F5110n&displaylang=en [retrieved: 02, 2013]

[25] Microsoft, “Windows media,” http://www.microsoft.com/windows/windowsmedia/ [retrieved: 02, 2013]

[26] Microsoft, “Internet explorer,” http://www.microsoft.com/windows/internet-explorer/default.aspx [retrieved: 02, 2013]

[27] Microsoft, “Communicate instantly with windows messenger,” http://www.microsoft.com/windowsxp/using/windowsmessenger/ [retrieved: 02, 2013]

[28] Eucalyptus Systems, “Eucalyptus sytems,” http://www.eucalyptus.com/ [retrieved: 02, 2013]

[29] VMware, “Vmware vsphere hypervisor (esxi): Virtualization made free and easy,” http://www.vmware.com/products/vsphere-hypervisor/ [retrieved: 02, 2013]

[30] Microsoft, “Get started using remote desktop with windows xp professional,” http://www.microsoft.com/windowsxp/using/mobility/getstarted/remoteintro.mspx [retrieved: 02, 2013]

[31] Microsoft, “Windows 7 features,” http://www.microsoft.com/windows/windows-7/default.aspx [retrieved: 02, 2013]

[32] Microsoft, “Windows 7 features: Homegroup,” http://www.microsoft.com/windows/windows-7/features/homegroup.aspx [retrieved: 02, 2013]

[33] Jooce, “Jooce,” http://jooce.com/ [retrieved: 02, 2013]

[34] Google, “Introducing the google chrome os,” http://googleblog.blogspot.com/2009/07/introducing-google-chrome-os.html [retrieved: 02, 2013]

[35] K. Takashio, G. Soeda and H. Tokuda, “A mobile agent framework for follow-me applications in ubiquitous computing environment,” in ICDCSW’01: Proceedings of the 21st International Conference on Distributed Computing Systems. Washington, DC, USA: IEEE Computer Society, 2001, p. 202-207.

[36] R.Want, T. Pering, G. Danneels, M. Kumar, M. Sundar and J. Light, “The personal server: Changing the way we think about ubiquitous computing,” in ACM International Conference on Ubiquitous Computing. Goteborg, Sweden: IEEE Computer Society, 2002, pp. 194–209.

[37] M. Kozuch and M. Satyanarayanan, “Internet suspend/resume,” in WMCSA ’02: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications. Washington, DC, USA: IEEE Computer Society, 2002, p. 40.

[38] T. Das, P. Padala, V. N. Padmanabhan, R. Ramjee, and K. G.Shin, “Litegreen: saving energy in networked desktops using virtualization,” in USENIXATC’10: Proceedings of the 2010 USENIX conference on USENIX annual technical conference. Berkeley, CA, USA: USENIX Association, 2010, pp. 3–3.

[39] S. Crosby and D. Brown, “The virtualization reality,” in ACM Q focus: computer architecture, 2006, pp. 34 – 41.

[40] Google, “Google exec: Desktops will be obsolete in 3 years,” http://www.maximumpc.com/article/news/google_exec_desktops_will_be_obsolete_3_years [retrieved: 02, 2013]

[41] Computerworld UK, “Ibm says desktop computers and phones will disappear,” http://www.computerworlduk.com/technology/networking/voip/news-analysis/index.cfm?articleid=1273 [retrieved: 02, 2013]

[42] C. Oliveira, “Load balancing on virtualized web servers,” in SRMPDS '12: Proceedings of the Eighth International Workshop on Scheduling and Resource Management for Parallel and Distributed Systems. Pittsburgh, USA.

[43] C. Oliveira, “An Architecture for Experimentation with Multi-tier Internet Applications in Virtualized Server Environments,” in WoSIDA’12: Proceedings of the Second Workshop on Autonomic Distributed Systems. Ouro Preto, Brazil: Brazilian Computer Society.