https://doi.org/10.48009/1_iis_2011_357-365

Issues in Information Systems

Volume XII, No. 1, pp. 357-365, 2011

EMPLOYING HOST VIRTUALIZATION AND SYMMETRIC MULTI-PROCESSING AS A STRATEGY FOR IMPROVING PERFORMANCE IN COMPUTENTIONALLY INTENSE PROBLEMS

Paul Safonov, Saint Cloud State University, MN, USA, Safonov@stcloudstate.edu
Dennis Guster, Saint Cloud State University, MN, USA, DGuster@stcloudstate.edu
Corey Hemminger, Saint Cloud State University, MN, USA, heco0701@stcloudstate.edu

ABSTRACT

This manuscript describes how a virtualized architecture can be utilized to help a corporate data center leverage the multi-core architecture which is often utilized in virtual computing. Further, the advantages of virtualization such as energy savings, isolating sensitive data for security purposes, ensuring data reliability and business continuity, while still maintaining sufficient performance are discussed. A case study examining a data center, in which numerous physical hosts have been virtualized into a single computer, is presented. The results indicate that the common multi-cored virtualization process can lead to significant improvement in performance. Because the potential benefits of virtualization in a multi-cored architecture are critical both financially and operationally to any organization operating on a global scale, the results presented herein is encouraging.

Keywords: virtualization, multi-cores, SMP, green computing, data center, high performance computing.

INTRODUCTION

We live in an information society. Most people live and work within the context of information technology [12, 11]. IT enhances leisure time and enriches culture by expanding the distribution of information, relieves pressures on urban areas by enabling individuals to work from home or remote-site offices [8], changes the way we work, and the way we work with one another. In terms of productivity and speed of communications, these changes have been mostly positive [15]. But what have been the costs? IT introduces change that creates new ethical issues. The US business model has recognized the importance of IT (information technology) to obtain a competitive advantage. The advent of the internet and its relationship to a global economy has made keeping pace with technology even more crucial [7]. As a result of this global model IT has had to become responsive to supplying timely computing resources all over the world. The change of reference from just a local to a national then to a global perspective has caused the volume of information to vastly grow and correspondingly the size and complexity of the hardware required to support the enterprise has expanded as well (if a physical computing model is followed). This growth as a result of the need to support worldwide applications has placed a major strain on the data center that supports global IT enterprises particularly from a performance perspective. In the past, the computing model invoked by many data centers utilized a hardware configuration based on many different physical hosts.

The basic logic of this model appears to be simple: one computer hosting one application. However, it often leads to a very substantial number of physical machines being deployed. Accordingly, this situation results in intense resource utilization which requires much physical space, electricity and much time for the system administrator to manage. Often the physical model has been justified to separate applications from one another with each application being housed in a separate physical computer. While this scenario does provide isolation it results in substantial wasted CPU cycles per unit. This problem is further compounded by the fact that each unit needs to be managed independently and this, of course, is not the best use of system administrator's time. Historically, physical separation has been the methodology of choice for a couple reasons such as improving performance and not disturbing well functioning existing applications. From a green computing perspective the cost and overhead of such a model is prohibitive. In fact, a physical model has several disadvantages such as it requires a lot of physical space to store multiple physical servers, the power consumption/heat generated can be substantial and the complexity and time required to manage multiple physical hosts can be problematic as well [20]. To illustrate the need for application isolation an example of some of the common applications or services that are supported in a data center follows. In the classical physical host model it is common to achieve isolation for various services such as DNS (domain name

service), SMTP (email), WWW (web server) and whatever other application servers there might be allocated to separate physical computers.

Virtualizing on the hosts level has proven to be a cost-effective alternative to the physical model [23]. Specifically, they state that virtualization can effectively support individual processes on a single physical machine that is logically divided into separate or virtualized zones. Further, Xu, Zhao, Fortes, Carpenter and Yousif [25], report that virtualization if properly implemented can still provide adequate performance while reducing the physical resources required and still provide the application isolation needed for security purposes. However, to achieve the performance goals it is common to configure the physical host to be virtualized with multiple CPU cores. To gain the fullest advantage of having multiple cores it is necessary to have some means of breaking the offered processes into subparts. To do this SMP (symmetric multi-processing) is often used. This allows regular processes to be distributed among available cores and allows complex jobs that would be handled in serial by a single processor to be reconfigured as a series of light weight processes(analogous to threads) and distributed across multiple processors.

Therefore, because of this multi-cored approach coupled with virtualized zones it is easy to provide application isolation and leverage unused CPU resources from any zone that has excess by implementing a dynamic CPU allocation policy. The work of Boyd and Dasgupta [6] illustrates that virtual operating systems can provide that needed isolation and allow those isolated zones to safely share the physical resources contained on a single physical computer. Further, the work of Singh, Korupolu and Mohapatra [22] has shown that load balancing techniques can be effectively used to effectively leverage unused CPU cycles especially of virtualization is used in a cloud computing architecture. It should further be noted that the limits of copper technology have been reached and CPU clock speeds have not increased dynamically in recent years so the need to rely on a multi-cored approach has been recognized in recent years [9]. Ascertaining the performance of such architectures is not as clear cut as on a single processor in which the clock speed is increased and hence testing performance through experimentation in the data center is needed to address this problem.

VIRTUALIZATION: EFFICIENT AND COST EFFECTIVE

In fact, the concept of a virtualized operating system on a PC (personal computer) architecture is not new, however the desire to promote green computing has led to a wide deployment of the concept [5]. Improved design methodology that transitions from best effort provisioning to stable platforms that do not compromise secure operation with a multi-core approach have led to increased functionality (Barham, et al. [3]) in recent years. The importance of sustainability is touted in the news media as being critical to the future well being of mankind. The result of this premise has been a nationwide (and even global) push for more efficient use of energy resources. Therefore, virtualization is attractive to this cause since it can reduce the number of computers needed to do the same job. Hence, virtualization promotes green computing. Virtualization has other advantages as well. For example, complexity has been problematic in data centers which are rapidly running out of physical space as well as electrical power. Therefore, there is need to embrace hardware architectures that are less complex which also require fewer personnel resources as well [17].

However, this reduction in the number of physical computers would not be as effective if it were not for the multi-cored capabilities of the current family of computers. The explosive growth of the internet and internet based commerce has resulted in a similar growth in browser based clients. This in turn has forced many data centers to expand to support client/server distributed systems. Because these systems were used to support E-commerce applications world-wide, the complexity of the systems grew and performance became even a larger issue. Therefore, having multiple cores in each physical host helped sustain the necessary performance. It is clear from prior research relating to the application of virtualization to a data center‘s infrastructure that it is certainly possible to reduce the number of physical computers required to perform the same functions. However, a well functioning implementation takes careful planning [21]. The advantages of virtualization can be taken one step further because when proper design guidelines are invoked this new architecture can significantly reduce power consumption, generate less heat and takes up less physical space than the traditional physical model of Armitage and Harrop [1].
The reduction of physical resources translates to a simplicity of design because fewer computers are required in the data center and that results in reduced personnel cost as well. As one would expect an additional advantage of fewer

computers is that less physical maintenance would be required and that is delineated by Lowell, Saito, and Samberg [18]. This simplicity also results in staff savings. While the cost savings in the areas of cooling, physical space, electrical and hardware itself are enticing their percentage of the budget is minimal compared personnel and therefore staffing costs often have the greatest impact to a company's bottom line [4].

In the literature the efficiency and cost saving of virtualization are well documented (Groen [10]; Green Electronics Council, 2006; Vrbsky [24]). However, the purpose of this paper is to build on this architecture, particularly the multi-core aspects to provide speed-up to computationally intense computing problems. To do this software particularly on the operating system level needs to be designed to help optimize the breaking down of a single problem into subparts or threads. The need to invoke such optimization is addressed by Joy [15], who believes that software optimization can be a key component. Specifically, Joy suggests that the management communication strategy needs improvement and calls for more efficient protocols to make hardware more efficient and recommends a shift from the concept of protocols to agents thus minimizing protocol compatibility issues. Since the work of Joy the software has continued to improve and the work of Arzuaga and Kaeli [2] is typical of the improvement realized in the last 10 years. While they report the effectiveness of their methodology within their testbed it is still important to test the efficiency of such software under a variety of conditions. The need for performance testing and scalability is also discussed in by Cencerrado, Senar and Cortes [8], who provide numerous examples to illustrate the point. That localized testing is a prime purpose of this study. In other words the goal is to ascertain to what degree a multi-core approach can improve performance within the current version of the virtualized Linux hosts being used in the author's data center.

Of course for virtualized computing to remain viable backup/replication is critical and addressed in works such as [13]. Also, virtualization creates a whole new set of security concerns which are also addressed in works such as Laureano, Maziero, and Jamhour [16]. However, the main thrust of this paper is to address the effectiveness of dynamic allocation of unused CPU cycles to solve computationally complex problems which might be viewed as a form of load balancing. There is research that supports the validity of load balancing techniques in improving computing performance [19]. Further, the fact that load balancing can improve performance is important given that the virtualized model in its basic form generally reduces an application's access to available computer resources unless a multi-cored approach is used and a dynamic algorithm is used to allocated unused computing resources within an effective priority schema. The case study that follows is designed illustrate how a virtualized multi-core device could be used to improve performance of computationally intense problems.

CASE STUDY

Computing Environment

The autonomous system (AS) used in this case study although used to support research and instruction is designed to mimic the functionality of an ISP/ASP (internet service/application service provider). To mimic that ISP/ASP world the AS utilizes a distributed architecture and robust internet connectivity. In an AS there are numerous physical production hosts that serve a variety of needs. If the traditional physical model were followed each service or application would be housed in separate physical computers to provide isolation for security purposes. Once again from a load balancing perspective the guiding principle in the design is that in all cases the performance of the separate physical host model was acceptable and that unused CPU cycles were available for all hosts if a multi-cored architecture was used. By following this concept one could expect that the core production hosts could be effectively converted to a virtualized host design. This resulting design would ensure that the virtual partitions would provide the required isolation needed for security purposes while the performance would still be tenable and a significant pool of CPU cycles made available. Therefore, it is assumed that because all the original physical hosts were using only a fraction of their computing resources that they would function effectively in a shared virtual host [17]. In this case study, 9 physical hosts were restructured into virtual zones in a single physical host which used a multi-cored architecture. In a static world that would mean that each host would each be limited to about 1/9 of the available computing resources. However, the optimization characteristics of the LINEX operating system allows allocation of resources dynamically which means that the total available resources can be viewed as a pool and any unused resources could be reallocated to any virtual partition. This applies to cores as well. This dynamic allocation

methodology would work well on even a single cored host as long as there are several fairly idle virtual machines. In cases where intense workload is distributed across all 9 virtual partitions on a single cored host performance could fall rapidly and therefore, a priority scheme would be needed to control resource allocation. However, if a multi-cored approach is utilized then each partition may have its own dedicated core and be able to dynamically receive cores as they become available resulting in much more stable performance. The effectiveness of that scenario will be described later in this paper. The purpose of each of the 9 original physical hosts providing IT infrastructure in the author's computing domain is described below in Table 1. Each partition corresponds to what originally was a separate physical host.

Table 1. Partitions (zones) In a Typical Multi-Cored Virtual Host

Partition number	Description
Host	Hypervisor: Time Server, Virtual Machine host(Running LINUX), host level management
1	Primary Client Access Server. SSH login server(Firewall will forward all port22 traffic to this partition)
2	Secondary Client Access Server, SSH login server as backup if partition 1 locks up (Firewall will forward alternate port to this partition, port22)
3	Global Authentication Server (OpenLDAP/Kerberos5-MIT) , allows single sign-in within the AS
4	Network Address Resolution Server (DNS/DHCP/LDAP) , required for IPv4
5	E-Mail Server (LINUX Mail Server installed), AS email
6	Web Server (Apache Tomcat) , supports web development
7	Global File System Server for Home folders (NFS Mounted), allows a user's home directory to follow them across hosts
8	Programming Application Development Server (java, C, C++ and etc.), for program development
9	Data Base Application Server (mySQL, SQL queries, DB tuning and etc.) , for database development and centralized security logging

Row one in the table describes the characteristics of the physical host for the 9 partitions (zones) that will follow. Specifically, this host contains the network time server (NTP) that is used to synchronize all hosts (both virtual and physical) contained within the domain. This process is crucial in client/server and database applications and all hosts contained within the domain need to be synchronized for data integrity and security purposes. This physical host and all subsequent partitions are configured with the Linux operating system although it is possible for any given partition to run other operating systems including windows.

The first partition contains the main client access server that is used to allow people within the organization and collaborators from all over the world access to the domain's resources in a secure manner. Because it is possible that the domain's clients could need access from anywhere in the world this domain is connected to the internet in a robust fashion (two internet connections at 20+ Mbs each). Hence, it is crucial to monitor client traffic for security purposes. In adhering to this end, both incoming and outgoing traffic arriving at this partition needs to be tightly filtered at the firewall level and terminal access is only available through secure shell (ssh) which uses a sophisticated encryption stream.

The second partition provides a secondary version of virtual zone 1 (described above). The primary purpose of this zone is to provide a backup to partition 1 so that clients can still gain access to the domain if partition 1 becomes corrupted. Because this domain uses a global authentication system (active directory) single sign-on can be provided to users so that they are granted access to both partitions one and two while still maintaining the same login information for both partitions.

The purpose of partition 3 then is to host the global authentication software. Specifically, global authentication is provided by lightweight directory access protocol which is well known for its robustness and scalability [11]. To

further protect this critical component in the IT security structure Kerberos (developed by MIT) is used to increase the robustness of the encryption. Further, this methodology uses a mechanism so that the password does not have to be sent remotely across the network.

Partition number four provides network address resolution services. In internet connected autonomous systems it is necessary to maintain an address index to external addresses. Domain name service (DNS) provides this functionality and along with the dynamic host configuration protocol (DHCP) is hosted in this zone. These services allow address resolution of network layer addressing to take place and client work stations to be allocated a temporary network (IP) address. Domain service needs to be provided to all domains within the company's enterprise because a given company might support more than one domain name (such as cady.com and cadilac.com).

Email services are housed in virtual zone 5 via SMTP (simple mail transfer protocol). Specifically, this virtual partition hosts a mail system which provides standard mail services and a browser based email interface that is linked to a standard apache tomcat web server (which is hosted in partition 6). To assure backward compatibility, both secure and unsecured mail services are supported. Table 1 also illustrates that standard web services are supported in partition 6 and as previously mentioned apache tomcat is the web server software of choice due to its wide-spread use. This web server also hosts domain related home pages, web interfaces for end-user, java applet development and E-commerce instructional/research activities.

Partition 7 hosts the global file system which uses the network file system (NFS) protocol. This file system can be described as the central depository of data within the domain and allows all end-user data to be stored in a central location for ease of access and management. An additional benefit of this global orientation allows a user's home directory (or directory maps) to follow that user no matter which host (virtual or physical) he/she is logged into. In other words, the user always has the same default directory no matter which host (virtual or otherwise) he/she is connected to. While the concept of centralizing all data at a single location provides management advantages however it involves risk. Using an old adage it is analogous to putting all your eggs (data) are in one basket (location). Therefore, it is crucial to distribute the data to multiple locations to achieve the multiple basket logic. Fortunately, virtualization is conducive to this goal because the functionality of the entire domain can be housed in a single physical computer. Therefore, by positioning several of those computers all configured with the same file synchronization logic at various locations (hopefully geographically separated) excellent fault tolerance and backup can be achieved.

Virtual partition 8 is designed to programming/ developmental related activities. In other words, applications using standard programming languages such as java can be developed. Direct access to this zone would not be allowed for sensitive applications so a user would have to authenticate through partition 1 (or 2) and obtain access the application via SSH or some form of tunneled web browser.

The last partition contains the virtual zone that contains the database application server. This partition would support applications such as database design, SQL code development and database tuning. Once again because security is critical direct access to this zone would not be allowed for sensitive data so a user would have to authenticate through partition 1 (or 2) and access the application via SSH or some form of tunneled web browser.

Dynamic Allocation of Cores in a Virtual Host

The first step in leveraging a multi-core strategy is to ensure that symmetric multi-processing (SMP) is in fact configured:

```
buster@mermes:~$ uname –a

Linux mermes 2.6.28-19-server #66-Ubuntu SMP Sat Oct 16 18:41:24 UTC 2010 i686
```

As illustrated above SMP appears in the basic configuration information about a virtualized host called mermes. It is next logical to ascertain if there are in fact multiple cores available and the mpstat (multi-processor statistics) command is used below to determine the number of cores available.

```
buster@mermes2:/$ mpstat -A
Linux 2.6.32-30-generic (mermes2)     05/07/2011      _x86_64_        (8 CPU)
10:39:24 AM  CPU    %usr   %nice    %sys %iowait    %irq   %soft  %steal  %guest   %idle
10:39:24 AM  all    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    0    0.00    0.00    0.01    0.02    0.00    0.00    0.00    0.00   99.97
10:39:24 AM    1    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    2    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    3    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    4    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    5    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    6    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
10:39:24 AM    7    0.00    0.00    0.01    0.00    0.00    0.00    0.00    0.00   99.99
```

The output indicates that there are eight cores and all of them are almost completely idle and could have unused CPU cycles to allocate elsewhere.

To get some idea of how the SMP component of the operating system might split a job to take advantage of multiple core we can use the LINEX ps –ALF command. The output below from virtual zone number 8 from Table 1 which is configured as a LINEX host illustrates this condition.

Eight Processors SMP enabled

```
                                NLWP            PSR#
dguster  16172 14738 16172  0   18 578273 41924  4 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16173 99   18 578273 41924  0 11:06 pts/0    00:00:02 java PrimeByVector
dguster  16172 14738 16174  0   18 578273 41928  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16175  0   18 578273 41928  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16176  0   18 578273 41928  0 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16177  0   18 578273 41928  0 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16178  0   18 578273 41932  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16179  0   18 578273 41932  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16180  0   18 578273 41932  0 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16181  0   18 578273 41932  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16182  0   18 578273 41932  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16183  0   18 578273 41936  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16184  0   18 578273 41936  2 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16185  0   18 578273 41936  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16186  8   18 578273 41936  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16187  6   18 578273 41940  2 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16188  0   18 578273 41940  2 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16172 14738 16189  0   18 578273 41940  1 11:06 pts/0    00:00:00 java PrimeByVector
dguster  16190 15808 16190  0    1  1760  1048   4 11:06 pts/1    00:00:00 ps –ALF
```

In the output above, a java program "PrimeByVector" (configured to find all prime numbers between 2 and 10 million) is allocated across 4 of the 8 available processors and is split into 18 subparts by the SMP logic. In other words, the SMP logic breaks the java program into 18 instances or light weight processes (NLWP=18). Further, an examination of the processors each LWP is assigned to reveals that four processors have been allocated to this job. These processors are processor number 0, 1, 2 and 4 (illustrated in the PSR# column).

In terms of performance gains related to a multi-cored approach, as one would expect that including additional processors can have an impact. To illustrate this principle the java program from above was run on static single core and multiple core configurations and the results appear below.

Single Static Processor SMP enabled

```
bcrl@t-ubuntu:~$ (2 threads across only one processor)
real    2m56.674s
user    2m53.820s
sys     0m2.520s
```

Multiple Processors Allocated Dynamically SMP enabled

```
dguster@mermes2:~/javaclass$ (18 threads across 7(out of 8 available) processors)
real    0m25.610s
user    0m23.520s
sys     0m2.830s
```

As expected the multi-processor example finished quicker in regard to elapsed time (real values 25.61 seconds versus 2 minutes, 56.674 seconds). Further, the second example put less of a load on the processor (sum of user and sys time) to which the root process of the java program was assigned ((23.520 + 2.830) vs. (2 53.820 + 2.520)).

The impact on performance can also be illustrated by looking at the idle percentage of the main processor assigned to the virtual zone. Using the LINEX vmstat (virtual memory statistics) command can provide such data and a representative example follows.

Single Static Processor SMP enabled

```
dguster@mermes2:~$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 1  0      0  78572 126948 117212    0    0     0     9   18   23  0  0 99  0
 2  0      0  78352 126948 117212    0    0     0     4 1007  939  8 28 64  0
 1  0      0  78352 126948 117212    0    0     0     0 1153 1040 10 32 58  0
 1  0      0  78352 126948 117212    0    0     0     0 1155 1063 10 25 65  0
 1  0      0  78336 126948 117212    0    0     0     0 1294 1047 10 31 59  0
 3  0      0  78188 126948 117212    0    0     0     0 1301 1132 11 32 57  0
```

Multiple Processors Allocated Dynamically SMP enabled

```
dguster@mermes2:/$ vmstat 1
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id   wa
 1  0      0 7598804 108696 204440    0    0     0     0    1    1  0  0 100 0
 1  0      0 7592152 108696 204440    0    0     0     0 1322 4766  8  3 89  0
 1  0      0 7584032 108696 204448    0    0     0     0 1215 3314  4  1 95  0
 0  0      0 7584780 108696 204448    0    0     0    36  676 1952  6  0 93  0
 0  0      0 7584652 108696 204448    0    0     0     0  242  253  0  0 100 0
```

An examination of the idle percentage of the main processor in each case (under main heading CPU, subheading id) consistently reveals smaller idle percentages for the static example, which means the processor is busier. Further, the number of processes waiting to be assigned to a CPU (under main heading procs, subheading r) is much greater in the first example. In other words the dynamic SMP configuration never has more than 1 process waiting whereas the static example has 2 or 3 waiting at various points in the data.

CONCLUSIONS

It is clear that a multi-cored architecture invoked as a result of virtualization can provide a distinct performance advantage while still maintaining adequate performance for each zone. The performance gain realized could easily translate to more cost-effective operations. These advantages can be easily exploited by a company that is connected to the internet and wishes to portray an international presence. Response time to the end user is critical in applications such as E-commerce. With the multi-cored approach the wait time for a CPU can be greatly reduced and the base applications can be speed up as well. While it is clear that computationally intensive applications would benefit, IO intensive applications could benefit as well. For example, Guster, D. C., Safonov, P. I., Brown, C., & Jansen, B. [14] found that the multi-cores could reduce the time to find and extract records from a database.

As stated earlier the rationale to justify the multi-cored approach can be embedded in virtualization logic which features numerous advantages of its own. Among these advantages is added fault tolerance which is critical in online business models that use internet connectivity such as E-commerce. The importance of fault tolerance in E-commerce cannot be underestimated. For example, if an E-commerce site is not available then often not only the current sale would be lost but all future sales from that customer. The use of host level replication makes it easy to use virtualization to provide fault tolerance. Further, host level replication besides providing fault tolerance can also

be used to provide another level of dynamic core allocation. In other words, instead of just dynamically allocating core across zones in a virtualized host one could allocated core across hosts which is one of the primary tenants of cloud computing.

In addition, that simplification typically provides ease of management and a reduction in personnel time which is often significant because personnel costs are often a major portion of the IT budget. Virtualization can often result in significant savings in electrical power consumption as well. For example, Guster, Hemminger and Krzenski [12] found that both electrical power consumption and cooling cost could be reduced by a factor of 5.

Therefore, future designs of IT infrastructures designated to support companies with a global presence should consider virtualization, although careful consideration of the physical configuration of the hosting computer is needed. The fact that virtualization reduces complexity, improves performance and is cost effective makes it easy to justify the research and development needed to bring an IT staff up to speed on its deployment. The leveraging the multi-cored technology further adds to its attractiveness.

REFERENCES

1. Armitage, G. and Harrop, W. (2005). Teaching IP Networking Fundamentals in Resource Constrained Educational Environments, *Australasian Journal of Educational Technology*, 21(2), 263-283.
2. Arzuaga, E. and Kaeli, D. (2010). Quantifying load imbalance on virtualized enterprise servers. Proceedings of the First WOSP/SIPEW International Conference on Performance Engineering, ACM Press.
3. Barham, P. et al. (2003). Xen and the Art of Virtualization, *Proceedings of the 19th ACM Symposium on Operating Systems Principles,* 164-177.
4. Bergeron, Francois; Rivard, Suzanne; and Serre, Lyne. 2008. "Investigating the Support Role of the Information Center," *MIS Quarterly*, (14: 3).
5. Borden, T., Hennessy, J. and Rymarczyk, J. (1989). Multiple Operating systems on One Processor Complex, *IBM systems Journal*, 28(1), 104-123.
6. Boyd, T. and Dasgupta, P. (2002). Process migration: A Generalized Approach Using a Virtual Operating System, *Proceedings of the 22nd International Conference on Distributed Computing Systems*, 385.
7. Davenport, T. and Prusak, L. (2000). Working Knowledge: How Organization Manage What They Know, *Ubiquity*, 1(24), Article 2.
8. Cencerrado, A., Senar, A. and Cortes, A. (2009). Support for Urgent Computing Based on Resource Virtualization. In Lecture Notes in Computer Science, edited by G. Allen, (5544), pp. 227-236.
9. Creeger, M. (2005). Multicore CPUs for the Masses.Queue Magizine, 3(7).
10. Groen, P. (2008). "Green" Information Technology (IT) Strategies and Practices, Working paper Shepherd University, WV.
11. Guster, D. C., Hall, C., Herath, S., Jansen, B., & Mikluch, L. (2008, April 24). Analysis of vulnerabilities in a global authentication system in a university based distributed processing laboratory. Presentation at the 3rd International Conference on Information Warfare and Security, Omaha, NE.
12. Guster, D., Hemminger, C., and Krzenski, S. (2009). Using Virtualization to Reduce Data Center Infrastructure and Promote Green Computing, A paper presented to the IABE Conference.
13. Guster, D. C., McCann, B., Krzenski, K., & Lee, O. F. (2008). A cost effective, safe, and simple method to provide a disaster recovery plan to small and medium sized businesses. *Review of Business Research*. 8(4), 63-71.
14. Guster, D. C., Safonov, P. I., Brown, C., & Jansen, B. (2008). An analysis of distributed database indexing method in regard to performance of extract/transform/load (ETL) processes. *Issues in Information Systems, IX*(2), 544-550.
15. Joy, B. (2000). Shift from Protocols to Agents, *IEEE Internet Computing*, 4(1), 63-64
16. Laureano, M., Maziero, C. and Jamhour, E. (2007). Protecting Host-based Intrusion Detectors Through Virtual Machines, *Computer Networks: The International Journal of Computer and Telecommunications Networking,* 51(5), 1275-1283.
17. Lee, Olivia F.; Dennis C. Guster; Mark B. Schmidt; and Brandon McCann. (2009). ―Applying the Scale-Free Degree Distribution Algorithm to Assess Communication Complexity and Failure Points in Disaster Recovery Models. *Journal of Information Technology Management*, XX(2).

18. Lowell, D., Saito, Y. and Samberg, E. (2004). Devirtualizable Virtual Machines Enabling General, Single Node, Online Maintenance, *Proceedings of the 11th International conference on Architectural Support for Programming Languages and Operating Systems*, 211-223.
19. Lin, Q., et al, (2007). Grid-based Large-scale Web3D Collaborative Virtual Environment, *Proceeding of the 12th International Conference on 3D Web Technology*, 123-132.
20. Safonov, P., Guster, D., and Hemminger, C. (2011). Employing Host Virtualization as a Strategy for Sustainability, Security and Reliability in a Corporate Data Center: Balancing Utility with Performance, A paper presented to the IABPAD Conference, Orlando, January 2011.
21. Schultz, B. (2008). How to Create a Business-boosting Virtualization Plan, *Network World*, 8, 1-5
22. Singh, A., Korupolu, M. and Mohapatra, D. (2008). Server-storage virtualization: integration and load balancing in data centers. Proceedings of the ACM/IEEE Conference on Supercomputing, IEEE Press.
23. Smith, J. and Nair, R. (2005). The Architecture of Virtual Machines, *Computer*, 38(5), 32-38.
24. Vrbsky, S. (2007). Introduction to Green Computing, University of Alabamba Lecture Notes, #475.
25. Xu J., Zhao, M., Fortes, J., Carpenter, R. and Yousif, M. (2008). Autonomic Resource Management in Virtualized Data Centers Using Fuzzy Logic-based Approaches, *Cluster Computing*, 11(3), 213