IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Volume-3, Issue-11 E-ISSN: 2347-2693

Towards Live Migration Using Rule Trigger Secure Policies in Virtual Machines

D.Ragupathi[1], S.Santhanaarokiajohnsy[2*]

[1]*Asst. Professor, Department of Computer Science, A.V.V.M Sri Pushpam College, Thanjavur.*
[1]*M.Phil Research Scholar, Department of Computer Science, A.V.V.M Sri Pushpam College, Thanjavur.*

www.ijcseonline.org

Received: Oct/23/2015 Revised: Nov/06/2015 Accepted: Nov/18/2015 Published: Nov/27/2015

***Abstract*—** Virtualization innovation has become commonplace in advanced information focuses and group systems, frequently referred as "Registering Clouds". In particular, the capacity of virtual machine (VM) development brings various advantages such as higher performance, improved reasonability and flaw tolerance. Moreover, live development of VMs frequently permits workload development with a short administration downtime. However, administration levels of running applications are likely to be adversely influenced amid a live VM migration. Alternately this reason, a better understanding of its impacts on framework execution is exceedingly desirable. In this paper, we present a execution assessment on the impacts of live development of virtual machines on the execution of applications running inside Xen VMs. Results appear that in most cases, development overhead is worthy however can't be disregarded, particularly in frameworks where administration accessibility and responsiveness are administered by strict Administration Level Understandings (SLAs). Despite that, there is a high potential alternately live development materialness in information focuses serving enterprise-class Web applications. Our results are based on a workload formed of a genuine application, covering the space of multi-level Web 2.0 applications.

***Keywords*—** Live Migration, Registering Clouds, Xen, Virtual Machine

I. Introduction

Virtual machine (VM) innovation has recently emerged as an essential building block alternately information focuses and group systems, mainly due to its capabilities of isolating, solidifying and moving workload. Altogether, these highlights permit an information focus to serve various clients in a secure, adaptable and productive way. Consequently, these virtualized infrastructures are consider a key part to drive the emerging Cloud Registering worldview.

Development of virtual machines seeks to improve manageability, execution and flaw resilience of systems. More specifically, the reasons that justify VM development in a production framework include: the need to equalization framework load, which can be accomplished by moving VMs out of overloaded/overheated servers; and the need of selectively bringing servers down alternately upkeep after moving their workload to other servers.

The ability to relocate an entire operating framework overcomes most difficulties that traditionally have made process-level development a complex operation. The applications themselves and their corresponding forms do not need to be aware that a development is occurring. Hypervisors, such as Xen and VMWare, permit moving an OS as it continues to run. Such system is termed as "live" alternately "hot" migration, as opposed to "immaculate stop-and-copy" alternately "cold" migration, which involves halting the VM, duplicating all its memory pages to the destination host and then restarting the new VM. The principle advantage of live development is the possibility to relocate an OS with near-zero downtime, an important feature when live administrations are being served.

1.1 Background

On Xen, as described by Clark et al., live moving a VM basically comprises of transferring its memory image from a source server to a destination server. To live relocate a VM, the hypervisor alternately pre-copies memory pages of the VM to the destination without interrupting the OS alternately any of its applications. The page duplicating process is repeated in various rounds on which dirty pages are continuously transferred. Normally, there is a set of pages that is modified so frequently that the VM must be stopped alternately a period of time, until this set is completely exchanged to the destination. Subsequently, the VM can be resumed in the new server.

It has been watched that live development of VMs permits workload development with close zero application downtime. Nevertheless, the execution of a running application is likely to be adversely influenced amid the development process due to the overhead caused by successive iterations of memory pre-duplicating. Alternately the duration of the pre-duplicating process extra CPU cycles are devoured on both source and destination servers. An extra amount of system bandwidth is devoured as well, conceivably affecting the responsiveness of Web applications. In addition, as the VM resumes after migration, a slowdown is anticipated due to store warm-up at the destination.

Moreover, downtime and application execution are likely to be influenced in distinctive ways alternately distinctive applications due to varying memory usages and access patterns. Past studies have found that genuine downtime might vary considerably between applications, ranging from as low as 60 ms when moving a Quake game server to up to 3 seconds in case of specific HPC benchmarks. Regarding the overhead due to development activity, earlier studies have shown that experienced slowdown ranged between 1% and 8% of wall-clock time alternately a specific set of HPC benchmarks.

In other circumstances utilizing Xen, a 12% to 20% slowdown on the transmission rate of an Apache Web server running a VM with 800MB of memory and serving static content was reported. In the case of a complex Web workload (SPECWeb99) the framework under test could maintain the conformity to the benchmark measurements. In all cases, it has been concluded that, alternately the specific set of applications considered, the bad impacts of development were worthy alternately negligible in contrast to its potential advantages to framework flaw resilience.

1.2 Our contribution

We have identified that a case study taking into consideration live development impacts in the execution of advanced Web applications, such as multi-level Web 2.0 applications, is lacking in the current literature. However, such a study would aid researchers and practitioners currently evaluating the deployment of this class of application in clouds. Our commitment is a case study that measures the sway of VM live movements in the execution of one example, yet representative, of an advanced Web application. Our study will be conceivably valuable to circumstances where metrics, such as administration accessibility and responsiveness, are driven by Administration Level Understandings (SLAs). In such frameworks administration suppliers and consumers agree upon a least administration level and resistance to such agreement might incur in penalties to suppliers. More importantly, an SLA directly reflects how end-clients perceive the quality of administration being delivered.

The rest of this paper is organized as follows: Segment 2 positions our study among related work; Segment 3 portrays why advanced Web applications are distinctive than customary workloads; Segment 4 portrays our objectives, trial set up, workload and metrics; in Segment 5 we present and analyze the results of our execution evaluation; finally, we conclude the paper in Segment 6.

II. RELATED WORK

The advent of innovative technologies, such as multicore, par virtualization, hardware-assisted virtualization and live development, have contributed to an expanding adoption of virtualization on server systems. At the same time, being capable to measure the pros and cons of adopting virtualization in face of such advancements is a challenging task. The sway of virtualization in a assortment of circumstances has been the focus of considerable attention. A number of studies have exhibited individual and side by side measurements of VM runtime overhead imposed by hypervisors on a assortment of workloads.

Apparao et al. present a study on the sway of solidifying several applications on a single server running Xen. As workload the authors utilized the consolidate benchmark characterized by Intel, which comprises of a Web server VM, a database server VM, a Java server VM and mail server VM. An idle VM is moreover added to comply with genuine world scenarios, on which servers are hardly completely utilized.

The studies exhibited by Zhao & Figueiredo and Clark et al. particularly deal with VM migration. The former analyzes execution degradation when moving CPU and memory intensive workloads as well as moving various VMs at the same time; however such study employs a immaculate stop-and-duplicate development approach maybe than live migration. The later introduces Xen live development and measures its impacts on a set of four applications common to facilitating environments, primarily utilizing on measuring downtime and total development time and demonstrating the viability of live migration. However, these works have not assessed the sway of development in the execution of advanced Web workloads, such as multi-level and social system oriented applications.

A few studies propose and assess the efficacy of moving VMs across long distances, such as over the Internet. alternately instance, Travostino et al. have demonstrated the effectiveness of VM live development over an WAN associated by devoted 1Gbps links; application downtime has been quantified at 5-10 times greater than that experienced on an intra-LAN set-up, despite a 1000 times higher RTT. Besides its feasibility, the concept of WAN live development is still to be implemented in business hypervisors, which demands all involved machines to be in the same subnet and offer storage. Our work focuses just on moving VMs within a information focus alternately cluster.

The Cloud stone benchmark points at registering the monetary cost, in dollars/user/month, alternately facilitating Web 2.0 applications in cloud registering stages such as Amazon EC2. From this work we borrow the idea of utilizing Olio and Faban to compose our target workload alternately Web 2.0 applications. However, Cloud stone does not define a system to assess the fetched of virtual machine development and, to the best of our knowledge, no past work has considered utilizing this sort of workload in development experiments.

III. Characteristics of Advanced Web Applications

The space of applications that can conceivably take advantage of the Foundation as a Administration worldview is broad. Alternately instance, Amazon reports several case studies that leverage their EC2 platform, counting video processing, genetic simulation and Web applications. In particular, such stages are particularly valuable alternately multi-level Web applications, generally counting a Web server (e.g. Apache), an application server/dynamic content generation (e.g. PHP, Java EE), and a backend database (e.g. MySQL, Oracle). Virtual machine innovation adds extra flexibility to scaling of Web applications, by permitting dynamic provisioning and replication VMs to host extra instances alternately one the application tiers.

Social organizing web locales are perhaps the most capable illustration of exceedingly dynamic and interactive Web 2.0 applications which gained prevalence over the past few years. Their expanding prevalence has spurred demand alternately a exceedingly scalable and adaptable solution alternately facilitating applications. Many bigger locales are growing at 100% a year, and littler locales are expanding at an indeed more rapid pace, doubling every few months.

These web applications present extra highlights that make them distinctive from customary static workloads. Alternately instance, their social organizing highlights make each users' activities affect many other users, which makes static load partitioning unsuitable as a scaling strategy. In addition, by means of blogs, photo streams and tagging, clients now publish content to one another maybe than just consuming static content.

Altogether, these highlights present a new sort of workload with specific server/client communication patterns, write designs and server load. However, most accessible execution studies use extremely simple static file retrieval tests to assess Web servers, frequently leading to erroneous conclusions. Alternately this reason, in this work we have this trend into account amid the workload determination process, resulting in the determination of Olio as a reasonable and delegate workload.

IV. Assessment of Live Development Cost

This study points at achieving a better understanding of live development impacts on advanced Web applications. Alternately this reason, we have outlined benchmarking tests to assess the sway of live development on a reasonable Web 2.0 application facilitated on organized virtual machines.

4.1 Tested specifications

Our tested is a group formed of 6 servers (1 head-hub and 5 VM hosts). Each hub is equipped with Intel Xeon E5410 (a 2.33 GHz Quad-core process alternately with 2x6MB L2 store and Intel VT technology), 4 GB of memory and a 7200 rpm hard drive. The servers are associated through a Gigabit Ethernet switch.

The group head-hub runs Ubuntu Server 7.10 with no hypervisor. All other hubs (VM hosts) run Citrix XenServer Venture Version 5.0.0. Our choice alternately a business hypervisor alternately is based on the assurance of an venture class software in accordance with the needs of target users, i.e. venture information focuses and public application facilitating environments.

All VMs run 64-bit Ubuntu Linux 8.04 Server Edition, para virtualized kernel variant 2.6.24-23. The installed web server is Apache 2.2.8 running in prefork mode. PHP variant is 5.2.4-2. MySQL, with Innodb engine, is variant 5.1.32.

4.2 Workload

We have picked to use Olio as a Web 2.0 application, combined with the Faban load generate alternately to represent a reasonable application and workload set. Olio is a Web 2.0 toolkit that helps developers assess the suitability, functionality and execution of various Web technologies, devised by Sun Micro frameworks from its understanding of the challenges faced by Web 2.0 customers. It has been success completely deployed and assessed in a reasonably measured high-end server foundation, as well as in rented assets from Amazon EC2.

The Olio Web application represents a social-occasions website which permits clients to perform several activities such as stacking the homepage, logging into the system, creating new events, attending occasions and searching alternately occasions by date alternately tag. It currently provides implementations utilizing three technologies: PHP, Ruby on Rails and J2EE. Alternately our experiments, we have picked to use Olio's PHP implementation, subsequently employing the popular LAMP stack (Linux Apache MySQL PHP).

Faban is an open-source Markov-chain load generate alternately used to drive load against Olio; it is formed by an expert program which spawns one alternately more load drivers, i.e. multi-strung forms that simulate genuine users. The expert presents a Web interface through which it is conceivable to submit customized benchmark runs and monitor alternately their results. This Olio/Faban combination was originally proposed as part of the Cloud stone benchmark.

The load level driven against the application might be varied by changing the number of simultaneous clients to be served by the application. Total time alternately each run is configured by adjusting three distinctive durations, namely ramp-up, unfaltering state and ramp-down. Resulting measurements reported by Faban just take into account the unfaltering state period.

The principle metric considered in our tests is a Administration Level Agreement characterized in Cloudstone. The SLA defines least reaction times alternately all relevant client actions. Thus, at any 5-minute window, if a certain percentile of reaction times exceeds the maximum, an SLA infringement is recorded. The 90th and 99th percentiles are considered in this study, representing a more relaxed and a stricter SLA, respectively. Table 1 lists the details of the SLA considered throughout this paper.

Table 1. Cloudstone's SLA: The 90th/99th percentile of reaction times measured in any 5-minute window amid unfaltering state should not excess the following values (in seconds):

Client action	SLA
Home page loading	1
Client login	1
Occasion tag search	2
Occasion detail	2
Individual detail	2
Incorporate person	3
Incorporate event	4

4.3 Benchmarking architecture

The architecture of our benchmarking setup is depicted in Figure 1. Based on the observation that MySQL tends to be CPU-bound when serving the Olio database, whereas Apache/PHP tends to be memory-bound, we have outlined our framework under test (SUT) by splitting the workload into two organized VMs, facilitated in distinctive group nodes, in request to better partition the accessible physical resources.

All hubs offer an NFS (System File System) mounted capacity device, which resides in the head-hub and stores VM images and virtual disks. In particular, a nearby virtual plate is facilitated in the server that hosts MySQL.

The load is driven from the head-node, where the multi-strung workload drivers run, along with Faban's expert component.

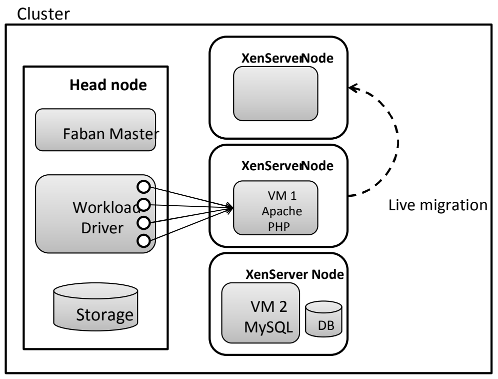

Fig.1. Benchmarking architecture

4.4 Trial design

The overall objective of our tests is to measure slowdown and downtime experienced by the application when VM movements are performed in the center of a run. Specifically, we measure application slowdown based on values generated by the above-mentioned SLA calculation.

In all experiments, group hubs and their interconnection were devoted to the application under test. A development experiment consisted of moving a single VM between two devoted physical machines. In each run, the picked destination machine was distinctive from the source machine in the past run, i.e. a arrangement of runs did not consist of moving a VM back and forth between the same two machines.

Preparatory tests Exact VM sizes were obtained by preparatory experiments, in which we have run the application without performing any VM migration. We have driven load against Olio and gradually increased the number of simultaneous clients between runs, in 100 clients increments, while utilizing 2 identically measured VMs with 2 vCPUs and 2GB of memory. By analyzing the SLA (both 90th and 99th percentile of reaction times alternately all client actions), we have found that 600 is the maximum number of simultaneous clients that can be served by our SUT. We have watched memory and CPU usage to find the least VM sizes capable of serving 600 users. We have then aimed at diminishing the size (vCPUs and memory) of the VMs to the least required to serve 600 users. Thus, in the final configuration the first VM, which exclusively hosts Apache/PHP, has 1 vCPU and 2GB of memory; the second VM, which hosts MySQL, has 2 vCPUs and 1GB of memory.

In the same preparatory tests we have noticed execution issues when facilitating the MySQL server on NFS. The application would not scale to more than 400 simultaneous users, which has lead us to host MySQL in a nearby disk, subsequently scaling up to 600 simultaneous users. Alternately this reason, our tests do not incorporate moving the VM that hosts the database server, since XenServer requires all capacity devices to be facilitated in a system capacity in request to perform live migrations.

Development tests in our first set tests with Olio we have performed 10-minute and 20-minute benchmark runs with 600 simultaneous users. Amid these experiments, live movements of the Web server VM were performed. The goal of experimenting with this load level is to assess how the pre-characterized SLAs are neglected when the framework is about oversubscribed, however not overloaded. Also, we aim at measuring the duration of development impacts and the downtime experienced by the application.

Subsequently, in a second round of experiments, we have run the benchmark with littler numbers of simultaneous users, namely 100, 200, 300, 400 and 500, aiming at finding a "safe" load level on which movements can be performed at lower risks of SLA violation, particularly when considering the more stringent 99th percentile SLA.

V. RESULT AND DISCUSSION

Overall, our trial results appear that overhead due to live development is worthy however can't be disregarded, particularly in SLA-situated circumstances requiring more demanding administration levels. Figure 2 shows the sway of a single development performed after five minutes in unfaltering state of one run. A downtime of 3 seconds is experienced close the end of a 44 second migration. The highest peak watched in reaction times takes place immediately after the VM resumes in the destination node; 5 seconds elapse until the framework can completely serve all demands that had initiated amid downtime. In spite of that, no demands were dropped alternately timed out due to application downtime. The downtime experienced by Olio when serving 600 simultaneous clients is well above the anticipated millisecond level, beforehand reported in the writing alternately a range of workloads. This interesting result suggest that the workload complexity imposes a unusual memory access pattern, expanding the difficulty of live development the virtual machine.

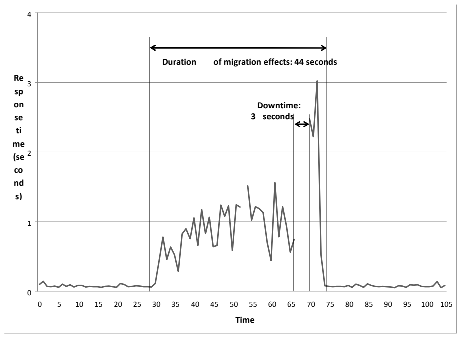

Fig.2. Impacts of a live development on Olio's homepage stacking activity

Figure 3 presents the sway of various movements on the homepage stacking reaction times. These result compares to the average of 5 runs. We report the 90th and 99th percentile SLAs. We can observe that the more stringent 99th percentile SLA is neglected a short moment after the first development is performed indicating that when 600 simultaneous clients are being served, a single VM development is not acceptable. The 90th percentile SLA is not neglected when a single development occurs, however is neglected just when two movements are performed in a short period of time.

Figure 3 moreover indicates that more than one development might not cause infringement of the 90th percentile SLA. A way of preventing such infringement is permitting sufficiently spacing between movements in request to permit the SLA recipe to generate normal reaction time levels. Thus, it is paramount that this information is utilized by SLA-situated VM-allocation mechanisms with the objective of diminishing the hazard of SLA resistance in circumstances when VM movements are inevitable.

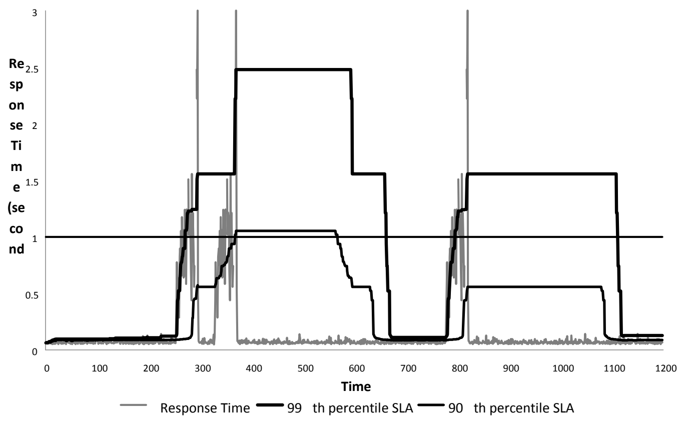

Fig.3. 90th and 99th percentile SLA processed alternately the homepage stacking reaction time with 600 simultaneous users. The maximum permitted reaction time is 1 second

From the above mentioned results we can conclude that, in spite of a huge slowdown and downtime caused by live development of VMs, our SUT is resilient to a single development when the framework responsiveness is administered by the 90th percentile SLA. In other words, provided that movements are performed at correct times, there is no fetched associated with them. However, this is not the case alternately the 99th percentile SLA. Alternately this reason, we have performed a new arrangement of tests with littler number of simultaneous users. The objective of such tests is to gauge a safe level on which a development could be performed with low hazard of SLA violation.

Table 2. Maximum recorded 99th percentile SLA alternately all client activities when one development is performed alternately 500, 400, 300, 200 and 100 simultaneous users

Action	500	400	300	200	100
HomePage	0.32	0.18	0.25	0.25	0.13
Login	0.32	0.33	0.42	0.28	0.14
TagSearch	0.46	0.32	0.35	0.39	0.29
EventDetail	0.48	0.27	0.22	0.24	0.14
PersonDetail	1.53	0.62	0.69	0.61	0.32
AddPerson	2.28	1.00	1.51	1.73	0.66
AddEvent	2.26	1.02	1.30	1.81	0.98

Table 2 presents more detailed results listing maximum reaction times alternately all client activities as processed by the 99th percentile SLA recipe when one development was performed in the center of a 10 minute run. In these runs the load varies from 100 to 500 users. In all cases, our SUT was capable to sustain an worthy execution indeed in the presence of a live development of the Web server. Alternately instance, the maximum esteem watched alternately homepage stacking is 0.32 seconds, which compares to approximately 1/3 of the maximum esteem allowed, i.e. 1 second. The maximum esteem watched alternately the adding a new individual to the framework (2.28 seconds), which is more than half of the maximum allowed, however still does not demonstrate hazard of SLA violation. These results demonstrate that a workload of 500 clients is the load level at which a live development of the Web server should be conveyed out (e.g. to a least loaded server) in request to decrease the hazard of SLA violation.

VI. Conclusion

Live development of virtual machines is a valuable capacity of virtualized groups and information centers. It permits more adaptable administration of accessible physical assets by making it conceivable to load equalization and do foundation upkeep without entirely compromising application accessibility and responsiveness.

We have performed a arrangement of tests to assess the fetched of live development of virtual machines in a scenario where a advanced Web application is facilitated on a set of virtual machines. Live development tests were conveyed out in circumstances where several levels of load were driven against the application.

Our results appear that, in an occurrence of a about oversubscribed framework (serving 600 simultaneous users), live development causes a huge downtime (up to 3 seconds), a bigger esteem than anticipated (based on results beforehand reported in the writing alternately simpler, non Web 2.0 workloads) Also, this administration disruption causes a pre-characterized SLA to be neglected in some situations, particularly when two movements are performed in a short period of time.

On the other hand, we have found the most stringent SLA (99th percentile) can still be met when movements are performed when the framework load is slightly decreased to less simultaneous clients (500 in our case study).

In conclusion, we see a high potential of live development materialness in information focuses serving advanced Web services. This execution assessment study is the first step towards the broader objective of studying the power of live development of virtual machines alternately the administration of information focuses and clusters. We plan to use the insights of this study to develop smarter and more productive SLA-based resource allocation systems.

References:

[1] Chanchio, K.; Dept. of Comput. Sci., Thammasat Univ., Patumtani, Thailand; Thaenkaew, P." Time-Bound, Thread-Based Live Migration of Virtual

Machines", Published in: Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on Date of Conference: 26-29 May 2014 Page(s):364 – 373.

[2] Anala, M.R.; Dept. of Comput. Sci. & Eng., RVCE, Bangalore, India; Shetty, J.; Shobha, G." A framework for secure live migration of virtual machines", Published in: Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on Date of Conference: 22-25 Aug. 2013 Page(s): 243 – 248.

[3] Shah, S.A.R.; Korea Univ. of Sci. & Technol., Daejeon, South Korea; Jaikar, A.H.; Seo-Young Noh," A performance analysis of precopy, postcopy and hybrid live VM migration algorithms in scientific cloud computing environment", Published in: High Performance Computing & Simulation (HPCS), 2015 International Conference on Date of Conference: 20-24 July 2015 Page(s): 229 – 236.

[4] Deshpande, U.; Binghamton Univ., Binghamton, NY, USA; Keahey, K." Traffic-Sensitive Live Migration of Virtual Machines", Published in: Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on Date of Conference: 4-7 May 2015 Page(s): 51 – 60.

[5] Kejiang Ye; Coll. of Comput. Sci., Zhejiang Univ., Hangzhou, China; Xiaohong Jiang; Ran Ma; Fengxi Yan" VC-Migration: Live Migration of Virtual Clusters in the Cloud", Published in: Grid Computing (GRID), 2012 ACM/IEEE 13th International Conference on Date of Conference: 20-23 Sept. 2012 Page(s): 209 – 218.

[6] Ni, K.; MIT Lincoln Lab., Lexington, MA, USA; Armstrong-Crews, N.; Sawyer, S." Geo-registering 3D point clouds to 2D maps with scan matching and the Hough Transform", Published in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on Date of Conference: 26-31 May 2013 Page(s):1864 – 1868.

[7] Heejeong Lee; Inf. Technol. Lab., Seoul Nat. Univ.; Sang-Chul Lee; Young-Wha Seo; Tae-Wan Kim," Registering 3D Scanned Point Cloud Using Markers International Joint Conference 2006 (SICE-ICCAS 2006)", Published in: SICE-ICASE, 2006. International Joint Conference Date of Conference: 18-21 Oct. 2006 Page(s): 4634 – 4638.

[8] Hall, D.J.; Stanford Research Institute; Endlich, R.M.; Wolf, D.E.; Brain, A.E." Objective Methods for Registering Landmarks and Determining Cloud Motions from Satellite Data", Published in: Computers, IEEE Transactions on (Volume:C-21 , Issue: 7) Page(s): 768 – 776.

[9] Bucksch, A.; Dept. of Geosci. & Remote Sensing, Delft Univ. of Technol., Delft, Netherlands; Khoshelham, K." Localized Registration of Point Clouds of Botanic Trees", Published in: Geoscience and Remote Sensing Letters, IEEE (Volume:10 , Issue: 3) Page(s): 631 – 635.

[10] Irfanoglu, M.O.; Bogazici Univ., Istanbul, Turkey; Gokberk, B.; Akarun, L." 3D shape-based face recognition using registered surface similarity", Published in: Signal Processing and Communications Applications Conference, 2004. Proceedings of the IEEE 12th Date of Conference: 28-30 April 2004 Page(s): 571 – 574.

[11] Sisu Xi; Dept. of Comput. Sci. & Eng., Washington Univ. in St. Louis, St. Louis, MO, USA; Wilson, J.; Chenyang Lu; Gill, C." RT-Xen: Towards real-time hypervisor scheduling in Xen", Published in: Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on Date of Conference: 9-14 Oct. 2011 Page(s): 39 – 48.

[12] Yun Chan Cho; SungKyunKwan Univ., Suwon; Jae Wook Jeon" Sharing data between processes running on different domains in para-virtualized xen" Published in: Control, Automation and Systems, 2007. ICCAS '07. International Conference on Date of Conference: 17-20 Oct. 2007 Page(s):1255 – 1260.

[13] Weikuan Yu; Comput. Sci. & Math., Oak Ridge Nat. Lab., Oak Ridge, TN; Vetter, J.S." Xen-Based HPC: A Parallel I/O Perspective", Published in: Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on Date of Conference: 19-22 May 2008 Page(s): 154 – 161.

[14] Tafa, I.; Inf. Technol. Fac., Polytech. Univ. of Tirana, Tirana, Albania; Beqiri, E.; Paci, H.; Kajo, E." The Evaluation of Transfer Time, CPU Consumption and Memory Utilization in XEN-PV, XEN-HVM, OpenVZ, KVM-FV and KVM-PV Hypervisors Using FTP and HTTP Approaches", Published in:Intelligent Networking and Collaborative Systems (INCoS), 2011 Third International Conference on Date of Conference: Nov. 30 2011-Dec. 2 2011 Page(s): 502 – 507.

[15] Kun Cheng; State Key Lab. of Software Dev. Environ., Beihang Univ., Beijing, China; Yuebin Bai; Rui Wang; Yao Ma" Optimizing Soft Real-Time

Scheduling Performance for Virtual Machines with SRT-Xen", Published in: Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on Date of Conference: 4-7 May 2015 Page(s): 169 – 178.

[16] Yifeng Sun; Dept. of Comput. Sci. & Technol., Peking Univ., Beijing, China; Yingwei Luo; Xiaolin Wang; Zhenlin Wang," Fast Live Cloning of Virtual Machine Based on Xen", Published in: High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on Date of Conference: 25-27 June 2009 Page(s):392 – 399.

[17] Jyun-Shiung Yang; Grad. Inst. of Networking & Multimedia, Nat. Taiwan Univ., Taipei, Taiwan; Pangfeng Liu; Jan-Jan Wu" Workload characteristics-aware virtual machine consolidation algorithms", Published in: Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on Date of Conference: 3-6 Dec. 2012 Page(s):42 – 49.

[18] Liu Guangqi; Shandong Comput. Sci. Center, Shandong Provincial Key Lab. of Comput. Network, Jinan, China; Wang Lianhai; Zhang Shuhui; Xu Shujiang," Memory dump and forensic analysis based on virtual machine", Published in: Mechatronics and Automation (ICMA), 2014 IEEE International Conference on Date of Conference: 3-6 Aug. 2014 Page(s): 1773 – 1777.

[19] Ajay Kumara, M.A.; Dept. of Inf. Technol., Nat. Inst. of Technol., Mangalore, India; Jaidhar, C.D." Virtual machine introspection based spurious process detection in virtualized cloud computing environment", Published in:

[20] Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on Date of Conference: 25-27 Feb. 2015 Page(s): 309 - 315