Short Paper: Autonomic Adaptation of Virtual Distributed Environments in a Multi-Domain Infrastructure*

Dongyan Xu, Paul Ruth, Junghwan Rhee
Department of Computer Science
Purdue University
West Lafayette, IN 47907, USA
{*dxu, ruth, rhee*}*@cs.purdue.edu*

Rick Kennell, Sebastien Goasguen
Rosen Center for Advanced Computing
Purdue University
West Lafayette, IN 47907, USA
{*linux, sebgoa*}*@purdue.edu*

Abstract

By federating resources from multiple domains, a shared infrastructure provides aggregated computation resources to a large number of users. With rapid advances in virtualization technologies, we propose the concept of virtual distributed environments as a new sharing paradigm for a multi-domain shared infrastructure. Such virtual environments provide users with confined, customized platforms to execute legacy parallel/distributed applications. Furthermore, we propose to support autonomic adaptation *of virtual distributed environments, driven by both dynamic availability of infrastructure resources and dynamic application resource demand. We identify new research challenges and describe our on-going work and preliminary results.*

1 Motivation

The growth of shared distributed infrastructures such as the Grid and PlanetLab has made computing and communication resources available to a large community of users. Meanwhile, virtualization technologies [2, 6, 8, 26, 27] have been increasingly deployed on top of such shared physical infrastructures [1, 7, 9, 15, 22, 23], supporting the customization, mutual isolation, and administrator privilege of virtual machines (VMs) running applications on behalf of different users. With recent advances in network virtualization techniques [10, 11, 13, 14, 24, 25], virtual private networked environments can also be created on top of a shared distributed infrastructure. For example, we have developed the VIOLIN virtual distributed environment (or VIOLIN for brevity) [13, 20], which consists of VMs connected by a virtual network and decouples the ownership, configuration, and administration from those of the underlying infrastructure. Within a VIOLIN, the user is able to execute and interact with unmodified parallel/distributed applications. Any negative impacts of these applications will be contained without causing damage to the infrastructure.

The all-software virtualization of distributed computing environments brings a new opportunity to make these environments autonomically adaptive [4, 19, 21, 25]. The vision is that a virtual distributed environment will cater to the application running inside, by dynamically adapting and re-locating itself across the multi-domain infrastructure. More specifically, for a distributed/parallel application, the underlying virtual environment may change its VM and link capacity, re-configure its topology and scale (e.g., by adding new VMs), or even migrate its VMs to more resource-sufficient locations in the infrastructure. Virtual environment adaptation is a unique capability that a physical environment cannot achieve dynamically at runtime. Complementing the traditional adaptation scheme where "applications adapt to dynamic environments", we advocate a new scheme where "(virtual) environments adapt to dynamic applications". The VNET system [25] is among the first to propose and demonstrate the power of runtime adaptation of virtual environments.

As a motivating example, consider a virtual distributed environment that executes NEMO3D [16], a nanoscience parallel simulation that provides quantitative predictions for nanometer-scaled semiconductor devices. The simulation runs in two phases (*Strain* and *Lanczos* phases). The communication pattern among the nodes is a bi-directional ring. While Phase 2 (*Lanczos*) is less network intensive than Phase 1 (*Strain*), it is much more CPU and

*This work was supported in part by National Science Foundation Grants OCI-0438246, OCI-0504261, CNS-0546173.

memory intensive than Phase 1. If the underlying virtual environment is able to adapt to the change of phase by increasing the VMs' CPU and memory allocation after Phase 1, it will result in a shorter execution time of Phase 2. If one or more VMs cannot scale up their capacity due to other workloads in the same physical hosts, the VMs may migrate to other hosts. With recent advances in virtual machine migration technique [5], the end-to-end execution time of NEMO3D (from tens of minutes to hours) can be significantly improved via virtual environment adaptation.

In summary, the adaptation of virtual distributed environments is driven by two main factors: (1) the dynamic, heterogeneous availability of infrastructure resources and (2) the dynamic resource needs of the applications running inside the virtual environments. The goal is to improve end-to-end application performance as well as infrastructure resource utilization. Furthermore, it is highly desirable that virtual environment adaptation (including re-location) be transparent to the application and to the user, giving the latter the illusion of a dedicated, well-provisioned networked runtime environment.

2 Challenges

A number of challenges arise in realizing the vision of autonomic virtual environment adaptation in a multi-domain shared infrastructure.

(1) Live adaptation mechanisms The first challenge is to support application-transparent adaptation of virtual distributed environments. Runtime resource (CPU, memory, and bandwidth) re-allocation capability has been supported by current virtual machine platforms [2, 26] while live VM migration has been enabled within a local-area network [5]. For a multi-domain infrastructure, we still need a solution that enables live migration *across* network domains without pausing or checkpointing the application. Such a solution has to meet two requirements not yet satisfied by current techniques. First, VMs in a virtual distributed environment need to retain the same IP addresses and remain connected to each other during the migration. Second, virtual environment migration across domains cannot rely on NFS to maintain a consistent view of the large VM image files.

(2) Logistic service for VM migration To support efficient VM migration across network domains, a logistic service consisting of distributed *depots* is desirable in the physical infrastructure. A depot is created in each infrastructure domain, where VM images are assembled using either local or transferred "parts" (e.g., OSes, libraries, packages, and COW (Copy-on-Write) files) of varying sizes. The function of the logistic service involves solving the following optimization problem: Given (1) a set of depots each with a set of locally available VM parts and (2) the configuration of VMs in a virtual distributed environment and the set of physical hosts to which the VMs will be migrated, how to compute a distributed schedule for VM parts delivery and assembly so that all VMs will be ready in their destination hosts no later than a certain deadline? Existing content distribution services [17], network storage services [3], and VM creation and configuration services [18, 12] are expected to be leveraged in developing such a VM logistics service.

(3) Adaptation decision making The third challenge is to automate the decision making process for virtual environment adaptation. A monitoring and control mechanism is needed to dynamically monitor and adjust the resource allocations and locations of virtual distributed environments, with minimum administrator intervention. More importantly, the following problems need to be addressed: How to "sense" that an application would need more resources to perform well? When the adaptation of one virtual environment negatively affects other virtual environments because of resource sharing, how to decide which one(s) will get more local resources and which one(s) will have to be migrated? If a VM in a virtual environment has to be migrated, where should it go, in the face of the tradeoff between host resource availability and migration overhead?

(4) Adaptation shepherding Finally, autonomic adaptation of virtual distributed environments may conflict with the goal of virtual environment confinement and self-discipline. Especially with the existence of software bugs, vulnerabilities, and exploiting mal-codes, the software inside a VM cannot be assumed trusted. As a result, the adaptivity of virtual environments can potentially be abused by mal-functioning, selfish, or even malicious programs or users to gain *excessive* amount of infrastructure resources. A dilemma exists between the advantage of autonomic adaptation (i.e., better application performance and higher resource utilization) and the negative impact of untrusted adaptation requests. An external, tamper-resistant shepherding mechanism is desirable to "justify and approve" adaptation requests, preventing the abuse of adaptations from inside the virtual environments.

3 On-going Work and Preliminary Results

We are currently addressing the challenges discussed in the previous section. As our preliminary solutions to challenges (1) and (3), we have developed a prototype

of *adaptive* VIOLIN [21] based on Xen 3.0 virtual machine platform. The prototype has been deployed and evaluated in a multi-domain infrastructure at Purdue University. A software adaptation manager oversees multiple adaptive VIOLINs in the infrastructure and dynamically adjust their resource shares and locations across the infrastructure domains, based on simple adaptation policies. The policies are heuristic and aim at balancing the workload within and between domains while minimizing the instances of VM migrations thus the resulting overhead. From the user's point of view, an adaptive VIOLIN is a stable, private LAN of machines dedicated to the user. From the software adaptation manager's viewpoint, the adaptive VIOLIN "catches up with" both the dynamic infrastructure resource availability and the application resource needs.

A key feature of the adaptive VIOLIN prototype is the live cross-domain migration capability. This is achieved by leveraging the Xen VM live migration mechanism [5] - in the *virtual* layer-2 network created by VIOLIN. In addition, the root file system image is also migrated instead of relying on the same NFS. To demonstrate the effectiveness of adaptation mechanisms and policies, we have performed a number of experiments using adaptive VIOLIN running real-world scientific applications. The experimental results show that a small amount of VIOLIN adaptation can lead to non-trivial application performance improvement and higher infrastructure resource utilization. Detailed results from a number of adaptation scenarios are presented in [21].

4 Conclusion

This paper motivates the vision of autonomic adaptation of virtual distributed environments in a shared, multi-domain infrastructure. The vision calls for further research to address new challenges in multiple aspects, including resource allocation, autonomic management, VM migration and logistics, application profiling, and security. Although far from its full realization, initial efforts of ours and others towards this vision have yielded promising results.

References

[1] S. Adabala, V. Chadha, P. Chawla, R. Figueiredo, J. Fortes, I. Krsul, A. Matsunaga, M. Tsugawa, J. Zhang, M. Zhao, L. Zhu, and X. Zhu. From Virtualized Resources to Virtual Computing Grids: The In-VIGO System. *Future Generation Computer Systems*, 2005.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the Art of Virtualization. In *ACM SOSP'03*, 2003.

[3] M. Beck, T. Moore, and J. Plank. An End-to-End Approach to Globally Scalable Network Storage. *ACM SIGCOMM 2002*, Aug. 2002.

[4] J. S. Chase, D. E. Irwin, L. E. Grit, J. D. Moore, and S. E. Sprenkle. Dynamic Virtual Clusters in a Grid Site Manager. In *IEEE HPDC-12*, 2003.

[5] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. In *USENIX NSDI'05*, 2005.

[6] J. Dike. User-Mode Port of the Linux Kernel. In *USENIX Annual Linux Showcases and Conference*, 2000.

[7] R. Figueiredo, P. A. Dinda, and J. Fortes. A Case for Grid Computing on Virtual Machines. *IEEE ICDCS'03*, 2003.

[8] R. Figueiredo, P. A. Dinda, and J. Fortes. Guest Editors' Introduction: Resource Virtualization Renaissance. *IEEE Computer*, 38(5), 2005.

[9] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayor, and X. Zhang. Virtual Clusters for Grid Communities. In *IEEE/ACM CCGrid'06*, 2006.

[10] A. Ganguly, A. Aagrawal, P. O. Boykin, and R. Figueiredo. IP over P2P: Enabling Self-configuring Virtual IP Networks for Grid Computing. In *IEEE IPDPS'06*, 2006.

[11] A. Ganguly, A. Agrawal, P. O. Boykin, and R. Figueiredo. WOW: Self-Organizing Wide Area Overlay Networks of Virtual Workstations. In *IEEE HPDC-15*, 2006.

[12] X. Jiang and D. Xu. vBET: a VM-Based Emulation Testbed. In *ACM Workshop on Models, Methods and Tools for Reproducible Network Research (MoMeTools'03)*, 2003.

[13] X. Jiang and D. Xu. VIOLIN: Virtual Internetworking on OverLay INfrastructure. Technical report, Purdue University, 2003.

[14] M. Kallahalla, M. Uysal, R. Swaminathan, D. E. Lowell, M. Wray, T. Christian, N. Edwards, C. I. Dalton, and F. Gittler. SoftUDC: A Software-Based Data Center for Utility Computing. *IEEE Computer*, 37(11), 2004.

[15] K. Keahey, K. Doering, and I. Foster. From Sandbox to Playground: Dynamic Virtual Environments in the Grid. In *IEEE/ACM Grid'04*, 2004.

[16] G. Klimeck, F. Oyafuso, T. B. Boykin, R. C. Bowen, and P. von Allmen. Development of a Nanoelectronic 3-D (NEMO 3-D) Simulator for Multimillion Atom Simulations and Its Application to Alloyed Quantum Dots. *Computer Modeling in Engineering and Science (CMES)*, 3(5):601–642, 2002.

[17] D. Kostic, A. Rodriguez, J. Albrecht, and A. Vahdat. Bullet: High Bandwidth Data Dissemination Using an Overlay Mesh. *ACM SOSP 2003*, Oct. 2003.

[18] I. Krsul, A. Ganguly, J. Zhang, J. Fortes, and R. Figueiredo. VMPlants: Providing and Managing Virtual Machine Execution Environments for Grid Computing. In *IEEE/ACM SC'04*, 2004.

[19] H. Liu and M. Parashar. Enabling Self-Management of Component Based High-Performance Scientifi c Applications. In *IEEE HPDC-14*, 2004.

[20] P. Ruth, X. Jiang, D. Xu, and S. Goasguen. Virtual Distributed Environments in a Shared Infrastructure. *IEEE Computer*, 38(5), 2005.

[21] P. Ruth, J. Rhee, D. Xu, R. Kennell, and S. Goasguen. Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure. *IEEE Int'l Conf. on Autonomic Computing (ICAC'06)*, June 2006.

[22] S. Santhanam, P. Elango, A. Arpaci-Dusseau, and M. Livny. Deploying Virtual Machines as Sandboxes for the Grid. *USENIX WORLDS'05*, 2005.

[23] A. Shoykhet, J. Lange, and P. A. Dinda. Virtuoso: A System For Virtual Machine Marketplaces. Technical report, NWU-CS-04-39, July 2004.

[24] A. Sundararaj and P. A. Dinda. Towards Virtual Networks for Virtual Machine Grid Computing. In *USENIX Virtual Machine Research and Technology Symposium (VM'04)*, 2004.

[25] A. Sundararaj, A. Gupta, and P. A. Dinda. Increasing Application Performance In Virtual Environments Through Run-time Inference and Adaptation. In *IEEE HPDC-14*, 2005.

[26] VMware. http://www.vmware.com.

[27] A. Whitaker, M. Shaw, and S. D. Gribble. Scale and Performance in the Denali Isolation Kernel. *USENIX OSDI 2002*, Dec. 2002.