Live Migration of Virtual Block Devices

Cristian Zamfir
Department of Computing Science
University of Glasgow
zamf@dcs.gla.ac.uk

Colin Perkins
Department of Computing Science
University of Glasgow
csp@csperkins.org

Peter Dickman
Department of Computing Science
University of Glasgow
pd@dcs.gla.ac.uk

I. Introduction

Today's computing environments do not provide support for seamless user mobility. We argue that the support for user mobility between various computing environments can be improved and present our initial steps in this direction. We are adddressing a scenario where a user can migrate his work environment without interruption across various computers. Migrating a user's computing environment packed as a virtual machine capsule has been recognized as a way to support seamless user mobility [7], [8]. The capsule can be suspended during a user's commute and resumed at the destination allowing the user to continue his work from where he left off. Our approach is new because we are using live migration [3] for the same purpose, allowing the user to continue his work without suspending the virtual machine during the commute. This has the advantage that network connectivity is not restarted, rather it is migrated with unnoticeable downtime to the new destination. Also, batch jobs such as updating the operating system with the latest security patches or a lengthy download can be performed during the commutes. Our approach has the advantage of running at near native performance and has no residual dependencies to the originating site.

In order to realize this scenario a few challenges have to be addressed. Firstly, memory live migration has to be optimized for low bandwidth commodity links such as DSL. Secondly, live migration of virtual machines currently works only with Network Attached Storage (NAS) [3], [5]. Remote storage devices are accessed over the network so that live migration is a straightforward consequence of migrating the network stack. However, to achieve near native performance in a wide area scenario where NAS and distributed file systems are unlikely to be deployed, the storage has to be accessed locally. Therefore, a solution to the problem of mirroring Virtual Block Devices (VBDs) during migration is necessary. Thirdly, we must address provisioning for a bi-directional trust relation between the migrating host and the migration destination.

II. Virtual Block Device Migration

We have developed and evaluated a prototype for Xen [2] live migration without NAS. Virtual block devices are live migrated with the aid of DRBD [6], a popular cluster disk mirroring software.

Our prototype extends Xen live migration with an XML-RPC client and daemon that handle VBD migration through a set of commands issued to the DRBD replicated devices. Live migration commands from Xen's privileged domains are intercepted and signal the destination to start the resynchronization process which copies to the destination the outdated blocks. Once the replicas are up to date, DRBD mirrors writes to the two virtual block devices synchronously meaning that at any moment the drives are up to date. From this moment on, the last stage of memory migration can start and the user's virtual machine can be resumed at the new site. At this point, the two replicas can be detached, leaving no residual dependencies to the originating site. Our prototype shows that live VBD migration is feasible and does not significantly impact migration time and downtime.

This solution works very well in a well-connected cluster. However, our performance evaluation showed a significant decrease in *I/O* performance during migration even for 100Mbps links due to the fact that writing to the two disks is done synchronously. Another shortcoming is that only two peers are supported. Supporting multiple peers and asynchronous disk writes during migration is the subject of future work.

III. Conclusions and Future Work

We have developed a prototype for Xen live migration without NAS. This is the first step towards the goal of using virtual machine live migration as a general-purpose mobility solution. We are also working on a P2P based VBD live migration tool which will be more suitable for our low bandwidth network links scenario. We are aware that a parallel effort [4], which is addressing a different scenario, reported success in live migrating local VBDs using a simple P2P protocol without focusing on the performance of VBD live migration. We plan to develop a performance enhanced design for our scenario and use a Bittorrent P2P protocol for efficiently mirroring VBD chunks during live migration. We shall perform a comparative study between our DRBD and P2P based solutions as well as quantify the impact of network parameters on Xen live migration without NAS.

Migrating a user's environment on top of an untrusted platform poses serious security and integrity risks. In order to provide the means for the hypervisor and the migrating virtual machine to establish a mutual trust relation before migration, we are investigating how to integrate Intel Trusted Execution Technology [1] into our design.

Appendix
Poster Content Summary

The poster will include an outline architecture diagram of our solution, a pictorial representation of the example scenario that motivates this work and a table of initial performance measurements. Key features of this and related projects will be presented alongside brief notes about the technical challenges that have been overcome and those that remain to be solved.

References

[1] Intel trusted execution technology. http://www.intel.com/technology/security/.
[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 *SOSP '03: Proc. of the 19th ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM Press.
[3] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *Proc. of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 273–286, Boston, MA, May 2005.
[4] A. Lagar-Cavilla, N. Tolia, R. Balan, E. de Lara, M. Satyanarayanan, and D. R. O'Hallaron. Dimorphic computing. *Tech. Report CMU-CS-06-123*.
[5] M. Nelson, B.-H. Lim, and G. Hutchins. Fast transparent migration for virtual machines. In *Proceedings of USENIX*, 2005.
[6] P. Reisner. Drbd v8 - replicated storage with shared disk semantics. In *Proc. of the 12th International Linux System Technology Conf*, 2005.
[7] C. P. Sapuntzakis, R. Chandra, B. Pfaff, J. Chow, M. S. Lam, and M. Rosenblum. Optimizing the migration of virtual computers. *SIGOPS Oper. Syst. Rev.*, 36(SI):377–390, 2002.
[8] M. Satyanarayanan, M. Kozuch, C. Helfrich, and D. R. O'Hallaron. Towards seamless mobility on pervasive hardware. In *Pervasive and Mobile Computing 1 (2005) 157–189*.