Virtual Network Management with XEN

Andreas Berl, Andreas Fischer, and Hermann de Meer

University of Passau, Innstr. 43, 94032 Passau, Germany
`{andreas.berl,andreas.fischer,hermann.demeer}@uni-passau.de`

1 Introduction

Due to the rise in hardware capabilities, virtualisation has been rediscovered as a valuable tool introducing an abstraction layer between software and the underlying hardware. One very flexible kind of virtualisation is called "system virtualisation" - it allows to virtualise whole operating systems, mediating access to the underlying hardware via a software layer called "hypervisor". System virtualisation provides the option to run several operating systems in parallel on a single hardware instance.

Recent innovations in hypervisor technology [1] - namely the concept of "paravirtualisation" - allow to virtualise hardware platforms (like the Intel x86 platform) that otherwise could not be virtualised at all [2]. Moreover, this kind of virtualisation is also able to offer quite good performance, due to the necessary cooperation of the guest operating systems. Unfortunately, the power wielded through system virtualisation does not come without cost - the use of virtualisation easily leads to an increased total number of systems that have to be managed.

Nevertheless, the option of using a highly performant form of virtualisation on readily available off-the-shelf hardware platforms opens up a host of new applications for virtualisation. One such application is the virtualisation of networks. Recent work in this area [3–5] shows both the possibilities of virtualised networks as well as the need for a well defined management interface.

2 Main

2.1 Virtualisation of networks

In a virtualised network, physical routers of a network are virtualised in order to get the flexible concept of virtual routers. They are running within a virtual machine (VM), controlled by a hypervisor software which mediates access to hardware between different VMs. Virtual routers offer some advantages in flexibility and management. First, different virtual routers are able to provide different protocol stacks on the same physical machine - possibly optimised for specific applications. Second, since VMs are usually stored in one or more files (or within a file structure), they can easily be moved or copied. Third, even a live migration of VMs is possible, allowing a running service to migrate from one physical hardware component to another with only minimal disruption in service. In general it is possible to start, stop, or pause a given VM at any time, allowing to release currently unused resources and use VMs only on demand. These basic functions also enable more complex scenarios - e.g. making a backup of a VR and restoring it in short time, when necessary. This enables a "rollback in time", as it is possible to reactivate a previously backed up router in a defined state. If VRs are interconnected to create networks, it is possible to define virtual topologies (VT) between them, which are different to the real physical topology (RT). Moreover, not only can the VT differ from the RT, but also bandwidth and other properties of links can be different in virtual networks formed by the VRs.

Experiments show that it is possible to change properties (e.g. bandwidth) of the virtual networks dynamically, based on the current demand of services. It is also possible to host multiple "paused" virtual networks with different topologies, which are preconfigured for a certain scenario (e.g. as an emergency procedure) and can be launched immediately in a predefined state. This means, that not only VTs can be different from RTs, but also that VTs can be dynamic, i.e. they change over time.

2.2 Management of virtual networks

To offer access for client software to virtualised resources, a clear definition of a virtualisation management interface is necessary. Several basic service primitives have been proposed, following the suggestions of the Distributed Management Task Force (DMTF) [6, 7]. These include functions to start, stop, pause, resume and/or move a virtual machine.

It is vital that service primitives also include functions for monitoring virtual and physical resources. This is necessary to be able to dynamically react to resource bottlenecks and to cater to the needs of new, unforeseen service requirements. These monitoring values include (among others) available RAM, available CPU cycles, and available harddrive space. However, in order to preserve the abstraction and the flexibility gained with it, the virtualisation management interface must not allow direct access to unmodified monitoring data of physical hardware. Instead, the underlying management software has to modify the monitoring values, accounting for any overhead produced by the virtualisation solution. This ensures that client software can remain ignorant about the actual virtualisation solution employed, being only concerned with advertised resources.

More complex, high level methods can be designed by grouping together basic service primitives. This avoids client software having to know about the entire management interface. Instead, clients can concentrate on a specific task, like performance-, fault-, or topology-management functions, disregarding all primitives not concerned with that task.

The interface also has to offer a way to allow client services to adhere to Quality of Service (QoS) requirements. In order to do so, it is necessary to assign assurable resources, allowing privileged services access to a guaranteed share of the available network services, despite of their competition with other services. The resulting hierarchy of different priorities with regard to services has to be taken care of. This has also an impact on failure recovery - in a QoS aware environment sudden disappearance of a virtual router may not be acceptable. Handling such situations requires sophisticated failure recovery strategies.

3 Conclusions and Future Work

The recent rise of virtualisation techniques also saw an increased interest in security issues inherent to virtualisation [8, 9]. A virtualisation management architecture has to take these new threats into account in order to create a dependable system. Moreover, the security implications of a virtual management software have to be considered.

The proposed architecture has yet to be verified at large, showing the feasability of using virtualisation as an enabler for more flexible computer networks. The scalability of XEN as a base for virtual routers has to be evaluated, as problems have been reported [10]. If problems remain, either XEN has to be modified, or the usefulness of other virtualisation solutions has to be examined. Also the applicability of this approach to a wider spectrum of network services, shifting the focus from virtual routers to generic virtual services, has to be researched.

Acknowledgments. Parts of the work in this paper were undertaken in the context of the AutoI project [11, 12], EuroFGI [13] and EuroNF [14] - Networks of Excellence, which are partially financed by the EU. This paper was also partly funded by the German Research Foundation (Deutsche Forschungsgemeinschaft - DFG), contract number ME 1703/4-2.

References

1. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles. (October)
2. Popek, G.J., Goldberg, R.P.: Formal requirements for virtualizable third generation architectures. Communications of the ACM **17**(7) (July)
3. Berl, A., Fischer, A., Meer, H.D., Galis, A., Rubio-Loyola, J.: Management of virtual networks. In: 4th IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management - EVGM2008, Samos Island, Greece (September 2008)
4. Davy, S., Fahy, C., Griffin, L., Boudjemil, Z., Berl, A., Fischer, A., Meer, H.D., Strassner, J.: Towards a policy-based autonomic virtual network to support differentiated security services. In: TEMU 2008 - International Conference on Telecommunications & Multimedia, Ierapetra, Crete, Greece (July 2008)
5. Fahy, C., Davy, S., Boudjemil, Z., der Meer, S.V., Rubio-Loyola, J., Serrat, J., Strassner, J., Berl, A., Meer, H.D., Macedo, D.: An information model that supports service-aware self-managing virtual resources. In: 3rd IEEE International Workshop on Modelling Autonomic Communications Environments - MACE2008, Samos Island, Greece (September 2008)

6. : System virtualization profile (August 2007) DMTF profile DSP1042, version 1.0.0a, http://www.dmtf.org/standards/published_documents/DSP1042.pdf.
7. : Virtual system profile (May 2007) DMTF profile DSP1057, version 1.0.0a, http://www.dmtf.org/standards/published_documents/DSP1057.pdf.
8. Rutkowska, J.: Subverting vista kernel for fun and profit. In: Black Hat 2006, Las Vegas, Nevada, USA (August 2006)
9. King, S.T., Chen, P.M., min Wang, Y., Verbowski, C., Wang, H.J., Lorch, J.R.: Subvirt: Implementing malware with virtual machines. In: IEEE Symposium on Security and Privacy. (May 2006)
10. Egi, N., Greenhalgh, A., Handley, M., Hoerdt, M., Mathy, L., Schooley, T.: Evaluating xen for router virtualization. In: 16th International Conference on Computer Communications and Networks - ICCCN 2007. (August 2007) 1256–1261
11. Bassi, A., Denazis, S., Galis, A., Fahy, C., Serrano, M., Serrat, J.: Autonomic internet: A perspective for future internet services based on autonomic principles. In: IEEE 3rd International Week on Management of Networks and Services End-to-End Virtualization of Networks and Services - Manweek 2007 / MACE 2007 2nd IEEE International Workshop on Modelling Autonomic Communications Environments, San Jos, California, USA (October 2007)
12. : Autonomic internet (autoi) (2007) EU FP7 IST Project, http://ist-autoi.eu/.
13. : Future generation internet (eurofgi) (2006) Network of Excellence, FP6, grant no. 028022, http://eurongi.enst.fr/.
14. : European network of the future (euronf) (2007) Network of Excellence, FP7, grant no. 216366, http://euronf.enst.fr/.