

# MetaCenter Virtual Networks

David Antoš, Jiří Sitera, Petr Holub, and Luděk Matyska

CESNET, z. s. p. o.  
Zikova 4, 160 00 Prague 6, Czech Republic  
{antos|hopet|ludek}@ics.muni.cz, sitera@civ.zcu.cz

## Abstract

The MetaCenter is both Czech national Grid and supercomputing infrastructure and a research project focused on sustainable advancement of this infrastructure. We describe a research activity bringing virtual network service to MetaCenter. It covers the reasons for implementing VirtCloud, system for managing virtual networks, expected use-cases, design and implementation constraints and interaction with other services (management of virtual machines, batch systems, network core, security and others). The paper is concluded by the first experiences with experimental deployment of the MetaCenter virtual networks.

## 1 Introduction

In the last years, advances of the MetaCenter, the Czech national Grid and supercomputing infrastructure, are tightly connected with virtualisation concepts and technologies. Major part of MetaCenter computation resources is currently virtualised, gaining flexibility for both end users and resource owners. Virtual machines are managed by Magrathea, a service we designed and implemented [10], while the virtual nature of the resources is mostly transparent for the end users thanks to integration with the job management system representing the primary interface for users.

As the MetaCenter computation nodes are scattered over the three major cities of the Czech Republic (Brno, Pilsen, Prague), the project traditionally builds on advanced services of the Czech National Research and Education Network CESNET2+<sup>1</sup>.

Recently, MetaCenter introduced the concept of *virtual clusters*, composed from a set of virtual machines running user specified environment (up to and including operation system). Although the network community already introduced the end-to-end services [9], they are still insufficient to support full virtual clusters. Based on the available end-to-end functionality, we have designed a VirtCloud system for interconnecting nodes of virtual clusters over a wide-area (state-wide) network, with a design not specific to the CESNET2+ network.

Traditionally, the network is understood as a “fixed resource” in Grid computing. The purpose of the network is just to transfer data, the network is omnipresent in the Grid, and no advanced functionality is expected beyond support of large bulk data transfers. The only parameters of interest are related to performance measures like throughput and latency.

---

<sup>1</sup><http://www.ces.net/>, topology on <http://www.ces.net/network/>

This view is not sufficient for virtual clusters. Virtual clusters are dynamically mapped to the physical infrastructure, this mapping is indirectly controlled by the user by means of Grid middleware. Logical structure of virtual clusters does not necessarily correspond to the physical network topology. We understand the network as “just another resource,” dynamically managed by the resource scheduling system, and adapting to the needs of virtual clusters (e.g., connecting to the outside Internet, filtering the traffic, user access to the clusters, ...).

In this paper, we study use cases that lead to the VirtCloud design (Sec. 2) and the design itself (Sec. 3). First experiences with the behaviour of the network are briefly discussed in Sec. 4. Related work is summarised in Sec. 5 and the conclusion is given in Sec. 6.

## 2 Use Cases and Design Considerations

We considered following use cases as typical requirements for the VirtCloud system. The use cases are not mutually disjoint, some of them lead to a single technical solution. We divide them roughly into several groups.

### 2.1 Privacy and Security Policies

Privacy and security use cases refer mostly to “protecting the cluster from the outside world” as well as “protecting the outside world against the cluster.”

**Mutual Isolation of Clusters.** This use case is an analogy of the increase of level of separation achieved by virtualisation. Processes belonging to distinct users are separated in a common operating system to a certain level, e.g., users can list all processes on the system but cannot modify/manipulate them. Providing virtual machines to the users, the level of separation increases together with the illusion of “owning” the infrastructure (e.g., a user cannot see processes running on other virtual machines on the same physical host). Nevertheless, if users have administrator privileges in the virtual machines (we will see later how this can be done in a secure manner), the network traffic must be separated among the clusters, otherwise a user could eavesdrop network traffic of others.

**User-Provided OS Images and Security of the Infrastructure.** We have two scenarios to consider.

1. The user runs MetaCenter approved virtual machine image without administrative privileges. The infrastructure owner can take full responsibility for security of the virtual machines, the machines can be directly connected to the Internet.
2. The user (a) runs his/her own virtual machine image and/or (b) he/she has administrator privileges in the virtual machine. In that case, it is not possible for the infrastructure owner to take responsibility for the security of the machines. Generally, the machines must not be accessible from the Internet using address space belonging to the infrastructure owner.

The type of network connectivity should be automatically decided by the scheduler when virtual cluster is allocated based on the requested type of OS images of computing nodes.

**Legacy Insecure Services and Components.** While user provided virtual machine images are by definition considered insecure, users may want to run insecure components even in case they do not use their own operating system images. Typically, legacy software may depend on libraries and components that are known to have security flaws (and upgrading the libraries breaks the software), which is unacceptable on a shared publicly accessible computation infrastructure. Requiring secure components is natural for any professional infrastructure provider, but difficult to explain to the user (“but this is no problem in our departmental cluster”). It can be solved by controlling access to the cluster network.

## 2.2 Networking Related Use Cases

**Limited Layer 3 Address Space.** The IPv4 address space is very tight even for the physical machines in the clusters. Adding virtual machines, the amount of necessary addresses per single physical node is practically unlimited. While IPv6 is the preferred way to solve limited amount of IP addresses, it has severe practical drawback: the support of IPv6 in applications is usually not in production quality [2].

Separating Layer 2 networks of virtual clusters allows arbitrary Layer 3 addressing schemes independent of actual network topology, e.g., using IPv4 addressed networks behind NAT even spread over the whole underlying network.

**Multiple Instances of Hardcoded IP Addresses.** One of MetaCenter user groups uses a set of applications with hardcoded IP addresses. A cluster of such applications can be run just in a single instance on a local network, otherwise the traffic of multiple instances of the cluster would obviously interfere. In order to allow running multiple instances of the cluster to run simultaneously, the clusters must be separated below network layer (i.e., either physically and/or at the link layer).

**User Access to the Cluster.** User access to the cluster must be provided by a tunnelling service, enabling a user’s workstation to become a part of the cluster.

**Cluster as a Part of User’s Address Space.** The user may want to publish the cluster to the Internet even in case of clusters that are considered “insecure” by the infrastructure provider. In that case, the user may connect the cluster to his/her local network by means of routing the tunnelled connection. As Layer 3 addressing scheme is sole discretion of the user, the cluster may be accessible through user’s router under public IP addresses, hidden behind NAT, etc. In all

cases, it is responsibility of the user to keep the cluster secure and the user is to blame in case of a security incident.

**Virtual Machine Migration.** Virtual machine migration increases the flexibility of the whole environment, but it needs specific network support. It is not possible to change Layer 3 (IP) address of the migrated machine as the application layer usually is not prepared to cope with such a dynamic change (e.g., in case of MPI jobs).

### 2.3 Other Restrictions

Reasonable low latency to set up network of the cluster is necessary in order to support interactive jobs.

Throughput of the private cluster network must not be (significantly) worse than that of native IP network. The expected lifetime of the virtual clusters is expected to range from hours to months (i.e., the cluster is not built for an ordinary single job), creating a virtual cluster is initiated by user requirement and managed by Grid middleware.

The network must be able to operate over a wide-area backbone. After initial configuration of the backbone allowing the system to operate, no other configuration on the backbone is acceptable at runtime.

## 3 VirtCloud Design and Implementation

It follows from the use cases that the only reasonable layer to separate the virtual clusters is the link layer (L2). Physical layer separation is much less flexible and scalable (even if state-of-the-art optical cross-connects were used). We therefore close the virtual clusters into Virtual Local Area Networks (VLANs). Each VLAN has a flat switched topology over at least all sites hosting physical computers participating in the virtual cluster. Switched network enables all use cases we have discussed above. The common Layer 2 network allows us to build geographically distributed clouds of virtual machines on a single logical network segment with the possibility of transparent migration without any noticeable borders previously represented by separate MetaCenter sites. With the possibility to dynamically create many virtual networks we can also make a topology totally independent of the physical one. Each user, group or a specific application can have its resources connected to its own virtual network.

VirtCloud consists of four levels: (a) core L2 network, (b) cluster site network, (c) host configuration, and (d) VLAN life cycle management service.

**L2 core network.** The core network has to maintain flat switched topology for all VLANs in the virtual clusters. Implementation depends on available networking equipment. For performance reasons, we are only interested in multipoint mechanisms supported directly by hardware of high-end research networks.

In the CESNET2+ network, following technologies can be used to fulfil the requirements of the VirtCloud system: (a) *IEEE 802.1ad (QinQ)* that allows encapsulation of 802.1q tagging into another 802.1q tagging, (b) *Virtual private LAN service (VPLS)* creating a shared broadcast domain using MPLS, and (c) *Cisco Xponder technology* creating a distributed switch on dedicated DWDM optical circuits.

**Cluster site network.** The site is required to support 802.1q VLAN trunking and capability of interfacing to the core network. In MetaCenter, each site is implemented using a mix of Force10, HP, and Cisco Ethernet switches. When building a virtual cluster, a VLAN number is allocated and switches configured to attach the VLAN to the chosen tunnelling mechanism through the core network.

**Host configuration.** Each physical host is connected to at least one interface that supports 802.1q trunking. In the implementation, hosts run Xen virtual machine monitor [3]. The hypervisor manages user domains and provides them with Ethernet bridges. Logical network interface of each domain is bridged to VLANs depending on membership of user domains in virtual clusters. Tagging the VLANs in the hypervisor is purely under control of Grid middleware, out of user's reach.

**VLAN life cycle management.** Life cycle of VLAN is closely related to the life cycle of virtual clusters themselves. We have developed a stateful service called SBF<sup>2</sup>. Virtual clusters are built upon user action, submitting a special job into a resource scheduler PBS. The PBS selects a set of physical machines to run the cluster and requests allocation of VLAN number from SBF. SBF configures the site switches and PBS with cooperation of Magrathea [10] configures bridges in Xen hypervisors and boots requested virtual machines. The tear-down phase (initiated by user or administrative action and/or timeout) proceeds in the reversed order.

**VirtCloud access service.** In the case of virtual network with restricted outside connection an important part of the VirtCloud architecture is a gateway providing access to the virtual network. We are currently testing a prototype of such service providing a VPN tunnel from the virtual network to its user. The tunnel can be used to implement both connectivity of a single user's machine into the virtual cluster or to provide stable "publication" of the cluster in a user's network space.

The prototype is based on OpenVPN server and is able to connect to appropriate virtual network based on authenticated user identity and authorization database. If the user is allowed to access more virtual networks he/she can select the desired virtual network using a MetaCenter attribute certificate service.

---

<sup>2</sup>Pronounceable abbreviation for Slartibartfast, Magrathean coastline designer from *The Hitchhiker's Guide to the Galaxy* by Douglas Adams

TCP: Brno–Prague	
Xponders, phys.	936 Mbit/s
Xponders, Xen	936 Mbit/s
VPLS, Xen	935 Mbit/s
Native IP, Xen	592 Mbit/s

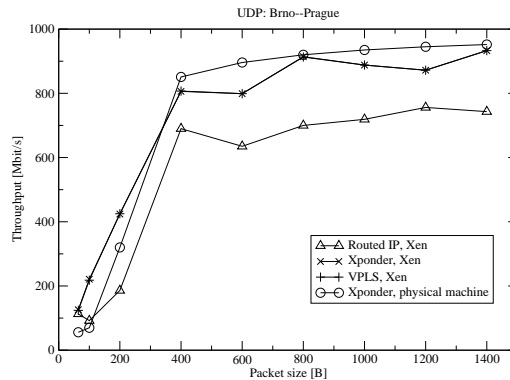


Fig. 1: TCP and UDP: Xponders, VPLS, and routed IP to Prague

We plan to enrich the access gateway prototype with support of SSH. It is for many users preferred way how to access a virtual cluster for occasional inspection.

## 4 First Experiences

A set of experiments has been run to show feasibility of the VirtCloud concept: performance of the virtualised network must not be significantly worse than the native IP network. In this paper, we show only a sample of the results, detailed set will be published in a technical report.

The machines used for the experiments are located in all MetaCenter sites: Brno, Pilsen, and Prague. In Brno, the node has two dual-core Intel Xeon 5160 3 GHz CPUs, 4 GB memory, and PCI Express GE adapter Intel 80003ES2. Prague node is a quad-core Intel Xeon X5365 3 GHz, 16 GB physical memory, and PCI Express GE adapter Intel 80003ES2. Pilsen node is a dual AMD Opteron 270, 2 GHz, 8 GB memory, and PCI GE adapter Broadcom NetXtreme BCM5704. All the machines run Xen 3.1.3, both hypervisors and user domains with kernel 2.6.22.17. Hypervisor domains have 1 GB of memory, user domains the rest of memory available on the particular machine.

Throughput was measured with iperf version 2.0.2, UDP throughputs are the highest possible that achieve packet loss at most 0.5%.

Figure 1 shows TCP and UDP throughput on the Brno–Prague line. The Xponders in physical machine, being a dedicated network without any Xen overhead, represent the practical limit. Virtualisation of the nodes (i.e., Xen results) show a small overhead in the Xponder network. The UDP performance of Xen is slightly better than of physical machines in case of shortest packets, this is probably due to larger buffers available in the virtual network interface. Surprisingly, Xponders and VPLS (where the traffic shares the standard backbone) reach practically identical performance. The native routed IP network is clearly worse in both TCP and UDP.

## 5 Related Work

Network virtualisation approaches can be roughly divided into three categories. Virtual LANs (Local Area Networks) provide virtual LAN over a more complex infrastructure, Virtual Private Networks simulate presence of a network interface in a distant network, and Overlay Networks are usually deployed to transport traffic through a adverse environment, e.g., firewalls, NATs, etc. Overlay Networks duplicate vertically a part of the network stack and use tunnelling from protocol point of view. Deployment of those techniques is tightly coupled with the environment that is to be spanned, topology of the network, geographical distribution, restrictions in the network, and the need of isolation.

Tunnelling methods are typically deployed in distributed networks unfriendly for transporting usual communication among clusters. Violin [5] is an overlay network based on UDP tunnels. In-VIGO [1, 7] sets a system of tunnels and VPNs to separate machines into logical clusters called VNET. VNET [11] builds a logical Ethernet bridge over IP network. Tunnelling the Ethernet traffic into IP, the performance of VNET is limited by the throughput of the underlying IP network. Hamachi<sup>3</sup> is an example of application-level overlay network for connecting computers behind firewalls and NATs.

When building a virtual cluster in an unrestricted local network, the requirements are driven by the necessity of cluster separation. It is possible to separate the clusters on network layer (L3), assigning mutually disjoint IP addresses to them, similarly to Cluster-on-demand [4]. This level of separation is not sufficient in our scheme: users having administrator privileges in their virtual machines are free to set arbitrary IP addresses, possibly also intruding other virtual clusters. Nakada et al. [8] describe a system for VLAN configuration for package system called Rolls. Wide-area network is nevertheless not considered. Nimbus (known as Virtual Workspace Service) [6] is a management system for virtual machines. Configuration of network interfaces is supported without the possibility of creating controlled network environment.

## 6 Conclusions and Future Work

The VirtCloud architecture we presented in this paper is based on requirements of the MetaCenter user community. The system is targeted for building virtual computing facilities over wide-area networks (covering the whole country), user controlled by means of Grid middleware, encapsulating the clusters and managing accessing them in a controlled manner. The experiences gained with the prototype implementation show the approach is feasible with currently available technologies.

Many questions left for further investigation remain. Layer 3 addressing scenarios must be prepared for detailed cluster usage patterns. Methods of efficient access to external resources in case of closed clusters must be also studied, as well as methods for publishing encapsulated clusters. Although conceptually

---

<sup>3</sup><http://logmeinhamachi.com/>

simple, they create many practical problems to investigate when implemented on real Grid infrastructure.

**Acknowledgements.** We would like to thank Zdeněk Salvét, Václav Novák, Josef Verich, Pavel Šmrha, Miroslav Ruda, Jiří Denemark, and Lukáš Hejtmánek from CESNET and Sitola Laboratory. This project has been supported by research intents “Optical Network of National Research and Its New Applications” (MŠM 6383917201) and “Parallel and Distributed Systems” (MŠM 0021622419).

## 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*, 21(6):896–909, 2005.
2. D. Antoš, J. Sitera, and D. Kouřil. IPv6 in METACenter. Technical Report 5/2008, CESNET, z. s. p. o., 2008.
3. 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: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM Press. ISBN 1-58113-757-5.
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 *Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing*, pages 90–100, 2003. ISBN 0-7695-1965-2.
5. X. Jiang and D. Xu. VIOLIN: Virtual Internetworking on Overlay Infrastructure. In *Proc. 2nd Int'l Symp. Parallel and Distributed Processing and Applications*, number 3358 in LNCS, pages 937–946. Springer-Verlag, 2004. ISSN 0302-9743.
6. K. Keahey, I. Foster, T. Freeman, and X. Zhang. Virtual workspaces: Achieving quality of service and quality of life in the Grid. *Scientific Programming*, 13(4):265–275, October 2005. ISSN 1058-9244.
7. I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. Figueiredo. VM-Plants: Providing and Managing Virtual Machine Execution Environments for Grid Computing. In *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*, page 7, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2153-3.
8. H. Nakada, T. Yokoi, T. Ebara, Y. Tanimura, H. Ogawa, and S. Sekiguchi. The Design and Implementation of a Virtual Cluster Management System. In *EVGM 2007, 1st IEEE/IFIP International Workshop on End-to-end Virtualization and Grid Management*, San Jose, CA, USA, October 2007.
9. V. Novák, P. Šmrha, and J. Verich. Deployment of CESNET2+ E2E Services. Technical Report 18/2007, CESNET, z. s. p. o., December 2007.
10. M. Ruda, J. Denemark, and L. Matyska. Scheduling Virtual Grids: the Magrathea System. In *Second International Workshop on Virtualization Technology in Distributed Computing*, pages 1–7, USA, 2007. ACM Digital Library.
11. A. I. Sundararaj and P. A. Dinda. Towards Virtual Networks for Virtual Machine Grid Computing. In *Proceedings of the 3rd USENIX Virtual Machine Research and Technology Symposium*, pages 177–190, 2004.