VirtCloud: Virtualising Network for Grid Environments—First Experiences

David Antoš, Luděk Matyska, Petr Holub, and Jiří Sitera
CESNET, z. s. p. o.
Zikova 4, 162 00 Prague, Czech Republic
antos@ics.muni.cz, ludek@ics.muni.cz, hopet@ics.muni.cz, sitera@civ.zcu.cz

Abstract

Networking infrastructure is a vital part of virtual computer clusters. This paper describes VirtCloud, a system for interconnecting virtual clusters in a state-wide network based on advanced features available in academic networks. The system supports dynamic creation of virtual clusters without the need of run-time administrative privileges on the backbone core network, encapsulation of the clusters, controlled access to external sources for cluster hosts, full user access to the clusters, and optional publishing of the clusters. The paper describes architecture of the system, and prototype implementation in MetaCenter (Czech national Grid infrastructure) using Czech national research network CESNET2. Feasibility of the concept is evaluated through a series of measurements demonstrating that the network performance of the system is satisfactory.

1. Introduction

Virtualisation has become one of the key paradigms in Grid computing in recent years. It enables tailoring of Grid environments to the needs of their users in previously unprecedented way, thus making them more attractive for broader user communities.

Virtualising computer clusters also involves the interconnecting networking infrastructure. While steps to virtualising the network inside the cluster have already been taken by several groups [1], [2], [3], [4], [5], this paper focuses on building sustainable virtualised networking infrastructure that scales enough to interconnect clusters in wide-area networks and that performs up to the expectations of the high-performance applications.

VirtCloud is a system for internetworking dynamic virtual clusters over a state-wide network, supporting encapsulation of the clusters and publishing them in controlled manner. This allows for both protecting the cluster from the outside world and protecting the world from the cluster (e.g., in case of user-supplied virtual images). The system is driven by Grid middleware. While VirtCloud uses services of the backbone network, it is designed to run without the need of run-time configuration of the core network. The design (described in Section 2) is not limited to our primary target network: as we discuss in Section 3, it is able to use several mechanisms for traffic encapsulation.

The architecture has been prototyped in the Grid environment of the MetaCenter project[1]—Czech state-wide Grid infrastructure—using the CESNET2 backbone network[2] that spans the whole Czech Republic with interconnects to other European and world-wide networks.

Interfering the networks in large areas can have serious performance implications. We have done a series of basic measurements to show performance feasibility of our approach (Section 4). Section 5 summarizes related work and the paper concludes with Section 6 providing final remarks.

2. VirtCloud Design

We describe analysis and design of VirtCloud system in this section. We start with design considerations (DC) that reflect usage patterns of the system and practical restrictions, followed with an overview of the architecture.

2.1. Design Considerations

We divided the design considerations into three categories that reflect different points of view. We start with the network considerations:

DC-1 *High-performance virtual private network* with performance not significantly worse than running the infrastructure with normal networking interconnects.

DC-2 *Dynamic virtual cluster network creation.* Virtual clusters have expected lifetime ranging from hours to months. Clusters are built upon user request and/or administrative action in case of long-term clusters for special user groups.

DC-3 *Encapsulation of virtual clusters.* No communication outside of the network unless specifically enabled due to security considerations (virtual cluster may run insecure images provided by the users).

DC-4 *Capable of being deployed in state-wide and international environments.* It needs to support sufficient encapsulation to avoid conflicts with services already

1. http://meta.cesnet.cz/
2. http://www.ces.net/

running in the network. Several mechanism of interfacing with backbone network need to be proposed to increase compatibility with different types of state-wide and international networks.

DC-5 *Operation without administrative privileges on the backbone networks.* After the initial configuration of the backbone networks is done to support VirtCloud, the configuration has to be limited to cluster hosting sites and there should only be well defined interfaces to the backbone networks.

Organization of virtual clusters leads to the following DCs:

DC-6 *Support for interactive jobs.* Low latency to set up the networking environment is required.

DC-7 *Access to the virtual cluster for its user(s).* User needs to be able to get secure interactive access to the virtual cluster, for interactive jobs or for preparation and control of batch jobs, efficient data transfer, etc. This requires more generic interface than, e.g., traditional web portals.

DC-8 *Optional publishing of the cluster.* Two basic scenarios are to consider: (a) a virtual cluster runs certified images and services, and the Grid service provider is responsible for its security, (b) a virtual cluster runs images provided by its users, where users are solely responsible for their security. While direct publishing (i.e., routing the cluster directly to the Internet) is possible and even suitable for performance reasons in the first case, the latter case requires indirect publishing through the network of the user, so that the user is fully responsible for possible security incidents. Closing the cluster into a VLAN is nevertheless reasonable even in case (a), the type of the cluster can change during its lifetime.

DC-9 *Jobs on the cluster may need to access external data and services.* For some job types, access to data and/or services residing on locations outside of the virtual cluster may be required. This should be implemented as a network connection initiated from inside (unidirectional in this sense), i.e., for this purpose, there should be no services running on the virtual cluster that would be available from outside of the cluster for security reasons.

DC-10 *Migration of virtual machines* has serious implications for applications if Layer 3 addresses change. For migration feasibility, Layer 3 addresses should be fixed.

DC-11 *Multiple simultaneous instances of the same virtual cluster with fixed Layer 3 addresses* (e.g., legacy applications with hard-coded addresses in user images) need sufficient encapsulation below Layer 3.

Interoperability and legacy considerations lead to the following DC:

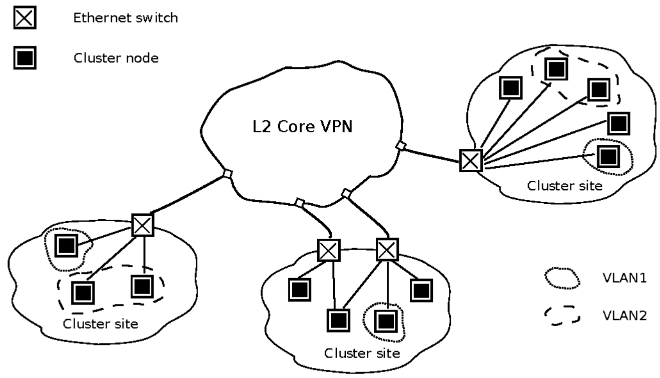

Figure 1. Architecture of the VirtCloud network.

DC-12 *Interoperability with Grid virtualisation system(s).* The proposed system must be compatible with existing systems for Grid virtualisation like Magrathea [6] or Nimbus [7], or requiring only modest adaptation of these systems.

2.2. VirtCloud Architecture

After defining DCs, we can proceed to description of VirtCloud architecture and show how it maps onto the DCs.

VirtCloud spans four levels: (1) L2 core network, (2) cluster site network, (3) host configuration, and (4) VLAN life cycle management service. Each virtual cluster VC_i uses its own private network, further denoted as $VLAN_i$. Overall scheme of the architecture is shown in Figure 1. Based on the requirements stated above, each VLAN uses flat switched (Layer 2) topology. The $VLAN_i$ provides encapsulation (DC-3) and spans over at least all the sites hosting computers participating in the VC_i. Switched topology of the VLANs enables easy low-latency migration of the virtual machines over the physical hosts (DC-10), which is fundamentally the same as migration of a networked device in switched local area network. It also supports running multiple simultaneous virtual clusters with the same addressing scheme (DC-11). There are several options how to implement such a network in large-scale infrastructure with respect to requirements DC-4, DC-5, and DC-6 as discussed in Section 3.

Host configuration. Each physical host is connected to the site network using one or more interfaces that support 802.1q trunking. This allows for multiple virtual hosts running on a physical host, each belonging to a different VLAN.

Site network. The site network is a switched network among the physical computer nodes and provides uplink to the core network. The site is required to support 802.1q trunking and be capable of interfacing to core network (which may pose some additional requirements depending on the configuration of the core network).

L2 core network. The core network has to maintain flat switched topology for all VLANs interconnecting virtual clusters, i.e., to provide a logical distributed Layer 2 switch with VLAN support. Actual implementation of the core network depends to some extent on available underlying networking facilities. There are many implementations of switched virtual networks ranging from systems supported directly by network hardware (e.g, IEEE 802.1ad) to application-level systems (e.g., OpenVPN[3], Hamachi[4]). However, for performance reasons, we only focus on virtual networks that can be supported by hardware in high-end academic and research networks (DC-1). Some protocols only support point-to-point bridging (e.g., L2TPv3 [8]) which excludes them from use in the core of the network.

Life Cycle of Virtual Networks. The life cycle of VLANs in the infrastructure reflects the life cycle of virtual clusters themselves (DC-2). Clusters are build upon user action—submission of a special job to the resource manager (DC-12). The resource manager configures network active elements in cluster sites and allocated physical machines to assign traffic from the virtual machines hosted on them to appropriate VLANs. Resource manager then boots requested virtual images. Layer 3 addresses are assigned to the virtual machines according to user needs.

2.3. Access from/to the Virtual Clusters

There are three cases to handle here: (1) user access to the cluster (including publishing it, DC-7, DC-8); access to data and services (DC-9) provided either (2) as a part of the Grid infrastructure or (3) as an external third-party service.

Remote access for the users is provided by several tunnelling services, be it SSH, OpenVPN, etc. Servers for the remote access become part of the cluster with their "inner" interfaces, having their "outer" interface publicly addressable and protected with a standard Grid authentication and authorisation. When the user wants to publish the virtual cluster, there are two ways to do it. If the cluster is built solely from a certified image, it can be published directly from one of the sites. Otherwise, the user may publish the cluster by creating a tunnel to it and providing access through his/her Internet connection—thus accountability for any security-related problems lies on the user.

The access to services that are part of the Grid infrastructure is based on integrating nodes that host these services into the virtual cluster. Choosing which nodes will be integrated into the virtual cluster depends primarily on user's request when building the virtual cluster.

When access to external data sources is necessary, the problematic possibility is using user-provided virtual machine images. The user can either use similar techniques like for publishing the cluster (and, e.g., keep the cluster in his/her address space), or—as an optimisation—some traffic can be administratively permitted and routed directly through one or more sites, preferably through a firewall. It naturally depends on type of virtual machine image used and needs careful judgement, as the Grid infrastructure provider takes part of responsibility over possible security problems. This is nevertheless considered a special feature.

3. VirtCloud Implementation in the MetaCenter using CESNET2 Network

MetaCenter as a national Grid infrastructure utilizes Czech national research and educational network CESNET2[5]. The CESNET2 network provides DWDM interconnects among major cities in the Czech Republic, production 10 Gbps IP backbone for normal traffic as well as experimental services available to other projects. For traffic engineering of the IP backbone, it uses Multi-Protocol Label Switching (MPLS).

MetaCenter project has its nodes in three cities in the Czech Republic: Prague (Praha), Brno, and Pilsen (Plzeň), all of them located close to the CESNET2 point of presence. The distances (over optical cable) are approximately 300 km between Prague and Brno and 100 km between Prague and Pilsen.

L2 core network. The following technologies has been identified to fulfil the requirements of the VirtCloud L2 core network, that can be implemented using CESNET2 network [9]:

- *IEEE 802.1ad (QinQ)* is a technology that allows encapsulation of the 802.1q tagging into another 802.1q VLAN. It has been designed for service providers to encapsulate customer-provided VLAN tagging. The standard was approved in 2005 and it is currently the most widely supported and easiest to deploy manufacturer-independent technology.
- *Virtual private LAN service (VPLS)* [10], [11] is a viable technology for the network that use MPLS traffic engineering. It creates a shared Ethernet broadcast domain.
- *Cisco Xponder technology* [12] uses Cisco 15454 platform to create a distributed switch based on dedicated DWDM optical circuit interconnects. This is an interesting option for the networks that support lambda services, without the need of additional VLAN encapsulation.

Site network. Each site uses Layer 2 infrastructure implemented on mix of Force10, Hewlett-Packard, and Cisco

3. http://openvpn.net/
4. https://secure.logmein.com/products/hamachi/vpn.asp
5. Topology can be found at http://www.ces.net/network/.

Ethernet switches as shown in Figure 2. Each site has parallel uplinks to public IP routed network, Xponder network and VPLS network. For production purposes, the Xponder network is used under normal circumstances as it provides higher capacity since the traffic does not mix with normal routed traffic on the MPLS backbone.

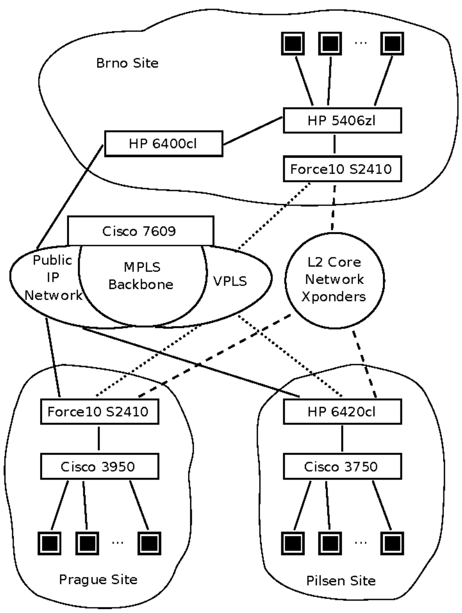

Figure 2. Site network setup.

When building a virtual cloud, a VLAN number is allocated and edge switches of each physical cloud are configured to send traffic of the VLAN through chosen tunnelling mechanism.

VLANs used for cluster communication must not interfere with VLANs used on a particular site for other purposes, therefore site local administrators have to provide a list of VLAN that may be used in the system. When allocating VLANs for clusters, only VLANs that are available on all sites participating in the virtual cluster can be used.

Host configuration. Hosts deploy Xen virtual machine monitor [13]. The hypervisor domain manages user domain virtual machines and provides network connection to them via an Ethernet bridge. Logical network interfaces of each user domain must be bridged to VLANs depending on membership of the user domain in virtual clusters. Taking into account that users may even have administrator privileges in their virtual machines, the tagging must necessarily be performed by the hypervisor, out of user's reach.

As shown in Figure 3, `eth0.vlan<n>` are virtual interfaces representing VLANs on the Dom0's `eth0` interface, `br<n>` are bridges that connect user domain traffic to VLAN interfaces.

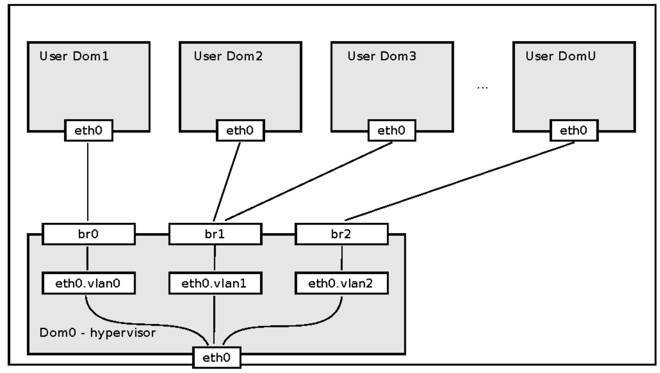

Figure 3. VirtCloud host configuration.

Addressing of the user domain interfaces can be either IPv4 or IPv6[6] and it can be fully controlled by the user. The user can use, e.g., private addresses and/or even addresses from user's organisation in order to publish the cluster machines.

VLAN life cycle implementation. VLAN allocation is controlled by a stateful service called SBF[7].

Users initiate building virtual clusters by means of submitting a special job to resource manager PBS[8]. The PBS allocates a set of physical nodes to run virtual cluster nodes and requests allocation of VLAN number from SBF. SBF configures active elements and returns a VLAN number. PBS in cooperation with Magrathea [6] configures bridging in Xen hypervisor domains and boots requested virtual machine images.

The configuration may be torn down by time-out, user and/or administrative action. Then the configuration is removed from all network elements and the VLAN number can be allocated to another virtual cluster.

Access from/to the Virtual Clusters. Currently we provide two services for the virtual clusters: file system access and user remote access. Both are implemented in similar way—NFSv4 file servers as well as OpenVPN server used for the remote access have access to all the VLANs of all the virtual clusters, thus becoming part of it. OpenVPN access implementation is very similar to what Nimbus system [2] uses for remote access.

4. First Experiences with VirtCloud

We have run a series of initial experiments in order to show feasibility of the whole concept: behaviour of the high-

6. While IPv6 is preferable because of possible merging of clusters, many applications (e.g., network file systems) don't support it reliably currently.

7. Easy-to-pronounce abbreviation for Slartibartfast, the Magrathean coastline designer from *The Hitchhiker's Guide to the Galaxy* by Douglas Adams.

8. http://www.openpbs.org/

performance virtualised network must not be significantly worse than the high-performance native routed IP network (which also represents the performance limit of application layer tunnelling solutions).

The system has two major network components, VLAN tagging in Xen itself and performance of the virtualised network in comparison to the routed one. We have tested tagging performance in a single site and compared virtualised and native network over the state-wide environment.

4.1. Experimental Setup

The machines we used for the experiments are located in three MetaCenter sites: Brno, Prague, and Pilsen. The topology of the network is described in Section 3.

In Brno, we used two identical machines `skirit82-1` and `skirit83-1`. Each of them has two dual-core Intel Xeon 5160 3GHz processors, 4 GB physical memory, and PCI Express gigabit network adapter Intel 80003ES2. The machines are interconnected with an HP 5406zl switch.

Prague node, `skurut9-1`, has two quad-core Intel Xeon X5365 3GHz processors, 16 GB physical memory, and PCI Express Gigabit Ethernet adapter Intel 80003ES2. Pilsen node, `konos23-1`, is a dual AMD Opteron 270 2GHz processor system with 8 GB physical memory, and PCI Gigabit Ethernet adapter Broadcom NetXtreme BCM5704.

All the machines run Xen version 3.1.3, hypervisor Linux kernel version is 2.6.22.17, user domains run 2.6.22.17, too, with the exception of `skurut9-1` having kernel 2.6.18. The distribution is SuSE Linux 10.0 on `skurut9-1` and Debian GNU/Linux 4.0 on the other machines. The hypervisor domains (Dom0) have 1 GB memory, user domains use the rest of available memory on a particular machine.

All the Xen tests were run among user domains. Processor planning was done by the Xen scheduler. Hypervisor domains had high priority (weight 256), user domains low priority (weight 1). In the standard configuration, a dynamic number of buffers is used in the implementation of virtual network interfaces between Dom0 and DomU. This turned out to be a performance bottleneck therefore we set the number of buffers to the maximum possible value (i.e., `/sys/class/net/<interface>/rxbuf_min` is set to the value of `rxbuf_max`).

In order to obtain comparison base not affected by virtualisation of the host machines themselves, we measured Xponders (a dedicated private network) using the same machines we described above without Xen virtualisation.

4.2. Measurement Software

Software tools used for measurement are

- iperf version 2.0.2 [14] with a set of patches by Andrew Gallatin originating in FreeBSD [15],
- Real-time UDP Data Emitter (RUDE) and Collector for RUDE (CRUDE) version 0.62 [16].

	Local network `skirit83-1`
Untagged	939 Mbit/s
VLAN tagging	936 Mbit/s

Table 1. TCP: price of VLAN tagging in Xen

We measured TCP throughput and UDP throughput for packet lengths 64 B, 100 B, 200 B, 300 B, ..., 1300 B, and 1400 B with iperf. Each result is an average of 60 1-second measurements taken continuously. As iperf sends UDP data (approximately) in the requested rate regardless of packet losses, we determine UDP throughput using a "first-fit convergence procedure."

The process goes as follows. Let us have the currently used bandwidth bw (the first measurement starts with the nominal bandwidth of the line, i.e., 1000 Mbit/s). We make a measurement in order to learn packet losses in this configuration, let the ratio of lost packets to the amount of sent packets be $loss$. If the $loss$ is at most 0.5% we take the measurement to be the final result and the process quits. If the $loss$ is higher than 0.005, we decrease the transmitted bandwidth according to formula

$$bw := \min\{bw(1 - 0.75loss), bw - 1\}$$

and go on repeating the measurement. The formula decreases the bandwidth at least by 1 Mbit/s to assure progress, and "less than to the number that came through" in order to make the measurement more precise.

We have also verified the iperf UDP throughput with a home-grown Real Time Protocol (RTP) benchmark called Generator7. The results were very similar to iperf's, we therefore omit them from the paper.

The rude/crude test is targeted primarily to the stability of the network. We send 1000 packets per second for 60 minutes and check whether all of them arrive and if they are in order.

4.3. Results and Discussion

The user domain test were run from `skirit82-1` (Brno) to the remaining machines, all of them using the native IP network and through a VLAN connected via Xponders and/or VPLS (the VLAN goes just through the local switch in case of `skirit83-1` machine). The comparison Xponder physical machine tests were also run from Brno to the remaining sites.

The results of all rude/crude tests via routed IP network, VPLS, and Xponders can be described easily—all packets in all configurations arrived in order, we therefore consider the network to be functional and stable.

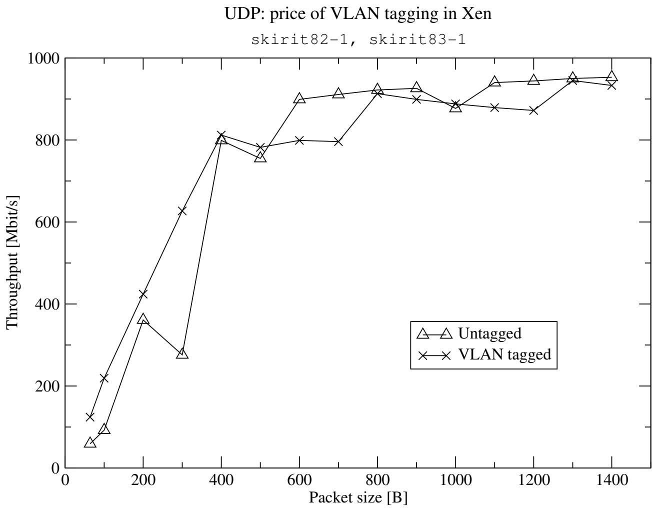

Figure 4. UDP: price of VLAN tagging in Xen

	Prague `skurut9-1`	Pilsen `konos23-1`
Xponders, phys.	936 Mbit/s	936 Mbit/s
Xponders, Xen	936 Mbit/s	936 Mbit/s
VPLS, Xen	935 Mbit/s	937 Mbit/s
Native IP, Xen	592 Mbit/s	362 Mbit/s

Table 2. TCP: Xponders, VPLS, and routed IP backbone

Let us study throughput of the network. The first test is concerned to the price of VLAN tagging in Xen bridge. Table 1 shows the TCP throughput between `skirit82-1` and `skirit83-1` for native untagged TCP traffic and with VLAN tagging on the Xen bridge. TCP traffic processing is not affected by VLAN tagging.

VLAN tagging of UDP traffic in Xen seems to bring a small overhead on the local network, as we can see on Figure 4.

Table 2 compares TCP throughput. The Xponders in physical machines represent the theoretically expected performance limit, being a dedicated network without any possible overhead caused by Xen. As we can see, Xen doesn't bring any overhead to TCP traffic. Moreover, VPLS, transported together with backbone commodity traffic, reaches the same performance as Xponder's dedicated network. In comparison, the throughput of the native routed IP is significantly worse—it is necessary to point out that the routes of the native connection are typically longer and more complex than of VPLS and Xponders.

Figures 5 and 6 show UDP performance from Brno to Prague and Pilsen nodes, respectively. Again, we take physical machines connected with the Xponder network as a base for our comparison. Virtualisation of the host machines brings acceptable overhead to the Xponder network. The measured performance of Xen virtualised hosts is slightly

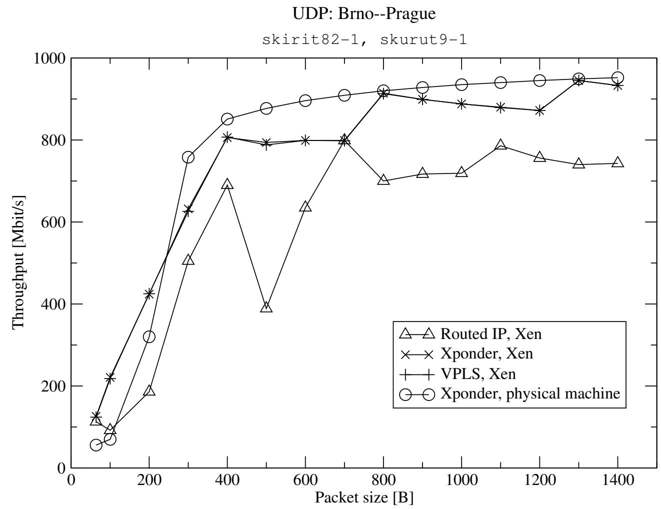

Figure 5. UDP: Xponders, VPLS, and routed IP to Prague

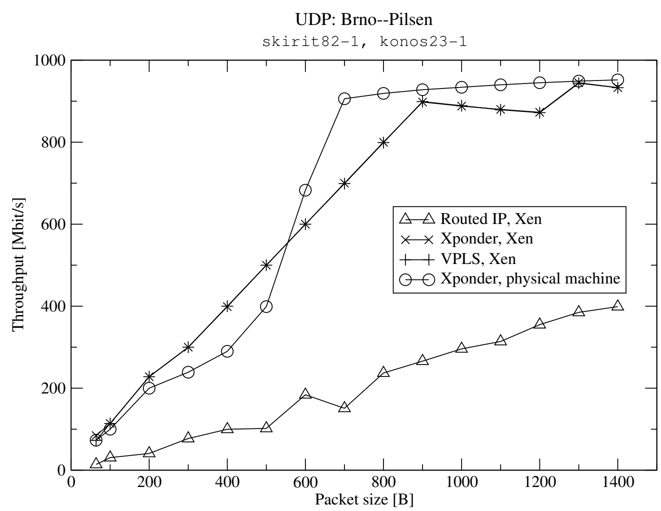

Figure 6. UDP: Xponders, VPLS, and routed IP to Pilsen

better than that of physical machines in case of small packets (up to 200 B for Prague and 500 B for Pilsen). This is most probably due to larger buffers available in the implementation of the virtual network interface.

Similarly to the TCP case, both Xponders and VPLS reach practically the same performance in Xen. The native routed network performance is clearly worse in case of Prague and significantly worse in case of Pilsen. For Pilsen, we attribute the result to rather complex IP routed network topology.

5. Related Work

Three mainstream approaches appear in the area of network virtualisation: Virtual Local Area Networks (illusion of local network over a more complex physical infrastructure), Virtual Private Networks (illusion of having a network interface in a distant network), and Overlay Networks (duplicating vertically part of network stack, usually in order to get traffic through an environment hostile in one way or another).

Previously described methods to building networks of virtual machines are based on assumptions about the available and requested network environment, mainly geographical distribution, restrictions placed in the network (Network Address Translation (NAT), firewalls), and isolation requirements.

Distributed networks are likely to be quite unfriendly for transporting usual internal cluster communication, therefore methods of tunnelling are necessary. In-VIGO [17], [18] uses a system of tunnels and VPNs to separate machines into logical clusters called VNET. VNET [4] is a software VLAN based on L2 tunnelling for clusters of virtual machines, building a logical Ethernet bridge over IP network. It uses the routed IP network for traffic tunnelling, therefore the performance of VNET cannot be better than performance of the IP network. Violin [5] is an overlay network based on UDP tunnels. Those methods are generally focused on traversing various types of NATs, firewall piercing, etc. It deploys a network of software routers and switches over the IP network (with performance implications similar to VNET, see our DC-1).

Building virtual cluster in unrestricted local network depends on the need of virtual cluster separation. Cluster-on-demand [1] separates virtual clusters on network level, addressing them with disjoint IP address spaces. Note that in case that users have administrator privileges in their virtual machines, it is easy for the users to intrude any virtual network in the site (cf. DC-3). Nimbus is a system for deployment and management of virtual machines (formerly known as Virtual Workspace Service) [2]. Nimbus supports configuring network interfaces of the virtual machines without creating closed or controlled network environment. Nakada et al. [3] describe a system for VLAN configuration for RedHat Linux based package system Rolls. Wide area network is not considered (DC-4).

Network performance of Xen virtual machine monitor [13] has been studies many times, e.g., [19], [20], [21], with results that are not easily comparable. The performance depends highly on many parameters like CPU allocation to domains, amount of memory, CPU scheduling, buffer sizes, etc.

6. Conclusions

We have presented VirtCloud, a system for internetworking dynamic virtual clusters over a large high performance network. The system is targeted for broadly distributed computing facilities, allowing to build virtual clusters (giving the users the possibility to fully manage their computation resources), encapsulate the clusters, and manage publishing and accessing the clusters in controlled manner.

Using our prototype implementation, we have tested feasibility of the concept and evaluated performance of VPLS and Xponder technologies used to build the core Layer 2 network.

Even though the approach turned out feasible and performing well, many questions left for deeper investigation remain. The methods of publishing encapsulated cluster must be studied thoroughly in order to provide more efficient ways to connect the cluster to user's machines. This problem is also related with scenarios of Layer 3 addressing the virtual clusters. Accessing external data and resources is another area for further research: while conceptually the problem is simple, it creates enormous amount of issues when implemented in the real Grid infrastructure.

Acknowledgements

We would like to thank Zdeněk Salvet, Václav Novák, Josef Verich, Pavel Šmrha, Miroslav Ruda, Jiří Denemark, Lukáš Hejtmánek. This project has been supported by a research intent "Optical Network of National Research and Its New Applications" (MŠM 6383917201) and "Parallel and Distributed Systems" (MŠM 0021622419).

References

[1] 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*, 2003, pp. 90–100, iSBN 0-7695-1965-2.

[2] K. Keahey, I. Foster, T. Freeman, and X. Zhang, "Virtual workspaces: Achieving quality of service and quality of life in the Grid," *Scientific Programming*, vol. 13, no. 4, pp. 265–275, October 2005, iSSN 1058-9244.

[3] 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.

[4] 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*, 2004, pp. 177–190.

[5] X. Jiang and D. Xu, "VIOLIN: Virtual Internetworking on Overlay Infrastructure," in *Proc. 2nd Int'l Symp. Parallel and Ditributed Processing and Applictions*, ser. LNCS, no. 3358. Springer-Verlag, 2004, pp. 937–946, iSSN 0302-9743.

[6] M. Ruda, J. Denemark, and L. Matyska, "Scheduling Virtual Grids: the Magrathea System," in *Second International Workshop on Virtualization Technology in Distributed Computing*. USA: ACM Digital Library, 2007, pp. 1–7.

[7] T. Freeman and K. Keahey, "Flying Low: Simple Leases with Workspace Pilot," in *Euro-Par 2008*, Las Palmas de Gran Canaria, Canary Island / Spain, August 2008.

[8] W. Luo, "Layer 2 Virtual Private Network (L2VPN) Extensions for Layer 2 Tunneling Protocol (L2TP)," RFC 4667.

[9] V. Novák, P. Šmrha, and J. Verich, "Deployment of CESNET2+ E2E Services," CESNET, z. s. p. o., Tech. Rep. 18/2007, December 2007.

[10] K. Kompella and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling," RFC 4761.

[11] M. Lasserre and V. Kompella, "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling," RFC 4762.

[12] Cisco Systems, Inc., "10 Gigabit Ethernet DWDM XPonder Card for the Cisco ONS 15454 MSTP," http://www.cisco.com/en/US/prod/collateral/optical/ps5724/ps2006/product_data_sheet0900aecd805ec093.html, 2008, accessed Sep 10, 2008.

[13] 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*. New York, NY, USA: ACM Press, 2003, pp. 164–177, ISBN 1-58113-757-5.

[14] "Iperf network performance measurement tool," http://dast.nlanr.net/Projects/Iperf/, 2005.

[15] A. Gallatin, "[PATCH] performance fixes for non-linux," Iperf users mailing list, http://lkml.org/lkml/2007/9/26/215, August 2007, accessed Sep 10, 2008.

[16] J. Laine, S. Saaristo, and R. Prior, "Real-time UDP Data Transmitter (RUDE)," http://rude.sourceforge.net/, 2002.

[17] 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*, vol. 21, no. 6, pp. 896–909, 2005.

[18] I. Krsul, A. Ganguly, J. Zhang, J. A. B. Fortes, and R. J. Figueiredo, "VMPlants: Providing and Managing Virtual Machine Execution Environments for Grid Computing," in *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*. Washington, DC, USA: IEEE Computer Society, 2004, p. 7, iSBN 0-7695-2153-3.

[19] P. Apparao, S. Makineni, and D. Newell, "Characterization of Network Processing Overheads in Xen," in *VTDC 2006: First International Workshop on Virtualization Technology in Distributed Computing (held in conjunction with SC06)*, Tampa, FL, USA, November 17 2006.

[20] L. Cherkasova and R. Gardner, "Measuring CPU Overhead for I/O Processing in the Xen Virtual Machine Monitor," in *USENIX 2005 Annual Technical Conference*, Anaheim, CA, USA, April 2005, pp. 387–390.

[21] A. Menon, J. R. Santos, Y. Turner, G. J. Janakiraman, and W. Zwaenepoel, "Diagnosing Performance Overheads in the Xen Virtual Machine Environment," in *VEE '05: Proceedings of the 1st ACM/USENIX international conference on Virtual execution environments*. New York, NY, USA: ACM Press, 2005, pp. 13–23, ISBN 1-59593-047-7.