Industrial Panel Talks

What Is the Impact of Cloud Computing on the Data Center Interconnect?

Co-chairs
Madeleine Glick, Intel
Ada Gavrilosvka, Georgia Tech

Speakers and Contact Information

Donn Lee, Facebook
Ashok Ganesan, Cisco
Anees Shaikh, IBM, “Cloud Networking Services for Enterprises”
Mark Shaw, Microsoft

Cloud Networking Services for Enterprises

(Invited Paper)

Anees Shaikh
IBM TJ Watson Research Center
Yorktown Heights, NY
Email: aashaikh@watson.ibm.com

***Abstract*—As more enterprises look to leverage the cost and flexibility advantages of cloud computing, the lack of rich networking support remains a challenge. In this paper, we discuss the requirements of enterprise line-of-business applications for additional network functions in the cloud, and argue that a service-level abstraction for the network is needed. After reviewing some of the trends in cloud network architecture, we describe two ongoing research efforts to develop networking services for multi-tenant enterprise clouds, and workload-optimized modular data centers.**

I. INTRODUCTION

The appeal of cloud computing as an IT consumption model for enterprises lies in its cost efficiency, as well as the ability to rapidly deploy new resources to handle changing business needs [12]. While adoption of cloud services among small and large enterprises continues to increase, there are a number of key challenges in moving line-of-business production workloads to the cloud. Some of these include the lack of end-to-end security, poor perceived reliability, compliance concerns, unpredictable performance, and lack of transparency and control over the service implementation [30], [37].

Underlying many of these challenges is the limited control available to customers to configure the network in current cloud environments. The cloud network model has largely focused on providing basic reachability using dynamic or static IP addresses assigned to customer VMs, with basic firewall capabilities available at each virtual server. Several key network functions are generally not available, e.g., fine-grained network isolation for security, policy-based routing through middleboxes (for intrusion detection or compliance), control over addressing, and optimizations like protocol acceleration, path control, and distributed caching for improved performance and availability. A richer set of network functions, coupled with an easy way to deploy and configure them, would help address several of the shortcomings in current cloud services (public and private).

Network services have started to receive more attention recently from cloud providers, but it is generally targeted at a small set of capabilities. For example, Amazon recently extended its VPN services to include secure connectivity to isolated virtual instances with the ability to segment them into subnets and specify private address ranges and more flexible network ACLs. Similarly, the Microsoft Windows Azure Virtual Network provides services for customers to integrate on-premise applications. There is also a developing ecosystem of third-party providers of network-related services delivered as virtual cloud appliances. Another model is via overlays using nodes in the cloud to provide services such as custom addressing and encrypted communications [9], [10].

In this short paper, we provide an overview of two ongoing efforts to develop cloud networking services for enterprise applications. Section III describes the design and implementation of a networking-as-a-service model for the cloud that unifies virtual server provisioning with new capabilities for simultaneously deploying virtual network services. We leverage emerging software-defined networking technologies and discuss some of the challenges in delivering network services for enterprise applications at cloud scale, as well as challenges that arise in handling cloud dynamics. In Section IV, we discuss the emergence of workload-optimized clouds that are tuned to specific enterprise applications. These depart from traditional modular data centers in eschewing the homogeneous, commodity infrastructure layer for one that uses specialized components to improve specific business outcomes. In particular, we discuss the implications of this data center model on multiple aspects of the network, including the interconnect topology and software. These ideas are in early stages and we are eager to engage the research community in further developing them.

II. CLOUD-SCALE NETWORK INTERCONNECT ARCHITECTURES

Although our focus in this paper is on cloud networking from a services perspective, it is important to also consider the requirements on the underlying network architecture to realize such services. In this section, we briefly review the characteristics of cloud data center networks that are designed for i) cloud-scale with tens of thousands of physical hosts (and ports), and an order of magnitude more virtual servers, ii) limited bandwidth oversubscription to support inter-VM communications patterns, and iii) support for virtualization and VM mobility (see [33] for a good overview).

As Figure 1 illustrates, several trends are driving the move away from the traditional 3-tier design to support cloud networking requirements:

- *flattening* of the network from a 3-tier tree-based structure with oversubscription ratios as high as 1:20 to a "spine-leaf" topology (combined with multi-path routing) to enable greater bisection bandwidth across the data center.
- *simplified management* models that treat the switches as a single large switch to apply consistent configuration and policies automatically across the network. The

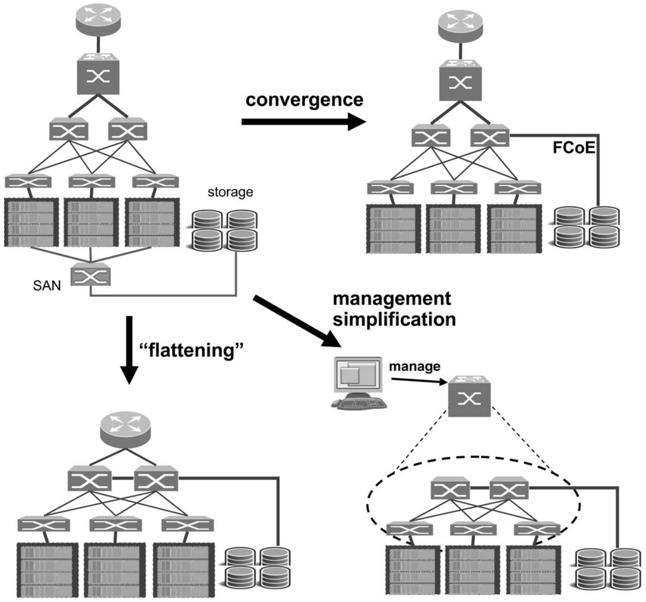

Fig. 1. Large scale data center network technology trends

control plane can also run as a single instance rather than separately in each switch to further the single large switch abstraction. This can be implemented along with the management functions, or in a separate centralized controller (e.g., in OpenFlow [21]).

- *convergence* of multiple traffic types onto a single Ethernet fabric. In particular the Fiber Channel over Ethernet (FCoE) and Data Center Bridging (DCB) standards aim to enable Ethernet to carry storage-area network traffic without loss using mechanisms for bandwidth allocation, congestion avoidance, and prioritized flow control. Although SANs are used primarily in enterprise settings, the move toward converged fabrics, while still in early stages, has goals similar to cloud networks, namely simplifying the network and reducing costs.

As cloud computing adoption continues to grow, commercial network equipment makers have introduced new "fabric" architectures and products that support these trends (e.g., [8], [29], [34]). A number of new architecture and topology proposals have also come from the research community, often adapting interconnect topologies from high-performance computing systems to achieve high bisection bandwidth using low-cost merchant silicon-based switches (e.g., [13], [15], [16], [25]). Many of these proposals support simplified management and support for VM mobility by creating a single large layer-2 domain.

New standards are also emerging to extend the cloud network into host hypervisors. Edge virtual bridging (IEEE 802.1qbg) provides a mode in which all packet switching, even between guests on the same host, is done through the attached physical switch. This avoids having to separately manage port profiles (e.g., VLAN ids, ACLs, etc.) in both hypervisor virtual switches and the physical switches. Instead, VM port profiles are made visible to the physical network and associated management tools.

Finally, overlay networks are also being used to implement network virtualization (usually on a private cloud scale). These schemes avoid dependency on the underlying physical network and use various forms of layer-2 and layer-3 encapsulation to implement end-to-end isolated virtual networks at each hypervisor virtual switch, with centralized management (e.g., [7], [27], [35]).

III. REALIZING CLOUD NETWORK SERVICES

Our realization of cloud networking aims to provide a services-level abstraction for customers to deploy and manage network functions in coordination with other virtualized infrastructure (e.g., virtual servers and storage) comprising their applications. In addition to standard network connectivity between virtual instances, we also want to allow customers to specify a number of network functions such as isolated domains, custom addressing, quality-of-service, and middlebox traversal.

While some of these functions are available as third-party add-ons via virtual appliances or overlays, using them requires customers to integrate a variety of point solutions each with their own varied interfaces and service models. In contrast, we unify network services in a single, extensible framework that is implemented within the cloud infrastructure. Hence, our network primitives are highly efficient and transparent to cloud customers and end-users.

Our design leverages a number of techniques, including software-defined networking to provide flexible and fine-grained control of the network, indirection to provide added control over addressing, and host-based virtual switches to extend the network edge into hypervisors. We integrate these mechanisms with the cloud provisioning system to extend VM provisioning to include network primitives. In this way, the desired network functions are integrated with the application, making it easy for the tenant to tailor the network support for the needs of the workload. We leverage several of the trends described in Section II in our design, such as management simplification through centralized control, and overlay networks for virtualization.

Our approach differs from others that aim to provide control over the network through a virtualized device view, for example a router or switch abstraction for each customer [7], [19]. We view a services abstraction as a more natural and less complex way for enterprise customers to reason about and express their applications' network requirements. A more closely related effort to add network service APIs to a cloud management system is in OpenStack [2], though it is still nascent and does not specify how network services should be realized.

Below we discuss our initial design and implementation, and also outline our ongoing work to address challenges of scalability, performance, and management. Additional details may be found in [5].

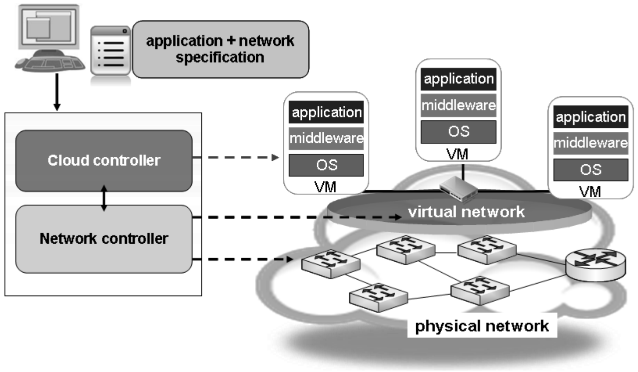

Fig. 2. Cloud networking service architecture

A. Architecture and implementation overview

The cloud network services architecture has three primary components, shown in Figure 2: i) customer application and network specification, ii) cloud controller, and iii) network controller.

Customers specify their application deployment, and corresponding network service configuration, using a simple **network policy specification** language. The basic abstraction is that of a virtual network segment that establishes connectivity between VMs. Various network services may be attached to a virtual network segment to define its behavior, for example minimum bandwidth, layer-2 isolation, or a list of middleboxes to be traversed. Traffic is only allowed to reach a VM over an explicitly defined virtual segment, hence providing a secure "default-off" model. Virtual network segments may be applied between all pairs of VMs in a predefined group, or between any pair of VMs across groups. This approach can be used to provide standard templates of network services and segments that implement pre-defined policies (e.g., for security). We have demonstrated a number of scenarios using these constructs, including complex multi-tier, clustered applications with a number of network QoS, middlebox, and isolation services.

The **cloud controller** manages physical resources in the cloud, and provides the standard functions of deploying virtual machines with the specified resources (CPU, memory, storage) and virtual image software stack. The server provisioning component also handles initial placement and subsequent VM operations such as monitoring. In our system, the cloud controller is extended to: i) interpret network policy specifications and generate a communication matrix for the network controller, ii) consult the network provisioning component to determine which physical hosts should be used for placing new VMs, and iii) tracking which hypervisor virtual switch each VM is attached to, and notifying the network controller when changes occur. The cloud controller is implemented with the OpenNebula 1.4 cloud management system, with modifications to support network service management.

The **network controller** configures virtual networks in the cloud by mapping the logical connectivity requirements in the communication matrix onto physical network resources. It also determines VM placement and performs re-mapping when available resources change, to ensure that customer requirements continue to be met when changes occur. The network controller gets information about the requested virtual network from the cloud controller, along with a list of physical hosts and their available resources. In addition, the network controller periodically collects the current status of switches and links (along with link utilizations) and the current mapping of virtual network flows to the corresponding physical cloud network devices. Based on these inputs, the controller invokes its placement optimizer to determine the best location to place VMs within the cloud (and reports it to the cloud controller for provisioning). The controller then provisions the network devices (including hypervisor virtual switches) to instantiate the customer's virtual network services. Details of the network controller algorithms for placement, path computation, and device management are available in [5].

As mentioned earlier, we adopt the software-defined networking paradigm, and more specifically the OpenFlow protocol, to enable the cloud network controller to manage network device configurations in the cloud. OpenFlow provides an API that allows an external software controller to manage the forwarding tables in network switches by dynamically installing, updating, and deleting flow entries in switch flow tables. Flow entries support specific actions on packets, such as forwarding to a particular output port, dropping the packet, or forwarding to the controller for further actions [21]. Our network controller is implemented as an application running on the NOX OpenFlow controller [14].

In our prototype and demonstrations, the cloud network consists of a laboratory testbed with commercial OpenFlow-enabled switches supporting OpenFlow v1.0. We replace the default KVM virtual switch with Open vSwitch in each hypervisor as it also supports OpenFlow. While our current design and implementation assumes OpenFlow switches deployed in the cloud, we are also pursuing designs that can provide some subset of network services with a more limited deployment. Given the widespread participation in the standardization efforts for software-defined networking across the networking industry [1], we expect to see more network switch vendors supporting OpenFlow, often through a simple firmware or software update.

B. Scaling Network Services for the Cloud

Scaling network services to cloud scale poses a number of important challenges for the network and management layers. For example, physical network devices typically are constrained in terms of their processing and memory resources. Moreover, in large scale clouds where cost efficiency is paramount (see Section IV-A for more on this), the common approach is to use commodity switches and merchant silicon which may be further limited. Another practical challenge is handling the dynamics of the cloud, where applications (and their corresponding virtual networks) are frequently deployed, modified, torn down, or scaled up and down. Each new request or change requires the network controller to re-establish (perhaps incrementally) the virtual network segments and their associated services. Finally, the cloud infrastructure architects and operations teams must account for network switch or link failures. The impact of even a single link failure is much more magnified in the cloud as it potentially affects a large number of virtual networks and cloud applications. Rapid recovery of the virtual networks is crucial for cloud networking services to be viable for enterprise use.

In implementing and evaluating our cloud networking prototype we have addressed a number of these challenges to varying degrees. A number of opportunities for further research and improvement remain, however, as described below.

Network switch limitations: OpenFlow-enabled switches offer an elegant and appealing model for fine-grained control of network traffic. But creating virtual networks for customers on cloud scale requires a large amount of forwarding state to be maintained in each switch. Forwarding entries are stored in ternary content-addressable memories (TCAMs) which are limited in size (e.g., a few thousand entries) – unless this memory is carefully managed, it may not be possible to support a large number of virtual networks in the cloud. We use several techniques to mitigate this limitation, for example storing per-flow forwarding rules in hypervisor virtual switches where memory is relatively plentiful, and aggregating certain flows along a small number of paths using standard destination-based forwarding rules. Also, by assigning addresses judiciously to VMs behind a shared top-of-rack switch, we can similarly aggregate forwarding entries with the same output port, for example. More work is needed, however to extend these types of optimizations more broadly to raise the ceiling on the number of virtual network services that can be supported.

Cloud dynamics: Network services must be able to operate with minimal disruption in the face of planned or unplanned events. Network or VM deployment changes, switch failures, or link failures are examples of cloud dynamics that require the network controller to quickly re-establish the affected virtual networks. In our prototype, for example, we employ classic methods such as precomputation and caching to reduce the time it takes to recover from failures. The network controller keeps track of which virtual networks are dependent on a particular physical switch or link. When a link or switch fails, the controller can quickly determine which virtual network segments need to be recomputed. To speed up this process, it also precomputes the state needed to re-establish virtual networks for certain high-impact failures, such as an aggregation layer switch or uplink.

Additional services: While our network services model provides a number of important network functions for cloud applications, additional services are required to move the architecture closer to an enterprise-grade service. One key requirement for many enterprises is to be able to monitor and manage their virtual networks in the cloud. For example, we can extend the network service specification framework to allow customers to attach monitoring, reporting, and logging functions to virtual network segments, along with policies for sampling and storing the data. The challenges here include the need to ensure privacy of the network data, and to limit the overhead of collecting and processing management data at cloud scale. Another example is the ability for customers to deploy WAN optimization or application delivery controllers (ADC). While several IaaS providers support load-balancing as core service, a more complete ADC function generally requires custom arrangements with the provider, or integration of a third-party offering. A more appealing model would be to include wide-area functions in the same framework as the virtual network services.

IV. FLEXIBLE NETWORKS FOR WORKLOAD-OPTIMIZED DATA CENTERS

Scale-out systems are often deployed as integrated, modular units that are preconfigured blocks of compute, storage, and connectivity. The market for modular systems (e.g., high-density multi-rack or blade-based) is well established. The latest step in this evolution toward highly integrated solutions is the growing interest in modules that comprise an entire data center, for example prebuilt and delivered in a standard shipping container. These self-contained data centers combine servers, storage, and networking infrastructure, with external connections for power, cooling, and networking [6], [18]. Since these integrated data centers can be built using standardized (often commodity) components and construction methods, they offer cost advantages over traditional data centers. For cloud service providers, these data centers offer the benefit of very rapid, incremental deployment and highly optimized power and cooling in addition to the aforementioned reduction in capital costs. The operational management cost is also reduced through a "fail in place" model in which failed components are not serviced, but rather dealt with through redundancy in the infrastructure as well as the software applications [18].

A. *Warehouse-scale vs. Workload-optimized*

Modular, or container-based, data centers have been widely adopted by global-scale online service providers, such as Google [23], Microsoft Online [22], Facebook [26], and Yahoo! [32]. When deployed in large scale (e.g., tens of thousands of servers) these data centers have been described as "warehouse-scale computers" with homogeneous infrastructure, system software, and management [4]. These data centers are designed to run a relatively small number of Internet-scale services like search and mail, with cost efficiency (including power efficiency in particular) as a primary driver. More recently, cloud service providers have also turned to modular data centers to enable rapid growth [24]. But unlike online service providers that deliver a few very large applications, cloud providers offer infrastructure-as-a-service to a very large number of tenants running a wide variety of applications.

Warehouse-scale computers and large multi-tenant cloud providers rely on commodity infrastructure and homogeneity to control costs and provide reasonable performance for a variety of workloads. For individual enterprises looking for optimal performance for their specific applications, however, the general-purpose approach taken in these environments may not be a good choice. Recent work, for example, has shown that even small differences in microarchitecture can provide an opportunity for significant performance improvement for scale-out applications [20]. Modeling approaches have also been applied to help automate the selection of cluster configurations (e.g., type of storage, CPU, etc.) in order to meet a target SLA for a given workload [31].

A new class of systems, sometimes referred to as "workload-optimized" or "outcome centric" systems integrate a tailored set of compute, storage, and network infrastructure, with specialized middleware or other software to provide a high-performance platform for specific workloads [17]. These tightly integrated systems are distinguished from general-purpose modular systems by their combination of specific infrastructure components and software to support different workloads such as online transaction processing (OLTP), data analytics, high-performance computing, or web application serving. Also, workload-optimized systems are designed for existing enterprise applications, where there is limited willingness to completely redesign the software to work with commodity, homogeneous components that are more prone to failures or offer relatively poor performance. In addition to the performance advantages of specialized infrastructure, enterprises also benefit from the integration of workload-specific platform software to enable a faster time-to-value to achieve desired business-level outcomes. Use of these workload-optimized systems among enterprises is expected to grow in the near future [3].

As the number of application instances deployed on workload optimized systems grows, we expect to see the emergence of workload-optimized environments that enable the agility and elasticity of clouds. In contrast to traditional clouds built from commodity general purpose components, these "workload-optimized clouds" consist of specialized architectures designed to run particular workloads well. The interface and usage model, however, has the same characteristics as a private cloud, and could be packaged and delivered similarly to the modular or container-based data centers described above.

B. Flexible architectures for cost-effective workload-optimization

From a IT provider point of view, there is a strong desire to offer a workload-optimized cloud in the same rapid, low-cost fashion as general purpose modular data centers. The challenge is how to create a versatile design process that can support workload-optimized environments for a variety of use cases, without requiring completely redesigned infrastructure, system, and middleware layers. Typically, workload optimization involves changes to the system architectures, I/O components, management software, and middleware features (see Table I for some examples).

Mixing components to service different workloads is not well-aligned to the standard modular data center design methodology which is geared toward homogeneous infrastructure and lacks integration with higher layers of the stack (e.g., middleware). New "flexible computing" models are emerging, however, that aim to address this requirement with a systems architecture that enables computing elements to be interchanged easily. The intent of these designs is to allow, for example, commodity x86 server elements to be exchanged easily with GPU-, FPGA-, or microserver-based compute nodes. The key is that other elements of the infrastructure, such as the storage volumes, network fabric, or software, can remain unchanged and reused, or also reconfigured for a new workload. To reduce the cost of optimizing for new workloads, the ability to reuse other IT components is crucial. Realizing this vision requires that IT components be designed to support this level of flexibility.

Component	Optimization feature	Workload / application
system architecture	high memory expansion	consolidation via virtualization
system architecture	FPGA-based compute engines	data analytics
I/O	low-latency adapters with hardware assist	high-frequency trading
network topology	low (or no) bandwidth oversubscription	business or data analytics
middleware	MapReduce / Hadoop autoconfiguration	large-scale analytics

TABLE I
EXAMPLES OF WORKLOAD OPTIMIZATION FEATURES

C. Flexible networks for workload-optimized clouds

The design of a modular approach for workload-optimized clouds requires new thinking at multiple layers, including power, cooling, packaging, platform architecture, storage and network design, systems software to manage the infrastructure layers, and middleware to provide key services to the target workload. In addition, understanding workload characteristics across various industries is important to provide a solution that can in fact be tailored for a broad set of use cases. Some workload attributes that will govern the cloud design include:

- scaling characteristics – *does the workload grow by adding more threads, cores, or nodes?*
- resource bottlenecks – *which system resources are likely to limit performance or scale?*
- cost sensitivity – *is the target industry or workload more sensitive to cost or other factors such as performance or availability?*
- availability and recoverability – *does the workload typically have strict disaster recovery or availability requirements?*

These and other questions impact the choice of individual architectural components (server platform, NICs, SSD storage, etc.) as well as system design at the data center level (level of redundancy, backup strategy, use of commodity vs. high-end components, etc.).

In the remainder of this section we focus on some of the important network features that we aim to make available in the flexible workload-optimized data center. The network infrastructure and communications software play an important role in application performance. Hence, any workload optimization effort must pay close attention to the network at multiple levels.

Network topology: The traditional 3-tier network topology still deployed in most enterprise data centers has well-known scaling limitations due to uplink oversubscription, underutilization of multiple paths through the network, and limited memory in switches to support a very large-scale flat addressing space (thus requiring division into subnets). New network architectures for data centers that address these shortcomings are largely motivated by the need to support full bisection bandwidth across the data center to enable any-to-any communications between VMs or hosts [13], [15], [16], [25]. For online service providers with large cluster-based applications (e.g., search, social networks with multi-server interactions), or multi-tenant clouds with unpredictable communications patterns (e.g., Amazon EC2 or Microsoft Azure), optimizing for cross data center ("east-west") traffic may be appropriate. However, for OLTP or web applications, communication may be more skewed toward client-server patterns. For future workloads, such as online game streaming, or virtual desktops, the pattern will consist of many "north-south" flows, each with significant demands on both throughput and latency. As one might expect, some recent work has demonstrated that the choice of network architecture does indeed play a role in application performance [38]. Hence, a "tunable" network architecture that can be adapted to different needs is a core requirement for workload-optimized clouds. A number of recently proposed techniques can be leveraged to provide mechanisms for tuning the topology without extensive reconfiguration or rewiring, for example hybrid electrical-optical architectures [36], programmable network devices [21], and offloading forwarding and routing to servers [15], [16].

Communications middleware and services: Above the network infrastructure layer, there is an opportunity to provide specialized networking support for specific workloads as middleware or services. We can observe examples of such services even in general purpose cloud platforms, including cluster load-balancing and content distribution. Workloads that scale by growing the number of nodes, or workloads that deliver multimedia content to end users, can benefit from these services, respectively. Another important service that could benefit a number of workload types is network QoS to reserve bandwidth or control network variability between particular elements of an application. More recently, commercial offerings provide a tuned middleware layer that is designed to take advantage of specialized high-performance interconnects such as Infiniband [11], [28]. Similar protocol optimizations may be integrated into middleware or libraries to facilitate compression or caching for workloads that provide mobile services, for example.

Network management: One of the primary value propositions of the workload-optimized cloud is that it should deliver superior application performance for target workloads. Hence, the management layer must provide the operator with sufficient visibility and control to ensure the environment is delivering the expected business outcomes. For the network, this requires full visibility into both the physical and virtual network devices, as well as a way to translate performance, availability, or security goals into relevant network configurations and policies, and then be able to report on how those goals are being met. In other words, the traditional network performance metrics of bandwidth utilization, TCP goodput, packet loss, etc. are less important than their effect on workload-level metrics. The required translations are likely be workload-specific, however, and represent a significant challenge for the management system.

V. SUMMARY

In this paper we argued for the need for richer networking features in the cloud to enable more enterprises to move line-of-business applications to the cloud and have similar networking capability as in their on-premise environments. We view a services level, rather than device level, abstraction as the right one for enterprises to deploy network services to better meet the need of their applications. While some capabilities are available in commercial cloud platforms, a comprehensive solution requires customers to integrate a number of third-party appliances and services, or confine themselves to specific cloud environments like VPN-connected private clouds. We presented an overview of our design and implementation of a unified cloud networking service framework for a multi-tenant cloud, with network functions that can support needs of a variety of applications. Our design leverages the flexibility and control of software-defined networks, and incorporates a number of techniques to address challenges arising from cloud scale and dynamics. We also described our exploratory ideas on flexible "workload optimized clouds." The vision is to design a modular cloud data center that enables the interchange of specific workload-specialized components and sharing of others that do not require optimization for the target workload. We focused in particular on how the network infrastructure and software could be designed in a tunable fashion to support specific workloads without requiring a costly wholesale redesign or reconfiguration.

Acknowledgment

The cloud networking services platform is an ongoing effort in collaboration with Prof. Aditya Akella and Theophilus Benson at the University of Wisconsin, and Sambit Sahu at IBM Research. The concepts of network architectures and services for workload-optimized clouds benefited from extensive discussions with Jack Kouloheris and Chung-Sheng Li, my colleagues at the IBM TJ Watson Research Center.

REFERENCES

[1] Open networking foundation. http://www.opennetworkingfoundation.org/.
[2] Openstack cloud software. http://wiki.openstack.org.
[3] M. Bailey et al. Worldwide enterprise server 2011 top 10 predictions: An outlook. Technical Report 226698, IDC, February 2011.

[4] L. A. Barroso and U. Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Synthesis Lectures on Computer Architecture. Morgan and Claypool Publishers, 2009.

[5] T. Benson, A. Akella, S. Sahu, and A. Shaikh. EPIC: Platform-as-a-service model for cloud networking. Technical Report 1686, University of Wisconsin, Dept. of Computer Science, February 2011.

[6] D. Cappuccio. Containers and modules: Is this the future of the data center? Technical Report G00211139, Gartner, April 2011.

[7] M. Casado, T. Koponen, R. Ramanathan, and S. Shenker. Virtualizing the network forwarding plane. In *Proc. of Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO)*, 2010.

[8] Unified access layer with Cisco Nexus 5000 series switches and Cisco Nexus 2000 series fabric extenders. white paper, March 2010. http://www.cisco.com/en/US/prod/collateral/switches/ps9441/ps10110/solution_overview_c22-588237.pdf.

[9] CloudSwitch Enterprise. http://www.cloudswitch.com/.

[10] VPN-Cubed. http://cohesiveft.com/vpncubed/.

[11] Oracle exalogic elastic cloud: A brief introduction. white paper, March 2011. http://www.oracle.com/us/products/middleware/exalogic-wp-173449.pdf.

[12] F. Gens. Idc it cloud services survey, 2q10. Technical Report 225021, IDC, September 2010.

[13] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. VL2: A scalable and flexible data center network. In *Proc. of ACM SIGCOMM*, 2009.

[14] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: towards an operating system for networks. *ACM Computer Communication Review*, 38(3), July 2008.

[15] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu. BCube: A high performance, server-centric network architecture for modular data centers. In *Proc. of ACM SIGCOMM*, 2009.

[16] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu. Dcell: a scalable and fault-tolerant network structure for data centers. In *Proc. of ACM SIGCOMM*, 2008.

[17] G. Haff. Horses for courses: Optimizing for workloads. Technical Report DIL03009USEN, Illuminata, October 2009.

[18] J. Hamilton. An architecture for modular data centers. January 2007.

[19] E. Keller and J. Rexford. The "platform as a service" model for networking. 2010.

[20] J. Mars, L. Tang, and R. Hundt. Heterogeneity in "homogeneous" warehouse-scale computers: A performance opportunity. *IEEE Computer Architecture Letters*, 2011.

[21] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling innovation in campus networks. *ACM Computer Communication Review*, 38(2), March 2008.

[22] R. Miller. Microsoft goes all-in on container data centers. Data Center Knowledge, December 2008. http://www.datacenterknowledge.com/archives/2008/12/02/microsoft-goes-all-in-on-container-data-centers/.

[23] R. Miller. Google unveils its container data center. Data Center Knowledge, April 2009. http://www.datacenterknowledge.com/archives/2009/04/01/google-unveils-its-container-data-center/.

[24] R. Miller. Amazon's cloud goes modular in oregon. Data Center Knowledge, March 2011. http://www.datacenterknowledge.com/archives/2011/03/28/amazons-cloud-goes-modular-in-oregon/.

[25] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. PortLand: A scalable fault-tolerant layer 2 data center network fabric. In *Proc. of ACM SIGCOMM*, 2009.

[26] Open compute project. http://opencompute.org/.

[27] Open vSwitch: An open virtual switch. http://openvswitch.org/.

[28] IBM pureScale application system. white paper, August 2010. http://www.ibm.com/software/data/purescale/.

[29] The Juniper networks QFabric architecture: a revolution in data center network design. white paper, March 2011. http://www.juniper.net/us/en/local/pdf/whitepapers/2000380-en.pdf.

[30] D. M. Smith, D. W. Cearley, Y. V. Natis, B. J. Lheureux, D. C. Plummer, M. W. Cain, and D. Scott. Predicts 2011: Cloud computing is still at the peak of inflated expectations. Technical Report G00209194, Gartner, November 2010.

[31] A. C. Snoeren, H. Madhyastha, J. McCullough, R. Kapoor, G. Porter, S. Savage, and A. Vahdat. BICMIC: Matching cluster configurations to application demands. In *Workshop on Content Delivery and Services for Mobile and Fixed Users (CloudS)*, 2010.

[32] Y. Sverdlik. Yahoo's data center coop. Datacenter Dynamics, November 2010. http://www.datacenterdynamics.com/focus/archive/2010/10/yahoo!s-data-center-coop.

[33] A. Vahdat, M. Al-Fares, N. Farrington, R. Mysore, G. Porter, and S. Radhakrishnan. Scale-out networking in the data center. *IEEE Micro*, 30(4):29–41, 2010.

[34] Deploying Brocade VDX 6720 data center switches in enterprise data centers. white paper, 2011. http://www.brocade.com/solutions-technology/technology/vcs-technology/index.page.

[35] Vmware vCloud director networking, January 2011. http://kb.vmware.com/kb/1026298.

[36] G. Wang, D. G. Andersen, M. Kaminsky, K. Papagiannaki, T. E. Ng, M. Kozuch, and M. Ryan. c-Through: part-time optics in data centers. In *Proc. of ACM SIGCOMM*, 2010.

[37] G. Wang and T. Ng. The impact of virtualization on network performance of Amazon EC2 data center. In *Proc. IEEE INFOCOM*, March 2010.

[38] Y. Zhang, A.-J. Su, and G. Jiang. Understanding data center network architectures in virtualized environments: A view from multi-tier applications. *Computer Networks*, 55(9):2196–2208, 2011.