

Trends and Challenges in Cloud Datacenters

Kashif Bilal, Saif Ur Rehman Malik, and Samee U. Khan,
North Dakota State University

Albert Y. Zomaya, University of Sydney

Next-generation datacenters (DCs) built on virtualization technologies are pivotal to the effective implementation of the cloud computing paradigm. To deliver the necessary services and quality of service, cloud DCs face major reliability and robustness challenges.

Cloud computing is the next major paradigm shift in information and communication technology (ICT). Today, contemporary society relies more than ever on the Internet and cloud computing. According to a Gartner report published in January 2013, overall public cloud services are anticipated to grow by 18.4 percent in 2014 into a \$155 billion market.¹ Moreover, the total market is expected to grow from \$93 billion in 2011 to \$210 billion in 2016. Figure 1 depicts the public cloud service market size along with the growth rates. We've seen cloud computing adopted and used in almost every domain of human life, such as business, research, scientific applications, healthcare, and e-commerce² (see Figure 2).

The advent and rapid adoption of the cloud paradigm has brought about numerous challenges to the research community and cloud providers, however. Datacenters (DCs) constitute the structural and operational foundations of cloud computing platforms.² Yet, the legacy DC architectures cannot accommodate the cloud's increasing adoption rate and growing resource demands. Scalability, high cross-section bandwidth, quality of service (QoS) concerns, energy efficiency, and service-level agreement (SLA) assurance are

some of the major challenges faced by today's cloud DC architectures. Multiple tenants with diverse resource and QoS requirements often share the same physical infrastructure offered by a single cloud provider.³ The virtualization of server, network, and storage resources adds further challenges to controlling and managing DC infrastructures.² Similarly, cloud providers must guarantee reliability and robustness in the event of workload perturbations, hardware failures, and intentional (or malicious) attacks³ and ultimately deliver the anticipated services and QoS.

The cloud computing paradigm promises reliable services delivered through next-generation DCs built on virtualization technologies. This article highlights some of the major challenges faced by cloud DCs and describes viable solutions. Specifically, we focus on architectural challenges, reliability and robustness, energy efficiency, thermal awareness, and virtualization and software-defined DCs.

Cloud Datacenter Architectures

The DC architecture plays a pivotal role in the performance and scalability of the cloud platform. Cloud computing relies on DCs to deliver the expected services.² The widespread adoption of the cloud paradigm mandates exponential growth in the DC's computational, network, and storage resources. Increasing the computational capacity of today's DCs is not an issue. However, interconnecting the computational resources to deliver high intercommunication bandwidth and specified QoS are key challenges. Today's DCs are not constrained by computational power but are limited by their interconnection networks.

Legacy, multirooted tree-based network architectures, such as the ThreeTier architecture, cannot accommodate cloud computing's growing demands.⁴ Legacy DC architectures face several major challenges: scalability, high oversubscription ratio and low cross-section bandwidth, energy efficiency, and fault tolerance.

To overcome these challenges, researchers have proposed various new DC architectures, such as FatTree, DCell, FiConn, Scafida, and JellyFish.² However, these proposed DC architectures overcome only a fraction of the challenges faced by legacy DC architectures. For instance, the FatTree architecture delivers high bisection bandwidth and a 1:1

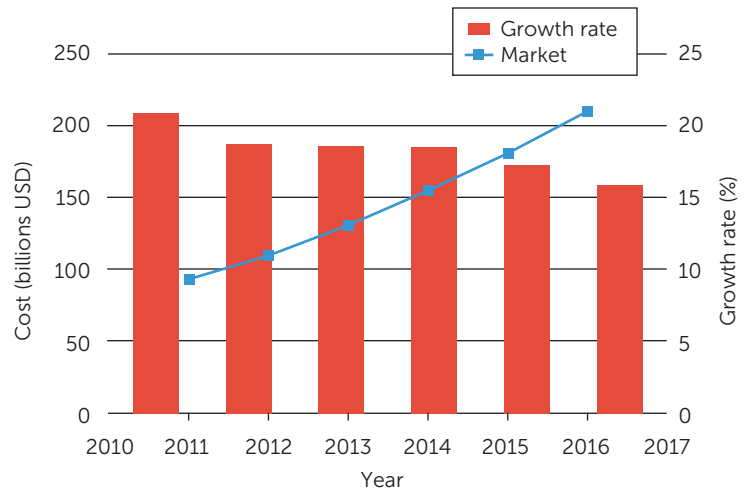


FIGURE 1. Market and growth rate of public clouds. The market is expected to reach more than \$200 billion by 2017.

oversubscription ratio, but it lacks scalability. The DCell, FiConn, Scafida, and Jellyfish architectures, on the other hand, deliver high scalability but at the cost of low performance and high packet delays with high network loads.

Because of the huge number of interconnected servers in a DC, scalability is a major issue. Tree-structured DC architectures, such as ThreeTier, VL2, and FatTree, offer low scalability. Such DC architectures are capped by the number of network switch ports. Server-centric architectures, such as DCell and FiConn, and freely/randomly connected architectures, such as JellyFish and Scafida, offer high scalability.²

DCell is a server-centric DC architecture, in which the servers act as packet-forwarding devices in addition to performing computational tasks.⁴ DCell is a recursively built DC architecture consisting of a hierarchy of cells called dcells. The dc_{0} is the building block of the DCell topology, which contains n servers connected through a network switch. Multiple lower-level dcells constitute a higher-level dcell—for instance, $n + 1$ dc_{0} builds a dc_{1} . A four-level DCell network with six servers in dc_{0} can interconnect approximately 3.26 million servers. However, such DC architectures can't deliver the required performance and cross-section bandwidth within a DC.⁴



FIGURE 2. Adoption of cloud computing in the information and communications technology (ICT) sector. In 2014, the amount spent on clouds is expected to reach \$55 billion annually.

Similarly, JellyFish and Scafida are nonsymmetric DC architectures that randomly connect servers to switches for high scalability. In the JellyFish architecture, the servers are connected randomly to switches such that a network switch can be connected to n servers. Each network switch is then connected to k other switches. The Scafida DC architecture has a scale-free network architecture. The servers are connected to switches using the Barabasi and Albert network-generation algorithm. Because of the large number of nodes within the network, DC architectures can't use conventional routing algorithms. The customized routing algorithms that DC architectures use, such as DCell Routing, perform poorly under high network loads and many-to-many traffic patterns.

In a previous study,⁴ we analyzed the network performance of state-of-the-art DC architectures with various configurations and traffic patterns.⁴ Our analysis revealed that server-centric architectures, such as DCell, suffer from high network delays and low network throughput compared with

tree-structured switch-centric DC architectures, such as FatTree and ThreeTier. Figure 3 shows that DCell experiences higher network delays and low throughput as the number of nodes within the DC architecture increases.⁴ This is because, for larger topologies, all the inter-dcell network traffic must pass through the network link that connects the dcells at the same level, resulting in increased network congestion. However, for smaller network topologies, the traffic load on the inter-dcell links are low and the links serve fewer nodes, resulting in high throughput. Moreover, the routing performed in DCell is not the shortest routing path, which increases the number of intermediate hops between the sender and receiver.

High cross-sectional bandwidth is a necessity for today's DCs. An industry white paper estimated that the cloud will process around 51.6 exabytes (Ebytes) of data in 2015.⁵ The network traffic pattern within a DC may be one-to-many, one-to-all, and all-to-all. For instance, serving a search query or social networking request, such as group chats

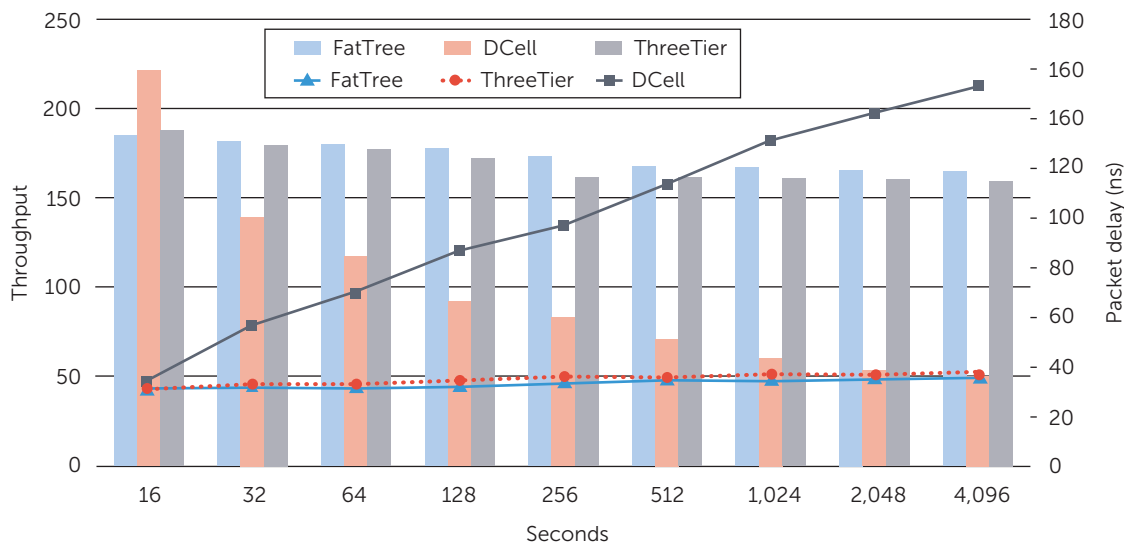


FIGURE 3. Average network throughput and packet delay of datacenter networks. As the number of nodes within the DC architecture increases, DCell experiences higher network delays and low throughput.

and file sharing, requires thousands of servers to act in parallel.⁶ The high oversubscription ratio within some DC architectures, such as ThreeTier and DCell, severely limits the internode communication bandwidth and affects performance. For instance, a typical oversubscription ratio in legacy DC architectures is between 4:1 and 8:1. An oversubscription of 4:1 means that the end host can communicate at only 25 percent of the available network bandwidth. The FatTree architecture offers a 1:1 oversubscription ratio by using a Clos-based interconnection topology. However, the FatTree architecture is not scalable and uses numerous network switches and network cables for interconnection. For example, a FatTree topology of 128 nodes (8 pod) requires 80 network switches to interconnect.

The industry is also considering the use of hybrid DC architectures (optical/electrical and wireless/electrical) to augment DC networks. Optical interconnects offer high bandwidth (up to terabytes per second per fiber), low latency, and high port density. Therefore, optical networks are certainly a possible solution for the ever-increasing bandwidth demands within DCs.⁷ Various hybrid (optical/electrical) DC architectures, such as Helios, c-Through, and HyPac, have been proposed recently to augment existing electrical DC networks.⁷ However, optical networks also face numerous challenges:

- high cost;
- high insertion loss;
- large switching and link setup time (usually 10 to 25 ms);

- inefficient packet header processing;
- unrealistic and stringent assumptions, such as networks flows without priorities, independent flows, and hashing-based flow distribution, that are effective but not applicable in real-world DC scenarios; and
- optical-electrical-optical (OEO) signal conversion delays caused by the lack of efficient bit-level processing technologies and incurred at every routing node when the optical links are used with electrical devices.⁶

Similar to the optical interconnects, emerging wireless technologies, such as 60-GHz communications, are also being considered to overcome various challenges faced by the current DC networks, such as cabling costs and complexity.⁸ However, 60-GHz technology in DCs is still in its infancy and faces numerous challenges, such as propagation loss, short communication range, line of sight, and signal attenuation.⁸ Hybrid, fully optical, and wireless DCs may be viable solutions for DC networks, but the aforementioned open challenges are currently a barrier to their widespread adoption.

Energy Efficiency in Cloud Datacenters

Concerns about environmental impacts, energy demands, and electricity costs of cloud DCs are intensifying. Various factors, such as the massive amounts of energy DCs consume, excessive greenhouse gas (GHG) emissions, and idle DC resources mandate that we consider energy efficiency as one of the foremost concerns within cloud DCs. DCs are one

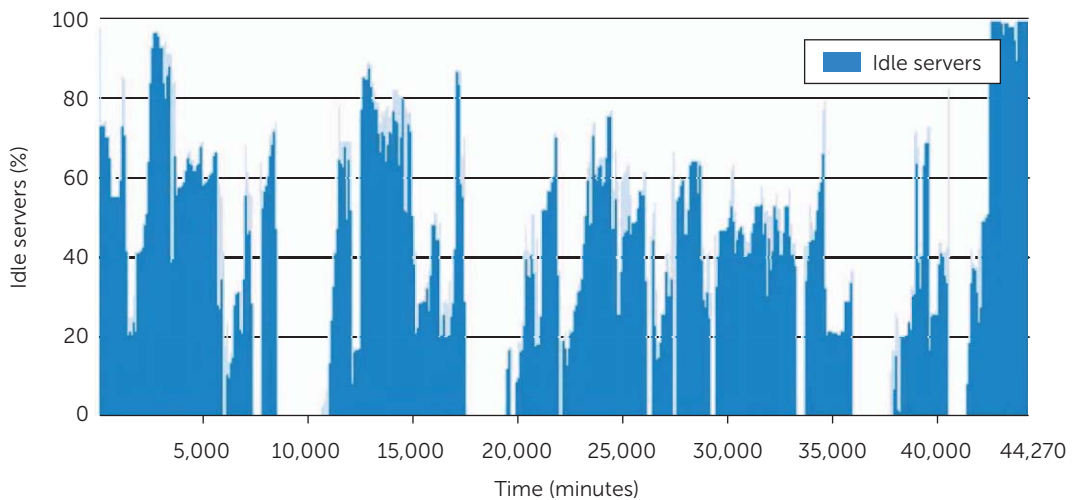


FIGURE 4. Idle servers in the University of New York at Buffalo datacenter. Careful workload placement and consolidation can result in better resource allocation and thus reduced energy consumption.

of the major energy-consuming entities worldwide. The cloud infrastructure consumed approximately 623 terawatt-hours (TWh) of energy in 2007.⁹ The estimated energy consumption of the cloud infrastructure in 2020 is 1,963.74 TWh.¹⁰ DCs are experiencing a growth rate of around 9 percent every year, and as a result, their energy demands, which have doubled in the last five years,¹¹ are continuing to increase as well.

Because of the increasing energy costs (around 18 percent in the past five years and 10 to 25 percent in the coming years), DC operational expenses are also increasing.^{11,12} The energy bill of a typical DC dominates its total operational expenses. For instance, approximately 45 percent of one IBM DC's operational expenses went to its energy bill.¹² IBM has reported that, over a 20-year period, a DC's operational expenses are three to five times that of its capital expenditures.¹² In certain cases, the energy costs may account for up to 75 percent of operational expenses.¹²

The enormous GHG emissions produced by DCs and the cloud infrastructure have intensified environmental concerns. The ICT sector's carbon footprint was approximately 227.5 metric tonnes in 2010, higher than that of the worldwide aviation industry.⁹ The cloud infrastructure's carbon footprint may be close to 1,034 Mt by 2020.¹⁰ Moreover, most of the electricity used by DCs is produced by "dirty" resources, such as coal.¹⁰ Coal-fired power stations are among the biggest sources of the GHG emissions, and the biggest source of GHG emissions in the US.

A typical DC experiences around 30 percent of average resource utilization,¹³ meaning that DC

resources are overprovisioned to handle peak loads and workload surges. Therefore, a DC's available resources remain idle most of the time. In fact, as much as 85 percent of the computing capacity of distributed systems remains idle.¹¹ As a result, significant energy savings are possible using judicial DC resource optimization techniques. We can achieve energy efficiency within DCs by exploiting workload consolidation, energy-aware workload placement, and proportional computing.

Consolidation techniques exploit resource overprovisioning and redundancy to consolidate workloads on a minimum subset of devices. Idle devices can be transitioned to sleep mode or powered off to save energy by using the Wake on LAN (WoL) feature on network interface cards (NICs). The Xen platform also provides a host power-on feature to sleep/wake devices. Researchers have recently proposed various workload consolidation strategies, such as ElasticTree and DCEERS, for energy savings within DCs.^{8,13} Such strategies use optimization techniques, such as calculating a minimum subset of devices to service the necessary workload. However, the aforementioned consolidation strategies do not consider two critical issues:

- How will the strategy handle workload surges and resource failures?
- When will the resources be transitioned to sleep/wake mode?

Activating a power off or sleep device requires a long delay (usually in seconds or minutes). Such extra delays are intolerable in SLA-constrained DC en-

vironments, so there is an immense need to consider system robustness while using energy-efficiency techniques. Similarly, energy-aware workload placement and on-the-fly task migration can help maximize resource utilization, minimize energy consumption, and control the network load.

We used a real DC workload from the University of New York at Buffalo to observe the impact of energy-aware workload placement and live migration to save energy. We observed that careful workload placement and consolidation can result in a high percentage of idle resources within a DC. Figure 4 shows that, with task migrations and proper workload placement, most of the resources remain idle in a DC. Such idle resources can be placed in sleep or low power mode to attain significant energy savings.

Proportional computing involves consuming energy in proportion to resource utilization. DC resources in an idle or underutilized state consume around 80 to 90 percent of the energy consumed during peak utilization. Proportional computing techniques, such as dynamic voltage and frequency scaling (DVFS) and adaptive link rate (ALR), aim to execute resources (processors and network links) in a scaled-down state to consume less power.⁸ Such techniques depend on a mechanism for efficiently scaling the power state of the resources up and down and a policy for determining when to alter the power state.

The DVFS technique is applied to processors to scale down a processor's voltage or frequency. However, the scaled-down state upturns the execution time of the tasks, leading to larger makespan. Similarly, ALR techniques are applied to network links to scale down the links' data rates for reduced energy consumption. The IEEE 802.3az standard uses the ALR technique to scale down Ethernet link data rates.⁸ However, the IEEE 802.3az standard only provides a mechanism to change the link's state, and true efficiency and energy savings depend on the proportional computing policies that decide when to change the state of the resources. Therefore, efficient and robustness-aware policies are mandatory to exploit the full potential of the proportional computing techniques.

Robustness in Cloud Datacenters

As the operational and architectural foundation of cloud computing, DCs hold a fundamental role in the operational and economic success of the cloud paradigm. SLA-constrained cloud environments must be robust to workload surges and perturbations as well as software and hardware failure³ to deliver the specified QoS and meet SLA requirements.

However, dynamic and virtualized cloud environments are prone to failures and workload perturbations. A small performance deprivation or minor failure in a cloud may have severe operational and economic impacts. In one incident, a small network failure in the O2 network (the UK's leading cellular network provider) affected around seven million customers for three days.³ Similarly, because of a core switch failure in BlackBerry's network, millions of customers lost Internet connectivity for three days. In other incidents, the Bank of America website outage affected around 29 million customers, and Virgin Blue airline lost approximately \$20 million because of a hardware failure in its system.¹⁴ Major brands faced service outages in 2013, including Google, Facebook, Microsoft, Amazon, Yahoo, Bank of America, and Motorola.

The cloud market is growing rapidly, and the European Network and Information Security Agency (ENISA) has projected that approximately 80 percent of public and private organizations will be cloud dependent by 2014. Many cloud service providers (CSPs) offer 99.9 percent annual availability of their services. However, a 99.9 percent availability rate still translates into 8.76 hours of annual downtime. For any cloud-dependent organization, around-the-clock availability is of utmost importance. Moreover, even a short downtime could result in huge revenue losses. For instance, in a survey of 200 DC managers, *USA Today* reported that DC downtime costs per hour exceed \$50,000.¹⁴ The business sector is expected to lose around \$108,000 for every hour of downtime. *InformationWeek* reported that IT outages result in a revenue loss of approximately \$26.5 billion per year.¹⁴

In addition to huge revenue losses, service downtimes also result in reputation damage and customer attrition. Therefore, robustness and failure resiliency within the cloud paradigm is of paramount importance. We analyzed the robustness and connectivity of the major DC architectures under various types of failures, such as random, targeted, and network-only failures.³ We found that the legacy DC architectures lack the required robustness against random and targeted failures. A single access layer switch failure disconnects all the connected servers from the network. The DCell architecture exhibits better connectivity and robustness against various types of failures. However, the DCell architecture cannot deliver the required QoS and performance necessary for large networks and heavy network loads.

Using consolidation, dynamic power (sleep/wake) management, and proportional computing

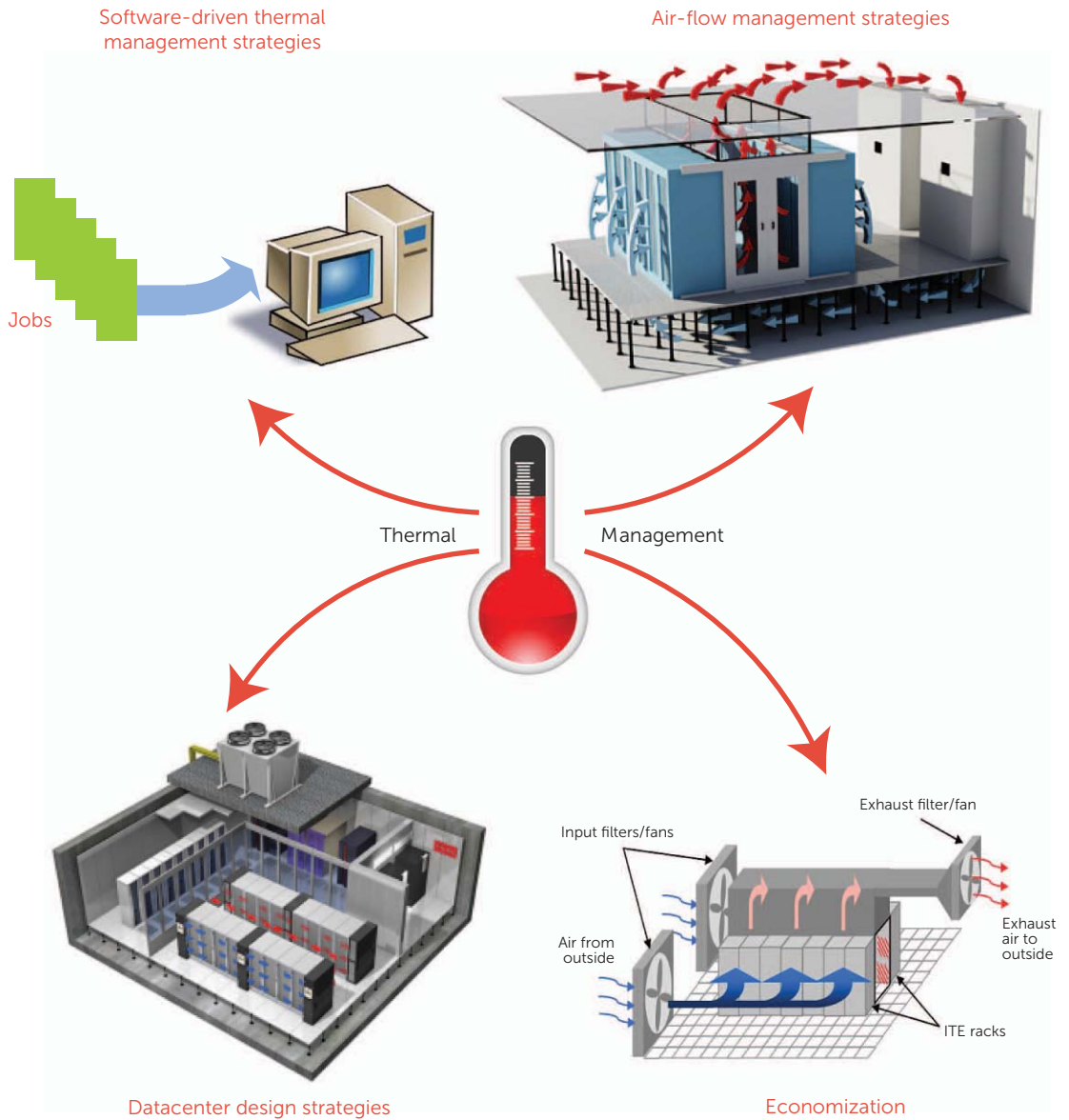


FIGURE 5. Thermal management strategies. Cloud DCs can utilize one or more strategies to regulate and manage operating temperatures.

techniques to save energy may also affect a cloud’s performance. Google reported a revenue loss of around 20 percent because of an additional response delay of 500 ms.³ Similarly, Amazon reported around 1 percent sales reduction because of an additional delay of 100 ms. In high frequency trading (HFT) systems, a delay of as small as nanoseconds may have huge financial effects.¹¹ Therefore, energy-efficient techniques must not compromise system robustness and availability. Activating sleep and power-off devices requires significant time. Similarly, scaling up a processor or network link may also result in an

extra delay and power spike. Therefore, appropriate measures must be taken to avoid any extra delays. The dynamic power management and proportional computing policies and consolidation strategies must be robust enough to handle workload surges and failures. Similarly, prediction-based techniques for forecasting future workloads and failures can also help enhance system robustness.

Thermal Awareness in Cloud Datacenters

As we stated earlier, electricity costs comprise a major portion of overall DC operational expenses. A

further breakdown of the energy consumption within a DC reveals that an overwhelming portion of those costs are incurred to stabilize the DC's thermal dynamics, such as the computer room air conditioning (CRAC) units, chillers, and fans. In a typical DC, the annual electricity cost of cooling alone is \$4 to \$8 million, including the cost of purchasing and installing the CRAC units.¹⁵ High operating temperatures can decrease the reliability of the underlying computing devices. Moreover, inappropriate air-flow management within DCs can create hotspots that may cause servers to throttle down, increasing the possibility of failures. The DC industry uses several strategies to stabilize thermal subtleties. As Figure 5 shows, we can broadly categorize such strategies into four areas:

- software-driven thermal management and temperature-aware strategies,
- DC design strategies,
- air-flow management strategies, and
- economization.

Software-driven thermal management strategies mainly focus on maintaining a thermal balance within the DC. The goal is to reduce the average heat dissipation of the servers to reduce the cost of running the CRAC unit. Such strategies adopt various methods for job allocation. For instance, genetic-algorithm-based job allocation¹⁶ attempts to select a set of feasible servers to minimize the thermal impact of job allocation, the integer linear-programming modeling approach¹⁷ aims to meet real-time deadlines while minimizing hotspots and spatial temperature differences through job scheduling, and thermodynamic-formulation and thermal-profiling-based strategies optimize the DC's thermal status.¹⁸ However, different software-driven thermal strategies produce different thermal footprints, depending on the nature of the workload being processed.

DC design strategies aim to build efficient physical DC layouts, such as a raised floor and hot and cold aisles. In a typical air-cooled DC, hot and cold aisles are separated by rows of racks. The CRAC unit's blower pressurizes the under-floor plenum with cold air that is drawn through the vents located in front of the racks in the cold aisle. The hot air coming out of the servers is pushed into the hot aisles. To enhance efficiency, DC managers have added containment systems that isolate hot and cold aisles to avoid air mixing. Initially, physical barriers, such as vinyl plastic sheeting or Plexiglas covers, were used for containment. However, today vendors

offer other commercial options, such as plenums that combine containment with variable fan drives to prevent air from mixing.

Other DC design strategies involve the placement of cooling equipment. For example, the cabinet-based strategy contains the closed-loop cooling equipment within a cabinet; the row-based strategy dedicates CRAC units to a specific row of cabinets; the perimeter-based strategy uses one or more CRAC units to supply cold air through plenums, ducts, or dampers; and the rooftop-based strategy uses central air handling units to cool the DC. Generally, equipment-placement strategies are adopted based on the physical room layout and the building's infrastructure.

The DC cooling system is significantly influenced by the air movement, cooling delivered to servers, and hot air removal dissipated from the servers. In this case, air-flow management strategies are adopted to appropriately maneuver the hot and cold air within the DC. Three air-flow management strategies are usually followed: *open*, where no intentional air flow management is deployed; *partial containment*, where an air-flow management technique is adopted but there is no complete isolation between hot and cold air flows (using hot and cold aisles); and *contained*, where the hot and cold air flows are completely isolated.

The economization strategy reduces the cost spent on cooling infrastructure by drawing in cool air from outside and expelling the hot air to outdoors. Intel IT conducted an experiment and claimed that an air economizer could potentially reduce annual operating costs by up to \$2.87 million for a 10-MW DC.¹⁹

A combination of all of the aforementioned strategies could be used to implement an efficient thermal-aware DC architecture.

Virtualization in Cloud Datacenters

The process of abstracting the original physical structure of innumerable technologies, such as a hardware platform, operating system, a storage device, or other network resources is called *virtualization*.²⁰ Virtualization is one of the key aspects used to achieve scalability and flexibility in cloud DCs and is the underlying technology that contributes application and adoption of the cloud paradigm. A virtual machine monitor (VMM) serves as an abstraction layer that controls the operations of all the VMs running on top of it. Every physical machine in the cloud hosts multiple VMs, which from a user's perspective, is equivalent to a fully functional machine. Virtualization ensures high resource utilization

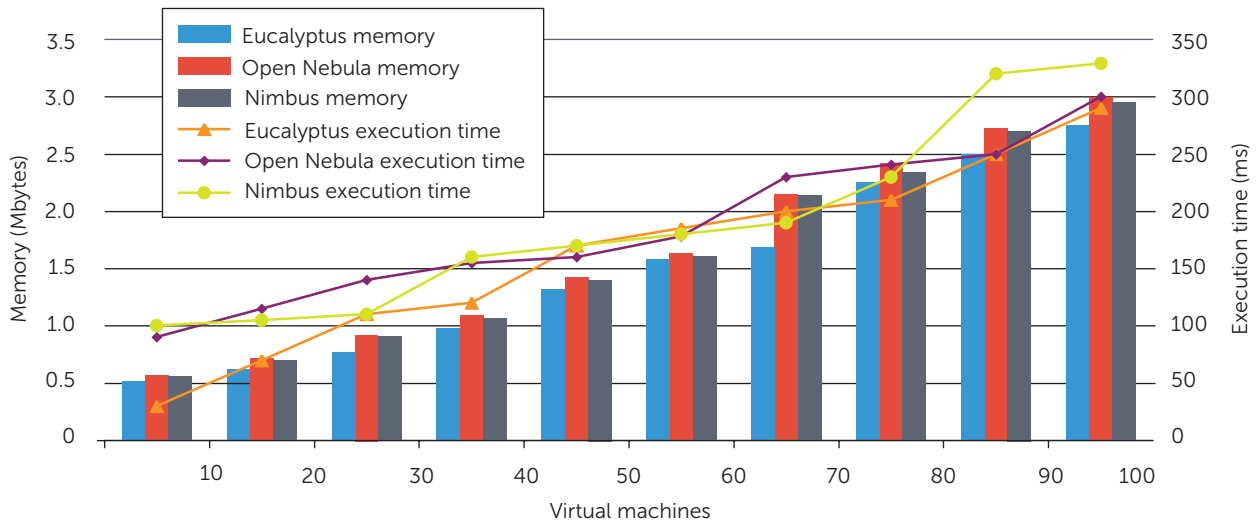


FIGURE 6. Verification time and memory consumed by VM-based cloud management platforms. The exercise to investigate the scalability of the models revealed they functioned appropriately as the numbers of VMs increased.

and thus leads to huge savings in hardware, power consumptions, and cooling. Several VM-based cloud management platforms are available, such as Eucalyptus, OpenStack, and Open Nebula.

Today, the primary focus for virtualization continues to be on servers. However, the virtualization of other components, such as storage and networks, is also evolving as a prominent strategy. Moreover, virtualization is also used in other areas: *application virtualization*, where every user has an isolated virtual application environment; *hardware-layer virtualization*, where a VMM runs directly on hardware, controlling and synchronizing the access to hardware resources; *OS-layer virtualization*, where multiple instances of the same OS run in parallel; and *full virtualization*, where the I/O devices are allotted to the guest machines by imitating the physical devices in the VMM.

Virtualization is experiencing an annual growth rate of around 42 percent.¹¹ According to a Gartner survey, the workload virtualization will increase from around 60 percent in 2012 to almost 90 percent in 2014.²¹ Several major reasons exist for this increase:

- scalability and availability,
- hardware consolidation,
- legacy applications continuing to operate on newer hardware and operating systems,
- simulated hardware and hardware configurations,
- load balancing, and,
- easy management of tasks, such as system migration, backup, and recovery.

Despite all the benefits, virtualization technology poses several serious threats and adds further challenges to efficiently and appropriately managing a DC. Moreover, network services in a virtualized environment have to look beyond the physical machine level to a lower virtual level. The advent of virtual switches and virtual topologies bring further complexity to the DC network topology. A legacy ThreeTier topology, for example, may grow to four or five tiers, which may be suboptimal and impractical in various cloud environments.¹¹ The MAC address management and scalability of the consolidated VMs is a major concern that must prevent the MAC tables from overloading in network devices.

Specifically, virtualization faces some key challenges, including *VM hopping*, where an attacker on one VM can access another VM; *VM mobility*, or the quick spread of vulnerable configurations that can be exploited to jeopardize security; *VM diversity*, where the range of operating systems creates difficulties when securing and maintaining VMs; and *cumbersome management*, where managing the configuration, network, and security-specific settings is a difficult task. The inception of the cloud is based on distributed (grid and cluster) computing and virtualization.

Previous research has focused on the computing and storage aspects of the cloud, while a crucial aspect, the connectivity (networking), is usually unaddressed.²² In a recent study, we performed formal modeling, analysis, and verification of three state-of-the-art VM-based cloud management platforms: Eucalyptus, Open Nebula, and Nimbus.²⁰ The exercise was to demonstrate the models' flexibility and

scalability. We instantiated 100 VMs and verified whether the models' functionality was affected by the increase in the number of instances. The results from the exercise revealed that the models were functioning appropriately as the numbers of VMs increased, as Figure 6 shows.

The increasing criticality and complexity involved in cloud-based applications, such as routing and VM management, necessary to deliver QoS has led to the maturity of formal method techniques. Such techniques aim to increase software quality, reveal incompleteness, remove ambiguities, and expose inconsistencies by mathematically proving program correctness as opposed to using test cases. Formal methods have gained a great deal of popularity since the famous Pentium Bug that caused Intel to recall faulty chips, resulting in a \$475 million loss.²³ Most of the well-known names connected to DCs, such as Microsoft, Google, IBM, AT&T, and Intel, have already realized the importance of formal methods and are using techniques and tools to verify the functionality and reliability of their respective software and hardware.

As we stated earlier, the occurrence of errors and miscalculations are hazardous and expensive in large-scale computing and critical systems, such as cloud and real-time systems. The *New York Times* reported one such error in August 2012: the Knight Capital Group lost \$440 million in just 45 minutes when newly installed trading software went haywire. Formal method techniques can be adopted to ensure system correctness, reliability, robustness, and safety by introducing rigorously and performing proofs and verification of the underlying systems.

Software-defined networking (SDN) involves separating the network control plane from the data plane within a network.¹¹ The data plane is considered a hardware-based portion of the device for sending and receiving network packets, whereas the control plane is the software-based portion of the network device that determines how the packets will be forwarded. SDN offers decoupled and centralized network management of the control plane to manage the whole network in a unified way. Control plane management is performed independently of the devices and the forwarding rules; for example, routing tables are assigned to the data plane on the fly using communication protocols. SDN offers high flexibility, agility, and control over a network using network programmability and automation.¹¹ The SDN market is expected to grow by \$35 billion in 2018.²⁴ Various SDN frameworks such as Cisco One and Open Daylight offer APIs, tools, and protocols for configuring and building a centrally controlled programmable network.

SDN-based automated DC networks are a possible solution to the various challenges faced by legacy DC networks, but such technologies are still in their infancy. Moreover, SDN deployment requires OpenFlow (or another SDN-based communication protocol) compliant network devices to operate, but legacy network devices do not support such communication protocols. In addition, a central SDN controller creates a single point of failure, and prevention of malicious misuse of the SDN platforms is a major security concern.

Energy efficiency, robustness, and scalability are among the foremost concerns faced by cloud DCs. Researchers and industry are striving to find the viable solutions for the challenges facing DCs. Hybrid DC architectures employing optical and wireless technologies are one of the strongest feasible solutions today. The SDN-based DCs architectures are also being considered to handle various network-related problems and to deliver high performance. The hybrid DC architectures and SDN-based DCs are still in their infancy, however. Therefore, serious research efforts are necessary to overcome the limitations and drawbacks of the emerging technologies to deliver the required QoS and performance. ●●●

References

1. Gartner, "Forecast Overview: Public Cloud Services, Worldwide," 2011–2016, 4Q12 Update, 2013.
2. K. Bilal et al., "A Taxonomy and Survey on Green Data Center Networks," to be published in *Future Generation Computer Systems*; doi:10.1016/j.future.2013.07.006.
3. K. Bilal et al., "On the Characterization of the Structural Robustness of Data Center Networks," *IEEE Trans. Cloud Computing*, vol. 1, no. 1, 2013, pp. 64–77.
4. K. Bilal et al., "Quantitative Comparisons of the State of the Art Data Center Architectures," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 12, 2013, pp. 1771–1783.
5. Bell Labs and Univ. of Melbourne, "The Power of Wireless Cloud: An Analysis of the Energy Consumption of Wireless Cloud," Apr. 2013; www.ceet.unimelb.edu.au/pdfs/ceet_white_paper_wireless_cloud.pdf.
6. A. Vahdat et al., "The Emerging Optical Data Center," *Proc. Conf. Optical Fiber Comm.*, 2011; www.opticsinfobase.org/abstract.cfm?URI=ofc-2011-otuh2.

7. C. Kachris and I. Tomkos, "A Survey on Optical Interconnects for Data Centers," *IEEE Comm. Surveys & Tutorials*, vol. 14, no. 4, 2012, pp. 1021–1036.
8. K. Bilal, S.U. Khan, and A.Y. Zomaya, "Green Data Center Networks: Challenges and Opportunities," *Proc. 11th IEEE Int'l Conf. Frontiers of Information Technology*, 2013, pp. 229–234.
9. G. Cook and J. Horn, "How Dirty Is Your Data, A Look at the Energy Choices That Power Cloud Computing," tech report, Greenpeace Int'l, 2011.
10. Greenpeace Int'l, *Make IT Green, Cloud Computing and its Contribution to Climate Change*, 2010; www.greenpeace.org/usa/Global/usa/report/2010/3/make-it-green-cloud-computing.pdf.
11. IBM, "IBM and Cisco: Together for a World Class Data Center," 2013.
12. S.L. Sams, "Discovering Hidden Costs in Your Data Center—A CFO Perspective," IBM, 2010.
13. J. Shuja et al., "Data Center Energy Efficient Resource Scheduling," *Cluster Computing*, Mar. 2014; <http://link.springer.com/article/10.1007%2Fs10586-014-0365-0>.
14. Evolgen, "Downtime, Outages and Failures—Understanding Their True Costs," Wind of Change blog, 18 Sept. 2013; <http://urlm.in/sjhk>.
15. E. Pakbaznia and M. Pedram, "Minimizing Data Center Cooling and Server Power Costs," *Proc. 14th ACM/IEEE Int'l Symp. Low Power Electronics and Design*, 2009, pp. 145–150.
16. Q. Tang, S. Gupta, and G. Varsamopoulos, "Energy-Efficient Thermal-Aware Task Scheduling for Homogeneous High-Performance Computing Data Centers: A Cyber-Physical Approach," *IEEE Trans. Parallel and Distributed Systems*, vol. 19, no. 11, 2008, pp. 1458–1472.
17. E. Kursun and C.Y. Cher, "Temperature Variation Characterization and Thermal Management of Multicore Architectures," *IEEE Micro*, vol. 29, 2009, pp.116–126.
18. J. Moore et al., "Making Scheduling 'Cool': Temperature-Aware Workload Placement in Data Centers," *Proc. Usenix Conf.*, 2005, pp. 61–75.
19. Intel Information Technology, "Reducing Data Center Cost with an Air Economizer," 2008; www.intel.com/it/pdf/Reducing_Data_Center_Cost_with_an_Air_Economizer.pdf.
20. S.U.R. Malik, S.U. Khan, and S.K. Srinivasan, "Modeling and Analysis of State-of-the-Art VM-Based Cloud Management Platforms," *IEEE Trans. Cloud Computing*, vol. 1, no. 1, 2013, pp. 50–63.
21. M. Warrilow and M. Cheung, "Will Private Cloud Adoption Increase by 2015?" research note G00250893, Gartner, May 2013.
22. F. Panzneri et al., "Distributed Computing in the 21st Century: Some Aspects of Cloud Computing," *Technology-Enhanced Systems and Tools for Collaborative Learning Scaffolding*, Springer, 2011, pp. 393–412.
23. T. Coe et al., "Computational Aspects of the Pentium Affair," *IEEE Computational Science and Eng.*, vol. 2, no. 1, 1995, pp. 18–30.
24. P. Bernier, "Openwave Exec Discusses the Benefits, Challenges of NFV and SDN," *SDN Zone Newsletter*, 12 Nov. 2013; <http://urlm.in/sjjj>.

KASHIF BILAL is a doctoral student at North Dakota State University. His research interests include cloud computing, datacenter networks, green computing, and distributed systems. Bilal has an MS in computer science from the COMSATS Institute of Information Technology, Pakistan. He is a student member of IEEE. Contact him at kashif.bilal@ndsu.edu.

SAIF UR REHMAN MALIK is a doctoral student at North Dakota State University. His research interests include formal methods, large-scale computing systems, cloud computing, and datacenters networks. Malik has a MS in computer science from COMSATS Institute of Information Technology, Pakistan. He is a student member of IEEE. Contact him at [saif.rehmanmalik@ndsu.edu](mailto:satif.rehmanmalik@ndsu.edu).

SAMEE U. KHAN is an assistant professor at North Dakota State University. His research interests include optimization, robustness, and security of cloud, grid, cluster and big data computing, social networks, wired and wireless networks, power systems, smart grids, and optical networks. Khan has a PhD in computer science from the University of Texas, Arlington. He is a senior member of IEEE. Contact him at samee.khan@ndsu.edu.

ALBERT Y. ZOMAYA is a professor at the University of Sydney. His research interests include areas of algorithms, complex systems, and parallel and distributed systems. Zomaya has a PhD from the Sheffield University in the United Kingdom. He is a fellow of IEEE. Contact him at albert.zomaya@sydney.edu.au.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.