Evaluation of Data-Center Architectures for Virtualized Network Functions

Sandra Herker, Xueli An, Wolfgang Kiess
DOCOMO Communications Laboratories Europe GmbH
Munich, Germany
{herker, an_de_luca, kiess}@docomolab-euro.com

Andreas Kirstädter
Institute of Communication Networks and Computer
Engineering (IKR), University of Stuttgart, Germany
andreas.kirstaedter@ikr.uni-stuttgart.de

Abstract—**Network Functions Virtualization (NFV) is a recent industrial trend which gains a lot of attention from telecom operators and vendors. The NFV concept delivers network services using commercial off-the-shelf hardware and IT virtualization technologies, thus virtualizing entire classes of network node functions. Traditional and recent data-center architectures are mostly built and optimized for data storage or web based applications. However, NFV type applications have different characteristics with respect to network load and needed computing capacity, which may impact the data-center design. Therefore in this paper, we are motivated to model the characteristics of NFV type applications and investigate the suitability of current data-center architectures for such new applications.**

I. INTRODUCTION

Network Functions Virtualization (NFV) is an emerging network architecture concept to virtualize network functions like firewalls, gateways, or carrier grade Network Address Translation (NAT). These are implemented as software on industry standard high volume servers, switches and storage using IT virtualization technologies into building blocks. These blocks may be connected together in the form of Virtualized Network Function (VNF) service chains to create network services. Such an approach promises to reduce Capital Expenditure (CAPEX) and Operational Expenditure (OPEX) for service deployment. Moreover, it can also reduce the time to market. Further it delivers agility and flexibility by quickly scaling up or down services to address changing demands and supporting innovation by enabling services to be delivered as software on any industry-standard server hardware [1].

Today's data-centers (DCs) are mostly designed for services where the amount of external traffic (arriving at the data-center) and the resulting traffic internal to the data-center are different. Examples are outward facing services like web type applications, or internal computing like search index calculation and data analytics, where small requests can trigger large amounts of internal communication. In comparison, NFV type applications are data intensive and require the processing of traffic streams. Here, the external traffic and the data-center internal traffic are at the comparable magnitude. For telecom grade clouds/DC hosting NFV type applications, there are four fundamental differentiating factors that need to be considered: locality, SLA management, security and trust management and the usage of inter-cloud technologies [2].

To the best of our knowledge, this work is one of the first efforts to examine the suitability of DC architectures for NFV type applications. After explaining the VNF characteristics and analyzing their differences to the web service different existing DC topologies are examined for NFV deployment in DC via simulations. The goal is to understand the DC topology influence, resource requirement, etc. for VNF chain deployment in a DC environment based on this we make recommendations on how to optimize DC architectures in order to support NFV-type workloads.

In the following sections, we first explain the related work in Section II, and then introduce the involved system components in Section III. The VNF service chain placement algorithms are introduced in Section IV, and the performance analysis for different type of DCs for VNF service chain placement is introduced in Section V. Finally, we will conclude our work in Section VI.

II. RELATED WORK

An NFV type application can be implemented as one virtual network function, or it can be implemented as a chain of VNFs. The sequence of these makes up a VNF forwarding graph, and we assume that the VNFs are run in individual Virtual Machines (VMs). Thus, VM placement plays a crucial role in the layout of a VNF in a data-center, and this has been investigated in many works. For instance in [3], Meng et al., considered VM placement with the objective of minimizing the communication cost using traffic-aware VM placement to improve the network scalability. By optimizing the placement of VMs on servers, traffic patterns among VMs can be better aligned with the communication distance between them, e.g. VMs with large mutual bandwidth usage are assigned to servers in close proximity. A comparative analysis on the impact of the traffic patterns and the network architectures (traditional DCs and recently proposed DC architectures like VL2, Fat-Tree and BCube) on the potential performance gain of traffic-aware VM placement was done. One result is that if a DC is devoted to just one application with a homogeneous traffic pattern among VMs, such as a map-reduce type of workload, then traffic-aware placement of the VMs provides little improvements. The results only indicate that a BCube architecture can greatly benefit in terms of its scalability with traffic-aware VM placement, while the VL2 sees the smallest

978-1-4673-6305-1/15/$31.00 ©2015 IEEE

benefit. The work from [4] addresses a real-time VM allocation problem for DC, which expands the technique of Markov approximation. They solve a joint tenant placement and route selection problem by exploiting multi-path routing capabilities and dynamic VM migration. In [5], two DC architectures are evaluated, FiConn and Fat-Tree, for usage of a three tier web service application in a virtualized environment. Similar to our work they use a local VM placement and compare it with a service fragmentation. Further they perform tests with failure resilience. They observe several fundamental characteristics like Fat-Tree shows less impact on the placement scheme. Being different from the above work, this paper focuses on the resource placement for NFV type applications in DCs with various architectures.

III. SYSTEM MODELING

A. Data-center topologies

Many different data-center (DC) topologies have been proposed in the literature. Our goal is to find a topology that minimizes the cost to deploy NFV type applications. We first introduce the general parameters for a DC before we explain the DC topologies.

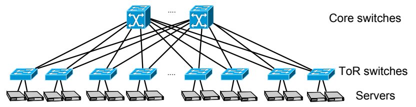
The entire DC can be modeled as a graph which consists of vertices and edges. The vertices represent switching nodes, i.e. core switch, aggregation switch and top of rack switches (ToR), and servers. The edges represent physical links between servers and switching nodes. Each server can be used to host one or multiple VMs. We assume that all servers have the same configuration in terms of CPUs, RAM and storage. In theory, the ToR switch should be capable to support bandwidth of all the servers, and the aggregation switch should support the bandwidth of all the racks connected under it. In practice if this is not the case, over-subscription is considered, which refers to a point of bandwidth consolidation where the ingress bandwidth is greater than the egress bandwidth.

The following DC topologies we are considering for our analysis.

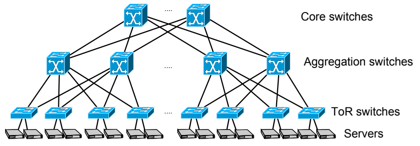
- Two-tier tree architecture as shown in Figure 1(a)
- Three-tier tree architecture as shown in Figure 1(b)
- Fat-Tree topology as shown in Figure 1(c) is a three-tier architecture that uses Clos topology [6].
- BCube as shown in Figure 1(d) is a recursively defined structure and uses servers and switches for packet forwarding [7].
- DCell as shown in Figure 1(e) is also defined recursively and uses servers and switches for packet forwarding [8].

B. Cost Modeling

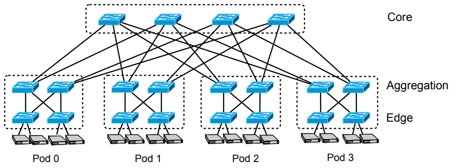
Switch and server costs are important factors to investigate DC topologies in terms of cost. However, it is not straightforward to obtain public information for such parameters directly. Therefore, we use assumptions and further analysis based on available vendor data, e.g. [9], [10]. The switch cost are different according to its corresponding speed interfaces, i.e. 10 GbE, 40 GbE or 100 GbE switch. We differentiate between 10 GbE ToR switches (monolithic architecture, up to 96 ports) and 10 GbE modular switches (up to 2048 ports). For 10 GbE ToR switches we chose per-port cost of 300 US$ (out of the range 200-450 US$ found in vendor data). For 10 GbE modular switches we chose per-port cost of 600 US$ (out of 500-900 US$ [9], [10]). For 40 GbE the per-port cost are usually 2-3 times higher than 10 GbE. We selected per-port cost of 1500 US$ at a maximum port count of 512 for our analysis. For 100 GbE we found a maximum port count of 192 and chose a per-port cost of 6000 US$. Based on the assumption that switch port cost are independent of the port count (up to the maximum port count) [11], it leads to a simple linear relation between port count and switch cost. Figure 2 shows the result of this switch cost modeling.



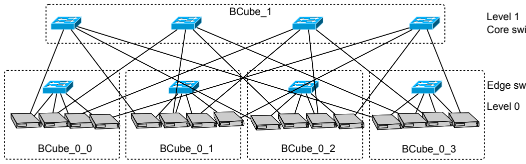
(a) two-tier tree architecture



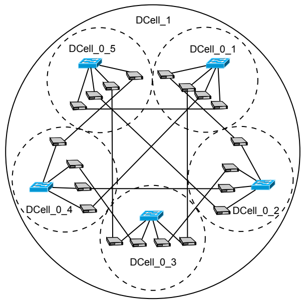
(b) three-tier tree architecture



(c) Fat-Tree architecture



(d) BCube architecture



(e) DCell architecture

Fig. 1. Data-center architectures

In addition to the switch cost, we further model the server cost. In order to highlight the bandwidth requirement for a server, the server cost are modeled by two components: server blade cost and server port cost. A server blade with ten cores can be found starting at 3000 US$ [1]. For the different DC

---

[1] http://www.dell.com/us/business/p/poweredge-blade-servers, Sept. 2014

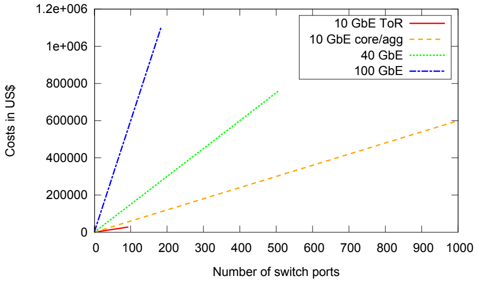


Fig. 2. Switch cost per port

topologies, 10 GbE and 40 GbE server ports are required. For a 10 GbE server port cost are 150 US$ per port. For a 40 GbE server port cost are 500 US$ per port [2]. For our DC cost modeling we did not consider cabling cost as it can be regarded as very small compared to the other cost components.

C. VNF Service Chain

VNF service chains provide typical network functions like DPI, firewall, encryption and tunneling. Such application is expected to process a large number of parallel flows, whereas a flow is defined as all packets exchanged between two end-systems that are located outside the DC. We assume that different flows do not have inter-dependencies between each other (e.g., the packets of one flow passing through our application do not influence the treatment of packets from another flow). If a VNF application needs to be run in a chain of VMs, the internal topology among VMs defines the sequence that a traffic flow passes through the DC. VNFs on the VNF service chain are deployed independently on VMs, which could be located on the same server or different servers. The VNFs can be provided by the same VNF vendor or different vendors. In the VNF service chain the flows need to traverse the function in a specific order.

Depending on the running VNFs, the traffic flow passing through a VM needs to be processed, for instance, the VM might drop packets from the flow or add additional header information to each packet. Thus, the output traffic load from the VM may not be the same as the input traffic load. However for simplicity reason, we assume that the traffic load is constant in our study. Further low latency for processing the data is needed, especially for mobile network functions.

IV. VNF SERVICE CHAIN EMBEDDING ALGORITHMS

The applied VNF Service Chain Placement (VSCP) strategy plays an important role in terms of consumed computing/storage resource and internal bandwidth of a DC. Using such strategies, we map the VNF service chain to the VM level in a DC. The goal is to map as many VNF service chain as possible in one DC to maximize operator's revenue. As the strategies how and where the operator will place the VNF functions in the DC are still unknown today, we developed the following three different VSCP strategies:

A. Local VSCP

The idea of the local placement is to keep all the VMs that run VNF application sub-functions as close as possible to minimize the DC internal consumed bandwidth. All the servers in a DC are assigned unique identifiers (ID) according to their location in the topology, e.g. server 1 is next or closest to server 2. A list is maintained with available servers, meaning, available VMs for service embedding. The available servers are sorted according to their IDs in increasing order. The server on the list top is selected for mapping a VNF service chain. If the resource on the selected server is not enough, the next server on the list is selected. For the link mapping, for instance, how to route the traffic within a DC if it is necessary, the shortest path algorithm is used.

B. Random VSCP

For this strategy a VM in the DC is chosen randomly to embed a function of a VNF service chain. The VMs can thus be on the same server or on different servers.

C. VNF Vendor Based VSCP

A VNF service chain may contain VNFs provided by different vendors, for instance, DPI from company A and tunneling from company B. In this scenario, to ensure full isolation of VNFs from different vendors and avoid the potential influence from the hypervisor and also security concerns, the servers can be pooled or clustered [12]. To cover this use case, we introduce a vendor based VSCP strategy: a server only contains VMs from a single vendor, however, they are allowed to be on the same rack. Notwithstanding, the VMs on the same chain should be placed as close as possible to the others in the VNF chain. Thus, the nearest server/nearest rack is selected via the lowest hop count between two servers. If two servers have the same hop count, the one with the lower ID is chosen.

V. PERFORMANCE ANALYSIS OF DIFFERENT DC TOPOLOGIES FOR NFV

In this section, we will examine the suitability and limitations of the different DC topologies for NFV usage while also comparing their cost via simulations.

A. Simulation setup

Our framework for analyzing the performance of the different DC topologies is custom-built and written in Java. The following DC parameters and VNF chain parameters are used in the simulations:

1) DC parameters: The DC size is determined by the amount of servers, which is within the range $[440, 16000]$. Each server has 10 cores and can host up to 10 VMs. Each VM can occupy one or multiple cores within a server.

The bandwidth allocation within a DC is shown in Table I. For the 2-/3-tier architectures (with 24 servers per rack), we use four core switches, the links between the servers and the ToR switches have a bandwidth of 10 Gbps and the links between ToR, aggregation and core switches have a bandwidth of 100 Gbps. For Fat-Tree the links between the servers and the

---

[2] http://www.mellanoxstore.com/categories/adapters.html, Sept. 2014

TABLE I
DATA-CENTER BANDWIDTH PARAMETERS

DC bandwidth	server-ToR	aggregation	core
2-tier	10 Gbps	-	100 Gbps
3-tier	10 Gbps	100 Gbps	100 Gbps
Fat-Tree	10 Gbps	20 Gbps	20 Gbps
BCube	20 Gbps	-	20 Gbps
DCell	20 Gbps	-	20 Gbps

ToR switches are also 10 Gbps while the links between ToR, aggregation and to core switches are 20 Gbps. For BCube, each link has a bandwidth of 20 Gbps and we use $k = 1$, which makes it a 2 layer switch architecture. For DCell, each link also has a bandwidth of 20 Gbps. To achieve 20 Gbps bandwidth, two 10 GbE links are used together, for instance by applying Ethernet link bundling.

2) VNF chain parameters: We assume that there are four VNFs per VNF service chain. Each of the NFVs can be built by up to 3 VMs in a row, meaning, it requires maximum 12 VMs in total to implement one VNF service chain. All the VMs are connected one after one to form a service chain. The incoming packets enter the VNF chain in the first VM and traverse all the other VMs and leave the chain at the last VM. We assume that the maximum traffic load can be processed by a VNF service chain is 5 Gbps. If there are more than one core switches in the DC, the incoming traffic will be routed into and also out of the DC using the same core switch. We assume that the required packet processing capability for each VNF is a random number between $0.65 - 2$ Gbps/core[3].

Therefore, if one VNF requires 1 Gbps/core processing capability, it needs a VM with 5 cores in order to process 5 Gbps incoming traffic.

B. Comparing of the cost of different DC topologies

First, we compare DCs with different topologies according to the major cost components for DC CAPEX, i.e. switch and server cost as shown in Figure 3. To compare the influence of DC topology on the cost, the amount of servers in the DCs is kept the same, and the total cost is calculated in order to support these amounts of servers in different DCs. The 2-tier architecture scales only to about 3500 servers. The 3-tier architecture scales up to about 8000 servers. Here we have configured the 3-tier architecture with no over-subscription rate in the aggregation layer. The higher the over-subscription rate, the lower the cost, will be. The DCell architecture shows the lowest cost compared to the other DC topologies, the reason of which is further investigated in Figure 4, which depicts the switch cost in relation to the total cost: for BCube and DCell the switch costs are less than the server cost in relation to the total cost. In general, the server cost are more than 50% of the total cost for all the DC topologies, among which, Fat-Tree has the highest proportion and the DCell has the lowest. Therefore, the reason that the DCell architecture shows the lowest cost compared to the other DC topologies is because it requires the smallest number of switches among all architectures, moreover, the switch cost used for DCell is much cheaper than 2-/3-tier. The above results provide us a general understanding about DC cost, and these DC architectures will be evaluated in the next section for VNF chain embedding.

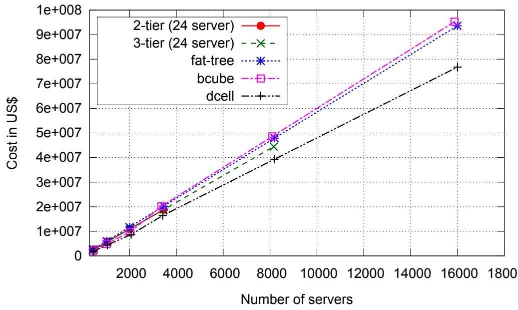


Fig. 3. Cost of the different architectures with different number of servers

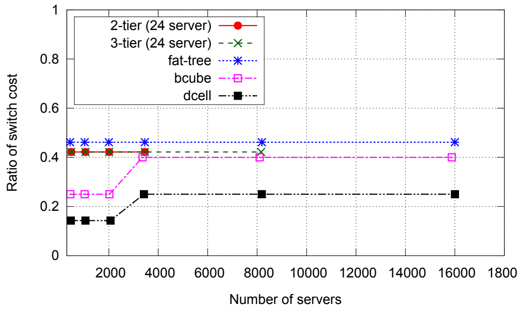


Fig. 4. Switch cost ratio of the DCs with different number of servers

C. VNF chain embedding results

1) Impact of DC topologies: We begin our simulation with short VNF chains: 4 VNFs per chain and each VNF uses one VM, which means, one VNF service chain consists of 4 VMs. The input traffic for each chain is 5 Gbps. During one simulation, we embed VNF service chain request one by one (by using different VSCP strategies) in a DC until the DC does not have enough resource to place further requests. Then we determine the number of successfully embedded VNF chains for each DC topology. Figure 5 and 6 show the results for the local and random VSCP strategy, respectively. The x-axis indicates the number of successful embedded VNF chains of each DC topology and the y-axis expresses the cost. For small scale DCs, the 2-/3- tier architectures has the cost advantage in terms of the number of embedded VNF chains. However, 2-/3-tier architectures are limited by the scalability issue. Further, 3-tier architecture with over-subscription in the aggregation results in lower number of successful VNF chain embeddings. For large scale DCs, Fat-Tree has the most successful number of embedded VNF chains for the same number of servers in the DC. It has also lower cost for the same number of embedded VNF chains compared to DCell and BCube architecture for

---

[3] One CPU core can forward 10 Gbps traffic in general [13]. For a typical middlebox application (e.g., a Firewall) the throughput per CPU core is 2.8 Gbps for a packet size of 64 byte and 10 Gbps for a packet size of 1024 byte [13][14]. Other functions like carrier grade NAT, Software BRAS and Intrusion Detection System have lower throughput, only about 1 to 1.7 Gbps for a packet size of 64 byte. Packet forwarding via the servers like in BCube and DCell is assumed with a rate of 10 Gbps per core [15].

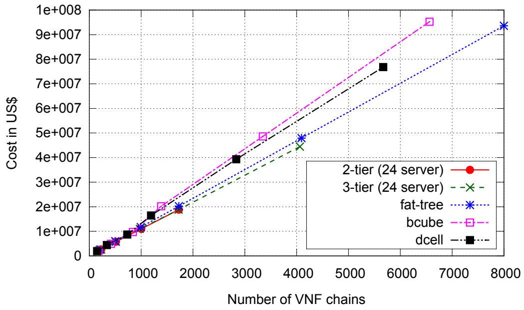


Fig. 5. Cost to successful embedded VNF chain for the local VSCP

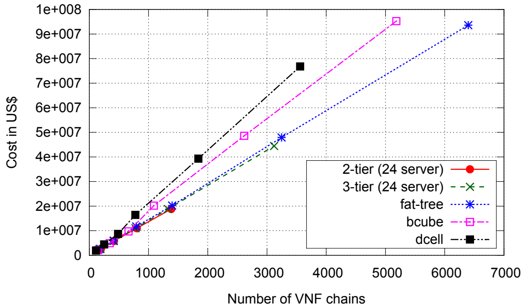


Fig. 6. Cost to successful embedded VNF chain for the random VSCP

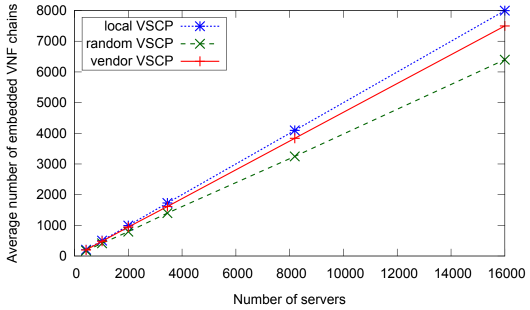


Fig. 7. Impact of VSCP strategies for Fat-Tree (with 5 Gbps chain capacity)

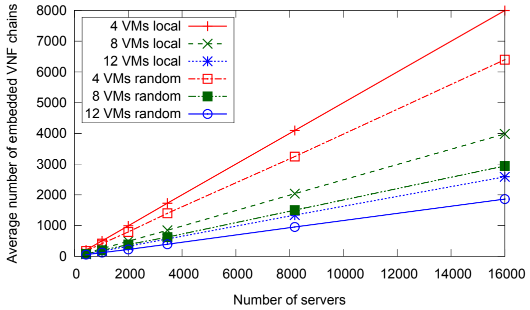


Fig. 8. Impact of VNF chain length for Fat-Tree (with 5 Gbps chain capacity)

both local and random VSCP. The reason is that part of the server computing resource is also used for packet forwarding for BCube and DCell.

We also observe that for the random VSCP the DCell performance decreases stronger than that of BCube. This is because the traffic load in DCell is less imbalanced than BCube: the level-0 links carry much higher traffic than the other links. As a result, the aggregate throughput of DCell is smaller than that of BCube.

2) Impact of VSCP strategies: An example result is shown in Figure 7 for the case of the Fat-Tree topology. The x-axis indicates the number of servers in the DC topology and the y-axis the number of successful embedded VNF chains. To test the performance of the vendor based VSCP, we assume that each VNF on the VNF service chain is from a different vendor and has to be placed on a different server. The performance of the vendor based VSCP strategy is closer to the local strategy as the VNF functions are placed also locally close but on different servers. We also simulated 2-/3-tier, BCube and DCell using different VSCP strategies. The general results trends are similar to Fat-Tree. For the random VSCP strategy, the path within a DC tends to be longer than the local VSCP for all DC topologies, which also results into more consumed bandwidth and fewer embedded VNF service chains. The difference between these two strategies gets bigger once the scale of a DC becomes bigger, as for larger DC size the hop count between the VMs becomes larger.

3) Impact of VNF service chain length: In Figure 8 the amount of embedded VNF service chains is depicted for the VNF chains with different length for DC with Fat-Tree topology. In general, a longer VNF service chain requires more computing and bandwidth resource from a DC. We compare an 8 VMs VNF chain with a 4 VMs one for the local VSCP, and observe that the number of successful embedded VNF chains is about half of the one with short VNF chain. For random VSCP the number of successful embedded VNF chains is less than half of the one with short VNF chain, which means that, random VSCP has more impact on the VNF chain length. For instance for a DC with size 16000 servers, using random VSCP has 20% performance drop in terms of the number of embedded chains with chain length 4 VMs, 26% drop for the chains with 8 VMs and 28% for the chains with 12 VMs. The reason is for the random VSCP with longer chain (e.g. 8 VMs) the embedding path lengths get about twice as long as for the short 4 VMs chain.

4) Impact of traffic load: The impact of different traffic input into the VNF chains is examined in this part to see if there is some performance advantage using smaller input traffic loads. The VNF chain processing capability is varied for 1 Gbps to 5 Gbps. Compared to 5 Gbps input traffic load, the average hop count between two VMs in the VNF chain is lower. If two VMs are on the same server the hop count is zero. The total traffic load (sum of the input load of each embedded VNF chains) is about the same for VNF service chains with 1 Gbps and 5 Gbps input traffic.

We further investigate the VSCP strategy impact on top of the VNF service chain capacity, the results are shown in Figure 9. To be able to compare all DC topologies together, we fix the number of server at around 3500. For all topologies, the random VSCP can embed fewer VNF chains than the local VSCP. This is because, when the random VSCP is used, the aggregation links becomes the bottleneck of 3-tier

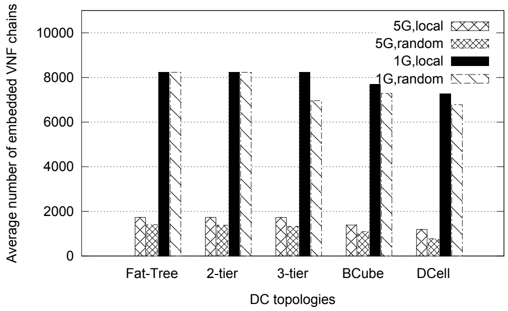


Fig. 9. Impact of VNF service chain capacity, with chain length 4 VMs

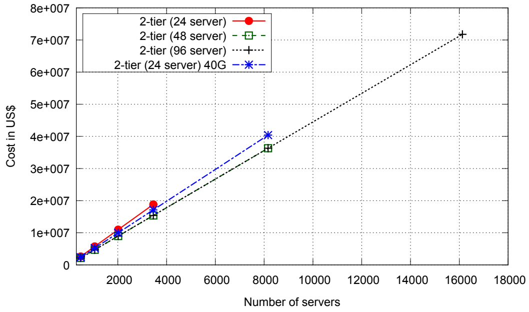


Fig. 10. Cost of 2-tier architecture with different number of servers per rack

architecture. For DCell the bottleneck are the links from server to switch. However, such difference is very small (almost no difference) for Fat-Tree and 2-tier when the VNF service chain capacity has a low value, i.e. 1 Gbps. Once the VNF service chain capacity is increased, meaning, there will be less VNF embedding requests, the Fat-Tree and 2-tier also show performance difference between random and local VSCP. The reason is that for higher input traffic also more CPU cores are required and for random VSCP it gets difficult to find servers with enough unoccupied cores after several embedded VNF chains. Therefore the CPU resources can not fully be utilized and as result less VNF chains were embedded. We investigate the Fat-Tree topology using local VSCP strategy. Using 1 Gbps input for the VNF chain results in not exactly 5 times more VNF chains as for the 5 Gbps input, which is about 2 to 5% lower. However, the bandwidth usage in the DC is much lower (about 60% lower). The reason is that with 1 Gbps input a 4 VMs service chain can always be embedded on an unoccupied server, which does not require inter-DC packets forwarding. However, this is not the case for 5 Gbps traffic load which requires more cores as on unoccupied server can offer.

5) Impact of the number of servers per rack: Further, we compared different number of servers (24, 48, or 96) per rack. Due to the space limit, we take the 2-tier architecture as the scenario for discussion. Increasing the rack size for 2-tier architecture will result in a bigger DC scale, which means that the rack size has a high influence on its corresponding DC size. Four-core switches are used for the scenario 24 and 48 servers per rack, and eight-core switches are used for the scenario 96 servers per rack. For 96 servers per rack the DC can at least scale up to 16000 servers. We also consider an alternative 2-tier architecture with 40 GbE links between ToR and 8 core switches. For the other 2-tier configurations this is not possible due to the limited number of server ports at the switches. The performance is similar to the 2-tier with 24 servers for local VSCP and for the random VSCP the larger rack size architecture have embedded about 3% less VNF chains. However, the cost for the larger rack size 2-tier architectures are a bit lower using bigger rack sizes as shown in Figure 10.

We have also done simulation with different rack size for 3-tier architecture. The overall cost can be reduced by using bigger rack size. However, the 3-tier architecture is more influenced from the rack size, because the bigger the rack size the more bandwidth is required in the aggregation links to keep the over-subscription rate low. With bigger rack size this gets difficult and therefore the performance decreased, few number of VNF service chain could be embedded, especially for the chains with long service chain length.

D. Summary and discussion

This section summarizes the DC impact factors to deploy high traffic volume NFV type applications. The 2-tier architecture performs well for VNF chains embedding with corresponding low cost compared to the other DC topologies. However, the number of ports per switch limits its scalability, because every core switch is connected to every ToR switch. For large DC size switches with high port numbers are required, however, this does not always exist (e.g. 100 GbE switches are only available with up to 192 ports). The 3-tier architecture also indicates good cost performance for VNF service chain embedding, however, it is not so robust with the VSCP strategies like random placement. The 3-tier architecture has a bandwidth bottleneck at the aggregation layer, even with no over-subscription at this layer. Comparing different over-subscription rates for 3-tier architecture, it shows that it plays an important role in the performance of the VNF chain embedding, i.e. low over-subscription rate results in good performance but high cost. In contrast, the higher the over-subscription rate, the lower the cost, which also results in the lower amount of embedded VNF service chains. With no or close to no over-subscription the performance of the 3-tier is similar to the 2-tier architecture and also shows the scalability problem due to the limited number of port of the switches.

Fat-Tree overcomes the bottleneck of conventional tree by introducing more bandwidth into the switches near the root. It is easy to scale up for large DC due to the low switch port number. The Fat-Tree performs well in terms of robustness when different VSCP strategies are applied for VNF chains embedding. However, it has the highest cost compared to the other DC topologies with the same number of servers because of the high amount of required switches.

The performance of BCube is lower compared to Fat-Tree and 2-tier architecture. The reason is the additional computing resource needed for the servers to execute packets forwarding instead of using switches for forwarding. It has a fully meshed architecture which makes the cabling more difficult compared to the tree-based architecture (2-/3-tier and Fat-Tree). The performance of DCell is lower than BCube. The reason is that, compared to BCube, more forwarding is done at server level for DCell, which also means that the links between servers and switches are also easily overloaded. Further the fully meshed structure makes the cabling even more difficult compared to the BCube architecture, which makes DCell difficult to be used for larger DCs. Also it is not straightforward to add any number servers in the DC, because of the the double-exponential growth of the servers in the network.

E. Recommendations for operators

The placement strategy is important for NFV type application deployment according to different vendor's policy and preferences. However, it does not have a big impact in terms of deployment cost. For lower input traffic (as e.g. 1 Gbps per service chain) the random and local VSCP perform similar - delivering nearly the same number of embedded VNF chains with short chain length.

The DC topologies can be evaluated from many aspects in terms of VNF service chain deployment. For instance, the architecture that can support high resource utilization or lower cost to construct such DC are preferred. Due to scalability issues it is not straightforward to compare deployment cost for NFV type applications for all type of DCs. The scenarios for small scale DCs (less than 4000 servers) and for large scale DCs (more than 8000 servers) have to be considered separately. For small DCs the 2-tier architecture performs well - while the Fat-Tree architecture performs well for large DCs and can scale to very large server numbers.

VI. CONCLUSION

In this paper we have examined DC architectures to deploy NFV type applications. We compared the cost of the different architectures and their performance for embedding VNF service chains. We showed the limitations of the different DC architectures. Further we gave some recommendations for future NFV type applications deployment in DCs. The"best" DC topology depends on the specific requirements of the operators. However, in general we can say that 2-tier tree architecture is a suitable solution for smaller size DC and the Fat-Tree for large DC.

REFERENCES

[1] M. Chiosi *et al.*, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action," White paper ETSI, Tech. Rep., 2012.

[2] S. Alliance, "Telecom Grade cloud Computing v1.0," White paper, Tech. Rep., 2011. [Online]. Available: http://scope-alliance.org/sites/default/files/documents/cloudComputing_Scope_1.0.pdf

[3] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–9.

[4] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang, "Joint VM placement and routing for data center traffic engineering," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 2876–2880.

[5] Y. Zhang, A.-J. Su, and G. Jiang, "Evaluating the impact of data center network architectures on application performance in virtualized environments," in *2010 18th International Workshop on Quality of Service (IWQoS)*. IEEE, 2010, pp. 1–5.

[6] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 63–74, 2008.

[7] 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," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 63–74, 2009.

[8] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "DCell: a scalable and fault-tolerant network structure for data centers," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 75–86, 2008.

[9] "Arista networks," August 2014. [Online]. Available: http://www.arista.com/

[10] "Cisco systems," August 2014. [Online]. Available: http://www.cisco.com/

[11] L. Popa, S. Ratnasamy, G. Iannaccone, A. Krishnamurthy, and I. Stoica, "A Cost Comparison of Datacenter Network Architectures," in *Proc. of the 6th International Conference*, ser. Co-NEXT '10. ACM, 2010.

[12] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, and X. Zhu, "VMware Distributed Resource Management: Design, Implementation, and Lessons Learned," in *VMware Technical Journal*, Spring 2012.

[13] "D5.3: Application Development and Deployment," FP7 CHANGE Project, Tech. Rep., 2013.

[14] J. Martins, M. Ahmed, C. Raiciu, V. Olteanu, M. Honda, R. Bifulco, and F. Huici, "ClickOS and the Art of Network Function Virtualization," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'14. USENIX Association, 2014.

[15] L. Popa, N. Egi, S. Ratnasamy, and I. Stoica, "Building Extensible Networks with Rule-based Forwarding," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'10. USENIX Association, 2010.