The Benefits of a Disaggregated Data Centre: A Resource Allocation Approach

Antonios D. Papaioannou, Reza Nejabati, Dimitra Simeonidou
High Performance Networks Group
University of Bristol
BS8 1UB, Bristol, United Kingdom
antonis.papaioannou@bristol.ac.uk

***Abstract*—Disaggregation of IT resources has been proposed as an alternative configuration for data centres. Comparing to the monolithic server approach that data centres are being built now, in a disaggregated data centre, CPU, memory and storage are separate resource blades and they are interconnected via a network fabric. That brings greater flexibility and improvements to the future data centres in terms of utilization efficiency and energy consumption. The key enabler for the disaggregated data centre is the network, which should support the bandwidth and latency requirements of the communication that is currently inside the server. In addition, a management software is required to create the logical connection of the resources needed by an application. In this paper, we propose a disaggregated data centre network architecture, we present the first scheduling algorithm specifically designed for disaggregated computing and we demonstrate the benefits that disaggregation will bring to operators.**

***Index Terms*—Disaggregated data centre; scheduling; resource allocation; energy consumption**

I. Introduction

Data centre infrastructure comprises a large number of servers interconnected by a network. From a high level point of view, a server is a fixed set of computing resources, i.e. processor (CPU), memory (RAM), networking and storage, all connected to a single motherboard that can only be used by the applications running on the given server. Despite the fact that data centres are able to simultaneously perform complex tasks and serve thousands of users at the same time, an alternative configuration has been suggested to take full advantage of their capabilities in entirety. In this configuration, the computing resources are disaggregated, meaning that each resource type (e.g. RAM, CPU, graphic processing units (GPU), etc.) will exist on a separate resource blades that are distributed across the data centre and are interconnected over the network. When a task is to be executed, there is a logical aggregation of the required resources as opposed to physical that exist in current data centres. Figure 1 shows an abstract overview of the differences between these two configurations.

Disaggregation of resources allows the data centre operators to have flexible and granular control over their resources. Further advantages of disaggregation include enabling the upgrade of any resources to the state-of-the art technology, customizing the existing resources or even adding specialized hardware (e.g. GPUs, media encoders/decoders) in an easy and cost effective way. Finally, disaggregation allows independent evolution of each technology and instant deployment.

In disaggregated data centres, network is the crucial element, which controls the overall performance, and the greatest challenge. On top of the usual data centre traffic, the network has to serve the additional communication that is currently inside the server, and is handled by the chipset and motherboard buses. Therefore, it is imperative that the network transforms to become faster both in terms of latency and capacity. In Table I, there is a summary of the network requirements in order for disaggregated data centres to operate successfully [1] [2]. Due to the combination of high bandwidth requirements and low latencies, optics are adopted for the disaggregated data centre network. In terms of capacity, traffic can be accommodated by existing optical technologies, however latency is the critical parameter. Faster interconnects and switches with low port-to-port latency are yet to be developed in order to enable disaggregated data centre scaling to thousands of blades. Proper network management that is specifically designed for disaggregated data centres is required in order ensure a smooth operation and a steady performance.

Disaggregating resources is a promising solution to improve resource utilization efficiency and energy consumption of data centres. Large scale data centres suffer from underutilization of their hardware resources. To elaborate, consider a commodity server with 8 cores and 64 GB of RAM in conjunction with an application request of 8 cores and 16 GB of memory. In a normal data centre, all CPU would be assigned and 48 GB of RAM would be completely unutilized and therefore wasted. On the other hand, in a disaggregated data centre, where the resources are organised into independent blades, the 64 GB memory blade will still have 48 GB available to be used by for another independent application. Hypervisors with over-subscription capabilities are currently dealing with this problem and improving the utilization of the servers [3], however the integration of hypervisors into a disaggregated data centre further increases utilization, since when one application is idle, the rest are still using the assigned resources. In summary, disaggregation transforms the data centre, making it more modular and flexible, providing a fine-grain control over the resources.

In a disaggregated data centre, the coordination of all the hardware and software falls under the control of the

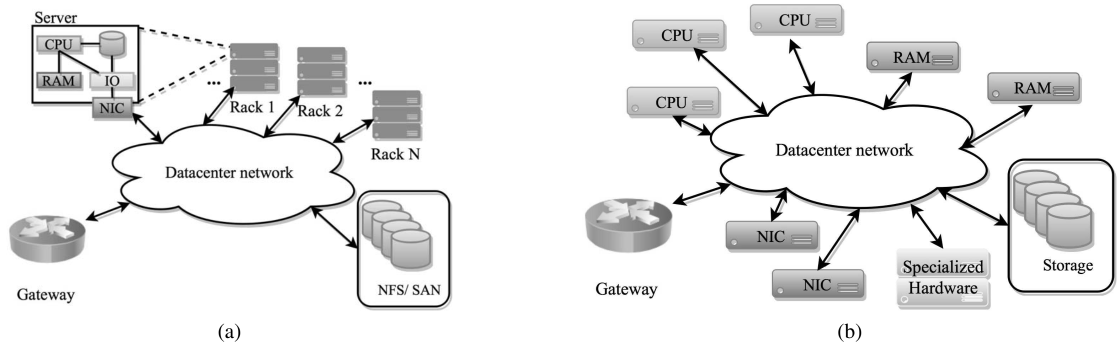

Fig. 1. (a) The traditional data centre architectural approach, (b) The disaggregated data centre approach

TABLE I
RESOURCE COMMUNICATION REQUIREMENTS FOR DISAGGREGATED DATA CENTRES

Communication	Latency	Bandwidth	Link Length
CPU - CPU	10 ns	200 - 320 Gbps/CPU	0.1 - 1 m
CPU - Memory	10 - 50 ns	300 - 800 Gbps/CPU	1 - 5 m
CPU - Disk	1 - 10 us	5 - 128 Gbps/device	5 m - 1 km

management software. A key component of the management software is the scheduler. In a normal data centre, scheduling is the process that decides upon the compute resource allocation for a virtual machine (VM). In a disaggregated data centre, network resources have to be taken into consideration in order to solve the VM placement problem, and satisfy the requirements presented in Table I.

This paper firstly addresses the network challenges for delay sensitive communications in disaggregated data centres. Next, the paper also proposes, for the first time, a resource scheduling and network management algorithm designed for a disaggregated data centre. It is a network aware algorithm aiming to increase the utilization of compute resources without affecting performance, while taking into account the special network latency requirements. The target is to create a technology agnostic algorithm that, with awareness of the specific needs of a disaggregated data centre, prioritizes the needs of an application, finds suitable resources to serve a task and brings up the benefits of disaggregation.

II. Related Work

Research effort, from both industry and academia, has been directed towards the realization of disaggregated data centres. SeaMicro has developed a single server architecture, where small compute units are interconnected through FPGAs, in order to serve computationally heavy tasks [4]. Intel and HP have extended this idea to work at a rack scale, with separated compute resources [5] [6]. Academia has also contributed in this direction with examples such as soNUMA [7] and Firebox [8] that allows the operating system to access the remote memory. In [9], page swapping techniques were proposed. Researchers have also focused on the network requirements in order to support disaggregated data centres [1] [2], and have indicated the need for a unified resource management software in order to take advantage of disaggregation.

To the best of the authors' collective knowledge, there is no previous work on scheduling for disaggregated data centres. To address the problem effectively, resource management in cloud data centres have been studied, with a special focus on networks, due to the disaggregated data centre requirements. Scheduling has gained a lot of attention from the research community. Energy consumption and utilization of servers are the main objectives for the scheduling software. However, most of them only focus on the compute resources and completely ignore the network. Considering that the network consumes approximately 15% of the total energy in a data centre [10], makes it a non-negligible source of consumption.

A network-aware resource allocation algorithm for cloud computing has been proposed in [11]. Authors assume that CPU/memory capacity tools have already decided the number of VMs a server can accommodate and have divided the server into slots. A slot refers to one CPU/memory allocation to a VM. They also consider static and single-path routing between a pair of hosts and calculate the communication cost. The problem itself is non-deterministic polynomial hard (NP - hard) and the authors resolve it with two heuristic algorithms. The first one implements the Cluster-and-Cut approach aiming VM pairs with heavy mutual traffic to be assigned to slot pairs with low-cost connections. The principle of the second one is divide-and-conquer. The algorithms partitions slots in slot-clusters and VMs in VM clusters and initially tries to associate slot-clusters to VM clusters and then recursively maps VMs to slots. The VM clustering is done through a Min-Cut algorithm that groups VMs with large traffic into the same group. Slot-clusters is defined according to slot's cost connection.

In [12], Biran *et al.* introduce the Min Cut Ratio-aware VM placement problem (MCRVMP). MCRVMP considers constraints on both IT resources, such as CPU and memory and network resources emerging from the complexity of network topologies and routing schemas in modern cloud data centres. MCRVMP starts with the assumption that traffic demands are time-variant and works on minimizing the maximum ratio of the demand and the capacity across the network, in order to find network cuts that can absorb unpredictable traffic bursts.

Since MCRVMP is a NP-hard problem, two heuristics have been applied. The first one places VMs interacting intensively in close proximity to each other. In more detail, the first phase assigns connected VMs to subtrees rooting to relevant switches recursively. In the second phase, it splits the connected VMs and tries to place according to the problem defined above, which is possible to solve on a small scale. The second heuristic follows a greedy approach. It provisions VMs on available hosts. The only objective is to identify the traffic demands from two connected VMs, sort them from high to low and then attempt to place the VM in the host that minimizes the maximum cut. The results suggest that MCRVMP placements improve the data centre scalability and support time-varying traffic demands. Heuristics can be efficiently implemented in medium size data centres in order to have a low overhead on the scheduling procedure.

These algorithms provide good indication towards the scheduling problem solution, aiming to improve the overall network utilization in a cloud environment. However, there are two drawbacks. They both assume that the management software knows beforehand the entirety of the VM requests that will arrive. In a real data centre scenario, the pattern of incoming VM requests and their requirements are not known, and the scheduling software should be able to adapt and take the best possible decision in real-time. Additionally, they are not designed for a disaggregated data centres, and hence, they don't consider that a logical combination of resources is required to complete a certain task. Finally, they both implement a best-effort approach for networking resources, while disaggregated data centres have networking standards.

III. Disaggregated Data Centre Architecture

The feasibility, implementation and scaling of a disaggregated data centre heavily depend on the network architecture. We are proposing a scalable network architecture which relies upon both optical and electrical network switching equipment, while simultaneously satisfying the latency requirements of a disaggregated data centre. Figure 2 depicts a high level view of this architecture, which is separated in two independent networks. There is an all optical network (lower yellow part of the image), which in this paper will be called "*fast backplane*", and a second network, which can be either electrical, hybrid or all optical that will be called the "*generic backplane*".

The reason of the existence of two different networks is that there are two traffic patterns with distinct requirements. The first one represents all the communication that occurs inside the servers within a traditional data centre, i.e. the communication between CPU and RAM, GPU, network cards, etc. As shown in Table I, this communication is characterized by high bandwidth and very low latency. Studies have shown that small variance of delay may cause serious performance degradation [2]. Optics can provide low and deterministic delay, and therefore a steady performance. To achieve 10-50 ns latency for CPU-to-RAM traffic, the network architecture must be simple, with few links. The optical links should be as short as possible and the port-to-port latency in switches must be low. Normal fibres have an approximate delay of 5 ns per meter. Recently, hollow core fibres have achieved 30% less latency than conventional fibres. This means that the fabric can increase from 5 m up to 9 m, depending on the network design. As measured in the lab, optical circuit switches (OCS) and wavelength selective switches (WSS) can provide low port-to-port switching delay - 5 ns and 20 ns respectively. The same measurements for Ethernet and INFINIBAND Top-of-Rack (ToR) switches are 250 ns and 100 ns [13]. Along with buffering in electronic switches and routers in data centres, classic network architectures are inappropriate for such latency sensitive communication. The communication between the CPU and the disk doesn't fall into this category since the latency requirements are quite looser and the bandwidth capacity needed is achievable by the "*generic backplane*". The discovery of available resources that satisfy the Table I criteria are controlled by the scheduling algorithm.

Fig. 2. Proposed Disaggregated Data Centre Architecture

The second traffic pattern represents the communication that currently exists in a data centre. The "*generic backplane*" would provide the necessary flexibility for VM-to-VM and storage communication. This traffic might have high bandwidth requirements, but is not latency sensitive. Therefore, it is not affected from the buffering in switches due to the varying buffer length.

The presence of two independent backplanes, allows easier integration of the disaggregated technology to existing data centres. The "*fast backplane*" will be exclusive for the resource blades, while the "*generic backplane*" is part of the existing operator network fabric and will serve outward communication.

IV. Scheduler

In a modern data centre, the scheduler is the element of the software stack that is responsible for finding available resources, applying the respective policies and deciding in which server the VM will be provisioned [14]. On top of the allocation of compute resources, the scheduler of a disaggregated data centre has to provision network resources. Therefore, it should have a global overview of the environment and knowledge of the specific characteristics and requirements of disaggregation.

In order to study the benefits of a disaggregated data centre, we have designed a scheduler, based on the following assumptions. According to [2], CPU-to-CPU communication is not possible with current technologies and will be challenging to implement in the future, due to its extreme requirements. Moreover, as long as processor power keeps pace with Moore's Law, it is safe to assume that there will be no application requiring more cores than a single server can provide. As the basic request for resources, a VM request is considered, since it is the standardized way to allocate resources in current data centres. However, that doesn't exclude the case of allocating resources on a task-to-task basis in the future. Finally, it is assumed that some cache memory is available on every CPU blade. Fetching memory blocks from the RAM blade for every memory access will multiply the traffic. In addition, local memory is necessary in order to have reasonable performance, since the access time is affected by network latency. So, the RAM blade can be considered as a memory expansion module that, as opposed to cache memory, can be allocated to any CPU blade.

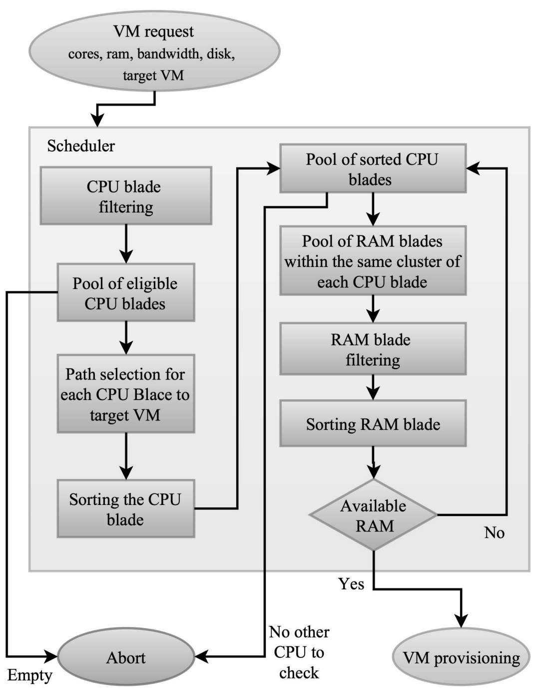

Fig. 3. Scheduler Flowchart

The flowchart of the proposed scheduler is depicted in Figure 3. It provides system administrators with the flexibility to define their own way of managing resources. This is achieved by applying policies which are enforced through three sets of weights. The scheduler takes the weights into account while ensuring smooth operation and the strict network requirements of a disaggregated data centre will be met. The scheduler is an iteration of filtering, prioritizing and sorting the network and computing resources that takes the best decision according to the policy provided. It is assumed that each VM request comes with a set of requirements, namely CPU cores, RAM size and network bandwidth for each of the target VMs that needs to be connected. The scheduler doesn't have knowledge of the upcoming VM requests, so it performs a real-time, one-off decision based on the current state of the disaggregated data centre. The steps undertaken by the scheduler to define the necessary resources are the following:

1) It filters all the CPU blades and determines those that have available cores and network bandwidth for VM-to-VM communication.
2) It prioritizes them according to whether they can accommodate the VM with the currently assigned memory.
3) For each eligible CPU blade, the scheduler finds the path that satisfies the following equation for each path connecting the given CPU blade with the target VMs:

$$min(wp_d * p_d + wp_b * p_b + wp_h * p_h) \qquad (1)$$

where $wp_d + wp_b + wp_h = 1$ and p_d, p_b, p_h are path's delay, bandwidth and hops respectively.

4) It sorts the CPU blades according to performance and utilization factors, as they have been defined by the system operator in the CPU weights:

$$wc_p * c_p + wc_u * c_u + wsp_b * sp_b + wsp_d * sp_d \qquad (2)$$

where $wc_p + wc_u + wsp_b + wsp_d = 1$ and c_p, c_u, sp_b, sp_d are the CPU blade priority and utilization and selected path bandwidth and delay, respectively. This equation maximizes for the best possible blade.

Finally, the scheduler takes the sorted list of available blades and tries to allocate an available RAM blade. If it finds a RAM blade, it stops and returns the resources that should be allocated. To find a suitable ram blade, the scheduler:

5) Filters the RAM blades and finds those that have enough space and have a path that can satisfy the latency requirement.
6) Sorts the RAM blades according to the latency to the target CPU blade and its utilization according to the following equation:

$$wr_u * r_u + wrp_b * rp_b + wrp_d * rp_d \qquad (3)$$

where $wr_u + wrp_b + wrp_d = 1$ and r_u, rp_b, rp_d are the RAM blade utilization and path bandwidth and delay for CPU-to-RAM communication, respectively. Again, the advantage of latency and bandwidth over utilization in decision process is defined by the system administrator through the set of weights.

In modern data centres, oversubscription of CPU and RAM is used to increase the utilization of the virtualized computing resources. In disaggregated data centres the same rule applies and can increase even further the resource utilization. Therefore, oversubscription ratios are considered in the scheduling algorithm when calculating available resources, if they are set by the administrator.

V. Simulation

In order to investigate the benefits of a disaggregated data centre and the effectiveness of the proposed algorithm a simulation has been created. Three scenarios have been simulated and compared:

1) The "*Smart*" or "*Network-aware*" scenario. A disaggregated data centre with the proposed scheduler.
2) The "*Round-Robin*" scenario. A disaggregated data centre with a scheduler following the Round-Robin approach and choosing the blades with the most available resources.
3) The "*Traditional*" scenario. A modern data centre with the default scheduling OpenStack algorithm, which implements Round-Robin logic and chooses the server with the most available RAM.

The amount of computing resources and the network configuration are the same in all scenarios to ensure fairness and to properly evaluate the proposed algorithm. We have considered a data centre with 900 devices split into 30 racks. In scenario 3, each of the 900 servers has 8 cores, 64 GB of RAM, one 1 G Ethernet card to connect to the "*generic backplane*" and one 40 G network card to connect to the "*fast backplane*". In scenarios 1 and 2, this amount of computing power translates into 600 CPU blades and 300 RAM blades. Each CPU blade has 12 cores, 24 GB of RAM, and is similarly connected to both backplanes. The RAM blades are equipped with 144 GB RAM and one 40 G network card and they are only connected in the "*fast backplane*". The oversubscription ratio for CPUs is 4, while it is 1.5 for RAM.

Although the scheduler is able to calculate the latency between and a RAM and a CPU blade before deciding whether the latency requirements are satisfied, the racks are organized into clusters. In the simulation, clusters are groups of racks that have been calculated offline for simplicity and ensure that the latency requirements for CPU-to-RAM communication will be preserved.

As shown in Figure 4, in the disaggregated environment, the "*fast backplane* is simulated as a single OCS switch per cluster. This implementation detail doesn't affect the results, since CPU blades can connect only with RAM blades within their cluster. In all cases, for VM communication, a hybrid leaf-spine network architecture has been simulated. The leaf switches are 1 G ToR switches with 10 G electrical uplink and 40 G optical uplink. The spine consists of both an electrical and an optical switch. Traffic from VMs that require large amount of bandwidth will be redirected through the optical link in order to improve the overall throughput.

All cases are tested under the same set of VM requests. The VM request arrival time and life duration is estimated according to the study of Peng *et al.* in [15], and the simulation runs for 40000 time units. The requirements of each VM request are chosen randomly between the options in Table II. The choices are the default cloudlets in DigitalOcean [16].

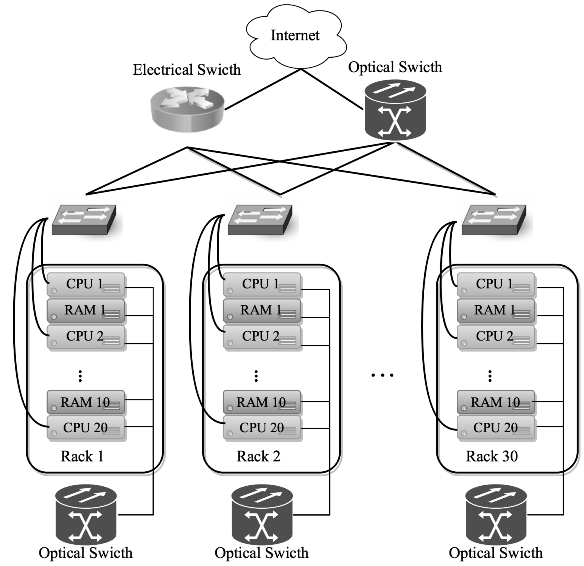

Fig. 4. Simulation Setup

TABLE II
VM Instances

Name	Cores	RAM (MB)	Bandwidth (Mbps)
Tiny	1	512	25
Small	1	1024	50
Medium	2	2048	75
Large	2	4096	100
Xlarge	4	8192	125
XXlarge	8	16384	150

Finally, the target VMs are again chosen randomly between the existing VMs. The number of target VMs is limited to 3.

The policy applied was balanced between performance and utilization criteria. The reason was that in a real data centre the operator would like to increase the utilization of each machine, but not stack all the VMs in a single server because this would increase the probability of a service level agreement (SLA) violation that would cost the operator more than distributing the load to multiple servers.

Storage load and utilization in a disaggregated data centre, assuming we are following the proposed architecture, is similar to modern data centres. There is extensive work that demonstrates techniques increasing the network's and storage servers' utilization and are applicable to a disaggregated data centre [17], therefore it is not part of the simulation.

VI. Results

The simulation results clearly demonstrate the benefits of the proposed scheduler in a disaggregated data centre in terms of active blades and average host utilization. With the word "active", we describe the blades, servers or switches that have assigned resources to one or more VMs. As shown in Figure 5, at the end of the simulation in its stable state, the proposed scheduler uses approximately 100 - 120 blades to serve the given load. On the other hand, in the "Traditional" scenario

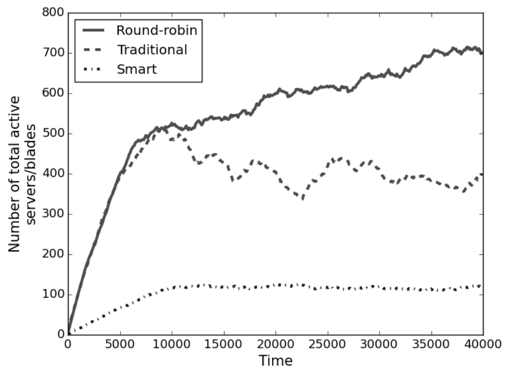

Fig. 5. Active CPU blades / servers

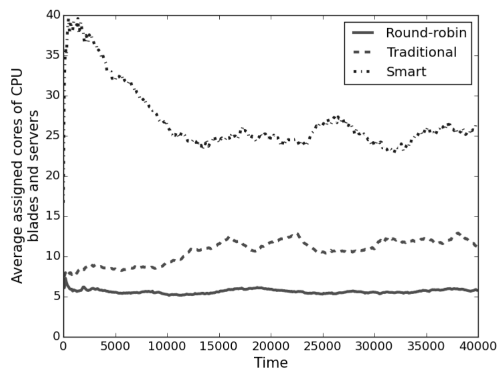

Fig. 6. Average assigned cores per active CPU blade/server

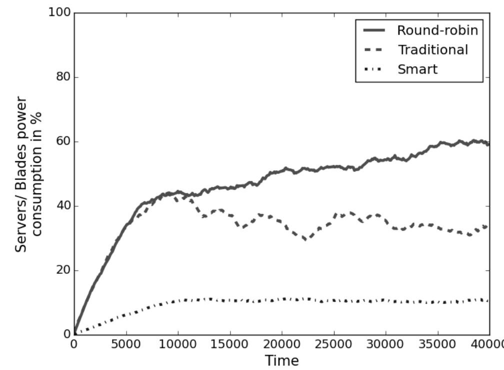

Fig. 7. Percentage of maximum power consumed for compute resources

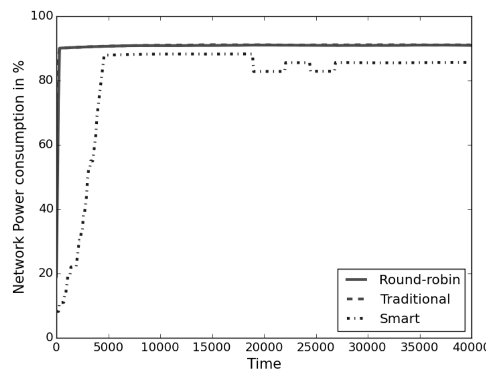

Fig. 8. Percentage of maximum power consumed for network resources

the number of the utilized servers is within the range of 320 - 420, representing an average increase of 260%, which is significant. Additionally, the total number of active servers and blades increases even further in the "Round-Robin" scenario, eventually approaching 700 blades. This reinforces our belief that schedulers should be redesigned in order to be able to properly serve a disaggregated data centre.

The lower number of active blades in the "Smart" scenario, is the result of the increase in the utilization of each blade. Therefore, in Figure 6, it is possible to infer that the proposed scheduler is capable of allocating almost triple and quadruple the number of cores per blade/server comparing to the schedulers in "Traditional" and "Round-Robin" scenarios respectively.

The actual CPU utilization, in terms of CPU cycles as is measured by monitoring tools, may vary according to the application that each VM is running. However, the more the assigned cores, the higher the CPU utilization would be. It should be noted that the percentage of the assigned cores per server/blade include the oversubscription ratio. That means that it is measured upon the number of virtual cores that each CPU blade/server can provide, which is 48 for "Smart" and "Round-Robin" scenarios and 36 for the "Traditional" scenario. Taking this into consideration, we see in Figure 6 that in the stable situation, the number of assigned cores is slightly above 50% of the total number of virtual cores. In that case, if all the VMs ask for the maximum of their assigned computing power, there would be a performance degradation and an SLA violation. However, the possibility is relatively low and the system operator, can change this behaviour by adjusting the weights of the scheduler or the oversubscription ratio.

The number of active devices and their utilization, directly reflect their energy consumption. We have performed an energy study for compute and network devices. Both types of resources have a similar behaviour regarding the energy consumption, which is in general linear and is reflected in Eq. 4 [18] [19]. The idle state energy consumption is usually around 75% for servers and 85-90% for electrical switches of the maximum energy that the device can consume and is represented with the variable a. The rest of the energy consumption depends on the utilization. We assume that the inactive devices, can be switched off.

$$E = (a + (1 - a) * utilization) * max\_power \qquad (4)$$

Figure 7, depicts the energy consumption comparison between the three scenarios and includes the computing resources, namely the total amount of blades or servers. The proposed scheduler managed the consumption of the disaggregated data centre to be only 30% of the energy that the "Traditional" scheduler consumed. Additionally, in Figure 8, the results show that in the "Smart" scenario, we managed to have some savings in the energy consumed in the network. What is more important though, is that the scheduler managed to adapt to the load variation and reduce the network energy consumption, when needed.

VII. Conclusion

In this paper we have demonstrated the first scheduling algorithm for a disaggregated data centre and the benefits of disaggregation. A disaggregated data centre architecture has been proposed in order to satisfy the high-capacity and low latency requirements of disaggregated computing resources. The simulation results show that the proposed scheduling algorithm can significantly improve resource utilization compared to state-of-the-art algorithms and thus reduce the energy consumption. Additionally, we have demonstrated that applying existing approaches for scheduling to a disaggregated data centre will have disastrous effects on the hardware utilization and its cost efficiency.

Acknowledgements

This work was supported by EPSRC grant EP/L020009/1: Towards Ultimate Convergence of All Networks (TOUCAN) and EPSRC grant EP/L026155/1: Introducing Insight into the Abstraction of Optical Network Infrastructures (INSIGHT).

References

[1] OIDA, " Photonics for Disaggregated Data Centers Workshop, OFC ," 2015.

[2] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, "Network support for resource disaggregation in next-generation datacenters," *HotNets-XII Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, pp. 1–7, 2013.

[3] R. Shi, J. Zhang, W. Chu, Q. Bao, X. Jin, C. Gong, Q. Zhu, C. Yu, and S. Rosenberg, "MDP and Machine Learning-Based Cost-Optimization of Dynamic Resource Allocation for Network Function Virtualization," *2015 IEEE International Conference on Services Computing* .

[4] SeaMicro, "Technology Overview," *http://www.seamicro.com/*.

[5] Intel, "Rack Scale Architecture," *http://www.intel.com/content/www/us/en/architecture-and-technology/intel-rack-scale-architecture.html*.

[6] HP, "Moonshot," *http://h20195.www2.hp.com/v2/getpdf.aspx/4AA4-6076ENW.pdf*.

[7] S. Novaković, A. Daglis, B. Falsafi, E. Bugnion, and B. Grot, "Scale-Out NUMA c," 2014.

[8] K. Asanović and D. Paerson, "FireBox : A Hardware Building Block for the 2020 WSC," *Proceedings of the 12th USENIX Conference on File and Storage Technologies (FAST 14) - Keynote presentation*, 2014.

[9] K. Lim, Y. Turner, J. R. Santos, A. AuYoung, J. Chang, P. Ranganathan, and T. F. Wenisch, "System-level implications of disaggregated memory," *HPCA '12 Proceedings of the 2012 IEEE 18th International Symposium on High-Performance Computer Architecture*, pp. 1–12, 2012.

[10] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree : Saving Energy in Data Center Networks," *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, pp. 17–17, 2010.

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

[12] O. Biran, A. Corradi, M. Fanelli, L. Foschini, A. Nus, D. Raz, and E. Silvera, "A stable network-aware VM placement for cloud systems," *Proceedings - 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012*, pp. 498–506, 2012.

[13] G. M. Saridis, E. H. Salas, Y. Yan, S. Yan, S. Poole, G. Zervas, and D. Simeonidou, "DORIOS : Demonstration of an All Optical Distributed CPU , Memory , Storage Intra DCN Interconnect," pp. 32–34, 2015.

[14] OpenStack, "https://www.openstack.org/."

[15] C. Peng, M. Kim, Z. Zhang, and H. Lei, "VDN: Virtual machine image distribution network for cloud data centers," *Proceedings - IEEE INFOCOM*, pp. 181–189, 2012.

[16] DigitalOcean, "https://www.digitalocean.com/."

[17] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," *Proceedings - 9th International Conference on Grid and Cloud Computing, GCC 2010*, pp. 87–92, 2010.

[18] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," *Future Generation Computer Systems*, vol. 28, pp. 755–768, 2012.

[19] P. Mahadevan, S. Banerjee, P. Sharma, a. Shah, and P. Ranganathan, "On energy efficiency for enterprise and data center networks," *Communications Magazine, IEEE*, vol. 49, pp. 94–100, 2011.