dReDBox: Materializing a Full-stack Rack-scale System Prototype of a Next-Generation Disaggregated Datacenter

M. Bielski[§] I. Syrigos[x] K. Katrinis[*] D. Syrivelis[*] A. Reale[*] D. Theodoropoulos[†] N. Alachiotis[†]
D. Pnevmatikatos[†] E.H. Pap[**] G. Zervas[‡] V. Mishra[‡] A. Saljoghei[‡] A. Rigo[§] J. Fernando Zazo[¶] S. Lopez-Buedo[¶]
M. Torrents[‖] F. Zyulkyarov[‖] M. Enrico[††] O. Gonzalez de Dios[‡‡]
[*]IBM Research, Ireland [†]FORTH, Greece [‡]University College London, UK
[§]Virtual Open Systems, France [¶]NAUDIT HPCN, Spain [‖]Barcelona Supercomputing Center, Spain
[**]SINTECS, Netherlands [††]HUBER+SUHNER Polatis, UK [‡‡]TELEFONICA, Spain [x]University of Thessaly, Greece

***Abstract*—Current datacenters are based on server machines, whose mainboard and hardware components form the baseline, monolithic building block that the rest of the system software, middleware and application stack are built upon. This leads to the following limitations: (a) resource proportionality of a multi-tray system is bounded by the basic building block (mainboard), (b) resource allocation to processes or virtual machines (VMs) is bounded by the available resources within the boundary of the mainboard, leading to spare resource fragmentation and inefficiencies, and (c) upgrades must be applied to each and every server even when only a specific component needs to be upgraded. The dRedBox project (Disaggregated Recursive Datacentre-in-a-Box) addresses the above limitations, and proposes the next generation, low-power, across form-factor datacenters, departing from the paradigm of the mainboard-as-a-unit and enabling the creation of function-block-as-a-unit. Hardware-level disaggregation and software-defined wiring of resources is supported by a full-fledged Type-1 hypervisor that can execute commodity virtual machines, which communicate over a low-latency and high-throughput software-defined optical network. To evaluate its novel approach, dRedBox will demonstrate application execution in the domains of network functions virtualization, infrastructure analytics, and real-time video surveillance.**

I. Introduction

Disaggregated computing aims to overcome the limitation of fixed resources in existing datacenter infrastructures. In principle, it considers the organization of resources into large homogeneous pools, such as compute, memory, and accelerator ones, and enables an on-demand, fine-grained assembly of computational platforms based on application requirements.

Recently published results [1] obtained through evaluation of an analytics workload on SparkSQL have shown that memory disaggregation can be feasible even with conventional 40Gbps interconnects. In a similar spirit, Gao et al. [2] evaluated using selectively emulation and simulation techniques the impact of memory disaggregation to a class of Big Data Analytics with benchmark-grade workloads. Custom designs for enabling dynamic scale-up to remote memory resources have shown the potential and challenges of the approach, either transparently to consuming applications [3], or by exposing remote memory via explicit programming models [4]. All of these previous efforts have evaluated the potential of memory disaggregation purely using simulation.

Further efforts have evaluated employing commercial of the shelf technologies to facilitate memory disaggregation, such as RDMA over Infiniband [5][6] and PCIe rack-scale switching [7]. Finally, research projects on next-generation datacenters consider the employment of hardware accelerators to improve system performance; for instance the Ecoscale project [8] proposes an architecture for automatic mapping and execution of HPC applications to platforms, supported by reconfigurable modules, whereas the Vineyard project [9] develops an integrated platform for heterogeneous accelerator-based servers.

Building upon the above knowledge, this paper reports on the progress of the dReDBox project [10][11] in materializing a full-stack prototype of a disaggregated datacenter at rack-scale, for the purpose of evaluating the value of disaggregation for service providers and full-fledged cloud applications at a high technology-readiness level. Towards this end, dReDBox proposes a customizable low-power architecture for next-generation datacenters, moving from the mainboard-as-a-unit paradigm to a flexible, software-defined block-as-a-unit one. Overall, the major project objectives are the following:

- Develop a vertical approach for a flexible modular datacenter-in-a-box architecture starting from the hardware platform.
- Deliver enhanced elasticity and improved process/virtual machine migration within the datacenter.
- Provide transparent access to remote memory with minimal latency over a scalable network architecture.
- Offer Type-1 full-fledged hypervisor functionality on top of the dReDBox platform, with innovative support for segmentation, and an appropriately revisited design of virtual memory ballooning subsystem for elastic distribution of disaggregated memory.
- Provide software-defined global memory and peripheral resource management.
- Offer fine-grained power management and aggressive power-aware resource management/scheduling.
- Decrease datacenter Total Cost of Ownership (TCO).

The remainder of this paper is organized as follows. Section II describes the dRedBox concept and its fundamental hardware building blocks, whereas section III describes the overall system network. Section IV discusses the system software layers, and section V presents the project pilot applications. Section VI discusses TCO analysis and compares conventional datacenter approaches against disaggregated platforms. Finally, section VII concludes the paper and states our next steps.

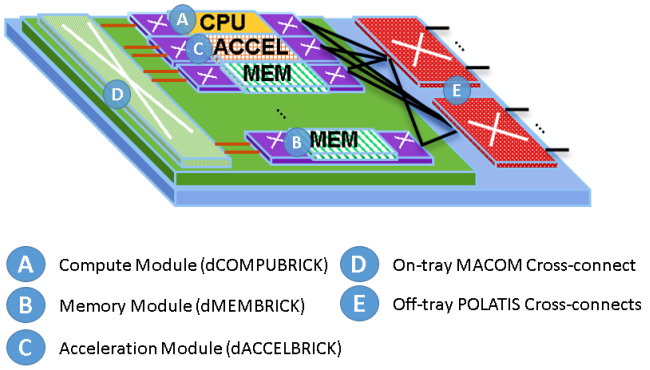

Fig. 1. The dReDBox disaggregated tray concept, comprising hot-pluggable modules that provide compute, memory and accelerator resources.

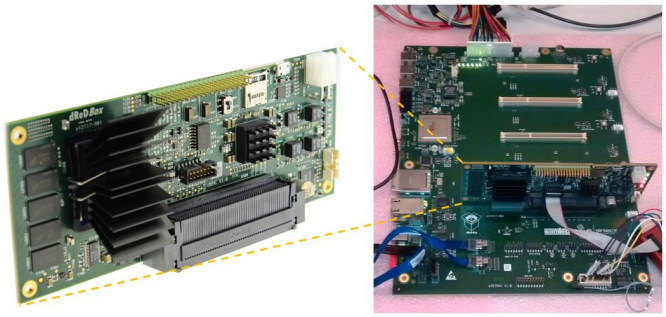

Fig. 2. (Left) Functional prototype of a "brick" used in the dReDBox vertical prototype; (Right) Functional downscaled testing prototype of the dReDBox tray featuring pluggable bricks and optical off-tray interconnection

II. dReDBox Disaggregated Datacenter Architecture

dReDBox pursues a customizable low-power datacenter architecture, based on a flexible, software-defined block-as-a-unit one. Figure 1 illustrates the tray-level pooling of resources, whereby a datacenter tray comprises hot-pluggable modules that provide three fundamental types of resources, namely compute, memory, and accelerators. dRedBox employs (a) micro-processor SoC modules (termed *compute bricks* or *dCOMPUBRICKs*), (b) high-performance RAM modules (termed *memory bricks* or *dMEMBRICKs*), and (c) accelerator (FPGA/SoC) platforms (termed *accelerator bricks* or *dACCELBRICKs)*, as the principal building blocks to form pooled, on-demand allocated hardware processing platforms. Figure 2 depicts the fully assembled prototyped board implementing a dReDBox brick module. Intra-tray bricks are connected over a low latency/high-throughput electrical circuit, whereas trays utilize optical networks for cross-tray, in-rack interconnection.

Figure 3 shows the block diagram of the dCOMPUBRICK architecture, based on the Xilinx Zynq Ultrascale+ MPSoC. The latter integrates a quad-core A53 ARM Application Processing Unit (APU) and a dual-core ARM Cortex R5 Real-time Processing Unit (RPU). Among others, this choice reduces the number of components (no separate FPGA chip needed), eventually leading to smaller block sizes and power consumption. Another benefit is their flexibility in terms of supporting various memory technologies (e.g., HMC).

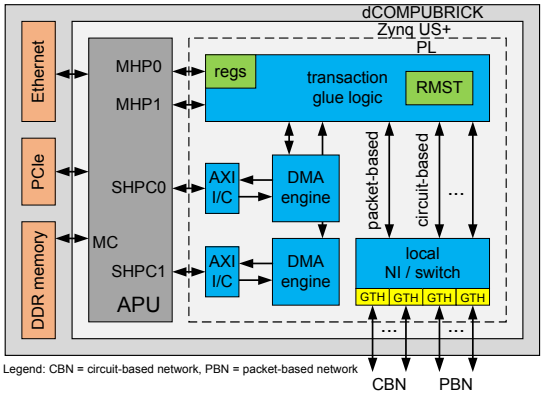

Fig. 3. A dCOMPUBRICK features a Xilinx Zynq Ultrascale+ MPSoC that integrates a quad-core ARMv8 Application Processing Unit (APU) for software execution.

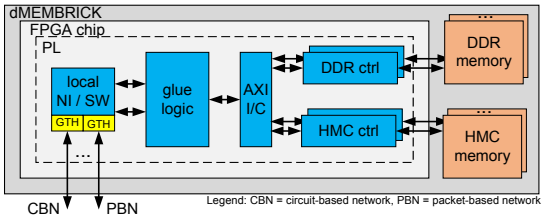

Fig. 4. dMEMBRICK architecture featuring the Xilinx Zynq Ultrascale+ MPSoC; a local switch forwards data to the memory brick glue logic to interface different memory module technologies.

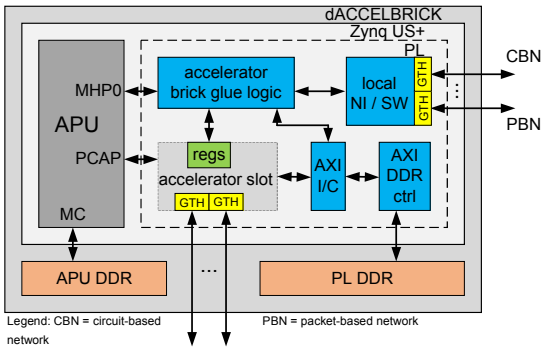

Fig. 5. The dACCELBRICK architecture based on a Xilinx Zynq Ultrascale+ MPSoC. It consists of dynamic and static infrastructure to host reconfigurable accelerators and external communication respectively.

A dCOMPUBRICK hosts local off-chip memory for low-latency and high-bandwidth instruction read and read/write data access, as well as Ethernet and PCIe ports for data and system communication and configuration, respectively. The dCOMPUBRICK APU can reach disaggregated resources, such as memory and accelerators, via a dReDBox-specific glue intellectual property (termed Transaction Glue Logic) on the data-path, and communication endpoints implemented on the MPSoC PL. The Remote Memory Segment Table (RMST) is a fully associative structure, whose entries identify large and contiguous portions of remote memory space hosted in dMEMBRICKs. The APU forwards remote memory requests to the TGL via its Master ports. The TGL identifies the remote memory segment that each transaction should access, and forwards it to the appropriate outgoing high-speed port, leading

to a circuit-switched path already set up via orchestration procedures. At an experimental level, the project is also exploring dCOMPUBRICK support for packet-level system/data interconnection, using Network Interface (NI) and a brick-level packet switch (also implemented on PL), on top of the inherent circuit-based interconnection substrate.

Figure 4 shows the dMEMBRICK architecture, which provides a large and flexible pool of memory resources that can be partitioned and (re)distributed among all processing nodes (and corresponding VMs) in the system. dMEMBRICKs can support multiple links. These links can be used to provide more aggregate bandwidth, or can be partitioned by orchestrator software and assigned to different dCOMPUBRICKs, depending on the resource allocation policy used. For ingress data, the glue logic forwards incoming transactions to the local memory controllers, whereas for egress data, it forwards transactions to the local switch for transmission back to the requesting dCOMPUBRICK.

A dMEMBRICK can be dimensioned in terms of memory size as well as the number of memory controllers it supports, so as to adapt to the size and bandwidth needs at the tray and system level. Moreover, the dMEMBRICK architecture is not limited to a specific memory technology, as long as this is supported by the transaction glue logic implementation. For example, the dMEMBRICK architecture can seamlessly support both DDR and HMC memory technologies; the glue logic is connected to an AXI interconnect, hence directly interfacing both Xilinx DDR and HMC controller IPs.

Figure 5 shows the dACCELBRICK architecture based on the Zynq Ultrascale+ MPSoC. dACCELBRICKs host accelerator modules for enhancing application execution based on a near-data processing scheme; instead of transmitting data to a remote dCOMPUBRICK, data are offloaded by remote dCOMPUBRICKs to dACCELBRICKs, thus improving performance and at the same time reducing network utilization.

The dACCELBRICK consists of dynamic and static infrastructure. The former consists of a predefined, reconfigurable slot within the PL that hosts hardware accelerators. An accelerator wrapper template integrates (a) a set of registers accessed by the glue logic for accelerator control and status monitoring, (b) a set of high-speed transceivers for direct communication with external resources, and (c) a local AXI DDR controller. The static infrastructure supports dynamic hardware reconfiguration, interfacing with the accelerators, and communication with remote dCOMPUBRICKs. To support hardware reconfiguration, the local APU executes a thin middleware responsible for (i) receiving and storing bitstreams from remote dCOMPUBRICKs, and (ii) reconfiguring the PL with the required hardware IP via the PCAP port.

III. Low-latency Memory Interconnection

The optical interconnect network that showcases the dReDBox architecture is shown in Figure 6. The resulting read/write memory requests and data transactions are sent to a dynamically controlled on-brick switch, whose role is to forward remote memory transactions (read/write requests and data) to the appropriate transceiver ports facing the circuit-switched optical interconnect, so that they reach the correct destination dBRICKs.

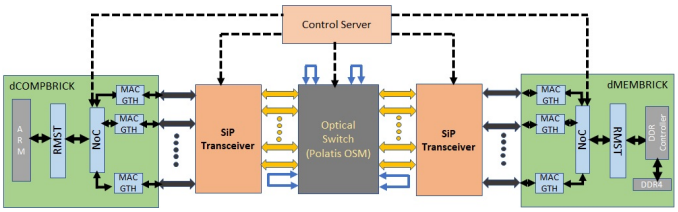

Fig. 6. Optical interconnection between dCOMPUBRICK and dMEMBRICK.

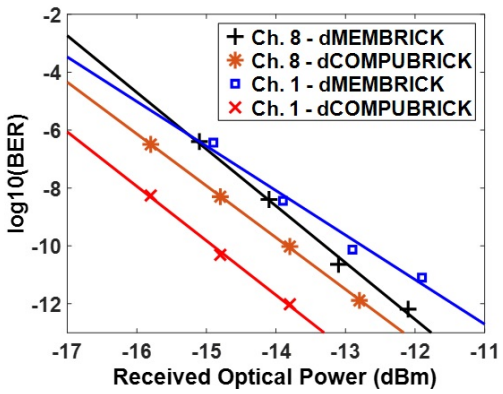

Fig. 7. BER Vs receiving optical power for both dBRICKS. dCOMPBRICK and dMEMBRICK are interconnected through ch-1 and ch-8.

Each of the physical incoming/outgoing ports on the dBRICKs is attached to a different channel on the multi-channel SiP Mid-board optics (MBO). The SiP MBO used has a total of 8 transceivers using external modulation and a shared laser operating at 1310 nm. Each channel on average has an optical output power of -3.7 dBm. The SiP MBO is connected to a low loss 48-port optical switch module provided by HUBER+SUHNER Polatis. Each hop through the optical switch module introduces approximately 1 dB of attentuation. The power consumption of this module is approximately 100 mW/port although the next generation of these switch modules is currently under development, doubling the optical port density and halving the per port power consumption. The dReDBox architecture requires a FEC-free optical interface between dBRICKs, as the presence of FEC can potentially introduce more than 100 ns of latency, which degrades the performance of a disaggregated system.

All bi-directional optical links between the dCOMPUBRICK and dMEMBRICK are able to achieve a bit error rate (BER) below 10^{-12} while all but one were traversing eight hops through the optical switch (with the remaining channel traversing six hops). The box plot in Figure 7 presents the bit error rate performance for two 10 Gb/s bi-directional optical links (channel 1 and channel 8) between the dCOMPUBRICK and dMEMBRICK, after traversing multiple hops through the optical switch as can be noted by the loss in received optical power. Our work is on-going to obtain similar evaluation results on higher throughput transceiver links.

In dReDBox, memory interconnection among modules occurs via electrical resp. optical circuit-switching, as a means of minimizing the critical KPI of remote access latency. Beyond this mainline approach, experimental work is put on explor-

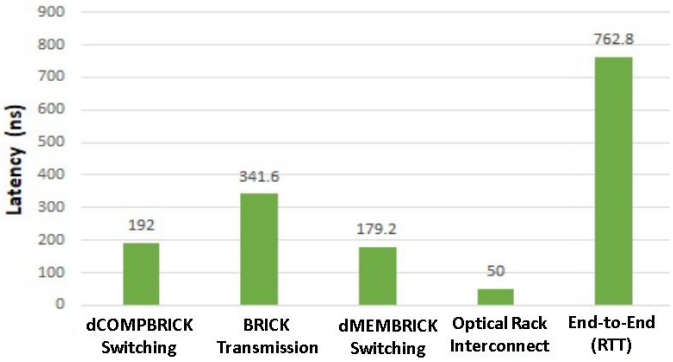

Fig. 8. Round-trip system latency breakdown for remote memory access.

ing packet-switching as a means of interconnecting pooled resources, particularly to cater for cases where the system is running low in terms of physical ports available to accomodate new circuits. In such a mode, dedicated switching and MAC/PHY blocks are used to forward memory transactions to on-brick destination ports as appropriate in a round-robin fashion. On the control-path, dedicated orchestration resources are required to make sure that packet-switch lookup-tables on dCOMPBRICKS/dMEMBRICKS are appropriately configured are runtime. Figure 8 shows a preliminary break down of (hardware-level) measured remote memory round-trip access latency using this exploratory approach. These latency results refer to contributions of the on-brick switch and the MAC/PHY blocks on both the dMEMBRICK and the dCOMPUBRICK, as well as the optical path propagation delay. Work is on-going on further optimizing IP designs to further decrease incurred latency.

IV. Disaggregation System Software

dRedBox features a fully-customized system software stack to facilitate disaggregation at all levels, as shown in Figure 9. The various components comprise a control plane that enables virtual machines and orchestration software to dynamically and safely request, attach and use remote memory on any given dCOMPUBRICK. An appropriately designed Scale-up API triggers the memory attachment process. The application notifies the Scaleup controller which in turn relays the request to the Software Defined Memory (SDM) Controller that manages the remote memory resources. Subsequently, the destination dCOMPUBRICK h/w glue logic is configured and the baremetal OS attaches remote memory and makes it available. Then control is handed back to the Scale-up controller which configures the hypervisor to dynamically expand the physical memory that it provides to the hosted VM. Below we provide a brief overview of the main components and the challenges that we have addressed.

A. Baremetal OS layer

A feature enabling memory resizing at OS level is called **memory hotplug**. As the name implies, the kernel attaches new physical page frames, by expanding the page table pool at runtime, after physical attachment process of remote memory is completed. We have implemented the memory hotplug linux kernel support for arm64 [12].

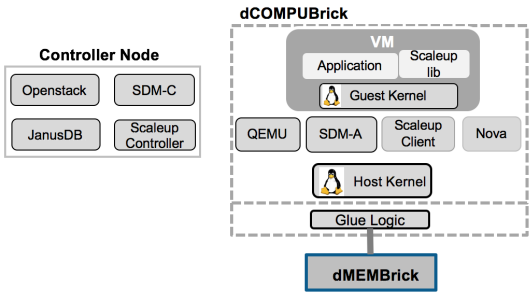

Fig. 9. High-level depiction of the dRedBox software stack

B. Virtualization layer

At the virtualization layer, we have developed appropriate memory hotplug support scheme for the QEMU hypervisor. The implementation adds new RAM DIMMs, at runtime, and makes them available to the guest OS. Subsequently, the guest kernel is leveraging the hotplug support that has been previously described for the baremetal kernel to use the remote memory. Scale up support is also implemented to enable applications that run within a VM to request the expansion of available system memory. In the future, the guest memory hotplug support will be enhanced to automatically protect the guest from running out-of-memory.

C. Orchestration layer

Orchestration of the disaggregated resources is performed by a software component integrated with OpenStack, namely the SDM Controller (SDM-C). The SDM-C runs as an autonomous service that primarily supports resource reservation and dynamic reconfiguration within a rack, by interacting with agents (SDM Agents) running on the OS of dCOMPUBRICKs, as well as with configurable switches to program circuit switches at runtime. The roles of this component are: a) to receive VM/bare-metal allocation requests from OpenStack b) safely inspect resource availability and make a power-consumption conscious selection of resources, c) safely reserve selected resources and d) generate all the necessary configurations and push them via appropriate interfaces to all involved devices.

In a preliminary evaluation setup, we have measured the competitiveness of the dReDBox software stack in terms of scale-up agility (delay in delivering dynamically scale-up memory to requesting VMs), when compared to conventional scale-out (i.e. spawning of additional VMs to facilitate memory addition to an application [13]). As shown in Figure 10, memory expansion agility is superior in the disaggregated approach, even under the most extreme scale-up concurrency conditions tested (number of VMs posting scale-up requests within a give time interval).

V. Pilot Applications

The concept of dReDBoX is aimed to be validated in three selected use cases that have the potential to significantly benefit either by improving their performance beyond current limitations, or by being able to share resources in a highly efficient manner.

The first application is a video analytics application that helps security organizations to carry out large investigations

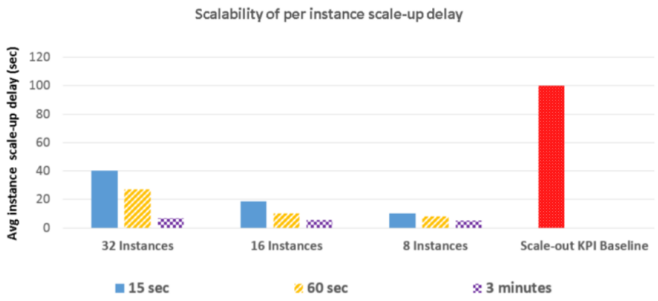

Fig. 10. Evaluation results of per VM average delay (in seconds) of dynamically scaling-up/down its memory resources (lower is better). The delay categories (depicted with different bar patterns/color) correspond to aggressiveness of scale-up concurrency among 32/16/8 VM instances respectively (lower is more aggressive), compared to elasticity through conventional VM scale-out

using video evidence and requires to search through thousands of video hours. In serious cases, including terrorist events, 100,000 hours of video or more may need to be reviewed quickly to find key intelligence. Video analytics algorithms are used to cut down this workload, but the computational requirements are event-driven and so cannot be scheduled or predicted. dRedBox provides new scalability features which greatly help to address this challenge.

The second use case aims at demonstrating the feasibility of dReDBox for Network Function Virtualization (NFV). In particular, the use case is aimed at edge computing and collaborative cryptography. In this use case, the server is split into two distinct entities, the edge server, which is the one that executes the edge computing and has direct contact with the user, and the key server that communicates over an encrypted and mutually authenticated channel (using for example TLS) and stores the private key. The load of NFV applications varies according to a daily traffic pattern, with a very low load at night and peaks during day hours. Given the sensibility of the information in the Key Server database, scale-out techniques should be avoided to replicate critical information and thus, elasticity in the memory usage provided by dRedBox can help to cope with the traffic peaks.

The third use case is related to Network Analytics at very high rates. Network analytics is a constantly-evolving field both in terms of resource demands and adaptability to more complex and innovative challenges. Such risks are natural and emerge with the revolutionary breakthroughs in the network infrastructures. Especially the decentralization of the servers with the cloud computing movement and the upgrade of conventional systems (standardization of 100GbE Ethernet links) turn the action of developing a flexible and adaptive monitoring probe to a real challenge. In the context of the dRedBox project and from the point of view of network monitoring, two clearly differentiated modes of operation are contemplated: **a) Online analysis** of the network: this concept refers to the mode of inspecting every single frame that travels across the physical link. A fast and accurate response must be offered without affecting concurrent workflows or the current provided service. Due to this restriction, the functionality is limited to the classification of packets as elements of interest (for further inspection) and the gathering of some basic metrics about the integrity and status of the network and **b) Offline analysis** of the network. Packets that were marked as relevant during the online analysis can be studied during a second stage

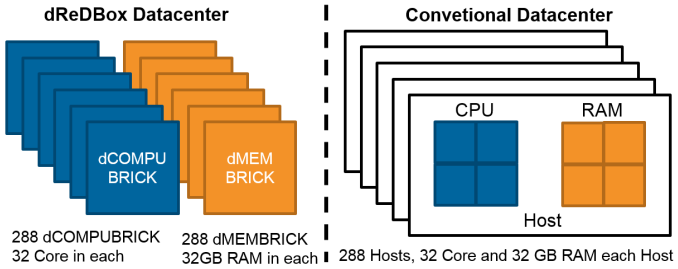

Fig. 11. dReDBox and a conventional datacenters both having the same compute and memory resources.

with a more exhaustive emphasis.

The dRedBox architecture offers two key concepts to the elevated value of the network-analytics use-case deployment in a datacenter environment: (a) Reconfigurable acceleration units (dACCELBRICKs) that can cope with the preprocessing of the incoming traffic and its dump for a future analysis, and (b) Dynamic access to the number of resources (dCOMPUBRICK, dMEMBRICK) that reduces the postponement of the offline analysis. The scheduling of this CPU-intensive tasks is not necessarily linked to a concrete physical node and could be run in one of the many dCOMPBRICK. With the dynamism of memory hotplugging, the process could be scaled down during peaks of memory intensive loads in the datacenter but, with the main improvement of being continuously executed. The second factor is critical for the performance and key evaluation of the system. The more responsiveness of the analysis tool, the faster a solution is offered to the user without a detriment of the QoS.

VI. TCO Value Proposition Case Study

One of the primary value propositions of disaggregated computing is in improving Total Cost of Ownership to datacenter/cloud service providers. This section presents a comprehensive study of the improvement to TCO that can be brought by a dReDBox-like datacenter. This first study focuses on evaluating the TCO savings in terms of the energy that can be saved by powering off unutilized resources.

In a conventional data center memory and CPU allocation depend on the availability of each on a given node. In a node of a conventional data center, when all CPUs are utilized, it will not be possible to allocate more memory and vice versa. Instead in dReDBox like datacenter each resource can be allocated independently and because of that we intuitively expect that fragmentation would be lower and the utilization higher. To quantify how much the dReDBox architecture can decrease TCO expressed through better utilization, we compare a dReDBox-like datacenter to a conventional datacenter built off commercial-off-the-shelf computer systems with compute and memory resources coupled on a single main board.

The TCO of the two types of datacenters is evaluated through simulation. The simulation uses a First Come First Served (FCFS) policy to schedule a given workload of virtual machines (VMs) with different requirements to each of the two datacenter types. Then it evaluates the number of unutilized individually powered units that can be powered off (i.e. ”bricks” in the dReDBox datacenter case and server nodes in the conventional datacenter case respectively). To deliver a fair comparison, in all experiments we consider that each

Configuration	vCPUs	RAM
Random	1-32 cores	1-32 GB
High RAM	1-8 cores	24-32 GB
High CPU	24-32 cores	1-8 GB
Half Half	16 cores	16 GB
More Ram	1-6 cores	17-32 GB
More CPU	17-32 cores	1-16 GB

TABLE I. VM WORKLOADS WITH DIFFERENT TYPES OF RESOURCE REQUIREMENTS USED FOR THE TCO STUDIES.

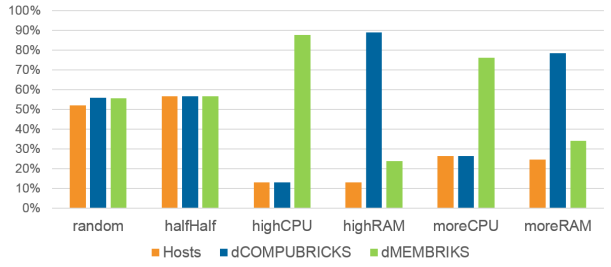

Fig. 12. Percentage of unutilized resources that can be powered off.

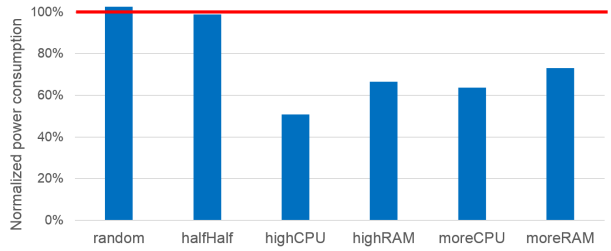

Fig. 13. Estimation of power consumption. Numbers are normalized to a conventional datacenter.

datacenter has the same aggregate amount of compute and memory resources as shown in Figure 11.

Table I summarizes the different types of VM workloads and their resource requirements we tested. Figure 12 depicts the percentage of resources that can be powered off to conserve energy in each of the two datacenter types. Our results suggest that the resource fragmentation in a dReDBox-like datacenter is significantly lower in scenarios where VMs have unbalanced compute and memory requirements (e.g., high memory vs. low compute). This is because resources in dReDBox can be better utilized by scheduling the VMs on dBRICKs which are already running a VM and the remaining dBRICKs where no VM is scheduled can be powered off. Depending on the different VM configurations in dReDBox, up to 88% of dMEMBRICKs or dCOMPUBRICKs can be powered off because they are not utilized, whereas in a conventional datacenter only 15% of the hosts can be powered off

The opportunity to power down resources may translate into almost 50% energy savings depending on the workload (Figure 13). Such levels of power savings can be achieved when the VM workloads have diverse and unbalanced resource requirements. The diverse resource requirements are usually the types of workloads being executed in large datacenters. Besides the energy savings obtained through powering off unutilized resources during the operation of a datacenter, the dReDBox architecture has other advantages that contribute to lowering the TCO. For example, the modularity and interchangeability of the dBRICKs plays a significant role in lowering the price of the procurement, as well in delivering technology refreshes at the component level instead of the server level. This study does not consider how these aspects and advantages of the dReDBox architecture affect the TCO; the latter is targeted by our on-going work.

VII. CONCLUSIONS

dRedBox addresses the limitations caused by the static resource proportionality of multi-tray datacenter systems. It proposes the organization of resources into compute, memory, and accelerator pools, to allow on-demand, fine-grained assembly of computational platforms. All the above is being materialized on a vertical rack-scale prototype to quantify the value of disaggregation, in terms of improved cost and performance, power consumption, and ultimately reduced datacenter TCO.

VIII. ACKNOWLEDGEMENTS

This work has been supported in part by EU H2020 ICT project dRedBox, contract #687632.

REFERENCES

[1] P. S. Rao and G. Porter, "Is memory disaggregation feasible? a case study with spark sql," in *2016 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, March 2016, pp. 75–80.

[2] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, "Network requirements for resource disaggregation," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)*. Berkeley, CA, USA: USENIX Association, 2016, pp. 249–264.

[3] K. Lim, J. Chang, T. Mudge, P. Ranganathan, S. K. Reinhardt, and T. F. Wenisch, "Disaggregated memory for expansion and sharing in blade servers," in *Proceedings of the 36th Annual International Symposium on Computer Architecture (ISCA)*. New York, NY, USA: ACM, 2009, pp. 267–278.

[4] S. Novakovic, A. Daglis, E. Bugnion, B. Falsafi, and B. Grot, "Scale-out numa," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*. New York, NY, USA: ACM, 2014, pp. 3–18.

[5] S. Liang, R. Noronha, and D. K. Panda, "Swapping to remote memory over infiniband: An approach using a high performance network block device," in *2005 IEEE International Conference on Cluster Computing*, Sept 2005, pp. 1–10.

[6] J. Gu, Y. Lee, Y. Zhang, M. Chowdhury, and K. G. Shin, "Efficient memory disaggregation with infiniswap," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. Boston, MA: USENIX Association, 2017, pp. 649–667.

[7] C.-C. Tu, C.-t. Lee, and T.-c. Chiueh, "Marlin: A memory-based rack area network," in *Proceedings of the Tenth ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*. New York, NY, USA: ACM, 2014, pp. 125–136.

[8] I. Mavroidis, et.al., "Ecoscale: Reconfigurable computing and runtime system for future exascale systems," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 696–701.

[9] C. Kachris, et. al., "The VINEYARD approach: Versatile, Integrated, Accelerator-based, Heterogeneous Data Centres," in *International Symposium on Applied Reconfigurable Computing (ARC 2016)*, 2016, pp. 3–13.

[10] K. Katrinis, et. al, "Rack-scale disaggregated cloud data centers: The dReDBox project vision," in *2016 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2016, pp. 690–695.

[11] D. Syrivelis, A. Reale, K. Katrinis, and et. al, "A software-defined architecture and prototype for disaggregated memory rack scale systems," in *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, 2017.

[12] Maciej Bielski, Andrea Reale, "Hotplug for Arm64," https://lkml.org/lkml/2016/11/17/49, [Online; LKML.org].

[13] M. Mao and M. Humphrey, "A performance study on the vm startup time in the cloud," in *2012 IEEE Fifth International Conference on Cloud Computing (CLOUD)*, June 2012, pp. 423–430.