# Disaggregated Compute, Memory and Network Systems: A New Era for Optical Data Centre Architectures

**G. Zervas[1], F. Jiang[2], Q. Chen[1,2], V. Mishra[1], H. Yuan[1], K. Katrinis[3], D. Syrivelis[4], A.Reale[3], D. Pnevmatikatos[5], M. Enrico[6], N. Parsons[6]**

*[1]University College London, United Kingdom, [2]University of Bristol, [3]IBM Research – Ireland, [4]UTH, [4]FORTH-ICS, [5]Huber-Suhner Polatis*

*g.zervas@ucl.ac.uk*

**Abstract:** The disaggregated dRedBox Data Centre architecture is proposed that enables dynamic allocation of pooled compute and memory resources. An orchestration platform is described and algorithms are simulated that demonstrate the efficient utilization of IT infrastructure.

**OCIS codes:** (060.4258) Networks, network topology, (200.4650) Optical interconnects, (200.6715) Switching

## 1. Introduction

Computer architectures have been based historically on a server-centric approach with fixed amounts of processor and directly attached memory resources within the boundary of a mainboard tray. Current Data Centres follow this model, but have to support highly diverse workloads ranging up to 4-orders of magnitude on memory/CPU demand to CPU usage [1]. The mismatch between fixed proportionalities and diverse set of workloads leads to substantially under-utilized resources (often at only 40%) that account for 85% of the total Data Centre cost [2]. Server-centric Data Centres use overlay networks, with various protocols and optimization goals, e.g. InfiniBand for low-latency, FibreChannel for Storage Area Networks. To consolidate I/O and switching infrastructure, cost and power and to increase network flexibility a reconfigurable network functions virtualization (NFV) system has been designed and implemented in [3] as a protocol-independent programmable switch and languages [4] can simplify development.

The vision of disaggregation is to depart from the traditional paradigm of the mainboard-as-a-unit (server-centric model) and enable the creation of "function block-as-a-unit" (resource-centric model) having a baseline disaggregated pool of components including a) compute, b) memory c) storage d) network and e) accelarators. The result is a new type of computing system that is network-centric and can offer immense flexibility that can potentially maximize resource utilization while enabling new workflows and applications with few resource boundaries. However, a number of fundamental challenges arise on such communication-centric computer architectures: a) latency overheads, compared to current direct-attached model, should be minimized, b) system should support substantially higher bandwidth and bandwidth density at very low cost and power consumption, c) network system should offer specific performance and services according to communication type (e.g., compute-to-memory, compute-to-storage) on same substrate for maximum flexibility and d) orchestration of compute, memory and network resources should maximize resource utilization and workload performance at minimum cost.

This paper describes the dReDBox (disaggregated Recursive Datacentre-in-a-Box) architecture [5] that aims to address the points listed in the previous paragraph. It is designed to a) bound compute-to-memory latency to few 100s of nanoseconds, b) deliver 400 Gb/s capacity per compute/memory chipset, c) scale-out to multiple Racks, d) offer accelerated protocol/function programmable ports on each system and e) deliver maximum resource utilization. Simulation results demonstrate some of the aspects of the work.

## 2. Disaggregated Rack-Scale Computer Architecture

The dRedBox architecture as shown on Figure 1 consists of dRACKs (disaggregated Racks) housing multiple interconnected dBOXes. Each dBOX hosts pluggable arbitrary combinations of compute/memory/accelerator bricks, an electronic cross-point circuit switch for intra dBOX connectivity and a set of optical switches for intra and inter dBOX networking. Each 2U rack mounted dBOX will support up to 16 bricks. Each brick will either support general-purpose processing (dCOMPUBRICK) or random-access memory (dMEMBRICK) or application-specific accelerator (dACCELBRICK). All bricks are interconnected to all other bricks on the same box by means of the electronic L1 crosspoint circuit switch and the optical circuit switch [6] (a miniaturized beam-steering switch). Communication between dBRICKs on different dBOXes is strictly via optical circuit switching. Crucially each brick apart from its main information technology purpose (compute/memory/acceleration) uses reconfigurable System on Chip to perform networking functions beyond interfacing as traditional network interface cards do. The brick can embed and support forwarding, switching, and aggregation at either packet or circuit level [3]. It can deliver protocol independent programmable ports to support protocols and functions that can best suit the type of communication (i.e. compute-to-memory, compute-to-end user, etc) required. To minimize footprint and power consumption while maximizing bandwidth-density each of the bricks will make use of on-board 200 Gb/s Silicon

Photonic (SiP) single-mode 1.3 μm transceivers. SiP transceivers and beam-steering switches allow for a transparent brick-to-brick multi-hop network with minimum possible latency. The combination of transparent switches with protocol programmable system on chip allows for a function and topology programmable architecture.

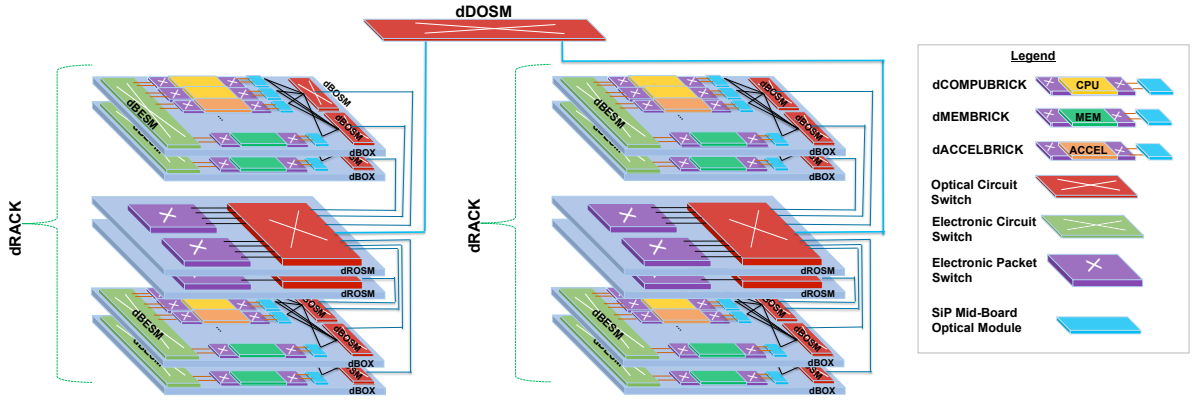


**Figure 1 dRedBox rack-scale architecture interconnected with hybrid optical and electrical switching.**

## 3. Compute, Memory and Network Orchestration

One main goal of disaggregation is to maximize resource utilization. An orchestrator is needed to first pool together, allocate and interconnect a set of resources (virtual CPUs, virtual memory) to compose a virtual compute entity able to satisfy the Virtual Machine (VM) or virtual tenant request triggered by a user. The orchestrator has to determine the set of compute bricks and memory bricks to reserve and interconnect before the system software can start a VM. Also, a running VM can dynamically shrink or expand its allocated memory using appropriate system support that enables the physical and logical attachment of memory regions at runtime.

A simulation platform has been developed to perform coordinated orchestration and allocation of IT resources together with reservation of their network bandwidth and interconnection. It investigates the importance of locality (placement of different bricks types in the same or different dBOXes or dRACKs) while offering resources with bounded latency. The parameters and values assumed are shown in Table 1. We have 1/3 ratio of compute, memory and storage resources across the whole multi-Rack system. In the top bar chart and middle plot of Figure 3 we consider dynamically generated VM requests following a Poisson distribution with an average inter-arrival time of 10 time units. Each request has a holding time that varies from 6300 to 9540 time units and contains the information of CPU core number, RAM size, storage size and CPU-RAM latency (0.3-0.6 μsec). Also the requests specify a fixed bandwidth of 5 Gb/s/unit for CPU to RAM and 1 Gb/s/unit for RAM to storage communication. For the bottom bar chart in Figure 3 we have another six different request types. The network topology reflects a double spine-leaf between Bricks, and optical switches (dBOSMs and dROSMs as per Figure 1) while Racks are fully-connected. We assume 8 channels per Brick each at 25 Gb/s reflecting 112 Tb/s maximum-offered capacity to the network.

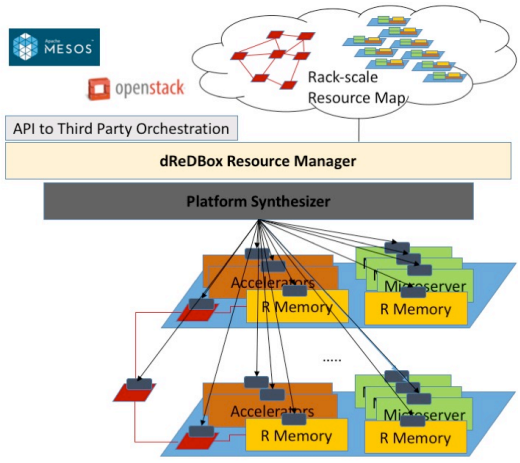


**Figure 2 Orchestration platform architecture.**

The algorithms developed and benchmarked include a) first fit (FF), b) best fit (BF), c) network unaware locality based (NULB) and d) network-aware locality based (NALB). The FF algorithm is the simplest since it allocates the first available IT resource types independently and without considering network availability. The BF is similar to the FF however it selects the best possible combination of IT resource types by scanning separately for them yet doesn't consider network resource status. The NULB is using the breadth-first search (BFS) algorithm to find IT resource types that are neighboring each other making it locality aware that is particularly useful for requests that have low-latency constraint. It starts scanning for the IT resource on bricks of a specific IT resource type that has the highest contention ratio (ratio between amount of resource type (i.e. 4 CPUs) required over total available) and looks for the other resource types neighboring this brick. NALB is similar to NULB however it uses a modified BFS algorithm that not only allocates IT resources that are neighboring each other but also considers network availability.

**Table 1 Simulation architectural resource assumptions and VM resource request profiles.**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **dRACKs** | 12 | dBOXes per dRACK | 6 | dBRICKs per dBOX | 8 | Units per dBRICK | 16 |
| CPU unit | 4 cores | RAM unit | 4 GB | STO unit | 64 GB | I/Os per dBRICK @ 25Gb/s | 8 |
| **Request Type** | (a) Random | (b)High RAM | (c) High CPU | (d) half and half | (e) More RAM | (f) More CPU | |
| CPU | 1-32 cores, | 1-8 cores | 24-32 cores | Type a (50%) | 1-16 cores | 17-32 cores | |
| RAM | 1-32 GB | 24-32 GB | 1-8 GB | Type b (50%) | 17-32 GB | 1-16 GB | |

We consider three type of dRACKs; a) Homogeneous dRACK and homogenous dBOXes (Type 1) meaning each Rack can only host one type of dBRICK (i.e. CPU), b) Heterogenous dRACK with homogenous dBOXes (Type 2) where dRACK can support multiple types of dBRICKs but only one type per dBOX and c) Heteregenous dRACK with heterogenous dBOX (Type 3) where each dBOX can host any type of dBRICK. The top plot in Figure 3 reflects a) the poor performance of the FF algorithm that deems it totally in-appropriate for any type of dRACK. The Type 1 dRACK delivers the highest blocking probability caused predominately by network congestion since all connections requests are between dRACKs that trigger increased network usage and high latency. The BF algorithm performs substantially better than FF in terms of blocking probability yet doesn't succeed on maximizing CPU utilization for minimum network resource usage as clearly reflected in middle plot in Figure 3. NULB and NALB offer almost identical and very low blocking probability. Both offer over 99% CPU utilization – assuming allocations abide by equivalent SLA policies – however on Type 2 dRACK, the cause of blocking on NULB is network bandwidth whereas for NALB is CPU unavailability. To clarify the advantage of NALB against NULB and the Type 3 dRACK against Type 2 dRACK we can only see that the combination of NALB and Type 3 require the least network resources (see Figure 3 middle plot) to achieve the same IT resource utilization. This keeps the added network cost and complexity to the minimum. To identify the benefits of disaggregation we benchmarked NALB using Type 3 dRACK against the traditional Server-centric architecture with equal total resources that consists of Servers each with 32 CPU Cores and 32 GB of RAM. All of the request types indicated above are within the upper limit of a single server. It is evident from Figure 3 bottom that the type of VM request substantially influence the results though disaggregation offers increased IT resource utilization on all cases scaling up to 30% improvement when have the highest diversity of requests (half and half case). However, we need to keep in mind that additional network resources (10-22%) are also required that increase the cost of the system.

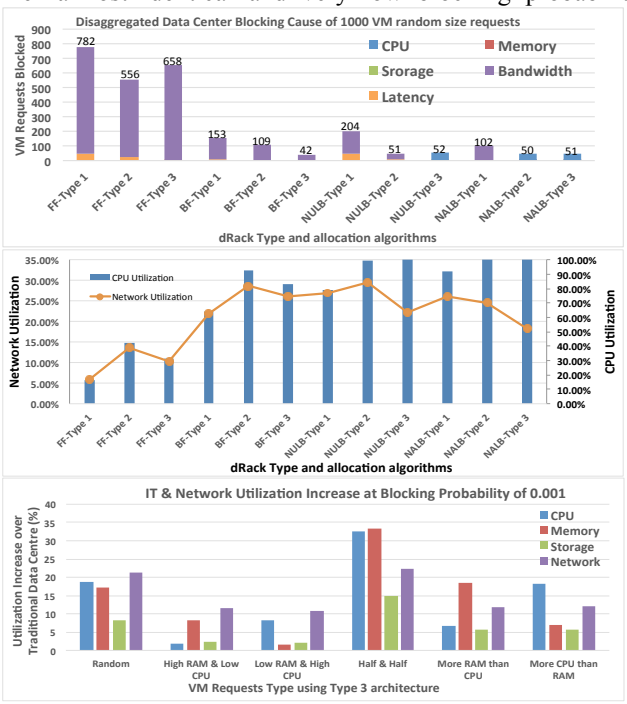


**Figure 3 VM blocking probability on dRACK types using four allocation algorithms (top), CPU and network utilization (middle) and resource utilization increase against standard server-centric Data Centres (bottom).**

## 4. Conclusions

The paper reports on a disaggregated Data Centre architecture that can orchestrate resources to achieve maximum IT utilization. It is architected to use on-board electrical network as well as on board SiP integrated transceivers (per dBRICK) together with miniaturized optical switches to scale-up and scale-out the system. The results of modeling of different algorithms for the placement and interconnection of compute and memory resources within the dReDBox architecture are described.

**5. Acknowledgements:** This work is supported by the EC H2020 dReDBox project with grant agreement 687632.

## 6. References

[1] S. Han et al., "Network support for resource disaggregation in next-generation datacenters," ACM Hot Topics in Networks, 2013
[2] Urs Hölzle, Luiz André Barroso, "The Case for Energy-Proportional Computing", Computer, vol. 40, Iss. 12, pp. 33-37, December 2007
[3] Q. Chen, et al., "Reconfigurable Computing for Network Function Virtualization: A Protocol Independent Switch", IEEE ReConFig 2016
[4] P Bosshart, et al., "P4: Programming Protocol-Independent Packet Processors", ACM, Vol 44, Iss. 3, SIGCOMM 2014
[5] K. Katrinis, et al., "Rack-scale Disaggregated cloud data centers: The dReDBox project vision", DATE, March 2016
[6] N. Parsons et al., "High Radix All-Optical Switches for Software-Defined Datacentre Networks", Proc. ECOC2016, paper W2F.1