Rack-scale Disaggregated cloud data centers: The dReDBox project vision

K. Katrinis*, D. Syrivelis†, D. Pnevmatikatos‡, G. Zervas§, D. Theodoropoulos‡, I. Koutsopoulos¶, K. Hasharoni‖, D. Raho**, C. Pinto**, F. Espina††, S. Lopez-Buedo††, Q. Chen§, M. Nemirovsky‡‡, D. Roca‡‡, H. Klos^x, T. Berends^x

*IBM Research - Ireland

† University of Thessaly

‡ FORTH-ICS & Technical University of Crete

§ HPN group, University of Bristol, UK

¶ Athens University of Economics and Business

‖ Compass-EOS

** Virtual Open Systems

†† NAUDIT HPCN

‡‡ Barcelona Supercomputing Center

^{x} SINTECS

Abstract—

For quite some time now, computing systems servers, whether low-power or high-end ones designs are created around a common design principle: the main-board and its hardware components form a baseline, monolithic building block that the rest of the hardware/software stack design builds upon. This proportionality of compute/memory/network/storage resources is fixed during design time and remains static throughout machine lifetime, with known ramifications in terms of low system resource utilization, costly upgrade cycles and degraded energy proportionality. dReDBox takes on the challenge of revolutionizing the low-power computing market by breaking server boundaries through materialization of the concept of disaggregation.

Besides proposing a highly modular software-defined architecture for the next generation datacentre, dRedBox will specify, design and prototype a novel hardware architecture where SoC-based microservers, memory modules and accelerators, will be placed in separated modular server trays interconnected via a high-speed, low-latency opto-electronic system fabric, and be allocated in arbitrary sets, as driven by fit-for-purpose resource/power management software. These blocks will employ state-of-the-art low-power components and be amenable to deployment in various integration form factors and target scenarios. dRedBox aims to deliver a full-fledged, vertically integrated datacentre-in-a-box prototype to showcase the superiority of disaggregation in terms of scalability, efficiency, reliability, performance and energy reduction which will be demonstrated in three pilot use-cases.

I. Introduction

The immense need for high-performing, parallel or distributed and heterogeneous computing is today cutting across industries: from close-to-the-sensor FPGA-based video analysis in national security applications, to chassis/desktop-sized GPU machines for fast sequencing in the health sector, to massively parallel supercomputers for advanced industrial manufacturing. Part of these industries have been traditionally employing embedded computing, primarily for achieving low-power, cost reduction - thanks to fit-for-purpose designs - and for abiding by stringent real-time constraints.

The increasing complexity of problems solved by traditional users of embedded computing has led to a significant ramp up of the computational power and integration plurality of such designs. It has enabled the reduction of component development and maintenance costs through repurposing of software/hardware used on conventional server systems. The latter trend, in conjunction with the explosion of the penetration of mobile into the consumer and enterprise market, is constantly pushing for higher performance low-power systems-on-a-chip (SoC). Such designs are being now capable of addressing use-cases that were once monopolized by the high-end server markets. SoC systems are already hugely outperforming traditional ones, across key performance indicators (KPIs) such as compute density, granularity of resource allocation, power consumption and virtual machine migration times. These KPIs challenge the way we need to design the next generation computing systems to guarantee environmental/societal and business sustainability.

High-end embedded computing systems, from small form-factor boards to multi-chassis systems, exhibit superior features compared to conventional high-end server designs (i.e. dual or quad-socket SMPs). Both technological trends are based on a common design axiom: *the mainboard and its hardware components form the baseline, monolithic building block that the rest of the system software, middleware and application stack build upon*. In particular, the proportionality of resources (processor cores, memory capacity and network throughput) within the boundary of the mainboard tray is fixed

978-3-9815370-6-2/DATE16/©2016 EDAA

during design time, as driven by the number of e.g. processor sockets and memory slots manufactured on a mainboard.

dRedBox proposes a customizable low-power datacenter architecture, moving from the paradigm of mainboard-as-a-unit to a flexible, software-defined block-as-a-unit. This approach allows an optimization of both performance and energy consumption. The baseline disaggregated building blocks to enable the on-demand hardware are: a) micro-processor SoC module, b) high-performance RAM module and c) accelerator (FPGA/SoC) module. Disaggregating components at that level significantly improves efficiency, increases resource utilisation and has the potential to revolutionize the way the datacenters are being built.

The rest of the paper is structured as follows: Section II motivates the need for further datacenter component disaggregation and outlines the dReDBox approach. Section III presents a high-level view of the vertical dReDBox architecture and discusses all challenges that will be taken on during the project. Last, section IV elevates the objectives of the project towards impact to vertical markets, stakeholders and users.

II. Motivation and Approach

There are three inevitable limitations that are side-effects of the way we build datacenters today:

1. Resource proportionality of the entire system follows the proportionality of the basic building block (mainboard), both at initial procurement time and during upgrade cycles. For instance, the decision of doubling the memory capacity in an operational system carries with it the parasitic capital and operational cost of procuring the rest of all additional but not necessary components that come with the upgrade server mainboard(s).

2. The allocation of resources to processes or virtual machines is upper bounded by the resources available within the boundary of the mainboard, leading to spare resource fragmentation and inefficiencies. For instance, if a CPU-bound application saturating 100% of processor cores uses only 40% of server memory, then no further workloads can be deployed on that server (due to lack of processing resources) and thus 60% of server memory is not usable.

3. Technology upgrades have to be carried out on each and every server board even when only a specific component needs to be upgraded (e.g. upgrading processor family) [1].

While the significant impact to Total Cost of Ownership (TCO) caused by point 3 above is straightforward, following two widely observed inflection points turn 1 and 2 into grave roadblocks to sustaining high-end computing efficiencies:

- Current and trending data-intensive workloads require a system-wide ratio of compute to memory/storage resources that is often disproportional to the fixed proportionality of the mainboard tray. In [2] a 4-order of magnitude range on memory/CPU demand to CPU usage is clearly indicated for a Google datacenter. In the same spirit, the RAMCloud project outlines that strategic use of (permanent) DRAM instead of storage significantly improves the execution time of specific, widely used, cloud applications. To achieve this, the RAMCloud software architecture aggregates memory modules that are located on different traditional mainboards, using a fast network and appropriate software, wasting this way processor resources and power [3]. Integrated CPU/memory/storage mainboard architectures can inherently create resource inefficiencies, both in terms of upgrades and workload-proportional system utilization. Delivering on this requirement with whole mainboard upgrades and/or server-bounded resource allocation coupled with process/VM migration is a bandage solution and a guaranteed sub-optimal investment of capitalization and operational expenses.
- Dynamic CPU, memory and accelerator scaling at runtime in response to dynamically changing service and application needs (elasticity) is suboptimal, for it is bounded by what is available on the mainboard tray. If more memory or CPU resources are required, VM migration to another tray is undertaken, incurring overhead and performance degradation.

The described limitations have been adequately addressed in modern datacentres at the peripheral level e.g. by Network Attached Storage (NAS) for persistent storage and PCIe off-board switches for network media. dReDBox aims at delivering memory disaggregation at the hardware integration level by interfacing the CPU chip memory controller (typically the Double Data Rate controller) with remote memory modules - located either on the same or a remote mainboard tray - using novel optical interconnection technology. For the most aggressive, widely used memory technology - namely DRAM - current state-of-the art DDR-to-DIMM interconnection is a tightly coupled parallel interface that can achieve a theoretical speed of $1800 million$ transfers per second, and latencies close to $10ns$.

The dReDBox architecture aims to approach these performance levels, while facilitating disaggregation, by employing a novel scalable optical network interconnecting memory controllers and memory modules in a datacentre configuration, offering multi-Tbps-level switch bisection, software-defined control capability and a minimum deterministic network (switch terminal I/O to switch terminal I/O) latency.

Hardware-level disaggregation and the software-defined wiring of resources undertaken by dReDBox will be matched with novel, required innovation on the system-software side. In particular:

- Delivering novel hypervisor distributed support to share resources that will allow the disaggregated architecture to bootstrap a full-fledged Type-1 hypervisor and execute commodity virtual machines. Taking advantage of the Type-1 hypervisor device driver techniques, the disaggregated memory support shall be further used for peripheral disaggregation with proper forwarding of Direct Memory Access (DMA) interrupts.
- Employing deep software-defined control of all resources at the hardware programmability level, including allocation of memory resources to micro-servers and software-defined network control. This hardware-orchestration

software, running off the data-path resources, is to be interfaced via appropriate Application Programming Interfaces (APIs) with higher-level resource provisioning, management and scheduling systems, notably cloud management (e.g. Openstack) and cluster management systems (e.g. Apache Mesos).

- Reducing the power consumption, which is a key parameter for dReDBox and will be attacked at all layers. At the platform level, power consumption of hardware platform components will play a major role in their selection. The optical network will utilize key technologies that significantly reduce power and improve latency. The hardware platform will provide a per component IPMIv2 interface that will allow the hypervisor and the orchestration tools to extensively control component mode of operation and also completely switch them off when not used. Note that dReDBox architecture will feature a central reservation system and will fully control the component interconnect, so in any given point in time it knows which components are in use. In this context, novel online myopic policies will be designed that can take instant decisions to migrate VMs and switch off resources. All approaches will be formulated as optimization problems that will be theoretically proven. The described efforts to reduce the power budget aim at improving power consumption of dReDBox platform by $10x$ compared to state-of-the-art.

The dReDBox platform will be embodied during the project through a vertically integrated prototype of working totalities for all the aforementioned subsystems, starting from small factor mainboard tray prototypes that can accommodate any combination of processor cores, memory and peripherals, all interconnected with an integrated modular, scalable and ultra-low latency optical network. The advantages of the dReDBox architecture and implementation will be quantified and demonstrated for proof-of-value to the end-user of the targeted systems. For this, dReDBox will demonstrate the value of the approach in terms of significant improvements in power consumption efficiency and elasticity, agility in resource allocation and reduction of data and binary migration overhead, using three real-life applications: a) Network Functions Virtualization (Telecom sector), b) Infrastructure Analytics (IT/Cloud sector) c) Real-time Video Surveillance (Security sector). These use-cases have been strategically selected as typical representatives spearheading the evolution of workloads needs increasingly prioritising for more parallel/distributed processing, being IO-/memory intensive and memory/storage capacity hungry and exhibiting temporal variability in utilisation of resources (hot-spot effects).

To provide an example, for the infrastructure analytics application network monitoring is considered. This is a difficult problem and currently it is very hard to scale efficiently in order to deal with the current speeds of corporate backbones (from 10 to 100 Gbps) [4]. Typically, packet processing involves a series of costly operations, which are expensive both in computational and memory access terms and, moreover, their cost depends heavily on the incoming traffic rate. If the latter is suddenly increased (e.g due to a DoS attack), the analytics system resource requirements scale in an instant. With conventional systems today, the analytics platform struggles to take maximum advantage of the motherboard resources that it was already running on, because migrating to other boards with more resources has the prohibitive cost of data movement. This is the reason why a disaggregated and scalable architecture such as the one envisioned in dReDBox is a perfect fit for network monitoring.

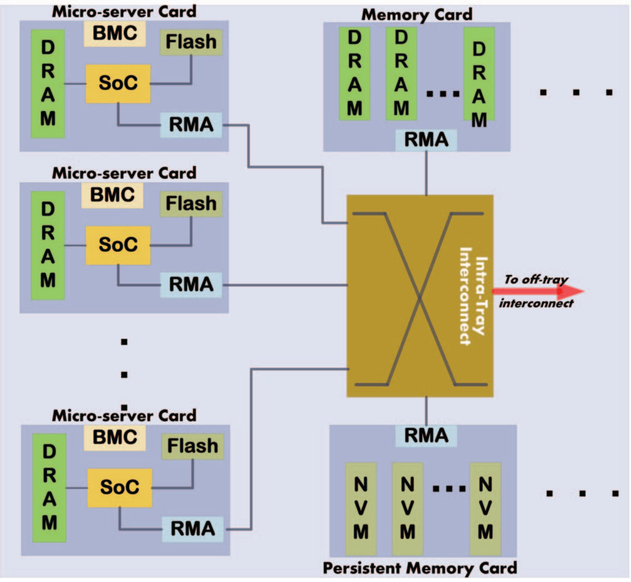

Fig. 1. Block Diagram of one type of dReDBox server

III. Architecture and Subsystems Overview

dReDBox adopts a vertical architecture to address the disaggregation challenges. At the lowest level, the optical interconnect architecture aims to be used for remote memory communication, with latencies in the order of tenths of nanoseconds. The interface facilitating remote memory communication will be decoupled from the processor unit, and appropriately integrated with the system interconnect. Forwarding operations at that level will be software controlled to provide the necessary support to other dReDBox components. The dRedBox datacenter will adopt the Virtual Machine (VM) as the execution container and several challenges will be addressed at the hypervisor and orchestration tools level like the implementation of RDMA-like support for peripheral communications using standard DMA programming as well as power consumption control support. In this section, we provided an overview of selected concepts and the target integrated architecture is summarized. Figures 1 and 2 depict high-level block diagrams of the target server- and rack-level architecture.

A. Server and Rack Level Architecture

dReDBox aims to deliver at least two types of hardware component blocks and one type of mainboard tray. Both component block types will be interfaced via Remote Memory

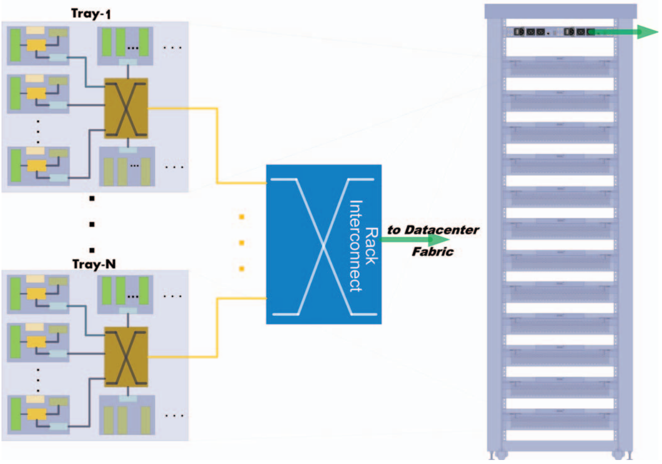

Fig. 2. Block Diagram of high-level dReDBox Rack-scale architecture

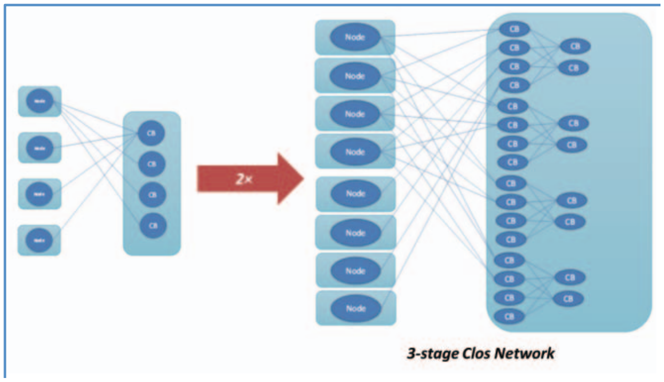

Fig. 3. Scaling of a Clos network element

Adapters (RMA) to the dReDBox mainboard. More specifically, the compute block shown in Figure 1 will feature a high performance SoC integrated with a local RMA, local memory, flash memory and an ethernet-based Board Management Controller (BMC). In Figure 1, two memory blocks are also depicted, featuring DRAM and NVM modules respectively and both interfacing to the rest of the chassis via a local RMA; the same block may also be serving as an accelerator module. In order to support the large scale network monitoring use case that exploits the dReDBox accelerator framework, we will explore interfacing the memory blocks with much higher bandwidth interfaces (40G/100G network interface)

The generic dReDBox tray is aimed to feature a series of memory slots, appropriately interconnected to provide: (i) power, (ii) a serial electrical interface to the electro-optical crossbar chip and (iii) PCIe interface, so that selected DIMM slots provide connectivity to a PCIe switch, and (iv) a per component IPMIv2 [5] interface that enables intelligent IPMIv2-based management from the orchestration tools. Two or more dReDBox mainboards are aimed to feature an appropriate number of interfaces to get interconnected with each other via the optical network. Fig. 2 depicts a set of dReDBox mainboard trays being interfaced to a rack-level interconnect, forming a disaggregated, Rack-Scale architecture. dReDBox aims to deliver a fully-functional PCB prototype of the described platforms in several copies to be used for the integration of the rest of the subsystems and use case demonstrations. The mainboard will be a small factor prototype appropriate for a datacentre-in-a-box configuration.

B. Electro-optical switched interconnect

The vision of disaggregation at rack-scale and with high-density servers poses various challenges to the system interconnect: scaling of the network, controlling end-to-end latency and power consumption requires the addition of switches, links, and hierarchy levels to support resource pooling. Each switch added to the network must be connected to both a higher and lower hierarchy level and has upfront implications to the switch radix and network topology selected. A challenging exercise to be carried out throughout the project is to decide the number of interfaces to be used for the various layers of the interconnection hierarchy (in-tray, off-tray, off-rack), as driven by workload requirements specification and technology density/cost. . In any network configuration, the design goal is to minimize the number of layers used since more layers imply more switch elements and thus higher cost, power consumption and higher latency as one moves from $2-hop$ to a multi-hop topology.

Fully non-blocking networks (logical full mesh topology) typically employ a Clos topology [6]. A Clos network (the most efficient topology) is possible when following two conditions are met: i)The node has enough interfaces to connect to all switches, and ii) The switch has enough interfaces to connect to all nodes and switches symmetrically.

When one of the two above conditions is not met, the Clos structure must use a higher number of switching stages to maintain its fully non-blocking nature. For instance, if the number of ports is for example 8, then a typical scheme would be to connect 4 compute nodes on one side with 4 higher level switches (1:1 oversubscription). This is shown on the left side of Fig. 3, assuming that the network capacity has to double from 4 to 8 nodes, Clos theory mandates that in order to maintain a non-blocking network, the number of switching elements must increase by $6x$ as shown on the right side of Fig. 3. Obviously, the limiting factor is the number of available ports on the switching device and since this number is limited, Clos network used are more than a single stage, with 3- and 5-stages being common in large datacentres.

Increasing the chip I/O bandwidth is not a simple task since the packaging technology is quickly reaching its limit. With more and more SERDES added to the package, the trace breakout into the PCB becomes a limiting factor since - at high line rate - traces cannot be brought close to each other without special design. The complexity associated with trace breakout implies that the SERDES can be placed only on the perimeter of the package. Their number is thus constrained by the package size and the number of BGA pins that may be used on the perimeter. Currently, the state of the art is roughly 150 SERDES/package and this number is not expected to exceed 250 with very complicated and power-hungry large package design.

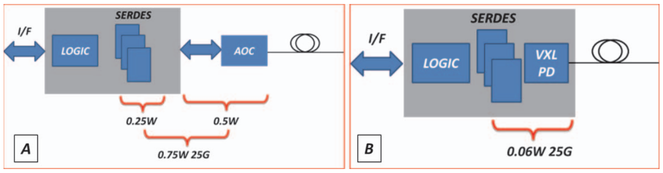

Fig. 4. Power consumption of an electrical I/O (A) and an optical I/O (B); the logic block is assumed to be the same in both cases

The solution to this problem that dReDBox introduces is to add optical I/O to the switching device as described in [7] and [8]. Integrating a large, two-dimensional optical interconnect directly on an ASIC has the benefit that low power SERDES can be used as they need to drive only a fiber link compared to driving a 30" metal trace on a board. The power saving is by a factor of $10x$ and is shown for the two cases in Figures 4.

Interfacing a parallel optical interconnect directly to the SoC ASIC allows to increase both the switch radix and the bandwidth-distance product of the device. Since data flow to and from the ASIC is routed via the optical interface, the number of SERDES used for I/O is the number of optical lanes utilized. The logic to be implemented is L2 forwarding with full crossbar functionality. A fiber matrix is assembled above the optoelectronic matrices to enable card-to-card and tray-to-tray connectivity. Naturally, some of the I/O can be still routed through the conventional electrical SERDES and thus, the overall I/O bandwidth of the device is the combination of both optical and electrical I/O.

With the optical interconnect integrated directly to the SoC ASIC, we expect a network size of $1.5Pb/s$ to be attainable in a single-stage fully non-blocking topology with about 10% of the number of devices needed with other technologies. The power saving and latency gains stemming from this technology and design are evident. Scaling of the data center network is thus enabled using this integrated electro-optical switch. Since the chip radix is in the $15Tb/s$ range, the size of the network that may be obtained in a single-stage Clos topology is larger than any network using conventional SERDES technology.

C. Memory Disaggregation

Memory disaggregation will require appropriate support starting from the lowest level, the memory interconnect architecture. In Non Uniform Memory Access (NUMA) architectures today, the Dual In-line Memory Modules are typically the slowest (but largest) elements in the high-performance memory access loop. The cache architecture is interleaved between processor and memory chips to improve performance. In dReDBox, we aim to disaggregate memory by placing modules on a dedicated memory card and interface them over the system interconnect to the remote memory adapter of processor cards (micro-server). Among the major challenges is to develop an appropriate memory interface and embodying logic for transmission over the optical network. dReDBox aims to integrate existing SoC architectures and aims to design and develop a "Virtual Memory DIMM" component that can be directly interfaced to a commodity DDR controller. This component will accept configuration via memory-mapped I/O using a special address range and aims to be capable of moving memory data to/from the optical network. Moreover a local DIMM module will be appropriately mapped and used for system software bootstrapping support. Virtual DIMM software-defined configuration will associate memory address ranges with optical network forwarding information so that remote DIMMs can be reached. On the memory card pool side a different version of the virtual DIMM will be directly interfaced and will be able to access physical modules on the memory card.

Another important challenge that dReDBox hardware memory interface design aims to address is the distribution of DMA transfer interrupts. In the dReDBox platform, the DMA chipsets that are integrated in microserver SoCs will be used. Each time a DMA transfer is programmed, a list of processor interrupt recipients will be configured at the remote Virtual DIMM. When the transfer is complete, interrupts are to be delivered to remote processor(s) accordingly. The described inter-tray interrupt mechanism will provide valuable support to the virtualization software and aims to enable integration of peripherals which will be driven by dedicated microservers.

D. Operating system support for disaggregation

dReDBox aims to provide a customizable commodity virtual machine execution unit to applications without compromising performance. In order to run the currently available virtual machines without modifications on the disaggregated platform, there are some important challenges that should be addressed at the OS and hypervisor layer.

The dReDBox hypervisor will be based on KVM [9], a kernel module that enables a standard Linux Operating System (namely the host system) to execute one or more virtual machines. Evidently, an instance of KVM will need to run on each microserver platform taking advantage of the locally available memory. Unlike the current server architecture case, the host system on each dReDBox microserver may not be able to detect all available platform components using the BIOS. In fact, the BIOS on each microserver may only provide locally attached component information. Therefore, during bootstrap, the host system should communicate with orchestration tools to get information about all available memory and peripherals of a dReDBox configuration.

Immediately prior to a virtual machine deployment, the hypervisor will interact with the dReDBox orchestration tools, asking to reserve resources and setup appropriate network state for reachability among pooled resources involved. We will explore advanced OS and virtualization techniques for dynamic disaggregated memory allocation at OS level; beyond memory modules, other peripherals are intended to be disaggregated and physically attached to dedicated dReDBox microservers. Such dedicated machines will be connected to peripherals through a PCIe switch. According to this approach, peripherals are aimed to be made available to virtual machines through: direct assignment, or remote para-virtualization. The former is aimed to be available for VMs running on the same server where peripherals are interconnected, and will leverage the current support from the KVM Hypervisor.

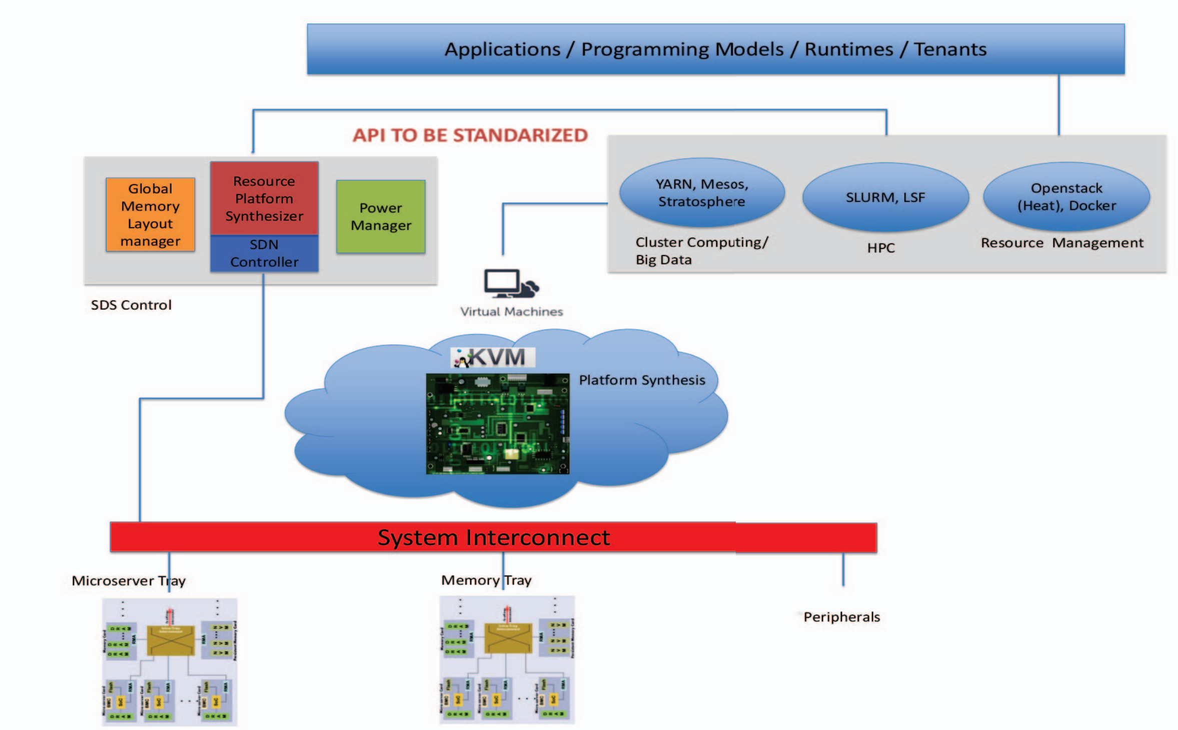

Fig. 5. Resource Allocation and Orchestration in dReDBox

E. Resource allocation and orchestration

dReDBox disaggregated platform needs orchestration support that is not currently available by state-of-the-art datacentre resource management tools. The related challenges are introduced by: (i) the switching network that requires forwarding information and can interconnect any combination of components, (ii) the need for distribution of a globally accessible, datacentre-wide physical memory address space, and (iii) novel per component IPMIv2 control. The dReDBox approach is to create a new orchestration tool layer that aims to (i) implement dynamic platform synthesis by analyzing hardware physical environment and software performance requirement to allocate components and set appropriate forwarding information to interconnect them, (ii) maintain a consistent distribution of physical memory address space and accordingly provide support to running hypervisors for memory segmentation and ballooning, and (iii) taking advantage of the component-level usage information and IPMIv2 control, significantly decrease the required power budget. This orchestration layer is aimed to be integrated via a standardized API with the resource management tools like Openstack. In this context, virtual machine deployment steps will be extended to include a resource scheduling and a platform synthesis step, which aims to reserve the required hardware and configure the platform interconnect in a power-budget conscious manner. Fig. 5 depicts the target dReDBox platform, featuring orchestration tools and relevant component interactions.

IV. Conclusions

The dReDBox project and architecture aims to disaggregate computing resources at the lowest possible level, which would result to a datacenter box that is fully configurable and can adapt to the target applications profile. For example, if applications are highly parallel and CPU intensive, more cores can be accommodated rather than memory, whereas when applications are I/O intensive, cores can be traded off for memory, disks and/or network media, as appropriate. dReDBox will also deliver a vertically integrated working prototype, featuring integration of hardware and system software derivatives and pilot applications porting on the embodiment of the dReDBox architecture.

References

[1] http://www.kitguru.net/components/cpu/anton-shilov/intel-bids-adieu-to-ddr3-majority-of-skylake-s-mainboards-will-use-ddr4/, [Online; accessed May 2015].

[2] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, "Network Support for Resource Disaggregation in Next-Generation Datacenters," in *ACM SIGCOMM, Hotnets-XII*, 2013.

[3] S. M. Rumble, A. Kejriwal, and J. Ousterhout, "Log-structured Memory for DRAM-based Storage, 2014," in *USENIX FAST*, 2014.

[4] V. Moreno, P. M. Santiago del Rio, J. Ramos, D. Muelas, J. L. Garcia-Dorado, F. J. Gomez-Arribas, and J. Aracil, "Multi-granular, multi-purpose and multi-Gb/s monitoring on off-the-shelf systems," *Int. J. Network Mgmt*, vol. 24, pp. 221–234, 2014.

[5] http://www.intel.com/content/www/us/en/servers/ipmi/second-gen-interface-spec-v2-rev1-4.html, [IPMIv2 specification:].

[6] C. Clos, "A Study of Non-Blocking Switching Networks," *Bell System Technical Journal 406-424 (1953)*, vol. 32, pp. 406–424, 1953.

[7] K. Hasharoni, "High BW Parallel Optical Interconnects," in *IAdv. Optical Communication Conference, PT4B.1*, 2014.

[8] K. Hasharoni, S. Benjamin, A. Geron, S. Stepanov, G. Katz, I. Epstein, N. Margalit, D. Chairman, and M. Mesh, "A 1.3 Tb/s parallel optics VCSEL link," in *Proc. SPIE 8991, Optical Interconnects XIV, 89910C*, 2014.

[9] http://www.linux-kvm.org/, [KVM hypervisor].