On Interconnecting and Orchestrating Components in Disaggregated Data Centers: The dReDBox Project Vision

K. Katrinis*, G. Zervas†, D. Pnevmatikatos‡, D. Syrivelis§, T. Alexoudi§, D. Theodoropoulos‡, D. Raho¶, C. Pinto¶, F. Espina‖, S. Lopez-Buedo‖, Q. Chen†, M. Nemirovsky**, D. Roca**, H. Klos†† and T. Berends††
*IBM Research - Ireland
†HPN group, University of Bristol, UK
‡FORTH-ICS & Technical University of Crete
§University of Thessaly
¶Virtual Open Systems
‖NAUDIT HPCN
**Barcelona Supercomputing Center
††SINTECS

***Abstract*—**

Computing systems servers -low- or high-end ones have been traditionally designed and built using a main-board and its hardware components as a "hard" monolithic building block; this formed the base unit on which the system hardware and software stack design build upon. This hard deployment and management border on compute, memory, network and storage resources is either fixed or quite limited in expandability during design time and in practice remains so throughout machine lifetime as subsystem upgrades are seldomely employed. The impact of this rigidity has well known ramifications in terms of lower system resource utilization, costly upgrade cycles and degraded energy proportionality.

In the dReDBox project we take on the challenge of breaking the server boundaries through materialization of the concept of disaggregation. The basic idea of the dReDBox architecture is to use a core of high-speed, low-latency opto-electronic fabric that will bring physically distant components more closely in terms of latency and bandwidth. We envision a powerful software-defined control plane that will match the flexibility of the system to the resource needs of the applications (or VMs) running in the system. Together the hardware, interconnect, and software architectures will enable the creation of a modular, vertically-integrated system that will form a datacenter-in-a-box.

I. Introduction

To address the increasing computing needs across all application domains, high-performance, parallel or distributed and heterogeneous computing is employed across industries. Solutions include massively parallel systems, and the use of GPU-, or even FPGA-based acceleration to achieve low-power, cost efficiency - thanks to fit-for-purpose designs - and for abiding to stringent performance and real-time constraints.

However, computing is co-located in large systems (actually, data-centers) that serve a multitude of applications, most often via virtual machines that shield the application user from the compute cycle provider. This trend is creating an environment where dimensioning a system can be viewed from at least two viewpoints: one is that of the system provider and the other from the application side. For such data centers key performance indicators (KPIs) are compute density, granularity of resource allocation, power consumption and virtual machine migration times. These KPIs challenge the way we design the next generation computing systems.

Data-center systems servers and desktop systems alike have been traditionally designed and built using a main-board and its hardware components as a "hard" monolithic building block; this formed the base unit on which the system hardware and software stack design build upon. This hard deployment and management border on compute, memory, network and storage resources is either fixed or quite limited in expandability during design time and in practice remains so throughout machine lifetime as subsystem upgrades are seldomely employed. The impact of this rigidity has well known ramifications in terms of lower system resource utilization, costly upgrade cycles and degraded energy proportionality. In particular, the proportionality of resources (processor cores, memory capacity and network throughput) within the boundary of the mainboard tray is fixed during design time, as driven by the number of e.g. processor sockets and memory slots manufactured on a mainboard.

In the dReDBox project we take on the challenge of breaking the server boundaries through materialization of the concept of disaggregation. The basic idea of the dReDBox architecture is to use a core of high-speed, low-latency opto-electronic fabric that will bring physically distant components more closely in terms of latency and bandwidth, creating in this way a customizable low-power datacenter architecture. 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. We also envision a powerful software-defined control plane that will match the flexibility of the system to the resource needs of the applications (or VMs) running in the system. Togeher the hardware, interconnect, and software architectures will anable the creation of a modular, vertically-

integrated system that will form a datacenter-in-a-box. The rest of the paper outlines the dReDBox project focus and vision.

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 suboptimal 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 system bus to remote memory controllers that govern DIMM 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 communication is a tightly coupled parallel interface that can achieve a theoretical speed of $1800million$ 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 remote memory controllers and 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.

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

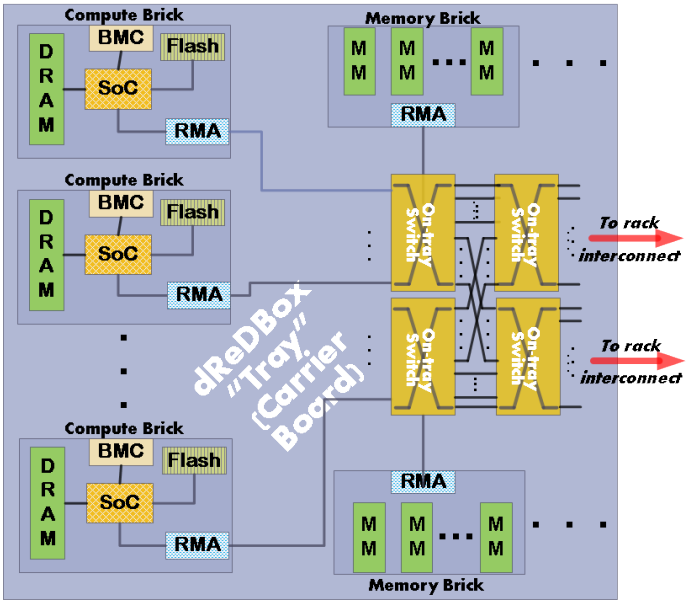

Fig. 1. Block Diagram of one type of dReDBox server tray. "MM" stands for a single memory module (e.g. DIMM or Memory package)

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 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.

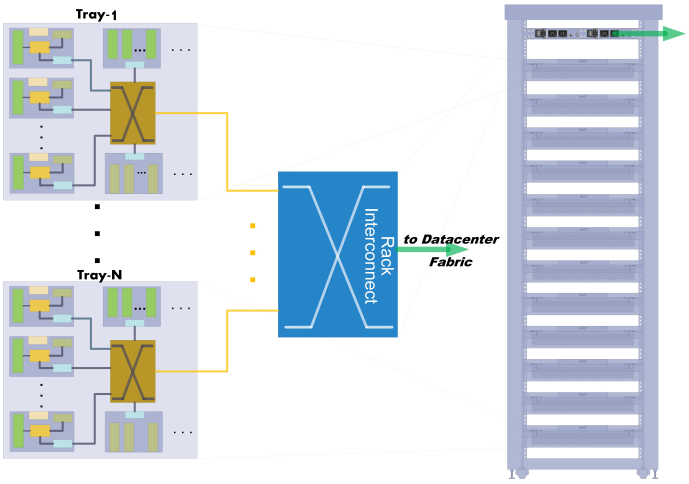

Fig. 2. Block Diagram of high-level dReDBox Rack-scale architecture

B. Electro-optical switched interconnect

Disaggregation puts immense pressure to the network, serving both remote memory/peripheral and data communication. Since the compute and memory function blocks are network attached ultimately their utilization and in turn application performance is directly dependent on network performance. dRedBox is investigating, proposing and implementing novel network architectures, protocols and technologies to deliver optimum performance in terms of network degree (that determines transport latency), level of connectivity and bandwidth granularity, bisection bandwidth, fibre complexity, spatial efficiency $(b/s/\mu^2)$, modularity, power and cost. It is investigating technologies that will best match the specifications of different sub-graphs or sub-systems (i.e. brick, tray, rack). For example, dense opto-electronic transceiver interfaces such as Vertical-Cavity Surface-Emitting Lasers (VCSEL)-arrays offering $12x14Gbs^{-1} or 12x28Gbs^{-1}$ (i.e. $Minipod^{TM}$, $Micropod^{TM}$ developed by Avago [6] , $LightABLE^{TM}$ from Reflex photonics [7]) and/or silicon photonics will be considered as mid board optics (MBO) solutions to maximize bandwidth density and minimize power consumption of each tray. Also we are investigating the best transmission band of operation (i.e. 850nm, 1310nm and 1550nm) that can satisfy such performance targets while considering dense optical switching technologies for minimum latency. Due to pooling of resources, the number of networked endpoint increases, thus requiring much higher port count switch elements and switch systems at all levels (in tray, in-rack, cross-rack); this is even more aggravated in an architecture where microservers are the base unit of compute allocation. While addressing the tray to tray networking aspects it is critical to highlight that there are certain latency bounds imposed by opto-electronic interfaces at each "brick" (e.g. CPU-Memory interface that requires PCS and Layer 2 protocols), both for transmission resilience and performance.

Focusing on memory interconnection elements, which exhibits the most stringent latency requirements, the aim is to explore and evaluate various switching options, such as all-optical circuit-switching and packet-switching. A potential architecture embodiment of the first is shown in Figure 1, whereby an array of crosspoint switches interconnects compute and memory cards ("bricks") within a single tray in a fully non-blocking manner. In such a setup, while full-mesh connectivity is obvious at tray level, maintaining the same degree of connectivity and levels of bisection bandwidth at rack level starts becoming a challenge. The main limitation here is the maximum number of transceivers per microserver card; dReDBox aims to address this challenge by pursuing work on miniaturization of network technologies. An alternative path we aim to explore are hybrid switching architectures, potentially including low-degree fine-granular fast switching with coarse-grained, high-degree slowly reconfigurable optical switching elements.

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 remote memory component that includes controllers and can be interfaced efficiently to a remote processor system bus. 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. Commodity local memory will be also available to support the system bootstrap process.

Another important challenge that dReDBox hardware memory interface design aims to address is the distribution of DMA transfer interrupts. In the dReDBox platform, the standard DMA chipsets that are integrated in microserver SoCs will be used. Each time a DMA transfer is complete, the local processor will be able to notify via interrupt any other processor in the datacenter about the completed memory tranfer to a shared remote module. The described inter-processor 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 [8], 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

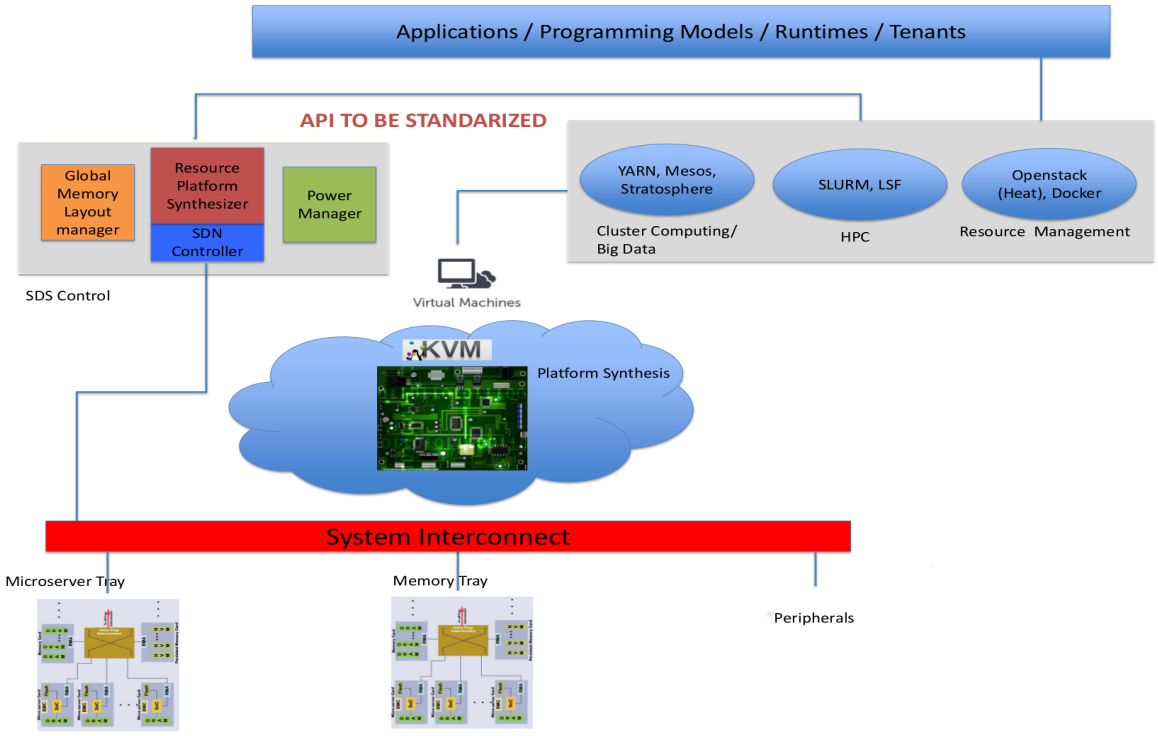

Fig. 3. Resource Allocation and Orchestration in dReDBox

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.

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. 3 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] http://www.avagotech.com/, [Online; accessed May 2015].

[7] http://www.reflexphotonics.com/products_3.html, [Online; accessed May 2015].

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