2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems

# Diluting the Scalability Boundaries: Exploring the Use of Disaggregated Architectures for High-Level Network Data Analysis

Carlos Vega*†, Jose Fernando Zazo*†, Hugo Meyer‡, Ferad Zyulkyarov‡, S. Lopez-Buedo*† and Javier Aracil*†

*Naudit HPCN,
Parque Científico de Madrid, C/Faraday, 7. 28049 Madrid, Spain
Email: {carlos.vega, josefernando.zazo, sergio, javier.aracil}@naudit.es

†Escuela Politécnica Superior, Universidad Autónoma de Madrid,
Francisco Tomás y Valiente, 11, Madrid, Spain
Email: {sergio.lopez-buedo, javier.aracil}@uam.es
Email: {carlosgonzalo.vega, josefernando.zazo}@estudiante.uam.es

‡Barcelona Supercomputing Center
Carrer de Jordi Girona, 29-31 08034, Barcelona, Spain
Email: {hugo.meyer, ferad.zyulkyarov}@bsc.es

***Abstract*—Traditional data centers are designed with a rigid architecture of fit-for-purpose servers that provision resources beyond the average workload in order to deal with occasional peaks of data. Heterogeneous data centers are pushing towards more cost-efficient architectures with better resource provisioning. In this paper we study the feasibility of using disaggregated architectures for intensive data applications, in contrast to the monolithic approach of server-oriented architectures. Particularly, we have tested a proactive network analysis system in which the workload demands are highly variable. In the context of the dReDBox disaggregated architecture, the results show that the overhead caused by using remote memory resources is significant, between 66% and 80%, but we have also observed that the memory usage is one order of magnitude higher for the stress case with respect to average workloads. Therefore, dimensioning memory for the worst case in conventional systems will result in a notable waste of resources. Finally, we found that, for the selected use case, parallelism is limited by memory. Therefore, using a disaggregated architecture will allow for increased parallelism, which, at the same time, will mitigate the overhead caused by remote memory.**

## 1. Introduction

NOWADAYS large data centers serve multitude of different applications, most often via virtual machines (from now on, VMs) that provide an additional level of abstraction. Traditionally, data center servers have been constructed on the basis of hard monolithic building blocks: Motherboards with a fixed number of processor and memory sockets. These building blocks define the characteristics of the VMs that will be run on the data centers. Applications must adapt to the characteristics of the VMs, and scalability is typically horizontal, achieved by instantiating more VMs.

The question is whether this rigid architecture, based on monolithic building blocks, is going to be able to satisfy the requirements of future data centers. A significant increase both in size of data centers [1, Fig. 4] [2, Fig. 2] and in volume of data [3, Fig. 5.8] [2, Fig. 4] is expected in the near future. Additionally, upcoming breakthroughs such as the Internet of Things (IoT) [4] and 100 Gbps networks [6, pg. 7] [7] will pose new challenges for future data centers.

Actually, the rigidity imposed by a monolithic building block draws a clear border on how computing, memory, storage and network resources may be expanded during future upgrades of a particular data center architecture. Decisions taken during design phase will then condition the way a system evolves, with a direct impact in terms of lower system resource utilization, costly upgrades, and poor energy efficiency.

Recently, disaggregated architectures have been proposed as an alternative to overcome the rigidity of conventional servers. The benefits of disaggregation have been previously discussed in the literature [8], either by improving vertical elasticity, proposing separate memory blades that disaggregate memory resources [9] [10]; or with the study of the network capabilities for the disaggregation of resources [11] [12] in data centers. Optical interconnections have also been addressed in the literature [13] for improving both low energy consumption in intra-rack interconnections through optical Top of Rack (ToR) switches and inter-rack communications with new optical switch architectures [14].

Furthermore, the uneven distribution of resources in

978-1-5386-2588-0/17 $31.00 © 2017 IEEE
DOI 10.1109/HPCC-SmartCity-DSS.2017.45

server-oriented architectures impacts energy consumption, which has been well addressed during the last years *"by reducing power draw during idle periods or when at low utilization"* [1, pg. 24]. Disaggregated architectures contribute to this trend offering a finer-grained control over resource provisioning and utilization. In addition to the misallocation of the spatial resources, highly changing workloads over time (e.g. day and night tasks) produce an unbalanced consumption of the available resources.

However, the cost effectiveness of disaggregation still remains a case of study [15], and is hard to quantify at the current stage. However, we strongly believe that cost savings might become one of its major assets due to the better design and low-power components used in these architectures.

For this purpose, the dReDBox[1] (Disaggregated Recursive Datacentre-in-a-Box) project took the challenge [25] of breaking the server boundaries aiming to materialize the concept of disaggregation, benefiting itself from the technological improvements of the interconnection components such as low-latency all-optical switches [16] [17]. The main idea of the dReDBox architecture is to dilute the base unit of data centers through a core of high-speed, low latency optoelectronic fabric that gathers together physically distant components in terms of bandwidth and latency. dReDBox proposes an adaptable low-power data center architecture, moving from the paradigm of mainboard-as-a-unit to a more flexible, software-defined block-as-a-unit schema.

During the development of the dReDBox project, different use cases were studied [18] as representatives of the very large class of possible applications that the system would host in production. The next subsection focus on the particular case of data analysis and how disaggregated architectures suit their requirements.

### 1.1. Data Analysis in disaggregated architectures

Data analytics tools usually show a highly varying demand of both processing and storage resources, which usually requires to squeeze either vertical or horizontal scalability, or even both. For instance, indexing a massive amount of documents in NoSQL databases such as Elasticsearch[2] or Apache Solr[3] may require heavy memory usage for caching and queuing documents, while, on the contrary, aggregation operations (e.g. calculating interval percentiles) in these search platforms cause a high CPU demand. Thus, resource fragmentation arises under heterogeneous and varying workloads like the aforementioned.

Network data analysis is no exception to the high variability of resource demands. Actually, network traffic in data centers varies widely over time and seasons. Figure 1 shows, for two different data centers, the high variability that exists in the number of records obtained per TB of captured network traffic. In this context, a traffic record corresponds to the series of statistics obtained per flow/transaction/conversation by the traffic dissection tools. There are different dissector tools for each application or

[1] http://www.dredbox.eu/

[2] https://elastic.co

[3] http://lucene.apache.org/solr/

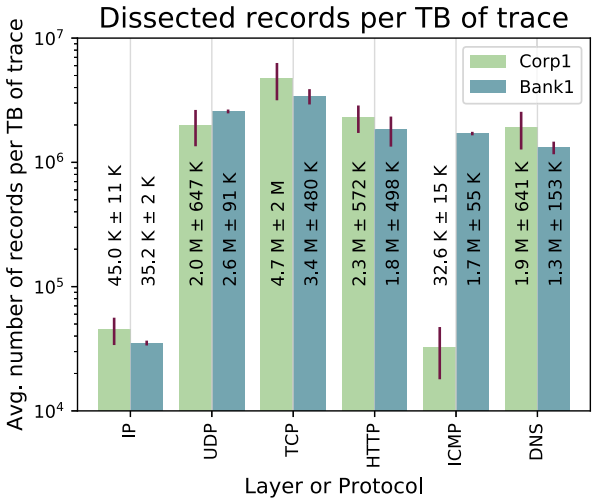



Figure 1. High variation in the number of traffic records per TB of trace from different enterprise networks, obtained by different traffic dissectors.

protocol layer; in the figure, the number of records from 6 different dissectors (IP, UDP, TCP, HTTP, etc.) is depicted.

Therefore, in traditional data centers, the shape and characteristics of data call for over-dimensioning of systems beyond the actual average needs, in order to deal with occasional workload peaks. This is, provisioning more vertical or horizontal resources than usually needed, if not both of them. Actually, horizontal scaling might not always be simple, as the nature itself of the information poses a challenge for an even distribution of the data. For example, for the network analytics problem, doing an even distribution of Internet flows between several systems is a challenging task. As it is explained in in [21], *"under certain Zipf-like flow-size distributions, hashing alone is not able to balance workload"*, that is, the straightforward solution of using a hash to horizontally distribute Internet flows is not enough.

Consequently, the vertical elasticity provided by disaggregated architectures is ideal to alleviate occasional workload peaks in proactive data analysis tasks. For instance, sales in U.S. department stores during the Christmas selling season usually register a 40% jump from the previous month [23] [24]. This increment would require higher amount of resources for the analysis of commercial transactions compared with the rest of the year. This seasonal peak of workload may not be worth the costs of having underutilized resources during the rest of the year. However, on a disaggregated data center, workload variations would be lessened, allocating unused remote resources to meet the particular demands.

In this paper we study the feasibility of using disaggregated architectures for intensive data analysis tasks by studying a case of use regarding network analysis. As noted before, disaggregation generally has a throughput penalty regarding remote resource latencies when the ratio of non-local operations increases. However, throughput penalties in disaggregated architectures is still an open research problem [22] and is strongly dependent on the task conducted and IO access patterns.

In light of the above, we now dwell on how the rest of the paper is structured: In Section 2 we present the dReDBox disaggregated architecture for data centers. An architecture simulator is then described in Section 3 as the testbed for later evaluations of the use case presented in Section 4, in which, we will further deepen on the particular data analysis case of high-level network analysis in disaggregated architectures. Finally, we summarize our conclusions about the evaluation and future expectations on the topic.

## 2. The dReDBox Architecture

To address these disaggregation challenges, the dReDBox project [25] adopts a vertical architecture, with the lowest level consisting of an interconnection system for remote memory communication that takes full advantage of optical solutions for latencies in the order of tenths of nanoseconds. By interconnecting remote memory controllers and modules with novel scalable optical networks we achieve multi-Tbps-level switch bisection.

Between the many innovation aims of the project, some stand out, particularly:

- Delivering a novel hyper-visor distributed support to share resources, to allow the execution of commodity VMs, which will be adopted as the execution container.
- Making use of a software-defined control for all resources at the hardware programmability level. This hardware orchestration software is to be interfaced via APIs with higher-level resource provisioning, management and scheduling systems.
- Reducing the power consumption at all layers. The hardware platforms provides an IPMIv2 interface on a per component basis, providing full orchestration control to the hyper-visor.

The dReDBox architecture allows resource usage to flow between the basic unit blocks in the same way the hosting machines provide resource flexibility to processes hosted in VMs, allocating slices of hardware within a server.

### 2.1. Server and Rack Architecture

The dReDBox platform interfaces different hardware blocks via Remote Memory Adapters (RMA), including a high performance System on a Chip (SoC) for the compute blocks, local memory, flash memory and an ethernet-based Board Management Controller (BMC).

The chosen system architecture for the SoC is based on ARMv8-A Cortex 64-bit [26] processors due to its low-power consumption and reduced cost. Although ARM-based application processors can be found in about an 85% share of mobile devices [27] there has been a recent strong interest for the use of ARM cores also in high-performance computing [28] [29]. ARM expects that *'25% share of servers shipped in 2020 will be using ARM-based chips."* [27].

Figure 2 depicts a high-level abstraction of the current server architecture, showing how the different components

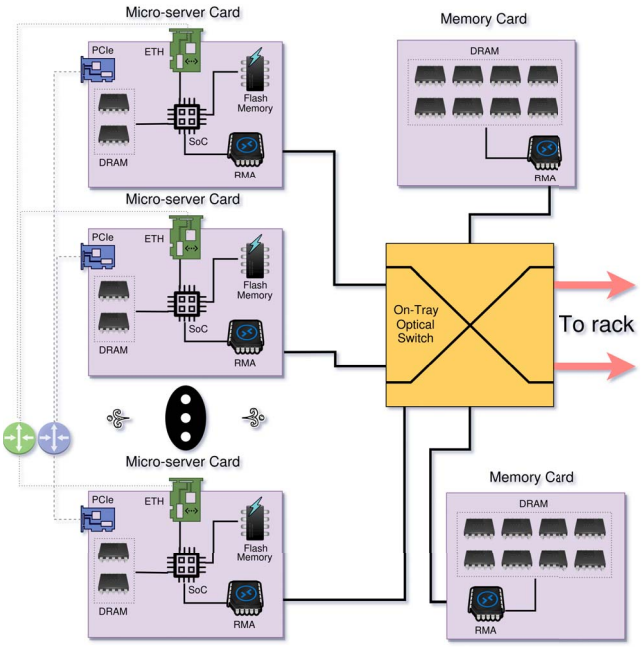



Figure 2. Block diagram abstraction of one type of dReDBox server tray. "ETH" stands for Ethernet and "RMA" for Remote Memory Adapters.

interconnect each other within a generic dReDBox mainboard tray. On the rack-level several trays interconnect themselves through a ToR switch.

### 2.2. Electro-optical switched interconnect

Disaggregation inherently depends on the network performance, which is crucial to serve remote resources. High level of connectivity, bandwidth granularity and low latencies are key for the development of such networks.

For instance, dense opto-electronic transceiver interfaces have been considered as mid board optics (MBO) due to its low-power consumption and high bandwidth. Manifold variables, such as the transmission band of operation, must be taken into consideration to deliver optimum performance.

With regard to memory interconnection, which always calls for the most demanding latency requirements, dReDBox architecture considers various switching options including all-optical circuit-switching, and also exploring hybrid switching architectures. The number of networked endpoints increases owing to pooling of resources, requiring switch systems at all levels (in tray, cross-rack, in-rack).

### 2.3. Memory Disaggregation

The dReDBox project disaggregates memory by placing modules on a dedicated memory card and interfacing them over the system and interconnecting them to the remote memory adapter. In this way, dReDBox integrates existing SoC and remote memory components. For that purpose, the development of a memory interface and embodying logic for transmission over the optical network is key. The remote memory component accepts configuration via

memory-mapped I/O using special address ranges. Commodity local memory is also available in order to support system bootstrap processes.

### 2.4. Operating system support for disaggregation

The dReDBox platform supports the virtual machine as the execution unit, providing customizable commodity virtual machine execution to applications without compromising their performance. Hence, applications, tools, or systems developed for running in commodity hardware will be able to be deployed without modifications on the disaggregated platform. Aforementioned ARM processors also offer a high level of compatibility to the platform.

The platform hyper-visor is based on *Kernel-based Virtual Machine* (KVM) [4], a kernel module which enables a standard Linux Operating System to host a number of VMs. Host systems on each dReDBox computing component may not be able to detect all the available components on the platform during local hardware initialization. Actually, they will only have information about the locally attached components. Hence, during bootstrap the host system should retrieve this information from the orchestration tools.

### 2.5. Resource allocation and orchestration

The orchestration in the dReDBox platform is key for an efficient allocation of the data center resources. Forwarding information through the switching network is essential for the interconnectivity of any combination of components. The dReDBox approach provides a physical memory address space available across the whole data center, maintaining a coherent distribution of it, with support for memory ballooning and segmentation. Also, the per component IPMIv2 control offers a potential decrease in the power consumption. A standardized API integrates the orchestration layer with resource management tools.

Bearing all of this in mind, prior to a virtual machine deployment, a resource scheduling and a platform synthesis step aim to allocate the required resources and set the platform interconnect in an efficient manner.

## 3. Simulating a disaggregated architecture

Prior to the discussion of our particular use case, in this section we describe the simulation process used for the evaluation of systems in the dReDBox architecture. This simulator is used to model the behavior of a disaggregated architecture, based on Queue Models. The iQ [31] model was designed to represent and analyze memory disaggregation, and a statistics-based queuing-based full system simulator was developed to analyze applications performance in disaggregated systems in a quick and accurate manner. These models employ queue structures, message passing, and latency accumulation in order to model such systems.

[4] https://www.linux-kvm.org/

TABLE 1. MEMORY DISAGGREGATION LATENCIES.

| Module Name | Current Latencies (ns) | Number of accesses per request |
|---|---|---|
| FPGA Addr. Transl. | 72 | 1 (CB) |
| Ingress/Egress | 6.25 | 4 (2 CB, 2 MB) |
| Network-on-Chip | 22.4 | 4 (2 CB, 2 MB) |
| PCS/PMA | 251 | 4 (2 CB, 2 MB) |
| DDR4 | 62.5 | 1 (MB) |

Particularly, this process begins when a message is received at the tail of a queue structure. Then, it is propagated until the head of the queue. Finally, a delay is added to emulate the amount of time that the action for that particular message or component takes. For example, in the case of an Integer ALU, the queue models the ALU input queue, where the message passed represents a particular arithmetic instruction (e.g. add, subtract, equals). Consequently, the delay added will depend on the amount of time the ALU unit typically takes to execute that arithmetic instruction.

Therefore, the simulation process occurs as a collection of discrete events which in our model are then represented as one computational cycle. Execution progress (including instruction generation, propagation, waiting, execution, and retirement as well as resource usage and branch and memory misses) is simulated within the queuing model for every event (i.e every cycle).

Total performance is measured as a collection of processed messages per total events, i.e., Instructions Per Cycle (IPC). The event driven queuing model we are proposing uses a modular combination of various queue structures, dependency tracking, and probabilistic execution flow to simulate particular systems.

The queue-model-based methodology emulates processor components by abstracting the implementation details into modular components composed of queue structures, delay parameters and probabilistic driven message generation and event control.

Figure 3 shows the bricks' interconnection through an optical switch for simplification purposes. In order to represent processors, memories and other components using queuing models, we have implemented a modular queue structure that models different behaviors through a set of variable configurations (the red **a** letter in Fig. 3 indicates the beginning of instruction processing). At the left-bottom

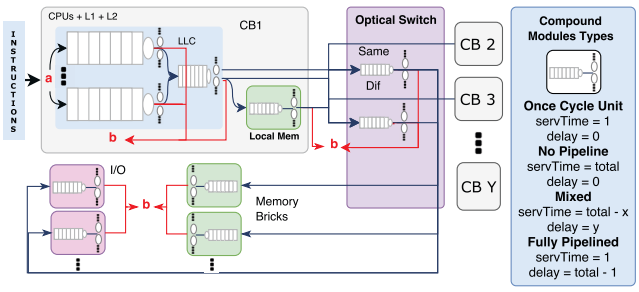



Figure 3. iQ model representing components of the dReDBox platform.

side of Figure 3 the module that is used to represent each component is depicted. This module is formed by a queue, a server and a delay. The queue length and delays required to process instructions are flexible and configurable. The length parameter is used to model resource contention and availability. The service time (*servTime*) represents the time needed to process an instruction until the following instruction may start to be processed. The instruction's total execution time inside a compound module will be the sum of its own *servTime* plus the service time of all previous executed instructions (pipelining). The lower the service time, the higher level of pipeline and vice versa. The delay (*delay*) parameter is used to complement the *servTime* to ensure the appropriate total delay is added to the instruction.

Instructions are generated according to the statistical information collected during the profiling stage making use of Linux perf[5] and pintools such as mica[6]. These instructions are then introduced at the entry point of the Compute Bricks as shown in Figure 3. Afterwards, depending on the probability values, the instructions will move from one queue to another or to the sink (point **b** in Fig. 3). Instructions move from the different levels of cache and the local memory. In the case that instructions need to access remote bricks (I/O, Memory, etc.), they may need to go through the Optical Circuit Switch (OCS).

For instance, in [31, Table 2] Meyer et al. depict the simulator validation with a comparison between real IPC and simulated IPC for different use cases. These results correspond to past evaluations using previous models but serve as a guide. The IPC is used as a performance indicator in all the experiments, and it helps to analyze the impact of disaggregation in the different use cases.

More specifically, during our evaluation the processor simulated was an Intel® Xeon® CPU E5-2630 0 @ 2.30GHz, including all its ALUs and cache levels. The values used to simulate memory disaggregation are presented in Table 1. This table also includes the number of times that a module is accessed per each remote memory request in the Compute Brick (CB) and in the Memory Brick (MB). The latency values presented in the Current Latencies column of Table 1 are preliminary results which correspond to the current dReDBox prototype currently under development. These latencies as well as the the performance degradation will be reduced during the refinement phase of the project, expecting to halve the total latency.

## 4. High-level Network Analysis

In order to evaluate the feasibility of using disaggregated architectures for intensive data analysis tasks, a use case of infrastructure analytics is considered. Network monitoring and auditioning is currently a problem very hard to scale when dealing with the current speeds of enterprise backbones (e.g. 10 to 100 Gbps) [30] [7].

Nowadays the analytics platform struggles to take full advantage of the motherboard capabilities of conventional systems, because migrating to other boards with more resources has the prohibitive cost of moving data to a different place. Also, due to the nature of the traffic, distribution of the capture and analysis process poses a major challenge. This is why disaggregated and scalable architectures such as the one conceived by the dReDBox project suit the needs of network monitoring and auditioning systems.

For our particular example we considered a traffic analysis solution, *FERMIN* [7] (Factual Executive Report of a Monitored IP Network) for the generation of automated reports aimed to improve the traffic monitoring in large IT infrastructures. Needless to say, service outages are one of the main troubles of any data center or network manager. For instance, in a sample of 69 data centers from 43 institutions, the average cost per minute of data center outage was about 7,908 USD [32]. To anticipate these incidents, proactive traffic analysis is an essential activity in network data center management, helping to proactively identify potential issues and their root cause, before they happen.

In our typical deployment scenario we consider an IT infrastructure that performs IT service continuity management (ITSCM), which conducts a risk analysis for each of the IT services in order to pinpoint the vulnerabilities, assets, threats and countermeasures for each of these services. As part of this process, a traffic analysis is performed either on a daily or weekly basis during the night, by means of an automated process, generating a traffic analysis report. Then, countermeasures and corrective actions are deployed, and monitoring and alarm systems are updated in consequence.

We note that the former are reactive monitoring systems, that in case of an incident, can provide microscopic information about a set of metrics. Usually, they are based on SNMP polling and highlight events like intensive CPU usage or link utilization. Many incidents happen as a consequence of a previously existing problem that was latent and remained undetected. Precisely, such sort of latent and underlying problems are the focus of automatic traffic analysis, which is mostly proactive. Therefore, risk assessment is a long-term proactive activity, based on macroscopic analysis of the traffic and logs. Both approaches are meant to be complementary and serve each other.

Figure 4 depicts this process: Firstly, traffic from the corporate network must be captured through specialized drivers capable of receiving network traffic at 10 Gbps such as the HPCAP network driver [30] for Intel® Ethernet

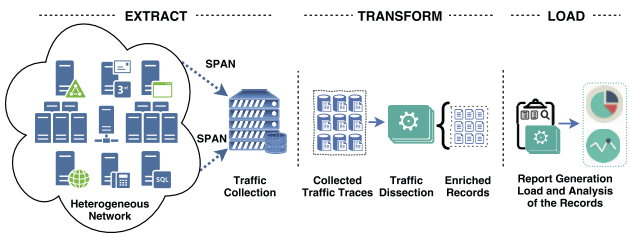
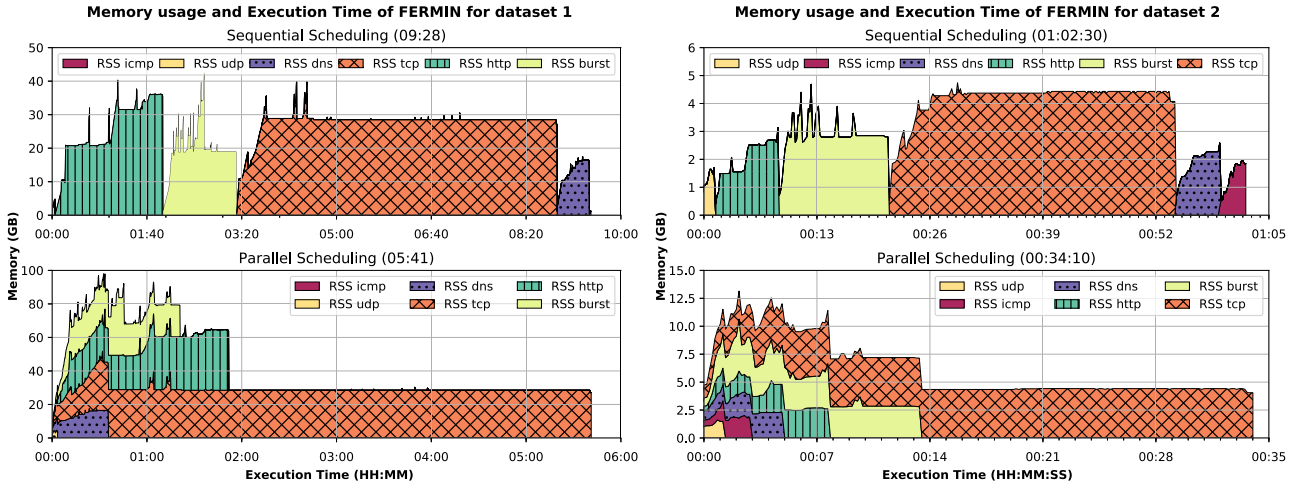



Figure 4. Different stages of the high-level network traffic analysis process.

[5] https://perf.wiki.kernel.org/ [6] http://kejo.be/ELIS/mica/

[7] http://www.naudit.es/fermin




(a) Stress case with high workload.

(b) Average workload case.

Figure 5. Stage-layered execution time and memory usage of the proactive network analysis in different scenarios. RSS stands for Resident Set Size.

10 Gb PCI Express NICs. During traffic dissection, the traffic trace is summarized into enriched records that contain amenable information for the analysis. Traffic dissectors provide records per service (e.g. HTTP, DNS, SQL, etc.), per protocol (e.g. TCP, UDP, ICMP, etc.). The origin of each type of these dissected enriched records may have different nature, since some dissection tools are able to dissect the traffic with streaming mechanisms, off-line techniques, or make use of log records from different services to provide aggregated information. In this respect, there is a clear compromise between real-time delivery of the records and accuracy of the obtained statistics.

For instance, traffic monitoring probes could provide enriched records in real-time with statistics such as the number of TCP zero window announcements from a given client or server in a particular flow. Nevertheless, not only the zero window announcement events matter, but also how long the client or server was not opening the window. If an endpoint announces a zero window just once in a TCP connection, then it is negligible. Nevertheless, if such an announcement is held for 10 seconds, then chances are that the server is heavily overloaded. As noted before, the latter number of zero windows announcements can be delivered in real time, by means of a counter per flow. However, the blocked time should be obtained off-line, not to increase the probe processing requirements while capturing and storing traffic at high speed.

Is not the purpose of this paper to discuss the particular insights of the tool, but we find necessary to summarize some of its main features. Besides macroscopic traffic analysis, *FERMIN* provides a proactive anomaly detection system to feed back the IT manager with a relevant briefing of the most abnormal and interesting events, all of that through a series of Key Performance Indicators (KPI), such as burst analysis, Red Amber Green (RAG) analysis of different protocols for the main servers, or Topology analysis of both MAC and IP levels, among others.

In order to address the multiple analysis requirements, which usually demand high-level statistical functions, *FERMIN* was developed in *Python* due to its high versatility and portability, and making use of statistical libraries such as *Numpy*[8] and *Pandas*[9], which allow us to update and evolve our system faster. We also developed several modules making use of C language for filtering and helping caching the data and deferring read operations until needed, efficiently.

Although networks show clear day-night or work-holiday behaviors, there are unexpected events that could significantly alter traffic, hence, the computational load of a network analytics problem may be unpredictable.

For example, Figure 5 show two different workload cases of a proactive network analysis conducted in a traditional network probe using an Intel® Xeon E5-2640 v3 @ 2.6Ghz with 128GB of RAM. We collected two groups of real network traces from different corporations captured by traffic probes. Both of these groups of traces, **Dataset 1** and **Dataset 2**, correspond to Spanish multinational companies. These two groups have different characteristics regarding the traffic. The 6.9 TB traffic traces from the stress case **Dataset 1** were captured from two different interfaces in the edge of their network. On the other hand, **Dataset 2** was captured in the the core distribution layer, providing 2 TB of traffic traces, which represent the average amount of traffic during an usual day.

As observed, the resources needed to address the stress case reach even one order of magnitude more than during the average use case. In fact, in order to tackle with these unpredictable peaks of traffic, we would need to provision quite more resources than usually needed. During peak dates, the main issue is not just the data volume variation but the huge increments in the number of IP conversations, UDP/TCP flows or HTTP transactions. Figure 6 represents the humongous increase in the number of records dissected corresponding to IP conversations, UDP/TCP flows and

[8] http://www.numpy.org/ [9] http://pandas.pydata.org/

HTTP transactions, etc. between the stress and average cases, as well as the differences in the analysis durations. On a disaggregated architecture workload peaks would be alleviated making use of unused remote resources.

## 5. Evaluation, Discussion, and Future Work

In this light, we conducted a profile evaluation of *FERMIN* in the way described in Section 3, with the purpose of collecting statistical information about memory access patterns and dependencies between instructions. The simulation was validated through this information with a comparison between real and simulated Instructions Per Cycle (IPC), yielding an error of 4.5% over the real profiled IPC. Afterwards the application was simulated on the disaggregated architecture model, see the results in Table 2. The simulated IPC without disaggregation is 1.37 ($\mathbf{IPC_{sim}}$). The table presents different results for memory cards with 1, 2, 4, or 8 communication endpoints, each endpoint provides 16 Gbps. The results show that for the current latencies the overhead is about 80%, which can be reduced to 66% if latency is halved (expected in future refinements of dReDBox architecture).

We note that access and provision of remote memory resources should not be the usual behavior but serve to alleviate stress cases such as those described in previous sections. In this sense, the trade-off between the latencies overhead and resource proportionality may be worthwhile for batch-processing systems, including proactive network analysis among others, with high workload variability.

Such systems do not require hard real-time constraints and can defer their processing tasks in favor of better resource provisioning and scheduling. In those cases, the benefits of a disaggregated architecture is that data analysis tasks, that otherwise would be impossible due to high memory requirements, can be done at the cost of a 66% performance overhead.

An additional benefit of provisioning remote memory resources is that parallelism is not limited by lack of memory.

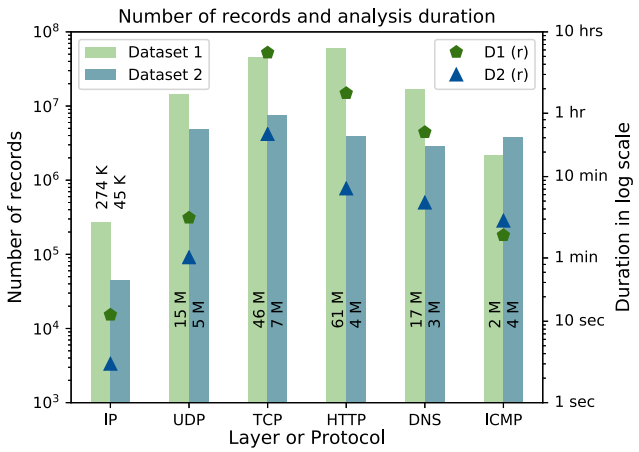



Figure 6. Amount of dissected records per layer/protocol together with their corresponding analysis processing time with FERMIN.

TABLE 2. SIMULATION RESULTS

| **Number of Endpoints** | **$IPC_{disagg}$ with current latencies** | **Overhead [%]** | $IPC_{disagg}$ with future latencies | Overhead[1] [%] |
|---|---|---|---|---|
| 1 | 0.25 | 81.75 | 0.456 | 66.72 |
| 2 | 0.26 | 81.02 | 0.463 | 66.20 |
| 4 | 0.29 | 78.83 | 0.469 | 65.77 |
| 8 | 0.30 | 78.10 | 0.465 | 66.06 |

[1] Overhead = $\frac{IPC_{sim} - IPC_{disagg}}{IPC_{sim}} * 100$

For example, consider Figure 6 where the execution time and memory usage of FERMIN is depicted for the stress case presented previously, but two different schedules for processes have been evaluated: parallel and sequential. Memory usage for the parallel schedule is notably higher, because at the beginning of the experiment several processes run in parallel, and each one allocates a significant amount of memory. On the contrary, the sequential schedule features a much lower memory usage (peak usage is 40 GB vs. 100 GB for the parallel schedule), but at the cost of almost doubling execution time. That is, if memory resources are limited, parallelism could be jeopardized, because it is going to be the lack of memory and not the number of cores available in the processors what is going to limit parallelism. Moreover, the overhead caused by using remote memory resources can be mitigated by increased parallelism.

To conclude with, although disaggregation has a clear throughput penalty when the ratio of nonlocal operations increases, this is, the access of remote resources and the induced latencies during the process, we believe that the strengths of such architectures in terms of scalability and resource balance pose a worthwhile compromise. We look forward to the interesting improvements in the interconnection systems, which will add elasticity to the monolithic building block of traditional data centers. Therefore, we expect the valuable information retrieved from our evaluation to help on the refinement phase of the dReDBox architecture.

## Acknowledgments

This work has been partially supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 687632 (dReDBox Project).

## References

[1] Shehabi, A., Smith, S. J., Horner, N., Azevedo, I., Brown, R., Koomey, J., and Lintner, W. (2016). United States data center energy usage report. Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775 Page, 4. https://eta.lbl.gov/publications/united-states-data-center-energy

[2] Networking, C. V. (2016). *Cisco Global Cloud Index*: Forecast and Methodology, 2015-2020. White paper. http://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.pdf [Online; accessed 15-Feb-2017].

[3] Peña-López, Ismael and others: *OECD Internet Economy Outlook 2012*, Chapter 4 (2012) http://dx.doi.org/10.1787/9789264086463-en

[4] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys and Tutorials, 17(4), 2347-2376. https://doi.org/10.1109/COMST.2015.2444095

[5] Manyika, J., Chui, M., Bughin, J., Dobbs, R., Bisson, P., and Marrs, A. (2013). Disruptive technologies: Advances that will transform life, business, and the global economy (Vol. 180). San Francisco, CA: McKinsey Global Institute. http://library.wur.nl/WebQuery/clc/2079131

[6] Arista in Q1 2017 https://s2.q4cdn.com/209832288/files/doc_presentations/2017/q1/2017-Highlights-Q1.pdf

[7] Zazo, J. F., Lopez-Buedo, S., Sutter, G., and Aracil, J. (2016, November). Automated synthesis of FPGA-based packet filters for 100 Gbps network monitoring applications. In ReConFigurable Computing and FPGAs (ReConFig), 2016 International Conference on (pp. 1-6). IEEE. https://doi.org/10.1109/ReConFig.2016.7857156

[8] Pags, A., Serrano, R., Perell, J., and Spadaro, S. (2017). On the benefits of resource disaggregation for virtual data centre provisioning in optical data centres. Computer Communications, 107, 60-74. https://doi.org/10.1016/j.comcom.2017.03.009

[9] Lim, K., Chang, J., Mudge, T., Ranganathan, P., Reinhardt, S. K., and Wenisch, T. F. (2009, June). Disaggregated memory for expansion and sharing in blade servers. In ACM SIGARCH Computer Architecture News (Vol. 37, No. 3, pp. 267-278). ACM. https://doi.org/10.1145/1555754.1555789

[10] Tu, C. C., Lee, C. T., and Chiueh, T. C. (2014, October). Marlin: A memory-based rack area network. In Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems (pp. 125-136). ACM. https://doi.org/10.1145/2658260.2658262

[11] Han, S., Egi, N., Panda, A., Ratnasamy, S., Shi, G., and Shenker, S. (2013, November). Network support for resource disaggregation in next-generation datacenters. In Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks (p. 10). ACM. https://doi.org/10.1145/2535771.2535778

[12] Greenberg, A., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2008, August). Towards a next generation data center architecture: scalability and commoditization. In Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow (pp. 57-62). ACM. https://doi.org/10.1145/1397718.1397732

[13] Kachris, C., and Tomkos, I. (2012). A survey on optical interconnects for data centers. IEEE Communications Surveys and Tutorials, 14(4), 1021-1036. https://doi.org/10.1109/SURV.2011.122111.00069

[14] Fiorani, M., Aleksic, S., Casoni, M., Wosinska, L., and Chen, J. (2014). Energy-efficient elastic optical interconnect architecture for data centers. IEEE Communications Letters, 18(9), 1531-1534. https://doi.org/10.1109/LCOMM.2014.2339322

[15] Abali, B., Eickemeyer, R. J., Franke, H., Li, C. S., and Taubenblatt, M. A. (2015). Disaggregated and optically interconnected memory: when will it be cost effective?. https://arxiv.org/abs/1503.01416

[16] Hasharoni, K. (2014, July). High BW parallel optical interconnects. In Photonics in Switching (pp. PT4B-1). Optical Society of America. https://doi.org/10.1364/PS.2014.PT4B.1

[17] Hasharoni, K., Benjamin, S., Geron, A., Stepanov, S., Katz, G., Epstein, I., and Mesh, M. (2014, March). A 1.3 Tb/s parallel optics VCSEL link. In SPIE OPTO (pp. 89910C-89910C). International Society for Optics and Photonics. http://doi.org/10.1117/12.2038073

[18] dReDBox Deliverable D2.1: Requirements specification and KPIs Document http://www.dredbox.eu/deliverables.html

[19] Vega, C., Roquero, P., and Aracil, J. (2017). Multi-Gbps HTTP traffic analysis in commodity hardware based on local knowledge of TCP streams. Computer Networks, 113, 258-268. http://doi.org/10.1016/j.comnet.2017.01.001

[20] Deri, L., Martinelli, M., Bujlow, T., and Cardigliano, A. (2014, August). ndpi: Open-source high-speed deep packet inspection. In Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International (pp. 617-622). IEEE. http://doi.org/10.1109/IWCMC.2014.6906427

[21] Shi, W., MacGregor, M. H., and Gburzynski, P. (2005). Load balancing for parallel forwarding. IEEE/ACM Transactions on Networking (TON), 13(4), 790-801. https://doi.org/10.1109/TNET.2005.852881

[22] Li, C. S., Franke, H., Parris, C., Abali, B., Kesavan, M., and Chang, V. (2017). Composable architecture for rack scale big data computing. Future Generation Computer Systems, 67, 180-193. https://doi.org/10.1016/j.future.2016.07.014

[23] Sales in U.S. Department Stores during the period 2000-2016 https://www.census.gov/econ/currentdata/dbsearch?program=MRTS&startYear=2000&endYear=2016&categories=4521I&dataType=MPCSM&geoLevel=US&notAdjusted=1&submit=GET+DATA&releaseScheduleId=

[24] The 2015 Holiday Season. U.S. Census CB15-FF.25 https://www.census.gov/newsroom/facts-for-features/2015/cb15-ff25.html

[25] Katrinis, K., Zervas, G., Pnevmatikatos, D., Syrivelis, D., Alexoudi, T., Theodoropoulos, D., and Chen, Q. (2016, June). On interconnecting and orchestrating components in disaggregated data centers: The dReDBox project vision. In Networks and Communications (EuCNC), 2016 European Conference on (pp. 235-239). IEEE. https://doi.org/10.1109/EuCNC.2016.7561039

[26] Grisenthwaite, R. (2011). ARMv8 technology preview. IEEE Conference. https://www.arm.com/files/downloads/ARMv8_Architecture.pdf

[27] ARM Strategic Report, 2015 v2. https://www.arm.com/company/investors/-/media/arm-com/company/Legacy%20Financial%20PDFs/ARM_Strategic_Report_2015_v2.pdf

[28] Rajovic, N., Carpenter, P. M., Gelado, I., Puzovic, N., Ramirez, A., and Valero, M. (2013, November). Supercomputing with commodity CPUs: Are mobile SoCs ready for HPC?. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (p. 40). ACM. https://doi.org/10.1145/2503210.2503281

[29] Durand, Y., Carpenter, P. M., Adami, S., Bilas, A., Dutoit, D., Farcy, A., and Matus, E. (2014, August). Euroserver: Energy efficient node for european micro-servers. In Digital System Design (DSD), 2014 17th Euromicro Conference on (pp. 206-213). IEEE. https://doi.org/10.1109/DSD.2014.15

[30] Victor Moreno, Pedro M. Santiago del Río, Javier Ramos, David Muelas, José Luis García-Dorado, Francisco J Gomez-Arribas, and Javier Aracil: *Multi-granular, multi-purpose and multi-Gb/s monitoring on off-the-shelf systems*. International Journal of Network Management (2014) http://doi.org/10.1002/nem.1861

[31] Hugo Meyer, Jose Carlos Sancho, Josue V. Quiroga, Ferad Zyulkyarov, Damian Roca and Mario Nemirovsky.: *Disaggregated Computing. An Evaluation of Current Trends for Datacentres*. International Conference on Computational Science (ICCS) 2017 http://doi.org/10.1016/j.procs.2017.05.129

[32] Cost of datacenter Outages, datacenter Performance Benchmark Series, Ponemon Institute, January 2016. http://www.ponemon.org/blog/2016-cost-of-data-center-outages