UNIVERSITY OF LEEDS

This is a repository copy of *Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server Design*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/101027/

Version: Accepted Version

Proceedings Paper:
Mohammad Ali, HM, Al-Salim, AMA, Lawey, AQ et al. (2 more authors) (2016) Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server Design. In: 18th International Conference on Transparent Optical Networks (ICTON 2016). ICTON 2016, 10-14 Jul 2016, Trento, Italy. IEEE . ISBN 9781509014675

https://doi.org/10.1109/ICTON.2016.7550696

(c) 2016, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Reuse
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

White Rose
university consortium
Universities of Leeds, Sheffield & York

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

Energy Efficient Resource Provisioning with VM Migration Heuristic for Disaggregated Server Design

Howraa M. Mohammad Ali, Ali M. Al-Salim, Ahmed Q. Lawey, Taisir El-Gorashi and Jaafar M. H. Elmirghani
School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, United Kingdom

ABSTRACT

This article introduces an energy efficient heuristic that performs resource provisioning and Virtual Machine (VM) migration in the Disaggregated Server (DS) schema. The DS is a promising paradigm for future data centers where servers' components are disaggregated at the hardware unit levels and resources of similar type are combined in type respective pools, such as processing pools, memory pools and IO pools. We examined 1000 VM requests that demand various processing, memory and IO requirements. Requests have exponentially distributed inter arrival time and with uniformly distributed service duration periods. Resources occupied by a certain VM are released when the VM finishes its service duration. The heuristic optimises VMs allocation and dynamically migrates existing VMs to occupy newly released energy efficient resources. We assess the energy efficiency of the heuristic by applying increasing service duration periods. The results of the numerical simulation indicate that our power savings can reach up to 55% when compared to our pervious study where VM service duration is infinite and resources are not released.

Keywords: Virtual Machine, VM Migration, Disaggregated Server, data center, energy consumption.

1. INTRODUCTION

Nowadays, virtualization within cloud computing are becoming prevalent technologies that greatly shape our lives, and accordingly, the energy consumption of the physical infrastructure that provides resources for the clouds is growing [1]. Thus, energy management is becoming a key challenge for data centers to reduce all their energy consumption and their total costs. A number of energy efficient data centre and inter data centre network architectures were proposed and studied in [2]-[7].

Serious concerns are raised currently regarding traditional server and networking design and significant research efforts were dedicated to improve the design of switches and servers, leading to new paradigms for future data centers designs [8]-[19].

In [8], InfiniBand was introduces as a type of communications link for data flow between processors and I/O devices. Similarly, at the 2013 Open Compute Project, Facebook and Intel presented their first disaggregated rack using Intel's new photonic architecture [9]-[11]. In [12], HP presented their software defined server. With the HP Moonshot 1500 chassis architecture, the network, storage and management fabrics are separated. In [13] the authors discussed the ability of the data centers network to support this new architecture. The idea of disaggregated servers has been discussed in [14]-[16], and different disaggregated server architectures were introduced by presenting the memory and IO ports as separate blades from the server block allowing resources to be disaggregated across a system to enable data centers vertical elasticity. In [17] and [18] the energy sufficiency potential of the DS architecture was studied considering the resource provisioning and VM allocation. In [19] a comprehensive review of the DS was introduced.

A DS based data centre, shown in Fig. 1, is a departure from the conventional (Single box Server) to disaggregated resources combined in resource pools. With DS, the different server resources are disaggregated from each other and combined with resources of the same type in resources pool constructing CPU pools, memory pools and IO pools. However the communication among resources within the same pool (Intra Rack Fabric in Fig. 1) or resources from different pools (Inter Rack Fabric in Fig. 1) is an important point that must be managed efficiently.

In this paper, we consider a DS-based data center as a new data center architecture that can address the emerged challenges facing traditional data centers. We especially address the increased energy need of current data center to serve the incoming VM requests. Optimizing resource provisioning and VM allocation as well as performing VM migration results in fewer working resources, and hence, lower power consumption in the data center by switching off unused resources. In section 2 we introduce the virtualization and VM migration in brief, while in section 3 we present our heuristic approach. Results are presented and discussed in Section 4, and section 5 concludes our work.

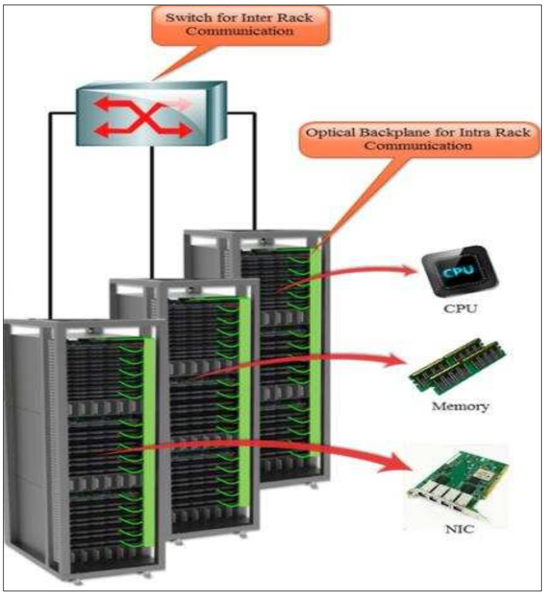


Fig. 1 Disaggregated server architecture.

2. VIRTUALIZATION and VM MIGRATIONH

Virtualization is a key technology for cloud computing as it allows several VM instances to be embedded on the same physical machine by running them on top of a software layer known as a hypervisor. The hypervisor enhances resources manageability by simulating the underlying hardware platform and provisions hardware resources to the VM requests, and consequently consents virtualization which brings set of new services and applications to the data centers [20]. In addition to enabling higher utilization of hardware resources, by packing VMs in minimum underlying physical resources, virtualization facilitates VM movement from one host to another which is an important feature considering server consolidation, power consumption and data center manageability.

The migration procedure is a heuristic approach. VMs are instated by hosting them in the optimal available server that meet some criteria such as resource availability, energy efficiency, latency bounds … etc. if a VM finished its service duration and released its occupied resources in a particular server, the following qualifications must be guaranteed before performing VM migration to that server: (i) the target server has enough capacity to host the migrated VM, (ii) the migration will save some resources, (iii) the migration will not increase the migrated VM duration. If all these conditions are satisfied, then the migration can be done [21].

Thus, currently there is huge drift toward virtualized data centers in order to address traditional data centers limitations, improve the data center efficiency, better resource utilization, simplified resource management and reduced power consumption.

3. EERPVMM-DS HEURISTIC APPROACH

In this work we develop a heuristic that completes our work appeared in [18] by considering VMs having finite service duration. The heuristic in [18] assumed that VMs have infinite service durations, therefore, VMs are assumed to arrive all at once and occupy the resources permanently. This might be a reasonable assumption for certain classes of VMs, such as IaaS [22], however, many other VMs request are instated for limited service duration, such as Amazon AWS [23]. We consider the VM arrivals following the Poisson distribution and associated with exponentially distributed inter arrival times (IAT). The IAT reflects how frequently jobs (i.e. VMs) are being submitted to the data center while the service duration is the total period a VM needs to finish its processing task and then leaves the server.

The EERPVMM-DS heuristic, Fig. 2, aims to pack incoming VMs in the minimum number of resources with minimal power to be consumed. According to eq. 1, resource power consumption is proportional to its Power Factor (PF, eq. 2), and its utilization (δ); therefore, resources with small PF are the optimum candidates to host a VM.

$$Pr = PMin + (PF \cdot \delta)) \qquad (1)$$

$$PF = \left(\frac{PMax - PMin}{Cap}\right) \qquad (2)$$

Where $PMax$, $PMin$ and Cap are the maximum active power, idle power and total capacity of the resource, respectively. In this study the resource type can be processor, memory or IO port. The heuristic first assesses the resources according to their PF and capacity. The heuristic then constructs sorted lists of the resources (i.e. one list per resource type) by organizing them in an ascending order starting with resources having smallest PF, and moving on to resources with higher PF, and if two resources of similar type have the same PF, then the resource with the highest capacity is favoured. The DS is presented with a set of requested VMs. The heuristic finds the total required time slots to serve all the requested VMs. The VMs to be served in each time slot are obtained by knowing the IAT of each VM and its service duration. A time slot is a unit of time and it could be in seconds, minutes or hours. Each VM duration is represented by the number of time slots it needs. Each VM is expected to occupy some resources for at least one time slot.

Then the heuristic creates a list for the number of VMs in each time slot by arranging the VMs in the time slots according to their arrival time and service duration. For example if VM1 arrives at the first time slot and needs 3 time slots then it will appear in time slots 1, 2 and 3. Similarly, if VM2 arrives at the third time slot and needs 7 time slots then it will appear in time slots 3-9. Subsequently, the heuristic must loop for all the time slots and serve the requested VMs in each time slot.

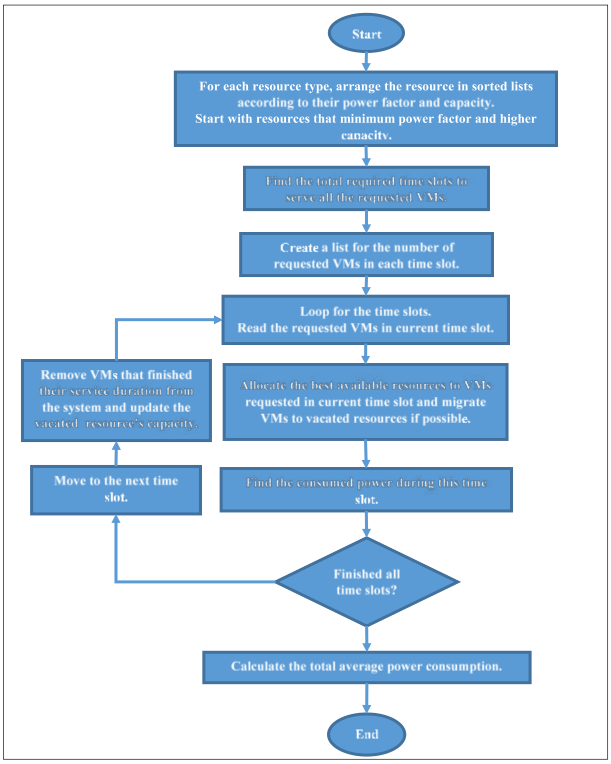


Fig. 2 Heuristic flowchart.

For each time slot, and for each VM in this time slot, the heuristic will select one of the resources from the top of each sorted list; recall that there are three lists, processors list, memories list and IO ports list. Similar to our work in [18], the heuristic will check the chosen resources to find out if there is enough capacity on each resource type to serve the VM request. First the processor will be tested, if it does not have enough capacity then the heuristic will pick up a new processor from the processor sorted list, otherwise, the heuristic will test the selected memory. Again, if the memory does not have enough capacity to serve the VM under consideration, a new memory must be retrieved from the memory list; otherwise the selected IO port must be tested. If the IO port can accommodate the network traffic requirements of the VM under consideration, it will be used directly; otherwise a new IO port must be retrieved from the IO list.

After choosing all three resources that can fulfil the current VM demands, the heuristic allocates these resources to the VM under consideration and updates the used resources' remaining capacity. In each time slot, the heuristic tries to occupy the remaining capacity of the already utilised resources by packing them with as many requests as they can serve and if any one of them cannot serve more VM requests then the heuristic proceeds to the next resource in the sorted list and use it. For each time slot and after serving all the VMs in current time slot, the heuristic calculates the total power consumed in that time slot .

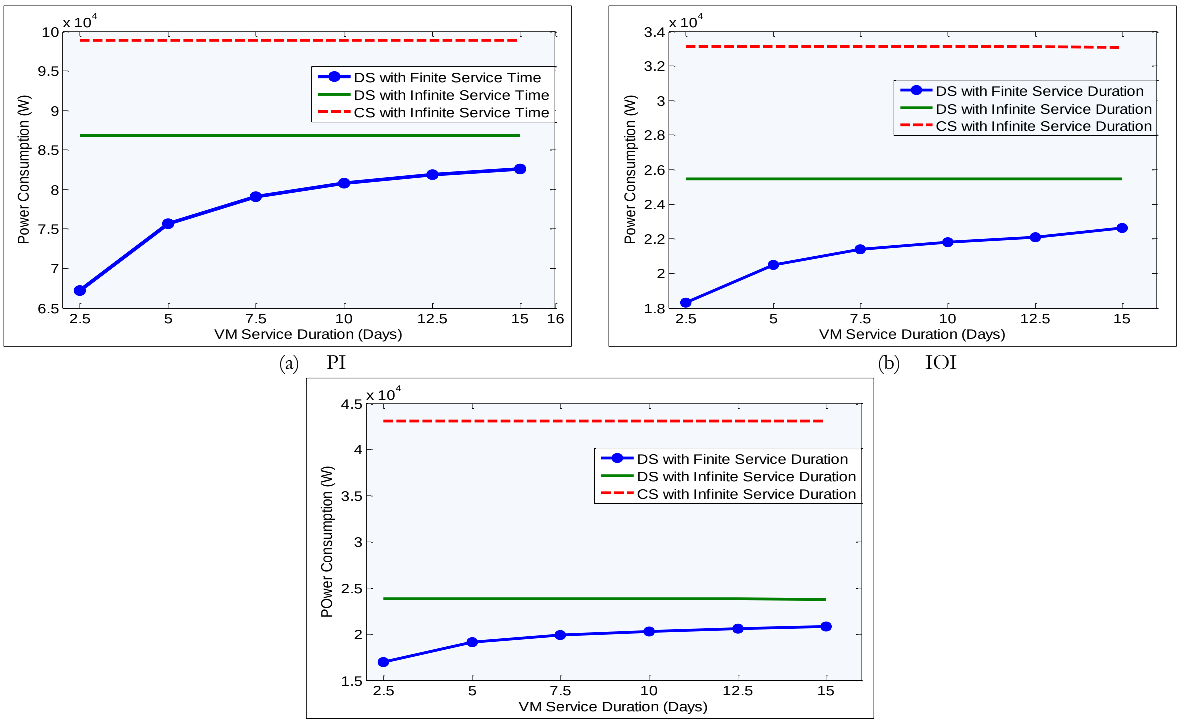
After that the heuristic will move on to the next time slot where the resources occupied by VMs that finishes in that time slot are to be released. The loop continues by repeating the above steps untill all VMs are served. Finally the heuristic calculates the average power consumption of the DS due to serving the current VMs set.

4. RESULTS

In this section we compare our current EERPVMM-DS heuristic and our previous heuristic appeared in [18], to show how considering VMs with finite serving duration affects the DS power consumption. We assume that the VMs IAT is exponentially distributed with mean of 1 minute, considering that we are dealing with data center of average access rates [24][25]. The IAT spans from few seconds to reach maximum value of 5 minutes, and the serving durations are generated using the uniform distribution. We consider the same three types of VMs as in [18]: (i) Processing Intensive Requests (PI), (ii) Memory Intensive Requests (MI) and (iii) IO Intensive Requests (IOI).

For the simulation and evaluation of the DS architecture we are considering a set of heterogeneous resources by disaggregating the IBM X3650 M3 server system [26] and using the same parameters in [17]. Note that, given our set of resources, processors are the most power consumers and memory resources are the least power consumers, while the IO ports' power consumption lies between the two.

The results in Fig. 3 show the power consumption considering the DS with infinite [18] and finite VM service durations, and also the Conventional Server (CS) power consumption with infinite service durations [17]. The graphs show clearly that considering VM with finite service durations with dynamic resource allocation has a positive impact on the total data center power efficiency.



(a) PI

(b) IOI

(c) MI

Fig. 3 Power consumption evaluation considering CS and DS with infinite service duration and DS with finite service duration

In our simulation, we are evaluating the power consumption of the different data centers designs considering 1000 VM requests, where each request has different resources requirements. We are assessing the DS with a range of VMs serving durations to show the effects of increasing the VM average service duration on the DS power consumption. Our findings states that when dealing with an approximately low service durations, resources can be used more efficiently as the number of VMs that finish their serving duration and leave the system are considerably high. Thus the resources can be reused to serve new incoming VMs, which eliminates or reduces the need to turn-on new resources. This is to be compared to our previous analysis in [17] where VMs were considered to have an infinite service duration and all VMs arrive at once.

With increasing the VM service duration, the number of VMs that leave the data center decreases, resulting in continuous increase in the number of working resources, and consequently increases the total power consumption. As limited number of VMs are leaving the system, new arriving VM requests will probably not be able to fit in any of the already working resources, thus new resources will be needed to be turned on, leading to an increase in the number of working resources.

Investigating Fig. 3 reveals that when handling VMs with average service duration around 15 days, which could be any cloud service such as SaaS or PaaS [22], our new heuristic will approach in behaviour our old heuristic in [17], which indicates that almost all the incoming VM requests are staying in the system as if they have infinite durations, and the used resources are remaining under the occupancy of the already arrived VMs.

From Fig. 3, it is apparent that serving VM requests with finite service duration will reduce the power consumption of the DS compared to the case when VMs were assumed to have infinite service duration, and the average power savings for the considered range of VM service durations are10% for the PI, 17% for IOI, and 18% for the MI. This is due to our given assortment of input parameters for the different resource types. Note that PI VMs scenario has the least power saving as the saving will be from the efficient use of the lower power consuming resources, i.e. the memory and IO ports, as compared to the high power consumption of processors. When considering the other two scenarios, i.e. MI and IOI VMs, the higher power saving will be achieved through the efficient use of the most power hungry processing resources, thus these scenarios will attain a higher power savings compared to the PI scenario.

What is interesting in the data given by Fig. 3 is that, comparing the power consumption of the EERPVMM-DS heuristic to the old CS scenario unveils massive power savings. About 55% of the consumed power in the old CS scenario will be saved when considering time for the MI VMs type, 36% for the IOI VMs type, and 21% for the PI VMs type in average.

This behaviour asserts the fact that considering VMs having finite service time, the resources will not be loaded with all VMs at the same time. Rather, after a VM finishes its serving duration, the used resources will be vacated from this VM, and therefore we can perform VM migration by moving VMs that still need further processing to emptied resources. Thus the most efficient resources, from the top of the resources' lists will be used more likely than other resources as they will be reused after being vacated from finished VMs. This alongside the better resource packing are the main factors that achieve this improved power efficiency.

In addition, considering the VM service durations are exponentially distributed will reduce the power consumption which means the average service duration that is needed to catch our old heuristic with infinite service duration is more than 15 days. Also increasing the mean IAT value for more than 1 minute will reduce the total power consumption values.

5. CONCLUSION

In this paper, we introduced our new energy efficient EERPVMM-DS heuristic that performs resource provisioning and VM migration in the Disaggregated Server (DS) paradigm. For simulation, we considered 1000 VM requests which have various processing, memory, IO, and serving duration demands, and consider a range of heterogeneous resources to be used for performing resource provisioning. The VMs are assumed to arrive with exponentially distributed inter arrival times and requesting uniformly distributed service durations. The heuristic optimises VMs allocation and dynamically migrates VMs to occupy newly released energy efficient resources. The heuristic results showed that the power consumption of the data center has been reduced remarkably as compared to our old work that compares Conventional Server (CS) with infinite service duration and DS with infinite service duration. Respectively, compared to the CS and DS with infinite service duration, EERPVMM-DS power savings are 55% and 18% when considering MI VM requests, 36% and 17% for the IOI VM requests and 21% and 10% when considering PI VM requests, in average. Our next step is to extend this work by designing a photonic communication fabric for the DS based data centre and revisiting our model and heuristics to obtain the relevant power savings when including the communication power consumption.

ACKNOWLEDGEMENT

The authors would like to acknowledge funding from the Engineering and Physical Sciences Research Council (EPSRC), INTERNET (EP/H040536/1) and STAR (EP/K016873/1). Mrs. Howraa M. Mohammad Ali would like to acknowledge The Higher Committee for Education Development in Iraq (HCED/IRAQ) for funding his scholarship. Mrs. Howraa M. Mohammad Ali would like to thank Mr. Mohamed O. I. Musa for his help and useful discussion.

REFERENCES

[1] J. Baliga, R. W. Ayre, K. Hinton, and R. Tucker, “Green cloud computing: Balancing energy in processing, storage, and transport,” Proceedings of the IEEE, IEEE, vol. 99, no. 1, pp. 149–167, 2011.

[2] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, “Green IP Over WDM Networks With Data Centers”, Journal of Lightwave Technology, VOL. 29, NO. 12, JUNE 15, 2011

[3] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, “On the Energy Efficiency of Physical Topology Design for IP Over WDM Networks”, Journal of Lightwave Technology, VOL. 30, NO. 12, JUNE 15, 2012

[4] X. Dong, T. El-Gorashi, and J. M. H. Elmirghani, “IP over WDM networks employing renewable energy sources”, Journal of Lightwave Technology, VOL. 29, NO. 1, JANUARY 1, 2013

[5] A. Q. Lawey, T. E. H. El-Gorashi, and Jaafar M. H. Elmirghani, “Distributed Energy Efficient Clouds Over Core Networks”, Journal of Lightwave Technology, VOL. 32, NO. 7, APRIL 1, 2014

[6] A. Q. Lawey, T. E. H. El-Gorashi, and Jaafar M. H. Elmirghani, “BitTorrent Content Distribution in Optical Networks”, Journal of Lightwave Technology, VOL. 32, NO. 21, NOVEMBER 1, 2014

[7] L. Nonde, T. E. H. El-Gorashi, and Jaafar M. H. Elmirghani, “Energy Efficient Virtual Network Embedding for Cloud Networks”, Journal of Lightwave Technology, VOL. 33, NO. 9, MAY 1, 2015

[8] InfiniBand Intro, Mellanox Technologies Inc. white paper, 2003.

[9] http://arstechnica.com/information-technology/2013/04/intel-wants-to-kill-the-traditional-server-rack-with-100gbps-links/.

[10] D. Version, “Design Guide for Photonic Architecture”, Intel draft document, January, 2013.

[11] The Case for Rack Scale Architecture, Intel Data Center Group. White paper, September 2014.

[12] H. P. Moonshot, F. P. Led, H. P. Proliant, M. Server, S. Module, S. F. P. U. Module, C. Management, C. Manager, R. Fabric, and T. M. Fabric, “HP Moonshot System The world’s first software defined servers”, HP technical white paper, April, 2013.

[13] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, “Network support for resource disaggregation in next-generation datacenters,” Proc. Twelfth ACM Work. Hot Top. Networks - HotNets-XII, pp. 1–7, 2013.

[14] P. Sv, B. Hudzia, J. Tordsson, and E. Elmroth, “Hecatonchire : Enabling Multi-Host Virtual Machines by Resource Aggregation and Pooling”, a report published by the Department of Computing Science, Umeå University, Sweden November 2012, available http://umu.diva-portal.org/smash/get/diva2:712161/FULLTEXT02.pdf, accessed 16/01/2016.

[15] K. Lim, J. Chang, T. Mudge, P. Ranganathan,S. K. Reinhardt, T. F. Wenisch, “Disaggregated Memory for Expansion and Sharing in Blade Servers”, ACM SIGARCH Comput. Archit. News, vol. 37, pp. 267, 2009.

[16] C. Tu and C. Lee, “Marlin: A Memory-Based Rack Area Network”, Proceedings of the tenth ACM/IEEE symposium on Architectures for networking and communications systems, pp. 125–135, 2014.

[17] H. M. M. Ali, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient Disaggregated Servers for Future Data Centers”, 20th IEEE Conference on Network and Optical Communications, NOC 2015, London, UK, June 30th to July 2nd, 2015.

[18] H. M. M. Ali, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Energy Efficient Resource Provisioning in Disaggregated Data Centers”, Asia Communications and Photonics Conference, ACP 2015, Hong Kong Hong Kong 19–23 November 2015.

[19] H. M. M. Ali, A. Q. Lawey, T. E. H. El-Gorashi, and J. M. H. Elmirghani, “Disaggregated Racks: Transformation in data center design”, Submitted to IEEE Communication Magazine - Green Communications and Computing Series.

[20] P. Savard, “Live VM Migration Principles and Performance”, LICENTIATE THESIS published by the Department of Computing Science, Umeå University, Sweden, May 2012, available http://umu.diva-portal.org/smash/get/diva2:707793/FULLTEXT02.pdf accessed 16/01/2016.

[21] K. Li, H.g Zheng, and J. Wu, “Migration-based Virtual Machine Placement in Cloud Systems”, IEEE 2nd International Conference on Cloud Networking (CloudNet), 2013.

[22] https://apprenda.com/library/paas/iaas-paas-saas-explained-compared/ accessed 09/03/2016.

[23] How AWS Pricing Works, Amazon web services white paper, 2015.

[24] D. Juan, L. Li, H. Peng, D. Marculescu, and C. Faloutsos, “Advances in Knowledge Discovery and Data Mining: Beyond Poisson: Modeling Inter-Arrival Time of Requests in a Datacenter”, 18th Pacific-Asia Conference, Taiwan, 13-16 May, 2014.

[25] Srikanth Kandula, Sudipta Sengupta, Albert Greenberg, Parveen Patel, Ronnie Chaiken, “The Nature of Datacenter Traffic: Measurements & Analysis”, Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference, ACM New York, NY, USA 2009.

[26] IBM, “IBM System x3650 M3”, available http://www.manualslib.com/manual/75575/Ibm-X3650-M3.html accessed 27/07/2015.