

Հ Հ Գ Ա Ա Ի Ն Տ Ո Ր Մ Ա Տ Ի Կ Ա Յ Ի Ե Վ Ա Վ Տ Ո Մ Ա Տ Ա Ց Մ Ա Ն  
Պ Ր Ո Ւ Բ Լ Ե Մ Ն Ե Ր Ի Ի Ն Ս Տ Ի Տ Ո Ւ Տ

Արամ Սեյրանի Քոչարյան

**Վ Ի Ր Տ Ո Ւ Ա Լ Հ Ա Մ Ա Կ Ա Ր Գ Ե Ր Ո Ւ Մ Հ Ի Շ Ո Ղ Ո Ւ Թ Յ Ա Ն  
Կ Ա Ռ Ա Վ Ա Ր Ո Ւ Մ Ը**

Ե 13.04 - «Հ ա շ վ ո ղ ա կ ա ն մ ե թ ե ն ա ն ե Ր ի ,  
հ ա մ ա լ ի Ր ն ե Ր ի , հ ա մ ա կ ա Ր գ ե Ր ի և ց ա ն ց ե Ր ի  
մ ա թ ե մ ա տ ի կ ա կ ա ն և ծ Ր ա գ Ր ա յ ի ն ա պ ա հ ո վ ու մ »  
մ ա ս ն ա գ ի տ ու թ յ ա մ Բ տ ե խ ն ի կ ա կ ա ն  
գ ի տ ու թ յ ու ն ն ե Ր ի թ ե կ ն ա ծ ու ի  
գ ի տ ա կ ա ն ա ս տ ի ճ ա ն ի հ ա յ ց մ ա ն  
ա տ ե ն ա խ ո ս ու լ թ յ ա ն

Ս Ե Ղ Մ Ա Գ Ի Ր

Ե Ր և ա ն - 2019

---

INSTITUTE FOR INFORMATICS AND AUTOMATION PROBLEMS NAS RA

Aram S. Kocharyan

**MEMORY MANAGEMENT IN VIRTUAL ENVIRONMENTS**

**ABSTRACT**

Of the dissertation for obtaining PhD degree in Technical Sciences  
on specialty 05.13.04 “Mathematical and Software Support  
of Computers, Complexes, Systems and Networks”

Yerevan - 2019

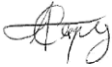
Ատենախոսությունը անթեման հաստատվել է ՀՀ  
ԳԱԱԻՆՖՆՈՐՄԱՏԻԿԱՅԻՆ ԱՎՏՈՄԱՏԱԳՄԱՆ  
ԱՊՐՈՔԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՆԻ ԳԻՏԱԿԱՆ  
ԽՈՐՀՐԴՈՒՄ

Գիտական ղեկավարներ՝ Հ. Ասցատրյան,  
տեխ. գիտ. թեկնածու  
Դանիել Հաջիմունթ,  
տեխ. գիտ. դոկտոր

Պաշտոնական ընդդիմախոսներ՝ Ս.  
Շուքուրյան,  
ֆիզ. մաթ. գիտ. դոկտոր  
Նոել Դեպալմա, տեխ. գիտ.  
դոկտոր

Առաջատար կազմակերպչությունը՝ Հայաստանի  
ազգային  
պոլիտեխնիկական  
համալսարան

Պաշտպանությունը կայանալու է 2019 թվականի  
հունիսի 2-ին, ժ. 15:00-ին ՀՀ ԳԱԱԻՆՖՆՈՐՄԱՏԻԿԱՅԻՆ  
ԱՎՏՈՄԱՏԱԳՄԱՆ ԱՊՐՈՔԼԵՄՆԵՐԻ  
ԻՆՍՏԻՏՈՒՆԻ ՄԳՈՐԾՈՂ 037 «ԻՆՖՈՐՄԱՏԻԿԱ»  
մասնագիտական խորհրդի նիստում:  
Հասցեն՝ Երեւան, 0014, Պ. Սևակի 1:  
Ատենախոսությունը կարելի է ծանոթանալ ՀՀ  
ԳԱԱԻՆՖՆՈՐՄԱՏԻԿԱՅԻՆ ԱՎՏՈՄԱՏԱԳՄԱՆ  
ՍԵՂՄԱԳԻՐՆ Առաքված  
է 2019թ.-ի մայիսի 24-ին:

Մասնագիտական խորհրդի  
գիտական քարտուղար, ֆ. մ.  Հ. Գ.  
Սարուխանյան

---

The topic of the dissertation was approved at the Scientific Council  
of the Institute of Informatics and Automation Problems of NAS RA

Scientific supervisors: Hrachya Astsatryan, PhD  
Daniel Hagimount, PhD, Doctor of Sciences

Official opponents: Samvel Shoukourian, PhD, D.Ph.M.S.  
M. Noel Depalma, PhD, Doctor of Sciences

Leading organization: National Polytechnic University of Armenia

The Defense will take place on July 2, 2019; at 15:00, at the Specialized Council 037  
«Informatics» at the Institute of Informatics and Automation Problems of NAS RA. Address:  
Yerevan, 0014, P. Sevak 1,

The Dissertation is available in the library of IIAP NAS RA.  
The abstract is delivered on May 24, 2019.

Scientific Secretary  
of the Specialized Council, D.Ph.M.S.

 Robert Sarukhanyan

## Executive Summary

**The problem statement:** Virtualization is a foundational element of cloud computing that encapsulates various services that can meet the user requirement in a cloud computing environment. This technology allows service providers to consolidate the servers to one physical machine as Virtual Machines (VMs) reducing the amount of the hardware in use. Cloud computing paradigm leverages virtualization and provides on-demand availability of computer system resources, especially high performance computational (HPC) and high-speed mass storage resources to meet the growing demand for computations and a large volume of data.

In addition to the air conditioning and cooling equipment, such computer system resources are the major source of the power consumption in cloud infrastructures. The power consumption of data centers (DC) is increasing due to several aspects, such as increasing the data volume to deal with or the need for more HPC facilities, which in its term leads to serious environmental issues (including e-waste and CO<sub>2</sub> emission). The energy consumption of DCs is already going upwards to 8-10% of global consumption and the total global footprint is 2% of global CO<sub>2</sub> emissions. Thus, reducing energy consumption will play an important role to decrease the total energy consumption of these centers.

Resource management is crucial for every DC provider that focuses on the efficient sharing of cloud resources among multiple users. The resource optimization is one of the most efficient ways to minimize the energy consumption of DCs. The RM aims to provide the same service with less resources and with the same quality (without Service Level Agreement (SLA) violation). The processor (CPU), Memory (RAM), storage (disk space) and network are the basic resources of DCs.

In the past a few years, the resource optimization challenges for cloud environment received great attention from the researchers. However, it is extremely complex to target all types of resources at once. Thus, there are many works in the field on CPU resource optimization, since it was considered to be the most expensive and main resource in DCs. However, the situation has changed during the years. Over last years, it has been viewed the emergence of new applications with growing memory demands, while hardware platforms' evolution continued to offer more CPU capacity growth than memory, referred to as the memory capacity wall.

**The Aim and objectives:** This dissertation aims to develop a memory management and optimization environment for virtualized DCs implemented in Unix like Operating System (OS) kernels. The studies follow the following objectives:

- To develop an accurate and non-intrusive working set estimation method focused toward track the working set size of a VM.
- To develop a single-node memory management system using above described working set estimation method.
- To develop multi-node memory management and optimization environment for virtualized DCs.
- To develop and implement software solutions for the above described objectives.

**Methods:** The following methods have been used to explore the memory resource management and optimization in datacenters:

- Linux kernel module development aiming to provide customized memory management aware kernels.
- Widely used memory management, migration and consolidation methods (such as memory ballooning, VM migration, consolidation etc).
- Distributed memory sharing mechanisms (remote swapping over RDMA).
- Well known benchmarking suits for data intensive and HPC workloads.

**Research Contributions:**

- An accurate and non-intrusive working set estimation method has been suggested to extend the kernel of a hypervisor and a Unix like OS.
- A single-node memory evaluation and distribution system has been developed allowing to reclaim unused memory from unsaturated VMs and land it to saturating VMs.
- A distributed memory management system has been developed relying on remote swapping technologies for DC-wide memory mutualization.

**Practical Significance:** All the developed methods and environments have been implemented in the cloud infrastructures provided by the IRIT and IIAP. The efficiency of the suggested solutions have been confirmed by several scientific applications and benchmarking suits.

**Presentation of Results:**

- Association for Computing Machinery’s Special Interest Group on Measurement and Evaluation, which specializes in the field of performance analysis, measurement, and modeling of computer systems (ACM *SIGMETRICS*) June 18-22, Irvine, CA, USA, 2018
- Institute of Electrical and Electronics Engineers (*IEEE*) Ivannikov memorial workshop (IVEM), 3 - 4 MAY, 2018 Yerevan, Armenia, 2018

- Conférence d'informatique en Parallélisme, Architecture et Système (*COMPASS*), 3-6 July Toulouse, France, 2018

**The Publications:** The scientific results of the PhD have been published in 4 scientific articles which are highlighted at the end of this abstract.

**Thesis Structure:** The manuscript consists of an introduction, 3 chapters and a conclusion. The manuscript covers 104 pages including including 94 references, 29 charts and 5 tables.

## **Introduction**

The introduction presents the research domain, problem statement, the aim and objectives as well as the focus of the research.

Besides, the introduction gives background information on the used technological stack:

- Main solutions provided by a Cloud provider:
- The technology behind cloud computing, including virtualization.
- The types of hypervisors, the benefits and drawbacks of each type.
- Resource managements that are widely used in virtual environments.
- Dynamic and static resource allocation approaches
- Memory management in virtual environment.

The section also represents the following three main steps that are used by all memory management systems:

- **Monitoring:** The working set of VMs is a challenge for datacenter providers as it allows to measure the memory need by VMs and the memory which can be reclaimed. The reclaimed memory can then be used to satisfy memory needs of other VMs in order to raise the consolidation ratio.
- **Reclamation:** Most of the modern hypervisors implement memory reclamation techniques (memory ballooning) to reclaim unused memory from VMs, thus avoiding resource waste. In such systems, the VM is equipped with a balloon driver, which can be inflated or deflated (by the hypervisor/dom0).
- **Re-Distributing:** Memory reclaimed by the hypervisor on one server can be granted to VM which lack memory on the same server. However, this reclaimed memory cannot simply be allocated to remote VMs.

## **Chapter 1: Studies on Working Set Size Estimation Techniques in Virtualized Environments:**

**Badis** This chapter presents a state-of-the-art survey on working set size estimation techniques and propose Badis, a system that can estimate a VM's working set size with high accuracy and no VM codebase intrusiveness.

Numerous DCs are relying on virtualization, as it provides flexible resource management means such as VM checkpoint/restart, migration and consolidation. However, one of the main hindrances to server consolidation is physical memory. In nowadays cloud, memory is generally statically allocated to VMs and wasted if not used. Techniques (such as ballooning) were introduced for dynamically reclaiming memory from VMs, such that only the needed memory is provisioned to each VM. However, the challenge is to precisely monitor the needed memory, i.e., the working set of each VM. In this context, it has been thoroughly reviewed the main techniques that were proposed for monitoring the working set of VMs. In this work, the main techniques (Geiger, VMware, Exclusive cache, Zballond, Selfballooning) have been implemented in the Xen hypervisor and it has been defined different metrics to evaluate their efficiency. Based on the evaluation results, Badis is proposed, a system which combines several of the existing solutions, using the right solution at the right time. It has been also proposed a consolidation extension, which leverages Badis to pack the VMs based on the working set size and not the booked memory. The implementation of all techniques, our proposed system, and the used benchmarks are publicly available to support further research in this domain.

In summary, the contributions of this work are the following:

- The evaluation metrics have been defined that allow to characterize WSS estimation solutions.
- The existing WSS techniques have been evaluated on several types of benchmarks. Each solution was implemented in the Xen virtualization system.
- Badis has been proposed, a WSS monitoring and estimation system which leverages several of the existing solutions in order to provide high estimation accuracy with no codebase intrusiveness. Badis is also able to dynamically adjust the VM's allocated memory based on the WSS estimations.
- A consolidation system extension is proposed which leverages Badis for a better consolidation ratio. Both the source and the data sets used for our evaluation are publicly available, so that our experiments can be reproduced.

The metrics for characterizing WSS estimation techniques are the following: the intrusiveness (requires the modification of the VM), the activeness (alters the VM's execution flow), the accuracy, the overhead on the VM (noted vm-over), and the overhead on the hypervisor/dom0 (noted hyper-over).

- Vm\_over: it directly impacts the VM performance. It could be affected by both the intrusiveness and the activeness.

- accuracy: a wrong estimation leads to either performance degradation (under-estimation) or resource waste (over-estimation).
- hyper-over: a high overhead could saturate the hypervisor/dom0, which are shared components. This could lead, in turn, to the degradation of VMs' performance (e.g. the I/O intensive VMs).

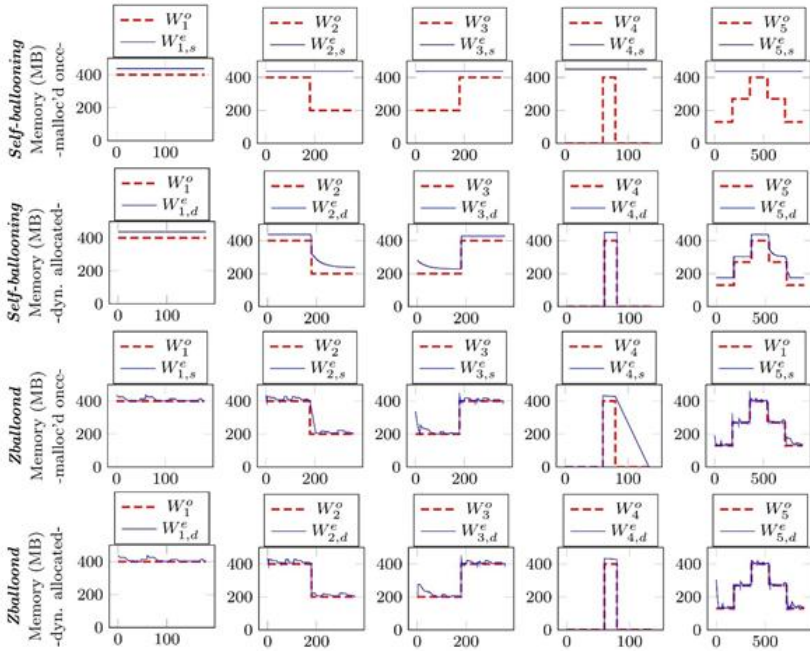


Figure 1. The ability of researched techniques to follow the actual working set of the application (part 1)

Fig.1 and Fig. 2 present the results for each workload and each WSS estimation technique. To facilitate the interpretation of the results, each curve shows both the original workload (noted  $W^o$ ) and the actual estimated WSSs (noted  $W_j^e$ ),  $1 < i < 5$  (represents the workload type) and  $j=s,d$  (represents the implementation type - static or dynamic).



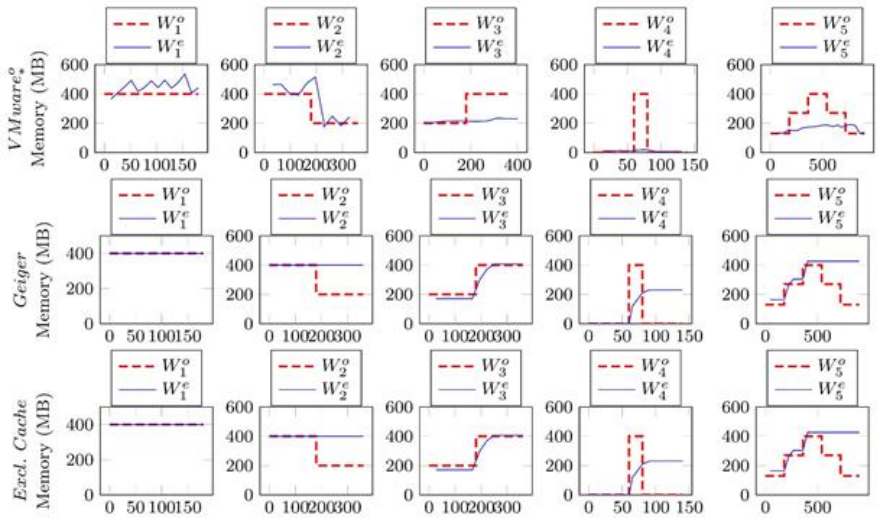


Figure 2. The ability of researched techniques to follow the actual working set of the application (part 2)

	Self-b.	Zballoond	VMware	Geiger	Excl. Cache
intrusive	yes	yes	no	no	no
active	no	yes	yes	no	yes
addressed situations	all	all	$S_{more}$	$S_{less}$	$S_{less}$

	Self-b.	Zballoond	VMware	Geiger	Excl. Cache
accuracy	depends on the app.	high	high in $S_{more}$ zero in $S_{less}$	high in $S_{less}$ zero in $S_{more}$	high in $S_{less}$ zero in $S_{more}$
vm_over	nil	almost nil	nil in $S_{more}$ high in $S_{less}$	almost nil	almost nil
hyper_over	nil	nil	almost nil	almost nil	not negligible

Table 1 Study synthesis of all WSS estimation techniques according to both qualitative and quotative metrics

Table 1 summarizes the characteristics of each technique according to both qualitative and quantitative criteria. Besides these criteria, the evaluation results reveal that not all solutions address the issue of WSS estimation in its entirety. Indeed, a WSS estimation technique must be able to work in the following two situations:

- ( $S_{more}$ ) the VM is wasting memory,
- ( $S_{less}$ ) the VM is lacking memory.

The VMware technique is only appropriate in (Smore) while Geiger and Hypervisor exclusive caches are effective in (S<sub>less</sub>). Only Zballoond and self- ballooning cover both (Smore) and (S<sub>less</sub>). Our study also shows that each solution comes with its strengths and weaknesses.

Badis is a system which smartly combines existing techniques in such a way that both (S<sub>more</sub>) and (S<sub>less</sub>) are covered with no codebase intrusiveness. Indeed, it has been determined that even if the VMware and Geiger solutions have a fairly high performance impact, they have no intrusiveness in the VM's codebase. The second observation is that these solutions are complementary (VMware addresses Smore while Geiger addresses S<sub>less</sub>). The Hypervisor exclusive cache is also a solution that only addresses (S<sub>less</sub>) but it has higher hyper-over. Thereby, a system which can combine VMware and Geiger satisfies all our requirements.

Benchmark and app.		Self-ballooning	Zballoond	Badis	
		vm_over	vm_over	vm_over	hyper_over
Dacapo	avroa	1	1.19	1.26	1.8
	batik	1	1.09	1.57	1.05
	eclipse	1	3.67	1	1.68
	h2	1	2	1.16	1.3
	jython	1	1.58	1.05	1.15
Cloud suite	Data Analytics	1.29	1.4	1.16	1.2
LinkBench	MySQL	1.11	2.92	1.09	1

Table 2 Evaluation of Badis

Table 2 presents the comparison of badis with existing solutions based on macro-benchmarks. Badis, a system which combines several of the existing solutions, using the right solution at the right time. In addition, a consolidation extension has been implemented which leverages Badis for an improved consolidation ratio. The evaluation results reveal a 2x better consolidation ratio with only 3% additional VM migrations.

**Chapter 2: Local Memory Mutualization Based on Badis** is set to reclaim memory from over-provisioned VMs to provide it to under-provisioned VMs. Three scientific applications, with different memory behaviors, have been studied to evaluate the effect of a system on the performance of applications.

Figure 3 illustrates our cooperative memory management system consisting of three main parts: Working set estimation technique which periodically calculates the working set size of each VM and updates the values. The memory manager adjusts the memory size of VMs according to the new working set values. If a VM is over-provisioned, then unused memory (according to the working set) is reclaimed and sent to the free memory pool. In the case of memory shortage of a VM, the memory needs can be satisfied from the free memory pool. The system guarantees that at least the VM's initially allocated memory size is allocated in case of memory shortage and some extra memory can be reallocated if the free memory pool

empty.

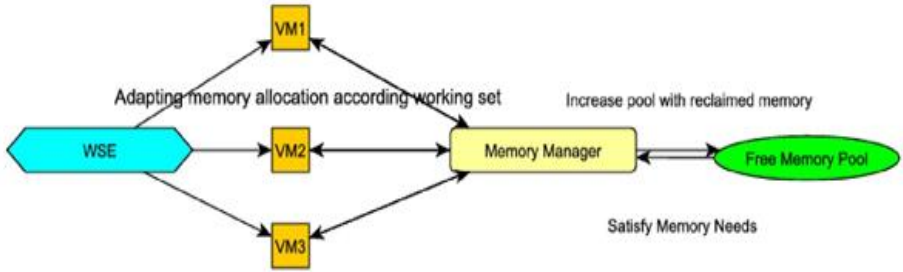


Figure 3 CMMS architecture

The system maintains 2 variables for each VM: initial allocation ( $Mem_{init}$ ) and current allocation ( $Mem_{act}$ ). The VMs are distinguished into two groups: Servers and Clients. Servers are the VMs that gave memory to the pool ( $Mem_{init} > Mem_{act}$ ) and Clients are the VMs that owe memory to the pool ( $Mem_{init} < Mem_{act}$ ).

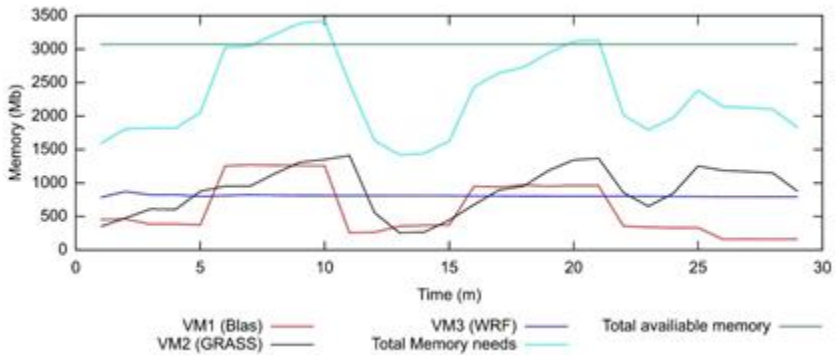
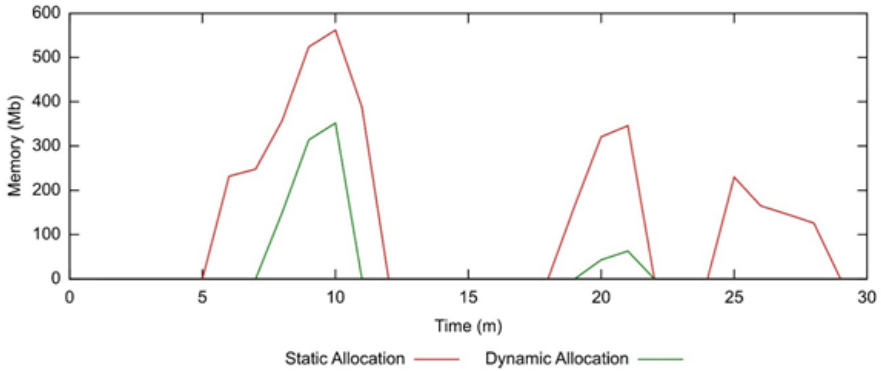


Figure 4 Memory behavior of VMs

The observations on Figure 4 show that VM1(solving linear algebra problems with BLASS library) and VM2(running geo-images processing using GRASS software) are crossing the line of 1GB several times during the running time of the applications, which is the initially allocated size of the VMs. Thus, in case of static allocation, these VMs are swapping during these periods. However, these peaks are supposed to be amortized with dynamic allocation. Furthermore, It is possible to notice that at some points, the total of the memory needs is higher than the size of the available memory on the physical machine. This means that at these points, the amount of memory in the pool is 0 and the memory management system faces a challenge of fair memory distribution when free resources are not enough. Figure 5 demonstrates that dynamic allocation may significantly reduce the amount of swapped out memory. The experiments show that by applying dynamic allocation, the amount of swapped out memory has been reduced by 4.2.



*Figure 5 Difference of swap activity in case of static and dynamic allocation*

### **Chapter 3: Memory Mutualization System For Virtualized Computing Infrastructures**

This contribution aims to improve memory management in such environments. The generally adopted approach is to monitor the working set of each VM and to reclaim weakly used memory (cold pages) without degrading the VM performance. Then, the reclaimed memory can be given to VMs with high memory requirements. However, this can only be done on a per server basis as reclaimed memory on one server can only be given to VMs running on that server. Therefore, it is mandatory to rely have to trust the placement and consolidation systems for gathering on the same server memory providing VMs and memory consuming VMs.

However, this approach is difficult to implement for two main reasons:

1. Consolidation limitations. Consolidation is known to be a NP hard problem, especially since it has to simultaneously take into account multiple resource types whose availability is

continuously varying. Therefore, it is a challenge to colocate VMs so that memory can be mutualized.

2. Infrastructure concerns. VMs' placement may be constrained by rules linked with the hardware type or with administration policies (e.g. different sub-clusters for HPC or BD applications), thus limiting the use of VM migration and dynamic consolidation.

Therefore, requiring VM colocation for memory mutualization appears to be a substantial limitation. The principle followed by the suggested contribution is to make the reclaimed memory accessible remotely.

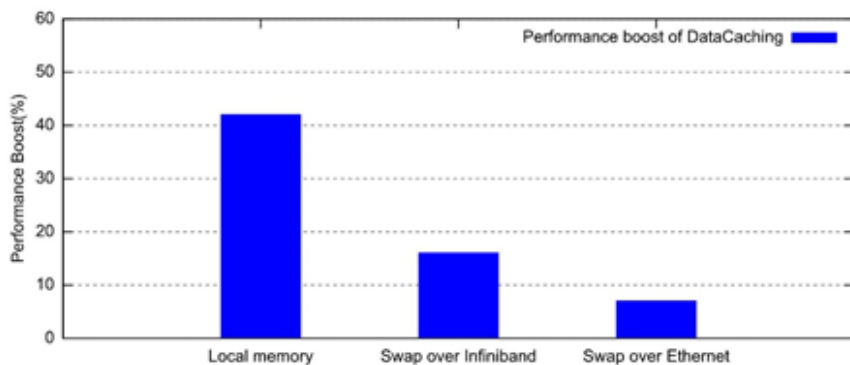
It is observed that VMs in HPC clusters are mainly CPU bound and their memory consumption is quite stable, allowing memory to be reclaimed to provision the memory reservoir. Most applications in BD clusters are memory and IO bound and can significantly benefit from extra memory from the memory reservoir.

Physical machines act either as a Client (memory consumer) or a Server (memory provider). A client machine can benefit from remote memory from server machines. A machine which does not use all of its memory becomes a server. A machine which requires more memory (than its capacity) becomes a client. However, every VM is guaranteed to have at least its initially allocated memory in case of memory shortage. Thus VMs and client machines can get their memory back in such cases.

The suggested system is composed of two parts: dynamic memory allocation within one node (local memory mutualization) and remote memory allocation from server machines (global memory mutualization).

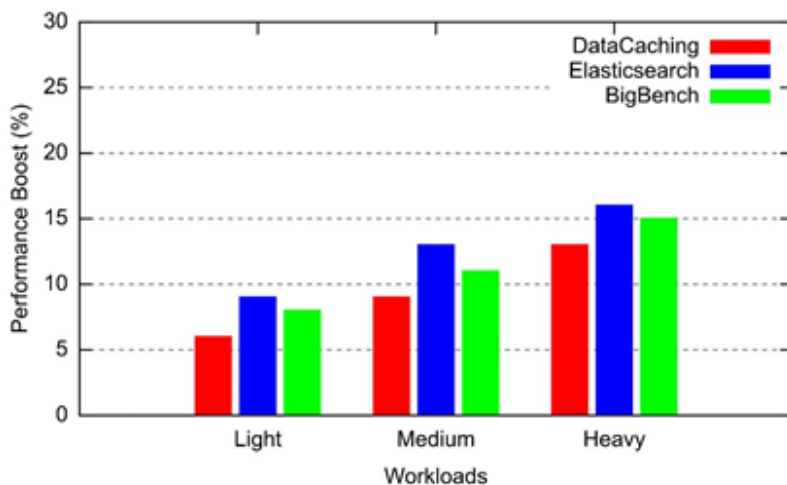
The design of our system relies on two main entities:

- A Local Memory Controller (*LMC*) is in charge of memory management within a single node. Every node (client or server) is running an *LMC*. The *LMC* manages a *Free Memory Reservoir*. This memory may be used for local or global mutualization.
- A Global Memory Controller (*GMC*) manages the coordination between machines (clients and servers). It is connected with all the *LMCs*. It implements a *Global Memory Reservoir* by federating the distributed *free memory reservoirs*. It is responsible for remote memory distribution among clients.



*Figure 6 Performance boost of DataCaching benchmark*

Fig. 6 shows the performance improvement obtained with the Data Caching benchmarks with a heavy workload. The baseline is the execution without memory extension, so that the required swap is managed on disk.



*Figure 7 Performance boot of memory intensive applications armed with our technique*

Fig. 7 shows the performance improvement for our 3 selected memory intensive benchmarks with the different workload sizes, when being provided memory extension over

Infiniband. Naturally, the improvement is proportional to the memory extension required by the workload. It is observe that the improvements are significant for all benchmarks.

The implemented platform enables to improve the memory management of HPC and BD infrastructures via dynamically monitoring the working set of each VM, aggregating this memory into a distributed memory reservoir, and making it available to requiring VMs. Microbenchmarks, memory intensive benchmarks and Big Data benchmarks were used to evaluate our contribution. The results show that remote memory mutualization can improve the performance of a standard Spark benchmark by up 17% with an average performance degradation of 1.5%.

### **Main Scientific results:**

- An accurate and non-intrusive working set estimation method has been suggested to extend the kernel of a hypervisor and a Unix like OS. [2]
- A single-node memory evaluation and distribution system has been developed allowing to reclaim unused memory from unsaturated VMs and land it to saturating VMs. [3]
- A distributed memory management system has been developed relying on remote swapping technologies for DC-wide memory mutualization. [1,4]

### **Publications:**

- 1) Hrachya Astsatryan, Wahi Narsisian, Aram Kocharyan, Georges Da Costa, Albert Hankel, Ariel Oleksiak, Energy optimization methodology for e-infrastructure providers, Concurrency and Computation: Practice and Experience Volume 29, Issue 10, Article No. e4073, 2017, DOI 10.1002/cpe.4073
- 2) Vlad Nitu, Aram Kocharyan, Hannas Yaya, Alain Tchana, Daniel Hagimont, Hrachya V. Astsatryan, Working Set Size Estimation Techniques in Virtualized Environments: One

Size Does not Fit All, Proceedings of the ACM on Measurement and Analysis of Computing Systems – SIGMETRICS, Volume 2, Issue 1, Article No. 19, 2018, DOI 10.1145/3179422

- 3) Aram Kocharyan, Boris Teabe, Vlad Nitu, Alain Tchana, Daniel Hagimont, Hrachya Astsatryan, Hayk Kocharyan, Intra-Node Cooperative Memory Management System for Virtualized Environments - 2018 Ivannikov Memorial Workshop (IVMEM), pp. 56-60, 2018, DOI 10.1109/IVMEM.2018.00018
- 4) Aram Kocharyan, Remote Swapping Mechanism Over Rdma: Speed-Up Memory Intensive Applications in Virtualized Environments, Вестник Российско-Армянского Университета, No 1, pp. 49-58, 2019



ԱՄՓՈՓՈՒՄ

Քոչարյան Արամ Սեյրանի

**ՎԻՐՏՈՒ ԱԼ ԶԱՄԱԿԱՐԳԵՐՈՒ ՄՅԻՇՈՂՈՒ ԹՅԱՆ  
ԿԱՌԱՎԱՐՈՒ ՄԸ**

**Խնդրի որվածքը.**

Վիրտուալիզացիան ամպային հաշվարկային մոդելի հիմնարար տարր է, որը ներառում է ամպային հաշվարկային միջավայրում օգտատիրոջ պահանջները բավարարող տարբեր ծառայություններ: Այս տեխնոլոգիան թույլ է տալիս ծառայություն մատուցողներին համախմբել սերվերները մեկ ֆիզիկական հանգույցում, որպես Վիրտուալ Մեքենաներ (ՎՄ), որը նվազեցնում է օգտագործվող սարքավորման քանակը: Ամպային հաշվարկային մոդելը և վիրտուալիզացիան տրամադրում է ըստ պահանջի ռեսուրսների բաշխման մեխանիզմը որը ապահովում է համակարգչային համակարգի ռեսուրսների հասանելիությունը, հատկապես՝ բարձր կատարողականություն հաշվարկային (HPC) և բարձր արագություն հիշողության պահեստավորման ռեսուրսներ՝ հաշվարկների աճող պահանջարկը և տվյալների մեծ ծավալը բավարարելու համար: Ի լրումն օդորակման և սառեցման սարքավորումների, այսպիսի համակարգչային համակարգի ռեսուրսները հանդիսանում են էներգիայի սպառման հիմնական աղբյուրը ամպային ենթակառուցվածքներում: Տվյալների կենտրոնների (ՏԿ) էներգիայի սպառումն աճում է մի քանի պատճառով, ինչպիսիք են՝ տվյալների ծավալի մեծացումը կամ ավելի մեծ բարձր կատարողականություն հաշվարկային օբյեկտների պահանջարկը, որն իր հերթին հանգեցնում է լուրջ բնապահպանական խնդիրների (այդ թվում՝ էլեկտրոնային աղբը և CO2 արտանետումները):

Տվյալների կենտրոնների էներգիայի սպառումն արդեն հասնում է գլոբալ սպառման 8-10%-ին, իսկ ընդհանուր գլոբալ տարածքում CO2 արտանետումների 2%-ը: Այսպիսով, էներգիայի սպառման նվազեցումը կարևոր դեր կխաղա այս կենտրոնների ընդհանուր էներգիայի սպառման նվազեցման համար:

Ռեսուրսների կառավարումը (ՌԿ) կենսական է ՏԿյուրաքանչյուր պրովայդերի համար, ում ուշադրություն կենտրոնում է գտնվում ամպային ռեսուրսների արդյունավետ համօգտագործումը՝ բազմակի օգտվողների շրջանում: Ռեսուրսների օպտիմալացումը ՏԿ-ների էներգիայի սպառման նվազեցման ամենաարդյունավետ եղանակներից մեկն է: ՌԿ-ի նպատակն է ապահովել նույն ծառայությունը քիչ ռեսուրսներով, բայց միևնույն որակով (առանց ծառայության մակարդակի համաձայնագրի (SLA) խախտման): Պրոցեսորը (CPU), օպերատիվ հիշողությունը (RAM), հիմնական հիշողությունը (սկավառակի տարողունակություն) և ցանցը ՏԿ-ի հիմնական ռեսուրսներն են:

Անցած մի քանի տարիների ընթացքում, ամպային միջավայրում ռեսուրսների օպտիմալացման մարտահրավերները մեծ ուշադրություն են արժանացել հետազոտողների կողմից: Այնուամենայնիվ, չափազանց բարդ է թիրախավորել միանգամից բոլոր ռեսուրսները: Այսպիսով, կատարվել են բազում հետազոտություններ պրոցեսորների ռեսուրսների օպտիմալացման ոլորտում, քանի որ այն համարվում է ամենաթանկ հիմնական ռեսուրսը ՏԿ-ում: Տարիների ընթացքում, սակայն, իրավիճակը փոխվել է: Անցած տարիների ընթացքում հիշողության աճի պահանջների հետմեկտեղ մենք տեսել ենք նոր հավելվածների առաջացման պահանջ, մինչդեռ սարքավորումների պլատֆորմի էվոլյուցիան շարունակել է առաջարկել ավելի շատ պրոցեսորներին, քան հիշողության

հնարավորություններ, որը դիտվում է որպես հիշողության հնարավորության պատ:

**Նպատակը և խնդիրները.**

Այս

ատենախոսության նպատակն է մշակել հիշողության կառավարման և օպտիմալացման միջավայր վիրտուալիզացիայի համար, որոնք օգտագործվում են Unix սերնդի օպերացիոն համակարգի (ՕՅ) միջուկների կողմից: Յետագոտությունները առաջադրել են հետևյալ խնդիրները.

- Մշակել ճշգրիտ և համակարգի աշխատանքը չծանրաբեռնող աշխատանքային բազմության գնահատման մեթոդ, որը նախատեսված է ՎՄ-ի աշխատանքային բազմության հետևելուն:
- Մշակել մեկ հանգույցի հիշողության կառավարման համակարգ, օգտագործելով վերը նկարագրված աշխատանքային բազմության գնահատման մեթոդը:
- Մշակել բազմահանգույց հիշողության կառավարման օպտիմալացման միջավայր վիրտուալացիայի համար:
- Մշակել և իրականացնել ծրագրային լուծումներ վերը նկարագրված նպատակների հասնելու համար:

**Մեթոդներ.**

Հիշողության ռեսուրսների կառավարման ընդլայնման և տվյալների կենտրոնների օպտիմալացման համար օգտագործվել են հետևյալ մեթոդները.

- Linux միջուկի մոդուլի ծրագրային մշակում, որի նպատակն է ապահովել հիշողության կառավարման հարմարեցված համակարգեր
- Լայնորեն կիրառվող հիշողության կառավարման, միգրացիոն և կոնսոլիդացիոն (ինչպիսիք են հիշողության վերաբաշխման, ՎՄ միգրացիան, ՎՄ կոնսոլիդացիան և այլն)

- Բաշխման հիշողության տարածման մեխանիզմներ (RDMA-ի տեխնոլոգիաներ)
- Ընդունված բենչմարքիև գործիքներն ախտատեսված HPC-ն տվյալների հետի նտեսնսիվաշխատանքի ծանրաբեռնվածության համար

Յետագոտնության արդյունքում .

- Առաջարկվել է ճշգրիտ և համակարգի աշխատանքը ծանրաբեռնող աշխատանքային բազմության գնահատման մեթոդորպես hypervisor-ի և Unix-երնդի ՕՏ միջուկի հավելված :[2]
- Մշակվել է հանգույցի հիշողության գնահատման և կառավարման համակարգ, որը թույլ է տալիս հետկանչել չօտագործված հիշողությունը թերբեռնված ՎՄ-ներից և տրամադրել գերբեռնված ՎՄ-ներին :[3]
- Մշակվել է բազմահանգույց հիշողության կառավարման միջավայր որը թույլ է տալիս կատարել հիշողության վերաբաշխում S4-ների մակարդակով :[1,4]

## РЕЗЮМЕ

Кочарян Арам Сейранович

### УПРАВЛЕНИЕ ПАМЯТЬЮ В ВИРТУАЛЬНЫХ СИСТЕМАХ

**Постановка проблемы:** виртуализация - это основополагающий элемент облачных вычислений, которая включает в себя различные сервисы, которые могут удовлетворить требования пользователей в среде облачных вычислений. Эта технология позволяет поставщикам услуг объединить серверы в один физический узел в виде виртуальных машин, уменьшая количество используемого оборудования. Парадигма облачных вычислений использует виртуализацию и обеспечивает доступность ресурсов компьютерной системы по требованию, особенно высокопроизводительных вычислительных (ВПВ) и высокоскоростных ресурсов большой емкости, для удовлетворения растущих потребностей в вычислениях и большого объема данных.

В дополнение к оборудованию для кондиционирования и охлаждения воздуха, такие ресурсы компьютерной системы являются основным источником энергопотребления в облачных инфраструктурах. Потребление энергии центрами обработки данных (ЦОД) увеличивается из-за нескольких аспектов, таких как увеличение объема данных для обработки или потребность в большем количестве объектов высокопроизводительных вычислений, которые в свою очередь приводят к

серьезным экологическим проблемам (включая электронные отходы и выбросы CO<sup>2</sup>). Энергопотребление ЦОД уже возрастает до 8-10% от мирового потребления, а общий глобальный след составляет 2% от глобальных выбросов CO<sup>2</sup>. Таким образом, снижение энергопотребления будет играть важную роль в снижении общего энергопотребления этих центров.

Управление ресурсами (УР) имеет решающее значение для каждого поставщика ЦОД, который сосредоточен на эффективном распределении облачных ресурсов между несколькими пользователями. Оптимизация ресурсов является одним из наиболее эффективных способов минимизации энергопотребления ЦОД. Управление ресурсами стремится предоставить ту же услугу с меньшими ресурсами и с тем же качеством (без нарушения Соглашения об уровне обслуживания (SLA)). Процессор, оперативная память, память (дисковое пространство) и сеть являются основными ресурсами центра обработки данных.

В последние несколько лет проблемам оптимизации ресурсов облачной среды исследователями уделялось большое внимание. Тем не менее, чрезвычайно сложно нацеливаться на все типы ресурсов одновременно. Поэтому, в этой области ведется много работ по оптимизации ресурсов процессора, поскольку он считается самым дорогим и основным ресурсом в ЦОД. Однако ситуация изменилась за эти годы. За последние годы мы стали свидетелями появления новых приложений с растущими требованиями к памяти, в то время как эволюция аппаратных платформ продолжала предлагать больший прирост процессорной мощности, чем памяти, называемой стеной емкости памяти.

**Цель и задачи:** Целью данной диссертации является разработка среды управления и оптимизации памяти для виртуализированных центров обработки данных, реализованных как ядра операционной системы (ОС) семейства Unix. Исследования преследуют следующие цели:

- Разработать точный и неинтрузивный метод оценки рабочего набора, ориентированного на отслеживание размера рабочего набора виртуальной машины.
- Разработать одноузловую систему управления памятью с использованием вышеописанного метода оценки рабочего набора.
- Разработать многоузловую среду управления и оптимизации памяти для ЦОД.
- Разработать и внедрить программные решения для вышеописанных целей.

**Методы:**

Для расширения управления ресурсами памяти и их оптимизации в центрах обработки данных были использованы следующие методы:

- Разработка модулей ядра Linux с целью предоставления настраиваемых ядер с поддержкой управления памятью.
- Широко используемые методы управления памятью, миграции и консолидации (такие как раздувание памяти, миграция виртуальных машин, консолидация и т. д.).
- Распределенные механизмы совместного использования памяти (удаленная замена через RDMA).

- Хорошо известные эталонные тесты для интенсивной работы с данными и рабочих нагрузок ВРВ.

**В результате исследований:**

- Предложен точный и не перегружающий работу системы метод оценки рабочего набора как приложение для расширения ядра гипервизора и ядра операционной системы (ОС) семейства Unix. [2]
- Была разработана система оценки и управления памятью виртуальной машины, позволяющая восстановить неиспользованную память незагруженных и направлять ее на перегруженные виртуальные машины. [3]
- Была разработана многоузловая среда управления памятью, позволяющая провести перераспределение памяти по всем центрам обработки данных (ЦОД). [1,4]

A handwritten signature in black ink, appearing to be 'Игорь', is written over a faint, light-colored rectangular stamp or watermark.