# On Optimizing Resource Allocation and Application Placement Costs in Cloud Systems

Cihan Seçinti and Tolga Ovatman
*Department of Computer Engineering, Istanbul Technical University, Maslak, Istanbul, Turkey*

Keywords: Virtual Machine Resource Allocation, Application Live-migration, Utilization Optimization.

Abstract: Resource utilization problem has been widely studied for cloud systems where a number of virtualized resources are shared among applications hosted as services. Resource utilization optimization can be confronted at two levels: allocating resources to virtual machines(VM) across physical machines or assigning applications to virtual machines present in the host. With the improving capabilities on virtualization technologies, realizing resource allocation at both levels is becoming more viable using VM reconfiguration and live-migration of applications. In this paper we investigate applying resource allocation optimization at these two levels and the emerging trade-off in deciding the appropriate technique to be used. We first analyze the effect of gradually increasing the amount of resources assigned to a virtual machine using VM reconfiguration and compare our results with fully assigning host's resources without reconfiguration. Later, we investigate the amount of utilization revenue when application live-migration is used for applications having smaller/larger performance needs. Finally, we compare the host utilization for different amounts of cost rates between live-migration and reconfiguration. Consequently our analysis results identify the cost rate and application granularity levels where it is optimal to apply live-migration or VM reconfiguration.

## 1 INTRODUCTION

The tremendous increase in employment of cloud systems through the recent years triggered the evolution of software towards cloud services. As the technological advancement continue improving the capabilities of virtualization and application hosting, it became viable to develop more efficient application placement strategies that can be used in provisioning cloud systems. Latest developments in virtual machine(VM) technology and application live-migration techniques added new dimensions to the problem of managing hosts' resources effectively.

Virtual machine(VM) reconfiguration can be described as the process of changing the amount of resources(e.g. CPU, memory, bandwidth, storage) assigned to the VM by the host(Verma et al., 2010). VM reconfiguration can simply be performed by shutting down the VM and tuning the amount of resources allocated for the VM. However some resources, like bandwidth and storage, are reconfigurable without shutting down VM and additionally recent advances are enabling "hot add" features for CPU and memory reconfiguration. This situation reveals on-the-fly resource allocation problem between the host and VM layers in a cloud system.

On the other hand, application live-migration is another development supported by most of the VM managers which enables moving a running application from a virtual machine to another without disconnecting the client or application(Clark et al., 2005). Application live-migration includes warm-up, copying and memory migration phases and requires a substantial amount of VM down-time during live-migration process. Being able to migrate applications with different levels of SLA requirements also reveals the problem of resource allocation by moving applications between VMs to utilize the host more effectively.

Although the resource management problems described above are encountered at different layers in the cloud system, both situations directly effect the resource utilization of the host. Moreover they both have their own costs(e.g., downtime, processing) which may change according to the technology and/or VM type being used.

In this paper, we investigated the resource allocation problem in both layers and reveal the situations where it is meaningful to apply VM reconfiguration or application live-migration. For VM reconfigura-

Seçinti C. and Ovatman T..
On Optimizing Resource Allocation and Application Placement Costs in Cloud Systems.
DOI: 10.5220/0004849605350542
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 535-542
ISBN: 978-989-758-019-2

tion we compared the case where available resources are assigned equally among the available VMs and the resources needed by a VM is gradually assigned to the VM as the applications continue to be deployed. We also investigated the effect of amount of initial resources assigned to VMs over the amount of final utilization of the host. We found out an initial resource allocation threshold can be identified where the final utilization starts to decrease as the initial resources assigned to each of the VMs start to increase.

We also analyzed the effect of application live-migration over the final utilization of the host with respect to the granularity of the applications being deployed to the VMs. We used round robin placement and mixed integer linear programming(MILP) in deciding the placement of applications having resource requirements inside certain intervals. This way we analyze if the effect of live-migrating smaller/bigger applications make any difference in utilization of the host.

Figure 1 summarizes the approach being used throughout the paper. The provisioning engine is responsible of resource allocation to virtual machines and carrying on application placement decisions. As increasing number of applications are being submitted to the host machine provisioning engine may migrate applications across VMs to generate additional resources. We use MILP to check for a possible applcation migration scheme that can discover necessary amount of resources. If this solution doesn't work additional resources can be assigned from the host to VMs if present. Our further analysis focus on making the decision of application migration versus ressource addition to VMs by reconfiguration. The tradeoff is analyzed regarding different rates of costs between migration and reconfiguration processes.

As mentioned above both of the resource allocation activities(migration and reconfiguration) described above have their own costs when applied. In our last stage of experiments we investigated the effect of different cost rates between these activities and discussed the thresholds when it becomes meaningful to apply each of these activities. Our results reveal the certain situations that can be used in provisioning process during the decision of using VM reconfiguration or live-migration technologies. Also by analyzing the trade-off between applying these technologies under certain cost rates we were able to explore conditions that can aid in deciding in which layer(host-VM, VM-application) it is meaningful to apply on-the-fly resource allocation.

In the next section, we continue by summarizing the related work in the literature. In Section 3 we explain the trade-off between VM resource allocation versus application live-migration in more detail and present the MILP model used in decision of application migrations in more detail. Section 4 presents the details of the experiments performed and a discussion of the results. Finally in Section 5 we conclude the study and present possible future directions.

## 2 RELATED WORK

Resource allocation is an important concept on systems that exhibits frequent application deployment. Nowadays, with development of cloud systems resource allocation regain importance on the problem of increasing utilization of the servers that are being run on the cloud infrastructure. Even before the advances in cloud computing resource allocation is vastly studied for clusters and data centers(Bennani and Menasc, 2005; Zhu and Singhal, 2001).

Migration algorithms can be applied to increase utilization of systems having a diverse range of characteristics. There were several studies based on increasing live-migration performance. In a recent study(Kikuchi and Matsumoto, 2012), live-migration is considered to be a serious factor in causing the response time degradations for applications. Thus, live-migration operations should be controlled to minimize the effect on the performance of applications running on the cloud system. In the study, the impact of transmission control protocol's behavior on live-migration is experimented per se to estimate the best live-migration strategy to be used. In our study instead on performing complicated estimation computations we have focused on detecting thresholds based on relative ratios which allows to perform decisions before the deployment process in a simple way.

Following the improvements of the cloud systems, live-migration is being used to increase utilization of the virtual machines on the cloud infrastructure and different algorithms are developed to increase usability of live-migration method. Yang(Yang et al., 2011) considers the different VM(and hence application) migration strategies adopted by the host machines that have different load states by taking into account four different resource types. The migration is done considering the amount of resource allocated to each VM, however the amount of resource utilization has not been taken into account. In a more recent study(Wang et al., 2013), migration is performed regarding the queue model of the time of application deployment requests from the clients. A centralized control management mechanism has been defined to inform the client which server is available at a time. However, Wang et al. assumed each request to be asigned to a

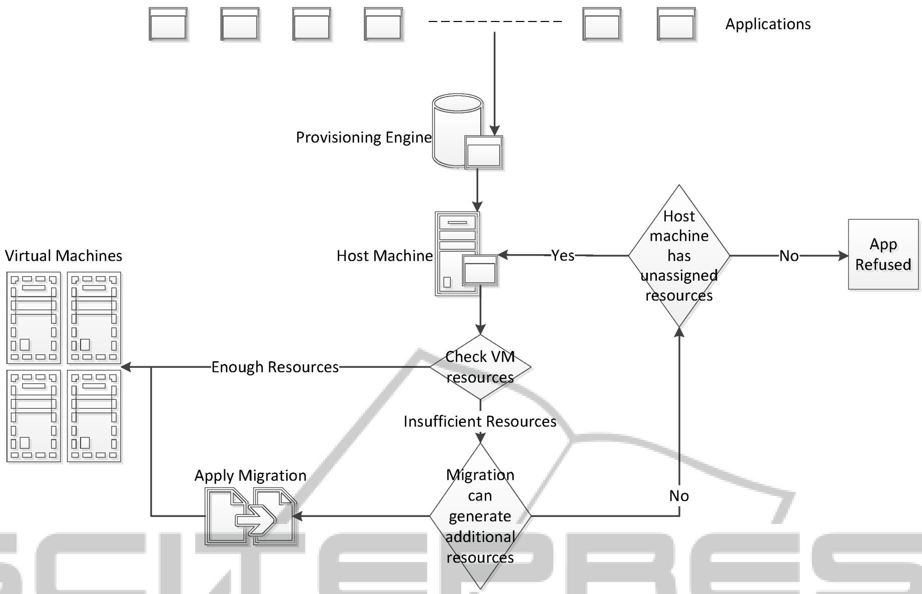


Figure 1: Overall MILP based provisioning process.



new virtual machine which ignores multiple applications deployed to a common virtual machine.

On the other hand, cloud reconfiguration research is mostly based on physical machine reconfiguration. In this particular area, the challange is more focused on figuring out how to place virtual machines to physical machines. In a recent study by He et al. it is claimed that utilization of the physical machines would be increased when the reconfiguration is applied to physical machines(He et al., 2012). In their study they focused on virtual machines placement on physical hosts rather than taking into account application placement as well.

There are a few studies which examine the tradeoffs between reconfiguration and migration technologies. A study by Chen et al. proposes a framework that uses live-migration to reduce cost of resource reconfiguration(Chen et al., 2012). They conduct experiments to illustrate the performance of the framework in maximizing utilization and reducing cost of the runtime reconfiguration. Also they propose a two-level runtime reconfiguration strategy to reduce virtual machine live-migration and shorten the total live-migration time. Earlier studies by Verla et al. examine the performance impact of reconfiguration in cloud systems and build a model to predict the duration and performance effects of such activities (Verma et al., 2010; Verma et al., 2011).

In our study we also focus on maximizing performance by considering the effect of migration and reconfiguration under different VM and application resource consumption characteristics. We have adopted the strategy to determine resource requirements of applications in a random way limited inside certain intervals. In the study by Hwang et al. it is considered that the exact resource requirements of any application could not be found easily(Hwang and Pedram, 2012). Therefore, calculating resource requirements in a randomized way is not a trivial approach. They also argue that by determining application resources randomly, error calculation can be done depending on standard deviation of randomized variables. We adopt this strategy by limiting our random resource determination process under certain limits which we use to represent the characteristics of the application size. This way we were able to fuzzily quantize the application sizes in our experiments.

As a final case to mention, we have used MILP techniques in deciding the optimal placement of application among VMs in our experiments. Utilization of ILP techniques is a known approach in resource optimization of cloud systems. A study by Borovskiy et al. treats the workload distribution issue as a set partitioning problem and devise a formal approach based on integer programming techniques(Borovskiy et al., 2011). In an earlier study a similar ILP based approach was applied on workload scheduling (Van den Bossche et al., 2010). Later, Alvarez et al. applied ILP techniques in data allocation problem by taking into account stroage and computational costs (Ruiz-Alvarez and Alvarez, 2012). Although we haven't dig into a single type of resource allocation problem deeply, we have built a simple model regarding four different types of resources altogether.

# 3 RESOURCE ALLOCATION AND APPLICATION LIVE-MIGRATION

An obvious problem in assigning a host's resources to a number of VMs is fragmentation caused by varying resource requirements of applications that are going to be placed in VMs. A straightforward strategy might be allocating all the resources present in the host equally among the VMs once and for all which requires no more reconfiguration in the future. A more sensible approach can be assigning a certain amount of host resources equally to the VMs and allocating remaining resources in appropriate amounts as the VMs require more resources in order to accept incoming applications. However this solution requires a reconfiguration cost, which may include shutting down a VM for a certain amount of time.

Another solution for more effective resource allocation is to use application live-migration. In this case, a reallocation of the present applications residing in VMs needs to be computed which contains two distinct challenges. The first challenge is leveraging the cost of VM down-time during the migration process where the second one is exploring a new placement for applications that contain minimal number of application migrations. Live migration of an application can be achieved in two different ways. Each VM in the system can be affiliated with a single application so that the application migration is achieved by migrating VMs across hosts. Another possibility is to redeploy the application to another VM which alsomakes it possible for a VM to host multiple applications. Our analysis consider the second case however the first case can also be supported by minor modifications by slightly transforming the problem from application migration among VMs to VM migration among hosts.

It can be crucial to decide which strategy to be used in resource allocation by considering different aspects of the system. These aspects include resource needs of the applications at hand as well as the cost rate between VM reconfiguration and application live-migration. In our experiments we use cost ratios in detecting the optimal results since each type of cost can change over time and VM type being used in the host system. By using profiling techniques the cost rate can be determined and appropriate strategy can be chosen accordingly. Moreover a priori knowledge on the resource consumption behavior of the applications can be used in deciding any of those resource allocation approaches as well. This a priori knowledge do not need to be very detailed; identifying applications as heavily resource consuming or lighter should be enough.

Before continuing to experiments performed to analyze the mentioned trade-offs we would like to explain an important step that takes a crucial role in determining migration costs. As mentioned above, selecting the migrating applications in determining a better placement for applications that reside in the host plays a key role in determining live-migration costs. The determined placement should include minimum number of application migrations in order to achieve a reordering process with minimal cost. In our experiments we mainly focus on cost rates between migration and reconfiguration, thus we have chosen to use symbolic costs for each of the concepts and perform our analysis based on different cost rates instead of specific cost values.

In order to determine an optimal application placement scheme we have used MILP techniques. By using MILP we minimize the number of migrations when determining the place of each application to the pool of VMs. In achieving this, for each solution we compare the place(VM) of each application with its former place and try to maximize the number of applications that doesn't change their place. To perform this, we provide MILP with Equation 1 to maximize. In Equation 1 $i$ index is used to select among VMs and $j$ index is used to select among applications. Since an application is either assigned or not to a VM we used assignment matrices of boolean variables. We try to maximize the sum of old and new application-VM assignment products which produces a value of 1 if the assignment didn't change and 0 otherwise if a migration is present.

$$\sum_{i=0}^{\#VMs}\sum_{k=0}^{\#Apps}(\text{newAssign}[i][k]\times\text{oldAssign}[i][k]) \quad (1)$$

In addition we also do apply a number of constraints in order to drive the MILP to produce meaningful results. In Equation 2 we guarantee that each application has been assigned to only one VM. Additionally in Equation 3 resource needs of each application that has been assigned to a specific VM for each type of resources are summed up to be less than the available resource assigned to the VM.

$$\bigwedge_{i=0}^{\#VMs}\left(\sum_{k=0}^{\#Apps}\text{newAssign}[i][k]=1\right) \quad (2)$$

$$\bigwedge_{i=0}^{\#VMs}\bigwedge_{j=0}^{\#Res}\left(\left(\sum_{k=0}^{\#Apps}\text{newAsgn}[i][k]\times\text{resNeed}[j][k]\right)\leq\text{resAv}[i][j]\right) \quad (3)$$

In our experiments we have used four different types of resources which are CPU, memory, bandwidth and storage requirements. We also use a total of four VMs which greatly reduces the computational complexity of the MILP. A more detailed discussion of the experimental environment is presented in the following section.

# 4 EXPERIMENTS ON OPTIMIZING PROVISIONING TRADEOFF

## 4.1 Experimental Setup

Our experiments on VM reconfiguration and application live-migration initially focus on separate aspects about applying resource allocation using each technique respectively. Initially, we focus on the difference in the host utilization when application migration is used for various application resource consumption characteristics. Secondly, we try to pinpoint the amount of initial resources need to be assigned to VMs to increase the final utilization of the host. And finally we combine both of our findings and analyze when it is meaningful to use each technique regarding the cost rate between migration and reconfiguration.

During our experiments we have used some assumptions which can be listed as follows:

- We have used four kinds of resources in our simulations: CPU, memory, bandwidth and storage. Each of our applications have four constant values indicating their resource need for a specific kind of resource. These values may point to SLA requirements of the applications need to be hosted by the cloud system.
- Each application's resource needs are set independently using a uniform random distribution. On the other hand in our experiments we produce those random values inside a number of intervals resembling cloud systems with applications in different granularity levels. For instance in Figure 2, 10 different granularity levels are used where randomly produced application resource needs vary around 5, 10, 15, ... units respectively.
- An important assumption is an application can be deployed on the cloud or migrated but it is not undeployed. A more detailed simulation can be performed by producing the incoming and leaving applications using a predefined distribution. In our simulations we mainly focus on fully utilizing the host so we only handle the case where applications never leave the host machine.

Our simulations are developed using JAVA programming language. We have used the native library lp_solve and Java ILP as API to solve the MILP problem introduced earlier. We have simulated four VMs residing in a host machine which let us to run our simulations on a decent laptop. In our simulations we have used POJOs to resemble hosts and virtual machines involving integer variables(resource variables) that represent the resources being used in simulations. Applications are added to system as discrete events causing the resource variables to be decreased according to the resource consumption values of the applications. The resource consumption values are determined randomly inside specific intervals. These intervals are indicated in the result charts presented later in the paper. Heavier analysis on a large number of VMs residing in federated cloud environments can also be simulated however MILP performance exponentially decrease for such cases.

## 4.2 Experiment Results

Our first experiment consists of assigning applications having different resource needs to the equally sized VMs in a round robin manner. When using round robin placement the applications are placed in VMs following a predefined order. This order can be determined using priorities but in our case all the virtual machines are assumed to have the same priority. When placing an application, if the next VM determined by the round robin is out of resources the application will be placed to the next VM in line. This process is carried on until an appropriate VM is found, otherwise the application is rejected due to the lack of resources.

In Figure 2 we consider using MILP to select a set of applications to be migrated across VMs when round robin algorithm is unable to find enough resource for any of the VMs to satisfy SLA needs of the submitted application. X-axis represents the intervals which is used to produce random values for application resource needs and y-axis shows the improvement of the host utilization before an application is rejected because of the unavailability of resources. The values in the y-axis are the ratio between premigration and post-migration host utilization. The intervals in the x-axis is determined resembling a sliding window interval. In each interval the median of the produced values is shifted 5 units, which we believe is granular enough to analyze the variations in the utilization.

For each set of experiments applications are produced with random resource needs inside the specific interval and assigned to an available VM using round robin. In one set of experiments we have also applied MILP and application migration after round robin is

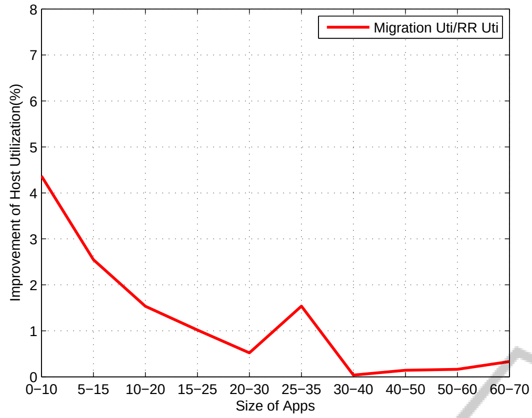


Figure 2: The improvement in host utilization when migration is used for different sized applications.

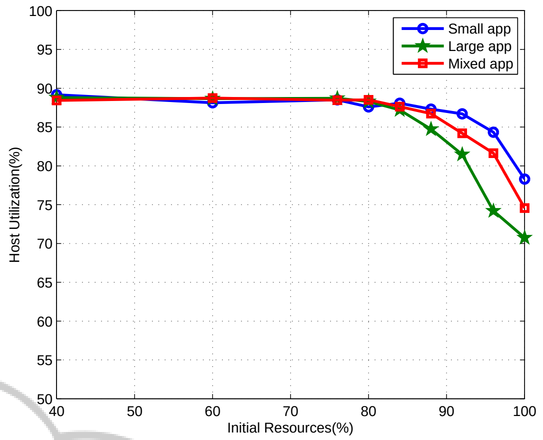


Figure 3: The impact of initial VM resource assignment over host utilization.

unable to find a VM with necessary amount of resources available. In our experiments we have used VMs having 125 units of resources available, so it can be seen that migration is applicable when the application sizes being deployed to the system is less than about 25% of VM sizes. The peak in 2535 point results from the uniform distribution around 30 unit application size being a factor of the 120 unit resource reserved for the host.

Our second experiment is to reveal the amount of initial resources that needs to be assigned to each VM in order to perform a better resource allocation. In this experiment instead of allocating all the resources equally among the VMs we start by allocating a predetermined amount of resource equally among VMs. Later when any of the VMs becomes full, instead of rejecting an application required amount of resources is granted to the VM so that the application can be deployed. This gradual increase continues until the host runs out of resources and reject incoming application. In Figure 3 we compare the amount of utilization of the host when the first application is rejected for applications with different resource needs. The applications used in the experiment can be either small(requiring 0%-25% of each resource), large(requiring 25%-50% of each resource) or randomly mixed applications of both. We can see that independent of the application size the host utilization starts to decrease after the case where 84% of initial resources are assigned.

Our final experiment compares the costs of application migration and VM reconfiguration. We have used only small applications(requiring 0%-25% of each resource) and compare the total cost for different cost rates assigned to reconfiguration and migration cases. Instead of using real cost values we rather chose to iterate over cost rates because cost values can vary significantly based on the hardware and software configurations of the system and technologied neing used. However cost rates can be used independently of these concepts because there should always be a rate between migration cost and reconfiguration cost.

In Figure 4 we indicate two thresholds which constrains the use of each concept during resource allocation process. The first threshold is the vertical initial resource allocation threshold which is the boundary when the total utilization starts to decrease, in other words it becomes less meaningful to apply reconfiguration. The horizontal threshold is the cost induced by solely using migration without reconfiguration, which marks the boundary where it becomes less feasible to apply migration and reconfiguration at the same time. A logical decision is to use the lower left region in the figure to use both concepts collaboratively.

Finally we would like to compare the performance of MILP in terms of host utilization and execution time for different number of VMs andusing small scale applications. In Figure 5 utilization achieved by applying MILP based replacement of the applications is compared with the execution time based performance of MILP. As expected, MILP performance is degraded as the number of VMs increase since there exist a larger number of possibilities to be checked during optimization process. However because of the same reason the optimal placement of applications on the VMs increases the host utilization further as the number of VMs increases. There exist a larger number of possible VMs to place the applications upon causing a better application scheme to be found by the MILP.

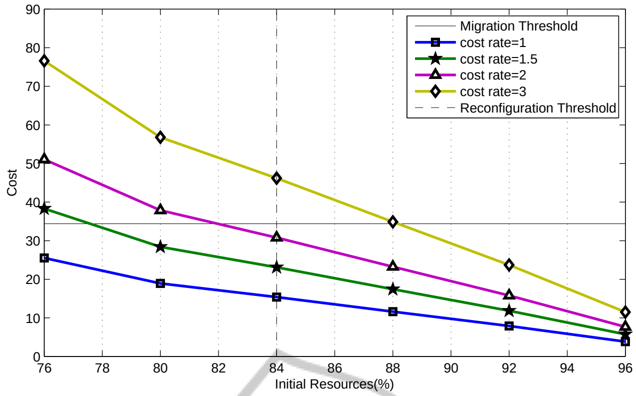


Figure 4: The trade-off between cumulative migration-reconfiguration cost and initial resource assignment for different cost rates between migration and reconfiguration.

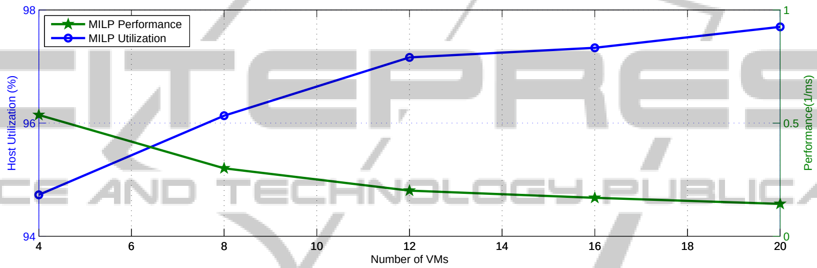


Figure 5: Utilization rate versus performance of MILP optimization for different number of VMs.



## 4.3 Evaluation and Analysis

By the experiment results explained above we have provided guidelines to be used in VM reconfiguation and application live-migration in cloud systems. By analyzing our first set of results in Figure 2 we have revealed an interval of 0% to 25% for application resource needs where using live-migration increases the total utilization of the host. This result makes sense because as the size of the application becomes larger it is less likely to allocate a large enough resource space from the leftovers in the VMs. By profiling or estimating resource need characteristics of the applications to be deployed to cloud system it is possible to decide on live-migration use strategy where such a strategy makes difference where smaller applications are dominant.

On the other hand a similar analysis can be made for reconfiguration purposes by examining Figure 3 as well. It is clear that instead of fully assigning all the resources equally to the VMs at hand, it is more preferable to assign around 80% of the resources and gradually reconfigure each VM according to needs. An interesting fact can be the stable utilization rate until 80%'s which may be tuned by taken the reconfiguration cost into account since assigning smaller amount of resources initially will clearly require a larger amount of reconfiguration.

Taking into account the costs of reconfiguration and live-migration techniques Figure 4 identifies a region where it is possible to leverage the trade-off between applying these two techniques. Since the cost of each operation can change from system to system we have chosen to use different cost rates in our analysis. We can interprete the regions formed by the thresehold values in the figure as follows: the upper left quadrant contains the cases where it is advantageous to perform configuration but the cost of using reconfiguration and live-migration together exceeds the cost of using migration solely. In this case it is meaningful to chose and use only one of the techniques at all. The lower left quadrant is the ideal case where both of the techniques can be used collaboratively. For the lower right quadrant using live-migration is preferrable since gains by reconfiguration starts to decrease. However, it can still be meaningful to use reconfiguration as well for lower percentages of initial resource allocation regarding the cost rates. The upper-right quadrant is the area where using solely live-migration should be preferred.

## 5 CONCLUSIONS AND FUTURE WORK

Advances in the cloud resource allocation technologies led to apply resource allocation strategies at different layers of a cloud system. In this paper we investigated resource allocation challenges at two different levels, more specifically between host and VM layer and between VM and application layer. Our analyses show that when the resource consumption of applications gets larger it becomes less meaningful to apply application live-migration at VM layer. Moreover it also becomes less feasible to use VM reconfiguration as the amount of initial resource assigned to each VM start to go beyond 84%. In deciding the level to apply resource allocation strategy VM reconfiguration is preferable when the amount of cost rate between reconfiguration and live migration is below 2. Moreover, the optimal cost is observed when an initial resource assigned to each VM is inside 80%-84% area. As a future study we plan to realize the cloud system environment in an actual testbed and replicate our results in actual workloads. Additionally we plan to extend our approach to support horizontal scaling and multiple hosts which require more advanced(even non-linear) optimization technniques.

## ACKNOWLEDGEMENTS

This work is performed in joint with "mCloud" project of Simternet Iletisim Sistemleri Reklam San. ve Tic. Ltd. Sti. "mCloud" project is supported by The Scientific and Technological Research Council of Turkey(TUBITAK)-TEYDEB project number 7130115. This work is also partly supported by ITU HP Cloud Computing Center.

## REFERENCES

Bennani, M. N. and Menasc, D. A. (2005). Resource allocation for autonomic data centers using analytic performance models. In *Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on*, pages 229–240.

Borovskiy, V., Wust, J., Schwarz, C., Koch, W., and Zeier, A. (2011). A linear programming approach for optimizing workload distribution in a cloud. In *CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 127–132.

Chen, W., Qiao, X., Wei, J., and Huang, T. (2012). A profit-aware virtual machine deployment optimization framework for cloud platform providers. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 17–24.

Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., and Warfield, A. (2005). Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pages 273–286, Berkeley, CA, USA. USENIX Association.

He, S., Guo, L., Ghanem, M., and Guo, Y. (2012). Improving resource utilisation in the cloud environment using multivariate probabilistic models. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 574–581.

Hwang, I. and Pedram, M. (2012). Portfolio theory-based resource assignment in a cloud computing system. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 582–589.

Kikuchi, S. and Matsumoto, Y. (2012). Impact of live migration on multi-tier application performance in clouds. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 261–268.

Ruiz-Alvarez, A. and Alvarez, M. (2012). A model and decision procedure for data storage in cloud computing. In *Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on*, pages 572–579. IEEE.

Van den Bossche, R., Vanmechelen, K., and Broeckhove, J. (2010). Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pages 228–235. IEEE.

Verma, A., Kumar, G., and Koller, R. (2010). The cost of reconfiguration in a cloud. In *Proceedings of the 11th International Middleware Conference Industrial track*, Middleware Industrial Track '10, pages 11–16, New York, NY, USA. ACM.

Verma, A., Kumar, G., Koller, R., and Sen, A. (2011). Cosmig: Modeling the impact of reconfiguration in a cloud. In *Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), 2011 IEEE 19th International Symposium on*, pages 3–11.

Wang, Y., Chen, S., and Pedram, M. (2013). Service level agreement-based joint application environment assignment and resource allocation in cloud computing systems. In *Green Technologies Conference, 2013 IEEE*, pages 167–174.

Yang, K., Gu, J., Zhao, T., and Sun, G. (2011). An optimized control strategy for load balancing based on live migration of virtual machine. In *Chinagrid Conference (ChinaGrid), 2011 Sixth Annual*, pages 141–146.

Zhu, X. and Singhal, S. (2001). Optimal resource assignment in internet data centers. In *Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 2001. Proceedings. Ninth International Symposium on*, pages 61–69.