Migrate or Not? Exploiting Dynamic Task Migration in Mobile Cloud Computing Systems

Lazaros Gkatzikis Iordanis Koutsopoulos

Abstract

Contemporary mobile devices generate heavy loads of computationally intensive tasks, which cannot be executed locally due to the limited processing and energy capabilities of each device. Cloud facilities enable mobile devices-clients to offload their tasks to remote cloud servers, giving birth to Mobile Cloud Computing (MCC). The challenge for the cloud is to minimize the task execution and data transfer time to the user, whose location changes due to mobility. However, providing quality of service guarantees is particularly challenging in the dynamic MCC environment, due to the time-varying bandwidth of the access links, the ever changing available processing capacity at each server and the time-varying data volume of each virtual machine. In this article, we advocate the need for novel cloud architectures and migration mechanisms that effectively bring the computing power of the cloud closer to the mobile user. We consider a cloud computing architecture that consists of a back-end cloud and a local cloud, which is attached to wireless access infrastructure (e.g. LTE base stations). We outline different classes of task migration policies, spanning fully uncoordinated ones, in which each user or server autonomously makes its migration decisions, up to the cloud-wide migration strategy of a cloud provider. We conclude with a discussion of open research problems in the area.

Index Terms

Mobile cloud computing, energy efficiency, multitenancy, task execution time, task migration.

- *L. Gkatzikis is with University of Thessaly (UTH) and I. Koutsopoulos is with Athens University of Economics and Business (AUEB) and the Center for Research and Technology Hellas (CERTH).*

***Primary contact author**: Lazaros Gkatzikis* ***E-mail**: lagatzik@uth.gr*

1 INTRODUCTION

Cloud computing is one of today's most rapidly evolving technologies and it is increasingly adopted by large companies to host various service platforms (e.g. iCloud by Apple, GoogleApps by Google, EC2 by Amazon, etc). A cloud facility is a network of geographically distributed datacenters, each consisting of hundreds of servers, that facilitates rapid and flexible access to a shared pool of dynamically configured resources, notably storage capacity and computational power. The two most pronounced advantages of cloud computing are the elasticity of resource provisioning and the pay-as-you-go pricing model, which enable users to use and pay only for the resources that they actually need.

Parallel to that, end-user mobility has become an essential feature of contemporary wireless networks. Mobile applications are now becoming more sophisticated than ever in terms of computing and storage requirements. Given the limited processing and energy capabilities of mobile devices, the proliferation of high-speed wireless access technologies (e.g. LTE, WiMax) provides ample promise in taking cloud computing to the next level, where mobile devices will outsource tasks to the cloud. The later asset has given rise to what is known as *mobile cloud computing* (MCC).

Virtualization is the key enabling technology of cloud computing that allows the simultaneous execution of diverse tasks over a shared hardware platform. Since each task is hosted in an isolated virtual machine (VM), collocated tasks do not interact with each other and each has access only to its own data. Virtualization provides the potential for on-the-fly and on-demand configuration of physical machines to run diverse tasks, hence avoiding resource waste.

On the other hand, encapsulation into VMs comes at a performance cost. Task consolidation brings with it increased and unpredictable *contention* for the shared system resources (e.g. CPU, caches, disks and network I/O), which in turn leads to significant performance degradation in terms of latency and execution time. This is known as the problem of *noisy-neighbours* or *multitenancy* and has been identified as a major issue in public cloud facilities [1]. Unstable and unpredictable performance due to multitenancy has been also observed when executing computationally intensive scientific tasks on commercial cloud facilities [2] and has even led companies to get off of the cloud [3].

Recently, [4] proposed that in order to best leverage cloud resources to execute mobile ap-

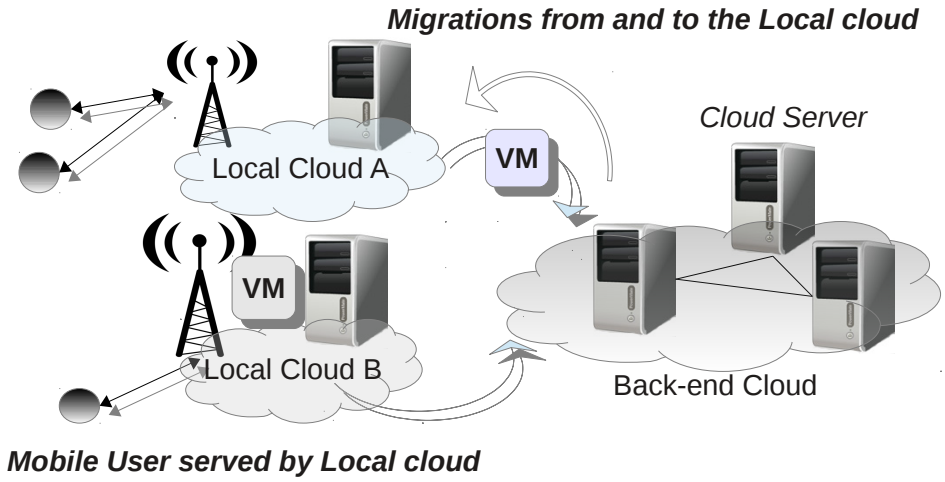


Fig. 1. An MCC system architecture that brings the cloud closer to the mobile user so as to avoid the communication latency of the Internet.

plications, we have to effectively *bring the cloud closer to the user*. In this article we consider the architecture of Fig. 1 that brings into stage fixed cloud infrastructure together with mobile wireless devices. Mobile devices access the cloud through readily available hubs like LTE base stations (BS) or WiFi access points. We consider a back-end cloud facility of servers interconnected through wire-line links, whereas the access link from the mobile device-client to the cloud is wireless. In addition, smaller-sized local clouds are attached to the points of wireless access. Local clouds are generally characterized by limited computing resources, but are directly accessible by the users and avoid the additional communication delay of the Internet. Both the local and the back-end cloud are managed by the same provider and they are connected as well.

In this article, we discuss VM migration mechanisms that enable cloud providers to adjust the load at each server at will. We delineate the performance benefits that arise for mobile applications and identify the peculiarities of the cloud that introduce significant challenges in deriving optimal migration strategies. Starting from the centralized approach where the cloud provider selects the cloud-wide optimal migration strategy, we move to distributed approaches where the decisions are made independently by each server or task. Finally, we outline open research directions that arise in the context of task migration.

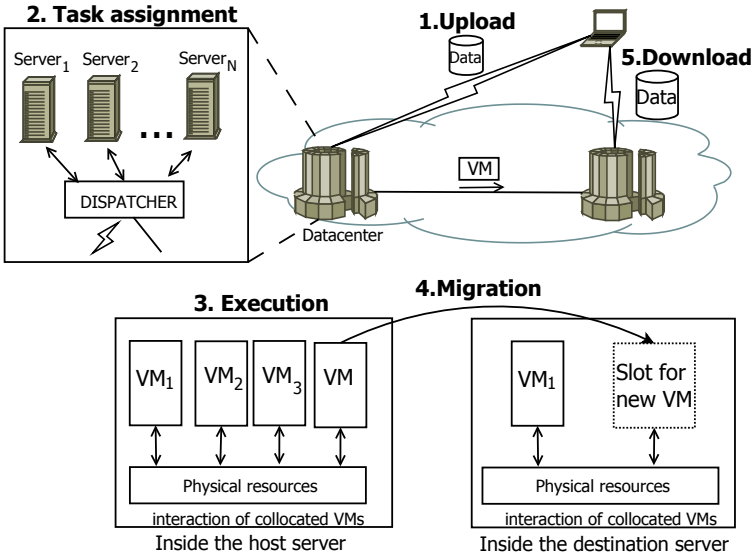


Fig. 2. The state evolution of a task during its lifetime in the cloud

2 THE LIFETIME OF A TASK IN THE CLOUD

In the context of MCC, new tasks arise continually in the cloud, as they are generated by mobile users at various locations. We use the term *task lifetime* to refer to the total time that a task spends in the cloud, including all the stages depicted in Fig. 2 and explained below:

1) **Upload**: Once a new task is generated by a user, the source code and any input data required for the initialization of the VM are uploaded to the cloud through the wireless link between the user and the corresponding point of wireless access. Then, the task is either executed at the local cloud, or its data are forwarded over the Internet to the back-end cloud.
2) **Task Assignment**: Once the required data have been uploaded to a datacenter, a local dispatcher is responsible for the assignment of the task to a specific server.
3) **Execution**: This is the actual processing within the cloud. A new virtual machine is created for the specific task at the selected server and execution starts immediately.
4) **Migration**: A VM may be transferred from its current server to a new one to continue execution there. This process is known as VM migration and may occur multiple times during task execution. For the further execution of the task, the accompanying data volume, that is required for the initialization of the new VM, has to be transferred to the new server. Migrations, though optional, may occur several times during the lifetime of a task.
5) **Download**: At this stage the mobile user retrieves the final results. We assume that once

processing is completed, the final data are immediately downloaded by the user through its current access technology. If the host server is not in the wireless range of the mobile user, data have to be transferred to an accessible server.

3 Task Scheduling and Migration in the Cloud

Once a task is issued by a user for execution in the cloud, a new VM has to be created and assigned to a physical server. This process is known as task placement (or assignment) and is performed by the VM manager. Task placement enables the cloud provider to control the load at each server up to some extent. Given that cloud is a highly volatile environment, task placement cannot ensure Quality of Service (QoS) guarantees though.

Taking the idea of task placement a step further, task *migration* has emerged as a promising option [5]. Migrations provide a more fine–tuned means of balancing the load throughout the system, since a migration may take place at any time during the lifetime of a task. However, each VM migration induces a delay that has also to be taken into account. This migration delay is defined as the time required for the VM i) to stop execution at the current server, ii) to move the accompanying data to the new one and iii) to initialize the new VM.

A cloud provider faces an inherent tradeoff in selecting the optimal migration strategy. On the one hand, it has to optimize its clients' quality of experience (QoE), which for a particular task translates into minimization of execution time. Reduced execution time allows the provider to enjoy competitive advantage over other providers. Besides, the resources of the cloud become more often free and available for other tasks, thus leading to larger residual processing capacity. On the other hand, the cloud provider aims to fully exploit task consolidation in order to reduce its operating costs (e.g. electricity cost by turning-off underutilized servers). In this article, we mainly focus on the former objective.

The question that naturally arises is when a migration should take place. In [6] it is proposed that a migration should be performed, only when the QoS requirement of a task is violated for a long enough period. As a rule of thumb, a VM migration is beneficial if the anticipated execution time at the new server is smaller than that at the current server. But where should a task migrate? In order to answer this question, information about the state of the current host and the tentative destination server is required. This includes both the number of tasks running

on each server but also their interaction.

In general, efficient VM scheduling calls for an accurate prediction of the corresponding multitenancy cost. In this direction, several works perform analytical [7] or experimental [8], [9] estimation of multitenancy effect. The former require a priori knowledge of the resource access pattern for each task in order to model contention. The latter perform extensive profiling of different types of cloud tasks. Since significantly diverse tasks exist in the cloud [10], a characterization through profiling is impractical. Instead, we suggest that multitenancy cost should be estimated through online measurements as tasks are being executed.

Finally, VMs are evolving entities; as time passes a task may generate new data, whereas others may become obsolete. Thus, the corresponding VM is characterized by a time-varying volume of accompanying data, which may be increasing, decreasing or constant with time, depending on the type of the task. For example, video compression is a typical CPU-intensive task of decreasing accompanying data pattern. Starting from an initial raw video of several gigabytes, we end up with a compressed video of some megabytes. As processing evolves, the already compressed part of the video becomes redundant, since in its place we get the much smaller compressed version. On the other hand, most of complex scientific calculations, that are uploaded for execution to the cloud, are of increasing data volume. New useful results are continuously generated and most are usually required for subsequent calculations.

Summarizing, efficient utilization of cloud computing infrastructure brings with it novel fundamental challenges that stem mainly from its highly dynamic nature, due to user mobility, the continuously evolving VM population at different servers, the time varying server processing capacity due to multitenancy, and the evolving accompanying data volume of each VM. Currently, a task is arbitrarily allocated to any server that can meet its processing and storage requirements [11]. Due to multitenancy though, VM scheduling and migrations should be performed according to the estimated performance. Additionally, in the context of MCC a VM should ”follow” the mobile user, so that latency during task execution and data download time to the mobile device are minimized.

4 Challenges in Designing Efficient Migration Mechanisms for the Cloud

Currently, cloud providers commit only to availability of their services, through Service Level Agreements (SLAs) with their clients. However, no QoS guarantees are provided, mainly due to the numerous factors that introduce uncertainty regarding the actual performance of a virtual machine. Efficient migration mechanisms though can improve exploitation of the available resources, reduce execution time and mitigate the effects of uncertainty. Efficient utilization of the cloud computing infrastructure faces the following fundamental challenges.

4.1 Workload uncertainty

The pattern of instantaneous resource demand varies with time and location, as new tasks arise continually at various locations while others complete service, leading to a highly unbalanced load distribution within the cloud. This indicates the need for mechanisms that exploit the available resources optimally by moving tasks from the overloaded servers to the underutilized ones.

4.2 Unpredictability of multitenancy effects

Availability of resources is also subject to continuous change. The actual processing capacity of virtualized cloud servers is time-varying due to the unpredictable degree of interaction of collocated VMs. Multitenancy unavoidably leads to contention for physical resources and introduces an overhead that is often task-dependent and difficult to model or predict.

4.3 Unknown evolution of accompanying data volume

Each task usually generates data whose volume varies with time. This data footprint may be increasing, decreasing or constant with time. For example, several types of tasks generate new information, others compress it as they evolve, and others do not generate new data at all. The time required to move a task from its current host to a new server depends decisively on this volume of data. Since the data footprint of a VM is not constant, selecting when a task should migrate requires knowledge of the its data evolution pattern. A migration within the same datacenter takes place over high speed Local Area Networks [5] and hence is in general less costly.

4.4 Time-varying network link capacity

The available communication capacity of the interconnection links among datacenters and the access links is time-varying. Especially, the latter that connect the mobile clients to the cloud are in general wireless and hence highly dynamic and prone to failure. Client mobility may contribute significantly to this variation. As the user moves from one location to another a different subset of servers is directly accessible, namely the ones comprising the corresponding local cloud. In general, the state of each access link is not known a priori. However, user mobility pattern dictates the most appropriate server for task uploading and data downloading, given the dynamically changing location of the user. Intuitively, as the task proceeds to completion, the migration strategy should favor the local clouds that are in range of the mobile user.

4.5 Partial availability of cloud-related information

In the cloud paradigm, the information required for identifying the optimal migration may not be available at the decision maker. The cloud provider is aware of the state of the cloud infrastructure, such as the capacity of the communication links, the available processing capacity and the number of collocated tasks. On the other hand, complexity and data volume evolution of each task is private information that is generally available only at the user side. Thus, any efficient migration scheme requires a supporting mechanism that makes all the required information available to the decision maker.

5 Different Modes of Task Migration

Cloud is a dynamically changing environment and its performance depends on numerous uncontrollable and unpredictable parameters. With this in mind, we advocate that novel online migration mechanisms are required to guarantee steady performance of mobile tasks in the cloud. Starting from the centralized approach where the cloud provider selects the migration strategy for the cloud as a whole, we move to distributed approaches where the decisions are made either by each server or even independently by each task. In order to adhere to a realistic scenario, we assume that information about the cloud dynamics is not available a priori, but rather it is presented online, just before a migration decision is taken. We take a discrete time approach, where from time to time the optimal VM migration is selected. For a given task, the

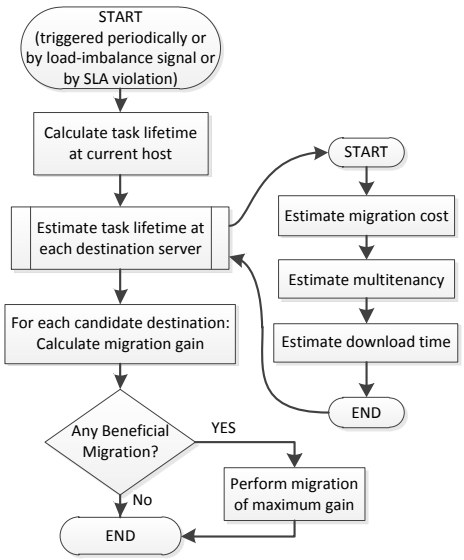


Fig. 3. Flowchart for the process of selecting the optimal destination server for a migrating task.

destination server is selected according to the mechanism described in Fig. 3, which serves as a building block of the following migration mechanisms.

5.1 Centralized migration policies

An important issue when scheduling multiple tasks in the cloud is that they compete for the same set of resources. Consider a task under tentative migration from its current host server to a new one. The migration affects the performance of all the tasks running at the current host and the destination server, since available processing capacity for each task is inversely proportional to the number of tasks running on the server. Thus, a migration leads to improved performance for the remaining tasks in the initial host, since the removal of a task alleviates both load and contention. On the contrary, the addition of a new task causes some performance degradation at the destination server, since more VMs share the same resources and the multitenancy overhead increases.

Regarding the task under migration, its performance may improve or deteriorate, depending on the load of the destination server compared to that of the current host. Obviously, any migration decision should also consider the time required for the migration itself, since within this period, generally referred to as *downtime*, the execution of the task is suspended. Finally, the time required so that the mobile user downloads the final data has to be taken into account.

From the provider point of view, an efficient migration mechanism should perform the mi-

grations that maximize the performance of the system as a whole. Thus, it should move tasks from overloaded servers to underutilized ones. Selecting the optimal migration requires a search over a three-dimensional space; for each server of the system, and for each of its active tasks the performance gain of a migration to each candidate server has to be estimated. However, instead of performing an exhaustive search, specific characteristics of the cloud can be used as a basis to reduce the search space significantly. In particular, any efficient migration strategy has to respect the following guidelines:

- **Data volume:** prioritize migrations of tasks of increasing data volume pattern. Since the migration cost of any such task increases as its processing advances, they should be preferred for migration.
- **Residual processing burden:** preferably migrate tasks of significant residual processing burden. Migrating a task that is close to completion, may not be beneficial even if the destination server is idle. In contrast, a task of substantial remaining processing time can better exploit the capacity available at the destination server.
- **Multitenancy:** preferably migrate tasks causing significant multitenancy cost. Although this cost is generally increasing in the number of collocated tasks, its exact impact is determined by the type of collocated tasks. For example multiplexing CPU-intensive tasks with network intensive ones generally reduces the multitenancy overhead [8].
- **Mobility:** perform migrations so that the task follows the mobile user, since communication delay is significantly higher when a VM is hosted in a distant server.

In this setting, all the decision making is performed by the cloud provider that is aware of the cloud infrastructure state. However, the remaining execution time and the accompanying data evolution of a task are not known a priori and hence they have to be estimated as task execution evolves.

5.2 Server-initiated migration: towards reducing the complexity of migration decision

The centralized nature and the increased complexity of the cloud-wide approach motivates the development of distributed migration mechanisms that minimize the information that has to be circulated within the cloud. In this direction, a server-level approach would enable each server to individually select which of its active tasks should migrate and where. The key underlying

TABLE 1
Characteristics of the different modes of migration

Migration policy	Decision making	Objective	Complexity
Cloud-wide	centralized	system execution time	high
Server-initiated	distributed	server execution time	medium
Task-initiated	distributed	task execution time	low

idea is that a migration should only occur if it is beneficial for the tasks hosted at the server, including the one under migration. A migration may be initiated whenever a server detects that it is overloaded compared to the average of the system or when an SLA agreement is about to be violated.

In this case, each server performs a two-dimensional search, i.e. for each active task it calculates the anticipated gain for each possible migration to a new server, and picks the one of maximum gain. The aggregate reduction of execution time of the hosted tasks and the task under migration is considered. Since the migrations are initiated by the host server, the impact of a migration on the tasks located at the destination server is unknown and hence not taken into account.

5.3 Task-autonomic migration

In an alternative distributed scenario, migrations could be initiated by the tasks themselves. Instead of delegating control to the cloud provider, each task may autonomously decide its migration strategy towards minimizing its own execution time. The key underlying idea is that a migration should only occur if it is beneficial for the anticipated execution time of the task itself, including the delay incurred by the migration. In this case, each task has to perform a one-dimensional search over the set of candidate servers.

Since the task is not aware of the state of the cloud system though, online interaction with the VM manager that monitors the load at each server and the link states is required. Besides, since each task is a self-interested entity such an uncoordinated approach could lead to extreme competition among the tasks for the least loaded servers.

The differences of the considered modes of migration are summarized in Table 1.

5.4 Evaluation of migration benefits and the impact of mobility

In order to stress the impact of mobility in migration decisions, we depict in Fig. 4 the lifetime of a task uploaded to a local cloud by a mobile user through its 3G access. We consider a task

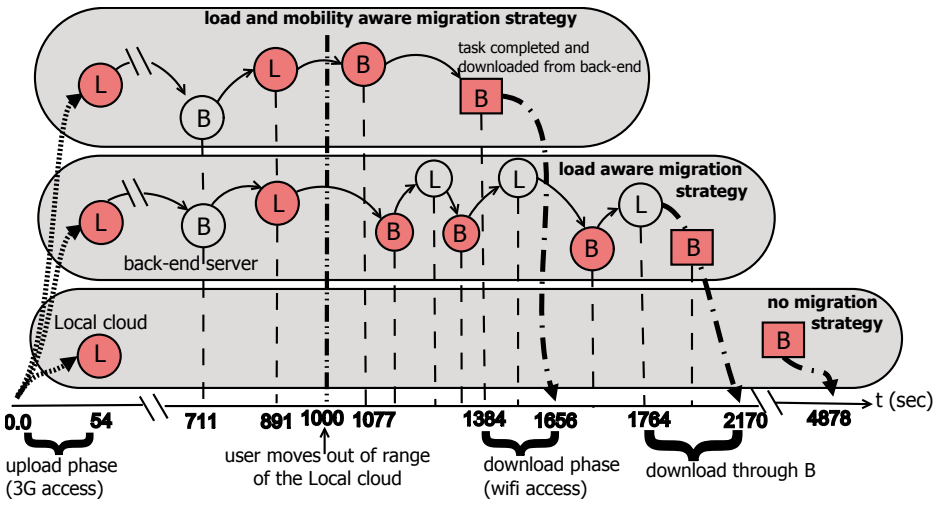
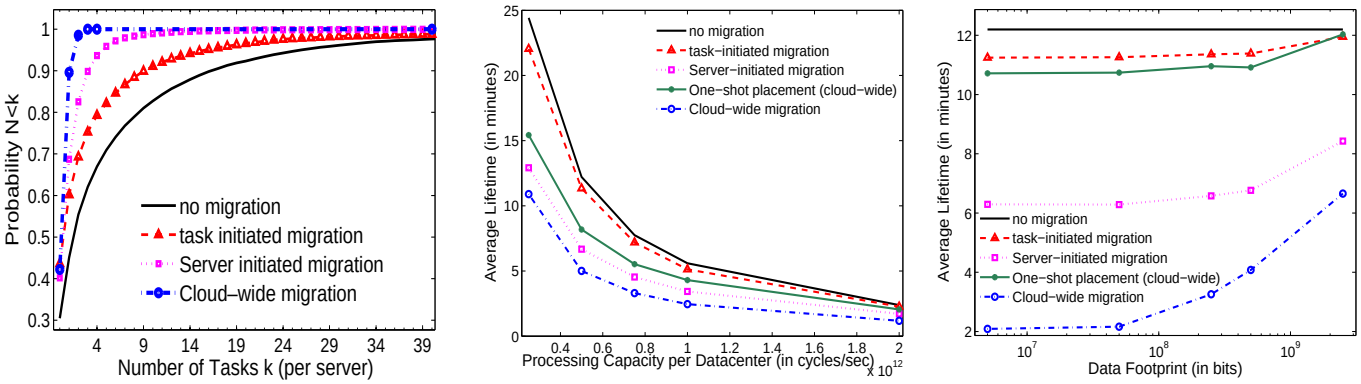


Fig. 4. Indicative migration path of a mobile task throughout its lifetime under a) a no migration strategy b) a load-aware migration strategy and c) a joint mobility and load-aware migration strategy

of increasing data footprint that can be executed either at the local (L) or the back-end cloud (B). We depict the task lifetime for the case of a) no migration, b) a strategy that does not consider migration cost and download time and c) the proposed mobility aware task-initiated migration strategy. For each time instance, we use shading to denote the closest to the user cloud facility in terms of communication delay. The no-migration strategy places the task at the local cloud and hence exhibits the worst performance. Initially, the migration cost is negligible since the data footprint of the task is small. Thus, as long as the user remains in range of the local cloud and migrations are costless, the other two migration strategies perform identically. Once the user moves out of range of the BS hosting the local cloud to a place closer to the back-end (WiFi access), the mobility-aware approach moves the task there. In contrast, the load-aware one performs several migrations to the least loaded server, in an attempt to exploit the best option in terms of available processing capacity, without considering though the increasing cost of each migration and the additional cost of downloading the final data from a distant server. Hence, it is outperformed by the mobility aware one.

Next, we quantify the performance of the different modes of task migration as tasks arise randomly in a cloud facility of 5 datacenters, each consisting of 100 servers. For comparison purposes, we depict also the performance of a one-shot placement scheme similar to our cloud-wide approach, but now the server that will host the task is selected only once, when the task arrives at the system.



(a) cdf of number of tasks per server (b) Average lifetime Vs Processing capacity (c) Average lifetime Vs Data footprint

Fig. 5. Performance evaluation of different migration policies

Initially, we investigate the expected number of tasks N hosted by each server, which is indicative of the load balancing behaviour of each approach. Since N is a random variable, we depict in Fig. 5(a) its cumulative distribution function (cdf), i.e. the probability $P[N \leq k]\ \forall k \in \mathbb{N}$. The slopes of the curves indicate how balanced the cloud is in each case. In the cloud-wide approach a server hosts at most 3 VMs. Instead, the probability of having more than 3 VMs running on a server is 10% for the server-initiated approach and 25% for the task-initiated one. Notice that in the latter case the probability of more than 20 tenants in a server is non-negligible ($\sim 4\%$); compared to the no migration case though the load is more balanced.

In Fig. 5(b) we quantify the impact of processing capacity on the average lifetime of tasks. As expected, the performance degrades as we move from the centralized approach that has system–wide information to the decentralized ones that reside only on local information. Increasing server capacity causes the performance gap of the proposed schemes to diminish, indicating that careful migration decisions are most important when the cloud is overcommitted. Interestingly, all approaches outperform the placement mechanism, except the task-initiated one. In the latter, although the average lifetime is similar to that of the no-migration case, the gain of individual tasks that exploit migrations is substantial. In Fig. 5(c) we depict the impact of data footprint on task lifetime. Low enough data footprint enables the cloud-wide approach to fully exploit the cost-free migrations, leading to significantly improved average lifetime. In contrast, performance deteriorates significantly, as the average footprint increases. In general, the "heavier" the tasks, the higher is the migration cost and hence the usefulness of migration schemes reduces.

6 FUTURE CHALLENGES AND OPEN DIRECTIONS

We now outline some open research directions that stem from task migration.

6.1 Energy efficiency considerations of task migration

An important consequence of task migration is the effective reduction of energy consumption at the cloud servers. Such an issue becomes especially critical in large-scale datacenter networks which consist of multiple dispersed server facilities.

First, the energy consumption of such a server facility depends on the number of active server machines. Further, for each active machine, energy consumption is an increasing function of the server load. Task migration can contribute to significant savings in energy consumption by (i) reducing the number of active servers through appropriate concentration of the tasks on fewer physical machines with the aid of virtualization (consolidation) (ii) reducing the energy consumption of individual servers by moving the processes from heavily loaded to less loaded servers (load balancing).

The challenge lies in that a migration has contradicting consequences due to the reasons above: by reducing the number of active servers as reason (i) dictates, the load is concentrated on fewer servers which are thus loaded more. On the other hand, by performing load balancing according to reason (ii), the heavy load of some servers is alleviated, but at the same time it is likely that more servers would need to be activated to accommodate it. The final choice will depend on the relative amounts of energy consumption of servers when they are idle and loaded, as well as on the precise dependence of energy consumption on physical machine load. An associated challenge would therefore be to derive analytical models for energy consumption.

6.2 Server load migration and integration of renewable sources

The prevalent policy in datacenter server farms is to build a collocated renewable energy source (e.g. wind turbine or photovoltaic) with the objective to minimize dependence on the main power grid. The output of these renewable sources is stochastic and hence highly time varying and unpredictable. On the other hand, the migration of tasks among different datacenters modifies the instantaneous energy consumption as outlined above, and therefore it changes the instantaneous power demand. The challenge is therefore to match the dynamic power supply

of renewable sources and dynamic datacenter power demand, by controlling the latter through migrations. A recent work along these lines is [12].

6.3 Impact of multitenancy

The recent work [13] discussed the application of max-weight inspired policies to maximize throughput through task allocation and VM configuration in dynamic cloud computing systems. The key idea is to relate the VM configuration (number of VMs assigned to each server) to the task queue evaluation rate. It would be interesting to enrich such stochastic approaches with precise models which capture the interdependence of multiple coexisting VMs in the same physical machines. Since multitenancy cost depends also on the types of collocated tasks, this is especially challenging due to the huge set of tasks in the cloud.

6.4 Modeling future cloud ecosystems

Currently, each cloud provider deploys its own infrastructure to provide efficient cloud services to its clients located throughout the world. Thus, all the decision making related to VM scheduling and migrations can be performed in a centralized way. In the near future it is expected that several providers will form coalitions, enabling access to each other's infrastructure, so as to reduce deployment cost and achieve more efficient utilization of the available resources. This way the Intercloud, the cloud of the clouds will be formed, introducing novel migration-related challenges. Indicatively, in this scenario an underlying mechanism that will coordinate the migration strategies of the providers is required.

Besides, a user may be contracted with more than one cloud providers, making possible the scenario of user-driven migrations between different providers. This scenario is captured by the autonomic task migration scheme described previously. In this case, each user has to decide whether and where to migrate according to the information announced by the cloud providers, such as the advertised residual capacities and the corresponding charges.

7 Conclusion

In this article we elaborated on efficient task scheduling in the context of MCC. We identified user mobility, data volume evolution and multitenancy as the main characteristics of the mobile cloud

paradigm that necessitate a fresh look at VM scheduling/migration. In particular, we showed that in such a dynamic environment, properly designed online migration mechanisms are required to tackle performance uncertainty and hence enable quality of service guarantees for mobile users. The crucial decision is which migration should be performed and when. Based on different design principles, we outlined three modes of online task migration that are characterized by different degree of autonomicity and complexity and discussed related open research problems.

8 Acknowledgements

L. Gkatzikis' work is co-financed by the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) "Research Funding Program: *Heraclitus II-Investing in knowledge society through the European Social Fund*".

I. Koutsopoulos acknowledges the support of ERC08-RECITAL project, co-financed by Greece and the European Union (European Social Fund) through the Operational Program "Education and Lifelong Learning" - NCRF 2007-2013.

The authors thank Dimitris Hatzopoulos for his help with numerical evaluation.

References

[1] "Has Amazon EC2 become over subscribed?" http://alan.blog-city.com/has_amazon_ec2_become_over_subscribed.htm.

[2] A. Iosup, S. Ostermann, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, pp. 931–945, June 2011.

[3] "Mixpanel: Why we moved off the cloud." http://code.mixpanel.com/2011/10/27/why-we-moved-off-the-cloud/.

[4] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," *IEEE Pervasive Computing*, vol. 8, pp. 14 –23, Oct. 2009.

[5] M. Mishra, A. Das, P. Kulkarni, and A. Sahoo, "Dynamic resource management using virtual machine migrations," *IEEE Communications Magazine*, vol. 50, no. 9, pp. 34–40, 2012.

[6] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," in *Proceedings of the 4th USENIX conference on Networked systems design & implementation*, NSDI, pp. 229–242, 2007.

[7] S.-H. Lim, J.-S. Huh, Y. Kim, G. M. Shipman, and C. R. Das, "D-factor: a quantitative model of application slow-down in multi-resource shared systems," in *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE conference*, pp. 271–282, ACM, 2012.

[8] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, "Understanding Performance Interference of I/O Workload in Virtualized Cloud Environments," in *IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pp. 51–58, Jul. 2010.

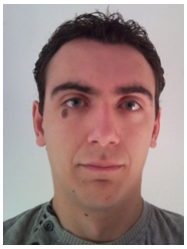
[9] Y. Koh, R. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, "An analysis of performance interference effects in virtual environments," in *IEEE International Symposium on Performance Analysis of Systems Software (ISPASS).*, pp. 200–209, Apr. 2007.

[10] A. K. Mishra, J. L. Hellerstein, W. Cirne, and C. R. Das, "Towards characterizing cloud backend workloads: insights from Google compute clusters," *ACM SIGMETRICS Performance Evaluation Review on Industrial Research*, vol. 37, pp. 34–41, Mar. 2010.

[11] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *IEEE Transactions on Services Computing*, vol. 3, pp. 266 –278, Oct.-Dec. 2010.

[12] D. Hatzopoulos, I. Koutsopoulos, G. Koutitas, and W. Heddeghem, "Dynamic virtual machine allocation in cloud server facility systems with renewable energy sources," in *Proceedings of IEEE ICC (to appear)*, 2013.

[13] S. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proceedings IEEE INFOCOM*, pp. 702–710, 2012.



Lazaros Gkatzikis (S '09, M '13) obtained the Diploma in Computer Engineering in 2006 and the M.S. degree in Communication engineering from the Department of Computer Engineering and Communications of University of Thessaly in 2008. Currently, he is a PhD student in the same department. He is also affiliated with Center for Research and Technology Hellas, Institute for Telematics and Informatics (CERTH-ITI). In Fall 2011 he was a Research Intern at the Technicolor Paris Research Lab. His research interests include optimization and game theory for mobile cloud computing and smart grids.

Iordanis Koutsopoulos (S '99, M '03, SM '13) is Assistant Professor at the Department of Computer Science of Athens University of Economics and Business (AUEB) since 2013. He obtained the Diploma in Electrical and Computer Engineering from the National Technical University of Athens (NTUA), Greece, in 1997 and the M.S and Ph.D degrees in Electrical and Computer Engineering from the University of Maryland, College Park (UMCP) in 1999 and 2002 respectively. He was Assistant Professor (2010-2013) and Lecturer (2005-2010) with the Department of Computer Engineering and Communications, University of Thessaly. He is also affiliated with Center for Research and Technology Hellas, Institute for Telematics and Informatics (CERTH-ITI).

During the summer of 2005 he was visiting researcher with the University of Washington, Seattle, USA. During his sabbatical in Fall 2012 he was a Visiting Research Scientist with Yahoo! Research Labs, Barcelona, Spain. From 1997 to 2002 he was a Fulbright Fellow and a Graduate Research Assistant with the Institute of Systems Research (ISR) of UMCP. He has held internship positions with Hughes Network Systems (HNS), Germantown, MD, Hughes Research Laboratories LLC, Malibu, CA, and Aperto Networks Inc., Milpitas, CA, in the summers of 1998, 1999 and 2000 respectively. He received the single-investigator European Research Council (ERC) competition runner-up award (co-funded by Greece and the European Union) for the project 'RECITAL: Resource Management for Self-coordinated Autonomic Wireless Networks'(2012-2015). His research interests are in the general area of network control and optimization.

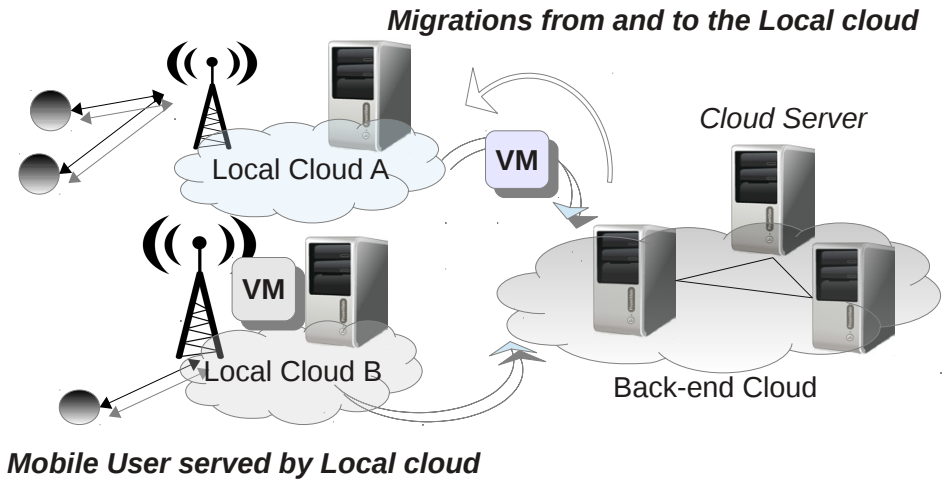


Fig. 1. An MCC system architecture that brings the cloud closer to the mobile user so as to avoid the communication latency of the Internet.

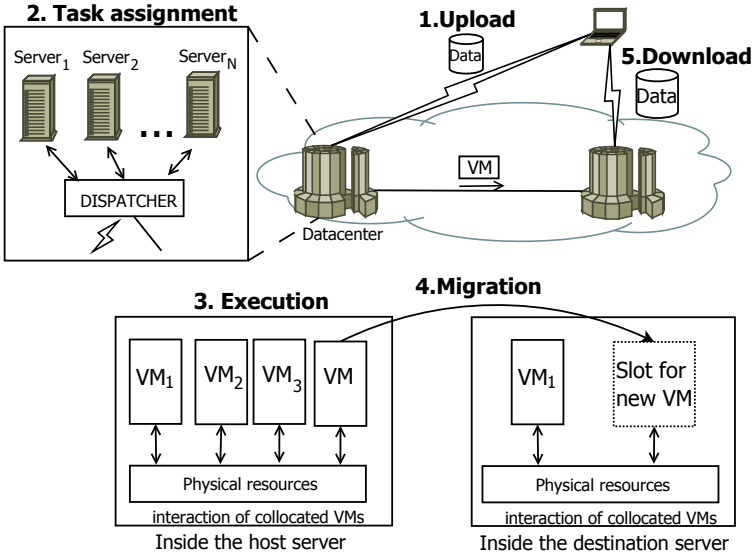


Fig. 2. The state evolution of a task during its lifetime in the cloud

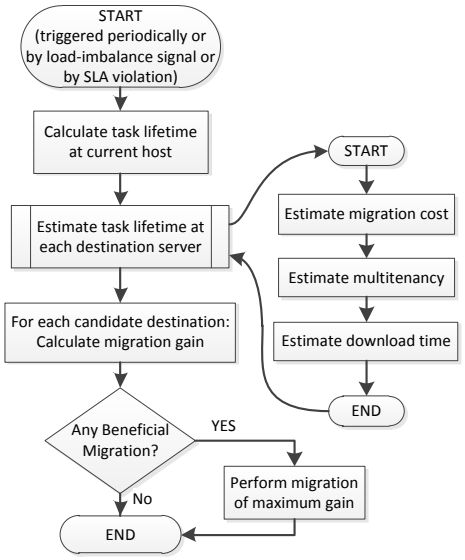


Fig. 3. Flowchart for the process of selecting the optimal destination server for a migrating task.

TABLE 1
Characteristics of the different modes of migration

Migration policy	Decision making	Objective	Complexity
Cloud-wide	centralized	system execution time	high
Server-initiated	distributed	server execution time	medium
Task-initiated	distributed	task execution time	low

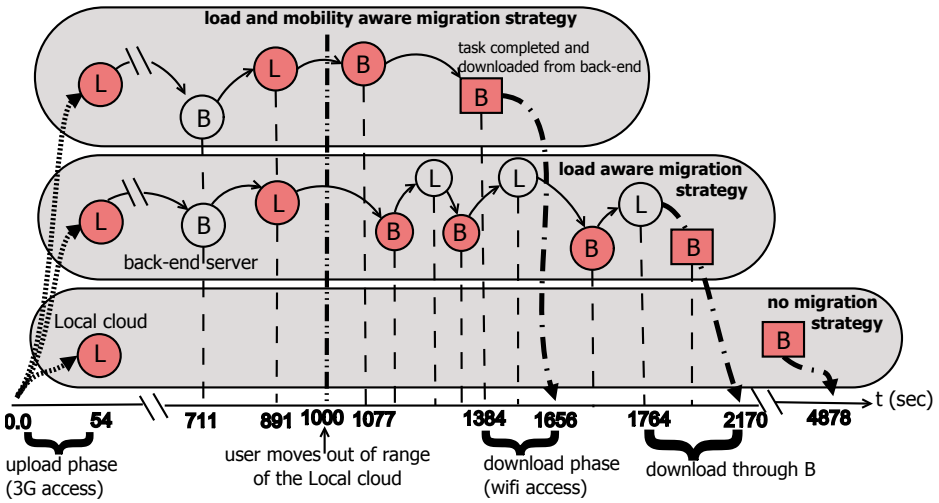
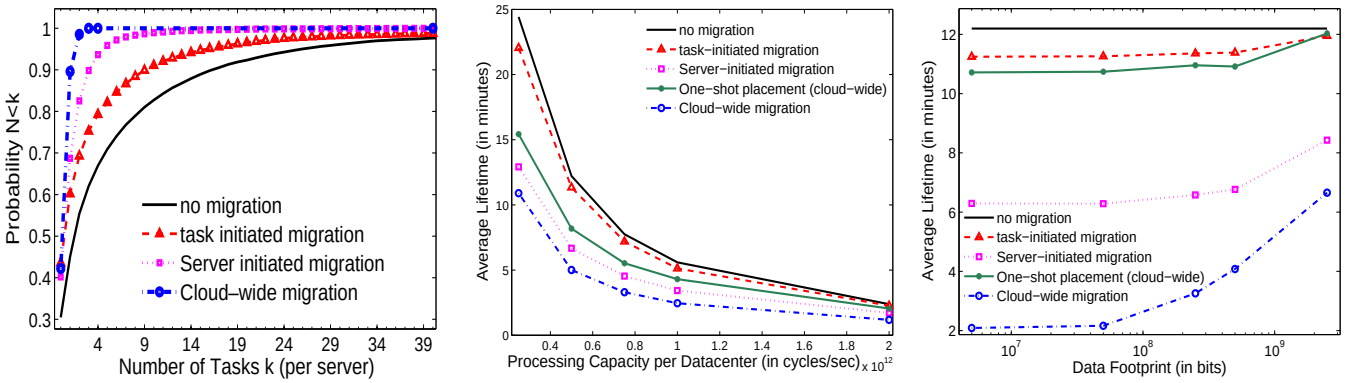


Fig. 4. Indicative migration path of a mobile task throughout its lifetime under a) a no migration strategy b) a load-aware migration strategy and c) a joint mobility and load-aware migration strategy



(a) cdf of number of tasks per server (b) Average lifetime Vs Processing capacity (c) Average lifetime Vs Data footprint

Fig. 5. Performance evaluation of different migration policies