

## Resource Management in Mobile Cloud Computing

Andrei IONESCU

Bucharest University of Economic Studies

andrei.ionescu@antiferno.ro

*Mobile cloud computing is a major research topic in Information Technology & Communications. It integrates cloud computing, mobile computing and wireless networks. While mainly built on cloud computing, it has to operate using more heterogeneous resources with implications on how these resources are managed and used. Managing the resources of a mobile cloud is not a trivial task, involving vastly different architectures. The process is outside the scope of human users. Using the resources by the applications at both platform and software tiers come with its own challenges. This paper presents different approaches in use for managing cloud resources at infrastructure and platform levels.*

**Keywords:** *Cloud Computing, Mobile Cloud Computing, Resource Management*

### 1 Introduction

International Telecommunication Union (ITU) predicts a mobile-broadband penetration rate at worldwide-level of 32% by the end of 2014. It is double the rate of the previous year and almost four times as much as five years ago. There is a huge gap between developed countries and developing countries, with a penetration rate of 84% for the former and 21% for the latter. Mobile-broadband is the highest growing market segment with an estimated 2.3 billion subscriptions by the end of 2014. At the same time, mobile-cellular subscriptions approach market saturation, at a level 7 billion subscriptions [1].

Of much more importance for mobile cloud computing, as both consumers and providers of cloud resources, are the 1 billion smartphones in use (2nd quarter, 2012), with a globally penetration rate of 16.7% [2]. It is estimated that in just three years (by 2015/2016) their number will reach 2 billion. The estimation for 2018 is for a 3.3 billion level. Another potentially mobile computing clients are the tablets, with sales 270 million units just in 2014.

In a short period of time, the Internet and mobile devices have become an integral part of the daily life, in developed world as well as in the developing countries [3]. A Nielsen study shows that in the United States of America, monthly usage of the Internet by adult people splits in 34 hours from mobile

devices opposed with 27 hours from desktop computers. Moreover, from the associated network traffic, only 14% is owed to web usage, the rest of 86% being generated by mobile applications [4].

One of the hottest research topics is Internet of Things (IoT). Smartphones, tablets and mobile terminals make a large part of the interconnected devices, but mobility comes with restrictions on physical dimensions, computing power, battery autonomy and power use. As the world goes mobile, the existence of these hard limits gets in direct conflict with the users' expectations and needs of a constant and predictable performance increase for all computing devices. On the mobile end, users requires more processing power and more autonomy. On the other hand, the infrastructure required to support the mobile services keeps expanding and its energy use develops with it. The small increases in energy efficiency did little to alleviate the constant grow in global power usage.

The shift to mobile was shown to be accompanied by problems such as insufficient resources, limited battery capacity and limits in network connectivity [5]. Decreasing applications' response time, efficient batter use of battery power and available network bandwidth are the main challenges for the future of mobile applications.

Mobile cloud computing is a solution for overcoming the barriers encountered by the rapid expanse of mobile use. Mobile cloud computing resides at the intersection of mobile computing and cloud computing. It is a major research topic for Information Technology and Communications (IT&C), shifting the point of use of computing resource from the end consumer to the resource reach data centers.

**2 Cloud Computing**

Cloud computing is defined by National Institute of Standards and Technologies (NIST) as a model which allows remote on-demand access to shared pool of computing resources (networks, servers, storage, application and services). The allocation and deallocation of these highly available resources are done with minimum management effort and without user interaction [6].

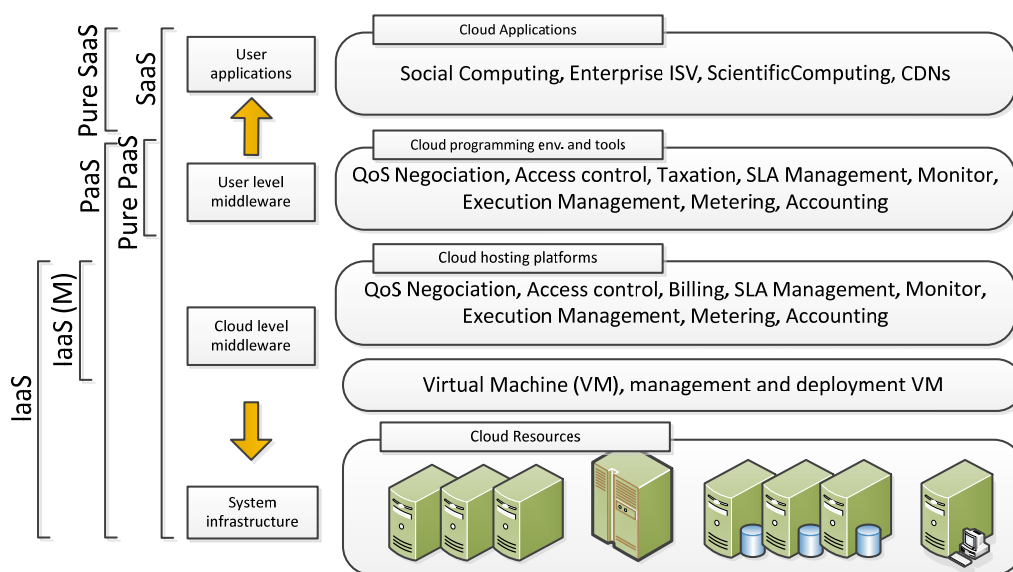
The main characteristics of the cloud computing model are, according to [7]:

- on-demand service. A service client with all the necessary permissions can have on-demand access to computing resources such as processor time or network storage space. The resources are allocated automatically, without

intervention from the supplier side, in the requested quantities.

- remote access. Cloud computing services are served over the Internet. The access is made through standardized mechanisms by heterogeneous clients (mobile phones, laptops, etc.)
- shared resources. Computing resources provided by the cloud computing suppliers form a pool shared by multiple clients using a multi-tenant model. The clients usually have no information on the geographical localization of the physical machines and equipment doing actual computations or providing storage space and other network services.
- scalable and elastic. Resources can be allocated to clients in a swiftly and elastic way, usually by an automated process.
- metered by use. Cloud computing providers automatically monitor and record resources provisioned and used by the clients. This allows a pay-per-use taxation model process, the model at the core of cloud computing.

There are three basic service models, as defined by NIST: Software-as-a-service, Platform-as-a-service and Infrastructure-as-a-service (Figure 1).



**Fig. 1.** Cloud architecture [8]

Software-as-a-service (SaaS). SaaS refers to complete applications provided over the network. Recent developments in web technologies as well as AJAX (Asynchronous JavaScript + XML) and permanent Internet access have allowed porting to web browsers of both functions and user interface of classical desktop applications. These applications and the infrastructure on which they are running are not directly administered by their users, though they might be able to change their behavior through changing internal settings.

Platform-as-a-service (PaaS). It is an intermediate tier providing environments, tools and programming language allowing users to build their own applications. As it is the case for SaaS, the client are not controlling the infrastructure in any way. They do have control over how the applications they're developing are to be implemented and, in some cases, they can specify the configuration of the host virtual machines.

Infrastructure-as-a-service (IaaS). In this model, cloud computing service clients are allowed to provision processing power, storage space, network resources. They then can use them to execute any kind of software, including operating systems and applications. The clients don't administer nor control the cloud's infrastructure. They do control the virtual machines running on the physical infrastructure and the software stack installed on them (operating system and applications). The clients can manage their own storage space and they might even control parts of the physical network infrastructure such as firewalls.

There are four types of cloud computing implementations:

- Public cloud. Infrastructure is owned by the cloud computing services provider and is publicly accessible.
- Community cloud. Physical infrastructure is shared between multiple organizations supporting a community with common purpose. It can be administered by the community itself or by an external entity.
- Private cloud. The infrastructure is used

exclusively by a single organization. It can be administered by the organization or by an external entity. From all the implementations, it usually has the smallest amount of shared resources available.

- Hybrid cloud. The infrastructure is made by integrating two or more public, community or private clouds. Using common protocols and interfaces, services and applications can be ported between implementations.

While there are no universal undisputed definitions of the four implementation types, it is widely accepted that in the following cases there is no cloud computing [9]:

- leasing physical dedicated servers in a data-center for a single activity, such as web hosting, even if the lease is subscription based.
- virtualization by itself, except for the cases when virtual servers can be dynamically created and destroyed as the result of client demand. These virtual servers shouldn't be under the control of services' provider.
- connecting to a remote computer through a Virtual Private Network (VPN).

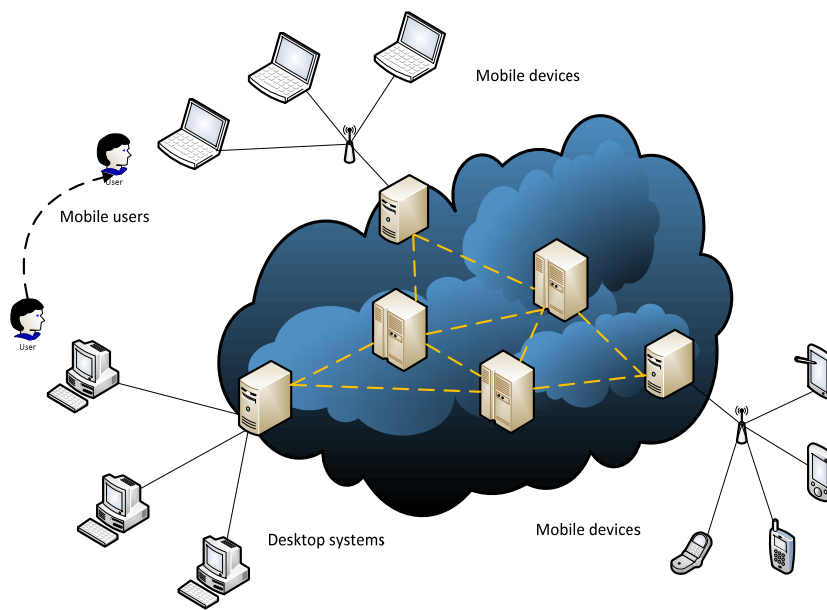
### 3 Mobile Cloud Computing

Mobile cloud computing is described in [10, p. 2] as providing cloud computing services in a mobile environment. In the same article we find a concise definition of mobile cloud computing: „Mobile cloud computing is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices.” [10, p. 1]

In literature, mobile cloud computing is studied through different perspectives, based on the weight granted to the mobile component. At every integration level, the limited resources of the mobile terminals (limited processing power, insufficient storage, lower and more expensive

bandwidth) bring with them different challenges and opportunities. Mobile cloud computing can be understood as regular cloud computing with the only difference the fact that the final users of the provided services are mobile devices (Figure 2). Under this perspective usually falls

Software-as-a-service: mobile consumers for cloud resources. The ongoing process of migrating client from desktop to mobile terminals, such as Bring your own device enterprise policy (BOD), greatly support this perspective.



**Fig. 2.** Mobile cloud computing

From another perspective, mobile terminals can be viewed as both consumers and providers of computational resources. In this case the cloud's infrastructure is dynamic, built ad-hoc from the participating devices, as opposed from the static infrastructure of classical clouds. Every node might belong to a different user being also mobile in most of the cases. There are advantages in using this model. Smartphone and tables have access to a large array of sensors such as cameras, microphones, GPS receivers, barometers, etc., that are not found in classical architecture. Transferring big volumes of data through the network might also be avoided by only having to communicate to nearby mobile neighbors.

The last perspective involves an architectural change, as it introduces a new intermediate tier between the server (cloud) and the clients (mobile devices) in form of cloudlets [11]. The cloudlets are powerful servers or clusters having fast network connections with the

actual cloud but deployed as close as possible to the final mobile users. The model involves rapid virtual machines instantiation running services accessed through wireless LANs. This solution addresses the lack of processing power and storage space, moved from terminals to cloudlets. It also tries to solve the big network latencies if WANs by only having to communicate locally. Because no big improvements in lowering WAN latencies are expected in the near future, great efforts are made for developing the cloudlet model.

There are two ways of bringing computing resources provided by the clouds closer to end-users, based architectural characteristics of the targeted clouds as well as the implementation types [12].

- for private clouds strictly localized within a building or a campus, resource access is made through local WiFi access points. There is no need to modify existing network infrastructure. A cloudlet will

- service one building.
- for clouds geographically distributed clouds, cloudlets are integrated with WiFi access points and deployed in the same way as public WiFi access points. In this case the cloudlets can be thought of as a “data-center in a box”.

#### 4 Resource Management in Cloud Computing

Resource allocation is the process of distributing available resources between the various applications running in a cloud environment. There are several problems addressed by an optimal resource allocation [13]:

- resource contention: multiple consumers are trying to gain access to the same resources at the same time.
- scarcity of resources: there are limited resources. Different cloud implementations must cope with different levels of resource scarcity. The problem is of bigger importance for private clouds.
- resource fragmentation: the resources are isolated. Even when there are enough resources, their potential consumers cannot gain access to them.
- over-provisioning: resources are reserved for a client’s exclusive use in quantities exceeding its needs.
- under-provisioning: opposed to over-provisioning, not enough resources are reserved for exclusive use of a client.

When discussing resources, a difference should be made between data-center resources such as available servers, storage space and network bandwidth and computing resources directly available to mobile devices.

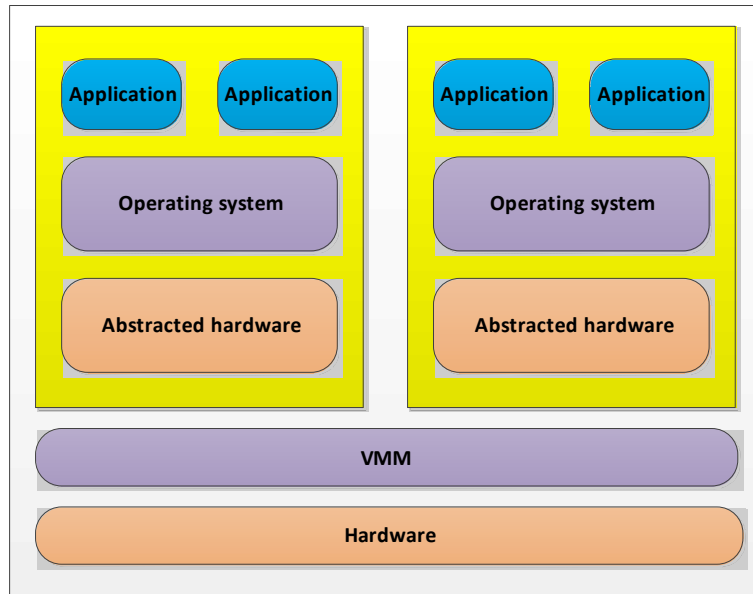
At data-center level there are multiple tiers for resource allocation and optimization: at cluster or supercomputer, virtual machine and operation system disk image levels. Moreover, for any of those levels different objectives can be pursued: increasing power

usage efficiency, increasing or insuring a predefined level of availability for provided services, increasing performance, lowering the data-center air conditioning costs or a combination of them.

#### *Energy efficiency*

Techniques of lowering the power usage developed for mobile clients (laptops, smartphones, etc.) are adapted to be used in the data-center for clusters and supercomputers. One of those techniques is Dynamic Voltage and Frequency Scaling (DVFS) [14]. Through DVFS, frequency and voltage are lowered when processing resources are not demanded. The main used technologies are Intel’s *SpeedStep* and AMD’s *PowerNow!*. Another techniques involves shutting down processor cores, processors and even physical machines when processing demand is low. The objective is to either lower the energy used by the data-center or lower the heat generated by operating the physical machines (and the energy spent keeping the temperature constant).

The next target for energy efficiency optimizations is the virtual machine (Figure 3). The same objectives as for the data-centers are used. Special algorithms are used to launch virtual machines and to allocate them on physical machines [15]. The virtual machines can be reconfigured at runtime, based on current load levels [16]. The reconfiguration is not a trivial task. There can be delays measured in minutes between the moment the new configuration is activated and the moment the virtual machine’s performance is stabilized [17]. Even identifying unused resources at virtual machines’ level is a costly operation and it spend CPU cycles by itself. While it is easily enough to determine the level of a virtual processor load, identifying unused memory pages and calculating total used memory for virtual machine is more complex.



**Fig. 3.** Abstraction through virtual machines

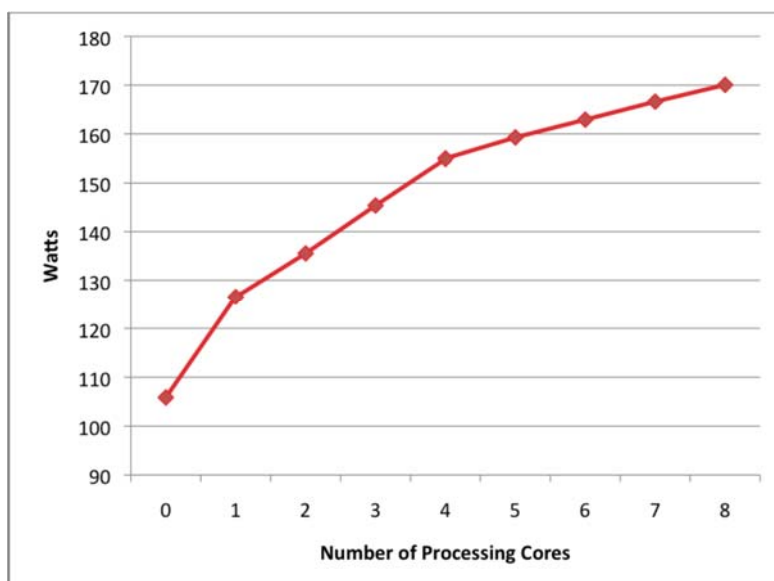
Execution of those virtual machines not running any applications can be suspended or their configuration can be dynamically altered (by changing the number of virtual CPU cores, available memory, and allocated network bandwidth). By borrowing techniques mainly used for insuring a predefined level of services or for isolating hardware defects, virtual machines can be migrated between physical servers or data-centers. Physical machines are loaded up to an optimum level from an energy efficiency point of view while still assuring a certain performance level. Those servers left unoccupied after this consolidation process

can be shut down completely or put in a suspended low-powered mode.

Increasing the number of jobs at the hardware level comes with gains in energy efficiency by itself. It is more efficient to have a higher load level on a single physical machine than to use multiple servers for the same workload. Power usage doesn't increase as fast as the number of active cores.

**Fig. 4** presents the link between power usage and number of active cores for an Intel Core i7 920 processor [15].

It shows the non-linear relation between the number of powered processor cores and energy consumption.



**Fig. 4.** Power use of an Intel Core i7 920

Cores of the same processor shares various components such as level 2 and 3 caches. This components need to be powered independently from the cores leading to the existences of a power consumption baseline. While saving on power usage, this techniques leaves the cloud more vulnerable to Denial of Service (DoS) cybernetic attacks. Servers loaded to a high level will hardly cope with a briskly increase in demands. Starting new physical machines and migrating running virtual machines is a costly process in both time and computing resources.

Virtual machines are launched based on saved system images. Even these images can be optimized for optimum power usage before having running instance.

#### *Job scheduling*

Job scheduling is used for sharing computing resources between client demands. Algorithms developed for distributed system are used. There are two main categories of job scheduling algorithms: Batch Mode Heuristic Algorithms (BMHA) and online mode heuristic scheduling algorithms [18]. Choosing one algorithm over another is made by taken into consideration the pursued objectives. Some of them try to minimize applications' execution time while others try to achieve a highest throughput. When using BHMA, the jobs are put on a waiting queue when they arrive in the system. Their scheduling starts after a fixed period of time:

- First-In-First-Out (FIFO) or First-Come-First-Serve (FCFS). It is a simple and fast algorithm. Jobs are queued as they arise. Being non-preemptive is its biggest disadvantage. A small task queued after a bigger one will have to wait for the first to finish its execution before having a chance to be executed itself.
- Round Robin: jobs are scheduled in a similar way to FIFO/FCFS but they are given an equal limited predefined CPU time. A job is interrupted and put at the end of the queue if its execution doesn't finished in the allotted time. The jobs are dispatched to the processing units trying to achieve equal load levels between

them. As it was previously show, this leads to power usage inefficiencies.

- min-min: the jobs with the smallest execution time are scheduled first. Those with the greatest execution time encounter the biggest delays.
- max-min: the jobs with the bigger execution time are scheduled first. Those with the smaller execution time suffer the greatest delays.

When using online mode heuristic algorithms, the scheduling process takes place as soon as a new job arrives into the system. These algorithms are better suited to the heterogeneous environment of the cloud computing. Examples of online mode heuristic algorithms:

- most fit task scheduling algorithm (MFTF): the job considered best fitted to the execution resources is scheduled first
- priority scheduling algorithm: jobs are scheduled based on priority values attached to them. Equal priority jobs are scheduled using FIFO/FCFS.
- earliest deadline first: jobs are ordered in a priority queued and whenever a scheduling event is triggered, the job with the earliest deadline is scheduled first. The algorithm is mainly used in real-time operating systems.
- greedy: scheduling is done in phases. In each phase, the best decision is made without taking into consideration consequences in the future.
- (Rank) Match-making: try to pair jobs with those virtual machines best fitted to execute them based on required resource. The matching is done using a rank value calculated based on monitoring data, filtering out unsuited virtual machines.
- genetic algorithms: algorithms based on strategies copied from natural selection. They can be applied on a big solutions space and they can accommodate complex objective functions.

Scheduling algorithms for the main open source and proprietary cloud computing platforms (**Table 1**. Scheduling algorithms) are presented in [19].

**Table 1.** Scheduling algorithms

Cloud computing platform	Open Source	Scheduling algorithms
Eucalyptus	Yes	- <i>Greedy first fit</i> - <i>Round robin</i>
Open Nebula	Yes	- <i>Rank match-making scheduling</i> - <i>Preemption scheduling</i>
Rackspace	Yes	- <i>Round robin</i> - <i>Weighted round robin</i> - <i>Least connections</i> - <i>Weighted least connections</i>
Nimbus	Yes	- Virtual machines schedulers PBS and SGE
Amazon EC2	No	- <i>Xen</i> - <i>Swarm</i> - <i>Genetic</i>
Red Hat	Yes	- <i>Breadth first</i> - <i>Depth first</i>
lunacloud	Yes	- <i>Round robin</i>

### 5 Resource Management in Mobile Clouds

Mobile cloud computing can be described as sharing computing resources within a mobile ecosystem. The mobile resources are insufficient and their availability is unpredictable. There are big efforts made for integrating mobile devices into cloud computing allowing the former access to bigger resources of the latter. While cloud storage is readily available, using processing power in the data center, or from an intermediate tier, for mobile needs is at its infancy.

#### Code offloading

One of the techniques used for code offloading is complete execution transfer. In the cloud are instantiated one or more virtual machines with identical software stacks as the targeted mobile device. Part or all of main processing is done in the cloud and the results are shown on the device. The users are given the illusion of using a more powerful terminal than in reality. The technique resembles cloudlets use, but these are not running replicas in VMs. The developing process of mobile applications is greatly eased. There are no explicit resource allocation and release in the cloud.

Mobile applications can be also partitioned into loosely coupled modules. Part or all of these modules can be executed in the cloud.

Partition might be done both during the development process, by the application developer, or dynamic, at runtime. Dynamically partitioning requires implementation of a dedicated specialized middleware to decide where the execution will take place and to dispatch the execution and reintegrate the results (**Fig. 5**). On the mobile device, minimum to be executed is the user interface. There are many platforms trying to implement this model: AlfredO [20], based on R-OSGi, an OSGi extension allowing services migration between Java virtual machines. MAUI [21], another code offloading platform has as its main objective maximizing battery autonomy through code offloading in the cloud. Methods allowed to be remotely executed are annotated and it is decided at runtime whether the execution should take place on the device or in the cloud [10].

Another technique developed for code offloading is using weblets. Applications are partitioned in modules called weblets, platform and programming language independent. They are locally executed or in the cloud as IaaS. The partitioning splits the user interface from the weblets. A weblet manager transparently decides at runtime weblet instantiation and/or migration.



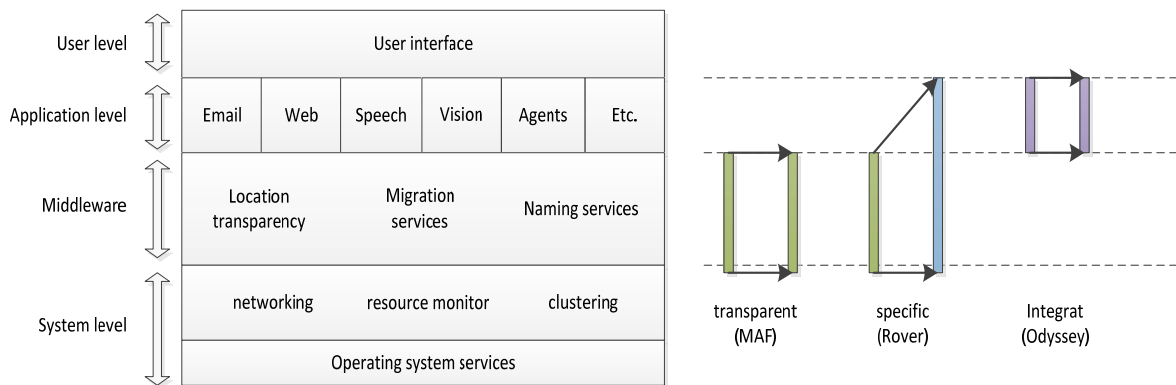


Fig. 5. Partitioning models

Migration of virtual machines at runtime can also be used as code offloading. Application execution status is migrated in the cloud and back, as needed. It is a more resource intensive process than selective tasks or modules migration.

When mobile devices form the node of a cloud infrastructure by also providing processing, storage, and network bandwidth resources, they form an ad-hoc cloud. The model might be an alternative for more traditional approaches when there is no connectivity to Internet or to the cloud provider, due to network problems, DoS attacks or lack of available resources. Higher communications cost can be avoided using this technique by only using WiFi LAN traffic. Furthermore, it can be used for building communities where users execute collaboratively shared tasks.

Using code offloading has other benefits also, such as increasing battery autonomy of mobile devices. CPU is big energy consumer; releasing it from part of its tasks allows lowering operating frequency, shutting down some of its cores or switching to low-power ones. Lowering 3G/4G network use by moving traffic to wireless LAN (for accessing cloudlets or an ad-hoc cloud) will also lower the overall power consumption.

Code offloading already entered commercial usage. Silk web browser from Amazon uses it to split a web-page into subsystems (communication, HTML, rasterisation) and execute them on Amazon's own EC2 cloud. Other uses for code offloading: Apple's Siri and Google's Voice Search.

*Pre-allocation and de-allocation*

Cloud computing providers might be bound through Service Level Agreements (SLA) to certain levels and availability for their services. In order to achieve it, resources can be redundantly allocated, identical virtual machines (replicas) can execute the same tasks. High availability might conflict with performance: more replicas and bigger redundancy have a cost besides the directly involved resources. Processing power, network usage, and storage space are needed for assuring consistency between replicated data or tasks [22].

One of the following strategies is used when pre-allocating resources:

- On Demand Activation (ODS): resources, virtual machines are created and/or activated whenever there is a new demand. It's the simplest strategy to implement. It has big delays because there is a time needed for creation/activation process.
- Spare Resource Strategy (SRS): a certain amount of resources is kept ready in advance for processing new requests. Virtual machines are pre-instantiated trying to minimize the delay from ODS. Deciding the amount of pre-allocated resources is done through the same process as resource allocation, using the same algorithms. There are disadvantages to this strategy: at high loads for the cloud it behaves just like ODS. Pre-allocated resources are unused resources in what involves the pay-per-use model

while still consuming power and limiting the amount of free resources.

Allocation strategies take into consideration the implementation model of the mobile cloud platform. Private clouds are usually based on lesser physical resources than the public ones. They cannot provide the user with the illusion of infinite processing power. [23] identifies three elements influencing allocation strategies for private clouds:

- budget and physical equipment limits.
- ever increasing number of users.
- difficulties in de-allocating resources blocked by inactive users.

While de-allocating difficulties are not specific to public cloud implementations, their providers should not ignore them. Cloud metering and taxation should reflect as precise as possible resources used by the clients. The basic pay-per-use model simply ignores those resources blocked by inactive users. Situations arise when resources are only intermittently used, skipping taxation for any intervals but those of peak usage. Thus, models have been proposed for taxing clients based on the time when the assigned resources were made unavailable for other uses [23].

## 6 Conclusions

Cloud computing as well as mobile computing are being a bigger and bigger part of today's information technology landscape. Efficient management of resources available in data-center has its role for both customer satisfaction and profit maximization. Energy usage has an ever increasing importance due to the increases in processing per unit of power not catching up with the increases in deployed processing power. All of these while, on the foreseeable future, energy cost will keep raising.

Regarding the mobile computing aspect, there is no industry wide mature technology for managing resources. By their nature, mobile applications resides at the opposed ends of the network spectrum. There are native offline applications which require none or seldom access to cloud and there are online applications which need to always

stay connected and exchange information. Any solution must cater to both extremes. Mobile cloud is a heterogeneous environment where the place of code execution is decided dynamically. Magic solutions are not to be expected in the near future. The increased application developing effort needed to both take advantage and cope with the mobile computing particularities is requiring greater abstraction at programming level [24].

## Future works

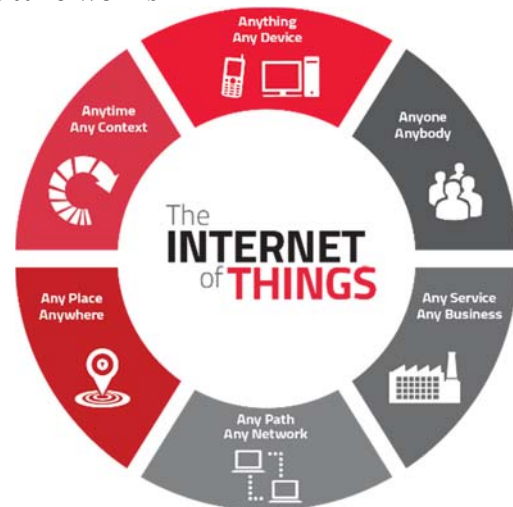


Fig. 6. Internet of Things [25]

Developing a methodology allowing comparison of different cloud computing platforms is of great importance. This is especially true for private clouds, being more resources constrained than the other implementation types. In the age of Internet of Things (**Fig. 6**), porting resource management solutions initially developed for mobile clouds should be easy. Embedded devices share a great part of restrictions with their mobile counterparts: small battery autonomy, limited processing power, and low network bandwidth as well as being data sources in form of sensors unavailable to classic client/server or cloud implementations. As the whole Internet of Things concept gains traction, the need to integrate the new devices with cloud base applications and services already in place will lead to an unavoidable integration of mobile concepts with non-mobile aware existing deployments.

## References

- [1] ITU, "The World in 2014: ICT Facts and Figures," 2014. [Online]. Available: <http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>. [Accessed May 2014].
- [2] A. Favell, "Global mobile statistics 2014 Part A: Mobile subscribers; handset market share; mobile operators," dotMobi, May 2014. [Online]. Available: <http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats/a>. [Accessed May 2014].
- [3] P. Research, "Emerging Nations Embrace Internet, Mobile Technology," 2014. [Online]. Available: <http://www.pewglobal.org/2014/02/13/emerging-nations-embrace-internet-mobile-technology/>. [Accessed 13 02 2014].
- [4] Nielsen, "What's Empowering The New Digital Consumer?," 2014. [Online]. Available: <http://www.nielsen.com/us/en/newswire/2014/whats-empowering-the-new-digital-consumer.html>. [Accessed 02 02 2014].
- [5] M. Satyanarayanan, "Fundamental challenges in mobile computing," in Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, New York, NY, USA, 1996.
- [6] W. Jansen and T. Grance, "NIST Guidelines on Security and Privacy in Public Cloud Computing," January 2011. [Online]. Available: <https://cloudsecurityalliance.org/download/nist-guidelines-on-security-and-privacy-in-public-cloud-computing/>. [Accessed May 2014].
- [7] D. E. Y. Sarna, Implementing and Developing Cloud Computing Applications, Auerbach Publications, 2010.
- [8] R. Buyya, C. Vecchiola and S. T. Selvi, Mastering Cloud Computing Foundations and Applications Programming, Waltham: Elsevier Inc., 2013.
- [9] M. I. Williams, A Quick Start Guide to Cloud Computing: Moving Your Business into the Cloud, London: Kogan Page Ltd., 2010.
- [10] D. Kovachev, Y. Cao and R. Klamm, "Mobile Cloud Computing: A Comparison of Application Models," arXiv.org, pp. 2-8, 2011.
- [11] M. Satyanarayanan, P. Bahl, R. Cáceres and N. Davies, "The Case for VM-Based Cloudlets in Mobile Computing," Pervasive Computing, IEEE, vol. 8, no. 4, pp. 14 - 23, 2009.
- [12] M. Satyanarayanan, "Mobile computing: the next decade," in MCS '10 Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, New York, USA, 2010.
- [13] V. Vinothina, R. Sridaran and P. Ganapathi, "A Survey on Resource Allocation Strategies in Cloud Computing," International Journal of Advanced Computer Science and Applications, vol. 3, no. 6, pp. 97-104, 2012.
- [14] L.-T. Lee, K.-Y. Liu, H.-Y. Huang and C.-Y. Tseng, "A Dynamic Resource Management with Energy Saving Mechanism for Supporting Cloud Computing," International Journal of Grid and Distributed Computing, vol. 6, no. 1, pp. 67-76, 2013.
- [15] A. Beloglazov and R. Buyya, "Energy Efficient Resource Management in Virtualized Cloud Data Centers," in 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, Melbourne, Australia, 2010.
- [16] G. A. Geronimo, J. Werner, C. Becker Westphall, C. Merkle Westphall and L. Defenti, "Provisioning and Resource Allocation for Green Clouds," in ICN 2013 : The Twelfth International Conference on Networks, 2013.
- [17] J. Rao, B. Xianping, K. Wang and C.-Z. Xu, "Self-Adaptive Provisioning of Virtualized," in Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, 2011.
- [18] P. Salot, "A Survey of Various

- Scheduling Algorithms in Cloud Computing Environment,” International Journal of Research in Engineering and Technology, vol. 2, no. 2, pp. 131-135, 2013.
- [19] C. T. Lin, "Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment," International Journal of Computer Science and Engineering, vol. 1, no. 1, pp. 17-23, 2013.
- [20] I. Giurgiu, O. Riva, D. Juric, I. Krivulev and G. Alonso, "Calling the cloud: Enabling mobile phones as interfaces to cloud applications.," in Middleware '09: Proceedings of the 10th ACM/IFIP/USENIX International Con-, New York, NY, USA, 2009.
- [21] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra and P. Bahl, "MAUI: Making Smartphones Last Longer with Code Offload," in MobiSys '10 Proceedings of the 8th international conference on Mobile systems, applications, and services, New York, USA, 2010.
- [22] R. Jhawar and V. Piuri, "Adaptive Resource Management for Balancing Availability and Performance in Cloud Computing," SECRYPT, pp. 254-264, 2013.
- [23] R. K. L. Ko, A. Y. S. Tan and G. P. Y. Ng, "'Time' for Cloud? Design and Implementation of a Time-Based Cloud Resource Management System," in Proceedings of the 7th IEEE International Conference on Cloud Computing (CLOUD'14), 2014.
- [24] T. Shelton, Business Models for the Social Mobile Cloud: Transform Your Business Using Social Media, Mobile Internet, and Cloud Computing, New Jersey: Wiley, 2013.
- [25] "The Way of the Future - Securing the Internet of Things (IoT)," [Online]. Available: <http://www.informationsecuritybuzz.com/way-future-securing-internet-things-iot>. [Accessed December 2014].



**Andrei IONESCU** obtained his Master of Science diploma in Economic Informatics in 2014. He has more than 10 years of experience as a software developer, currently working as an independent contractor. Since 2014 he is a PhD candidate at the Bucharest Academy of Economic Studies with a thesis in the field of Resource Management in Cloud Computing.