Resource Allocation for Distributed Cloud: Concepts and Research Challenges

Patricia Takako Endo, André Vitor de Almeida Palhares, Nadilma Nunes Pereira, Glauco Estácio Gonçalves, Djamel Sadok, and Judith Kelner, Federal University of Pernambuco
Bob Melander and Jan-Erik Mångs, Ericsson Research

Abstract

In a cloud computing environment, dynamic resource allocation and reallocation are keys for accommodating unpredictable demands and, ultimately, contribute to investment return. This article discusses this process in the context of distributed clouds, which are seen as systems where application developers can selectively lease geographically distributed resources. This article highlights and categorizes the main challenges inherent to the resource allocation process particular to distributed clouds, offering a stepwise view of this process that covers the initial modeling phase through to the optimization phase.

Current cloud computing providers mainly rely on large and consolidated datacenters in order to offer their services. This predominantly centralized infrastructure brings many well-known challenges, such as a need for resource overprovisioning and costly heat dissipation and temperature control, and it also naturally increases the average distance to end users [1].

In contrast, the authors in [2] introduce what they refer to as "embarrassingly distributed applications." These are, according to them, cloud services that do not require massive internal communication among large server pools, and are created out of *small distributed datacenters*. Under this model, one may understandably take advantage of *geo-diversity* to potentially improve cost and performance. However, the authors propose using public infrastructure for communication between datacenters and also with end users. The drawback we see with this approach is that it transfers traffic control to Internet service providers, who may lack the bilateral agreements that would adequately support cloud traffic constraints.

Authors in [3] make use of *distributed voluntary resources* to form what they call "nebulas" with the goal of building clouds that are more dispersed and have low costs of deployment. Some specific classes of applications fit within this idea, such as experimental cloud services, dispersed data-intensive services, and shared services. However, the lack of central management is a major issue with regard to reliability and state maintenance in the presence of failures.

To overcome these limitations, we choose a generic and distributed solution that may be used in the context of many types of services (describing their requirements at different abstraction levels). We refer to this concept as a *distributed cloud*. In such a scenario, cloud providers hire infrastructure on demand, and acquire dedicated connectivity and resources from communication providers. It is important to highlight that the infrastructure may range from routers and links to servers and databases.

Distributed clouds have similar characteristics to current cloud providers. In addition to their essential offerings, such as scalable services, on-demand usage, and pay-as-you-go business plans, distributed clouds also take advantage of geo-diversity. However, unlike in [2], a higher level of governance may be exercised.

An interesting application area that stands to benefit from offering resource allocation in geo-distributed scenarios is that of network virtualization (NV) [4]. Authors in [5] define NV as a system that supports "multiple coexisting heterogeneous network architectures from different service providers, sharing a common physical substrate." In a network virtualization environment (NVE), virtual networks (VNs), composed of virtual routers and virtual links, are deployed on a shared physical network, called substrate network (SN). The selection and span of VNs may be achieved under distributed geolocation constraints to improve user satisfaction and/or provider investment return. Thus, the main NV problem consists of choosing how to allocate a VN over an SN, meeting requirements and minimizing resource usage of the SN.

Although NV and distributed clouds are subject to similar problems and scenarios, there is an essential difference between them. While NV commonly models its resources using graphs only (requests are always virtual network ones), a distributed cloud allows many abstraction levels of resource modeling (requests may be for different types of applications). This way, one may see NV just as a particular instance of the distributed cloud.

There are some NV projects that already work with the idea of a geographically distributed cloud. PlanetLab is a popular project that provides geographically distributed virtualized nodes. VINI offers a network infrastructure in which researchers can test new ideas from the field of NV. SAIL is an FP7 European research project that aims to provide resource virtualization in order to allow researchers to investigate novel networking technologies, offering them what they call cloud networking.

This article gives special emphasis to the challenges for

0890-8044/11/$25.00 © 2011 IEEE

resource allocation in *distributed clouds*, focusing on four fundamental points:

- Resource modeling
- Resource offering and treatment
- Resource discovery and monitoring
- Resource selection

There is, in our view, very little literature available on this different cloud computing paradigm, and we expect to present the reader with useful related insights.

This article is organized as follows. We provide some basic definitions; we state relevant research challenges about resource allocation; we discuss resource allocation challenges and NV; and finally, we draw some conclusions.

Definitions

This section is particularly important as it highlights the main differences between the traditional cloud and a distributed one. It also establishes the nomenclature used in the rest of the article. Figure 1 shows the four entities that typically compose the distributed cloud computing ecosystem: end user, cloud user, cloud provider, and cloud applications. Furthermore, it shows the resource allocation system and some interfaces, described later.

The *cloud user* is located in the middle, between end users and the cloud provider, and is responsible for providing applications. A cloud user can be seen as a service provider, who leases resources/services offered by the provider in order to host applications that will be consumed by end users. In turn, the *end user* is the customer of an application that simply uses applications, generating demand for the cloud. It is important to highlight that in some scenarios (e.g., scientific computation or batch processing) cloud users may behave as end users to the cloud.

The *cloud provider* is the owner of the infrastructure. In this way, a provider is responsible for managing physical and virtual resources to host applications. These *cloud applications* may be of different types (a farm of web servers, a scientific application, etc.) that all have different requirements. For example, in the case of NV, a request for a VN may be represented with constraints associated with nodes (e.g., CPU and physical location) and links (e.g., delay, bandwidth, and jitter). For each VN request, the provider has to assign virtual resources to be hosted on its physical resources [4].

An essential feature of resource allocation mechanisms in cloud computing results from the need to guarantee that the requirements of cloud applications are met. According to [6], resource allocation must be "robust against perturbations in specified system parameters." In other words, it must limit the degradation in performance to a certain acceptable range.

To this end, allocation mechanisms should know the status of each element/resource in the distributed cloud environment and, based on them, intelligently apply algorithms to better allocate physical or virtual resources to applications according to their pre-established requirements. This way, we may consider that cloud resources, resource modeling, application requirements, and provider requirements constitute the input used by a resource allocation mechanism (Fig. 2).

These resources are located in a distributed pool and shared by multiple users. Each provider is free to model its resources according to its business model.

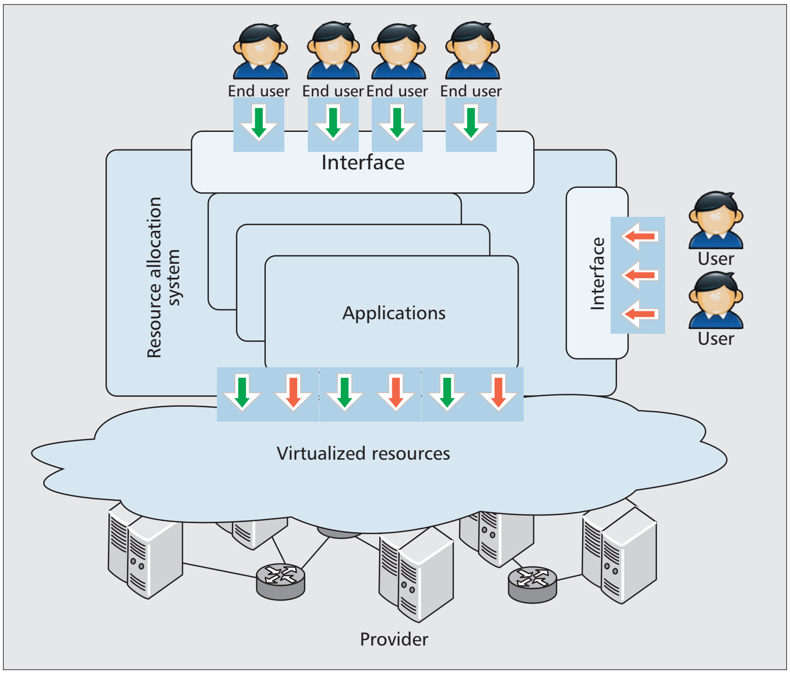

Figure 1. *Entities in the cloud computing ecosystem.*

Research Challenges Inherent to Resource Allocation

One of the most important aspects of cloud computing is the availability of "infinite" computing resources that may be used on demand. Users may rely on this "infinite" resource feeling because the distributed cloud — through the resource allocation system (RAS), which is shown in Fig. 2 — tries to deal with end users' demands in an elastic way. This elasticity allows the statistical multiplexing of physical resources, avoiding both under- and overprovisioning, as is the case in most corporate information technology (IT) infrastructures.

Furthermore, there is a need to cope with resource heterogeneity. This can be seen in distributed clouds, which are composed of computational entities with different architectures, software, and hardware capabilities. Thus, the development of a suitable resource model is the first challenge that an RAS must deal with.

The RAS for a distributed cloud also faces the challenge of representing cloud applications and describing them in terms of what is known as *resource offering and treatment*. Together with traditional network requirements (bandwidth and delay) and computational requirements, (CPU and memory), new requirements (locality restrictions and environmental necessities) are now part of the distributed cloud's additional requirements. Similarly, the right mechanisms for *resource discovery and monitoring* should also be designed, allowing the RAS to be aware of the current status of available resources. Based on this information, the RAS is then able to optimize already allocated resources, and can also elect available resources to fulfill future demands.

In Fig. 3, we see how the four challenges above are related. First, the provider faces the problems grouped together in the *conception phase*, where the provider should model resources according to the kind of service(s) it will supply and the type of resources it will offer. The next two challenges are faced in the scope of the *operational phase*. When requests arrive, the RAS should be aware of the current status of resources in order to determine if there are available resources in the distributed cloud that could satisfy the present request. Then, if this is the case, the RAS may select and allocate them to serve the request.

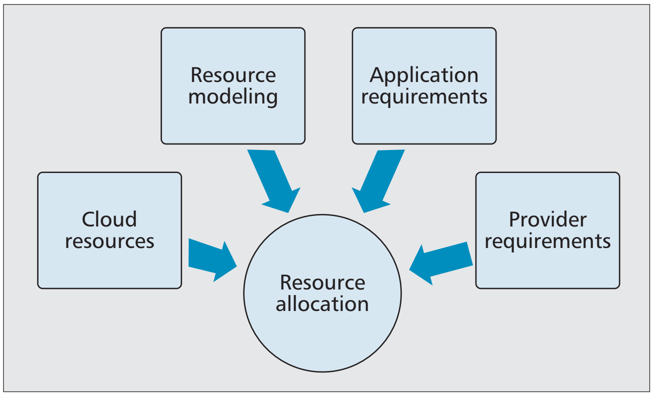

Figure 2. *Resource allocation inputs.*

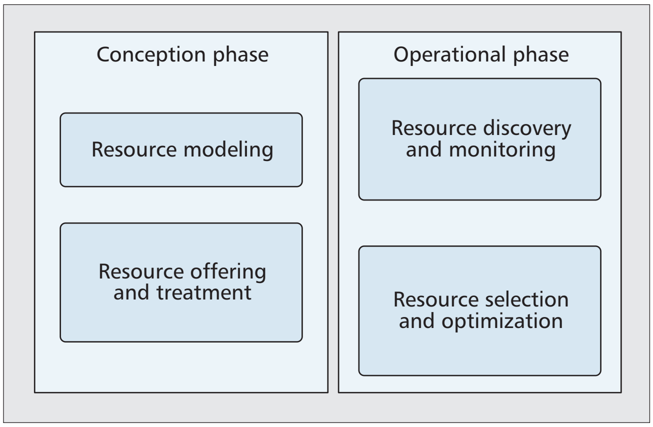

Figure 3. *Relationship between resource allocation challenges.*

When conceiving a distributed cloud, it is natural for its provider to choose the nature of its offering: service, infrastructure, and platform as a service (SaaS, Iaas, and PaaS).

The next sections describe each of these four challenges.

Resource Modeling

The cloud resource description defines how the cloud deals with infrastructural resources. This modeling is essential to all operations in the cloud, including management and control. Optimization algorithms are strongly dependent on the resource modeling scheme used.

Network and computing resources may be described by several existing notations, such as the Resource Description Framework (RDF) and Network Description Language (NDL). However, in a cloud environment, it is very important that resource modeling take into account schemas capable of representing virtual resources, virtual networks, and virtual applications. According to [7], virtual resources need to be described in terms of properties and functionalities, much like services and devices/nodes are described in existing service architectures.

The *granularity of the resource description* is another important point. The amount of detail that should be taken into consideration when describing resources is related to the difficulty of achieving a generic solution for distributed clouds. If resources are described using many details, there is a risk that the resource selection and optimization phase could become hard and complex to handle. On the other hand, more details allow more flexibility and leverage in the usage of resources.

Additionally, resource modeling is associated with a big challenge in current cloud computing: *interoperability*. The author in [8] describes the "hazy scenario," wherein large cloud providers use proprietary systems, hampering integration between different and external clouds. In this way, the main goal of interoperability in clouds is to realize the seamless flow of data across clouds, and between clouds and their local applications [9]. Solutions such as intermediary layers, standardization, and open application programming interfaces (APIs) are interesting options for interoperability.

According to [10], interoperability in the cloud faces two types of heterogeneities: vertical and horizontal. The former is intra-cloud interoperability, and may be addressed by middleware and enforcing standardization. The authors highlight the Open Virtualization Format (OVF) as an interesting option for managing virtual machines (VMs) across heterogeneous infrastructures. The latter heterogeneity type is more difficult to address because it is related to clouds from different providers. Once each provider manipulates and describes their resources at their own abstraction level, the challenge is how to lead with these differences to permit interaction between clouds. A high level of granularity in the modeling may help to address this type of problem, but perhaps at the cost of losing information.

Distributed clouds may take advantage of accruing horizontal interoperability. In such a scenario, a provider may receive a request with specific locational constraints, and for some reason (e.g., the unavailability of resources close to the requested location) cannot fulfill that request. Then, as an alternative, the provider may "borrow" resources from another one by dynamically negotiating these.

Resource Offering and Treatment

Once the cloud resources are modeled, the provider may offer interfaces that are elements of the RAS, as shown in Fig. 1. The middleware should handle resources (at a lower level) and, at the same time, deal with the application's requirements (described at a higher level).

It is important to highlight that resource modeling is possibly independent of the way they are offered to end users. For example, the provider could model each resource individually, like independent items on a fine-grained scale, such as the gigahertz of CPU or gigabytes of memory, but offer them as a coupled collection of items or a bundle, such as VM classes (high memory and high processor types).

Since a distributed cloud craves a generic solution (i.e., to support as many applications as possible), resource offering becomes very cumbersome. Questions like *"how can one achieve a good trade-off between the granularity of the resource modeling, and the ease of dealing with the generality level?"* and *"how many types of applications may one support to be considered generic enough?"* must be considered by providers.

Furthermore, handling resources requires that the RAS implement solutions to control all the resources in the cloud. Such control and management planes would need a complete set of signaling protocols to set up hypervisors, routers, and switches. Currently, to deal with these tasks, each cloud provider implements their own solution, which generally inherits a great deal from datacenter control solutions. They also employ solutions for the integrated control of hypervisors. In the future, new signaling protocols can be developed for resource reservation in heterogeneous distributed clouds.

The RAS must ensure that all requirements may be met with the available resources. These requirements have been defined previously between the provider and each cloud user, and may be represented by service level agreements (SLA) and ensured by the provider through continuous monitoring [11].

You may recall that, in addition to common network and computational requirements, new requirements are present under distributed cloud scenarios. Below, we describe some of these. The list is merely illustrative, since there are many distinct use scenarios, each with possibly differing requirements.

The *topology of the nodes* may be described. In this case, cloud users are able to set inter-node relationships and communication restrictions (e.g., downlinks and uplinks). This is illustrated in the scenario where servers — configured and managed by cloud users — are distributed (at different physical nodes), while it is necessary for them to communicate with each other in a specific way.

Jurisdiction is related to where (physically) applications and their data must be stored and handled. Due to restrictions such as copyright laws, cloud users may want to limit the locations where their information can be stored (e.g., countries or continents). This requirement should be re-evaluated to ensure that it does not conflict with topology requirements.

The *node proximity* may be seen as a constraint, where a maximum (or minimum) physical distance (or delay value) between nodes is imposed. This may also have direct impact on other requirements, such as topology. Although cloud users do not know about the actual topology of the nodes, here they may merely request a delay threshold, for example.

The *application interaction* describes how applications are configured to exchange information with each other. Cloud users may introduce some limitations (e.g., access control) according to their policies. Thus, application interaction and topology requirements may also be strongly related to each other.

The cloud user should also be able to define *scalability* rules. These rules would specify how and when the application would grow and consume more resources from the cloud. Work in [12] defines a way of doing this, allowing the cloud user to specify actions that should be taken (e.g., deploying new VMs) based on thresholds of observed metrics.

Resource Discovery and Monitoring

Resource discovery stems from the provider needing to find appropriate resources (suitable candidates) to comply with requests. In addition, questions like *"how can one discover resources with (physical/geographical) proximity in a distributed cloud?"* and *"how can one minimally impact the network, especially costly interdomain traffic?"* also fall within the responsibility of resource discovery, and cannot be answered trivially. Furthermore, considering distributed clouds, any new signaling overhead should not affect other essential quality-of-service requirements.

A simple implementation of the resource discovery service uses a discovery framework with an advertisement process, and has been described in [7] for the NV scenario. It is used by brokers to discover and match available resources from different providers. It consists of distributed repositories responsible for storing resource descriptions and states.

Considering that one of the key features of cloud computing is its capability of acquiring and releasing resources on demand [13], *resource monitoring* should be continuous, and should help with allocation and reallocation decisions as part of overall resource usage optimization. A careful analysis should be done to find an acceptable trade-off between the amount of control overhead and the frequency of resource information refreshing.

The above monitoring may be passive or active. It is considered *passive* when there are one or more entities collecting information. The entity may continuously send polling messages to nodes asking for information or do this on demand when necessary. On the other hand, the monitoring is *active* when nodes are autonomous and may decide when to send asynchronously state information to some central entity.

Naturally, distributed clouds may use both alternatives simultaneously to improve the monitoring solution. In this case, it is necessary to synchronize updates in repositories to maintain consistency and validity of state information.

Resource Selection and Optimization

With information regarding cloud resource availability at hand, a set of appropriate candidates may be highlighted. Next, the resource selection process finds a configuration that fulfills all requirements and optimizes the usage of the infrastructure. In virtual networks, for example, the essence of resource selection mechanisms is to find the best mapping of the virtual networks on the substrate network with respect to the constraints [14]. Selecting suitable solutions from a set is not a trivial task due to the dynamicity, high algorithm complexity, and all the other different requirements relevant to the provider.

Resource selection may be done using optimization algorithms. Many optimization strategies may be used, from simple and well-known techniques such as simple heuristics with thresholds or linear programming to newer, more complex ones, such as Lyapunov optimization [15]. Moreover, artificial intelligence algorithms, biologically inspired ones (e.g., ant colony behavior), and game theory may also be applied in this scenario. Authors in [16] define a system called Volley to automatically migrate data across geo-distributed datacenters. This solution uses an iterative optimization algorithm based on weighted spherical means [16].

Resource selection strategies fall into a priori and *a posteriori* classes. In the a priori case, the first allocation solution is an optimal one. To achieve this goal, the strategy should consider all variables influencing the allocation. For example, considering VM instances being allocated, the optimization strategy should figure out the problem, presenting a solution (or a set of possibilities) that satisfies all constraints and meets the goals (e.g., minimization of reallocations) in an optimal manner.

In an *a posteriori* case, once an initial allocation that can be a suboptimal solution is made, the provider should manage its resources in a continuous way in order to improve this solution. If necessary, decisions such as to add or reallocate resources should be made in order to optimize the system utilization or comply with cloud users' requirements.

Since resource utilization and provisioning are dynamic and changing all the time, it is important that any *a posteriori* optimization strategy quickly reach an optimal allocation level, as a result of a few configuration trials. Furthermore, it should also be able to optimize the old ones, readjusting them according to new demand. In this case, the optimization strategy may also fit with the definition of a priori and dynamic classification.

Discussion

In this section we discuss the challenges of resource allocation, seeing the distributed cloud allocation problem partially as a NV allocation problem. This is one of many views of the problem. We see that the NV view is important for distributed clouds, essentially because it can easily model the geographic location of the allocated resources, as can be seen in [17].

The authors of [17] describe the problem of NV on a substrate network. The resource modeling and offering approach is generally based on graphs. The SN and virtual network requests can be seen as sets of nodes and edges, forming the substrate graph. Bandwidth and CPU (or memory requirements) can be modeled as capacities associated with each link or node. An assignment can be seen as a simple mapping from the virtual nodes of the request to the substrate nodes and from the virtual links to the substrate paths.

With regard to *resource monitoring*, the solution is totally indifferent as to how the information on resource and network states is provided or obtained. The algorithm just considers that this information exists and uses it to perform optimal allocation.

Because the VN allocation problem is NP-hard [5], many approaches require some heuristic solutions and approximation algorithms. The *resource usage optimization* presented in [17] is applied a priori to optimize the revenue and cost to the provider. Given the model, the algorithm allocates virtual networks in consideration of constraints such as CPU, memory, location, bandwidth, and an objective function. The authors reduce their problem to a mixed integer programming problem and then relax the integer constraints to solve the problem with a polynomial time algorithm. An approximated solution for the initial problem is obtained through this method. Two approximation algorithms have been used. The first uses a deterministic approximation, and the other uses a random approach. Other approaches in [18] first allocate nodes and then the virtual links between them in separated steps using both a priori and *a posteriori* techniques.

Final Considerations

Our contributions are twofold. First, we establish and enforce the definition of what is seen as a distributed cloud. Next, the four main challenges for such a cloud paradigm are described. These are:

- Resource modeling
- Resource offering and treatment
- Resource discovery and monitoring
- Resource selection

Some solutions for these have been pointed out.

Although they present special challenges requiring new research, distributed clouds are promising and may grow to be seen in various contexts.

Acknowledgment

This work was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil.

References

[1] V. Valancius *et al.*, "Greening the Internet with Nano Data Centers," *Proc. 5th Int'l. Conf. Emerging Networking Experiments and Technologies*, 2009, pp. 37–48.

[2] K. Church, A. Greenbreg, and J. Hamilton, "On Delivering Embarrassingly Distributed Cloud Services," *VIII Hotnets*, Citeseer, 2008.

[3] A. Chandra, and J. Weissman, "Nebulas: Using Distributed Voluntary Resources to Build Clouds," *Proc. 2009 Conf. Hot Topics in Cloud Computing*, 2009.

[4] A. Haider, R. Potter, and A. Nakao, "Challenges in Resource Allocation in Network Virtualization," *20th ITC Specialist Seminar*, 18–20 May 2009, Hoi An, Vietnam.

[5] N. M. K. Chowdhury and R. Boutaba, "A Survey of Network Virtualization," *Computer Networks: Int'l. J. Comp. and Telecommun. Networking*, Apr. 2010, pp. 862–76.

[6] S. Khan, A. Maciejewsk, and H. Siegel, "Robust CDN Replica Placement Techniques," *IEEE Int'l. Symp. Parallel & Distrib. Processing*, 2009.

[7] I. Houidi *et al.*, "Virtual Resource Description and Clustering for Virtual Network Discovery," *Proc. IEEE ICC Wksp. Network of the Future*, Dresden, Germany, June 2009.

[8] M. Nelson, "Building a Open Cloud," *Science*, vol. 234, 2009, pp. 1656–57.

[9] T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," *IEEE Int'l. Conf. Advanced Info. Networking and Apps.*, 2010, pp. 27–33.

[10] A. Sheth, and A. Ranabahu, "Semantic Modeling for Cloud Computing, Part I," *IEEE Computer Society — Semantics & Services*, 2010.

[11] C. A. Yfoulis, and A. Gounaris, "Honoring SLAs on Cloud Computing Services: A Control Perspective," *Proc. EUCA/IEEE Euro. Control Conf. 2009.*

[12] C. Chapman *et al.*, "Software Architecture Definition for On-Demand Cloud Provisioning," *Proc. 19th ACM Int'l. Symp. High Performance Distrib. Computing*, 2010, pp. 61–72.

[13] Q. Zhang, L. Cheng, and R. Boutaba. "Cloud Computing: State-of-the-Art and Research Challenges," *J. Internet Services and Apps.*, Springer, 2010, pp. 7–18.

[14] I. Houidi, W. Louati, and D. Zeghlache, "A Distributed Virtual Network Mapping Algorithm," *Proc. IEEE ICC*, 2008, pp. 5634–40.

[15] R. Urgaonkar *et al.*, "Dynamic Resource Allocation and Power Managementin Virtualized Data Centers," *IEEE NOMS)*, 2010.

[16] S. Agarwal *et al.*, "Volley: Automated Data Placement for Geo-Distributed Cloud Services," *Proc. 7th USENIX Conf. Networked Sys. Design and Implementation*, 2010.

[17] N. M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Network Embedding with Coordinated Node and Link Mapping," *IEEE INFOCOM*, 2009, pp. 783–91.

[18] Y. Zhu, and M. Ammar, "Algorithms for Assigning Substrate Network Resources to Virtual Network Components," *Proc. IEEE INFOCOM*, 2006.

Biographies

PATRICIA TAKAKO ENDO (patricia@gprt.ufpe.br) received her M.S degree from the Federal University of Pernambuco (UFPE), Recife, Brazil, in 2008. She is currently a Ph.D. candidate in science computing at the same institution, and a professor of computer networks and distributed systems at the University of Pernambuco (UPE). She also works at GPRT, a research group in the areas of computer networks and telecommunications. Her current research interests include quality of service and cloud computing.

ANDRÉ VITOR DE ALMEIDA PALHARES (andre.vitor@gprt.ufpe.br) graduated in 2009 in computer engineering from the Computer Science Department of UFPE. Currently he is a Master's degree candidate at the same university and a researcher at GPRT. His current areas of interest are cloud computing, and network virtualization and optimization algorithms.

NADILMA NUNES PEREIRA (ncvnp@gprt.ufpe.br) received her M.S. degree from UPE in 2010. She is currently a Ph.D. candidate in science computing at UFPE. Her current research interests include ad hoc networks, routing, and artificial intelligence.

GLAUCO ESTACIO GONÇALVES (glauco@gprt.ufpe.br) received his M.S. degree from UFPE. He is currently a Ph.D. candidate in science computing at the same university. He also works at GPRT. His current research interests include systems performance evaluation, network management, and cloud computing.

DJAMEL HADJ SADOK (jamel@gprt.ufpe.br) received his Ph.D. degree from Kent University in 1990. He is currently a professor in the Computer Science Department of UFPE. He is one of the cofounders of GPRT. His current research interests include traffic engineering, wireless communications, broadband access, and network management. He leads a number of research projects with many telecommunication companies. He has authored many papers and registered some U.S. patents.

JUDITH KELNER (jk@gprt.ufpe.br) received her Ph.D. from the Computing Laboratory at the University of Kent at Canterbury, United Kingdom, in 1993. She is currently a professor at the Computer Science Department of UFPE. She also works at GPRT. Currently she is involved in a number of research projects in the areas of network management, multimedia systems, the design of virtual reality systems, and advanced communication devices.

BOB MELANDER (bob.melander@ericsson.com) received his Ph.D. degree from Uppsala Universitet in 2003. He is currently a research engineer at Ericsson Research.

JAN-ERIK MÅNGS (jan-erik.mangs@ericsson.com) is currently a senior researcher engineer at Ericsson Research.