Cost Optimization of Virtual Machine Provisioning in Federated IaaS Clouds

Fermin A. Armenta-Cano and A. Tchernykh are with *Computer Science Department, CICESE Research Center, Ensenada, B.C., México (e-mail: jcortes@cicese.edu.mx, chernykh@cicese.mx).*
Ramin Yahyapour *with GWDG - Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Germany, (e-mail: ramin.yahyapour@gwdg.de)*
Jarek Nabrzyski *with Center for Research Computing, University of Notre Dame, (e-mail: naber@nd.edu)*

Abstract.

In this paper, we present cost optimization model in cloud computing, and formulate the cost-aware resource allocation problem that provides cost-efficiency in the context of the cloud federation. Our model assumes a cloud provider with multiple heterogeneous resources or data centers. The provider needs to control amount of resources to avoid overprovisioning and increasing capital costs. To reduce an importance of known Build-To-Peak approach that means building infrastructures for top demands with over-provisioning in total operating time, cloud provider has to collaborate with other providers to be able to fulfil requests during peak demands by using idle resources of other peers. In this scenario, it is important to find a trade-off that allows reducing the total investment and operational cost. We address cost minimization problem in the hierarchical federated cloud environment, where external clouds are parameterized by renting costs per time unit. We discuss several cost optimization algorithms in distributed computer environments with the goal to understand the main characteristic of the cost optimization. We conclude by showing how none of these works directly addresses the problem space of the considered problem, but do provide a valuable basis for our work.

Keywords. Cloud computing, IaaS, Operational Cost, Service Level Agreement.

I. Introduction

Cloud Computing is an innovative distributed computing paradigm that is widely accepted by public and private organizations. It focuses on providing three main types of services through the Internet with quality of services to their customers: SaaS, PaaS and IaaS.

Software as a Service (SaaS) is a model of software deployment whereby a provider offers a software application on the internet rather than a software package to be buying it for the costumer. Examples are online email providers like Google Gmail, Microsoft hotmail, Google docs, and Microsoft Office 365. Platform as a Service (PaaS) fills into the system level, which provides platform to run end user's applications without downloads or installation. Examples are the Google App Engine, which allows applications to be run on Google's infrastructure.

Infrastructure as a Service (IaaS) model the provider offers hardware resources such as storage capacity and power computing resources like CPU capacity and memory capacity as a service over the

internet. In this way the costumers rent only the necessary resources for they need instead to buy all the equipment. One of the leading vendors that provide this service is Amazon Web Services (EC2 and S3) for processing and storage.

In this paper, we focus on the IaaS type of clouds.

Service Level Agreement (SLA) is a business component of extreme importance in Cloud computing, which represents a contract that specifies the minimum obligations of the provider to its customers, or expectations of the customers to receive in exchange of the price paid.

Two most important IaaS cloud challenges that providers must addressed are the energy efficiency and cloud provider costs.

IT companies must meet global and national goals for carbon-footprint reduction, and must compensate for noticeable increases in energy expenditures. Thus, technological energy saving measures are mandatory ingredients for any emerging information and communication technology. In this context, IaaS cloud providers must evaluate and assess different strategies to improve energy efficiency in their data centers, including computing, cooling, and power supply equipment. This involves defining and using unified metrics, such as the power usage effectiveness (PUE) or data center infrastructure efficiency (DCIE) metrics, among others. These help cloud providers measure their data centers' energy efficiency, compare the results against other cloud infrastructures, and decide what improvements are necessary.

Some open issues include developing more efficient energy-proportional servers that consume power proportionally to utilization level and improving cloud resource allocation, consolidation, and migration strategies that consider each particular service's workload profile. For example, the reallocation of service components for consolidating purposes can be efficient from a power-saving perspective, but can be counterproductive for service performance when workloads have tightly coupled communications requirements. Another future direction is to study mechanisms for advanced resource provisioning, based on a service's historical execution profile, to predict the resources that the service will consume, allowing for optimal provisioning those results in lower energy consumption.

II. Provider Costs

The costs are changing slightly depending on whether the cloud is public or private, their general structures are similar.

Provider costs are primarily tied to their assets and the maintenance of these assets. For example, providers have an infrastructure that needs to be powered and cooled. Similarly, storage providers have storage arrays containing storage disks, and these arrays are connected to chassis which are all housed in data centers. So, major provider costs can be categorized as follows [1]:

1. Servers cost (compute, storage, software)
2. Infrastructure cost (power distribution and cooling, data center building, etc.)
3. Power draw cost (electrical utility costs)
4. Network cost (links, transit, equipment)

A number of other costs exist.

Optimization is very important for providers to

offer competitive prices to prospective customers. Inefficient resource management has a direct negative effect on performance and cost.

In the shared environments, it is often difficult to measure costs, usage, and value of virtual and physical resources and the services they provide. It is a challenge optimizing the costs of running workloads, usage costs, and defining billing rates to cover total cost of ownership. Detailed cost management can optimize resource usage and improve profitability. An effective step, then, is to measures resource usage at granular levels.

The main objective of IaaS providers is to obtain maximal profits and guarantee QoS requirements of customers. Efficient resource allocation strategies should be exploited in dynamic environment to provide needed quality of service. Graham [2] demonstrated that the problem of scheduling jobs on a set of heterogeneous resources is NP-complete.

On the other hand, more challenges should be met, when providers do not have enough resources to satisfy all customers. Several policies could be used:

- The provider may buy services from other providers to meet the customer‘s requirements established in the SLA. In this scenario, if a new task arrives, the scheduler must analyse whether it is more efficient to allocate the task to other cloud providers or reallocate existing tasks on external resources.
- The provider could invest in additional computational resources.
- Redistribute own computational resources from other services to increase cloud service capability. The main idea is to set a cloud resource admissibility threshold, and dynamically adapt it to cope with different objective preferences, and workload properties.

The multi objective nature of the scheduling problem in clouds makes it difficult to solve. In different cases, we have to consider different performance criteria such as response time, number of deadline violations, resource cost, operational cost, income, energy consumption, etc.

Federated cloud unifies different cloud resources and computing capability, even if they are owned by different organizations, to overcome resource limitations of each cloud, and to enable an unlimited computing capability.

Due to this technology, enterprises can choose an on-demand computing environment.

Cloud federation allows reducing an importance of known Build-To-Peak approach that means building infrastructures for top demands with over-provisioning in total operating time. The infrastructure can be scalable up to the certain limit but not for peak requirements, when resources of other providers can be used. In this scenario, it is important to find a trade-off that allows reducing the total investment and operational cost. Providers must propose efficient resource allocation strategies to guarantee QoS based on the SLA with customers, and make an intelligent decision on outsourcing requests to other providers.

On the other hand, providers should propose efficient cost minimization strategies and optimize different operating expenses that will be generated during the execution of a given workload due to most of the resource allocation strategies today

are non-provider-pricing-based. To this end, we have to look at the total costs in details to discover how much expense is incurred in different usage categories.

In this study, we consider four categories:

1. The cost of resources that are turned off.
2. The cost of resources that are turned on but not used.
3. The cost of resources that are in use.
4. The cost of resources of other providers in the federated cloud.

The first category includes capital costs on hardware/software, upgrades, etc. The second one includes running and maintenance costs, electricity billing, etc. The third category includes additional expenses for extra power consumption of loaded processors, extra cooling, for using hard disks, memory, databases, repairing, etc. The last one includes costs of the use of resources of other providers when local provider requests during peak demands.

Taking a resource offline does not mean shutting off services, and the operating spends for unused or under-utilized resources.

Cloud providers need to control amount of resources to avoid overprovisioning and increase capital costs. Also they need to optimize provisioning local and external resources requested by own customers and by other providers, so they can stay within budgets.

They need also to be able to manage their pricing plans to cover costs and meet profitability targets. This issue is not addressed here.

III. Problem definition

We address cost minimization problem in the hierarchical federated cloud environment, where independent clouds of different providers collaborate to be able to fulfill requests during peak demands and negotiate the use of idle resources with other peers.

A. Infrastructure model

Cloud computing infrastructure model for resource management typically assumes a homogeneous collection of hardware in one data center. Here, we extend this model to provide a cloud-based access to several data centers of one provider, which contain heterogeneous architectures, with different number of cores, execution speed, energy efficiency, amount of memory, bandwidth, operational costs, etc.

Let us consider that the cloud C consists of m nodes (data centers, sites) $D_1, D_2,\ldots,D_m$. Each node D_i, for all i=1..m, consists of b_i servers (blades, boards) and p_i processors per board. We assume that processors in the data center are identical and have the same number of cores. Let m_i be the number of identical cores of one processor in D_i. We denote the total number of cores belonging to the data center D_i by $m_i = b_i \cdot p_i \cdot m_i$, and belonging to all data centers of the cloud C by $m = \sum_{i=1} m_i$. The processor of data center D_i is described by a tuple $\{m_i, s_i, mem_i, band_i, eff_i\}$, where s_i is a measure of instruction execution speed (MIPS), mem_i is the amount of memory (MB), $band_i$ is the available bandwidth (Mbps), and eff_i is energy efficiency (MIPS per watt). We assume that data centers have enough resources to execute any job but their resources are limited.

In addition, to satisfy requests during the peak demands that exceed the capacity of the cloud C, it collaborates with k external independent clouds

(sites) $C_1,C_2,\ldots,C_k$. Each cloud C_i is characterized by the given price per time unit of the allocated instances on a pay-as-you-go basis $q_1,q_2,\ldots,q_k$.

B. Cost model

In cloud computing, a critical goal is to minimize the cost of providing the service. In particular, this also means minimizing energy consumption and maximizing resource utilization. In this section, we first present the cost model of running workloads in the cloud C. Then we define the cost model of cloud federation.

As we mentioned in Section 2, we look at the total costs in details to discover how much expense is incurred in different usage categories.

In this study, we consider three costs of resources: turned off (off); turned on but not used (idle), in use (used).

Let $q\_off_i^{core}$, $q\_idle_i^{core}$, $q\_used_i^{core}$, $q\_off_i^{proc}$, $q\_idle_i^{proc}$, $q\_off_i^{server}$, $q\_idle_i^{server}$, $q\_off_i^{site}$, $q\_idle_i^{site}$ be the operational costs (prices) per time unit of resources (core, processor, server, data center) when they are turned off (standby mode), turned on, but not used, and in use, respectively.

Let q_i be a price per time unit of the request from local cloud C to cloud C_i to use external resources.

The operational cost of a core at time t consists of a constant part $q\_off_i^{core}$ (cost in the off state) and two variable parts $q\_idle_i^{core}$, and $q\_used_i^{core}$: $q_i^{core}(t)=q_{off_i}^{core} + o_i(t)*(q_{off_i}^{core} + w_i(t)*q\_used_i^{core})$, where $o_i(t)=1$, if the core is on at time t, otherwise, $o_i(t)=0$, and if the core is in operational state at time t, $w_i(t)=1$, otherwise $w_i(t)=0$.

When a core is off, it has an operational cost $q\_off_i^{core}$; when it is on, it has extra cost $q\_idle_i^{core}$, even if it is not performing computations. Therefore, the model assumes that cost of all system components has a constant part regardless of the machine activity. Hence, core in the idle state includes the cost of the core and extra costs related with power consumption, cooling, and maintenance cost. In addition, the core has extra cost $q\_used_i^{core}$, when the core is loaded (in operational mode).

The operational cost of all cores in the processor is

$$q^{cores_{proc}}(t) = \sum_{i=1}^{mi} q_i^{core}(t)$$

The operational cost $q_i^{proc}(t)$ of processor at time t consists of a constant part $q\_off_i^{proc}$(cost in the off state) and one variable parts $q\_idle_i^{proc}$:

$$q_i^{proc}(t) = q_{off_i}^{proc} + o_i(t) * \left(q\_idle_i^{proc} + q^{cores}(t)\right)$$

where $o_i(t) = 1$, if the processor is on at time t, otherwise, $o_i(t) = 0$.
The operational cost of processors in a server is

$$q^{proc}_{server}(t) = \sum^{pi}_{i=1} q^{proc}_{i}(t)$$

The operational cost $q^{server}_i(t)$ of a server at time t consists of a constant part $q\_off^{server}_i$ (cost in the off state) and one variable parts $q\_idle^{server}_i$:

$$q^{server}_i(t) = q_{off\,i}{}^{server} + o_i(t) * \left(q\_idle^{server}_i + q^{cores}_{server}(t)\right)$$

where $o^i(t)=1$, if the server is on at time t, otherwise, $o^i(t)=0$.
The operational cost of the servers in the site is

$$q^{server}_{site}(t) = \sum^{pi}_{i=1} q^{server}_i(t)$$

The operational cost $q^{site}_i(t)$ of a site at time t consists of a constant part $q\_off^{site}_i$ (cost in the off state) and one variable parts $q\_idle^{site}_i$:

$$q^{site}_i(t) = q_{off\,i}{}^{site} + o_i(t) * \left(q\_idle^{site}_i + q^{server}_{site}(t)\right)$$

Total operational cost of the site D_i is

$$Q_i = \sum^{Cmax}_{i=1} q^{site}_i(t)$$

The total cloud C operational cost

$$Q^{cloud} = \sum^{m}_{i=1} Q_i$$

In addition, we consider costs associated with using resources from other providers.
The cost of execution of n_i jobs in cloud C_i is

$$Q^{extcloud}_i = \sum^{n_i}_{j=1} q_i \cdot p_j$$

where q_i is a price per time unit in cloud C_i, and p_j is the job execution time.
The total cost Q^{fed} is calculated as follows

$$Q^{fed} = \sum^{k}_{i=1} Q^{extcloud}_i$$

The total cost that will be generated during the execution of a given workload is defined as

$$Q = Q^{cloud} + Q^{fed}$$

In this paper, in order to evaluate the provider's expenses we consider total operational cost Q criterion. This metric allows the provider to measure the system performance in terms of parameters that helps him to establish utility margins.

In this paper, we consider only the part of the total cost optimization problem assuming given infrastructure.

We do not consider capacity planning for clouds that address the finding near optimal size of the cloud (best size/configuration of the infrastructure) that minimize total investment and operational costs.

C. Job model

We consider n independent jobs $J_1,J_2,\ldots,J_n$ that must be scheduled on federation of clouds [3]. The job J_j is described by a tuple $J_j=(r_j,p_j,d_j,SL_j)$, where r_j is the released time, p_j is the processing time of the job, SL_j is the SLA from a set $SL=\{SL_1,SL_2,\ldots,SL_j,\ldots,SL_k,\}$ offered by the provider [4]. Each SLA represents a SL guarantee, and d_j is the deadline. The release time of a job is not available before the job is submitted, and its processing time is unknown until the job has completed its execution.

Due to the virtualization technique and resource sharing the resources are constantly changed, which causes uncertainty in the assignation of the jobs. A job can be allocated to one cloud only, replication of jobs is not considered. Jobs submitted to the one cloud can be migrated to another one.

The admissible set of data centers for a job J_j is defined as a set of indexes $\{a^j_1,\ldots, a^j_1\}$ of data centers that can be used for allocation of the job J_i.

D. Allocation strategies

To design cost efficient allocation strategies we have to answer several important questions: How much allocation strategies can minimize the operation cost of the provider? How intelligent decisions on outsourcing requests or renting

resources to other providers could be made in the context of multiple IaaS providers? Which additional metrics are useful to describe and analyse trade-off between minimization of the provider expenses and efficiency of the service providing? What type of performance guarantees can be secured when the scheduler is based on the provider cost model? How a balance between under-provision and over-provision be obtained in cloud computing? What changes have to be made to known resource allocation algorithms to use them in cloud environment? How cost-aware scheduling algorithms can change the model complexity?

Table 1 shows the summary of cost-aware allocation strategies.

Table 1.
Cost-Aware Allocation Strategies

Strategy	Description
Random	Randomly allocates jobs to the admissible cloud
MQp -(Min Cost per processor)	Allocates job j to the cloud with the least cost of the cloud per processor at time r_j: $min_{i=1..m}\left(\frac{q_i}{m_i}\right)$. The motivation behind this strategy is to balance the cost between clouds.
MinQ (Min Cost)	Allocates job j to the cloud with the least cost for the job
LBal_Q (Load Balance Cost)	Allocates job j to the cloud with the least standard deviation of the cost per processor (taking into account all clouds) when job j is assigned to it. $\min_{q=1..m}\sqrt{\frac{1}{m}\sum_{i=1}^{m}(Q_i^q-\bar{Q})^2}$, where $Q_{i=1..m}^q = \frac{1}{-}\sum_{a_k=i}(q_k + q_j^q)$ and q_j^q is the cost of job j

IV. Related work.

Several works have addressed operational cost reduction in grid and cloud computing environments. Most of these works evaluated user operational cost, while few of them have considered a provider cost optimization.

CESH-Cost-Efficient Scheduling Heuristics. In [5] authors propose a set of heuristics to cost-efficiently schedule deadline-constrained computational applications on both public cloud providers and private infrastructure. They focus on the optimization problem of allocating resources from both a private cloud and multiple public cloud providers in a cost-optimal manner, with support for application-level quality of service constraints such as minimal throughput or completion deadlines for the application's execution.

FVPM-Federated VEE Placement Optimization Meta-Problem [6]. Authors address efficient provisioning of elastic cloud services with a federated approach, where cloud providers can subcontract workloads among each other to meet peaks in demand without costly overprovisioning. They define a novel Federated Virtual Execution Environments Placement Optimization Meta-Problem (FVPM), where each cloud autonomously maximizes the utility of Virtual Execution Environment (VEE) placement using both the local capacity and remote resources available through framework agreements, and present integer linear program formulations for two local VEE placement optimization policies: power conservation and load balancing.

CMHC- Cost Minimization on Hybrid Cloud [7]. Authors address the problem of task planning on multiple clouds formulated as a mixed integer nonlinear programming problem. The optimization criterion is the total cost, under deadline constraints. Their model assumes multiple heterogeneous compute and storage cloud providers, such as Amazon, Rackspace, ElasticHosts, and a private cloud, parameterized by costs and performance. Results show that the total cost grows slowly for long deadlines, since it is possible to use resources from a private cloud. However, for short deadlines

it is necessary to use the instances from public clouds, starting from the ones with best price/ performance ratio.

Table 2.
Areas Of Algorithms Application

	Algorithm	Data centers	Cloud Computing	Cloud federation	Net data storage	Two level (tier)
CESH	Cost-Efficient Scheduling Heuristics	•		•		•
FVPM	Federated VEE Placement Optimization Meta-Problem	•		•		•
CMHC	Cost Minimization on Hybrid Cloud			•	•	•
AMAW	Autonomic Management of application Workflow			•		•
TBPC	Tradeoffs between Profit and Customer Satisfaction	•		•		•
SSSC	Six Scheduling Strategies on Clouds	•	•			•
RAGF	Resource Allocation Game in Federated Cloud			•		•

Table 3.
Areas Of Algorithms Application

	Centralized	Decentralized	Hybrid	On line	Off line	Clairvoyant	Nonclairvoyant	QoS	Migration cost	Network delay	Static	Dynamic	Migration
CESH	•			•		•		•		•		•	
FVPM		•			•	•		•	•		•		•
CMHC			•		•	•		•	•	•	•		•
AMAW			•		•		•					•	
TBPC	•			•		•		•		•		•	
SSSC			•		•	•		•				•	
RAGF	•		•	•			•	•				•	

Table 4.
Evaluation Criteria

	Utilization	Performance	Response time	Scalability	Provider Cost	User Cost	Throughput	Energy	Deadline
CESH					•		•		•
FVPM	•				•			•	
CMHC		•		•		•			•
AMAW		•		•		•			•
TBPC			•		•				
SSSC		•	•			•			•
RAGF			•			•			•

V. Conclusion

We address the problem of the resource allocation on cloud computing with provider cost-efficiency in the context of the cloud federation, considering QoS.

We present cost model and discuss several cost optimization algorithms in distributed computer environments.

A fundamental design decision in the cloud is how many servers in one data center and how many data centers are optimal. In the future work, we apply our cost model to optimize capacity planning for clouds and find investment cost and operational cost trade off. This is difficult problem to answer even when simple cloud architecture is considered.

References

1. **Greenberg, A., Hamilton, J., Maltz, D. A., & Patel, P. (2008).** "The cost of a cloud: research problems in data center networks," SIGCOMM Comput Commun Rev, vol. 39, no. 1, pp. 68–73.

2. **Graham, R. L., Lawler, E. L., Lenstra, J. K. & Rinnooy Kan, A. H. G. (1979).** "Optimization and Approximation in Deterministic Sequencing and Scheduling: a Survey," in Annals of Discrete Mathematics, vol. Volume 5, E. L. J. and B. H. K. P.L. Hammer, Ed. Elsevier, pp. 287–326.
3. **Schwiegelshohn, U. & Tchernykh A. (2012)**. "Online Scheduling for Cloud Computing and Different Service Levels," in Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International, pp. 1067–1074.
4. **Barquet, A. L., Tchernykh, A. & Yahyapour R. (2013).** "Performance Evaluation of Infrastructure as Service Clouds with SLA Constraints," Comput. Sist., vol. 17, no. 3, pp. 401–411.
5. **Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2011)**. "Cost-Efficient Scheduling Heuristics for Deadline Constrained Workloads on Hybrid Clouds," in 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), pp. 320–327.
6. *Breitgand, D., Marashini, A., & Tordsson, J. (2011).* "Policy-Driven Service Placement Optimization in Federated Clouds," IBM Haifa labs technical report h-0299 .
7. **Malawski, M., Figiela, K., & Nabrzyski, J. (2013).** "Cost minimization for computational applications on hybrid cloud infrastructures," Future Gener Comput Syst, vol. 29, no. 7, pp. 1786–1794.
8. **Kim, H., El-Khamra, Y., Rodero, I., Jha, S., & Parashar, M. (2011)**. "Autonomic management of application workflows on hybrid computing infrastructure," Sci Program, vol. 19, no. 2–3, pp. 75–89.
9. **Chen, J., Wang, C., Zhou, B. B., Sun, L., Lee, Y. C., & Zomaya, A. Y. (2011)**. "Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud," in Proceedings of the 20th international symposium on High performance distributed computing, New York, NY, USA, pp. 229–238.
10. **De Assunção, M. D., Di Costanzo, A., & Buyya, R. (2009).** "Evaluating the CostBenefit of Using Cloud Computing to Extend the Capacity of Clusters," in In Proceedings of the International Symposium on High Performance Distributed Computing HPDC 2009, pp. 11–13.
11. **Xu, X., Yu, H., & Cong, X. (2013)**. "A QoS-Constrained Resource Allocation Game in Federated Cloud," in 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 268–275.

Andrei Tchernykh is a researcher in the Computer Science Department, CICESE Research Center, Ensenada, Baja California, Mexico. From 1975 to 1990 he was with the Institute of Precise Mechanics and Computer Technology of the Russian Academy of Sciences (Moscow, Russia). He received his Ph.D. in Computer Science in 1986. In CICESE, he is a coordinator of the Parallel Computing Laboratory. He is a current member of the National System of Researchers of Mexico (SNI), Level II. He leads a number of national and international research projects. He is active in grid-cloud research with a focus on resource and energy optimization. He served as a program committee member of several professional conferences and a general co-chair for International Conferences on Parallel Computing Systems. His main interests include scheduling, load balancing, adaptive resource allocation, scalable energy-aware algorithms, green grid and cloud computing, eco-friendly P2P scheduling, multi-objective optimization, scheduling in real time systems, computational intelligence, heuristics and meta-heuristic, and incomplete information processing.

Ramin Yahyapour is the executive director of the GWDG University of Göttingen. He has done research on Clouds, Grid and Service-oriented Infrastructures for several years. His research interest lies in resource management. He is a steering group member and on the Board of Directors in the Open Grid Forum. He has participated in several national and European research projects. Also, he is a scientific coordinator of the FP7 IP SLA@SOI and was a steering group member in the CoreGRID Network of Excellence.

Fermin Alberto Armenta-Cano received a bachelor's degree in Electronics Engineering from Technological Institute of Sonora, Mexico in 2005. He earned the M.S. degree in Computer Sciences from CICESE Research Center Ensenada, Baja California, Mexico in 2011. Actually he is a Ph.D. student working on distributed computing and Cloud computing scheduling problems.

Jarek Nabrzyski is the director of the University of Notre Dame's Center for Research Computing. Before that he led the Louisiana State University's Center for Computation and Technology, and earlier he was the scientific applications department manager at Poznan Supercomputing and Networking Center (PSNC). His research interests are resource management and scheduling in distributed and cloud computing systems, decision support for global health and environment and scientific computing. Nabrzyski was involved in more than twenty EC funded projects, including GridLab, CrossGrid, CoreGrid, GridCoord, QoSCoSGrid, Inteligrid and ACGT. Within his last five years at Notre Dame Nabrzyski has been focused on building the Center for Research Computing, a unique, interdisciplinary environment, where strong groups of computational and computer scientists and research programmers work side by side with