

Collaborative Marketplaces for eScience: a Medical Imaging Use Case

Mengsong Zou¹, Javier Diaz-Montes¹, Ivan Rodero¹, Manish Parashar¹, Ioan Petri²,
Omer Rana³, Xin Qi^{4 5} and David J. Foran^{4 5 6}

¹ *Rutgers Discovery Informatics Institute, Rutgers University, USA*

² *School of Engineering, Cardiff University, UK*

³ *School of Computer Science & Informatics, Cardiff University, UK*

⁴ *Department of Pathology and Laboratory Medicine, Rutgers - Robert Wood Johnson Medical School, USA*

⁵ *Center for Biomedical Imaging and Informatics, Rutgers Cancer Institute of New Jersey, USA*

⁶ *Department of Radiology, Rutgers - Robert Wood Johnson Medical School, USA*

contact author: javier.diazmontes@gmail.com

Abstract—The development of digital imaging technology is creating extraordinary levels of accuracy that provide support for improved reliability in different aspects of the image analysis such as content-based image retrieval, image segmentation and classification. This has dramatically increased the volume and generation rates of data, which make querying and sharing non-trivial, and render centralized solutions infeasible. Moreover, in many cases this data is naturally distributed or has to be shared across multiple institutions requiring decentralized solutions. In this paper, we present a federation approach that can take advantage of advanced cyber-infrastructure (ACI) by seamlessly and securely interacting with information/data located across geographically distributed resources. We describe and evaluate the establishment of a federated marketplace across resources from UK and US in the context of collaborative research in medical imaging. We show how users from different sites can transparently access to remote data by placing computational requests in the marketplace. Moreover, we evaluate different pricing policies to demonstrate how site providers and users can benefit from such federation.

I. INTRODUCTION

Medical imaging is a very intensive area of study with, by 2010, more 5 billion medical imaging studies been conducted worldwide [32]. It includes millions of applications in different disciplines such as radiology, pathology and quantitative biology. Multimodal datasets, from macro to micro resolutions, will make studies ranging from human organ, tissue to cell and molecular levels a reality. These studies enable personalized medicine with medical decisions, practices and/or products being tailored to individual patients. Digital image processing is an interdisciplinary field at the intersection of biomedical imaging, computer science, data science, mathematics, physics and medicine. This field develops computational and mathematical methods for solving problems pertaining to medical images and their use for

biomedical research and clinical care. These methods can be grouped into several broad categories namely: image segmentation, content-based image retrieval, and image-based physiological modelling.

Content-based image retrieval (CBIR) applies computer vision techniques to search for specific content in digital images from large datasets. The image-based content refers to various features, colors [7, 33, 36], shapes [25], textures [18, 21] or any other information [34, 36] that can be derived from query and candidate images accompanying the searching process to the one similar to query image. The performance of the retrieval and usability are usually ranked by some criteria, such as appearance similarity or diagnostic relevance. Over the last decade, there has been a growing interest in the predictive assessment of disease or therapy course. Image-based physiological modelling, such as biomechanical or physiological nature, can extend the possibilities of medical imaging from a descriptive to a predictive angle of view. The International Union for Physiological Science (IUPS) has been sponsoring the IUPS Physiome Project for more than a decade [4, 15]. This is a worldwide public domain effort to provide a computational framework for understanding human physiology. It aims at developing integrative models at levels of biological organization, from genes to the whole organism via gene regulatory networks, protein pathways, integrative cell functions, and tissue and whole organ structure/function relations. Such an approach aims at transforming current practice in medicine and therapy to a new era of computational medicine [41].

Near real-time processing of images has become essential for a wide range of these biomedical researches, especially for handling with the multimodal datasets. Large-scale, multiple sites collaboration has become indispensable for a wide range of research that rely on the capacity of individuals do dynamically acquire, share, and assess images and correlate data. To design, develop,

evaluate, and maintain a web-based model for interactive telemedicine, intelligent archiving, and automated decision support in biomedical imaging is an overwhelming trend in the near future. However, the heterogeneous execution speed and memory capacity of distributed computing infrastructures makes processing large data sets with multiple sites challenging.

Federated computing has been explored in various contexts and has been demonstrated as an attractive and viable model for effectively harnessing the power offered by distributed resources [3, 5, 13, 17, 27, 30]. More recently, cloud federations are being explored as means to extend as-a-service models to virtualized data-centers federations [11, 23, 31, 39]. Clouds offer on-demand access to computing utilities, an abstraction of unlimited computing resources, customizable environments, and a pay-as-you-go business model. They have a potential for scale-up, scale-down and scale-out as needed, and for IT outsourcing and automation. By federating private and public clouds with local data centers, it is possible to create hybrid cloud infrastructures. This creates interesting marketplaces where users can take advantage of different types of resources, quality of service (QoS), geographical locations, and pricing models.

In this work, we describe how a federated marketplace can provide participants access to a larger number of resources and types of data. We use an application use case in the context of cancer research and present a scenario where different sites (e.g. hospitals, research centers) have different types of data that they are willing to share with other participants. We show how the federation is able to overcome the computational limitations of a single site by collaboratively distributing the workload across sites. Moreover, we show how sites providers can increase their profit by having a dynamic pricing policy.

The rest of the paper is organized as follows. Section II collects the related work. Sections III and IV present the medical imaging techniques and our federation model, respectively. Section V describes the federated environment used for our experiments and Section VI details the behaviour of the marketplace. Finally, in Section VII we discuss the results of our experiments and in Section and VIII we present the conclusions of this work.

II. RELATED WORK

Cloud computing can provide effective platforms for certain classes of applications, for example high-throughput computing (HTC) applications [11]. There have been several early projects that have reported successful deployments of applications on existing clouds [10, 12, 16, 38]. Additionally, there are efforts exploring other usage modes [23] and to combine clouds, such as Amazon EC2 [1], with integrated computing

infrastructures. Villegas et al. [39] proposed a composition of cloud providers as an integrated (or federated) cloud environment in a layered service model. Assuncao et al. [9] described an approach of extending a local cluster to cloud resources using different scheduling strategies. Along the same lines, Ostermann et al. [22] extended a grid workflow application development and computing infrastructure to include cloud resources, and experimented with Austrian Grid and an academic cloud installation of Eucalyptus using a scientific workflow application. Similarly, Vazquez et al. [37] proposed architecture for an elastic grid infrastructure using the GridWay meta-scheduler, and extended grid resources to Globus Nimbus; Vockler et al. [40] used Pegasus and Condor to execute an astronomy workflow on virtual machine resources drawn from multiple cloud infrastructures based on FutureGrid, NERSC's Magellan cloud and Amazon EC2; Gorton et al. [14] designed a workflow infrastructure for Systems Biology Knowledgebase (Kbase) and built a prototype using Amazon EC2 and NERSC's Magellan cloud; and Bittencourt et al. [6] proposed an infrastructure to manage the execution of service workflows in the hybrid system, composed of the union of a grid and a cloud.

Given the growing popularity of virtualization, many commercial products and research projects, such as OpenNebula [35], OpenStack [43], Nimbus [42], Eucalyptus [20], IBM Smart Cloud [2], Amazon EC2, and VMware vCloud Connector are being developed to dynamically overlay physical resources with virtual machines. Analogously, Riteau et al. [28] proposed a computing model where resources from multiple cloud providers are leveraged to create large-scale distributed virtual clusters. They used resources from two experimental testbeds, FutureGrid in the United States and Grid'5000 in France. Celesti et al. [8] proposed a cross-federation model based on using a customized cloud manager component placeable inside the cloud architectures. Other example are the Reservoir [29] and Siena [44] that aims at contributing to best practices with a cloud and federation architecture and defining standards for cloud interoperability, respectively. These efforts are intended to extend the benefits of virtualization from a single resource to a pool of resources, decoupling the VM not only from the physical infrastructure but also from the physical location.

III. CONTENT-BASED IMAGE RETRIEVAL

With the advent of whole-slide virtual microscopy technologies, the emphasis has shifted to the development of methods capable of performing reliable, sub-region searching to identify image patches exhibiting specific patterns or structural characteristics. Researchers at the Cancer Institute of New Jersey recently

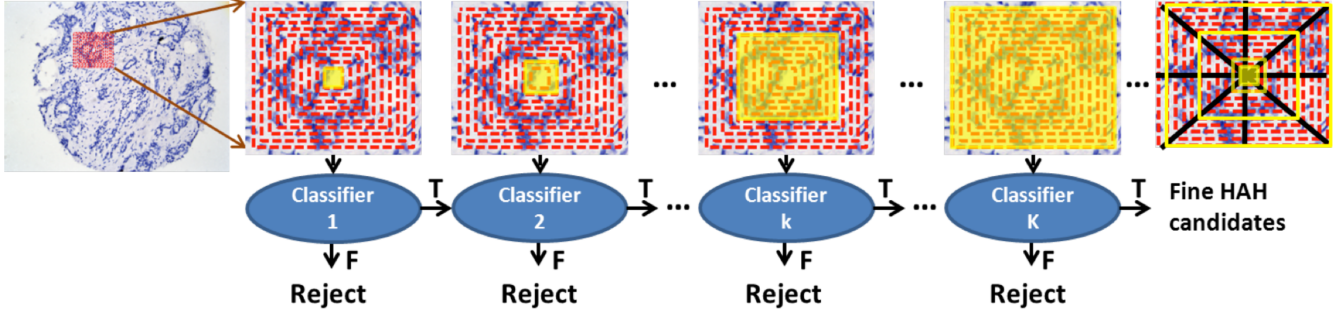


Figure 1: A demonstration of hierarchical searching process based on HAH and fine HAH. In every iteration, a certain percentage of candidates will be discarded and the survived candidates will be refined in the second stage

focused our efforts on developing methods that enable pathologists to select an input query (area or object of interest) within a digitized specimen to reliably identify other regions within either the same specimen or across multiple specimens corresponding to a large patient cohort to support characterization of different types and stages of disease. It includes content localization, hierarchical searching process, and performance evaluation. The performance of the retrieval and usability are usually ranked by some criteria, such as appearance similarity or diagnostic relevance.

Xin et.al. [26] present a novel feature measure called hierarchical annual histogram (HAH) and fine HAH. For a given image patch, first it is segmented into several closed bins with equal intervals with high weights in center and decreasing weights outwards. Next, color histogram of each bin is calculated on each color channel and concatenated together. For fine HAH, the whole patch is equally divided into eight segments, from which each HAH is calculated and concatenated together. The characteristics of HAH and fine HAH are as following: (1) it is scale and rotation invariant; (2) it captures spatial configuration of image features; and (3) it is suitable for hierarchical searching in the sub-image retrieval. Because HAH and fine HAH significantly improve discriminative power of feature descriptor, they can easily differentiate image patches with similar intensity distribution. The hierarchical searching process contains three stages: rough searching, fine searching, and clustering. The hierarchical searching scheme is an iterative process that discards less similar candidates within each iteration. The hierarchical searching procedure can greatly reduce the computation load as it only computes one bin of HAH or any image texture feature and rejects a large portion of candidates within each iteration. The number of candidates moving into the next iteration is largely reduced by rejecting the obvious negative ones. Mean-shift clustering [127] is applied at last stage of searching to remove the redundancy of the top ranked candidates,

which could be caused by the high correlation among those adjacent candidates. Figure 1 demonstrates the whole hierarchical searching process.

IV. FEDERATION MODEL

Our federation model [11, 24] is designed to be dynamically updated as it is created in a collaborative way, where each site communicates with others to identify itself, negotiate the terms of interaction, discover available resources, and advertise their own resources and capabilities. In this way, a federated management space is created at runtime and sites can join and leave at any point. This federation model does not have any centralized component and users can access the federation from any site, which increases the fault tolerance of the overall federation, see Figure 2. Another key benefit of this model is that since each site can differentiate itself based on the availability of specialist capability, it is possible to schedule tasks to take advantage of these capabilities.

The federation model is based on the Comet [19] co-ordination “spaces” (an abstraction, based on the availability of a distributed shared memory that all users and providers can access and observe, enabling information sharing by publishing requests/offers to/for information to this shared memory). In particular, we have decided to use two kinds of spaces in the federation. First, we have a single federated management space used to create the actual federation and orchestrate the different resources. This space is used to exchange any operational messages for discovering resources, announcing changes at a site, routing users’ request to the appropriate site(s), or initiating negotiations to create ad-hoc execution spaces. On the other hand, we can have multiple shared execution spaces that are created on-demand to satisfy computing needs of the users. Execution spaces can be created in the context of a single site to provision local resources or to support a *cloudburst* to public clouds or external high performance computing systems. Moreover, they can be

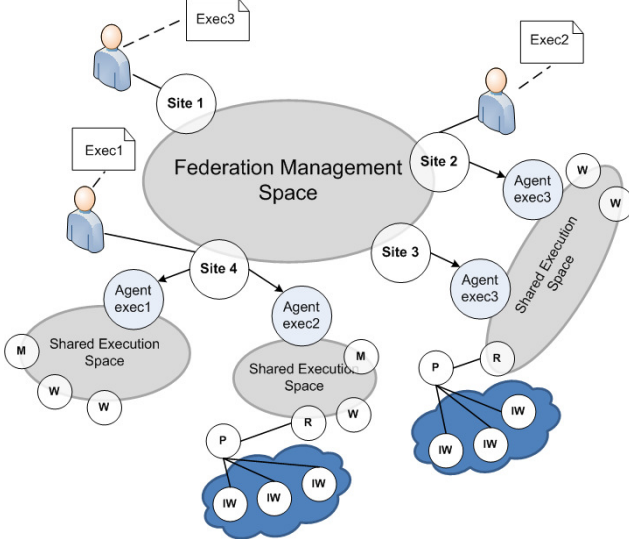


Figure 2: The overall Federation Management Space, here (M) denotes a master, (W) is a worker, (IW) an isolated worker, (P) a proxy, and (R) is a request handler.

used to create a private sub-federation across several sites. This case can be useful when several sites have some common interest and they decide to jointly target certain types of tasks as a specialized community.

As shown in Figure 2, each shared execution space is controlled by an agent that initially creates such space and subsequently coordinates access to resources for the execution of a particular set of tasks. Agents can act as a master node within the space to manage task execution, or delegate this role to a dedicated master (M) when some specific functionality is required. Moreover, agents deploy workers to actually compute the tasks. These workers can be in a trusted network and be part of the shared execution space, or they can be part of external resources such as a public cloud and therefore in a non-trusted network. The first type of workers are called secure workers (W) and can pull tasks directly from the space. Meanwhile, the second type of workers are called isolated workers (IW) and cannot interact directly with the shared space. Instead, they have to interact through a proxy (P) and a request handler (R) to be able to pull tasks from the space.

V. METHODOLOGY

In this paper, we investigate the use of sub-image CBIR in several key pathology applications including peripheral blood smears (BS), glomeruli of the kidney (GK), and breast tissue microarrays (BTMA). We present an scenario where different sites offer different datasets. We consider that each dataset has a single type of pathology data, see Table I. This represents

a typical scenario where different hospitals or research centers have agreements on sharing their databases to support collaborative research. Although the data is anonymized to protect patients' privacy, our federation also implement security mechanisms to encrypt the communications across sites. In our experiments we assumed that the data is not allowed to be staged outside of its origin and therefore we transfer the required data every time a task needs it.

Table I: Location of each dataset

Site	Dataset
Cardiff	Breast tissue microarrays (BTMA)
FutureGrid	Glomeruli of the kidney (GK)
Rutgers	Blood smears (BS)

CometCloud software stack has been developed using Java and it is shipped in a convenient package ready to use. The federation is created ad-hoc where peers talk with each other to negotiate the terms of adhesion. In order for a site to join the federation, it has to support Java, have valid credentials (authorized SSH keys), and configure some parameters (i.e. address, ports, number of workers). Once this configuration is done, we can easily launch the different CometCloud components using simple scripts and the site will join the federation, or it will create it if it is the first one.

The federated marketplace used here is composed by three different sites geographically distributed. Cardiff site is located at Cardiff University, Cardiff, UK; FutureGrid site is located at Indiana University, Indiana, USA; and Rutgers site is located at Rutgers University, New Jersey, USA. Table II shows the bandwidth of the interconnection network between sites. Table IV collects the prices assigned to each site.

Table II: Network speed in MB/s.

Network	Cardiff	FutureGrid	Rutgers
Cardiff	-	5	5
FutureGrid	5	-	10
Rutgers	5	10	-

Table III: Price assigned to the resource of each site (based on Amazon EC2 pricing)

Site	Computing	Network
Cardiff	0.000033 \$/s	0.0001 \$/MB
FutureGrid	0.000017 \$/s	0.0001 \$/MB
Rutgers	0.000067 \$/s	0.0001 \$/MB

Next, we provide additional implementation details within the three federated sites.

Cardiff Federation site: We deploy our architecture on a cluster-based infrastructure with 12 dedicated cluster machines. Each machine has 12 CPU cores at 3.2 GHz CPU speed. Each physical machine uses a KVM (Kernel-based Virtual Machine) virtualization environment with each entity (master, workers, request-handler) within our system, occupying a single virtual machine. Each virtual machine runs Ubuntu Linux utilising one 3.2GHz core with 1GB of RAM and 10 GB storage capacity. The networking infrastructure is Gigabit Ethernet with a speed of 1 gigabit per second (Gbps). The measured latency on the network is 0.706 ms on average.

Rutgers Federation site: The Rutgers federation site is deployed on a cluster-based infrastructure with 32 nodes. Each node has 8 CPU cores at 2.6 GHz, 24 GB memory, 146 GB storage and Gigabit Ethernet connection. The measured latency on the network is 0.227 ms on average.

FutureGrid Federation site: The FutureGrid federation site is deployed on a cloud infrastructure based on OpenStack. In particular, we have used the infrastructure located at Indiana University. We have used instances of type medium, where each instance has 2 cores and 4 GB of memory. The networking infrastructure is DDR Infiniband and the measured latency of the cloud virtual network is 0.706 ms on average.

We have three types of tasks, one per type of pathology, where each task involves the computation of multiple sub-tasks depending on the size of the dataset. In our case we consider that all three datasets have 100 images. Table IV collects the average execution time for each type of pathology for different sites, based on actual executions. The task distribution is generated using a Poisson distribution and the size of the input data is randomly generated between 1MB and 100MB, which amounts to 5GB.

Table IV: Execution time for single sub-tasks in seconds

Site	BS	GK	BTMA
Cardiff	360s	600s	830s
FutureGrid	500s	700s	900s
Rutgers	300s	650s	770s

VI. MARKETPLACE DETAILS

In our federated marketplace, each site subscribes to the type of tasks that they are interested on computing. When a new request is submitted into the tuple-space, a blind auction process is initiated. Here, each site that is interested in that type of computation is notified. Then, each site calculates the time to complete (TTC) all the sub-tasks involved in the request and if they can complete the task before the required deadline, an offer is made considering the cost of the computation and the

desired profit. Once all offers are collected, the client that put the task into the space decides which site has the best offer and therefore executes the task.

- *Cost:* The cost of computing a task is calculated as the sum of the cost of computing each sub-task and the cost of transferring the required input data. $cost = \sum_{i=1}^n (exec_time_i \times c + input_size_i \times nc)$, where $exec_time_i$ is the average execution time of a sub-task, c represents the cost per core per second, $input_size$ is the size of the input data required by the sub-task i and nc is the network cost per megabyte (MB). Different sites have different prices, see Table III, and execution time for each type of task, see Table IV. When the required data is stored locally to the site, we do not incur in the network cost.
- *Profit:* The computation of a task involves a cost, but also provides a benefit. Therefore, the total profit of a site Pr is calculated using the benefit b obtained from each task (price that user pays) and the cost $cost$ of computing it, $Pr = \sum_{i=1}^n (b - cost)$.

The auction winning criteria in these experiments will be the ratio between *cost* and TTC.

VII. EVALUATION

We have performed a set of experiments in the previously described infrastructure. This infrastructure includes all the mechanisms needed for the coordination across computational resources which are used by our experiments. However, we have emulated the execution of the tasks to better analyze the behaviour of the marketplace.

In these experiments, we consider two use cases with different pricing policy. We study how the pricing policy affects to the overall profit of each site and the price that users pay for the computation. First, we consider that each site always charges the same amount of money to compute tasks (static pricing). This price is calculated adding a 10% to the *cost* formula described in Section VI, which depends on the computation cost of the site resources and the network cost when needed. Results are collected in Figure 3. We can observe that most of the sites obtain the maximum revenue from the type of tasks that are local to them. The main reason is that data transfer cost has influence in the overall cost of a task (up to 8%). Then, if a site has the data locally available, it eliminates the need for moving data, which reduces the overall cost as well as the TTC. Since it can complete the execution faster, it has higher chances of winning auctions of this type of task and obtain higher profit from it. However, we can see that FutureGrid, which hosts the GK data, obtains the maximum profit from the BTMA. Although, it executed more tasks of the GK

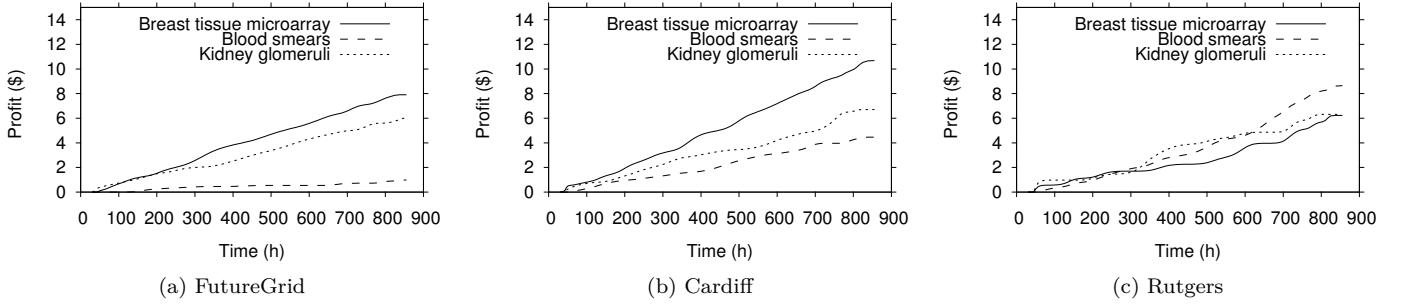


Figure 3: Profit obtained by each site for different type of computation when using a static pricing policy.

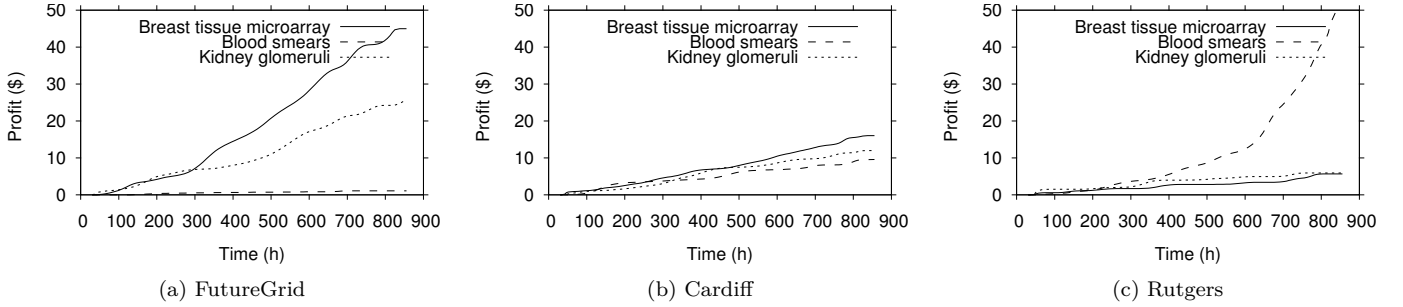


Figure 4: Profit obtained by each site for different type of computation when using a dynamic pricing policy.

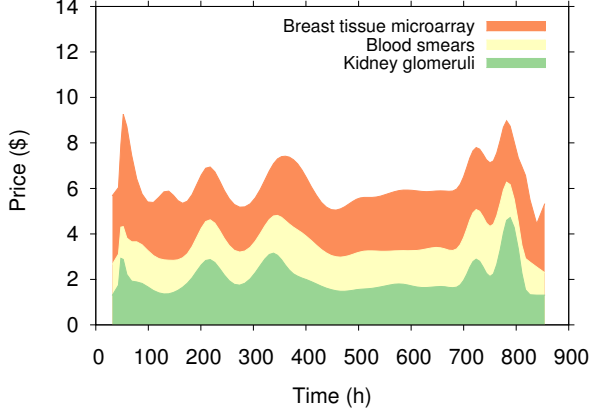


Figure 5: Market price for each type of task when using a static pricing policy.

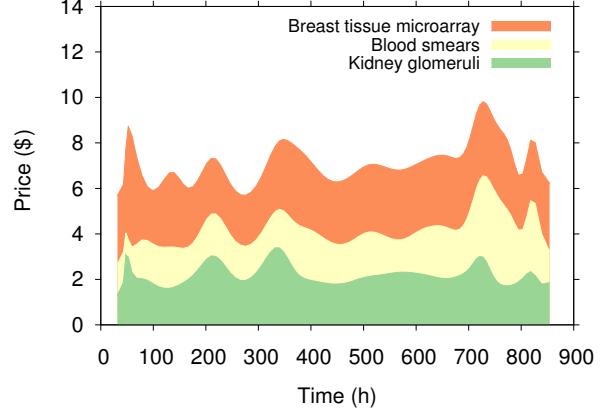


Figure 6: Market price for each type of task when using a dynamic pricing policy.

type, but the BTMA tasks provide more profit due to the longer duration of individual sub-tasks, see Table IV.

Figure 5 shows the market price of each type of computation. Average prices are \$2.5, \$1.6, and \$ 2.1 for BTMA, BS, and GK type of task, respectively. However, the standard deviation is around \$1 in all cases. This high variation is influenced by different sites winning different types of tasks. The reason for this to happen is the different prices of each site and specially the

limitation in the available resources.

Next, we consider that sites have dynamic pricing, in an attempt to maximize their profit. Here, every time that a site wins the same type of task consecutively, this site increases the overall price of such type of task by a 10% over the current market price. When a site wins consecutively the same type of tasks, it can be as a consequence of undercharging for the use of its resources. Therefore, the site progressively increases its prices until

it stops winning that type of tasks. Once this happens, the site know what is the actual market price for that type of computation and it can maximize its profit by staying around that price. Figure 4 collects the profit over time of each site for each type of task.

We can observe, in Figure 4, that as in the previous case, each site tends to obtain higher profit from the tasks that do not involve data movement. In this case, we observe that the overall profit of each site is much higher than in the static pricing case. In particular, by using dynamic pricing, sites can obtain between 40% and 80% more profit.

In Figure 6 we can observe the market price of each type of computation. In this case, average prices are \$2.9, \$2, and \$2.2 for BTMA, BS, and GK type of task, respectively. Again, the standard deviation is around \$1 in all cases due to different sites winning different type of tasks to optimize the use of their resources. The increase in the market price for different type of tasks suppose between 5% and 20% compared with the static pricing policy. Although, sites tried to increase the market prices as much as possible, the competition in our marketplace kept prices within reasonable levels, preventing users from being abused.

VIII. CONCLUSION

In this work we showed how federated computing can help different sites to work in a collaborative way to share their resources and obtain access to data that otherwise will not be possible. We demonstrated that a federated marketplace can be beneficial for both users and site providers. We showed how static pricing policies are not beneficial for site providers as they get very low profit. On the other hand, by introducing price competition in the marketplace, we observed that providers significantly increased their profit by adjusting their prices to their competitors. Moreover, the competition itself prevented users from being unreasonably charged for their computation. Since the federation prevented vendor lock-in, users were able to select the cheapest resource at any given time.

ACKNOWLEDGEMENT

This work is supported in part by the NSF under grants ACI-1339036, IIP-0758566, DMS-0835436, CNS-1305375, ACI-1310283, CAREER-1149365, PHY-0941576, and by IBM via OCR and Faculty awards. This project used resources from FutureGrid supported in part by NSF OCI-0910812.

REFERENCES

- [1] Amazon EC2. <http://aws.amazon.com/ec2/>.
- [2] IBM Smart Cloud. <http://www.ibm.com/cloud-computing/us/en/>.
- [3] G. Allen and D. Katz. Computational science, infrastructure and interdisciplinary research on university campuses: Experiences and lessons from the center for computation & technology. Technical Report CCT-TR-2010-1, Louisiana State University, 2010.
- [4] J. B. Bassingthwaite. Strategies for the physiome project. *Ann. Biomed. Eng.*, 28:1043–1058, 2000.
- [5] F. Berman, G. Fox, and A. Hey. *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons, 2003.
- [6] L. F. Bittencourt, C. R. Senna, and E. R. M. Madeira. Enabling execution of service workflows in grid/cloud hybrid systems. In *Network Operations and Management Symp. Workshop*, pages 343–349, 2010.
- [7] C. Carson, S. Belongies, H. Greenspan, and J. Malik. A generative model for image segmentation based on label fusion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 42–51, 1997.
- [8] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. How to enhance cloud architectures to enable cross-federation. In *IEEE CLOUD*, pages 337–34, 2010.
- [9] M. D. de Assuncao, A. di Costanzo, and R. Buyya. Evaluating the cost-benefit of using cloud computing to extend the capacity of clusters. In *ACM HPDC*, pages 141–150, 2009.
- [10] E. Deelman, G. Singh, M. Livny, et al. The cost of doing science on the cloud: the Montage example. In *Proc. SC*, 2008.
- [11] J. Diaz-Montes, Y. Xie, I. Rodero, J. Zola, B. Ganapathysubramanian, and M. Parashar. Exploring the use of elastic resource federations for enabling large-scale scientific workflows. In *Proc. of Workshop on Many-Task Computing on Clouds, Grids, and Supercomputers (MTAGS)*, pages 1–10, 2013.
- [12] G. Fox and D. Gannon. Cloud Programming Paradigms for Technical Computing Applications. Technical report, Indiana University, 2012.
- [13] G. Garzoglio, T. Levshina, M. Rynge, et al. Supporting shared resource usage for a diverse user community: the OSG experience and lessons learned. *J. of Physics: Conf. Series*, 396, 2012.
- [14] I. Gorton, Y. Liu, and J. Yin. Exploring architecture options for a federated, cloud-based system biology knowledgebase. In *IEEE Intl. Conf. on Cloud Computing Technology and Science*, pages 218–225, 2010.
- [15] P. J. Hunter and T. K. Borg. Integration from proteins to organs: The physiome project. *Nature Rev. Mol. Cell Biol.*, 4:237–243, 2003.
- [16] K. Keahey and T. Freeman. Science clouds: Early experiences in cloud computing for scientific applications. In *Proc. Cloud Computing and Its Applications (CCA)*, 2008.
- [17] A. Kertész, I. Rodero, and F. Guim. Data model for describing grid resource broker capabilities. In *Grid Middleware and Services*, pages 39–52, 2008.
- [18] W. J. Kuo, R. F. Chang, C. C. Lee, W. K. Moon, and D. R. Chen. Retrieval technique for the diagnosis of solid breast tumors on sonogram. *Ultrasound Med. Biol.*, 28:903–909, 2002.
- [19] Z. Li and M. Parashar. A computational infrastructure for grid-based asynchronous parallel applications. In *HPDC*, pages 229–230, 2007.
- [20] D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov. The eucalyptus open-source cloud-computing system. In *IEEE/ACM CCGRID*, pages 124–131, 2009.
- [21] M. Ortega, Y. Rui, K. Chakrabarti, K. Porkaew, S. Mehrotra, and T. S. Huang. Supporting ranked boolean similarity queries in mars. *IEEE Trans. Knowledge Data Eng.*, 10:905–925, 1998.
- [22] S. Ostermann, R. Prodan, and T. Fahringer. Extending grids with cloud resource management for scientific computing. In *IEEE/ACM Grid*, pages 42–49, 2009.
- [23] M. Parashar, M. AbdelBaky, I. Rodero, and A. Devarakonda. Cloud Paradigms and Practices for Computational and Data-Enabled Science and Engineering. *Computing in Science and Engineering*, 15:10–18, 2013.
- [24] I. Petri, T. Beach, M. Zou, and et. al. Exploring models and mechanisms for exchanging resources in a federated cloud. In *Intl. Conf. on cloud engineering (IC2E 2014)*, 2014 - Accepted.
- [25] T. Pfund and S. Marchang-Maillet. Dynamic multimedia annotation tool. In *Proceedings of SPIE Photonics West Conference on Internet imaging*, pages 216–224, 2002.
- [26] X. Qi, R. H. Gensure, D. J. Foran, and L. Yang. Content-based white blood cell retrieval on bright-field pathology images. *Proceeding of SPIE Medical Imaging*, 2013.
- [27] P. Riteau, M. Tsugawa, A. Matsunaga, et al. Large-scale cloud computing research: Sky computing on FutureGrid and Grid’5000. In *ERCIM News*, 2010.
- [28] P. Riteau, M. Tsugawa, A. Matsunaga, J. Fortes, and K. Keahey. Large-scale cloud computing research: Sky computing on future-

- grid and grid'5000. In *ERCIM News*, 2010.
- [29] B. Rochwerger, D. Breitgand, E. Levy, A. Galis, et al. The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53, 2009.
 - [30] I. Rodero, F. Guim, J. Corbalan, and J. Labarta. How the jsdl can exploit the parallelism? In *Cluster Computing and the Grid, 2006. CCGRID 06. Sixth IEEE International Symposium on*, volume 1, pages 8 pp.–282, May 2006.
 - [31] I. Rodero, D. Villegas, N. Bobroff, Y. Liu, L. Fong, and S. M. Sadjadi. Enabling interoperability among grid meta-schedulers. *J. Grid Comput.*, 11(2):311–336, 2013.
 - [32] C. A. Roobottom, G. Mitchell, and G. Morgan-Hughes. Radiation-reduction strategies in cardiac computed tomographic angiography. *Clin Radiol*, 65:859–967, 2010.
 - [33] S. Sclaroff, L. Taycher, and M. L. Cascia. Imagerover: a content-based browser for the world wide web. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pages 2–9, 1997.
 - [34] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the edn of the eacery years. *IEEE Trans. Pattern Anal. Machine Intel.*, 22:1349–1380, 2000.
 - [35] B. Sotomayor, R. Montero, I. Llorente, and I. Foster. Virtual infrastructure management in private and hybrid clouds. *IEEE Internet Computing*, 13:14–22, 2009.
 - [36] D. M. Squire, W. Muller, H. Muller, and T. Pun. Content-based query of image databases: inspirations from text retrieval. *Pattern Recognition Letters*, 21:1193–1198, 2000.
 - [37] C. Vazquez, E. Huedo, R. Montero, and I. Llorente. Dynamic provision of computing resources from grid infrastructures and cloud providers. In *Grid and Pervasive Computing Conf.*, pages 113–120, 2009.
 - [38] C. Vecchiola, S. Pandey, and R. Buyya. High-performance cloud computing: A view of scientific applications. In *Proc. Int. Symp. on Pervasive Systems, Algorithms, and Networks (ISPAN)*, pages 4–16, 2009.
 - [39] D. Villegas, N. Bobroff, I. Rodero, et al. Cloud federation in a layered service model. *J. of Computer and System Sciences*, 78(5):1330–1344, 2012.
 - [40] J.-S. Vockler, G. Juve, and M. R. Ewa Deelman and. Experiences using cloud computing for a scientific workflow application. In *2nd Workshop on Scientific Cloud Computing in conjunction with ACM HPDC*, pages 402–412, 2011.
 - [41] R. L. Winslow, N. Trayanova, D. Geman, and M. I. Miller. Computational medicine: Translating models to clinical care. *Sci. Trans. Med.*, 4:158–169, 2012.
 - [42] Nimbus Project. <http://www.nimbusproject.org/>.
 - [43] OpenStack Project. <http://www.openstack.org/>.
 - [44] Siena Initiative. <http://www.sienainitiative.eu>.