A Model for Evaluating the Economics of Cloud Federation

George Darzanos[*†], Iordanis Koutsopoulos[*†], George D. Stamoulis[*]
[*]Athens University of Economics and Business, AUEB, Athens, Greece
[†]Centre for Research and Technology Hellas, CERTH, Greece
{ntarzanos, jordan, gstamoul}@aueb.gr

***Abstract*—We consider the problem of formation of an economically sustainable computational resource federation by Cloud Service Providers (CSPs). The federation aims at providing improved quality of service, expressed in terms of average total delay per job provided to clients served by the CSPs. Each CSP is modeled as an M/M/1 queue. The queue may serve requests coming from that CSP's own client, as well as requests coming from clients of other CSPs. Our model includes all salient factors to evaluate the economics of such a federation: we model the energy consumption cost of each CSP as a function of its resource utilization factor, and we model the CSP's revenue by a delay-dependent pricing function according to which each CSP charges its clients. We propose a model for the formation and evaluation of a cloud federation according to which each CSP may transfer a portion of the requested workload from its clients to other CSPs within the federation. A federation policy is specified by the portions of workload transferred from each CSP to other CSPs. We formulate the problem of finding the federation policy that maximizes the total profit (revenue minus cost) of CSPs and we also deal with the incentives of individual CSPs. Finally, we conduct several experiments under different setups. The numerical results show that according to our model CSPs can maximize the total profit of the federation and also achieve a nearly optimal QoS.**

Keywords*—Cloud federation, Queueing theory, Resource pooling, Pricing, Cooperation, Profit maximization.

I. Introduction

Cloud Service Providers (CSPs) usually have geographically dispersed servers in order to satisfy client requests through their storage or computational resources. Client requests are themselves arising in different locations. The stream of client requests has time-varying characteristics. Hence, the load at the servers of a CSP is time-varying, and thus the quality of provisioned service (e.g. the job execution delay) is also time-dependent and unpredictable. A solution to alleviate the temporal variation of load requests, would be to invest more in resources (e.g. servers and computational capacity) at the expense of increased costs. A natural means to refrain from this investment is to respond to such load variations by employing *cloud resource federation* policies. In a cloud federation, two or more CSPs offer their resources in a common pool so that resources owned by one CSP can also be used to serve tasks coming from clients of other CSPs.

Several instances of academic cloud federations or commercial platforms already enable in reality the concept of cloud federation through different private and public clouds. The European Grid Infrastructure Federated cloud [1] is a seamless grid of academic private clouds and virtualized resources, that serves the needs of the scientific community. The OnApp Federation [2] constitutes a network of Infrastructure as a Service (IaaS) among CSPs and connects them through the OnApp market, where each member of the federation can buy and sell capacity on demand. The Arjuna's Agility framework [3] has been developed in order to deliver the service agreements and policies that are needed in federations. Another free and open-source cloud computing software that can be used to enable the CSP federation at the IaaS level is OpenStack. Recently, Rackspace and CERN started working together on the CERN Openlab project [4], aiming to build a seamless federation among multiple private and public cloud platforms on OpenStack. Finally, the European FP7 project BonFIRE [5] offers a federated testbed that supports large-scale testing of cloud services over multiple, geographically distributed, heterogeneous cloud and network testbeds.

In a cloud federation, CSPs cooperate and *pool* together their resources in order to improve the QoS of their client requests in a seamless manner. The coordination mechanism of resource pooling should be agreed a priori between the CSPs, and the CSP whose resource is actually utilized to serve the client request is indifferent to the client. Cloud federation mechanisms should be designed to be flexible enough so that any CSP should take part in a federation regardless of the amount of owned resources. On the other hand, the federation mechanism should be appropriately designed so that it provides *incentives* to CSPs to pool (part of) their resources and devote them to serve requests from other clients. These incentives should include a mechanism for profit sharing among the federated CSPs in a way that does not discourage them from joining a federation. There arise several advantages for CSPs when they get involved in a federation. First, a CSP can expand its geographic coverage range and come closer to the client if it uses servers of some other CSP. Moreover, CSPs do not need anymore to over-dimension their infrastructure, since dynamic inter-cloud load balancing can be achieved by outsourcing jobs to federated CSPs in response to peak-demand workloads. This migration of jobs within the federation may (and in fact should) have positive repercussions both for the CSP in terms of cost (e.g. energy) reduction and for the clients in terms of improvement in QoS for their requests.

Different modes of cloud federation have been proposed in the literature; they can be classified into three categories: *(i) Cloud infrastructure aggregation* [6], [7], where different CSPs integrate their infrastructures into one unique virtualized infrastructure, *(ii) hybrid cloud federations* [8], which combines the infrastructures of private and public clouds, and *(iii) brokering* [9], [10], where the cloud federation brings together multiple CSPs into a global marketplace where each participant buys and sells computational capacity on demand.

In this work, we model and study the problem of cloud federation, whereby federated CSPs pool (part of) their infrastructure, with emphasis on the formation of an economically

978-1-4673-9501-4/15/$31.00 ©2015 IEEE

viable federation. Our technical contributions are as follows.

- We develop an abstraction model for the CSP's infrastructure and service through an M/M/1 queueing system.
- We model the salient factors that determine the net benefit (profit) of a CSP, i.e. a pricing function that each CSP uses to charge its clients, and the cost from energy consumption at servers.
- We model a federation policy agreed among CSPs as the portion of jobs' requests transferred from each CSP to other CSPs within the federation in order to be served through their server infrastructure.
- We formulate the problem of finding the federation policy that maximizes total profit for the CSPs, we find the optimal federation policy as the solution of non-linear optimization problem and we provide a rule for the sharing of the generated profit of the federation.

In order to demonstrate our model, we restrict our attention to federations comprising two CSPs. The paper is organized as follows. In section II, we present our model for the CSP as well as our assumptions. In section III we define the federation policy, and we formulate and solve the relevant optimization problem. In section IV we present our numerical results, in section V we discuss some related work, and in section VI we present our conclusions.

II. The Model

A. M/M/1 queueing model abstraction of the CSP

We consider a set of N CSPs. In order to demonstrate our approach, we take $N = 2$. For CSP i, let C_i denote its computational capacity (in flops/sec). The job requests from all clients of CSP i arrive according to a Poisson process of rate λ_i (jobs/sec). The size of the job is expressed in terms of the number of operations it entails. We assume that this size follows the exponential distribution with mean number of operations per job L. Hence, the average service rate (in jobs/sec) for CSP i is $\mu_i = \frac{C_i}{L}$, while the service time of a job is exponentially distributed with mean $\frac{1}{\mu_i}$.

Average task execution delay as a measure of client QoS. Each CSP offers a level of QoS for the tasks of its clients. This QoS is the average task execution delay. By standard queueing theory for the single-server M/M/1 queue, the average delay d_i for jobs served by CSP i infrastructure is given by:

$$d_i = \frac{1}{\mu_i - \lambda_i}. \tag{1}$$

Why is a single-server M/M/1 queueing model reasonable? A typical CSP consists of multiple physical hosts that serve the incoming job requests. When the requests arrive in the CSP, they are translated into VMs and then they are served by the CSP's virtualized infrastructure. Each job spends some time in the system until it is finally served. This time duration depends on the size of the job, the number of existing requests that wait to be served and on the availability of resources when the request arrives; hence, a queueing model is applicable. In order to abstract the system of a CSP with multiple servers and queues, we further assume that perfect dispatching and scheduling of requests without idling of resources is applied. If the CSP constitutes of n identical servers of computational capacity C/n each, then this optimal intra-CSP dispatching and scheduling policy should achieve the same average utilization level ρ of all CSP servers. Under these assumptions, we can model the CSP as a single-server M/M/1 system computational capacity C with utilization ρ. While this is a simplification that allows the mathematical treatment of our paper, it is also reasonable enough to capture the reality.

B. Energy Consumption Cost

In order to develop a model for the power consumption of each CSP, we have again to take into account the multiple servers of the CSP. According to state-of-the art literature [11], the power consumption of a single server is linearly increasing in its utilization factor, $\rho = \frac{\lambda}{\mu}$. The power consumed is the sum of idle and dynamic power consumption. The idle power W_0 is the power consumed when the server is powered on and does not serve any request. The dynamic power consumption depends on the utilization ρ. If we denote by W_1 the power of the server when it is fully utilized (namely at $\rho = 1$), the range of dynamic energy consumption is $[0, W_1 - W_0]$. The total power consumption of the server as function of ρ is:

$$W(\rho) = W_0 + (W_1 - W_0)\ \rho. \tag{2}$$

Power consumption of a CSP. Now, we aim to show that we can use the same type of power consumption function (i.e. $\alpha + \beta\rho$) to model the total power consumption of a CSP. The power consumption of a CSP i is the aggregate power consumption of its servers. As we have already mentioned, we assume that the CSP achieves the same average level of utilization ρ in all its servers. Thus, the idle and dynamic power consumption of the CSP are the corresponding aggregates of power consumptions of all servers. If i has M_i servers, and if $W_{0,ij}$ and $W_{1,ij}$ denote the idle and total power consumption of the j-th server of i, the power consumption of the i is

$$\begin{aligned} W_i &= \sum_{j=1}^{M_i} W_{0,ij} + \frac{\lambda_i}{\mu_i} \sum_{j=1}^{M_i} \Big(W_{1,ij} - W_{0,ij} \Big) \\ &= W_{0,i} + (W_{1,i} - W_{0,i})\ \frac{\lambda_i}{\mu_i}, \end{aligned} \tag{3}$$

where $W_{0,i}$ and $W_{1,i}$ denote the idle and total power consumption of i's infrastructure. Therefore, the single-server model for the CSP also applies for the power consumption too. Furthermore, given a price q_i that CSP i should pay per Watt·sec, the cost of energy consumption per unit of time is:

$$E_i = q_i W_i. \tag{4}$$

C. QoS-dependent Pricing and CSP Profit

We reasonably assume that the CSP charges its clients based on the load served and on the QoS-level offered, for which we take the average execution delay per a task as a proxy. The pricing function $p_i(\cdot)$ of a CSP i should be decreasing in the average delay d_i per job experienced by its clients. Furthermore, it should also be convex, because a marginal change of delay is perceived more by the client for smaller values of the delay. A pricing function definition that satisfies the above conditions is

$$p_i(d_i) = x_i\ e^{-d_i/d^*}, \tag{5}$$

where x_i denotes the price per job that i charges for offering the service in the best possible level of QoS (ideally for $d_i \to 0$), while d^* is a parameter that specifies the sensitivity of the price to QoS degradation. Assuming that all the requests that arrive in the CSP's queue are always executed, the revenue rate in monetary units per unit of time for CSP i is

$$R_i = \lambda_i\ p_i(d_i). \tag{6}$$

In practice, the pricing function for each CSP is also driven by the competition in the market. In our approach, we assume that each CSP has made a decision a priori on its pricing function, taking also into account this competition. We assume that CSPs do not change their pricing functions and also that their clients are committed by some contract i.e. they do not have the freedom to change their serving CSP.

D. CSP Profit

Considering both the revenue (cash inflow rate) and the energy cost (cash outflow rate), the profit rate for CSP i is:

$$P_i = R_i - E_i. \tag{7}$$

III. CSP Federation policies

A. Model

Our federation model has at its core the transfer of a portion of the stream of requests from a CSP to other CSPs within the federation. In order to demonstrate our approach, we restrict our attention here to the case of two CSPs; the extension to more CSPs is trivial. For each CSP $i = 1, 2$, we define a variable α_i (with $0 \leq \alpha_i \leq 1$) that denotes the portion of the stream of requests (of rate λ_i) from clients of CSP i that are transferred to the other CSP. A federation policy is specified by a pair (α_1, α_2). We assume that the requests transferred from one CSP to the other experience an additional average delay D. This models the delay introduced by the transfer process by the intervening Internet links between servers of the two CSPs and various other causes of delay in between. In this work, we take D to be fixed and known by virtue of some measurement process that has taken place before the federation.

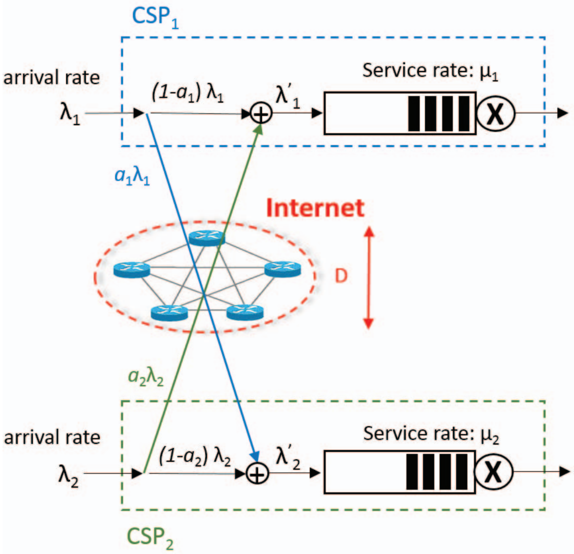

Fig. 1: Federation scenario for two Cloud Service Providers, each modeled through an M/M/1 queue. The job-request traffic that is transferred to the other CSP incurs a fixed delay D that models the various causes of intermediate delays it experiences.

We assume that the jobs in both CSPs have the same mean size in flop requirements L. Otherwise, due to the jobs from clients of a CSP that migrate to the other CSP, the queue of the destination CSP would have to support two classes of jobs and the M/M/1 model would need to be extended.

The average arrival rate of requests transferred from CSP$_1$ to CSP$_2$ is $\alpha_1 \lambda_1$ and are fed in the queue of CSP$_2$. Likewise, the average arrival rate of requests that are transferred from CSP$_2$ to CSP$_1$ is $\alpha_2 \lambda_2$ (Fig.1). Therefore, the total request arrival rate at the input of the queue of CSP$_1$ depends both on α_1 and α_2 and is given by $\lambda_1^{'}(\alpha_1, \alpha_2) = \big(1 - \alpha_1\big)\lambda_1 + \alpha_2 \lambda_2$. Similarly, at the input of the queue of CSP$_2$ we have $\lambda_2^{'}(\alpha_1, \alpha_2) = \big(1 - \alpha_2\big)\lambda_2 + \alpha_1 \lambda_1$. Consequently, the average delay d_i of requests that are served by the queue of CSP i, is:

$$d_i(\alpha_1, \alpha_2) = \frac{1}{\mu_i - \lambda_i^{'}(\alpha_1, \alpha_2)}. \tag{8}$$

Hence, part of the arriving requests from each CSP's clients is served by that CSP's own infrastructure, while another part is served by the infrastructure of the other CSP. Therefore, the average delay per job experienced by the clients of each CSP depends on the average delays at both CSPs' queues, $d_1(\alpha_1, \alpha_2)$ and $d_2(\alpha_1, \alpha_2)$. Thus, the average delay per job T_i experienced by clients of CSP i, for $i = 1, 2$ are given by:

$$\begin{aligned} T_1(\alpha_1, \alpha_2) &= \big(1 - \alpha_1\big)\ d_1(\alpha_1, \alpha_2) + \alpha_1\ \big(d_2(\alpha_1, \alpha_2) + D\big) \\ T_2(\alpha_1, \alpha_2) &= \big(1 - \alpha_2\big)\ d_2(\alpha_1, \alpha_2) + \alpha_2\ \big(d_1(\alpha_1, \alpha_2) + D\big). \end{aligned} \tag{9}$$

Note the subtle difference between $d_i(\cdot)$ and $T_i(\cdot)$: while $d_i(\cdot)$ denotes the average delay of *any job* served by the queue of CSP i regardless of whether it originated from clients of CSP$_1$ or CSP$_2$, $T_i(\cdot)$ denotes the average delay of jobs originated from clients of CSP i, regardless the server they are actually served.

We now revisit the definitions of revenue and energy cost. In the presence of federation, we re-define the power consumption $W_i(\alpha_1, \alpha_2)$, energy consumption cost $E_i(\alpha_1, \alpha_2)$, revenues $R_i(\alpha_1, \alpha_2)$ and profit functions $P_i(\alpha_1, \alpha_2)$ in order to be applicable in the federation. The power consumption is

$$W_i(\alpha_1, \alpha_2) = W_{0,i} + \Big(W_{1,i} - W_{0,i}\Big)\ \rho_i\ , \tag{10}$$

where $\rho_i = \frac{\lambda_i^{'}(\alpha_1, \alpha_2)}{\mu_i}$ since $\lambda_i^{'}(\alpha_1, \alpha_2)$ denotes the total incoming request rate at the queue of CSP i that causes the server utilization ρ_i. Thus, the energy cost per unit of time is

$$E_i(\alpha_1, \alpha_2) = q_i\ W_i(\alpha_1, \alpha_2). \tag{11}$$

Since the of jobs originated from clients of CSP i are served in both CSPs, the pricing of the clients of CSP i should be based on delay $T_i(\cdot)$ rather than on $d_i(\cdot)$. Therefore, the pricing function becomes $p_i(\alpha_1, \alpha_2) = x_i e^{-T_i(\alpha_1, \alpha_2)/d^*}$, and thus the revenue rate of CSP i in a federation is

$$R_i(\alpha_1, \alpha_2) = \lambda_i\ p_i(\alpha_1, \alpha_2). \tag{12}$$

Finally, the profit rate of CSP i is

$$P_i(\alpha_1, \alpha_2) = R_i(\alpha_1, \alpha_2) - E_i(\alpha_1, \alpha_2). \tag{13}$$

B. Cooperative Total Profit Maximization

We assume that the CSPs that participate in the federation are fully cooperative. That is, each CSP abides to cooperation rules that have been agreed a priori between the CSPs as to their participation in the federation and the rules for sharing the additional incurred profit of the federation (see below). Therefore, the federated CSPs jointly decide on the best possible request load transfer policy to each other, and each CSP always serves requests from the clients of the other CSP. The determination of the optimal federation strategy reduces to solving an optimization problem. The output of this problem is the optimal pair (α_1^*, α_2^*) of the portions of the job request traffic at the input of each CSP queue that are routed to the other CSP, such that the total profit of federated CSPs is maximized. The optimization problem is as follows:

$$\begin{aligned} \max_{\alpha_1,\alpha_2} \quad & P_1(\alpha_1,\alpha_2)+P_2(\alpha_1,\alpha_2) \\ \text{s.t.} \quad & 0 \le \alpha_i \le 1 \quad , \quad i=1,2 \\ & \lambda_i^{'}(\alpha_1,\alpha_2) < \mu_i \quad , \quad i=1,2 \end{aligned}$$

The second constraint is due to stability in the queues of each CSP, so that the rate of the stream of incoming requests does not exceed the service rate of the CSP. If that constraint were not included in the formulation, the delay would grow unbounded. This is a non-linear optimization problem that can be solved with standard methods, i.e. formation of the Lagrangian and statement of the necessary and sufficient KKT conditions that the pair (α_1, α_2) should satisfy for optimality.

We intuitively expect that the following properties apply for the optimal solution of the problem above: For a given $D > 0$, there exists a unique pair (α_1^*,α_2^*) that maximizes the objective above, and for this optimal pair it should be $\alpha_1^*\alpha_2^* = 0$. Therefore, we always have unilateral service delegation, i.e. at most one of the two CSPs transfers a portion of its request load to the other. This is because the optimal solution essentially entails an optimal load balancing , for which unilateral shift of load suffices. However, if $D = 0$, there exist in general multiple optimal solution pairs (a_1^*, a_2^*). In particular, the solution can be succinctly described as a pair $(z(\alpha_2^*), \alpha_2^*)$, where $z(\cdot)$ is an increasing function.

Profit sharing. Our problem formulation guarantees that the aggregate profit in the federation will be maximized, however this is not the case for the individual profit of each CSP, which may in fact deteriorate for one of them due to the formation of federation. Thus, we propose a policy that splits the profit among the CSPs in such a way that each of them has at least the same or higher profit compared to the standalone operation. In particular, the solution of the optimization problem leads to a total profit $P_{tot} = P_1(a_1^*, a_2^*) + P_2(a_1^*, a_2^*)$ that may exceed or be equal to the corresponding total profit of CSPs if these were not involved in a federation. The latter is attained for $(a_1, a_2) = (0, 0)$ and thus can be written as $P_{tot}(0, 0) = P_1(0, 0) + P_2(0, 0)$.

If the federation is beneficial for the CSPs as a whole, then the issue arises, how to share the profit incurred by the federation. By incurred profit we mean the difference $P_{tot}(a_1^*, a_2^*) - P_{tot}(0, 0)$. Recall that under the optimal federation policy, load is delegated only to one CSP by the other. This extra workload increases the energy consumption cost the CSP to whom load is delegated, due to the higher utilization and thus higher power consumption of its infrastructure. Therefore, this CSP has reduced profits and may be unwilling to conform to the federation, unless some rule is applied for compensating it for these losses. Since the total profits of the federation exceed those of the standalone case, the CSP that delegates part of its workload definitely has higher profit than before. This CSP should compensate the other for loss in profit. Thus, the CSPs should reach an agreement for the sharing of the additional profit that satisfies both of them. A cooperative sharing policy that serves the above objective is one where each CSP gets at least the profit it had in the no-federation case, while the extra profit generated from federation is shared according to some proportionality rule. If this rule concerns the served request load, then CSP i gets profit. Thus, the payoff that CSP i eventually obtains is given by

$$\frac{\lambda'_i(\alpha_1^*,\alpha_2^*)}{\lambda_1+\lambda_2}\left(P_{tot}(a_1^*,a_2^*)-P_{tot}(0,0)\right)+P_i(0,0) \ , \qquad (14)$$

where the second term represents the profit of CSP i in the standalone operation, while the first term is the share of the extra profit induced by the federation that is given to CSP i.

C. Problem Generalization

In the general case of N CSPs, where $N > 2$, the federation policy is defined as a $N \times N$ matrix $\mathbf{A}$, whose entries α_{ij} determine the percentage of job requests of traffic of CSP i that is sent to CSP j. Consequently, the objective of our optimization problem in the general case is to derive the optimal matrix $\mathbf{A}$ that maximizes the total profit of the CSPs while maintaining stability in the queue of each CSP. Thus, extending our notation, we obtain the following problem:

$$\begin{aligned} \max_{\mathbf{A}} \quad & \sum_{i=1}^{N} P_i(\mathbf{A}) \\ \text{s.t.} \quad & 0 \le \alpha_{ij} \le 1 \quad , \quad i,j = 1,..,N \\ & \lambda_i^{'}(\mathbf{A}) < \mu_i \quad , \quad i=1,..,N \\ & \sum_{j=1}^{N} \alpha_{ij} = 1 \quad , \quad i=1,..,N \end{aligned}$$

,where $P_i(\mathbf{A})$ denotes the dependence of profit of CSP i on matrix $\mathbf{A}$.

IV. Numerical results

We simulate an environment of two CSPs both in standalone and federated operation. Recall that by standalone we mean that CSPs act in isolation from each other and serve only their own clients. We assume that the average number of processor flops L that a job requires in order to be completed is the same for both CSPs and equals to 2. Regarding the network delay D that models the average intermediate delay experienced by a request that is transferred from one CSP to the other, we assume that it is a small fraction of d_1, d_2. Thus, we set D to be an order of magnitude lower than d_1 and d_2.

A. Symmetric CSPs

In the first set of experiments we assume that CSP$_1$ and CSP$_2$ are symmetric with respect to their infrastructure $C_1 = C_2 = 20$ flops/sec. For the power consumption of the servers we take $W_0 = 300$ MWatt and $W_1 = 1000$ MWatt. We also assume that both CSPs pay the same price, namely $q = 15$ \$/MWatt·hour to their electricity provider, while they charge their clients according to the same pricing function, with the same maximum price x \$/job when $d_i \rightarrow 0$ and sensitivity parameter $d^* = 1$ sec. In our experiments, we select the value of x taking as input the electricity price q. In particular, given the price q we find the value of x for which the profit of CSP becomes zero when the utilization factor approaches 1. This guarantees that both CSPs in standalone operation will not have negative profit under any value of utilization ρ. Next, we assume that the CSP$_2$ has fixed rate of incoming requests λ_2 and we set values for λ_1 in the range of values for which the queues of both CSPs are stable (there should apply $\frac{C_i}{L} > \lambda_i$), from 1 to 9.9 with a step of 0.1. We run this type of experiment for different fixed values of λ_2 from 1 to 9.9.

The results in Fig. 2 show that for $\lambda_2 = 9$ and for low to medium load λ_1, federation leads to 50-100% more total profit compared to the case when each CSP serves its own clients. The benefits of the federation are seen to diminish as

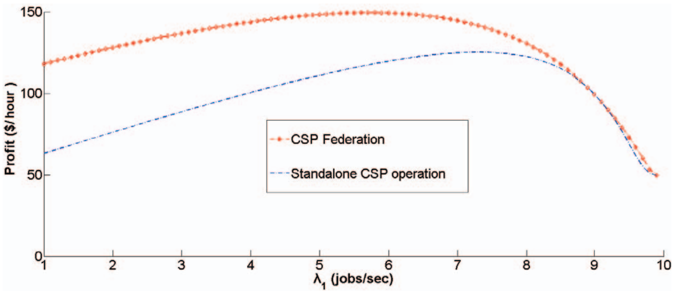

Fig. 2: Maximum total profit of federation for $\lambda_2 = 9$ and $\lambda_1 \in [1, 9.9]$.

λ_1 tends to λ_2. In the case where both CSP input loads are equal ($\lambda_2 = \lambda_1 = 9$) the profit of the federation vanishes. These results provide valid guidelines for when a federation is most profitable. That is, the more diverse the input loads are, the more pronounced the benefit of the federation is.This should have been intuitively expected, because for two CSPs with the same computational capacity the optimal federation tends to balance their loads.

In Fig. 3 we provide results for the average delay. For the average delay of clients of federated CSPs, it is shown that the optimal federation policy achieves an average delay that coincides or is close to the optimal average delay with respect to a QoS-based federation policy, namely the optimal policy when the objective function of the optimization problem includes only the delay.

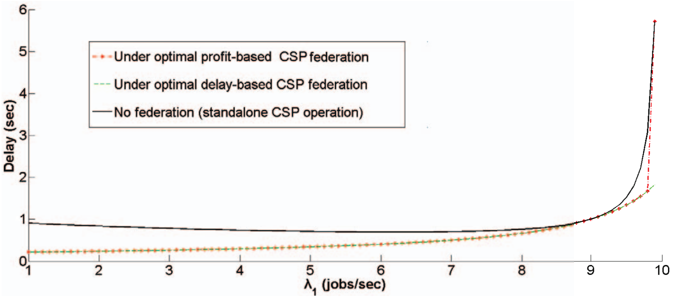

Fig. 3: Average delay of all clients in the environment of two CSPs under different policies.

In Fig.4, we can observe that in the solution of the optimization problem that gives the optimal federation policy, at least one of α_1 and α_2 equals to zero, while the non-zero value always refers to the most utilized CSP. Our results confirm the *unilateral service property* we discuss in section III-B.

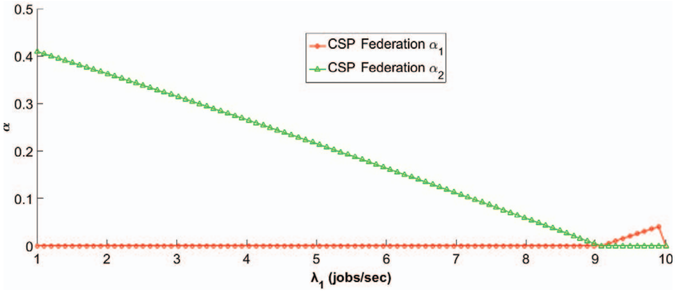

Fig. 4: Optimal pairs (α_1, α_2) that denote the portions of request traffic transferred from one CSP to the other, for $\lambda_2 = 9$ and $\lambda_1 \in [1, 9.9]$.

Moreover, the results in Fig. 5 show that as the intermediate delay D increases, the CSPs follow a more conservative job transfer strategy, and when D exceeds a certain high value, both α_1^* and α_2^* becomes zero. Consequently, as D increases, the effectiveness of federation decreases, and thus the maximum total profit decreases and after a certain value the total profit is maximized when the CSPs do not federate.

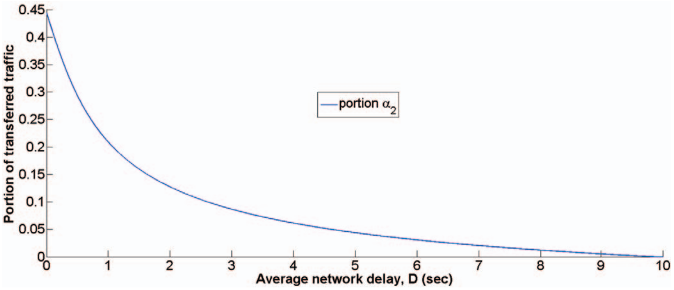

Fig. 5: Optimal pairs (α_1, α_2) that denote the portions of request traffic transferred from one CSP to the other as a function of D, for $\lambda_1 = 1$ and $\lambda_2 = 9$. Note that $\alpha_2 = 0$.

B. Asymmetric CSPs

Asymmetric infrastructure - symmetric pricing. We run the same type of experiments for asymmetric CSPs with respect to their infrastructure, i.e. $C_1 \neq C_2$. Since we assume that the power consumption of servers is related to their processing power, the CSPs are also asymmetric with respect to their power consumption. Therefore, we consider three different values of CSP dimensioning and the related power consumption, namely $C = \{10, 20, 40\}$ and $(W_0, W_1) = \{(300, 1000), (600, 2000), (1200, 4000)\}$. Then, we try all possible combinations of elements in the sets above for CSP$_1$ and CSP$_2$. The results reveal that the parameter that affects more the effectiveness of federation is again the utilization factor of the infrastructure of each CSP. Also, when the largest CSP has a high utilization factor and the other has a low to medium utilization factor, the federation can achieve higher benefit than in the opposite case of asymmetric CSPs. However, for both cases of asymmetric CSPs, forming a federation is in general more beneficial than for symmetric CSPs.

Asymmetric infrastructure and pricing. We also run a set of experiments for symmetric infrastructure, but now we assume that the energy prices q_i and the maximum prices per job x_i of the two CSPs are asymmetric. In particular we consider three different values of electricity price $q_i \in \{5, 10, 15\}$ and we assign all possible pairs of them to two identical CSPs together with the x_i's produced from them. In the case where utilization factors of the CSPs differ significantly, it is shown that when the highly utilized CSP is the one with the highest value of x_i, the benefit of federation is higher and the portion of requests that are outsourced increase up to 25% compared to the case of symmetric pricing. On the other hand, when the CSP with the lowest utilization has the highest x_i, the benefits and the portion of outsourced requests decrease to 25%. The effect of price asymmetry is less pronounced when the CSPs have the same utilization level.

V. Related Work

Cloud federation concept. The authors in [6] propose the reservoir model, a modular cloud architecture that allows multiple CSPs to pool their resources in order to provide services as a federation of CSPs. In [9] the architectural elements of a service-oriented, market-based architecture of cloud federation among IaaS CSPs is presented. The basic entities of the architecture are a cloud broker (buyer) per client, a cloud coordinator (seller) per CSP and a cloud exchange as the market maker. Finally, in [7] a definition of the cloud federation as a concept of service aggregation is suggested, focusing on two types of federation, the redundancy and the migration federations. The former is used when more than one CSPs offering a service together are able to achieve better utility than any single CSP, while the migration federation is triggered when a CSP offering a new service achieves better utility for the client than any previously used service of another cloud provider (or federation), and thus the job should migrate from the old service to the new one.

Resource allocation in distributed or federated clouds. A class of published works deal with the resource allocation in geographically distributed CSPs or federated environments of CSPs. In [12] the authors design algorithms for inter-cloud resource trading and scheduling in a federation of geo-distributed clouds, by applying a double-auction based mechanism. In [13] and [14] cloud federation formation is modeled as a coalitional game where CSPs dynamically decide to form a cloud federation and to allocate Virtual Machines (VMs) based on their clients' requests. Additionally, in [14] an energy-aware mechanism coping with hosts and their energy consumption is proposed.

In [15] the authors investigate the distributed VM resource allocation problem in dynamic cloud federation platforms, where IaaS CSPs are at the same time buyers and sellers of computational resources. A Stackelberg game is presented in [10]; the game is between the Application Service Providers (followers) that aim at optimizing their offered QoS and the CSPs (leaders) that set prices of resources to maximize their own benefit. In [16], the cloud federation is presented as a solution to the problem of resource provisioning in the presence of large workload variations. A global scheduler is proposed with the goal to maximize the CSPs' utility by deciding between VM migration or VM shutdown. Finally, the authors of [17] consider an environment of multiple CSPs where each CSP maintains a number of heterogeneous servers and model each server as a queueing system. Then, they formulate the problem of resource provisioning and management among different CSPs as a game among rational CSPs that aim to maximize their own profit taking into account the SLA agreements.

Some of the works above provide an overview of the architectural elements of a system that will enable the federated operation of multiple CSPs. In our work on the other hand, we develop an economic model of the federated environment of CSPs and we investigate profit-based optimal federation formation policies. Further, most of existing works on resource allocation do not take into account the QoS offered to CSPs' clients in their optimization approach. In our work, the federation policy is optimal with respect to total CSPs' profit, but it is also beneficial and caters for client utility, since a better QoS for clients leads to higher revenue for the CSP.

VI. Conclusions

In this paper, we provide the economic modeling and the policies for the formation of profitable service-oriented cloud federations. The results show that the formation of the optimal federation increases the profit of the CSPs that join the federated environment and it achieves a QoS that approaches the optimal one. The key factor that federation takes advantage of is the utilization factor of CSPs, but federation can achieve further benefits by taking advantage of asymmetries in infrastructure such as the computational capacity C_i or in pricing, i.e. different prices charged. In this work we adhered to a cooperative formulation in an effort to quantify the benefits of the system of federated CSPs that emerge from the solution of the optimization problem. The cooperative case is actually in the core of federation, and that is why we address it first. Our main objective was to expose the model, hence we started with the case of two CSPs, which is also likely to occur in reality. The transfer of our theory to the case of more than two CSPs requires certain extensions, particularly in designing modes of profit sharing; we intend to address this case in our future work. Finally, given that each CSP is a selfish entity, we have also considered the profits and incentives of each individual CSP. However, several other interesting game-theoretic aspects arise, which we plan to address in the future.

VII. Acknowledgments

The work of G.D.Stamoulis and G. Darzanos was partly funded by the EU ICT Project SmartenIT (FP7-2012-ICT-317846). Also, G. Darzanos and I. Koutsopoulos acknowledge the support of ERC08- RECITAL project, co-financed by Greece and the European Social Fund through the Operational Program Education and Lifelong Learning- NCRF 2007-2013.

References

[1] https://www.egi.eu/infrastructure/cloud.

[2] http://onapp.com/federation.

[3] http://www.arjuna.com/federation.

[4] http://openlab.web.cern.ch.

[5] http://www.bonfire-project.eu.

[6] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Muoz, and G. Tofetti, "Reservoir - when one cloud is not enough," *Computer*, vol. 44, no. 3, pp. 44–51, March 2011.

[7] T. Kurze, M. Klems, D. Bermbach, A. Lenk, S. Tai, and M. Kunze, "Cloud federation," in *Proc. of Cloud Computing*, 2011.

[8] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *Proc. of IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010.

[9] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proc. of the 10th International Conference on Algorithms and Architectures for Parallel Processing - Volume Part I*, 2010.

[10] H. Roh, C. Jung, W. Lee, and D.-Z. Du, "Resource pricing game in geo-distributed clouds," in *Proc. of IEEE INFOCOM*, 2013.

[11] M. Steinder, I. Whalley, J. Hanson, and J. Kephart, "Coordinated management of power usage and runtime performance," in *Proc. Network Operations and Management Symposium (NOMS 08)*, 2008.

[12] H. Li, C. Wu, Z. Li, and F. Lau, "Profit-maximizing virtual machine trading in a federation of selfish clouds," in *Proc. of IEEE INFOCOM*, 2013.

[13] L. Mashayekhy, M. Nejad, and D. Grosu, "Cloud federations in the sky: Formation game and mechanism," *IEEE Trans. on Cloud Computing*, vol. 3, no. 1, pp. 14–27, Jan 2015.

[14] M. Guazzone, C. Anglano, R. Aringhieri, and M. Sereno, "Distributed coalition formation in energy-aware cloud federations: A game-theoretic approach (extended version)," *CoRR*, vol. abs/1309.2444, 2013. [Online]. Available: http://arxiv.org/abs/1309.2444.

[15] M. Hassan, B. Song, and E.-N. Huh, "Distributed resource allocation games in horizontal dynamic cloud federation platform," in *Proc. of IEEE 13th International Conference on High Performance Computing and Communications (HPCC)*, 2011.

[16] I. Goiri, J. Guitart, and J. Torres, "Characterizing cloud federation for enhancing providers' profit," in *Proc. of IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010.

[17] Y. Wang, X. Lin, and M. Pedram, "A game theoretic framework of sla-based resource allocation for competitive cloud service providers," in *Proc. of Sixth Annual IEEE Green Technologies Conference*, 2014.