



HAL
open science

Incentivising Resource Sharing in Federated Clouds

Eduardo De Lucena Falcão, Francisco Brasileiro, Andrey Brito, José Luis Vivas

► **To cite this version:**

Eduardo De Lucena Falcão, Francisco Brasileiro, Andrey Brito, José Luis Vivas. Incentivising Resource Sharing in Federated Clouds. 15th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS), Jun 2015, Grenoble, France. pp.45-50, 10.1007/978-3-319-19129-4_4 . hal-01775034

HAL Id: hal-01775034

<https://inria.hal.science/hal-01775034>

Submitted on 24 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Incentivising Resource Sharing in Federated Clouds

Eduardo de Lucena Falcão, Francisco Brasileiro, Andrey Brito, and José Luis Vivas

Federal University of Campina Grande, Department of Computing and Systems
Campina Grande, Brazil
eduardofalcao@lisd.ufcg.edu.br, {fubica, andrey}@computacao.ufcg.edu.br,
jlvivas@lisd.ufcg.edu.br

Abstract. An important cloud federation enabler is the development of suitable business models, which has so far received relatively little attention from investigators. Few efforts have been dedicated to investigation of business models for enabling cloud federations, especially with regard to distribution issues, in which centralized solutions are usually preferred. Hence, in our work we focus on the use of fully decentralized mechanisms supporting federation of private clouds based on barter mechanisms. We analyze the adaptation of fully decentralized incentive mechanisms previously used in the context of P2P desktop grids, and show that they are not suitable for federated cloud systems because fairness cannot always be guaranteed. We show initial results concerning the use of a mechanism intended to guarantee a higher level of fairness and thereby to promote voluntary participation in a decentralized federation without any central or trusted enforcing authority.

Keywords: cloud and grid computing, cooperation incentives and fairness, peer to peer computing

1 Introduction

Organizations with variable and peaky demand patterns often turn to public clouds (cf. “cloud bursting” [4]) in order to meet unexpected or short-term needs. However, during off-peak times resources might become idle, which constitutes an efficiency loss for the organization. An alternative for cloud providers is to participate in a federation for exchanging idle resources. In particular private cloud providers, due to the usually limited amount of owned resources, would greatly benefit from this [2].

From the architectural perspective, cloud federations can be either centralized or Peer-to-Peer (P2P) [3]. In centralized architectures, resource allocation is typically performed by a trusted central entity that is able to prevent free riding and to perform the best matching of consumers and providers. In P2P federations, on the other hand, participants must communicate and negotiate directly with each other. The advantages of decentralized topologies include extensibility, deployment, management, use and growth. The drawbacks, however,

include difficulties in discovery, routing, security, and the fact that participants are mostly unknown to each other and cannot be assumed to be trustworthy or collaborative. Moreover, peers should be assumed to be selfish and to have an economic incentive to become free riders.

Similarly, from a market perspective cloud federations can be also classified as centralized or decentralized. In a centralized market system, bids and requests are collected by a central entity or market auctioneer that matches them and decide the best matching of buyers and sellers. In decentralized markets, on the other hand, buyers and sellers must explore the market by themselves and bargain directly with each other. Participants may provide resources in exchange for payment or by bartering. Payment schemes would require the introduction of complex management mechanisms and procedures, whereas bartering schemes can be implemented e.g. in the shape of flexible credit and debit local annotations. The latter would be very suitable for the federation of private clouds, since it is a wholly money-less and distributed scheme that can do without any trusted centralized entities. Obviously, there is a close relation between the architecture and the market structures of private cloud federations. In this work, we propose a lightweight P2P cloud federation infrastructure implementing a decentralized market system.

An important challenge concerns the promotion of cooperation among rational selfish individuals in a decentralized context with no central and trusted enforcing authority. In this kind of system, participants are usually left to themselves and only with limited information about the actual behavior or intention of other actors, may not keep promises, and must rely solely on their own experience, acquired through interactions with each other, in order to decide to what extent they should trust other partners. It is therefore natural to expect that, at first hand, participants will prefer to act as free riders. Moreover, collaborative partners may defect from the federation if they are not satisfied with the results of participation. Some form of individual incentives must therefore be enforced in order to ensure sustained voluntary participation. Our aim is to propose one such mechanism enabling the collaborative peer to make efficient decisions that guarantees both its satisfaction, defined as the ratio between received and requested resources (which should ideally be one), and fairness, defined as the ratio between the amount of resources obtained and the amount of resources provided (which should ideally be approximately one on the long run). The challenge is to find a scheme that guarantees that the levels of both fairness and satisfaction are good enough to ensure that most participants will not defect and free riders will be isolated or kept with a low degree of satisfaction.

For the reasons explained above, incentives or punishment procedures must exist in order to promote cooperation and keep the federation alive. The scheme we propose in this paper leverages the notion of the Network of Favors (NoF) [1], an incentive mechanism for resource sharing in P2P opportunistic desktop grids. In NoF, each peer uses only its own locally stored interaction history with other peers, which is basically the balance of favors exchanged (total amount of favors a peer A consumed from peer B minus total amount of favors a peer A do-

nated to peer B), in order to decide which ones will be given priority in resource requests. We call this notion of NoF the *Satisfaction-Driven* NoF (SD-NoF). In the SD-NoF a collaborator always supply all of its idle resources to the federation, with the expectation of accumulating credits with other peers, which may be converted to favors in the future. This ensures the best possible levels of satisfaction to collaborators, independently of the level of resource contention. This scheme is suitable for opportunistic desktop grids, since the costs of providing resources, which are intended primarily for in-house consumption, are assumed to be so low that collaborators may disregard fairness and focus only on maximizing satisfaction. However, in the case of private and P2P federated clouds, resources are dedicated, and the associated overhead costs (management staff, energy and space) may not be negligible. Fairness may thus become an important goal, which may be achieved by limiting the amount of supplied resources, thereby isolating free riders more efficiently specially in scenarios with low resource contention. To this end, we introduce a feedback control loop mechanism regulating the amount of supplied resources. Briefly, in order to reward cooperative actors and isolate free riders, a peer in the federation will draw upon its current assessment of fairness in order to define the amount of offered resources, even if it would have at any given moment more idle resources than those offered. By contrast to the SD-NoF, we call this new scheme *Fairness-Driven* NoF (FD-NoF).

The outcome is a mechanism that more closely conforms to the game theoretical results derived from the notion of reciprocal altruism [5], which shows how selection can operate against cheaters or non-reciprocators (the free riders in the NoF). Altruistic behavior is defined as a behavior that benefits another unrelated individual at a certain cost for the contributing individual. Selection would seem to favor “cheaters”, i.e. those individuals that fail to reciprocate favors. However, selection might also discriminate against cheaters if the altruist is able to curtail future altruistic gestures to those individuals, assuming that the sum of the benefits for the altruist of those lost acts outweighs their cost. As a result, the altruist will prefer to exchange altruistic acts with other altruist individuals, not with cheaters.

2 Feedback Control Loop Mechanism

Andrade *et al.* [1] proved that whenever there is resource contention¹ between collaborators (κ), the SD-NoF works well and prioritizes collaborators before free riders. However, only prioritizing collaborators is not enough in scenarios with low resource contention ($\kappa < 1$), since in this case all the surplus resources will nevertheless be offered, thus benefiting free riders. To avoid this, collaborators could try to regulate the amount of resources provided, thereby indirectly increasing the value of κ . Collaborators are not able to directly control the value of κ in a decentralized P2P system, and may not even have an interest in doing

¹ Contention is a metric that characterizes the degree of competition for offered resources, defined as the ratio of requested resources by provided resources.

it. However, contention may be indirectly increased when collaborators try to balance their level of fairness. The simplest way to do it is by decreasing the amount of provided resources. If fairness is close to 1 for each collaborator, the value of contention will also be close to 1. However, the overall level of satisfaction may decrease thereby, and affect not only free riders but collaborators as well.

Each collaborator is then equipped with the feedback control loop mechanism and must define on its own a threshold value τ for the desired minimum level of fairness. The mechanism enhances fairness by controlling the amount of provided resources until its value becomes higher than τ ; once the desired level of fairness has been achieved, the algorithm will instead focus on enhancing satisfaction by increasing the amount of resources provided. Roughly, the feedback control loop mechanism decides whether to decrease or increase the amount of resources provided by a fixed value at each step of the simulation, in this work chosen to be 5% of the total resources capacity, in order to achieve the desired levels of both fairness and satisfaction. By this mechanism, a collaborator could momentarily “leave” the federation, by supplying no resources, until it receives again some favors.

2.1 Simulation Model

To assess the behavior of the participants in the FD-NoF, we have built a simulator for a simplified model of a resource sharing P2P federation of cloud providers. The federation consists of a community of n peers, with $(1 - f) \cdot n$ collaborators and $f \cdot n$ free riders, $0 \leq f \leq 1$. We are interested in understanding how the set of collaborators perform in relation to the set of free riders. Thus, to eliminate the influence that different kinds of participants might have in the results, we assume that all collaborators have the same capacity and needs, and free riders the same needs. The simulation proceeds in steps. At each step each collaborator can be in a consumer state with probability of π or in a provider state with probability $1 - \pi$. Each collaborator is assumed to have a total resource capacity of C , and the amount of demand requested by any consuming collaborator is $D \cdot C$, where $D \geq 0$. In the SD-NoF, when in a provider state a collaborator will always offer all of its resources, i.e. C . At each step, the resources are given to one or more peers selected from the set of consuming peers according to the balance of favors exchanged (more details of NoF in [1]). Free riders, in contrast, never provide resources and are always in a consumer state, each one requesting $D \cdot C$ resources at every step.

2.2 Scenarios

Our goal is to understand the behavior of the peers in scenarios with low, moderate and high levels of κ , with values ranging over the set $\{0.5, 1.0, 2.0, 4.0\}$.

We are interested in the scenarios in which the amount of free riders may seriously affect the level of fairness for collaborators. The higher the percentage of free riders in the system, the more resources, which are provided solely by

collaborators, they will collectively consume in total, thereby affecting negatively the level of fairness for collaborators. Thus, we chose $f = 0.75$ for all scenarios.

The larger the number of participants in the federation, the worse will be the situation to the free riders, even with constant f . This is because the larger the total amount of collaborators, the less will be at each step the deviation from the expected value of the total amount of providers, and hence the probability that a significant amount of resources will be available to free riders, which would imply that the number of providers is a bit higher than average. Then, we fix n to 100 in all scenarios.

Finally, for simplicity we consider $C = 1$. In summary, our design of experiments will have three parameters with constant values, $n = 100$, $f = 0.75$ and $C = 1$, and two variables with changing values: D and π — which both generates the different levels of κ . We run the simulations in 4000 steps, sufficient for ensuring that the fairness of collaborators achieves stability.

3 Results and Analysis

In order to evaluate the Feedback Control Loop we simulated the scenarios presented in section 2.2 in both SD-NoF and FD-NoF. Figure 1 shows the average fairness of collaborators (lines) and the individual fairness of each one of them (circles) along the four κ values, at the last step of the simulation in both SD-NoF and FD-NoF, the latter with $\tau \in \{0.8, 0.95\}$.

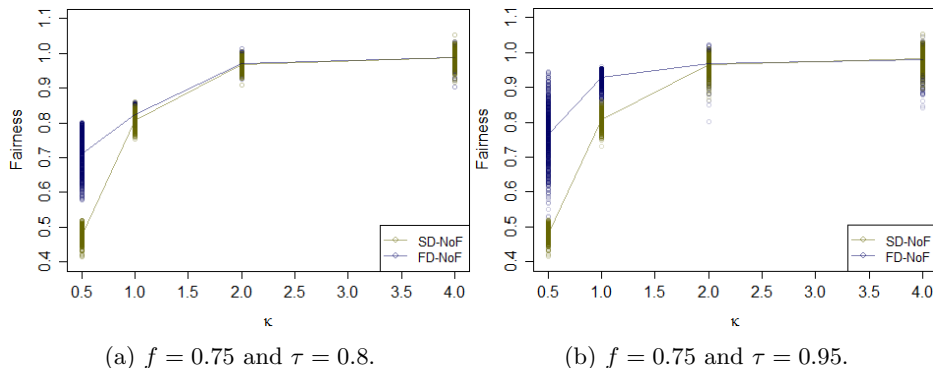


Fig. 1: Average and density of fairness to collaborators along $\kappa \in \{0.5, 1, 2, 4\}$ with SD-NoF and FD-NoF.

From Figure 1 we may observe that the Feedback Control Loop Mechanism was able to increase the values of fairness in low and moderate κ scenarios. Because f is high (75%), although free rider's individual satisfaction tends to be low (due to the high resource contention between themselves), their collective action strongly affects the fairness of collaborators in SD-NoF, which is on average

0.48 and 0.81 for contention values of 0.5 and 1, respectively (see yellow line on Figures 1a and 1b). In this case, FD-NoF increased the fairness of collaborators on average by 60% and 15%, respectively, when τ was set to 0.95, and by 48% and 3%, respectively, when τ was set to 0.8.

Moreover, in scenarios with high values of κ , FD-NoF behaves similarly to SD-NoF, regardless the high f value and τ , which is good since SD-NoF already works fine in these scenarios. One can also notice that the higher is the value of τ , the tougher is the task of the controller. Thus, not surprisingly, when we used $\tau = 0.95$, only few collaborators could indeed approach it when $\kappa = 0.5$. Obviously, as κ increases, the Feedback Control Loop Mechanism gets more efficient. Also, smaller values of τ lead to less variability among collaborators.

In summary, FD-NoF achieves its goal: collaborators improve their level of fairness by controlling the amount of supplied resources.

4 Conclusions and Future Works

In our work we introduce the FD-NoF, an enhancement of the NoF [1], by defining a Feedback Control Loop mechanism to regulate the amount of resources provided. With the aid of this mechanism collaborators may achieve greater levels of fairness by indirectly increasing the levels of resource contention. FD-NoF behaves similarly to SD-NoF in high resource contention scenarios, which is a positive result since SD-NoF by itself already provides good values of fairness.

As future work, we will implement the FD-NoF as part of the middleware², and deploy and evaluate the performance of the FD-NoF on the federated cloud that comprises the computing infrastructure of the EUBrazilCC project³. Moreover, we will refine the FD-NoF in order to accelerate convergence by introducing delegation schemes enabling the donation of services on behalf of a third agent.

References

1. N. Andrade, F. Brasileiro, W. Cirne, and M. Mowbray. Discouraging free riding in a peer-to-peer cpu-sharing grid. In *High performance Distributed Computing, 2004. Proceedings. 13th IEEE International Symposium on*, pages 129–137, June 2004.
2. Eduardo R. Gomes, Quoc Bao Vo, and Ryszard Kowalczyk. Pure exchange markets for resource sharing in federated clouds. *Concurrency and Computation: Practice and Experience*, 24(9):977–991, 2012.
3. Nikolay Grozev and Rajkumar Buyya. Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, 44(3):369–390, 2014.
4. P Marshall, K Keahey, and T. Freeman. Elastic site: Using cloudsto elastically extend site resources. In *Conference on Cluster, Cloud, and Grid Computing (CC-GRID)*, pages 43–52, May 2010.
5. Robert L. Trivers. The evolution of reciprocal altruism. *The Quarterly Review of Biology*, 46(1):35 – 57, 1971.

² <http://fogbowcloud.org/>

³ <http://eubrazilcloudconnect.eu/>