

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/281452196>

# Controlando a Contenção de Recursos para Promover Justiça em uma Federação Peer-to-Peer de Nuvens Privadas

Conference Paper · May 2015

DOI: 10.13140/RG.2.1.2462.2489

READS

32

4 authors:



[Eduardo de Lucena Falcão](#)

Universidade Federal de Campina Grande (...)

22 PUBLICATIONS 7 CITATIONS

[SEE PROFILE](#)



[Francisco Brasileiro](#)

Universidade Federal de Campina Grande (...)

201 PUBLICATIONS 1,991 CITATIONS

[SEE PROFILE](#)



[Andrey Brito](#)

Universidade Federal de Campina Grande (...)

34 PUBLICATIONS 151 CITATIONS

[SEE PROFILE](#)



[José-Luis Vivas](#)

Universidade Federal de Campina Grande (...)

15 PUBLICATIONS 94 CITATIONS

[SEE PROFILE](#)

# Controlando a Contenção de Recursos para Promover Justiça em uma Federação Peer-to-Peer de Nuvens Privadas

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

<sup>1</sup>Universidade Federal de Campina Grande  
Departamento de Sistemas e Computação  
Campina Grande, Brasil

eduardolfalcao@lsd.ufcg.edu.br, {fubica, andrey}@computacao.ufcg.edu.br,  
jlvivas@lsd.ufcg.edu.br

**Abstract.** *Private cloud providers would greatly benefit from combining their efforts in a cloud federation. This would allow a better conciliation of peak and off-peak demands occurring in same times at different providers. In this work, we propose the use of a decentralized and money-less market for resource sharing in a P2P federation of private cloud providers. The proposed scheme leverages the notion of the Network of Favors (NoF) proposed earlier as an incentive mechanism in the context of P2P opportunistic desktop grids. An extension of the NoF is put forward consisting of the introduction of a feedback control loop mechanism regulating the amount of resources that each cloud provider should offer to the federation with the aim at achieving suitable levels of fairness.*

**Resumo.** *Provedores privados de computação na nuvem poderiam obter considerável benefício mútuo ao operar suas infraestruturas de forma federada. Tal operação permite que a demanda excedente de um provedor possa ser atendida por outros provedores que estejam experimentando uma baixa demanda naquele mesmo instante. Neste trabalho, propomos a utilização de um mercado descentralizado baseado em escambo para o compartilhamento de recursos em uma federação P2P de provedores privados de computação na nuvem. Utilizamos a noção da Rede de Favores e propomos a introdução de um laço de controle retroalimentado que regula a quantidade de recursos que cada participante deveria oferecer para a federação para assegurar níveis adequados de justiça.*

## 1. Introdução

Organizações cujo grau de demanda apresenta alta variabilidade geralmente recorrem às nuvens públicas (cf. “*cloud bursting*” [Marshall et al. 2010]) a fim de atender necessidades inesperadas ou de curto prazo. Com uma ágil alocação de recursos, tais organizações conseguem evitar possíveis falhas no cumprimento de acordos de nível de serviço possibilitando a oferta da qualidade de serviço acordada, mesmo em momentos de pico. Por outro lado, recursos próprios podem se tornar ociosos durante períodos de baixa demanda e embora nesse caso os custos operacionais possam ser reduzidos, bastando simplesmente desligá-los, os custos de amortização decorrentes da compra e manutenção dos recursos de TI ainda constituem uma perda de eficiência para a organização.

Nestas circunstâncias, o *cloud bursting* por si só não é suficiente para atingir eficiência máxima de custo. Uma alternativa que os provedores de nuvem, especialmente os privados, podem considerar vantajosa é a colaboração de forma voluntária em um ambiente federado através do escambo de recursos ociosos [Gomes et al. 2012]. Neste caso, assume-se que cada membro da federação atua tanto como provedor quanto como consumidor de recursos durante momentos de vale e de pico, respectivamente.

No que concerne à arquitetura, federações de nuvens privadas podem ser categorizadas como centralizadas ou Peer-to-Peer (P2P) [Grozev and Buyya 2014]. Em arquiteturas centralizadas, a alocação de recursos é normalmente realizada ou facilitada por uma entidade central confiável cujas responsabilidades podem incluir o registro de recursos disponíveis além das funções convencionais de um mercado. Em contrapartida, participantes de federações P2P precisam se comunicar e negociar diretamente entre si, sem mediadores. As vantagens de topologias descentralizadas incluem a extensibilidade, implantação, gerenciamento, uso e escalabilidade. Obviamente, elas são independentes de qualquer infraestrutura centralizada e autoridade confiável, o que facilita a adesão de novos membros. As desvantagens, por outro lado, incluem dificuldades na descoberta, roteamento, segurança, e o fato de que os participantes são em sua maioria desconhecidos uns dos outros e não podem ser considerados confiáveis ou colaborativos.

Da mesma forma, sob uma ótica de mercado, federações de nuvens também podem ser categorizadas como centralizadas ou descentralizadas. Evidentemente, há uma estreita relação entre as arquiteturas e as estruturas de mercado de tais federações. Geralmente, não faz muito sentido implementar um sistema de mercado centralizado em uma arquitetura P2P descentralizada. Neste trabalho, propomos um sistema de mercado descentralizado para uma infraestrutura leve de federação P2P de provedores de nuvens privadas. Esta abordagem é adequada para federações com um grande número de nuvens participantes, o que aumenta as chances de que a escassez e excesso de recursos ocorram simultaneamente com mais frequência nos diferentes membros da federação.

No contexto do mercado descentralizado o principal desafio é a promoção de cooperação entre indivíduos egoístas racionais em um cenário que não dispõe de uma autoridade central e confiável. Neste tipo de mercado, os participantes possuem apenas uma quantidade limitada de informação sobre o sistema, adquirindo noção do comportamento real de outros atores através de sua própria experiência e baseando-se em interações passadas para decidir até que ponto eles devem confiar em outros participantes. Por isso, é natural esperar que, a priori, os participantes prefiram agir como caronas – indivíduos que apenas consomem recursos da rede – uma vez que este é o comportamento que inicialmente seria mais vantajoso para eles. Além disso, os participantes colaborativos podem sair da federação se estiverem insatisfeitos com os resultados de suas interações. Portanto, alguma estratégia que maximize os benefícios de participar da federação deve ser considerada, a fim de mantê-la proveitosa para os participantes. Tal estratégia deve auxiliar um participante colaborativo a tomar decisões eficientes – quanto devo doar? e para quem devo doar? – que garantam tanto sua satisfação (a frequência com que as requisições são atendidas) como sua justiça (se os recursos doados estão sendo retribuídos).

Neste trabalho reutilizamos a noção da “Rede de Favores” (NoF, do inglês *Network of Favors*), proposta anteriormente como um mecanismo de incentivo a colaboração em sistemas P2P. Resumidamente, a abordagem da NoF garante a

priorização de concessão de favores aos colaboradores em relação aos caronas. A NoF foi inicialmente concebida para o contexto de grades computacionais P2P oportunistas [Andrade et al. 2004], e devido aos recursos serem geralmente de baixo custo operacional, prioriza a satisfação do colaborador ao oferecer todos os recursos ociosos, na expectativa de acumular créditos com outros participantes, que podem ser convertidos em favores. Isto garante os melhores níveis possíveis de satisfação aos colaboradores, qualquer que seja o nível de contenção de recursos. Por este motivo, chamamos esta NoF de NoF Dirigida a Satisfação ou, abreviadamente, SD-NoF (do inglês Satisfaction-Driven NoF). No entanto, ofertar sempre todos os recursos ociosos pode prejudicar a justiça dos colaboradores já que em cenários de baixa contenção de recursos os caronas poderiam aumentar seu nível de satisfação consumindo os recursos excedentes do sistema.

Quando o cenário é a exploração oportunista de processadores ociosos em grades computacionais, os custos de amortização são geralmente desprezíveis, uma vez que os recursos são computadores pessoais que normalmente permanecem ligados durante parcelas de tempo consideráveis. Desse modo, o custo de fornecimento dos recursos excedentes não é muito maior do que mantê-los ociosos, e portanto a justiça não é mais importante do que a satisfação para os cenários nos quais a NoF foi originalmente pensada. Quando mudamos o contexto para federação P2P de provedores de nuvens privadas as prioridades passam a ser diferentes. Nuvens privadas proveem recursos dedicados e em larga escala, o que torna a justiça um fator extremamente relevante. A priorização de concessão de favores aos colaboradores que é realizada pela NoF já garante níveis adequados de justiça, mas somente em cenários de contenção de recursos. O objetivo deste trabalho é estender a NoF com algum mecanismo que permita que também em cenários sem contenção um participante colaborativo possa garantir bons níveis de justiça.

Esta nova estratégia, a qual chamamos de NoF Dirigida a Justiça ou, abreviadamente, FD-NoF (do inglês Fairness-Driven NoF), consiste na introdução de um mecanismo baseado na Teoria do Controle Retroalimentado [Doyle et al. 2013], que regula a quantidade de recursos que cada nuvem privada disponibiliza para a federação. Tal mecanismo se baseia na justiça atual proporcionada pela rede para definir a quantidade de recursos oferecidos à federação. Com isto, quando os colaboradores tentam melhorar seu nível de justiça através do controlador, eles regulam a contenção de recursos mantendo-a em um nível adequado, mesmo que este não seja seu objetivo principal, e por consequência diminuem a satisfação dos caronas com mais eficiência, principalmente quando o nível de contenção era considerado baixo. O desafio é projetar um algoritmo para o controlador que garanta que tanto o nível de justiça como o de satisfação sejam suficientemente bons para assegurar que a maioria dos participantes não saia do sistema, e que os caronas sejam mantidos com um baixo grau de satisfação.

O restante deste trabalho está organizado da seguinte forma. Na Seção II, discutimos alguns trabalhos relacionados. A Seção III detalha o funcionamento da Rede de Favores. Na Seção IV, apresentamos o problema específico que estamos lidando neste trabalho. A Seção V é dedicada à apresentação do mecanismo proposto que consiste em um Laço de Controle Retroalimentado para regular a quantidade de recursos disponibilizados ao sistema. As Seções VI e VII apresentam respectivamente os principais resultados obtidos e suas correspondentes análises. A Seção VIII finaliza este trabalho resumindo as principais conclusões e propostas de trabalhos futuros.

## 2. Trabalhos Relacionados

Muitas abordagens já foram exploradas com o objetivo de promover colaboração em sistemas P2P. Uma das soluções é usar esquemas de reputação, nos quais através de informações sobre o comportamento passado de outros nós, um nó pode decidir melhor sobre suas colaborações futuras. Na Rede de favores [Andrade et al. 2004], P2PRep [Damiani et al. 2003] e EigenRep [Kamvar et al. 2003], cada nó armazena localmente informações sobre suas próprias interações com outros nós, as quais são utilizadas para determinar suas reputações. Tais estratégias são suficientes para promover a colaboração e discriminar os caronas. No entanto, nenhum desses trabalhos lidam com o problema de simultaneamente fornecer níveis adequados de satisfação e justiça.

Satsiou e Tassiulas (2010) também propuseram um esquema baseado em reputação que controla dinamicamente a disponibilidade de recursos em sistemas de compartilhamento de arquivos em que a largura de banda dos nós são compartilhadas para capacidade de *download* e *upload*. A exemplo da nossa abordagem, o algoritmo utilizado para o controlador é simples, porém efetivo. Se durante determinado período a largura de banda que um dado nó receber for menor que a capacidade dedicada para *download*, o nó aumenta a capacidade de *upload* por um valor constante, visando assim melhorar sua reputação. Por outro lado, se o nó receber a mesma quantidade que a capacidade dedicada para *download*, isso significa que ele poderia aumentar esta capacidade para maximizar sua utilidade. O mecanismo utilizado garante a justiça dos participantes colaborativos assegurando que estes recebam recursos na mesma proporção de suas contribuições.

Com respeito a nuvens federadas, de forma semelhante à nossa estratégia para aumentar a utilidade dos recursos, Goiri, Guitart e Torres (2012) também tentam aumentá-la tomando a melhor decisão entre oferecer, requisitar, ou desligar recursos ociosos baseados em parâmetros como lucro, receita e custo. Mihailescu e Teo (2010) e Gomes et al. (2012) adotam a estratégia de precificação dinâmica para melhorar os benefícios da federação, configurando o preço ótimo dos recursos de acordo com a oferta e demanda. Em ambos os trabalhos não há nenhum problema com indivíduos caronas já que suas abordagens são monetárias. Tendo em conta que seu principal objetivo é a maximização da utilidade, esses trabalhos compartilham idéias semelhantes com o nosso, mas uma vez que o mercado que propomos é baseado em escambo, nosso trabalho maximiza a utilidade dos recursos ao discriminar os caronas por meio do controle da contenção.

Rochwerger *et al.* (2011) utiliza estratégias orientadas a políticas para encontrar de forma otimizada a melhor alocação de máquinas virtuais em máquinas físicas levando em conta aspectos econômicos de desempenho e de disponibilidade. Este trabalho não resolve de maneira efetiva o problema do parasitismo dos caronas, uma vez que as interações passadas não são levadas em conta para futuras decisões de cooperação. Adicionalmente, enquanto na nossa abordagem duas nuvens privadas conseguem ingressar na federação de forma autônoma para trocar recursos, no RESERVOIR a colaboração entre duas nuvens deve ser pré-definida em um acordo que especifica a capacidade em termos de número e tamanhos de máquinas virtuais que estão disponíveis para a federação, em conjunto com outras restrições como custo, QoS e mecanismos de autenticação. Estes requisitos impõem limites na escala que a federação pode alcançar. Como mencionado anteriormente, quanto maior for a federação, maiores serão as chances de correspondência simultânea entre picos e vales na demanda de membros diferentes.

### 3. A Rede de Favores

Na Rede de Favores (NoF), o valor de um favor é associado com o tempo e poder de processamento fornecidos. Assume-se que cada nó mantém um registro local do valor total de todos os favores concedidos e recebidos de cada nó com quem interagiu no passado. Sempre que um nó concede ou recebe um favor, o valor balanceado de todos os favores trocados com o nó com quem colabora é atualizado. Com base nesses valores, cada nó calcula um valor de reputação local associado com os nós com quem colabora. Para um nó A, a reputação do nó B só aumenta se A receber algum favor de B. A reputação atual que o nó A tem relacionada a outros membros competindo por seus recursos é usada por A para decidir para qual dos nós solicitantes será fornecido o recurso. Com esta abordagem o nó A sempre priorizará a concessão de favores a nós colaborativos. Isso garante que sempre que houver contenção para um recurso de propriedade de A, A priorizará os solicitantes com maior reputação.

Um nó A calcula  $r_A(B)$ , a reputação local do nó B, usando apenas duas informações: o valor total dos favores que recebeu de B e o valor total dos favores que concedeu a B. Considerando  $v(A, B)$  como sendo o valor total dos favores que A doou para B, em um dado momento, relativo a todas as interações passadas, podemos calcular a reputação  $r_A(B)$  como uma função de  $v(A, B)$  e  $v(B, A)$ . A reputação  $r_A(B)$  aumenta sempre que B concede um favor para A, e diminui sempre que A concede um favor para B. Além disso, se A e B nunca interagiram anteriormente,  $r_A(B) = 0$ . A definição mais simples de  $r_A(B)$  que satisfaz essas condições é:

$$r_A(B) = v(B, A) - v(A, B). \quad (1)$$

Em [Andrade et al. 2004], mostrou-se que com a ajuda deste simples mecanismo de reputação autônomo, os participantes que contribuem pelo menos tanto quanto eles consomem recebem na maior parte do tempo prioridade na concessão de favores de outros nós colaborativos.

O esquema de reputação da NoF é local, uma vez que os nós não usam informações de terceiros, e isto evita que os indivíduos maliciosos manipulem os valores de reputação. No entanto, um nó pode ser capaz de manipular sua reputação alterando sua identidade e aparecendo como um nó recém-chegado, sem interação prévia com os outros membros da federação. Em redes P2P, por exemplo, este é geralmente um modo simples de se alterar a identidade. Para resolver esse problema, Andrade *et al.* (2004) propuseram que a NoF deveria forçar a reputação de todos os nós ser não-negativa, da seguinte forma:

$$r_A(B) = \max\{0, v(B, A) - v(A, B)\}. \quad (2)$$

A equação 2 impede a priorização de indivíduos que maliciosamente mudam suas identidades ao invés dos colaboradores que tenham consumido mais recursos do que contribuído. Entretanto, um colaborador A não pode distinguir um nó malicioso B que nunca doou, de um nó colaborador C que já doou para A no passado mas consumiu de A a mesma quantidade. Para distingui-los, Andrade *et al.* (2004) introduziram na função de reputação  $r_A(B)$  um identificador histórico que leva em consideração a quantidade de doações que A recebeu de B no passado. Para evitar uma grande diferença entre as reputações de nós colaboradores já conhecidos e de nós recém-ingressados no sistema, o

que complicaria a entrada desses novos participantes, a NoF usa uma função sub-linear de  $v(B, A)$ <sup>1</sup> como o identificador histórico na definição de  $r_a(B)$ :

$$r_A(B) = \max\{0, v(B, A) - v(A, B) + \sqrt{v(B, A)}\}. \quad (3)$$

#### 4. Definição do Problema

Andrade *et al.* (2004) provaram que sempre que há contenção de recursos a NoF, a qual nos referimos também por SD-NoF, funciona bem e prioriza os colaboradores em relação aos caronas. Contudo, apenas priorizar os colaboradores não é suficiente em cenários com baixa contenção de recursos, uma vez que neste caso os caronas não são isolados de forma eficiente e o sistema se torna mais vulnerável a eles. Sempre que um colaborador doa recursos para caronas, ele está perdendo a oportunidade de beneficiar outros colaboradores, reduzindo, portanto, as chances de receber futuros favores. Além disso, os custos provenientes dos recursos doados para caronas jamais serão retribuídos.

Para verificar este fato e compararmos a eficiência da SD-NoF com a FD-NoF, precisamos mapear o comportamento de todos os nós em cenários de baixa contenção de recursos. Definimos  $c(A, t)$ ,  $d(A, t)$  e  $r(A, t)$  como, respectivamente, o montante total de recursos consumidos, doados, e requisitados por A até o tempo  $t$ . O valor de justiça de um nó A que doou pelo menos uma vez é, então, definido por  $j = \frac{c(A, t)}{d(A, t)}$ , e a satisfação de um nó A que realizou pelo menos uma requisição é dada por  $s = \frac{c(A, t)}{r(A, t)}$ .

##### 4.1. Modelo de Simulação

Para avaliar o comportamento dos participantes na SD-NoF, construímos um simulador para um modelo simplificado de uma federação P2P de nuvens privadas para compartilhamento de recursos. A federação é composta por uma comunidade de  $n$  nós, com  $(1 - f) \cdot n$  colaboradores e  $f \cdot n$  caronas,  $0 \leq f \leq 1$ . Estamos interessados em compreender como o conjunto de colaboradores se comporta em relação ao conjunto de caronas. Deste modo, para eliminar a influência que diferentes características dos participantes teriam nos resultados, assumimos que todos os colaboradores têm as mesmas necessidades de capacidade e demanda. Os caronas também têm as mesmas necessidades de demanda. Neste modelo a simulação avança em turnos. Em cada turno, cada participante colaborador pode estar em um estado consumidor com probabilidade  $\pi$  ou em um estado provedor com probabilidade  $1 - \pi$ . É assumido que cada colaborador têm capacidade total de recursos  $C$ , e demanda  $D \cdot C$ , onde  $D \geq 0$ . Na SD-NoF, quando em estado provedor, um colaborador sempre oferece todos os seus recursos, i.e.,  $C$ . Em cada turno os recursos são doados a um ou mais colaboradores selecionados entre os nós consumidores de acordo com o registro local de reputação. Os caronas, por outro lado, nunca doam e estão sempre em estado consumidor, cada um solicitando  $D \cdot C$  recursos em cada turno.

##### 4.2. Cenários

Contenção é uma métrica que define o grau de competição por recursos ofertados em cada turno da simulação, e é dada pela razão entre os recursos requisitados e disponibilizados. Assim, considerando apenas os colaboradores, a contenção de recursos é definida por:

$$\kappa = \frac{\pi \cdot (1 - f) \cdot n \cdot D \cdot C}{(1 - \pi) \cdot (1 - f) \cdot n \cdot C} = \frac{\pi \cdot D}{1 - \pi} \quad (4)$$

<sup>1</sup>Neste trabalho, utilizamos a raiz quadrada como função sub-linear

De acordo com a definição 4,  $\kappa$  depende exclusivamente de  $\pi$  e  $D$ . No intuito de comparar os casos com níveis de contenção baixo, moderado e alto, focamos em nosso projeto de experimentos os seguintes cenários:  $\kappa = 0.5$  (baixo),  $\kappa = 1.0$  (moderado), e  $\kappa \in \{2.0, 4.0\}$  (alto). Para simular estes cenários, definimos dois grupos de experimentos: 1)  $D$  fixo e  $\pi$  variável; 2)  $\pi$  fixo e  $D$  variável. No grupo 1, foi definido que  $D$  varia de 1 (valor baixo) a 3 (valor alto) com 15 valores equidistantes. Para cada valor que  $D$  assume, quatro valores de  $\pi$  são escolhidos de tal modo que conceba um cenário para cada um dos quatro níveis de  $\kappa$  no conjunto  $\{0.5, 1, 2, 4\}$ . No grupo 2,  $\pi$  varia de 0.2 (valor baixo) a 0.8 (valor alto) com 15 valores equidistantes, e para cada valor assumido por  $\pi$ , quatro valores correspondentes são escolhidos para  $D$  de modo que conceba um cenário para cada um dos quatro valores desejados de  $\kappa$ . Portanto, cada grupo de experimentos dá origem a 15 microexperimentos, cada um contendo quatro cenários – os quatro níveis desejados de  $\kappa$ .

Em nossa análise, estamos interessados nos casos em que os caronas possam afetar consideravelmente o nível de justiça dos colaboradores. Nesse sentido, dois pontos importantes devem ser considerados. Quanto maior a percentagem de caronas no sistema, mais recursos eles irão consumir no total, o que afeta a justiça dos colaboradores. Em contrapartida, menor será a quantidade de recursos que cada um deles será capaz de consumir. Isto acontece pois o montante total de recursos excedentes a ser consumido pelos caronas será partilhado por uma população maior, diminuindo seu nível de satisfação. Para avaliar esses dois pontos, definimos dois novos cenários para cada cenário discutido anteriormente: um com um alto valor<sup>2</sup> de  $f$  — escolhemos  $f = 0.75$ , e outro com o maior valor de  $f$  na qual a contenção entre os caronas seja baixa. Neste último caso, visamos situações em que os caronas produzirão o maior impacto sobre a federação ao mesmo tempo em que ainda preservam um bom nível de satisfação. Nós chamamos de f-ótimo os cenários em que  $f$  é escolhido dessa forma. Nesses cenários,  $f$  é calculado de modo que o número de caronas seja o mínimo necessário para consumir a média de recursos excedentes fornecidos por colaboradores em estado provedor, e não consumidos pelos colaboradores no estado consumidor. A expressão abaixo formaliza esta situação.

$$n \cdot f \cdot C \cdot D = (n \cdot (1 - f) \cdot (1 - \pi) \cdot C) - (n \cdot (1 - f) \cdot \pi \cdot C \cdot D), \quad (5)$$

o que leva a:

$$f = \frac{(1 - \pi) - (\pi \cdot D)}{D + (1 - \pi) - (\pi \cdot D)}. \quad (6)$$

Note que isto só se aplica quando  $\kappa < 1$ . Nos cenários em que  $\kappa \geq 1$ , configuramos f-ótimo de modo que  $n \cdot f = 1$ , caso em que não há disputa entre os caronas.

Quanto maior a quantidade de participantes na federação, pior será a situação para os caronas, mesmo que se mantenha o valor de  $f$ . Isto se deve ao fato de que quanto maior o número total de colaboradores, menor tende a ser o desvio do valor esperado do número de provedores em cada turno, i.e.,  $n \cdot (1 - f) \cdot (1 - \pi)$ . Em consequência disto, também será menor a probabilidade de que, dadas as diferentes proporções, uma quantidade significativa de recursos esteja disponível para os caronas em um dado turno. Assim, fixamos  $n$  em 100 para todos os cenários, levando a  $f = 1\%$  nos cenários para f-ótimo quando

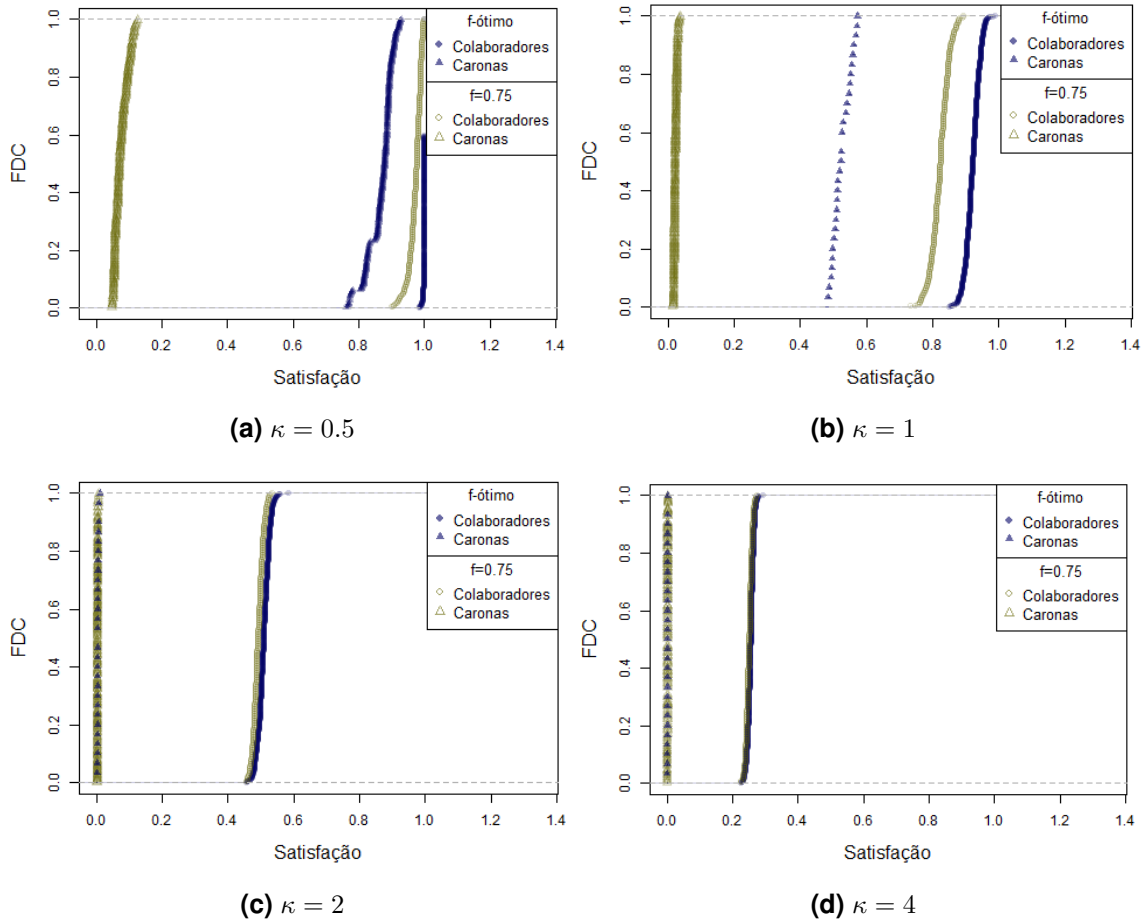
<sup>2</sup>Apesar do valor  $f = 0.75$  não ser comum em sistemas P2P, o escolhemos com o propósito de avaliar o sistema em um caso extremo onde existiria um excesso de caronas, ou um só carona entraria no sistema com múltiplos identificadores na tentativa de consumir mais recursos.



$\kappa \geq 1$ . Finalmente, por questões de simplicidade, consideramos  $C = 1$ . Simulamos todos os cenários com 4000 turnos, o suficiente para garantir que as métricas avaliadas converjam. Em resumo, o nosso projeto de experimentos tem dois valores constantes,  $n = 100$  e  $C = 1$ , e três fatores com valores variáveis:  $D$ ,  $\pi$  e  $f$ .

### 4.3. Resultados

Nas Figuras 1a, 1b, 1c e 1d, podemos avaliar os níveis de satisfação dos colaboradores e caronas, observando os valores para a Função de Distribuição Cumulativa (FDC) de cada participante no último turno da simulação para  $\kappa$  variando entre  $\{0.5, 1, 2, 4\}$ . A Figura 2 mostra o valor médio da justiça dos colaboradores (linhas azuis e amarelas) e sua dispersão<sup>3</sup> em cada um dos quatro cenários de  $\kappa$  para as duas configurações de  $f$ .

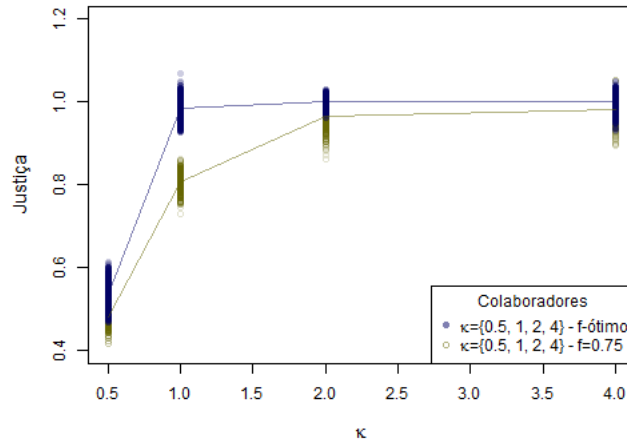


**Figura 1. FDC da satisfação dos colaboradores e caronas na SD-NoF.**

### 4.4. Análise

A primeira conclusão que podemos tirar da Figura 1 é que, mesmo sob as circunstâncias mais vantajosas para caronas – cenários de baixa contenção de recursos – a SD-NoF já assegura excelentes níveis de satisfação para os colaboradores em relação aos caronas. Além disso, as Figuras 1a e 1b mostram que um número maior de caronas teria pouco

<sup>3</sup>Para tanto, plotamos a justiça de cada um dos  $n \cdot (1 - f)$  colaboradores para cada valor de  $\kappa$  e de  $f$ .



**Figura 2. Justiça dos colaboradores na SD-NoF para  $f$ -ótimo e  $f = 0.75$ .**

impacto sobre a satisfação dos colaboradores, enquanto reduziria consideravelmente a própria satisfação dos caronas. Por outro lado, uma maior percentagem de caronas causa um impacto significativo sobre a justiça de participantes colaboradores em cenários de contenção baixa e moderada. Este fato pode ser observado na Figura 2, onde para  $\kappa \in \{0.5, 1\}$  e  $f = 0.75$  o nível de justiça oscila visivelmente abaixo do valor 1.

As Figuras 1c e 1d confirmam a conclusão de Andrade *et al.* (2004), observando-se que a eficácia da SD-NoF em termos de isolamento dos caronas torna-se ainda mais forte quando  $\kappa$  aumenta, independentemente da quantidade de caronas. Nós complementamos tais resultados mostrando que a SD-NoF não consegue garantir bons níveis de justiça em cenários de baixa, e até moderada, contenção de recursos. Por isso, defendemos que os colaboradores deveriam ser equipados com um mecanismo que lhes permitissem regular a quantidade de recursos doados, afetando indiretamente o nível de contenção de recursos do sistema. Isto deve ser feito de modo que altos valores de justiça sejam alcançados sem prejudicar os valores de satisfação. Tal mecanismo é proposto na próxima seção.

## 5. Laço de Controle Retroalimentado

Na SD-NoF, quando em estado provedor, um colaborador sempre disponibiliza todos os seus recursos ociosos ao sistema. O mecanismo adotado na SD-NoF permite priorizar os colaboradores em relação aos caronas, porém, quando  $\kappa < 1$ , todos os recursos excedentes serão provavelmente consumidos por caronas. Para evitar isto, os colaboradores podem regular a quantidade de recursos oferecidos, tentando aumentar indiretamente o valor de  $\kappa$ . Os colaboradores não têm conhecimento, e talvez nem mesmo interesse, para controlar diretamente o valor atual de  $\kappa$ , uma vez que a SD-NoF é projetada para sistemas P2P descentralizados. No entanto, é de interesse dos colaboradores tentar melhorar o seu nível de justiça, e isto é o que pode os levar a indiretamente aumentar  $\kappa$ . A maneira mais simples de aumentar a justiça é diminuindo a quantidade de recursos disponibilizados ao sistema. Se o nível de justiça é próximo de 1 para cada colaborador, o valor de  $\kappa$  estará perto de 1 conforme desejado. Contudo, embora a justiça seja importante, devemos também considerar o nível de satisfação, pois quanto menor a quantidade de recursos ofertados, menor também será o nível de satisfação dos participantes da federação, tanto

dos caronas como dos colaboradores.

O mecanismo proposto se baseia no algoritmo de *hill climbing* [Hinson and Staddon 1983], tentando alcançar um máximo local alterando a capacidade fornecida ao sistema baseado nos dois valores mais recentes de justiça. Ele também inclui uma simples abordagem para permitir que os colaboradores também tenham um bom nível de satisfação. Cada colaborador deve definir para si um valor limite  $\tau$  para seu nível mínimo desejado de justiça. O algoritmo básico de *hill climbing* é ativado sempre que a justiça de um colaborador for inferior a  $\tau$ . Uma vez que um colaborador atinge o nível mínimo desejado de justiça, o mecanismo buscará melhorar a satisfação do colaborador, aumentando gradativamente (turno a turno) a oferta de recursos até oferecer  $C$  (vide Algoritmo 1).

Através deste mecanismo, um colaborador poderia momentaneamente “sair” da federação, i.e., disponibilizando  $C = 0$ , até que receba alguns favores e decida que a federação está novamente rentável. O mecanismo considera um valor fixo que é adicionado ou deduzido do valor atual de recursos disponibilizados, neste trabalho escolhido para ser  $0.05 \cdot C$ , a fim de alcançar os níveis desejados de ambos justiça e satisfação.

---

**Algoritmo 1** Laço de Controle Retroalimentado

---

```
1: if justiçaAtual  $\leq$  justiçaAnterior then  
2:   if decisão = diminuir then  
3:     decisão  $\leftarrow$  aumentar  
4:   else if justiçaAtual  $<$   $\tau$  then  
5:     decisão  $\leftarrow$  diminuir  
6:   alteraCapacidade()
```

---

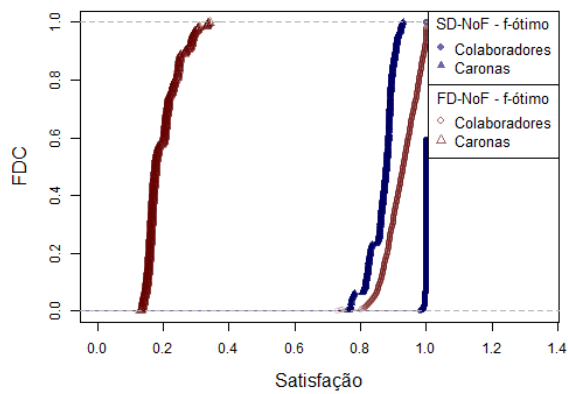
## 6. Resultados

Com o objetivo de avaliar o Laço de Controle Retroalimentado simulamos os mesmos cenários apresentados na seção 4.2, mas com o mecanismo ativado, i.e., utilizando a FD-NoF. Para estes experimentos configuramos  $\tau = 0.95$ .

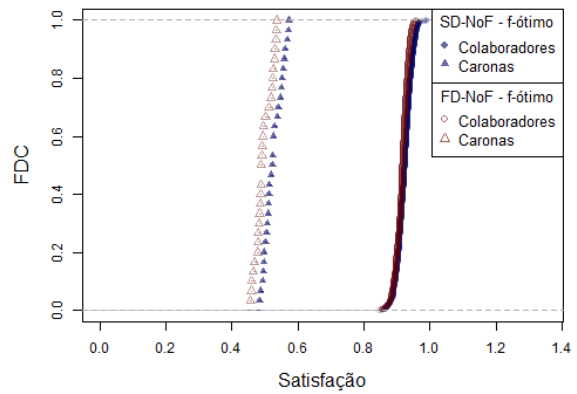
A Figura 3 apresenta a FDC dos valores de satisfação dos colaboradores e caronas no último passo da simulação, em ambas SD-NoF e FD-NoF com  $\tau = 0.95$ . Exibimos apenas os resultados para f-ótimo visto que este cenário é o mais favorável à satisfação dos caronas. A Figura 4 mostra a justiça média dos colaboradores e sua dispersão em todos os cenários nas quatro configurações de  $\kappa$ , no último passo da simulação, em ambas SD-NoF e FD-NoF com dois valores de  $\tau$  (0.8 e 0.95). Neste caso, mostramos apenas os cenários em que  $f = 0.75$ , já que estes têm um impacto mais forte sobre a justiça dos colaboradores. Finalmente, a Figura 5 mostra o impacto da percentagem de caronas na convergência do nível de contenção para os diferentes valores de  $\kappa$ .

## 7. Análise

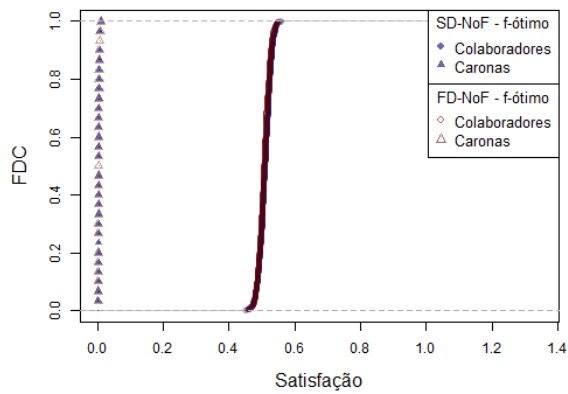
Das Figuras 3 e 4, podemos observar que a FD-NoF é mais eficaz do que a SD-NoF, como esperado. Em relação à satisfação dos colaboradores, a Figura 3a mostra que, quando  $\kappa = 0.5$  e  $f$  é ótimo, o melhor cenário para caronas, a FD-NoF diminui a satisfação dos caronas de uma média de 0.87 para uma média de 0.20 causando pouco impacto



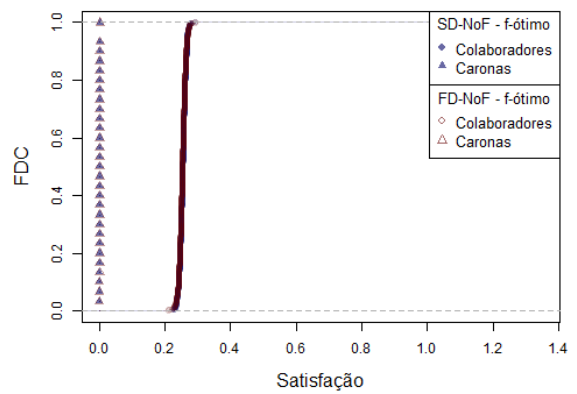
(a)  $\kappa = 0.5$ .



(b)  $\kappa = 1$ .

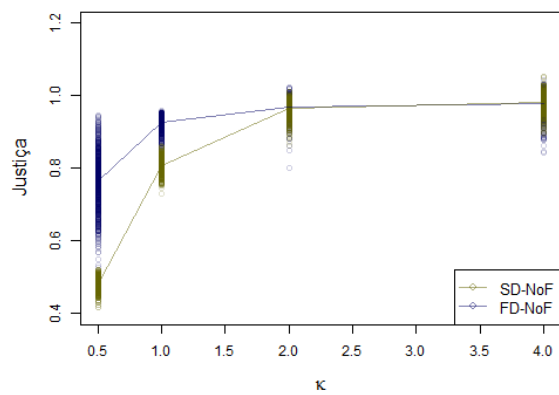


(c)  $\kappa = 2$ .

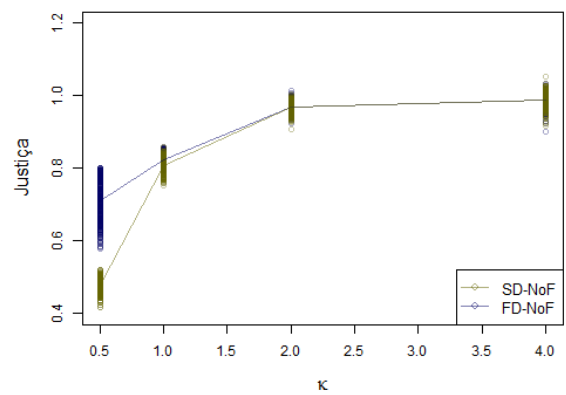


(d)  $\kappa = 4$ .

Figura 3. FDC da satisfação dos colaboradores e caronas na SD-NoF e FD-NoF.

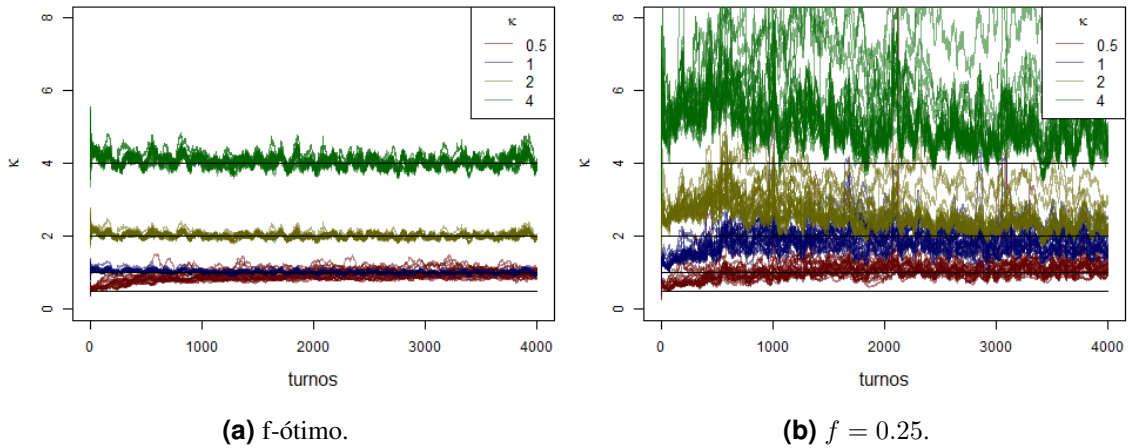


(a)  $f = 0.75$  e  $\tau = 0.95$ .



(b)  $f = 0.75$  e  $\tau = 0.8$ .

Figura 4. Justiça dos colaboradores na SD-NoF e FD-NoF.



**Figura 5. Média móvel de  $\kappa$  nos últimos 50 turnos com FD-NoF e  $\tau = 0.95$ .**

sobre a satisfação dos colaboradores, que diminui de uma média de 1.00 para uma média de 0.93. O cenário para  $\kappa = 1$  e f-ótimo também é muito vantajoso para os caronas, porque neste caso (e para todos os valores maiores de  $\kappa$  e f-ótimo) existe apenas um único carona no sistema que recebe todas os recursos excedentes. Contudo, a FD-NoF é ainda um pouco mais eficaz do que a SD-NoF em isolar os caronas, com um efeito desprezível na satisfação dos colaboradores. Em relação à satisfação, em níveis mais elevados de  $\kappa$  (Figuras 3c e 3d), o mecanismo de Laço de Controle Retroalimentado é inteligente o suficiente para não aumentar os já elevados valores de  $\kappa$  (buscando, por exemplo, melhorar a justiça) e, conseqüentemente, não diminui os já baixos valores de satisfação dos colaboradores. Por outro lado, o mecanismo não é capaz de diminuir um elevado valor de  $\kappa$  devido ao nosso modelo de simulação simplificado. Os valores iniciais de  $\kappa$  para os cenários da simulação dependem de  $\pi$  e  $D$ , que não se alteram ao longo dos turnos da simulação, além do valor inicial e máximo de  $C$ . Por este motivo, nos cenários de  $\kappa$  elevado, a satisfação dos colaboradores continua baixa, mesmo na FD-NoF.

Quando o valor de  $f$  é alto, apesar da satisfação individual dos caronas tender ser baixa, sua ação coletiva afeta fortemente a justiça dos colaboradores na SD-NoF, que é em média 0.48 e 0.81 para valores de contenção de 0.5 e 1, respectivamente (vide Figura 4b). Neste caso, a FD-NoF aumentou em média a justiça dos colaboradores por 60% e 15%, respectivamente, quando  $\tau$  foi ajustado com 0.95, e por 48% e 3%, respectivamente, quando  $\tau$  foi ajustado para 0.8. Adicionalmente, em cenários com altos valores de  $\kappa$ , a FD-NoF se comporta de forma semelhante à SD-NoF, independentemente de  $f$  e  $\tau$ , o que não é surpreendente, uma vez que a SD-NoF já funciona bem nesses cenários. Também notamos que quanto maior é o valor de  $\tau$ , mais difícil será a tarefa do controlador. Assim, não surpreendentemente, quando usamos  $\tau = 0.95$ , apenas alguns colaboradores conseguiram aproximar sua justiça a  $\tau$  quando  $\kappa = 0.5$ . Obviamente, à medida que  $\kappa$  aumenta, o mecanismo de Laço de Controle Retroalimentado se torna mais eficiente – a alta contenção de recursos por si só já assegura bons níveis de justiça, e nesse caso o controlador simplesmente aumenta  $C$  visando aumentar o nível de satisfação do colaborador. Além disso, menores valores de  $\tau$  levam a uma menor variabilidade entre os colaboradores.

Finalmente, a Figura 5 mostra como o valor de  $\kappa$  varia ao longo do tempo em dois cenários com valores distintos para  $f$ :  $f$ : f-ótimo e  $f = 0.25$ . Estes resultados mostram que a percentagem de caronas tem um efeito importante sobre o valor final em que o nível de contenção irá estabilizar. Quando o percentual de caronas é baixo, o sistema estabiliza bem em torno do  $\kappa$  desejado, i.e., 1 sempre que  $\kappa \leq 1$ , e  $\kappa$  sempre que  $\kappa \geq 1$  (lembrando das limitações do modelo de simulação simplificado). Quando o número de caronas aumenta, o não-determinismo também aumenta, levando o sistema a estabilizar  $\kappa$  em valores ligeiramente mais elevados do que o esperado. Porém, isso tem impacto limitado sobre a satisfação e justiça alcançadas por colaboradores. Em resumo, a FD-NoF atinge seu objetivo: os colaboradores melhoram seu nível de justiça ao controlar a quantidade de recursos fornecidos, sem sacrificar significativamente seu nível de satisfação.

## 8. Considerações Finais

Neste trabalho, tomamos como base a Rede de Favores e propomos a inclusão de um mecanismo de Laço de Controle Retroalimentado para regular a quantidade de recursos disponibilizados à federação e melhorar a justiça dos colaboradores. A NoF foi originalmente projetada para o contexto de grades computacionais P2P oportunistas e, portanto, tem como objetivo principal garantir um alto nível de satisfação aos colaboradores. Em contrapartida, quando se trata do contexto de federação de nuvens privadas, a justiça se torna uma prioridade, devido aos custos envolvidos na aquisição e manutenção de *datacenters*. A NoF originalmente introduzida por Andrade *et al.* (2004) é adequada para cenários com moderada e alta contenção de recursos. A introdução de um mecanismo de Laço de Controle Retroalimentado tem a intenção de aumentar o nível de justiça dos colaboradores através do controle da quantidade de recursos ofertados, que por consequência, aumenta o nível de contenção quando ele for considerado demasiadamente baixo para assegurar um bom nível de justiça.

Com o auxílio de um modelo simplificado de um sistema de P2P, em uma série de simulações avaliamos os resultados de nossa abordagem (FD-NoF), e os comparamos com os resultados obtidos na NoF original (SD-NoF). Essencialmente, mostramos que mesmo com um algoritmo simples para o Laço de Controle Retroalimentado, a FD-NoF cumpre seu objetivo. Em cenários com elevado percentual de caronas ( $f = 0.75$ ) e baixo nível de contenção ( $\kappa \in \{0.5, 1\}$ ), graças ao mecanismo de controle proposto, a FD-NoF melhora a justiça média dos colaboradores em pelo menos 15% e em até 60%, quando  $\tau = 0.95$ . Simplesmente controlando a quantidade de recursos disponibilizados ao sistema, os colaboradores indiretamente aumentam os níveis de  $\kappa$ , o que impacta negativamente os níveis de satisfação dos caronas, como desejado. Por outro lado, quando  $\kappa = 0.5$ , o nível de satisfação dos colaboradores diminui um pouco, o que é aceitável já que aqui o objetivo principal dos colaboradores é a melhoria da justiça (visando economizar despesas relacionadas, e.g., custos com energia). Além disso, a FD-NoF se comporta de forma semelhante à SD-NoF em cenários de alta contenção de recursos, o que é um resultado positivo uma vez que a SD-NoF por si só já assegura bons valores de justiça e os melhores valores de satisfação possíveis aos colaboradores.

Trabalhos futuros incluem a avaliação do desempenho da FD-NoF na nuvem federada composta pela infraestrutura computacional do projeto EUBrazilCC (<http://eubrazilcloudconnect.eu/>).

## 9. Agradecimentos

Este trabalho está inserido no contexto do *middleware* fogbow (<http://fogbowcloud.org/>) e é parcialmente financiado pelo CNPq, CAPES e pela Comissão Europeia através do projeto EUBrazilCC. Francisco Brasileiro é pesquisador CNPq.

## Referências

- [Andrade et al. 2004] Andrade, N., Brasileiro, F., Cirne, W., and Mowbray, M. (2004). 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.
- [Damiani et al. 2003] Damiani, E., di Vimercati, S. D. C., Paraboschi, S., and Samarati, P. (2003). Managing and sharing servants' reputations in p2p systems. *IEEE Transactions on Data and Knowledge Engineering*, 15(4):840–854.
- [Doyle et al. 2013] Doyle, J. C., Francis, B. A., and Tannenbaum, A. R. (2013). *Feedback control theory*. Courier Dover Publications.
- [Goiri et al. 2012] Goiri, I., Guitart, J., and Torres, J. (2012). Economic model of a cloud provider operating in a federated cloud. *Information Systems Frontiers*, 14(4):827–843.
- [Gomes et al. 2012] Gomes, E. R., Vo, Q. B., and Kowalczyk, R. (2012). Pure exchange markets for resource sharing in federated clouds. *Concurrency and Computation: Practice and Experience*, 24(9):977–991.
- [Grozev and Buyya 2014] Grozev, N. and Buyya, R. (2014). Inter-cloud architectures and application brokering: taxonomy and survey. *Software: Practice and Experience*, 44(3):369–390.
- [Hinson and Staddon 1983] Hinson, J. M. and Staddon, J. E. R. (1983). Matching, maximizing, and hill-climbing. *Journal of the Experimental Analysis of Behavior*, 40(3):321–331.
- [Kamvar et al. 2003] Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web, WWW '03*, pages 640–651, New York, NY, USA. ACM.
- [Marshall et al. 2010] Marshall, P., Keahey, K., and Freeman, T. (2010). Elastic site: Using clouds to elastically extend site resources. In *Conference on Cluster, Cloud, and Grid Computing (CCGRID)*, pages 43–52.
- [Mihailescu and Teo 2010] Mihailescu, M. and Teo, Y. M. (2010). Dynamic resource pricing on federated clouds. In *Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on*, pages 513–517.
- [Rochwerger et al. 2011] Rochwerger, B., Breitgand, D., Epstein, A., Hadas, D., Loy, I., Nagin, K., Tordsson, J., Ragusa, C., Villari, M., Clayman, S., Levy, E., Maraschini, A., Massonet, P., Munoz, H., and Toffetti, G. (2011). Reservoir - when one cloud is not enough. *Computer*, 44(3):44–51.
- [Satsiou and Tassiulas 2010] Satsiou, A. and Tassiulas, L. (2010). Reputation-based resource allocation in p2p systems of rational users. *Parallel and Distributed Systems, IEEE Transactions on*, 21(4):466–479.