

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Coordenação de Pós-Graduação em Ciência da Computação

Uma Política Autônoma de Compartilhamento  
para Grades Entre-Pares

Matheus Gaudencio do Rêgo

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores e Sistemas Distribuídos

Francisco Vilar Brasileiro

Nazareno Andrade

(Orientadores)

Campina Grande, Paraíba, Brasil

©Matheus Gaudencio do Rêgo, 27/09/2011

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

R343p Rêgo, Matheus Gaudencio do.  
Uma política autonômica de compartilhamento para grades  
entre-pares / Matheus Gaudencio do Rêgo. - Campina Grande, 2011.  
62f. : il.

Dissertação (Mestrado em Ciência da Computação) – Universidade  
Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.  
Orientadores: Prof. Francisco Vilar Brasileiro e Prof. Nazareno Andrade.  
Referências.

1. Sistemas Distribuídos. 2. Grades Computacionais.  
3. Compartilhamento. I. Título.

CDU 004 (043)

## Resumo

Grades computacionais buscam agregar diversos domínios administrativos para compartilhar recursos para a realização de determinada tarefa. Desta forma, estes recursos são compartilhados entre entidades com interesses próprios. Para definir uma forma de governança deste sistema, políticas são implementadas: sejam políticas definidas explicitamente entre os participantes da grade ou políticas gerais adotadas de forma autônoma. Em uma grade aberta, a definição de uma política geral é essencial pois a definição de políticas negociadas caso a caso inviabiliza a escalabilidade e a natureza desses sistemas. Uma política existente para grades computacionais entre-pares (*peer-to-peer*), a Rede de Favores (NoF, do inglês *Network of Favors*), trabalha na busca por reciprocidade buscando com que cada domínio administrativo da grade, representado por um nó no sistema entre-pares, seja retribuído de maneira proporcional ao quanto o mesmo doa para o sistema. Entretanto, a NoF não é capaz de privilegiar completamente o comportamento de colaboração. Este trabalho apresenta uma nova política de compartilhamento de recursos, a Rede de Colaboradores NoC (do inglês *Network of Collaborators*), que é capaz de privilegiar mesmo os doadores de alto consumo e de marginalizar apenas os nós que não colaboram com o sistema de forma alguma.

## **Abstract**

Grids assemble resources from broadly administrative domains that seek to share those resources among them. Explicit agreements can be made to define how resources will be shared among grid participants. Autonomus policies can also be used to coordinate this resource sharing. Open grids provides a scalable system which participants are free-to-join and follows a broad autonomus policy to provide such scalability. To encourage resources contributions, a grid should provide resource sharing policies that incentives donation. Network of Favours (NoF) is a currently used resource sharing mechanism that provides those incentives by returning donations using a tit-for-tat strategy. But NoF policy doesn't efficiently incentives collaboration behaviour. In some scenarios, freeriders can also obtain more benefits from a grid that uses NoF. We created a new resource sharing mechanism, the Network of Collaborators (NoC), that is capable to incentive collaboration even for peers that provides less resources than what they consume. Also, NoC is able to completely avoid resource usage from peers that never contributes to the grid.

## Agradecimentos

Aos meus orientadores, Fubica e Nazareno. Primeiro pela paciência, segundo, pela paciência e depois por todas as discussões e aprendizados que não se resumem apenas a este trabalho, mas ao longo do tempo que tive contato com ambos.

Aos meus familiares, Gabriel, Rogéria, Rômulo e Thaís. Por não perguntarem “Como está o mestrado” e “Defende quando?”. Imensamente grato pelo apoio sempre incondicional.

Aos colegas do LSD e de universidade. Sempre tive o prazer de trabalhar para mulheres como Érica, Zane, Priscylla, Bárbara e Ayla. Estas me ensinaram muito durante a graduação, e este aprendizado se seguiu na pós-graduação. Aos professores, especialmente, Dalton e Jorge, que me acolheram no estágio docência e me tornaram fascinado na pesquisa do ensino de programação. Agradecimentos aos colegas de luta Andréa, Edigley, Jaíndson, Lesandro, Lorena, Mariana e Marcus: mestrados que se apoiavam quando preciso. A Carla e Ricardo, por todas as companhias de almoço.

Ao pessoal da Copin, Vera, Rebecka, Hyggo e Aninha. Pela ajuda e acompanhamento neste mestrado. Por me ajudarem com toda a papelada ao longo destes anos. E por se preocuparem comigo e meu mestrado.

Aos colegas da vida, Cris, Felipe, Ponesa e Thábata. Por me escutarem e proporcionarem bons momentos de descontração quando necessário.

A todos que se importam com políticas: dentro e fora da grade. São as políticas que definem nossa sociedade e é a busca de melhores políticas que definem o que queremos para o futuro.

Mantenho as palavras do Prof. Dalton Guerrero, previamente parafraseadas pelo meu orientador Nazareno para os agradecimentos finais: “à parcela da população brasileira que, sem saber nem poder, financiou minha formação”.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	3
1.2	Contribuições . . . . .	3
1.3	Organização da Dissertação . . . . .	4
<b>2</b>	<b>Trabalhos Relacionados</b>	<b>5</b>
2.1	Grades Computacionais . . . . .	5
2.2	Sistemas Entre-pares . . . . .	7
2.3	Considerações do Estado da Arte . . . . .	8
<b>3</b>	<b>Modelo do Sistema</b>	<b>9</b>
3.1	Modelo da Grade . . . . .	9
3.1.1	Identificação do Nó . . . . .	10
3.1.2	Objetivo da Grade . . . . .	11
3.1.3	Objetivo dos Nós . . . . .	12
3.1.4	Retribuição dos Nós . . . . .	13
3.2	Modelo da Política de Compartilhamento . . . . .	13
3.2.1	Objetivos . . . . .	14
3.2.2	Arcabouço Operacional . . . . .	15
3.3	Sumário . . . . .	15
<b>4</b>	<b>Políticas de Compartilhamento</b>	<b>17</b>
4.1	Política da Rede de Favores . . . . .	17
4.1.1	Iniciação . . . . .	17
4.1.2	Fase de Recebimento . . . . .	18

4.1.3	Fase de Doação . . . . .	18
4.1.4	Fase de Priorização . . . . .	18
4.1.5	Exemplo de uso . . . . .	19
4.1.6	Comportamento da NoF . . . . .	20
4.2	Rede de Colaboradores . . . . .	25
4.2.1	Iniciação . . . . .	25
4.2.2	Fase de Recebimento . . . . .	25
4.2.3	Fase de Doação . . . . .	26
4.2.4	Fase de Priorização . . . . .	27
4.2.5	Exemplo de Uso . . . . .	28
4.3	Complexidade de Implementação das Políticas . . . . .	30
4.4	Sumário . . . . .	31
<b>5</b>	<b>Avaliação</b>	<b>32</b>
5.1	Metodologia . . . . .	32
5.1.1	Modelo de Simulação . . . . .	33
5.1.2	Métricas de Sucesso . . . . .	34
5.1.3	Cenários de Simulação . . . . .	35
5.2	Avaliação . . . . .	37
5.2.1	Objetivo 1 – Maximizar Satisfação dos Colaboradores . . . . .	38
5.2.2	Objetivo 2 – Tornar a satisfação do nó colaborador proporcional a sua retribuição . . . . .	41
5.2.3	Objetivo 3 – Ser robusta contra os ataques de múltiplas identidades e de mudança de identidade . . . . .	44
5.3	Sumário . . . . .	45
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>47</b>
6.1	Trabalhos Futuros . . . . .	48
6.1.1	Ataques Contra a NoC . . . . .	48
6.1.2	Uma Política para Nós Voluntários . . . . .	49

# Lista de Figuras

3.1	Modelo geral de uma grade entre-pares . . . . .	10
4.1	Representação dos estados de uma cadeia markoviana de uma relação par-a-par	21
4.2	Probabilidade do sistema estar em um estado cinza com 50% de probabilidade de transição . . . . .	23
4.3	Probabilidade do sistema estar em um estado cinza com 20% e 80% de probabilidade de transição . . . . .	24
5.1	$\mathcal{S}$ dos nós do início da simulação até o turno atual . . . . .	38
5.2	$\mathcal{S}$ para os nós com $c = 0.9$ . . . . .	41
5.3	$\mathcal{S}$ para os cenários onde $c = (0.6, 0.75, 0.9)$ . . . . .	42
5.4	$\mathcal{S}$ para os cenários onde $c = (0.3, 0.6, 0.9)$ . . . . .	43
5.5	$\mathcal{S}$ do nó malicioso $M$ controlando múltiplas identidades . . . . .	45



# Lista de Tabelas

4.1	NoF – Saldo inicial dos nós . . . . .	19
4.2	NoF – Saldo dos nós após consumo do nó $A$ . . . . .	19
4.3	NoF – Saldo dos nós após a doação do nó $A$ . . . . .	20
4.4	NoF: Saldo dos nós após a doação do nó $A$ . . . . .	20
4.5	NoC – Valores iniciais de $D$ , $C$ e $\mathcal{P}$ . . . . .	29
4.6	NoC – $D$ , $C$ e $\mathcal{P}$ após o consumo de recursos pelo nó $A$ . . . . .	29
4.7	NoC – $D$ , $C$ e $\mathcal{P}$ após as doações do nó $A$ . . . . .	30
5.1	Parâmetros do modelo de simulação . . . . .	34
5.2	Parâmetros avaliados . . . . .	37
5.3	$\mathcal{S}$ e $FR$ para os nós caroneiros na NoF no cenário onde $c = (0.3, 0.6, 0.9)$ . . . . .	39
5.4	$\mathcal{S}$ para os cenários com $c = (0.3, 0.6, 0.9)$ e com nós caroneiros . . . . .	40

# Capítulo 1

## Introdução

A tecnologia de grades computacionais permite o compartilhamento de recursos de diferentes domínios administrativos e o uso coordenado destes recursos [16]. Esta tecnologia possibilita principalmente a agregação de um número muito grande de recursos, provendo uma infra-estrutura que permite o processamento massivamente paralelo a um custo muito baixo, quando comparado a outras alternativas.

Uma grade pode operar através de acordos mútuos entre os ingressantes (*Virtual Organizations* como proposto por Foster *et al.* [17]) para ter garantias sobre o compartilhamento destes recursos. Estes acordos são definidos previamente ao ingresso da grade e são realizados de modo não autônomo, isto é, algum tipo de negociação prévia, possivelmente realizada fora da grade. Para a manutenção dessa grade, é necessário um esforço para tornar a grade conhecida, uma equipe com conhecimento para manter a infraestrutura da grade e unir os participantes em torno de um objetivo comum.

Entretanto, nem todo domínio administrativo possui o suporte ou recursos dedicados necessários para se juntar a tais grades. Ou ainda, alguns domínios não trabalham com um objetivo em comum. Para agregar recursos destes domínios criou-se a comunidade OurGrid [1; 10]. A comunidade OurGrid permite que diferentes domínios administrativos participem de uma grade computacional de livre acesso, utilizando o conjunto de recursos da grade de forma oportunista. Permite ainda tornar o sistema escalável e com baixo custo de manutenção ao adotar uma arquitetura entre-pares no seu sistema.

O modelo de uma grade aberta entre-pares provê políticas de compartilhamento que são gerais e adotadas por todos os membros durante o ingresso. Grades abertas apresentam uma

---

flexibilidade na construção de políticas próprias entre seus membros e são estas relações que governam a interação entre os diferentes domínios administrativos.

O OurGrid utiliza a política definida pela Rede de Favores (NoF<sup>1</sup>) [3] para determinar como as doações de recursos são priorizadas por um nó da grade entre-pares quando mais de um nó requer, simultaneamente, a utilização dos recursos daquele nó. O objetivo da política é incentivar doações através da reciprocidade. Para isso, nós da grade que mais doam recursos são os nós que devem receber mais recursos da mesma. Ou seja, espera-se que haja uma retribuição proporcional à quantidade de recursos que são doados por cada nó.

Para o seu funcionamento, a NoF se baseia em um valor de saldo. Cada nó armazena um saldo das interações par-a-par passadas para cada um dos nós da grade com os quais uma interação prévia já tenha ocorrido. Esta saldo registra um balanço dos favores realizados em cada relação entre-par. Caso um nó receba recursos, este incrementa os valores dos saldos associados aos nós que lhe doaram recursos. Caso um nó doe recursos, este decrementa os saldos dos nós favorecidos. A NoF contabiliza apenas favores positivos, nunca decrementando o saldo para um valor menor do que zero: isto garante algumas proteções ao sistema. A contabilidade de saldo define a política de incentivo de reciprocidade adotada na grade pois os nós que mais doarem terão mais valor de saldo e serão mais priorizados na grade.

No entanto, este funcionamento pode implicar na penalização de certas categorias de nós do sistema. Em particular, nós que ainda colaboram, mas que retribuem pouco, por consumirem mais do que doam, tendem a zerar rapidamente o seu saldo. Dentre os nós que apresentam saldo zero, estão os nós caroneiros. Estes nós tem interesse apenas em consumir recursos sem retribuir ao sistema. Ao zerar o saldo de nós que ainda colaboram com doações, mas apresentam baixa retribuição, estes são equiparados aos nós caroneiros e passam a disputar por recursos com uma categoria de nós que nunca irá retribuir ao sistema.

Buscando melhor priorizar toda categoria de nós colaboradores, ou seja, todos os nós que realizam alguma doação ao sistema, este trabalho propõe uma nova política de compartilhamento: a “Rede de Colaboradores” (NoC<sup>2</sup>). A NoC é capaz de diferenciar completamente os nós colaboradores dos nós caroneiros. Ao diferenciar estas duas categorias e aumentar o retorno dado aos nós colaboradores, a NoC aumenta o incentivo dos nós à doação, mesmo

---

<sup>1</sup>Do inglês *Network of Favors*

<sup>2</sup>Do inglês *Network of Collaborators*

para aqueles nós que não possam retribuir muito ao sistema.

## 1.1 **Objetivos**

Este trabalho tem como objetivo apresentar uma nova política de compartilhamento que é capaz de melhorar a priorização de recursos para os nós colaboradores da grade marginalizando o comportamento de nós do tipo “carona”. Esta política deve incentivar a doação de recursos na medida em que mantém o retorno obtido do sistema de forma proporcional à retribuição que cada nó dá ao sistema.

Assim, é objetivo geral deste trabalho:

- Apresentar uma nova política capaz de melhor satisfazer os nós colaboradores do sistema, inclusive aqueles nós de menor retribuição.

Como objetivos específicos, tem-se:

- Definir os objetivos gerais de uma política de compartilhamento de incentivos para nós colaboradores;
- Propôr um arcabouço comum que descreve o funcionamento geral de uma política de compartilhamento.

## 1.2 **Contribuições**

Este trabalho apresenta uma nova política de compartilhamento de recursos, a NoC, que é capaz de marginalizar o comportamento de carona e incentiva os nós a colaborarem com o sistema. A NoC é descrita seguindo um arcabouço de base para a descrição de uma política de compartilhamento e é analisada de acordo com os objetivos gerais propostos para tais políticas.

Como resultado, a NoC é uma política de compartilhamento robusta que considera a satisfação dos nós colaboradores ao marginalizar o comportamento caroneiro. Esta política ainda apresenta características de incentivo à doação pois, quando recursos são doados ao sistema, maior é o retorno oferecido pela grade.

## **1.3 Organização da Dissertação**

Este trabalho se organiza da seguinte forma; no Capítulo 2 discute-se os trabalhos relacionados. Em seguida, apresenta-se o modelo de grade e modelo de política de compartilhamento a serem explorados neste trabalho para, no Capítulo 4, apresentar uma discussão sobre o funcionamento geral da NoF e da NoC. No Capítulo 5 tais políticas são avaliadas e discutidas. O trabalho é finalizado com um capítulo dedicado às conclusões e trabalhos futuros.

# Capítulo 2

## Trabalhos Relacionados

Para a análise dos trabalhos relacionados, avalia-se as publicações de maior impacto no contexto de grades computacionais e de sistemas entre-pares buscando identificar quais os objetivos e funcionamento das políticas de compartilhamento apresentadas nesses trabalhos. Esta análise é feita através de uma discussão sobre os trabalhos relacionados nas áreas gerais sobre grades computacionais e sistemas entre-pares e concluída com considerações gerais sobre a literatura existente.

### 2.1 Grades Computacionais

O Globus Toolkit [17] é um middleware utilizado na construção de grades computacionais. Parte da infraestrutura provida pelo Globus define como diferentes domínios administrativos se organizam em uma grade computacional através de organizações virtuais (VO<sup>1</sup>) para implantar políticas de compartilhamento entre si. A definição desta política é algo a ser definido por cada um dos membros e em cada relação par-a-par das VOs. Apesar da grade poder definir uma política global que cada VO precisa assumir, a garantia e cumprimento destas políticas envolve acordos fora da grade feitos pelos participantes. Para poder realizar tais acordos fora da grade, é preciso então que cada nó possua uma identidade forte, ou seja, que cada nó da grade tenha sua identidade associada a uma entidade real que é responsável pelo nó na grade. Mais do que forte, tal identidade é custosa na medida de que sua obtenção não é trivial ao sistema. O Globus difere da proposta atual, na medida em que não considera a pos-

---

<sup>1</sup>Do inglês *Virtual Organization*

sibilidade da ausência de uma identidade que seja custosa aos participantes e não apresenta uma política global e autonômica a ser seguida pelos nós da grade.

O Condor [26] é um sistema que permite a agregação de recursos e execução de aplicativos de um grupo de usuários. Cada domínio administrativo representa um *Condor Pool*. O *Condor Flocking* [14] agrega diversos Condor Pools: uma ponte é construída de forma transparente, de modo que, em um Condor Pool, um recurso representa outro Condor Pool.

No Condor Flocking as políticas de compartilhamento são definidas entre os Condor Pools. Outra possível construção de Flocking é permitir o acesso de um usuário a diversos Condor Pools. Neste cenário, as políticas de compartilhamento são definidas entre o usuário e os diversos Condor Pools acessados. Por fim, o Condor-G [18] permite o acesso de um usuário a sistemas construídos no Globus Toolkit e Condor. Ainda assim, a política de compartilhamento é definida por contrato entre os usuários e os sistemas a que eles têm acesso, sempre utilizando uma identidade com grande custo associado para permitir a garantia dos contratos, encontrando-se a mesma limitação citada do Globus Toolkit.

A comunidade OurGrid [7] é uma iniciativa de uma grade aberta onde usuários podem doar e utilizar recursos seguindo uma política de incentivos discutida na introdução deste trabalho, a Rede de Favores [3] (NoF). A NoF também foi explorada no LBG Grid [9]. O LBG é um sistema que permite a construção e simulação de grades computacionais. Esse último trabalho apresenta um estudo sobre o uso de duas políticas de compartilhamento: FIFO e NoF. Ao contrário da nova política proposta, o modelo FIFO não apresenta incentivos à doação de recursos.

No sentido de expandir uma grade computacional, fala-se da ligação entre diversos tipos de grades. Trabalhos como o EDGeS [15] e de Brasileiro *et al.* [8] focam seus esforços no desenvolvimentos de mecanismos que permitem a coexistência de uma grade fechada com máquinas dedicadas e outros modelos de grades computacionais. Com o aumento de recursos, espera-se aumentar a escala da grade e contemplar novos cenários de uso. Entretanto, o trabalho do EDGeS ainda apresenta uma política semelhante à definida pelo Globus Toolkit e Condor, sem a flexibilidade que é foco deste trabalho. O trabalho de Brasileiro *et al.* utiliza a NoF como mecanismo de integração entre grades oportunistas e de serviço, sem apresentar uma nova proposta no funcionamento da política de compartilhamento.

## 2.2 Sistemas Entre-pares

Dentre os trabalhos de sistemas entre-pares, é possível observar mecanismos e políticas que busquem indentificar e melhor qualificar a reputação dos nós. Tais técnicas buscam privilegiar os nós que são mais capazes de uma melhor retribuição, e assim, melhorar a qualidade dos nós desse sistema.

Walsh *et al.* [27] descreve o *Credence* como mecanismo a ser utilizado no combate ao SPAM e a criação de falsos objetos em sistemas de compartilhamento de arquivos entre-pares. Neste mecanismo, a reputação do valor de cada recurso é definido por cada nó e por um consenso para a reputação deste recurso no sistema. Ao contrário do *Credence*, o trabalho atual busca operar na reputação de cada nó e não sob cada recurso provido por um nó.

De Costa e Almeida [11] cita o *Scrubbler* como um modelo para controle de reputação que faz uso ordenado do histórico dos nós. Ao contrário do *Credence*, a decisão da reputação passa a existir para definir a reputação de cada nó. Este mecanismo oferece uma melhor capacidade de discernir recursos maliciosos e opera de melhor forma quando comparado com o *Credence*. O *Scrubbler* é complementar ao trabalho atual na medida em que permite oferecer mecanismos para proteger uma grade computacional contra a presença de nós com recursos maliciosos. Mas ao contrário do *Scrubbler*, este trabalho busca trabalhar com a priorização dos diferentes tipos de nós colaboradores em um sistema.

No trabalho de Meulpolder *et al.* [23], ao comparar enxames de *BitTorrents* públicos e privados, é possível observar uma diferença maior de utilidade no sistema para os nós de enxames privados. Esta comparação tem uma aplicação direta neste trabalho. Nós de enxames públicos não armazenam qualquer informação sobre a história de seus usuários, enquanto, nos nós privados, há um controle rigoroso sobre o quanto estes nós podem baixar ou não, bem como um custo associado a criação da identidade deste nó. O sistema proposto neste trabalho trata de um sistema aberto em que nós tem livre ingresso, mas ao mesmo tempo apresenta uma proposta de política de compartilhamento que é capaz de definir sobre estes nós.



## 2.3 Considerações do Estado da Arte

As grades que trabalham com políticas próprias definidas por um contrato sobre a provisão e disponibilidade de recursos fazem uso de uma identidade de alto custo. Tais modelos de grade não são abertas e dificultam o livre acesso de um nó ao sistema. Para tais sistemas, a política de compartilhamento é definida par-a-par pelos participantes da grade para cada relação que possuem.

Em um modelo de grades abertas e entre-pares, a NoF é a política proposta para tal tipo de sistema. Entretanto, a NoF não apresenta um incentivo forte a todos os participantes da grade. Nós colaboradores de baixa retribuição por vezes se encontram em uma situação na qual são equiparados aos nós caroneiros. Basicamente, a NoF não mantém histórico e assim, é incapaz de diferenciar efetivamente tais categorias.

Nesta dissertação, apresenta-se uma nova política de compartilhamento, a NoC, que avança o estado-da-arte ao oferecer uma nova política que, ao contrário da alternativa existente, a NoF, é capaz de aumentar o retorno e incentivo à doação dos nós colaboradores, mesmo àqueles com baixa retribuição para o sistema. Para apresentar a nova política, este trabalho também apresenta os objetivos gerais de uma política de compartilhamento para uma grade computacional aberta entre-pares e um arcabouço que descreve o funcionamento geral destas políticas.

# Capítulo 3

## Modelo do Sistema

Toda política de compartilhamento define um comportamento geral de governança de um sistema. Este modelo de governança deve levar em consideração os participantes do sistema e o propósito desse sistema para definir os objetivos gerais da política bem como seu funcionamento.

Este capítulo apresenta o modelo geral da grade estudada e as características dos nós nessa grade. Descreve o sistema explorado e os comportamentos e interesses dos nós a serem estudados além dos objetivos e funcionamento geral destas políticas de compartilhamento. A Seção 3.1 apresenta o modelo geral de uma grade computacional aberta e entre-pares incluindo a descrição do funcionamento da grade e interesses dos nós. A Seção 3.2 apresenta o modelo de uma política de compartilhamento para a grade proposta: seus objetivos e em que momentos a política atua no sistema.

### 3.1 Modelo da Grade

Para entender as políticas propostas, é preciso primeiramente modelar o sistema a ser estudado. A Figura 3.1 representa um exemplo de uma grade entre-pares a ser estudada. Cada nó representa um domínio administrativo diferente que mantém relações par-a-par com os demais nós. Não há necessidade de um elemento central de controle e cada nó desse sistema se comunica com os demais nós para requisitar ou doar recursos num instante de tempo. Uma requisição de recursos não satisfeita num instante de tempo é descartada.

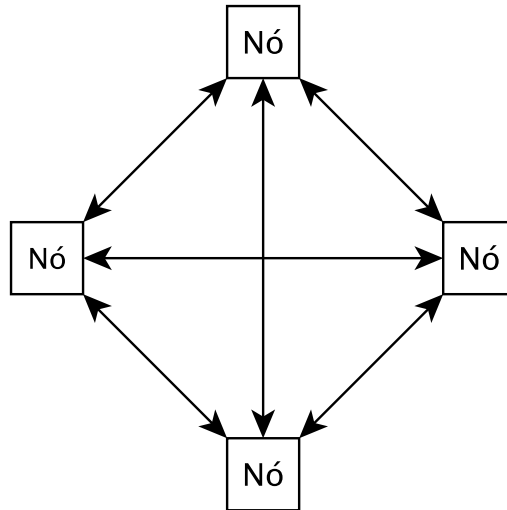


Figura 3.1: Modelo geral de uma grade entre-pares

### 3.1.1 Identificação do Nó

Em uma grade computacional, o recurso é o processamento computacional. Em um sistema onde todos os nós apresentam máquinas idênticas, o recurso se traduz em tempo de computação realizável por cada nó. Dada uma tarefa oriunda da grade, um nó pode processar esta tarefa doando assim o seu recurso, ou seja, processamento computacional. Ao contrário de alguns modelos de grade, para a troca de arquivos onde um arquivo é o recurso, como proposto por Bharambe *et al.* [6], em uma grade computacional o recurso não é duplicável. Isto implica que a troca de um recurso não aumenta a disponibilidade do mesmo no sistema.

Um nó pode se juntar à grade a qualquer momento. Para isto, o nó cria uma identificação única que busca: *i)* permitir que o nó seja reconhecido nas suas interações com os seus pares, e *ii)* garantir que nenhum outro nó possa assumir essa identidade. Estes requisitos são implementáveis e aplicáveis com o uso de criptografia assimétrica através da criação de uma chave pública e privada como proposto por Figueiredo *et al.* [12]. A chave pública assume a identificação do nó e a chave privada é utilizada para garantir a autenticidade das interações realizadas com um determinado nó.

Caso dois nós mantenham a mesma identidade nas comunicações par-a-par, estes nós se reconhecem e podem usar o histórico de suas interações para definir como será feita a troca de recursos. Um nó pode também gerar uma nova identidade com o objetivo de zerar seu histórico nesta relação entre-pares. Ao contrário de grades que forçam a presença de uma

identidade de alto custo que associa o nó a uma entidade real, o modelo proposto permite que um nó possa trocar sua identidade caso assim deseje. Quando o nó realiza esta troca para tentar obter mais ganhos do sistema do que o planejado, diz-se que este realiza um *whitewashing*. Tal técnica representa um ataque ao sistema e foi discutida por Friedman e Resnick [19]. Os autores argumentam que, em um sistema sem custo para a troca de identidade, apenas a presença de incentivos à manutenção da identidade é capaz de evitar tal tipo de ataque.

Há ainda outro ataque citado na literatura, o uso de múltiplas identidades [5; 13]. Quando não há custo na criação de identidades, uma possibilidade de tentar obter ganho do sistema de forma contrária à definida pela política é a criação de vários nós na grade sob o controle do mesmo domínio administrativo.

No modelo de grade proposto, não há custo em criar uma nova identidade. Esta identidade se apresenta ao sistema como um nó que nunca interagiu com outro nó. Entretanto, uma política que incentive a doação de recursos, deve associar um custo para que um nó receba privilégios de outro nó do sistema. Numa política de compartilhamento de recursos, o custo de tornar um nó prioritário para o recebimento de recursos é representado pelas doações realizadas por este nó que requisita recursos. Ao retribuir apenas os nós doadores de um sistema, uma política de compartilhamento implementa um mecanismo robusto de proteção contra ataques de múltiplas identidades pois, por mais que seja fácil criar uma nova identidade, há um custo associado na manutenção desta identidade para que este nó possa ser retribuído. Neste sistema, o custo existe na forma de doações realizadas.

### 3.1.2 Objetivo da Grade

O objetivo de uma grade computacional entre-pares é agregar os recursos disponíveis nos diferentes domínios administrativos e permitir que os recursos de um nó em um determinado instante de tempo possam ser usados por outros nós que tenham demanda por recursos naquele instante de tempo. Cada nó apresenta uma carga de trabalho particular e uma quantidade própria de recursos disponibilizados para o sistema. Para a realização de um trabalho, o nó pode utilizar seus recursos e pedir por mais recursos aos demais nós da grade. Quando não há trabalho a ser realizado por um nó, este pode disponibilizar seus recursos para serem utilizados pelos demais nós. Por esse motivo, este tipo de grade é dito oportunista.

A grade deve oferecer uma política geral de compartilhamento que incentive a participação dos nós com o aumento de recursos disponíveis no sistema levando em consideração: *i)* o interesse dos nós no ganho obtido (satisfação), e; *ii)* as características de comportamento dos nós em relação à sua carga de trabalho e à quantidade de recursos disponibilizados no sistema (retribuição).

### 3.1.3 Objetivo dos Nós

Para modelar os nós, considera-se que cada nó assume um comportamento egoísta. Um nó espera obter o máximo de recursos possíveis da grade da forma mais rápida possível, diminuindo o tempo de realização de suas tarefas (*makespan*), e contribuindo apenas o necessário para atingir tal objetivo.

Neste trabalho, o ganho obtido pelo nó em um determinado intervalo de tempo, ou seja, a satisfação de um nó é definida por  $\mathcal{S}(\Delta t)$  (fator de satisfação). Onde, para o intervalo de tempo ( $\Delta t$ ),  $\mathcal{S}(\Delta t) = 1$  representa a máxima satisfação do nó e  $\mathcal{S}(\Delta t) = 0$  representa um caso onde o nó não é, em nada, satisfeito.

Para entender  $\mathcal{S}(\Delta t)$  é preciso considerar duas dimensões: a quantidade de recursos recebidos e o tempo necessário para receber estes recursos. É objetivo geral do nó ser capaz de realizar suas tarefas da forma mais rápida possível, ou seja, mais do que receber recursos da grade, o nó precisa receber estes recursos rapidamente.

Assim, modela-se  $\mathcal{S}(\Delta t)$  como a capacidade de receber recursos na medida em que há requisições em uma janela de tempo, ou seja,  $\mathcal{S}(\Delta t) = \frac{\text{recursos recebidos}_{\Delta t}}{\text{recursos requisitados}_{\Delta t}}$ , sendo  $\text{recursos requisitados}_{\Delta t} > 0$ . O nó de comportamento egoísta é idealmente satisfeito se todas as requisições de recursos realizadas em um  $\Delta t$  forem atendidas em  $\Delta t$ , para todo intervalo  $\Delta t$  onde há requisição de recursos.

Nesta métrica, para todo  $\Delta t$  onde  $\text{recursos requisitados}_{\Delta t} > 0$ ,  $\mathcal{S}(\Delta t) = 1$  é o objetivo ideal de todos os nós do sistema. Um nó de comportamento egoísta que obtem este fator de satisfação foi capaz de realizar seu trabalho ao receber todos os recursos requisitados imediatamente. Isto acontece, por exemplo, em sistemas sem contenção pois há um excesso de recursos que sempre satisfaz a demanda dos nós. Num intervalo de tempo  $\Delta t$  onde há contenção dos recursos nem todos os participantes podem obter o valor  $\mathcal{S}(\Delta t) = 1$ . Entretanto, tais cenários de contenção existem em grades devido à variação de carga de trabalho e de

recursos disponíveis como apresentado por Iosup *et al.* [21].

### 3.1.4 Retribuição dos Nós

Como os nós assumem comportamentos distintos na grade, em especial, por conta da variação da carga de trabalho que eles impõe ao sistema e do número de recursos que eles disponibilizam ao sistema, define-se esta variação como a capacidade que cada nó tem em retribuir à grade. Isto representa uma característica do nó que traduz como o mesmo deseja doar recursos em relação ao que deseja consumir do sistema. Difere de  $\mathcal{S}$  pois é uma medida que descreve o padrão do nó, ao contrário de uma medida obtida pela a avaliação do modelo. Assim, o valor da retribuição de cada nó é capaz de caracterizar os diferentes tipos de nós de um sistema.

Um nó pode disponibilizar muitos recursos, mas ter uma carga intensa de trabalho. Outro nó pode disponibilizar poucos recursos, mas passar a maior parte do tempo doando ao sistema. Define-se como  $\mathcal{R}(\Delta t)$  (fator de retribuição) o comportamento que leva em consideração a variação da quantidade de recursos presentes por nó e o quanto cada nó deseja consumir do sistema em um determinado intervalo de tempo em que há requisição de recursos. Assim,  $\mathcal{R}(\Delta t) = \frac{\text{recursos disponibilizados}_{\Delta t}}{\text{recursos requisitados}_{\Delta t}}$ , para qualquer intervalo de tempo  $\Delta t$  onde  $\text{recursos requisitados}_{\Delta t} > 0$ .

No modelo, adota-se três categorias distintas de retribuição dos nós de comportamento egoísta ao longo do seu tempo de vida: *i*) os nós caroneiros, onde  $\mathcal{R}(\Delta t) = 0$ ; *ii*) os nós que desejam consumir mais ou igual ao que doam ao sistema, onde  $0 < \mathcal{R}(\Delta t) \leq 1$ , e; *iii*) os nós que doam mais do que desejam consumir do sistema, onde  $\mathcal{R}(\Delta t) > 1$ .

## 3.2 Modelo da Política de Compartilhamento

Toda política objetiva satisfazer interesses intrínsecos dos participantes do sistema e do próprio sistema. Assim, uma política determina qual o comportamento ideal esperado dos nós e como atuar no sistema para atingir tais objetivos.

No modelo apresentado anteriormente, é de interesse dos nós obter  $\mathcal{S}(\Delta t) = 1$  para cada instante de tempo  $\Delta t$  onde há requisições de recursos. A grade por sua vez deseja prover um ambiente robusto e com incentivos à doação de recursos. Assim, os objetivos de

uma política de compartilhamento devem considerar tais interesses para operar no sistema. Esta seção descreve tais objetivos bem como um arcabouço operacional para as políticas de compartilhamento. Este arcabouço mostra as fases de decisão em que os nós do sistema devem atuar de acordo com uma política.

### 3.2.1 Objetivos

São apresentados três objetivos básicos de uma política de compartilhamento para o modelo de grade proposto. Estes objetivos refletem os interesses dos nós e da grade para guiar uma política de compartilhamento.

**Objetivo 1.** *Maximizar a satisfação dos nós colaboradores.*

Define-se como objetivo primário maximizar a satisfação dos nós colaboradores do sistema. Nós de comportamento egoísta estarão satisfeitos quando conseguirem realizar rapidamente toda sua carga de trabalho. Idealmente, uma política busca fazer com que todo nó obtenha  $\mathcal{S}(\Delta t) = 1$ . Entretanto, este cenário não é sempre possível haja vista que pode haver ocasiões em que há contenção na grade; com menos recursos, alguns nós poderão não atingir este grau de satisfação. Para aumentar a satisfação dos nós colaboradores e assim incentivar o comportamento de doação, a política não deve dar incentivos aos nós em que  $\mathcal{R}(\Delta t) = 0$ , dado qualquer instante  $\Delta t$  em que tais nós requisitam recursos.

**Objetivo 2.** *Tornar a satisfação do nó colaborador proporcional a sua retribuição.*

Para aumentar o número de recursos disponíveis na grade respeitando as características de doação e de uso de recursos dos nós da grade, determina-se como objetivo de uma política ser capaz de satisfazer um nó de forma proporcional à retribuição deste no sistema. Assim, o Objetivo 2 busca incentivar as doações ao sistema, na medida que se garante que os nós com maior valor de  $\mathcal{R}(\Delta t)$  são os mais satisfeitos da grade.

**Objetivo 3.** *Ser robusta contra os ataques de múltiplas identidades e de mudança de identidade.*

Discutiu-se anteriormente a possibilidade de ataques através de manipulação de identidades. Toda política deve ser robusta o suficiente para evitar que exista a possibilidade do

uso de algum mecanismo oferecido pelo sistema, como a troca ou criação de novas identidades, para aumentar a satisfação de um nó de forma contrária à definida pela política. Não se esgotam as possibilidades de existirem outros tipos de ataque. Neste trabalho, busca-se a garantia mínima de proteção da política contra estes dois ataques mais conhecidos em grades e sistemas entre-pares.

### 3.2.2 Arcabouço Operacional

Seguindo o modelo entre-pares proposto, um nó ao requisitar por recursos, procura obter estes recursos dos demais nós do sistema. Ao realizar tal requisição, os demais nós do sistema determinam a prioridade que este nó tem em receber recursos. Um nó, ao ser priorizado, recebe recursos do nó doador. Assim, uma política de compartilhamento atua em três fases distintas da operação de um nó:

**Recebimento** O momento em que se recebe recursos de um nó;

**Doação** O momento em que uma doação foi realizada para outro nó;

**Priorização** Fase em que um nó recebe requisições e precisa definir que nós receberão seus recursos.

É importante observar que, durante a oferta de recursos, um nó requisitante pode receber mais ofertas de doação do que necessita. Nesta situação este nó deve fazer uma escolha pelos recursos que este considera obter um melhor retorno, dada a sua carga de trabalho. Esta decisão não é definida por uma política de compartilhamento global, mas sim por uma política de alocação local. Ou seja, cabe apenas ao nó que requisitou recursos decidir quais os recursos que serão usados. Esta alocação é discutida em outros estudos, como a proposta de Nóbrega-Júnior *et al.* [24], e não é foco deste trabalho.

## 3.3 Sumário

Este capítulo apresentou um modelo de sistema que representa uma grade computacional aberta entre-pares. Uma grade é composta de nós que apresentam como interesse aumentar sua satisfação, ou seja, aumentar o retorno do sistema de forma a realizar rapidamente sua



carga de trabalho. Para que haja um incentivo à doação dos recursos de um nó, a grade precisa operar de forma que este retorno do sistema seja proporcional à retribuição de cada nó ao sistema.

Através do modelo de grade, define-se um modelo de política de compartilhamento. Este modelo caracteriza os objetivos gerais de uma política de compartilhamento para o modelo de grade apresentado. São estes objetivos: *i)* maximizar a satisfação dos nós colaboradores; *ii)* tornar a satisfação do nó colaborador proporcional a sua retribuição, e; *iii)* ser robusta contra os ataques de múltiplas identidades e de mudança de identidade.

Definido os objetivos gerais de uma política, construiu-se um arcabouço operacional para tais políticas. Este arcabouço identifica as fases de atuação em que uma política opera. Cada fase representa uma interação entre os nós do sistema e, assim, um momento para atuação da política de compartilhamento.

# Capítulo 4

## Políticas de Compartilhamento

No capítulo anterior, foi apresentado o modelo geral de uma política de compartilhamento; descrevendo objetivos gerais e seu arcabouço de operação. Este capítulo descreve a NoF [4] na Seção 4.1 seguindo o arcabouço proposto. É feita uma discussão sobre as implicações do uso de tal política levando em consideração os objetivos apresentados. A Seção 4.2 apresenta o funcionamento da nova política proposta (NoC). Em seguida, na Seção 4.3, descreve-se a complexidade computacional das políticas discutidas. O capítulo é concluído com um sumário que sintetiza as discussões apresentadas ao longo da descrições das políticas.

### 4.1 Política da Rede de Favores

A política de compartilhamento da Rede de Favores, ou NoF, usa a relação de consumo par-a-par para determinar a distribuição dos recursos na grade. Todos os recursos de um nó colaborador são distribuídos de forma proporcional ao valor do saldo de favores que cada nó armazena em relação aos nós que estão pedindo recursos. Quanto maior o saldo de um nó, maior a prioridade deste nó em receber recursos.

#### 4.1.1 Iniciação

A NoF inicia o saldo dos nós da rede com o valor zero. Ao entrar no sistema, um nó não realizou nenhuma operação de troca de recursos. Assim, para este nó, o saldo de qualquer outro nó com quem venha a interagir no futuro é iniciado com o valor zero.

### 4.1.2 Fase de Recebimento

Ao receber recursos de um outro nó  $P$ , um nó da grade contabiliza este saldo de favores recebidos. Neste caso a atualização de saldo é feita com o incremento do saldo atual, ou seja,  $Saldo[P] = Saldo[P] + recursos\ recebidos$ .

### 4.1.3 Fase de Doação

Ao realizar uma doação de recursos a um nó  $P$ , o nó doador decrementa do saldo atual de  $P$  o valor de recursos doados. Caso o nó consumidor utilize mais recursos do que o registrado no saldo, o novo saldo assume o valor 0. Assim, o cálculo do saldo é feito da seguinte maneira:  $Saldo[P] = \max(0, Saldo[P] - recursos\ doados)$ .

Não há contabilidade negativa de favores, do contrário, se incentivaria determinados nós do sistema a assumirem uma nova identidade. Por exemplo, caso o saldo negativo fosse contabilizado, seria vantajoso que um dado nó  $B$ , ao atingir um saldo negativo em um nó  $A$ , saísse do sistema e retornasse como um novo nó  $C$ . Para um nó  $A$ ,  $Saldo[C]$  seria zero, e não um valor negativo como seria o valor de  $Saldo[B]$ . A NoF busca ser robusta para satisfazer o objetivo 3 ao não incentivar o uso de *whitewashing* pois a troca de identidade não coloca o nó  $B$  em vantagem no sistema.

### 4.1.4 Fase de Priorização

Quando dois ou mais nós requisitam recursos, um nó colaborador que queira realizar uma doação entra na fase de priorização. Nesta situação, este nó doador precisa definir quais nós receberão seus recursos ociosos. Para tanto, define-se o valor de saldo como medida de prioridade de doação. Cada nó que requisita recursos recebe, do nó doador, o valor de saldo como valor de prioridade.

Dado este valor de prioridade, Andrade *et al.* [4] citam dois modelos de doação: a doação direta e total de todos os recursos ao nó com maior saldo, e a doação proporcional de acordo com o saldo. No modelo em que há alocação direta, o nó de maior prioridade recebe todos os recursos do nó doador. No modelo proporcional, cada saldo é avaliado e cada nó requisitante recebe uma quantidade de recursos proporcional a este saldo. No primeiro modelo, caso haja empate entre os nós que requisitaram recursos, um destes é escolhido aleatoriamente para

receber tais recursos. No modelo proporcional, os recursos são distribuídos primariamente entre os nós de  $Saldo > 0$ . Ainda neste modelo, caso existam recursos disponíveis, estes são distribuídos equalitariamente entre os nós requisitantes de  $Saldo = 0$ . Neste modelo, um recurso é dado a cada nó consumidor até que não haja mais recursos disponíveis pelo provedor. De acordo com Andrade *et al.* [4], não há diferença entre os dois modelos de alocação no ganho geral dos nós, sendo a alocação proporcional atualmente utilizada pela comunidade OurGrid.

Ao priorizar a doação aos nós de maior valores de saldo, a NoF busca cumprir o Objetivo 2: a satisfação dos colaboradores aumenta de forma proporcional à retribuição do nó no sistema. Quanto maior a doação e menor o consumo, maior o valor de saldo gerado e, consequentemente, maior será a prioridade deste nó em, posteriormente, receber recursos.

#### 4.1.5 Exemplo de uso

Como exemplo, considere três nós:  $A$ ,  $B$  e  $C$ . Inicialmente, o saldo de cada par de nós é zero, como exibido na Tabela 4.1.

Tabela 4.1: NoF – Saldo inicial dos nós

	$A$	$B$	$C$
$Saldo[A]$	–	0	0
$Saldo[B]$	0	–	0
$Saldo[C]$	0	0	–

Considerando que todos os nós apresentam recursos com mesmo poder computacional, caso o nó  $B$  doe 10 horas de processamento para o nó  $A$  (seja provendo 1 recurso por 10 horas, ou 10 recursos por 1 hora) e o nó  $C$  doe 4 horas de processamento para o nó  $A$ , o saldo de  $A$  é atualizado conforme os valores da Tabela 4.2.

Tabela 4.2: NoF – Saldo dos nós após consumo do nó  $A$

	$A$	$B$	$C$
$Saldo[A]$	–	10	4
$Saldo[B]$	0	–	0
$Saldo[C]$	0	0	–

Considerando o modelo de alocação proporcional e que os nós  $B$  e  $C$  requisitaram 30 recursos cada, e que o nó  $A$  tenha 7 recursos disponíveis para doação, eles receberão 5 e 2 recursos respectivamente. Caso estes façam uso destes recursos por 1 hora, o saldo dos nós são atualizados conforme a Tabela 4.3.

Tabela 4.3: NoF – Saldo dos nós após a doação do nó  $A$ 

	$A$	$B$	$C$
$Saldo[A]$	–	5	2
$Saldo[B]$	5	–	0
$Saldo[C]$	2	0	–

Caso o nó  $B$  faça novamente o uso dos recursos do nó  $A$  e utilize 10 recursos no total, o valor de  $Saldo[B]$  para o nó  $A$  é igual a zero. O resultado final dos saldos, nesta situação, é demonstrado na Tabela 4.4.

Tabela 4.4: NoF: Saldo dos nós após a doação do nó  $A$ 

	$A$	$B$	$C$
$Saldo[A]$	–	0	2
$Saldo[B]$	15	–	0
$Saldo[C]$	2	0	–

#### 4.1.6 Comportamento da NoF

A política definida pela NoF permite que o compartilhamento de recursos privilegie os nós de melhor saldo buscando atingir o objetivo de tornar a satisfação dos nós proporcional à retribuição dos nós do sistema.

No entanto, um dos aspectos não cobertos pela NoF é a proteção contra caroneiros de forma efetiva. Os nós com mais doação do que consumo (maior retribuição) são representados e recebem recursos de acordo com o saldo gerado. Entretanto, os nós com maior consumo do que doação de recursos, podem se encontrar em uma situação de desvantagem perante os nós caroneiros, ou seja os nós que nunca contribuem com o sistema.

É possível observar este cenário quando um nó colaborador  $A$  interage com outro nó colaborador  $B$  de baixa reciprocidade ao sistema. Como exemplo, o nó  $B$ , ao consumir mais

do que doa de  $A$ , acaba fazendo com que  $Saldo[B]$  em  $A$  seja zero. Este nó, mesmo que tenha contribuído recursos para  $A$  no passado, se encontra na mesma situação de um nó caroneiro.

Assim, a NoF apesar de apresentar uma política em que a satisfação de um nó seja proporcional a sua retribuição, não é capaz de fazê-lo de forma plena para os nós colaboradores de baixa reciprocidade. Isto significa que a NoF não é capaz de maximizar a satisfação dos colaboradores por permitir que os nós de menor reciprocidade se encontrem equiparados aos nós caroneiros do sistema em determinadas situações. A NoF não atende o Objetivo 1 proposto no modelo de uma política de compartilhamento na Seção 3.2.

Para compreender melhor este cenário foi realizada uma simulação baseada em um modelo de uma cadeia markoviana. A Figura 4.1 mostra como os estados deste modelo foram construídos. Cada elemento do modelo representa o valor de  $Saldo[B]/Saldo[A]$  de dois nós  $A$  e  $B$ , respectivamente, que interagem entre si requisitando apenas um recurso por turno.

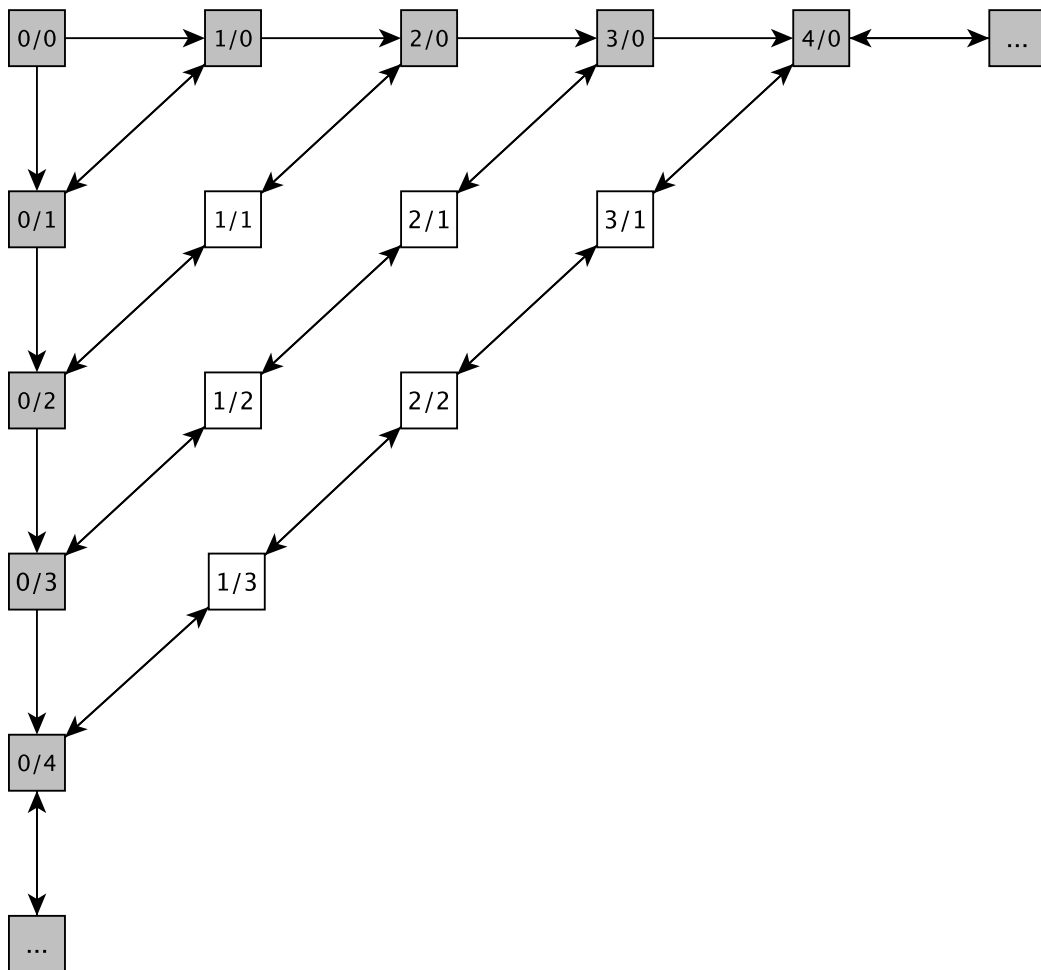


Figura 4.1: Representação dos estados de uma cadeia markoviana de uma relação par-a-par

A probabilidade de transição de estados é definida pelo consumo dos dois nós. Tome como exemplo o cenário onde o nó *A* apresenta 50% de chance de consumir o recurso do nó *B*. Quando *A* não está consumindo recursos, o nó *B* passa então a consumir o recurso do nó *A*. Neste cenário, dado um estado  $x/y$ , há 50% de probabilidade do nó *A* doar um recurso para o nó *B* e assim transitar para o estado  $x/y + 1$ ; e há 50% de chance do nó *B* doar 1 recurso para o nó *A*. Neste segundo caso, a simulação da cadeia markoviana irá do estado  $x/y$  para  $1 + x/y$ .

Quando a cadeia markoviana se encontra em um dos estados cinzas, um dos nós se situa na situação em que, caso esteja doando recursos, o nó que requisitou recursos apresenta a mesma prioridade que um nó caroneiro. Como exemplo, considere o estado 1/0. Nele, o nó *B* apresenta um saldo positivo para o nó *A*. Caso o nó *B* requisite recursos de *A* ao mesmo tempo que um nó caroneiro *C*, o nó *A* privilegia o nó *B*. Entretanto, caso o sistema se encontre neste mesmo estado e o nó *B* está doando recursos e os nós *A* e *C* estão requisitando recursos, o nó *B* é incapaz de diferenciar estes dois nós, mesmo que o nó *A* já tenha realizado uma doação previa ao nó *B*.

Executando uma simulação completa que avalia a possibilidade de encaminhamento total dos estados na cadeia markoviana, é possível identificar qual a probabilidade do sistema entrar nos estados cinzas. Para uma primeira avaliação do comportamento da NoF, utiliza-se um cenário onde as transições entre os estados são definidas com 50% de probabilidade de ocorrência. O comportamento da execução desta simulação completa com 500 turnos para o primeiro cenário é descrito na Figura 4.2. Nela, é possível visualizar a probabilidade do sistema estar em um estado cinza e como esta métrica avança ao longo dos turnos.

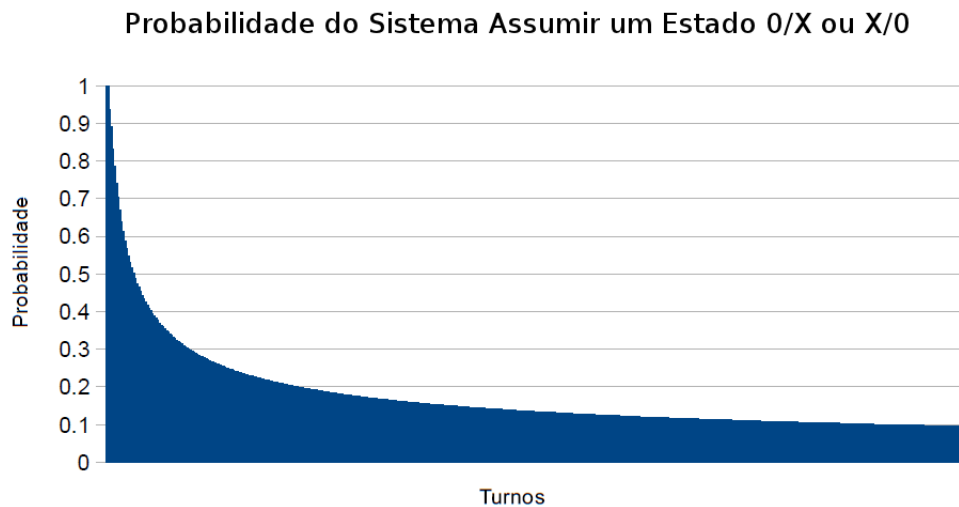


Figura 4.2: Probabilidade do sistema estar em um estado cinza com 50% de probabilidade de transição

Da Figura 4.2 é possível observar uma tendência de queda e, após 500 interações, este valor se encontra em torno de 10%. Com os nós interagindo de forma semelhante, eles tendem a adquirir um valor de conta positivo em ambos os nós na relação entre-pares quanto maior for a interação entre eles.

Uma nova simulação foi executada com novos parâmetros de transições de estado. Nesta nova cadeia, a probabilidade do nó *A* consumir um recurso é de 20%. Quando o nó *A* não está consumindo, o nó *B* passa a consumir recursos. Assim, a transição de, por exemplo,  $x/y$  para  $x + 1/y$  ocorre com 20% de probabilidade e de  $x/y$  para  $x/y + 1$  com 80% de probabilidade. Isto implica em uma menor reciprocidade de *B* para o nó *A*, pois este está consumindo muito mais do que doando para *A*. O resultado desta simulação é exibido na Figura 4.3. Esta figura mostra a probabilidade do sistema estar em um estado cinza ao longo de 500 turnos de simulação.



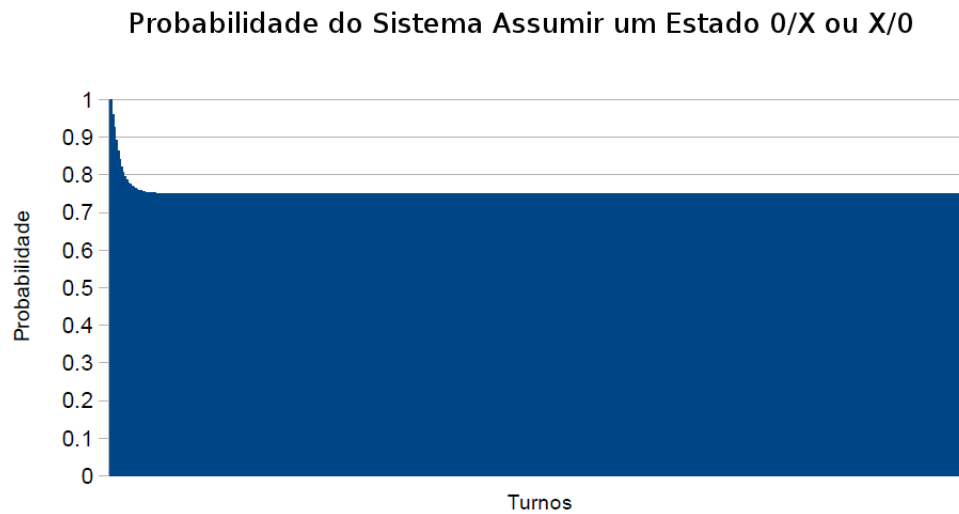


Figura 4.3: Probabilidade do sistema estar em um estado cinza com 20% e 80% de probabilidade de transição

Avaliando a Figura 4.3, a probabilidade de se estar em um estado cinza rapidamente se estabiliza em torno de 75%. Ou seja, os nós tendem a se manterem em uma situação em que, há apenas 25% de probabilidade de um dos nós não estar em uma situação em que sua prioridade é a mesma de um nó caroneiro.

Este último resultado demonstra que, mesmo que um nó tenha contribuído previamente, este pode se colocar em uma situação em que a política não o diferencia de um nó que nunca tenha realizado uma contribuição.

Ao ser incapaz de diferenciar efetivamente certos nós colaboradores de nós que nunca contribuíram ao sistema, também é possível que a NoF esteja sujeita a um ataque de múltiplas identidades. Como a distribuição de recursos para os nós de saldo zero é equalitária e não há custo na criação de identidades de saldo zero, um nó malicioso poderia criar diversas identidades que requisitam recursos a um nó doador. Um nó colaborador, mas de saldo zero, acabará sendo cada vez menos privilegiado pelo sistema na medida que novas identidades são criadas pelo nó malicioso aumentando a concorrência dos recursos de um nó doador. Uma análise em ambiente de simulação avaliando estes possíveis cenários são apresentados na Seção 5.2.

## 4.2 Rede de Colaboradores

Dado as características da grade, os objetivos determinados como ideais de uma política de compartilhamento, e as questões levantadas sobre a NoF, este trabalho apresenta uma nova política para os nós colaboradores: a Rede de Colaboradores, ou NoC.

Na NoC, o nó guarda o quanto ele recebeu e o quanto ele doou para os demais nós da grade. Estes valores são registrados, respectivamente, em dois vetores  $C$  e  $D$  que são armazenados em cada um dos nós e atualizados conforme as interações deste nó com os demais nós do sistema. Ao utilizar estes dois vetores, a política pode atuar na priorização privilegiando também os nós colaboradores que antes se encontravam em situação de saldo zero na NoF.

Ao doar recursos, um nó do sistema está aumentando sua prioridade. Ao consumir recursos, este mesmo nó perde prioridade de recebimento de recursos. A NoC prioriza especialmente os nós de maior retribuição. Em particular, quando o valor de doação de um nó é maior do que o de consumo, este nó é priorizado perante um nó que tenha um menor valor de doação quando comparado ao valor de consumo.

### 4.2.1 Iniciação

A iniciação do sistema na NoC ocorre através da criação dos vetores  $C$  e  $D$ . Estes vetores, por armazenarem a quantidade de recursos obtidos e consumidos dos demais nós são iniciados com zero pois o sistema ainda não realizou interações entre os nós.

### 4.2.2 Fase de Recebimento

É natural imaginar que nesta fase seria necessário apenas a atualização do vetor  $C$  no nó que recebe recursos. No entanto, isto não pode ser feito pois afetaria a robustez da NoC. Como dito anteriormente, a NoC privilegia primeiramente os nós que doaram mais do que consumiram de um nó. O incremento direto do vetor  $C$  incentivaria os nós doadores a uma troca de identidade. Um nó  $B$  que é colaborador mas que estima que, em um determinado nó  $A$ ,  $C[B] < D[B]$ , estaria em melhor situação se, antes da doação para o nó  $A$ , assumisse uma identidade  $C$ . Após a realização de favores, o nó  $A$  passaria a dar uma maior prioridade para a nova identidade  $C$  já que  $C[C] > D[C]$ .

Para incentivar a manutenção de uma identidade, a NoC não faz este incremento direto do vetor  $C$ . No lugar desta atualização, a NoC determina que esta doação deve ignorar o histórico prévio de consumo de um nó doador, caso este esteja em uma situação de desvantagem no sistema. Ou seja, se  $C[B] < D[B]$ : defina novamente  $D[B] = 0$  e  $C[B] = \text{recursos recebidos}$ , caso contrário,  $C[B] = C[B] + \text{recursos recebidos}$ .

Esta estratégia é semelhante à utilizada pela NoF. Na primeira política apresentada, quando um nó consome mais do que doa, este permanece em uma situação de saldo zero até que realize uma doação. Quando uma doação é realizada, o nó que recebe recursos coloca este nó colaborador com o saldo igual à quantidade de recursos imediatamente doados. A NoC segue uma estratégia semelhante e coloca o nó doador, para o nó que recebe recursos, em uma situação positiva imediatamente após o recebimento de uma doação. Em resumo, as atualizações dos vetores  $D$  e  $C$  da política NoC nesta fase são feitas de acordo com o Algoritmo 1.

---

**Algoritmo 1** Atualizando D e C no nó consumidor após doação de P

---

**if**  $C[P] \leq D[P] + \text{recursos recebidos}$  **then**

$D[P] = 0$

$C[P] = \text{recursos recebidos}$

**else**

$C[P] = C[P] + \text{recursos recebidos}$

**end if**

---

Com esta alteração, não há motivação para a troca de identidade de um nó. Em nenhuma ocasião uma nova identidade assumiria um valor maior de prioridade, desincentivando assim o comportamento de troca de identidades.

### 4.2.3 Fase de Doação

Quando um nó doa recursos para um determinado nó  $A$ , ele atualiza o valor de  $D[A]$  incrementando a quantidade de recursos doados, ou seja,  $D[A] = D[A] + \text{recursos doados}$

#### 4.2.4 Fase de Priorização

Para determinar a prioridade de cada nó, a NoC define  $\mathcal{P}(P)$  como a função de priorização. Esta função é calculada toda vez que é preciso decidir como realizar a doação de recursos para um determinado nó  $P$  e determina qual a prioridade de um nó do sistema tem para receber recursos. A função de priorização segue a proposta apresentada; os nós de maior doação são mais privilegiados no sistema e os nós colaboradores, mesmo com baixa reciprocidade, são identificados e priorizados. A função de priorização é definida no Algoritmo 2.

---

##### Algoritmo 2 Função de Priorização

---

```

if  $C[P] > D[P]$  then
    return  $C[P] - D[P]$ 
else if  $C[P] > 0$  then
    return  $\frac{C[P]}{D[P]}$ 
else
    return 0
end if

```

---

A função de priorização apresenta três situações distintas de comportamento quando um determinado nó  $A$  deseja calcular  $\mathcal{P}(B)$  para um nó  $B$ . Caso o nó  $B$  obtenha  $\mathcal{P}(B) > 1$ , isto implica que o nó  $B$  realizou mais doações para o nó  $A$  do que  $A$  realizou para  $B$  no seu histórico de doações. Caso  $0 < \mathcal{P}(B) \leq 1$ , então o nó  $B$  já doou uma determinada quantidade de recursos mas não retribuiu mais do que consumiu. E caso  $\mathcal{P}(B) = 0$ , isto significa que este nó  $B$  nunca doou recursos para o nó  $A$ . A NoC aloca os recursos de forma proporcional primeiramente para o grupo de nós onde  $\mathcal{P}(P) > 1$ . Em seguida, caso haja recursos excedentes, os nós onde  $\mathcal{P}(P) > 0$  são privilegiados. Qualquer recurso que sobrar é distribuído para os demais nós de forma equalitária. Caso haja empate, os recursos são distribuídos de forma equalitária entre os nós.

Para cada um destes grupos de diferentes retribuições, a NoC age de maneira distinta no cálculo da função  $\mathcal{P}$ . Para  $\mathcal{P}(P) > 1$ , é visível o comportamento linear de retribuição; independente da taxa contribuída, ou seja, independente do valor de  $\frac{C[P]}{D[P]}$ , a priorização ocorre pelo saldo gerado. Isto descreve um comportamento linear da função em relação a  $C[P]$  e  $D[P]$ . Entretanto, quando  $0 < \mathcal{P}(P) \leq 1$ , este saldo não é mais relevante, passando a taxa

de  $\frac{D[P]}{C[P]}$  ser o fator determinante desta priorização. É importante entender a escolha destes comportamentos no modelo de priorização.

### **Função de Priorização: Comportamento Linear e Não-Linear**

Quando a função de priorização precisa calcular a prioridade de nós que mais doaram do que consumiram do sistema, segue-se um modelo linear de priorização semelhante ao utilizado pela NoF. Esta estratégia permite que cada nó que se encontra nesta situação seja retribuído devido sua alta reciprocidade do sistema. Ao utilizar o valor de  $C[P] - D[P]$ , um nó estará incentivando os nós de maior reciprocidade, pois estes realizam mais doações do que consumo.

No entanto, na situação em que é preciso diferenciar os nós que consumiram mais do que doaram, a utilização direta de um comportamento linear para a função não incentiva a doação como ocorre anteriormente. Primeiro, dois nós só se encontrarão nesta situação durante o período de tempo em que, após uma doação de  $x$  recursos, pelo menos  $x + 1$  recursos foram consumidos. Neste momento, o nó consumidor entra em débito e é preciso definir um novo mecanismo de priorização.

Considere, por exemplo, dois nós  $A$  e  $B$  que doaram, respectivamente, 10 e 20 recursos para um nó  $C$ . Depois desta doação, ambos consumiram de  $C$  20 e 40 recursos, respectivamente. Ao priorizar  $A$  por apresentar uma menor diferença entre  $C[A]$  e  $D[A]$ , o nó doador não incentiva a doação de recursos, impactando especialmente o nó  $B$ : este estaria em uma melhor situação se fizesse uso de duas identidades, cada uma com os mesmos valores de doação e consumo do que o nó  $A$ .

Assim, a taxa  $\frac{C[P]}{D[P]}$  captura uma doação que é proporcional a reciprocidade nesta situação: ambos os nós são igualmente priorizados na medida que não aumentam seu consumo de forma desproporcional a sua doação. Assim, ambos os nós serão satisfeitos em uma medida proporcional a sua reciprocidade.

#### **4.2.5 Exemplo de Uso**

Para exemplificar o funcionamento da NoC, são considerados três nós:  $A$ ,  $B$  e  $C$ . Inicialmente, o valor de recursos recebidos e doados para todo par de nós é zero, como exibido na

Tabela 4.5.

Tabela 4.5: NoC – Valores iniciais de  $D$ ,  $C$  e  $\mathcal{P}$ 

	$A$	$B$	$C$
$D[A], C[A]$	–	(0, 0)	(0, 0)
$D[B], C[B]$	(0, 0)	–	(0, 0)
$D[C], C[C]$	(0, 0)	(0, 0)	–
$\mathcal{P}(A)$	–	0	0
$\mathcal{P}(B)$	0	–	0
$\mathcal{P}(C)$	0	0	–

Caso o nó  $B$  doe 10 recursos para o nó  $A$  e o nó  $C$  doe 4 recursos para o nó  $A$ , a quantidade de recursos obtidos e recebidos dos nós são atualizadas conforme os valores da Tabela 4.6.

Tabela 4.6: NoC –  $D$ ,  $C$  e  $\mathcal{P}$  após o consumo de recursos pelo nó  $A$ 

	$A$	$B$	$C$
$D[A], C[A]$	–	(0, 10)	(0, 4)
$D[B], C[B]$	(10, 0)	–	(0, 0)
$D[C], C[C]$	(4, 0)	(0, 0)	–
$\mathcal{P}(A)$	–	10	4
$\mathcal{P}(B)$	0	–	0
$\mathcal{P}(C)$	0	0	–

Por enquanto, o modelo de priorização se mantém igual à NoF. Considerando que ambos o nó  $B$  e o nó  $C$  requisitaram 30 recursos cada, e o nó  $A$  tenha 7 recursos disponíveis para doação, eles receberão, assim como no exemplo da NoF, 5 e 2 recursos respectivamente no modelo de alocação proporcional. A atualização dos vetores  $C$  e  $D$ , bem como o novo valor de  $\mathcal{P}$  que cada nó assume após estas doações estão exibidos na Tabela 4.7.

Tabela 4.7: NoC –  $D$ ,  $C$  e  $\mathcal{P}$  após as doações do nó  $A$ 

	$A$	$B$	$C$
$D[A], C[A]$	–	(5, 10)	(2, 4)
$D[B], C[B]$	(10, 5)	–	(0, 0)
$D[C], C[C]$	(4, 2)	(0, 0)	–
$\mathcal{P}(A)$	–	5	2
$\mathcal{P}(B)$	$\frac{5}{10}$	–	0
$\mathcal{P}(C)$	$\frac{2}{4}$	0	–

Observe que a NoC é capaz de capturar os nós colaboradores que se encontram em uma situação onde houve mais consumo do que doações. O nó  $A$  consegue identificar o nó  $B$  como um colaborador. Isto permite que  $A$  distinga o nó  $B$  dos nós caroneiros. Caso o nó  $B$  e um nó caroneiro peçam recursos, a NoC permite privilegiar o nó  $B$ .

### 4.3 Complexidade de Implementação das Políticas

Para permitir a implementação da NoC, não se pode impor um custo excessivo de tempo e memória. Políticas complexas podem ser tornar vulneráveis a ataques de negação-de-serviço pois um nó malicioso pode gerar requisições elaboradas para preencher a memória de um doador, ou ocasionar um atraso de disponibilidade pela exigência de cálculos computacionais complexos.

Em termos de memória e complexidade computacional, a NoF é um mecanismo extremamente simples e eficiente; em cada nó colaborador, o saldo é armazenado apenas para os nós com saldo positivo de favores. Assim, a complexidade de memória é  $O(\mu_1)$ , onde  $\mu_1$  é o número de colaboradores de saldo positivo.

Por outro lado, a NoC precisa receber apenas um recurso de um nó do sistema para manter o registro do mesmo. Este registro é mantido mesmo quando o nó passa, a partir de então, a apenas consumir recursos. Assim, a NoC apresenta uma complexidade de memória de  $O(\mu_2)$ , onde  $\mu_2$  é o número de nós que doaram previamente ao nó. Observe que  $\mu_2 > \mu_1$ .

Em ambos os cenários, o tamanho de memória só é alterado após o consumo de um recurso doado. Não há como realizar um ataque visando o incremento repentino e incontro-

lável de ambas as políticas. Mesmo que um nó não consiga armazenar o registro de todos os demais nós do sistema, ambas as políticas sabem o exato momento em que a memória aumenta no sistema. Assim, seria factível, caso o sistema passe por um momento de contenção pelo excesso de colaboradores em seu registro, eliminar os nós de menor prioridade do registro do sistema.

Em termos computacionais, a complexidade não é novamente impactante. No modelo de alocação direta, o custo computacional é apenas o de calcular o maior elemento de todos os nós mantidos no sistema. Este cálculo é realizado em tempo  $O(n)$  para ambas as políticas, onde  $n$  representa a quantidade de nós requisitando recursos. No modelo de alocação proporcional, o sistema passa antes por uma fase de ordenação dos nós de acordo com sua ordem de prioridade. Isto incorre em complexidade  $O(n * \log(n))$ , com  $n$  limitado pelo número de nós colaboradores. Em ambos os casos esta decisão é local e não faz uso da rede ou de uma entidade remota; isto significa que nenhum atraso na rede ou demora para decisão de um consenso impactam na ordenação e na escolha dos nós a serem privilegiados.

Assim, mesmo em uma grade computacional com um número muito grande de nós e recursos, a implementação de tais políticas é plausível e escalável. Mesmo que estes nós passem por dificuldades de contenção computacional ou de memória, há como diminuir este custo eliminando o registro dos nós de menor prioridade. Esta decisão pode ser feita no momento em que um nó recebe um recurso, que é o momento em que há incremento no tamanho de memória utilizada pelo sistema.

## 4.4 Sumário

A partir da apresentação das políticas de compartilhamento, foi feita uma discussão de como estas políticas satisfazem os objetivos apresentados anteriormente. A NoF apresenta um custo maior para os nós colaboradores de baixa retribuição. A NoC aparece como uma nova política que é capaz de capturar e priorizar os nós colaboradores perante os nós caroneiros.

Ainda, discute-se sobre a praticidade e possibilidade de implementação de tais políticas considerando sua complexidade. Pelas considerações apresentadas, as políticas são de fácil implementação e viáveis do ponto de vista de complexidade espacial e computacional.



# Capítulo 5

## Avaliação

Este capítulo descreve a avaliação das políticas de compartilhamento NoF e NoC perante os objetivos apresentados no Capítulo 3. A Seção 5.1 apresenta a descrição da metodologia utilizada neste trabalho, incluindo o modelo de simulação, as métricas de sucesso para cada política e a construção dos cenários avaliados. Em seguida, os resultados da avaliação são mostrados e discutidos. A última seção apresenta um sumário dos resultados da avaliação apresentada neste capítulo.

### 5.1 Metodologia

Dado os objetivos gerais de uma política de compartilhamento definidos na Seção 3.2, deseja-se avaliar como a NoF e a NoC são capazes de atender tais objetivos. Para isto, observa-se o comportamento das políticas em uma grade simulada de acordo com o modelo de grade apresentado.

Nesta simulação, observa-se o retorno de cada nó colaborador para o sistema observando qual o impacto que cada política apresenta em  $\mathcal{S}(\Delta t)$  e qual a relação entre  $\mathcal{S}(\Delta t)$  e  $\mathcal{R}(\Delta t)$ . O modelo de simulação foi adotado por permitir avaliar a satisfação dos nós controlando a retribuição dos nós em cenários variados.

Nesta seção, também são apresentadas as métricas de sucesso de cada política, ou seja, como cada objetivo é atendido por uma política de acordo com os resultados da simulação. Por fim, é feita uma descrição dos cenários avaliados.

### 5.1.1 Modelo de Simulação

Para avaliar as políticas deste trabalho considerando  $\mathcal{S}(\Delta t)$ , a característica  $\mathcal{R}(\Delta t)$  dos nós e os objetivos propostos, usa-se um modelo de simulação baseado no modelo proposto por Andrade *et al.* [3], o *OurGame*. Trata-se de uma simulação de tempo discreto, onde deseja-se entender o comportamento geral da política em relação às características do comportamento dos nós e do sistema.

No modelo de simulação o tempo é discretizado em turnos ( $t$ ). Cada turno de simulação representa uma unidade de tempo em que os nós interagem consumindo e doando recursos. Assim, ao avaliar um intervalo  $\Delta t$  que compreende vários turnos, é possível observar o ganho do nó no sistema ( $\mathcal{S}(\Delta t)$ ) e a retribuição deste nó ao sistema ( $\mathcal{R}(\Delta t)$ ).

Considerando todo tempo de simulação, há dois tipos gerais de nós: *i*) colaborador,  $\mathcal{R}(\Delta t) > 0$ , e; *ii*) caroneiro: onde  $\mathcal{R}(\Delta t) = 0$ . Na simulação, em nenhum momento o nó caroneiro irá doar recursos ao sistema.

Cada nó tem uma quantidade de recursos ( $r$ ). A cada turno, um nó pode assumir um comportamento de consumo de acordo com a frequência de consumo ( $c$ ) do nó. Esta escolha do comportamento do nó em um turno acontece de forma aleatória. A cada turno, cada nó escolhe um valor aleatório entre 0 e 1 e caso este valor seja menor do que  $c$ , o nó passa este turno tentando consumir recursos. Quando o nó assume este comportamento, ele tenta consumir da grade o mesmo número de recursos que possui. Nos turnos em que um nó não está consumindo recursos, ele estará ocioso, caso seja um nó caroneiro ou estará disponibilizando seus recursos para doação, caso seja um nó colaborador.

É possível estabelecer uma relação direta entre  $\mathcal{R}(\Delta t)$  e  $c$  no sistema. Considerando um número grande de turnos,  $\mathcal{R}(\Delta t) \rightarrow \frac{1-c}{c}$ , pois cada nó tende a requisitar  $c \times r \times \Delta t$  recursos nos turnos que atuar como consumidor e disponibilizar  $(1 - c) \times r \times \Delta t$  recursos nos turnos que se colocar como doador.

Caso o nó se apresente como colaborador, este utiliza um modelo de alocação para determinar como distribuir seus recursos entre os nós consumidores. Dois modelos de alocação foram descritos na Seção 4.1, o modelo de alocação direta e o modelo de alocação proporcional. No modelo de alocação direta, o nó de maior prioridade recebe todos os recursos disponíveis de outro nó. No modelo de alocação proporcional, os recursos de um nó são distribuídos de forma proporcional aos valores de prioridade. Andrade *et al.* [4] discute que

não há impacto na política devido à escolha do modelo, sendo o modelo de alocação proporcional o único utilizado pelo OurGrid e também pelo modelo de simulação usado nesta dissertação.

Os parâmetros do modelo de simulação estão descritos de forma geral na Tabela 5.1.

Tabela 5.1: Parâmetros do modelo de simulação

Parâmetro	Descrição
$t$	Número de turnos
$p$	Política utilizada
$n_d$	Quantidade de nós colaboradores
$n_c$	Quantidade de nós caroneiros
$r$	Quantidade de recursos que cada nó possui
$c$	A probabilidade de um nó, em um determinado turno, assumir o comportamento de consumo

Cada turno é executado atomicamente. Todas as atualizações que poderiam impactar na política de distribuição ou alocação não são alteradas até o final de toda a distribuição de recursos durante um turno. Cada nó armazena apenas a informação parcial inerente ao nó, ou seja, das relações par-a-par construídas até o turno atual da simulação.

Da simulação são extraídos: *i*) os recursos consumidos por cada nó, e; *ii*) os recursos requisitados por cada nó. Devido à quantidade de dados gerados pelo sistema, tais métricas são obtidas a cada 10 turnos de simulação.

### 5.1.2 Métricas de Sucesso

Andrade *et al.* [3] analisam duas métricas de resposta: o fator de escanteamento dos caroneiros, ou seja, o quanto os nós doadores prevalecem sobre os nós caroneiros na forma de recursos obtidos da grade; e o fator de reciprocidade definido pela razão entre quantidade de recursos recebidos sobre a quantidade de recursos doados.

Este trabalho avalia as políticas de compartilhamento através do fator de satisfação,  $\mathcal{S}(\Delta t)$ , obtido pelos nós. O fator de satisfação permite avaliar o objetivo dos nós em reduzir o tempo de execução de sua carga de trabalho. Ao contrário de avaliar a reciprocidade

quantitativa do sistema, ou seja, se um nó recebe a mesma quantidade de recursos que doa, avalia-se quantos recursos um nó consegue obter em relação a quantidade de recursos requisitados ( $\mathcal{S}(\Delta t) = \frac{\text{recursos recebidos}_{\Delta t}}{\text{recursos requisitados}_{\Delta t}}$ ).

Para avaliar o primeiro objetivo proposto, o de maximizar a satisfação dos nós colaboradores, este trabalho simula uma grade e analisa a satisfação dos nós colaboradores na presença de nós caroneiros. Uma política atinge esse objetivo quando é capaz de privilegiar ao máximo os nós colaboradores do sistema, fazendo com que  $\mathcal{S}(\Delta t) \rightarrow 1$  para intervalos de tempo  $\Delta t$  em que existem requisições de recursos dos nós colaboradores.

Em seguida, para avaliar se uma política apresenta a satisfação de um nó proporcional a sua retribuição para o sistema, são construídos cenários avaliando o comportamento de  $\mathcal{S}(\Delta t)$  em relação a nós com diferentes valores de  $\mathcal{R}(\Delta t)$ . Caso  $\mathcal{S}(\Delta t)$  nos nós apresente um comportamento proporcional a  $\mathcal{R}(\Delta t)$ , esse objetivo é atendido.

Avalia-se também a robustez das políticas. Uma política robusta deve oferecer mecanismos para evitar o ataque de múltiplas identidades e de mudança de identidade. Para isto, uma política deve oferecer incentivos à manutenção da identidade e não oferecer ganho na criação de identidades de baixo custo.

Três políticas são comparadas: a NoF, a NoC e uma política Ideal. A política Ideal difere da NoC apenas durante a “Fase de Recebimento”. No lugar de um mecanismo robusto para atualização da contabilidade de recursos recebidos, a política Ideal apenas incrementa o registro que armazena a quantidade de recursos recebidos por um nó, ou seja, ao invés de atualizar  $C[P]$  como indicado no Algoritmo 1, a política Ideal faz simplesmente  $C[P] = C[P] + \text{recursos recebidos}$ . Esta política Ideal serve como referencial para avaliar o impacto do mecanismo de robustez implementado na NoC (Algoritmo 1) e é mais justa na medida que privilegia os colaboradores que realizam mais doações.

### 5.1.3 Cenários de Simulação

O número de turnos ( $t$ ) é definido arbitrariamente alto (10 000) para atingir estabilidade das métricas avaliadas. Para este trabalho, estabilidade foi definida como o momento em que os nós atingem  $\frac{\partial \mathcal{S}(\Delta t)}{\partial t} \approx 0$  com um intervalo de confiança menor do que 0.01 (confiança de 95%), onde  $\Delta t$  é o intervalo de tempo do começo de simulação até o turno em que  $\frac{\partial \mathcal{S}(\Delta t)}{\partial t} \approx 0$ . Como cada métrica se torna estável, o  $\Delta t$  utilizado para a avaliação deste

trabalho representa o tempo total de simulação.

A quantidade de nós colaboradores ( $n_d$ ) é variada em 150, 300 e 600 nós. Estas quantidades de nós representam grades suficientemente grandes e representativas para a avaliação das políticas de compartilhamento.

A frequência de consumo  $c$  foi variada para que se possa avaliar o impacto de  $\mathcal{R}(\Delta t)$  em  $\mathcal{S}(\Delta t)$ . Como há uma relação direta de  $c$  e  $\mathcal{R}$ , os valores de  $c$  foram definidos de forma a avaliar a interação entre os grupos de alta e baixa retribuição. Um nó colaborador com  $\mathcal{R}(\Delta t) > 1$  faz parte do grupo de alta retribuição pois estes nós doam mais recursos do que consomem. Já um nó colaborador com  $\mathcal{R}(\Delta t) < 1$  faz parte do grupo de baixa retribuição, pois consomem mais do que doam ao sistema.

A política de compartilhamento só é capaz de atuar quando há contenção de recursos na grade. Assim, é apenas útil para a avaliação das políticas apresentadas comparar os cenários onde hajam: *i*) apenas nós de baixa retribuição, ou; *ii*) nós de baixa e alta retribuição. Em um cenário onde há apenas nós de alta retribuição, não há contenção na grade e assim todo nó é satisfeito de forma ideal.

Para facilitar a análise do impacto da retribuição na satisfação dos nós em cada política, divide-se igualmente a população de nós colaboradores e caroneiros do sistema em três grupos, cada grupo apresentando o mesmo  $c$ , e assim, o mesmo  $\mathcal{R}(\Delta t)$ . Para avaliar o comportamento das políticas nos cenários de baixa retribuição, definiu-se um grupo de nós com  $c = 0.6$  ( $\mathcal{R}(\Delta t) = 0.66$ ), outro com  $c = 0.75$  ( $\mathcal{R}(\Delta t) = 0.33$ ) e o último com  $c = 0.9$  ( $\mathcal{R}(\Delta t) = 0.11$ ). Para permitir a avaliação da interação de grupos de baixa e alta retribuição, definiu-se um grupo onde  $c = 0.3$  ( $\mathcal{R}(\Delta t) = 2.3$ ) e outros dois grupos: um com  $c = 0.6$  ( $\mathcal{R}(\Delta t) = 0.6$ ) e outro com  $c = 0.9$  ( $\mathcal{R}(\Delta t) = 0.1$ ).

O número de recursos por nó foi definido como  $r = 10$ . Este valor representa a quantidade de recursos que se espera de pequenos laboratórios. Estes são os maiores interessados em se agregar a uma grade aberta entre-pares como o OurGrid.

A quantidade de nós caroneiros no sistema varia de acordo com o objetivo avaliado. Para avaliar o Objetivo 1, é acrescida à grade uma certa quantidade de nós no caroneiros. Inicialmente,  $n_c = 60$ , depois  $n_c = 120$  e, por último,  $n_c = 180$ . Isto permite acompanhar o comportamento da satisfação dos colaboradores à medida que mais caroneiros ingressam no sistema. Para avaliar o Objetivo 2, utiliza-se apenas cenários onde  $n_c = 0$ .

Todos os parâmetros de simulação dos cenários avaliados estão resumidos na Tabela 5.2.

Tabela 5.2: Parâmetros avaliados

Parâmetro	Descrição
$t$	10 000
$p$	NoF, Ideal, NoC
$n_d$	150, 300 e 600
$n_c$	0, 60, 120, 180
$r$	10
$c$	(0.3, 0.6, 0.9) e (0.6, 0.75, 0.9)

## 5.2 Avaliação

Após a descrição dos cenários utilizados nesta avaliação, esta seção apresenta uma discussão sobre os resultados obtidos. Observa-se como os objetivos 1, 2 e 3 são satisfeitos no sistema. Durante a avaliação do desempenho das políticas para cumprir tais objetivos, discute-se também o impacto e os mecanismos de robustez implementados pelas políticas.

Para esta avaliação, as simulações são repetidas de forma que os resultados apresentem um intervalo de confiança para a NoF de 0.01 (com confiança de 95%). Com este mesmo número de repetições, tanto a NoC como a política Ideal apresentam um intervalo de confiança de 0.02 nos resultados apresentados. Isto implica que, há maior variação dentro de cada categoria para estes dois últimos grupos, mas indica também que esta variação não é relativamente alta. A Figura 5.1 mostra o comportamento da satisfação dos nós numa rodada de execução do cenário onde  $n_d = 300$ ,  $n_c = 180$ , e  $c = 0.3$  (cinza claro), 0.6 (cinza) e 0.9 (preto). É utilizado o  $\Delta t$  do turno zero de simulação até o turno atual.

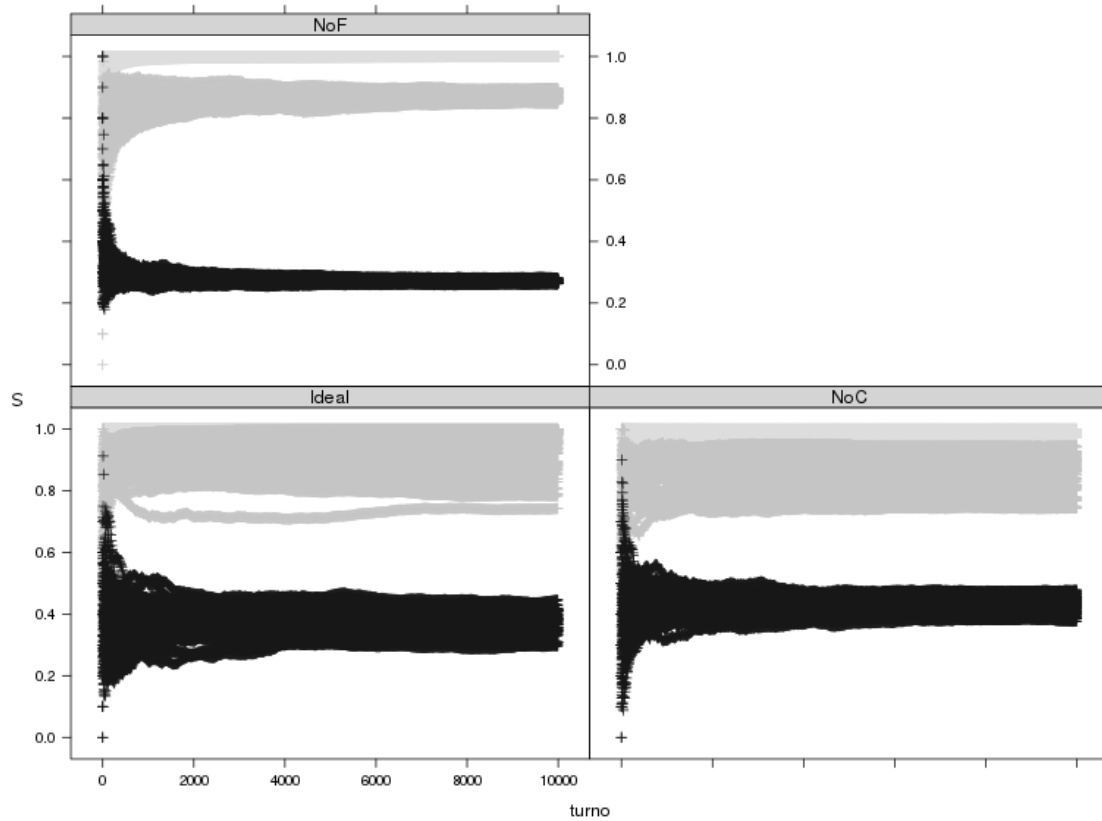


Figura 5.1:  $\mathcal{S}$  dos nós do início da simulação até o turno atual

Pela figura, é possível observar uma maior variação de  $\mathcal{S}(\Delta t)$  para cada fator de retribuição do nó nas políticas NoC e Ideal enquanto a NoF apresenta uma menor variação de  $\mathcal{S}(\Delta t)$ . Observa-se também uma rápida tendência de estabilidade, com os nós se agrupando em torno de um valor médio de  $\mathcal{S}(\Delta t)$  de acordo com seu valor de  $\mathcal{R}(\Delta t)$  com menos de 1 000 turnos de simulação.

A variação de parâmetro  $n_d$  não apresentou alterações nos resultados da avaliação. Uma outra avaliação [20] feita pelos autores deste trabalho utilizando  $n_d = 200$  e com um número maior de turnos corroboram com estes resultados. Para simplificar a apresentação dos resultados obtidos, serão discutidos os resultados dos cenários onde  $n_d = 300$ .

### 5.2.1 Objetivo 1 – Maximizar Satisfação dos Colaboradores

Para avaliar o quanto cada política prioriza os nós colaboradores, foram executados cenários onde nós caroneiros são acrescentados ao sistema e o impacto do fator de satisfação avaliado.

Foram avaliados cenários com  $n_d = 300$ , acrescentando-se 60, 120 e 180 nós caroneiros ao sistema.

Nos cenários onde os nós participantes apresentavam baixa retribuição, ou seja, para  $c = 0.6, 0.75$ , ou  $0.9$ , todas as três políticas apresentaram  $\mathcal{S}(\Delta t) \approx 0$  para todos os nós caroneiros independente da variação de  $n_c$ . Este resultado é justificável. Nesta situação, todos os nós colaboradores do sistema apresentam uma alta frequência de consumo. Na ausência de um grupo de alta retribuição, há uma tendência dos nós colaboradores a doarem recursos e imediatamente consumí-los. Como estes nós colaboradores passam a maior parte do tempo consumindo, qualquer saldo que seja gerado no sistema em um dos nós é utilizado pelos colaboradores do sistema quando estão consumindo recursos.

Entretanto, os cenários onde há a presença de colaboradores com  $\mathcal{R}(\Delta t) > 1$  apresentaram resultados diferentes do anterior para a NoF. Equanto na política Ideal e na política NoC os nós caroneiros obtiveram  $\mathcal{S}(\Delta t) \approx 0$ , com intervalo de confiança de 0.01, a política NoF não foi capaz de obter o mesmo resultado. A Tabela 5.3 apresenta o valor, na NoF, de  $\mathcal{S}(\Delta t)$  obtido pelos nós caroneiro para  $n_d = 300$  e as variações de  $n_c$ . Na tabela a métrica  $FR$  é definida como  $\frac{\text{Total de recursos obtidos pelos caroneiros}}{\text{Total de recursos disponíveis na grade}}$  e serve para ajudar na discussão sobre os resultados apresentados.

Tabela 5.3:  $\mathcal{S}$  e  $FR$  para os nós caroneiros na NoF no cenário onde  $c = (0.3, 0.6, 0.9)$

$n_c$	$\mathcal{S}$	$FR$
60	0.17	5.1%
120	0.13	7.8%
180	0.11	9.9%

Quanto maior o número de caroneiros no sistema, menor é a satisfação destes nós. Entretanto, o aumento do número de nós caroneiros implicou em uma quantidade maior de recursos sendo utilizados por esta categoria de nós. Como há um aumento na quantidade de recursos destinados aos nós caroneiros sem, no entanto, existir um aumento de nós colaboradores no sistema, é natural que ocorra um impacto na satisfação de alguns dos nós colaboradores na NoF na medida que  $n_c$  aumenta. A Tabela 5.3 apresenta  $\mathcal{S}(\Delta t)$  dos nós com  $c = 0.3$ ,  $c = 0.6$  e  $c = 0.9$ .



Tabela 5.4:  $\mathcal{S}$  para os cenários com  $c = (0.3, 0.6, 0.9)$  e com nós caroneiros

$p$	$n_c$	0.3	0.6	0.9
NoF	60	0.99	0.91	0.32
	120	0.99	0.90	0.29
	180	0.99	0.89	0.26
Ideal	60	0.99	0.96	0.35
	120	0.99	0.96	0.35
	180	0.99	0.95	0.36
NoC	60	0.99	0.90	0.40
	120	0.99	0.88	0.41
	180	0.99	0.86	0.42

É possível observar que tanto a NoC como a política Ideal foram eficazes em manter  $\mathcal{S}(\Delta t)$  dentro do intervalo de confiança para todas as variações de  $n_c$ . Este resultado é esperado já que nestas políticas, para os nós caroneiros,  $\mathcal{S}(\Delta t) \approx 0$ . Já a NoF apresentou uma degradação estatisticamente significativa de  $\mathcal{S}(\Delta t)$  com o aumento de  $n_c$  para os nós colaboradores para os nós de menor retribuição ( $c = 0.9$ ). Este resultado está exibido na Figura 5.2. Pela figura, é possível visualizar como o aumento de nós caroneiros impacta nos nós de menor retribuição.

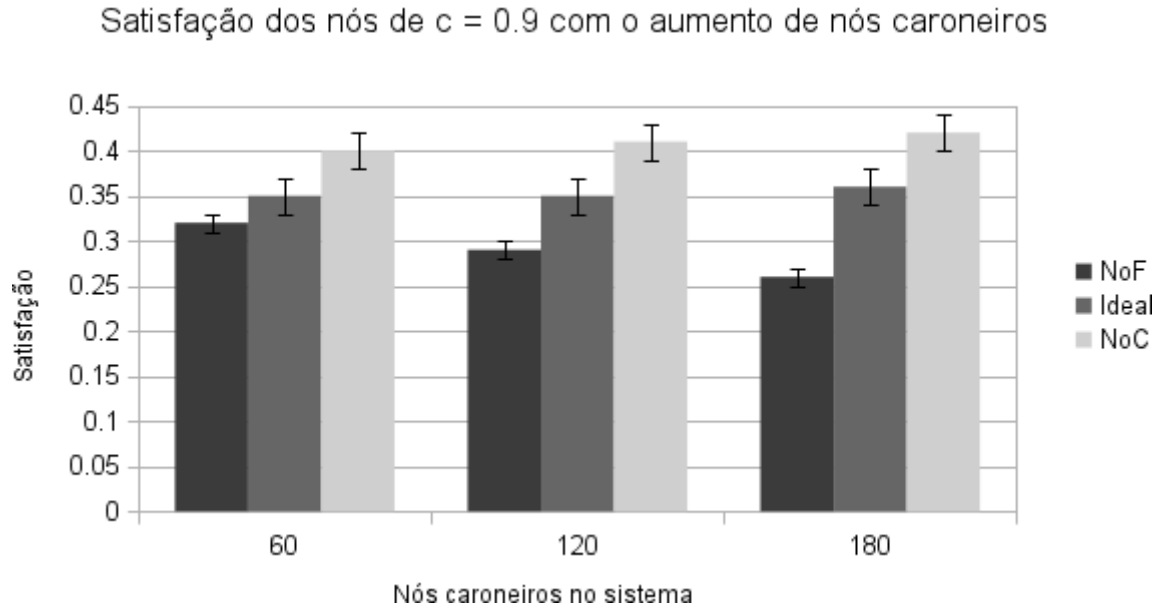


Figura 5.2:  $\mathcal{S}$  para os nós com  $c = 0.9$

Ao contrário do cenários sem nós de alta retribuição, nestes cenários há nós disponibilizando mais recursos do que consumindo. Na NoF, quando um nó de alta retribuição interage com um nó de baixa retribuição, este último tende a zerar rapidamente o seu saldo. Quando isto ocorre, este nó colaborador fica equiparado ao nó caroneiro. Ao aumentar o número de nós caroneiros no sistema, a chance deste nó obter recursos de um nó de alta retribuição diminui pois há cada vez mais nós caroneiros concorrendo por recursos.

### 5.2.2 Objetivo 2 – Tornar a satisfação do nó colaborador proporcional a sua retribuição

Com o intuito de avaliar como a satisfação e retribuição estão relacionadas nas diferentes políticas, foram simulados dois cenários: um com a presença de nós colaboradores de baixa e alta retribuição; e outro cenário onde apenas nós de baixa retribuição atuaram no sistema.

No cenário onde todos os nós tem  $\mathcal{R}(\Delta t) < 1$ , usando as políticas NoF e Ideal, os nós apresentam  $\mathcal{S}(\Delta t)$  similares, considerando o intervalo de confiança de cada categoria. Na NoC, os nós colaboradores apresentaram uma diferença em  $\mathcal{S}(\Delta t)$  estatisticamente significativa quando comparado à NoF ou à política Ideal. Estes resultados estão apresentados na Figura 5.3.

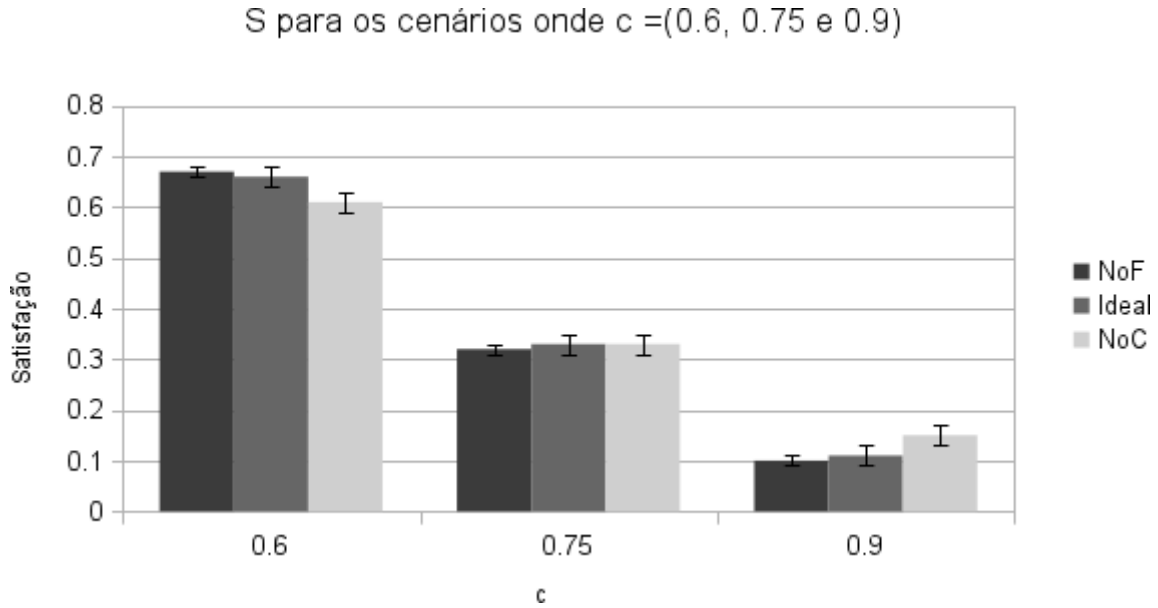


Figura 5.3:  $S$  para os cenários onde  $c = (0.6, 0.75, 0.9)$

Neste cenário, na NoF, como todos os nós consomem mais do que doam, há uma tendência de que cada nó seja satisfeito de maneira diretamente proporcional a sua retribuição. Isto ocorre pois cada nó do sistema doa  $(1 - c) \times r$  recursos por turno quando assume o comportamento de doação. Quando assume o comportamento de consumo, cada nó requisita  $c \times r$  recursos do sistema. Como  $(1 - c) < c$ , o valor médio de satisfação de um nó do sistema é o saldo que este foi capaz de obter como doador sobre a quantidade de recursos que este nó requisitou. Como estes nós não conseguem acumular um saldo, nem há a presença de um nó que doe mais do que consome do sistema, probabilisticamente, ao avaliar o valor de satisfação no tempo de vida do nó,  $\mathcal{S}(\Delta t) = \frac{(1-c) \times r}{c \times r} = \mathcal{R}(\Delta t)$ .

A política Ideal também é capaz de satisfazer cada nó proporcionalmente de acordo com seu fator de retribuição. Como em todos os nós desta grade,  $0 < \mathcal{R}(\Delta t) \leq 1$ , ou seja, todos consomem mais recursos do que doam, a priorização de um determinado nó oscilará em torno de 1 para todos os nós do sistema de forma que a quantidade de recursos recebidos seja igual à quantidade de recursos doados por cada nó. Como a quantidade de recursos doados por cada nó é  $(1 - c) \times r$  ao longo do tempo de vida de um nó, e a quantidade de recursos requisitados é  $c \times r$ , conclui-se, assim como na avaliação deste cenário feita para a NoF, que  $\mathcal{S}(\Delta t) = \frac{(1-c) \times r}{c \times r} = \mathcal{R}(\Delta t)$ .

Entretanto, na NoC, a satisfação dos nós foi diferente para os nós da maior e de menor

retribuição. Esta diferença ocorre pelo mecanismo de robustez implementado pela NoC quando comparado com a política Ideal. No entanto, os valores de satisfação ainda são proporcionais à retribuição do nó, condição satisfeita para atender o Objetivo 2 neste cenário.

Ao considerar o cenário onde nós com alta e baixa retribuição interagem entre si, constatou-se uma diferença dos resultados de satisfação para as três políticas comparadas. Os valores de satisfação para os nós de  $c = 0.3$ ,  $c = 0.6$  e  $c = 0.9$  estão apresentados na Figura 5.4.

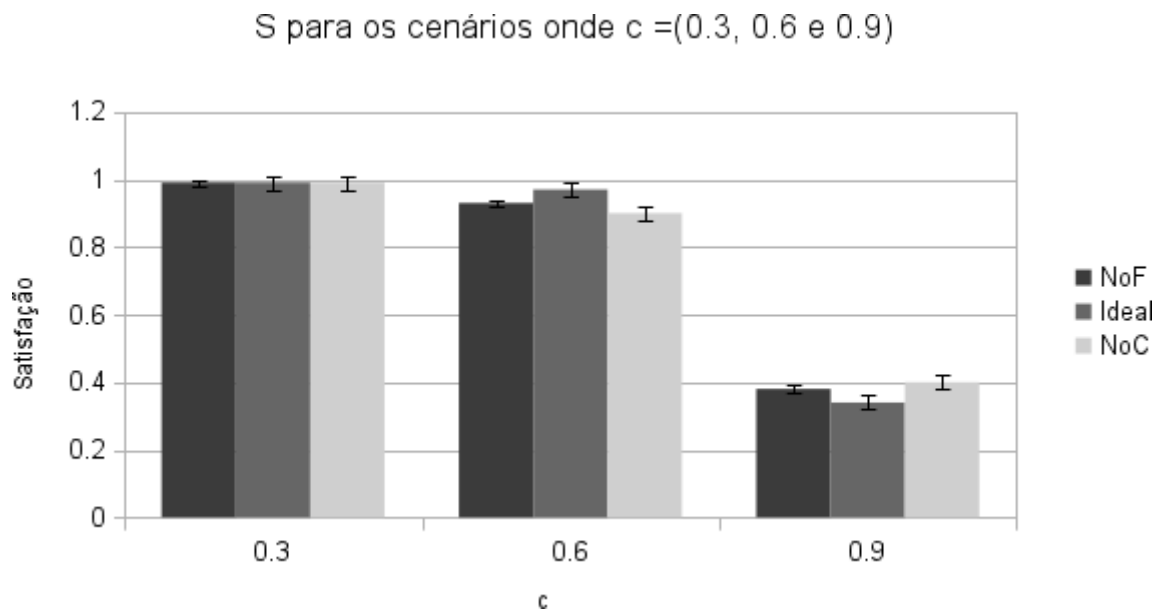


Figura 5.4:  $S$  para os cenários onde  $c = (0.3, 0.6, 0.9)$

Com a presença de um nó colaborador que doa mais do que consome, os nós de menor retribuição passam a utilizar tais recursos excedentes. Todos os nós de maior retribuição são praticamente satisfeitos de forma ideal. Para os demais parâmetros, há uma grande diferença entre os resultados da política Ideal perante as demais políticas. Como a política Ideal é capaz de armazenar a retribuição dos nós, os recursos excedentes dos nós onde  $c = 0.3$  tendem a ser completamente doados para os nós com segundo maior privilégio no sistema, ou seja, os nós onde  $c = 0.6$ . Os recursos que não forem consumidos pelos nós onde  $c = 0.6$  são então consumidos pelos nós de mais baixa retribuição.

Na NoF e na NoC, os mecanismos de robustez do sistema fazem com que a doação excedente dos nós de alta retribuição não seja repassado completamente para nós os onde  $c = 0.6$ . Estes nós, devido ao seu consumo, acabam gerando situações onde, na NoF, o saldo

se torna nulo assim como em alguns nós onde  $c = 0.9$ . A NoC acaba também priorizando os nós onde  $c = 0.9$  por permitir que o histórico de doação e consumo sejam reiniciados quando o nó desta categoria realiza uma doação. Ainda assim, a satisfação dos nós se mantém proporcional à retribuição dos recursos.

### 5.2.3 Objetivo 3 – Ser robusta contra os ataques de múltiplas identidades e de mudança de identidade

Todas as três políticas incentivam a manutenção da identidade dos nós do sistema. Se um nó tiver doado mais do que consumido, não há vantagem na troca de identidade pois há uma maior satisfação para o nó que apresentar esta característica. Se um nó consome mais do que doa, também não há vantagem na troca de identidade pois o nó assumirá uma situação pior ou igual a que possui.

Durante a avaliação do Objetivo 1, nos cenários onde há presença de nós caroneiros e há também nós colaboradores com  $\mathcal{R}(\Delta t) > 1$ , foi possível observar o aumento do número de recursos destinados aos nós caroneiros. A primeira implicação deste resultado é de que a NoF não é capaz de satisfazer plenamente os nós colaboradores. No entanto, outra implicação deste resultado é que pode ser vantajoso a um nó inundar a grade com múltiplas identidades que requisitem recursos na NoF. Um nó malicioso pode fazer uso de múltiplas identidades que não colaboram para o sistema pois há um possível ganho no sistema com o uso destas identidades.

Ao criar múltiplas identidades e inundar os demais nós com requisições por recursos, um nó malicioso pode acabar se tornando mais satisfeito do que caso colaborasse com o sistema. Considere um nó malicioso  $M$  com  $c = 0.9$ . Num cenário onde não há nós caroneiros, como o exibido na Figura 5.4, este nó espera obter, caso colaborasse com o sistema,  $\mathcal{S}(\Delta t) = 0.38$ . Se, no mesmo cenário, este nó não colaborar com o sistema e buscar obter recursos através do controle de múltiplas identidades, este pode obter  $\mathcal{S}(\Delta t) > 0.38$ . Considere o mesmo cenário, onde  $n_d = 300$  e os nós assumem  $c = 0.3$ ,  $c = 0.6$  e  $c = 0.9$ . O nó  $M$  passa a criar várias identidades que não irão contribuir com o sistema, mas apenas requisitar recursos. É possível acompanhar o aumento de  $\mathcal{S}(\Delta t)$  do nó  $M$  na medida que o número de identidades criadas por este nó aumenta. O resultado desta simulação está apresentado na Figura 5.5.

S do nó malicioso  $M$  de acordo com o aumento do número de múltiplas identidades

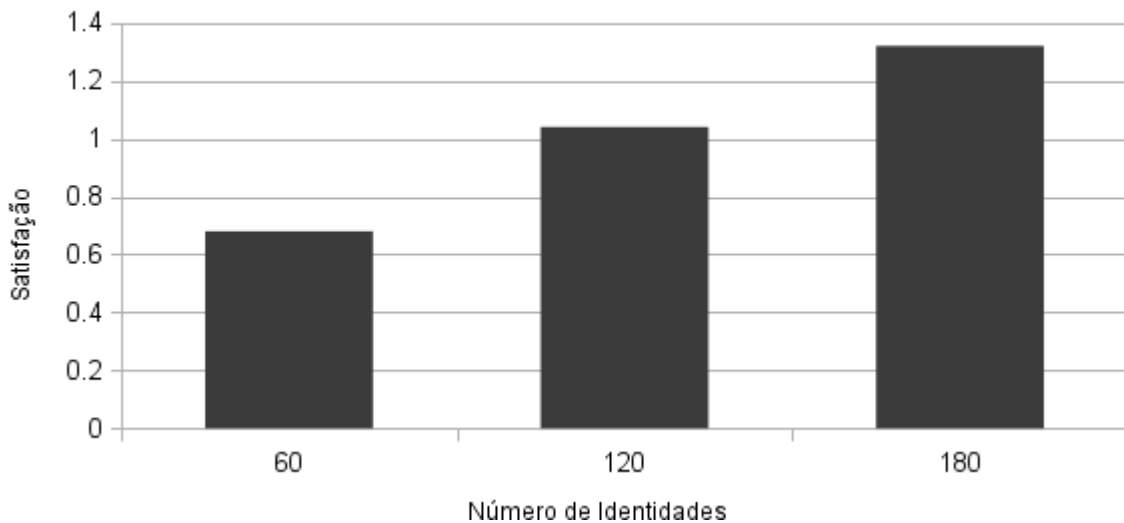


Figura 5.5:  $S$  do nó malicioso  $M$  controlando múltiplas identidades

Observe que  $S(\Delta t) > 1$  simboliza que o nó seria capaz de receber mais do que necessita. Tanto a NoC como a política Ideal não beneficiam a criação de uma identidade sem custo como a do nó caroneiro. Nestas políticas, a satisfação dessas múltiplas identidades se mantêm nula, e assim, um nó malicioso não pode obter um ganho maior com este ataque. A NoF, ao contrário, apresenta um cenário onde o uso de múltiplas identidades sem nunca colaborar com o sistema aumenta a satisfação de um nó, mais do que o ganho obtido por um colaborador da mesma categoria. Desta forma, a NoF falha em satisfazer o Objetivo 3 por não ser robusta contra este tipo de manipulação.

### 5.3 Sumário

Durante a avaliação do desempenho das políticas em atender o objetivo de maximizar a satisfação dos nós colaboradores, foi possível observar na NoF um cenário em que a adição de nós caroneiros impacta negativamente na satisfação dos nós. Ainda, o aumento do número de nós caroneiros gera uma diminuição cada vez maior na satisfação dos nós, especialmente aqueles de baixa retribuição. A política Ideal e a NoC conseguem diferenciar os nós colaboradores dos nós caroneiros por completo, incentivando os nós a realizarem uma retribuição ao sistema, mesmo que pequena.

Ao avaliar se as políticas de compartilhamento satisfazem os nós de forma proporcional a sua retribuição, todos os nós conseguiram atingir este objetivo. Durante esta avaliação, foi possível avaliar o impacto do mecanismo de robustez da NoF e NoC quando comparadas a uma política Ideal. O mecanismo de robustez da NoC não gera um resultado diferente daquele obtido na NoF e, ao contrário da política Ideal, acaba por priorizar mais os nós de baixa retribuição. Isto acontece mantendo proporcional a relação entre satisfação e retribuição.

Quanto a robustez do sistema, todas as políticas são efetivas ao proteger o sistema de *whitewashing*. Entretanto, a NoF ainda permite com que a criação de múltiplas identidades privilegie um nó no sistema. Isto ocorre porque a NoF não é capaz de identificar alguns nós que colaboraram previamente ao sistema, o que representa um custo na manutenção de uma identidade. Ao ignorar estas contribuições, um nó malicioso pode obter vantagens do sistema com a criação de múltiplas identidades. A NoC e a política Ideal fazem com que toda contribuição seja privilegiada. Desta forma, a criação de múltiplas identidades não contribuem para a satisfação de um nó malicioso.

# Capítulo 6

## Conclusões e Trabalhos Futuros

Toda grade que deseja compartilhar recursos entre diferentes domínios administrativos deve seguir uma política de compartilhamento. Tais políticas de compartilhamento devem levar em consideração as características dos nós e da grade.

Em um modelo de grade computacional aberta entre-pares, os nós buscam obter rapidamente os recursos que requisitam ao sistema. A grade precisa oferecer incentivos à doação de recursos pelos nós, para que mais recursos se tornem disponíveis no sistema. Assim, uma política deve apresentar dois objetivos básicos: aumentar a satisfação dos nós colaboradores e incentivar que os nós retribuam ao sistema. Também é objetivo de uma política ser robusta a ponto de evitar que nós façam uso de mecanismos do sistema para obter mais vantagens do mesmo.

A NoF [4] é uma política que busca incentivar a doação de recursos através de um modelo de troca de favores. Neste trabalho, a NoF foi avaliada pela: *i*) sua capacidade de maximizar o comportamento de colaboração dos nós; *ii*) a capacidade satisfazer nós de acordo com a retribuição dos mesmos, e; *iii*) ser robusta contra ataques de manipulação de identidade.

Enquanto a NoF é capaz de satisfazer os nós de acordo com a retribuição destes, a NoF não atende bem os outros demais objetivos. Há cenários onde o comportamento de carona é incentivado e o uso de múltiplas identidades é vantajoso para nós caroneiros.

Este trabalho apresenta uma política de compartilhamento de recursos, a NoC. Nesta política, os nós colaboradores são inteiramente privilegiados perante os nós caroneiros. Ainda, a NoC oferece incentivos à doação na medida que é capaz de satisfazer os nós do sistema de acordo com a retribuição dos mesmos e protege o sistema do ataque de múltiplas identida-



des, impedindo que os nós colaboradores sejam penalizados por tal estratégia maliciosa e de baixo custo.

Parte da pesquisa deste trabalho envolveu discussões que não cabem diretamente aqui neste trabalho, mas que podem ser de interesse pro pesquisador da área. Tais discussões, bem como o código-fonte das ferramentas utilizadas neste trabalho, encontram-se disponíveis em: <http://redmine.lsd.ufcg.edu.br/projects/matheusgr>.

## 6.1 Trabalhos Futuros

Nesta seção são apresentados possíveis trabalhos futuros que servem de continuidade para este trabalho. Discuti-se a possibilidade de outro tipo de ataque a NoC e como novas pesquisas podem caminhar na direção de tornar tal política robusta. Outro trabalho a ser considerado é a presença de nós voluntários e como gerar uma política justa para a distribuição de recursos nestes nós voluntários.

### 6.1.1 Ataques Contra a NoC

Neste trabalho, explorou-se dois tipos de ataques contra a política NoC. Durante a apresentação deste trabalho, um possível tipo de ataque foi detectado e é um trabalho futuro a ser realizado uma discussão aprofundada sobre impacto e viabilidade de obter vantagens do sistema com este ataque.

Neste ataque, um nó que não tem recursos (caroneiro) passa a atuar como um pseudo-colaborador. Este nó, quando ocioso, busca consumir recursos de um nó colaborador e realizar a doação destes recursos para outro nó colaborador que requisite recursos. Com o uso deste artifício, tal nó pseudo-colaborador ganha uma vantagem inicial perante o nó colaborador que estiver consumindo recursos.

Primeiramente, é preciso analisar a viabilidade de tal ataque. Dentro do modelo apresentado, o nó caroneiro ao atuar como pseudo-colaborador gera um aumento de recursos disponíveis na grade, mas cabe ao nó colaborador que está doando recursos decidir para que nó fazer a doação. Para que o nó pseudo-colaborador possa receber recursos para realizar este ataque, este precisa ter uma melhor prioridade do que, inclusive o próprio consumidor do nó pseudo-colaborador. É importante lembrar que o nó colaborador consumidor e o nó

pseudo-colaborador estão ambos concorrendo pelo mesmo recurso do nó colaborador que está ofertando recursos.

Uma primeira etapa para avaliar a viabilidade deste ataque é executar uma nova avaliação com a presença destes nós detectando se estes obteriam mais vantagens do que os nós colaboradores do sistema.

### 6.1.2 Uma Política para Nós Voluntários

A continuidade deste trabalho também pode ser realizada através de uma análise de novos tipos de comportamentos de nós no sistema, como os nós da computação voluntária [25]. Sistemas como o BOINC [2] e o zeroHUB [22] permitem a construção de grades computacionais onde a doação de recursos é livre e feita de forma voluntária. Os nós desse sistema não apresentam comportamento egoísta e carga de trabalho a ser realizada.

As políticas abordadas como a NoF e NoC não determinam uma política própria para nós voluntários. Nessas políticas, o nó doara equalitariamente seus recursos entre todos os nós que requisitarem por recursos. É de interesse definir como estes voluntários poderiam compartilhar seus recursos ao sistema, especialmente considerando mecanismos autonômicos e permitindo que a contribuição desses nós ajude a aumentar o incentivo à doação de recursos para o sistema.

Apresenta-se uma proposta para uma política que pode ser explorada em trabalhos futuros. Tal política busca doar recursos analisando o histórico prévio de consumo dos demais nós do sistema em um nó voluntário. Basicamente, esta política contabilizaria apenas a quantidade de recursos consumidos pelos seus participantes. Desta forma, ao contrário da NoC e NoF, um nó voluntário armazena apenas um vetor  $C$  representando a quantidade de recursos consumidos do nó voluntário para os demais nós do sistema. Um determinado nó  $B$ , consumidor da grade, apresenta inicialmente  $C[B] = 0$  no nó voluntário.

Quando um recurso é consumido deste nó voluntário por um determinado nó  $B$ ,  $C[B]$  é atualizado decrementando o atual valor de  $C[B]$  pelo número de recursos doados pelo nó voluntário. Este nó voluntário armazena também o valor *MaiorDebito* que representa o maior valor dentro do vetor  $C$ , ou seja, o de maior débito dentre todos os consumidores, a cada turno.

Quando o nó voluntário interage pela primeira vez com um nó consumidor,  $C[B]$  recebe

o valor *MaiorDebito*. Isto permite com que, caso um nó queira entrar com uma nova identidade no sistema, sua situação será a do pior consumidor anteriormente. Isto representa um desincentivo a troca de identidade pois, ao possuir uma nova identidade, o nó consumidor se encontrará sempre em situação igual ou pior que tinha anteriormente.

Definido como  $C$  é atualizado nesta nova política, é preciso agora imaginar como poderia ser feita a distribuição de recursos através do mecanismo de priorização. Caso não haja contenção na grade, o nó consumidor irá receber os recursos do nó voluntário independente do seu valor em  $C$ . Este mecanismo de priorização define um valor de prioridade ( $\mathcal{P}$ ) de acordo com o valor de consumo de cada nó que está requisitando recursos.

O valor de  $\mathcal{P}$  é calculado, para um nó  $B$ , como  $\mathcal{P} = \text{MaiorDebito} - C[B]$ . Ao fazer isto, um nó  $B$  terá uma prioridade maior o quanto menor for  $C[B]$ , ou seja, quanto menor este tiver consumido do nó voluntário.

Ao definir este valor de  $\mathcal{P}$  as políticas de alocação direta e alocação proporcional são válidas para este nó voluntário: utilizando a prioridade, o nó voluntário pode alocar todos os seus recursos para o nó de maior prioridade, ou alocar de forma proporcional ao consumo de todos os nós da grade.

# Bibliografia

- [1] Ourgrid status. Disponível em: <http://status.ourgrid.org/>.
- [2] David P. Anderson. Boinc: A system for public-resource computing and storage. In *GRID '04: Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.
- [3] Nazareno Andrade, Francisco Brasileiro, Walfredo Cirne, and Miranda Mowbray. Automatic grid assembly by promoting collaboration in peer-to-peer grids. *Journal of Parallel and Distributed Computing*, 67:957 – 966, 2007.
- [4] Nazareno Andrade, Walfredo Cirne, Francisco Brasileiro, and Paulo Roisenberg. Ourgrid: An approach to easily assemble grids with equitable resource sharing. In *9th Workshop on Job Scheduling Strategies for Parallel Processing (JSSPP)*, Jun 2003.
- [5] Rida A. Bazzi and Goran Konjevod. On the establishment of distinct identities in overlay networks. In *Proceedings of the twenty-fourth annual ACM symposium on Principles of distributed computing*, PODC '05, pages 312–320, New York, NY, USA, 2005. ACM.
- [6] A. R. Bharambe, C. Herley, and V. N. Padmanabhan. Analyzing and Improving BitTorrent Performance. Technical report, Microsoft Research.
- [7] Francisco Brasileiro, Eliane Araujo, William Voorsluys, Milena Oliveira, and Flavio Figueiredo. Bridging the high performance computing gap: the ourgrid experience. In *CCGRID '07: Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid*, pages 817–822, Washington, DC, USA, 2007. IEEE Computer Society.

- [8] Francisco Brasileiro, Matheus Gaudencio, Rafael Silva, Alexandre Duarte, Diego Carvalho, Diego Scardaci, Leandro Ciuffo, Rafael Mayo, Herbert Hoeger, Michael Stanton, Raul Ramos, Roberto Barbera, Bernard Marechal, and Philippe Gavillet. Using a simple prioritisation mechanism to effectively interoperate service and opportunistic grids in the eela-2 e-infrastructure. *Journal of Grid Computing*, pages 1–17, 2011. 10.1007/s10723-010-9177-5.
- [9] Cyril Briquet. *Systematic Cooperation in P2P Grids*. PhD thesis, Université of Liège, 2008.
- [10] Walfredo Cirne, Francisco Brasileiro, Nazareno Andrade, Lauro Costa, Alisson Andrade, Reynaldo Novaes, and Miranda Mowbray. Labs of the world, unite!!! *Journal of Grid Computing*, 4(3):225–246, 2006.
- [11] Cristiano Costa and Jussara Almeida. Reputation systems for fighting pollution in peer-to-peer file sharing systems. *Peer-to-Peer Computing, IEEE International Conference on*, 0:53–60, 2007.
- [12] Flavio de Figueiredo, Matheus Gaudencio, Thiago Cunha, Rodrigo Miranda, and Francisco Brasileiro. Um portfolio de segurança para um sistema de grade entre pares de livre entrada. In *VI Workshop on Grid Computing and Applications (WCGA)*, Rio de Janeiro, RJ, May 2008.
- [13] John Douceur. The sybil attack. In Peter Druschel, Frans Kaashoek, and Antony Rowstron, editors, *Peer-to-Peer Systems*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer Berlin // Heidelberg, 2002.
- [14] D. H. J. Epema, M. Livny, R. vanDantzig, X. Evers, and J. Pruyne. A worldwide flock of condors : Load sharing among workstation clusters. *FGCS. Future generations computer systems*, 12(1):53–65, 1996.
- [15] Gilles Fedak, Haiwu He, Oleg Lodygensky, Zoltan Balaton, Zoltan Farkas, Gabor Gombas, Peter Kacsuk, Robert Lovas, Attila Csaba Marosi, Ian Kelley, Ian Taylor, Gabor Terstyanszky, Tamas Kiss, Miguel Cardenas-Montes, Ad Emmen, and Filipe

- Araujo. Edges: A bridge between desktop grids and service grids. *ChinaGrid, Annual Conference*, 0:3–9, 2008.
- [16] Ian Foster. What is the Grid? - a three point checklist. *GRIDtoday*, 1(6), July 2002.
- [17] Ian Foster, Carl Kesselman, and Steven Tuecke. The anatomy of the grid - enabling scalable virtual organizations. *International Journal of Supercomputer Applications*, 15:2001, 2001.
- [18] James Frey, Todd Tannenbaum, Miron Livny, Ian Foster, and Steven Tuecke. Condor-g: A computation management agent for multi-institutional grids. *High-Performance Distributed Computing, International Symposium on*, 0:0055, 2001.
- [19] Eric J. Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, 2000.
- [20] Matheus Gaudencio and Francisco Brasileiro. Uma nova política de incentivos para grades computacionais entre-pares. In *Anais do XXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, Campo Grande, Brasil, may 2011. SBC.
- [21] Alexandru Iosup, Hui Li, Mathieu Jan, Shanny Anoep, Catalin Dumitrescu, Lex Wolters, and Dick H.J. Epema. The grid workloads archive. *Future Generation Computer Systems*, 24(7):672 – 686, 2008.
- [22] Gerhard Klimeck, Michael McLennan, Sean P. Brophy, George B. Adams III, and Mark S. Lundstrom. nanohub.org: Advancing education and research in nanotechnology. *Computing in Science and Engg.*, 10(5):17–23, 2008.
- [23] M. Meulpolder, L. D’Acunto, M. Capotă, M. Wojciechowski, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. Public and private bittorrent communities: a measurement study. In *Proceedings of the 9th international conference on Peer-to-peer systems, IPTPS’10*, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.
- [24] Nelson Nóbrega-Júnior, Leonardo Assis, and Francisco Brasileiro. Scheduling cpu-intensive grid applications using partial information. In *Proceedings of the 2008 37th International Conference on Parallel Processing, ICPP ’08*, pages 262–269, Washington, DC, USA, 2008. IEEE Computer Society.

- 
- [25] Luis F. G. Sarmenta. *Volunteer Computing*. PhD thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY, Jun 2001.
- [26] Douglas Thain, Todd Tannenbaum, and Miron Livny. Distributed computing in practice: The condor experience. *Concurrency and Computation: Practice and Experience*, 17:2–4, 2005.
- [27] Kevin Walsh and Emin Gün Sirer. Fighting peer-to-peer spam and decoys with object reputation. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, P2PECON '05, pages 138–143, New York, NY, USA, 2005. ACM.