Um Mecanismo Eficiente de Confiança para a Detecção e Punição de Usuários Maliciosos em Grades Peer-to-peer

Igor A. Chaves [1], Reinaldo B. Braga [1], Rossana M. C. Andrade [1], José N. de Souza [1] e Bruno Schulze [2] *

[1]Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Universidade Federal do Ceará (UFC)

[2]Laboratório Nacional de Computação Científica (LNCC)

{igor,reinaldo,rossana,neuman}@great.ufc.br,schulze@lncc.br

Abstract. *Peer-to-Peer grid environments demand the correct execution of tasks to guarantee good performance. However, in these environments, there are malicious users that affect the grid performance. These users modify the tasks results and even cheat security mechanisms. Thus, it is necessary an effective mechanism able to detect and punish these malicious users. This paper presents then a reactive trust-based security mechanism that aims at detecting and punishing malicious users who corrupt the tasks results of a P2P grid. These results are obtained with simulations of grid P2P environments. We also present the results analysis that shows the proposed mechanism is efficient and manages to detect and punish 100 % of the stations that modify the tasks results of the P2P grid.*

Resumo. *A correta execução das tarefas em ambientes de grades peer-to-peer (P2P) é fundamental para o seu bom desempenho. Entretanto, nestes ambientes, existem usuários maliciosos que prejudicam o desempenho da grade modificando os resultados das tarefas e, até mesmo, burlando os mecanismos de segurança. Desta forma, faz-se necessário um mecanismo eficiente capaz de detectar e punir os usuários maliciosos da grade P2P. Este trabalho apresenta um mecanismo reativo de segurança baseado em confiança, tendo como objetivo principal detectar e punir os usuários maliciosos que corrompem o resultado das tarefas da grade P2P. Os resultados apresentados foram obtidos através de simulações de ambientes de grades P2P. Ao analisar os resultados, é possível concluir que o mecanismo proposto é eficiente, detectando e punindo até 100% as estações que modificam os resultados das tarefas da grade P2P.*

1. Introdução

As grades computacionais são formadas a partir de recursos computacionais heterogêneos e distribuídos geograficamente, que possibilitam a criação de um ambiente com alto poder de processamento e de armazenamento [Foster e Kesselman, 2004]. O ambiente de grades fornece recursos para diversas aplicações, tais como armazenamento, análise e virtualização de dados. As grades *peer-to-peer* (P2P) se caracterizam por sua infraestrutura descentralizada, o que aumenta a escalabilidade quando comparada com as grades convencionais [Marsh et al., 2008]. Elas também se caracterizam pelo fato de

*Este trabalho foi realizado com recursos do CNPq (projeto SIMEGRID)

os usuários, além de doarem seus recursos, também poderem executar suas aplicações na grade P2P [Uppuluri et al., 2005]. Visto que a grade P2P pode ser composta por máquinas heterogêneas, localizadas em diferentes domínios de redes e com variados tipos de usuários, é possível, então, que existam usuários maliciosos, cujo comportamento pode prejudicar o desempenho da grade. Um dos exemplos mais conhecidos é o caso do SETI@home [Seti@Home, 2009], no qual voluntários alteravam sua quantidade de trabalhos realizados com o objetivo de aumentar sua reputação no *ranking* de maiores colaboradores do projeto SETI@home. Outro exemplo bastante conhecido é o dos usuários maliciosos que modificam os resultados das tarefas da grade. Estes usuários têm como objetivo principal enviar uma resposta aleatória para as estações da grade, pois isso faz com que a estação maliciosa não gaste os seus recursos processando uma tarefa e não seja excluído da grade, por estar respondendo-a. Já outros visam apenas prejudicar o funcionamento da grade.

Diversas soluções já foram propostas com o objetivo de aumentar a precisão dos resultados das tarefas executadas pela grade. Um dos exemplos é a votação majoritária, na qual uma tarefa é replicada e enviada para vários usuários executarem e, de acordo com as respostas recebidas, o usuário que enviou a tarefa decide se o resultado desta é aceito ou não [Sarmenta, 2001]. Outra solução utilizada é o *testjob*, que corresponde à submissão de uma tarefa de resultado já conhecido para um usuário executar. A resposta desse usuário é comparada com o resultado conhecido e, a partir daí, é possível saber se ele está executando de forma correta as tarefas enviadas para ele.

Baseados nessas primeiras soluções, diversos mecanismos foram propostos para tentar identificar usuários maliciosos, como [Sarmenta, 2001], que combina *testjobs* com votação majoritária com o objetivo de alcançar uma probabilidade mínima de o resultado da tarefa estar correto. Outro modelo é proposto em [Martins et al., 2006], no qual usuários maliciosos são detectados a partir de um mecanismo baseado em *testjobs* e reputação. Os usuários possuem três níveis de confiabilidade: executores, testadores e ultra-confiáveis (UR). Através do envio de *testjobs* entre esses usuários é possível identificar usuários maliciosos. A solução proposta em [Zhao e GauthierDickey, 2005] sugere a criação de pacotes, chamados de *Quiz*, contendo diversas tarefas que serão enviadas para execução. Essas tarefas são divididas em duas: tarefas normais e *testjobs*, com o objetivo de dificultar a identificação de *testjobs* pelos usuários maliciosos. A partir do resultado dos *testjobs* que vão dentro do pacote, o usuário decide aceitar ou não os resultados das aplicações restantes do pacote.

Além dos mecanismos de detecção e punição dos usuários maliciosos, existem propostas baseadas em confiança para detectar os usuários cujo comportamento prejudica o desempenho da grade [Azzedin e Maheswaran, 2002] [Virendra et al., 2005] [Yu et al., 2004] [Liu e Issarny, 2004a] [Liu e Issarny, 2004b]. Esses mecanismos tratam de diferentes formas as informações adquiridas localmente e as informações obtidas por meio dos outros usuários da grade.

Visando evitar a presença de usuários maliciosos na grade P2P, neste trabalho é proposto um mecanismo de confiança para a detecção e a punição de usuários maliciosos. A proposta apresenta uma solução de segurança reativa baseada nos resultados de *testjobs* inseridos em tarefas normais e analisados em janelas independentes de tempo. A solução foi implementada e analisada em um simulador de grades computacionais. A partir dos

resultados obtidos, pode-se concluir que a proposta é eficiente para detectar e punir os usuários que modificam as respostas das tarefas.

Este trabalho está organizado da seguinte forma: na Seção 2 são apresentados e discutidos os trabalhos relacionados; na Seção 3 é apresentado um mecanismo de segurança baseado em confiança, como forma de solucionar os problemas apresentados; na Seção 4 são definidas as variáveis do ambiente de simulação e são mostrados e discutidos os resultados dessas simulações; finalmente, na Seção 5 são apresentadas as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

Sarmenta propõe um modelo baseado em votação majoritária e em *testjob* [Sarmenta, 2001]. Nesta proposta, é necessário que uma quantidade mínima de respostas corretas seja atingida para que o resultado da tarefa seja considerado correto e, portanto, aceito pela estação de origem. Entretanto, a utilização do mecanismo de votação majoritária pode gerar um alto *overhead*, o que diminui o desempenho da grade computacional.

Com base na solução de *testjobs*, [Martins et al., 2006] propõe um mecanismo hierárquico. Para detectar um usuário malicioso, os usuários testadores analisam os usuários executores e repassam as informações de detecção para os usuários ultra-confiáveis (UR). Desta forma, os usuários UR decidem por detectar/punir ou não o usuário em questão a partir das informações passadas pelos usuários testadores. Para aumentar a segurança nas informações passadas, os usuários testadores também são analisados através de *testjobs* enviados pelos usuários UR. Além disso, os usuários UR são analisados por outros usuários UR. No entanto, mecanismos de detecção de usuários maliciosos que utilizam informações de terceiros são passíveis de detecções/punições incorretas, pois usuários maliciosos podem enviar falsas acusações sobre os usuários normais da grade.

Ao observarem esse tipo de usuário malicioso, que difama os usuários normais da grade, diversas propostas de detecção de intrusão adicionaram modelos de confiança em seus mecanismos de detecção e punição de usuários maliciosos [Azzedin e Maheswaran, 2002] [Virendra et al., 2005] [Yu et al., 2004] [Liu e Issarny, 2004a] [Liu e Issarny, 2004b]. Nestes trabalhos, são utilizadas formulações matemáticas que representam os modelos de confiança. Para relacionar a detecção e a punição com os modelos de confiança são utilizadas as informações locais de detecção, assim como, as informações de confiança passadas por terceiros. Desse modo, o mecanismo pode decidir por punir ou não um determinado usuário. Como dito anteriormente, quando informações de confiança são recebidas de terceiros, existe a possibilidade dos usuários normais serem detectados como maliciosos e de usuários maliciosos não serem detectados.

Com o objetivo de solucionar os problemas citados anteriormente, neste artigo é proposto um mecanismo de segurança baseado em confiança para detectar e punir usuários maliciosos. Esse mecanismo utiliza *testjobs* camuflados nas tarefas normais da grade. Estes *testjobs* são analisados em diferentes intervalos de tempo. Para aumentar a eficiência da proposta, as informações de confiança passadas pelos usuários da grade P2P não são utilizadas diretamente para detectar os usuários maliciosos. Estas informações são usadas para definir o tamanho do intervalo de tempo da análise de *testjobs*. Desta forma, pode-se definir um grau de tolerância para os testes que estão sendo realizados na

grade, evitando a ocorrência de falsas detecções, ou seja, de falso-positivos.

3. Mecanismo Proposto

Neste trabalho é apresentada uma solução de segurança reativa baseada nos resultados de *testjobs* enviados para os usuários em função do tempo. Como o principal objetivo da proposta é identificar e punir usuários de forma reativa, é considerada a existência de um modelo de segurança preventivo de autenticação. As verificações de comportamento são realizadas de forma independente, desse modo, cada usuário pode avaliar o comportamento dos outros usuários que estão fornecendo seus recursos para a grade. A partir do mecanismo de segurança proposto é possível detectar e punir os usuários maliciosos cujo comportamento inadequado compromete o desempenho da grade P2P.

3.1. Cálculo da Confiança

Diversos modelos de confiança existentes na literatura [Azzedin e Maheswaran, 2002] [Virendra et al., 2005] [Yu et al., 2004] [Liu e Issarny, 2004a] [Liu e Issarny, 2004b] utilizam as informações passadas pelos usuários sobre o comportamento de terceiros. Além disso, consideram as informações locais de detecção para inserir alguma punição aos usuários maliciosos. Porém, esses modelos recaem na mesma fragilidade: a possível geração de falso-positivos. Isso ocorre porque o modelo de segurança utiliza as informações passadas pelos outros usuários da grade para a detecção de usuários maliciosos. Entretanto, as informações de confiança passada pelos usuários da grade podem não ser verdadeiras.

Portanto, partindo desses argumentos, é proposta uma métrica de confiança, representada por C. Essa métrica é utilizada para detectar e punir os usuários maliciosos e é calculada somente a partir de testes realizados pelo próprio usuário. Assim, evita-se a punição dos usuários normais devido às falsas informações enviadas por usuários difamadores.

Os testes utilizados na proposta seguem o modelo de *Quiz* [Zhao e GauthierDickey, 2005]. O *Quiz* é baseado na criação de pacotes de tarefas, que são enviados para serem executados na grade. As tarefas que irão compor esse pacote são de dois tipos: tarefas normais e *testjobs*. Esse modelo tem como objetivo dificultar a identificação dos *testjobs* pelos usuários maliciosos, já que são enviados diferentes testes para os usuários.

Para o cálculo da confiança C, apresentado nas Equações 1 e 2, são atribuídos pesos a cada um dos resultados dos testes. Esses pesos formam uma progressão aritmética crescente de razão igual a 1, sendo r_1 referente ao resultado do primeiro teste efetuado e o r_k referente ao resultado do *k-ésimo* teste efetuado. Portanto, se o resultado do teste é correto, então $r = 1$.

$$C = \frac{1 \cdot r_1 + 2 \cdot r_2 + \ldots + (n-1) \cdot r_{n-1} + n \cdot r_n}{1 + 2 + ... + (n-1) + n} \tag{1}$$

$$C = \frac{2[1 \cdot r_1 + 2 \cdot r_2 + \ldots + (n-1) \cdot r_{n-1} + n \cdot r_n]}{n \cdot (n+1)} \tag{2}$$

Um peso maior aos resultados dos testes mais recentes, ou seja, os últimos testes realizados têm uma maior contribuição no cálculo da confiança e, consequentemente, na

detecção e punição de usuários maliciosos. A distribuição de pesos se torna mais clara quando o número de testes realizado é grande e os pesos referentes aos resultados dos testes são iguais a 1. Este é o caso, por exemplo, de um usuário que realiza 30 tarefas normalmente e que, a partir de um determinado instante, decide agir maliciosamente modificando o resultado das tarefas processadas por ele. Pode-se observar que esse tipo de usuário está tentando burlar o mecanismo de segurança. Com base nesse exemplo, é observado que, se os resultados dos testes obtidos mais recentemente têm o mesmo peso dos resultados obtidos inicialmente, então, ocorrerá uma demora na detecção e punição deste usuário malicioso.

Ao observar este exemplo, é possível perceber a importância em utilizar a ponderação com peso maior nos eventos detectados recentemente. Desta forma, a partir do momento em que algum usuário começa a modificar o resultado das tarefas, esses resultados terão um peso maior no cálculo da confiança em relação aos resultados obtidos anteriormente, facilitando a detecção e a punição de usuários que modificam o resultado das tarefas da grade.

3.2. Intervalo de Tempo

Ao observar a forma como é calculada a confiança, nota-se que, na medida em que o tempo passa, o número de tarefas executadas e o número de testes realizados aumentam. Assim, n aumenta e a contribuição de cada teste no cálculo da confiança diminui, fazendo com que o mecanismo se torne mais tolerante aos erros. Isso significa que, quando n é muito grande, existe uma tendência de o mecanismo depender de um maior tempo de interação para detectar e punir os usuários maliciosos que modificam o resultado das tarefas.

Partindo dessa análise, observa-se a necessidade de ajustar a quantidade de testes a serem utilizados para o cálculo da confiança. Propõe-se, então, a utilização de um intervalo de tempo para delimitar um conjunto de testes a serem considerados para o cálculo da confiança. Desse modo, os testes realizados anteriormente a esse determinado intervalo de tempo são desconsiderados para o cálculo da confiança e, consequentemente, para uma eventual detecção.

Analisando a variação desse intervalo de tempo pode-se fazer algumas considerações. A primeira delas é que quanto maior for o valor do Δt, maior será a quantidade de testes utilizados para o cálculo da confiança. Consequentemente, maior será a tolerância a erros, tornando a proposta favorável somente aos usuários que são considerados confiáveis pela grade. Contrário a essa situação, temos que quanto menor for o valor do Δt, menor será a quantidade de testes considerados para o cálculo da confiança. Isso acarreta em uma menor tolerância aos erros, o que é bem interessante para usuários de comportamento malicioso.

Após essa análise pode-se concluir que o principal desafio está relacionado ao cálculo desse intervalo de tempo. Para tentar resolver esse desafio, propõe-se que Δt seja calculado a partir da multiplicação da função $f(C_{grade})$ por Δt_{ant}, como mostra a Equação 3. Sendo C_{grade} calculado a partir da confiança que os usuários da grade passam sobre o usuário a ser avaliado e Δt_{ant} o intervalo de tempo imediatamente anterior a esse.

$$\Delta t = f(C_{grade}) \cdot \Delta t_{ant} \tag{3}$$

3.2.1. Cálculo da função *f*

A função f funcionará como um fator multiplicativo que será responsável pelo aumento ou diminuição do intervalo de tempo de um determinado usuário. Esse fator será calculado utilizando informações de confiança, C_{grade}, passadas pelos usuários da grade.

O objetivo da função f é variar Δt de modo a obtermos uma melhor capacidade de análise de um determinado usuário. No caso de $f > 1$, haverá um aumento em Δt, o que é interessante, como discutido anteriormente, para usuários considerados confiáveis pela grade. No caso de $f < 1$, haverá uma diminuição do Δt, o que interessa aos usuários de comportamento malicioso. Para definir se um usuário é considerado confiável pela grade definimos um limiar L, onde se $C_{grade} < L$ o usuário é considerado malicioso, e se $C_{grade} > L$ o usuário é considerado confiável pela grade.

A partir dessa discussão é feito um esboço do gráfico de f apresentado na Figura 1. Nota-se que se $C_{grade} > L \Rightarrow f > 1$, se $C_{grade} < L \Rightarrow f < 1$ e se $C_{grade} = L \Rightarrow f = 1$. Sendo C_{grade} a confiança da grade em um usuário e L o limite que define a confiabilidade de um usuário, percebe-se que quando $C_{grade} > L$, ou seja, quando o usuário é considerado normal pela grade, tem-se um aumento no valor de Δt. Quando $C_{grade} < L$, ou seja, quando o usuário é considerado malicioso pela grade, tem-se uma redução no valor de Δt. Por fim, quando $C_{grade} = L$, o valor do Δt não é alterado. Nota-se, também, que f é limitada superiormente por f_{max}, já que $C_{grade} \leq 1$. Esse valor máximo ocorre quando a confiança da grade em certo usuário for máxima, ou seja, quando $C_{grade} = 1$, e por isso o valor de Δt terá o maior aumento possível.

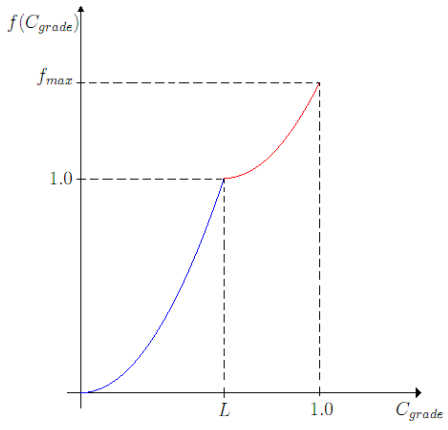

Figura 1. Esboço do gráfico da função f.

A partir do gráfico apresentado na Figura 1 pode-se observar o comportamento de f quando C_{grade} está próximo de L. Quando $C_{grade} > L$ e mais próximo de L significa que a confiança desse usuário está próxima do limiar, então, é importante que o Δt para esse usuário aumente lentamente. Portanto, se o usuário está próximo da faixa para ser considerado malicioso, não é adequado que a tolerância desse usuário seja am-

pliada rapidamente. De forma oposta, quando $C_{grade} < L$, porém não muito menor do que L, isso significa que o usuário é considerado não confiável pela grade. Apesar da confiança do usuário estar próxima do limite, isso acarreta em uma queda mais abrupta do Δt. Portanto, caso esse usuário esteja em um grande intervalo de tempo e comece a agir de forma maliciosa, é importante que ocorra uma diminuição mais acentuada do Δt. Assim, é importante que ocorra uma rápida redução da tolerância contra os erros para esse usuário, diminuindo o tempo de detecção. Essa queda acentuada do valor de f, que, por conseguinte, gera uma queda acentuada do valor de Δt.

As duas curvas observadas no gráfico da Figura 1 como parte da função f são definidas como duas parábolas. A partir dessas duas parábolas, é definida a função f, apresentada na Equação 4. Sendo a_1, b_1 e c_1 coeficientes da primeira parábola e a_2, b_2 e c_2 coeficientes da segunda parábola. Desta forma, é necessário encontrar cada um desses coeficientes para que f se comporte tal como apresentado no gráfico.

$$f(C_{grade}) = \begin{cases} a_1 \cdot C_{grade}^2 + b_1 \cdot C_{grade} + c_1, & se\ \ C_{grade} < L \\ \\ a_2 \cdot C_{grade}^2 + b_2 \cdot C_{grade} + c_2, & se\ \ C_{grade} \geq L \end{cases} \tag{4}$$

Esses coeficientes são encontrados através da resolução de um sistema de equações, construído a partir dos pontos observados nos gráfico da Figura 1, sendo f_{max} e L duas constantes a serem definidas pelo usuário. Para encontrar os coeficientes da primeira equação, são utilizados como pontos: a origem, o x do vértice e o ponto $(L, 1)$.

$$\begin{cases} f(0) = 0 & \rightarrow c_1 = 0 \\ \\ X_v = \frac{-b_1}{a_1} & \rightarrow \frac{-b_1}{a_1} = 0 \qquad \rightarrow b_1 = 0 \\ \\ f(L) = 1 & \rightarrow a_1 \cdot L^2 = 1 \quad \rightarrow a_1 = \frac{1}{L} \end{cases} \tag{5}$$

Para encontrar os coeficientes da segunda equação também são utilizados três pontos: x do vértice, y do vértice e o ponto $(1, f_{max})$.

$$\begin{cases} X_v = L = \frac{-b_2}{a_2} \\ \\ Y_v = 1 = \frac{-b_2^2 + 4a_2 \cdot c_2}{4a_2} \\ \\ f(1) = f_{max} = a_2 + b_2 + c_2 \end{cases} \Rightarrow \begin{cases} a_2 = \frac{-b_2}{L} \\ \\ b_2 = \frac{2L\ (f_{max} - 1)}{-L^2 + 2L - 1} \\ \\ c_2 = \frac{2 - L \cdot b_2}{2} \end{cases} \tag{6}$$

A partir da definição das duas constantes L e f_{max} são encontrados, através de uma simples substituição, os coeficientes a, b e c necessários para achar a função $f(C_{grade})$, tal como apresentada no gráfico da Figura 1.

4. Parâmetros e Resultados

Antes de apresentar o ambiente de simulação, é importante destacar os tipos de usuários maliciosos utilizados nesse ambiente. São utilizados dois tipos de comportamento malicioso que esses usuários poderão ter: modificar o resultado das tarefas com

uma determinada probabilidade e enviar falsas reputações de usuários para a grade. A partir disso, são definidos cinco tipos de usuários maliciosos, classificados no ambiente da seguinte forma:

- **Modificadores:** esse tipo de usuário malicioso corrompe o resultado das tarefas enviadas para ele. Para analisar as variações deste tipo de usuário, foram observadas diferentes probabilidades desses usuários se comportarem de forma maliciosa.
- **Inteligentes:** esse usuário ganha confiança da grade agindo normalmente por um período, executando normalmente as tarefas enviadas para ele. Após certo tempo, ele começa a corromper o resultado das tarefas também com certa probabilidade.
- **Difamadores:** usuários que difamam outros usuários da grade. Em outras palavras, estes usuários enviam falsas informações de confiança sobre terceiros para os outros usuários da grade.
- **Modificadores e Difamadores:** usuários que, além de modificar o resultado das tarefas com certa probabilidade, difamam os usuários da grade.
- **Inteligentes e Difamadores:** usuários maliciosos inteligentes, mas que também enviam falsas informações de confiança para a grade.

Para validar o modelo de segurança proposto, foi simulado um ambiente de grades p2p utilizando o simulador Gridsim [Buyya e Murshed, 2002]. Nessa seção são apresentados os parâmetros e os resultados do ambiente de simulação do trabalho.

4.1. Ambiente de Simulação

Para avaliar o impacto da utilização do mecanismo de segurança proposto, foram simulados diferentes cenários de grades p2p. Os cenários simulados utilizaram um total de 60 usuários. Desse total, 15 usuários eram maliciosos, ou seja, 25%. Cada usuário gera 1500 pacotes (tarefa + *testjob*), que são executados pela grade p2p. Os usuários executores das tarefas são escolhidos aleatoriamente pelos usuários que as submetem. Durante a simulação, a partir do instante em que um usuário detecta um terceiro como malicioso, ele não envia nem recebe pacotes relacionados ao usuário malicioso. Portanto, o usuário malicioso fica bloqueado para realizar qualquer tipo de execução com o usuário que o detectou. Para cada cenário foram realizadas 30 simulações e todos os resultados são apresentados com um intervalo de confiança de 95%.

A porcentagem de 25% de usuários maliciosos foi escolhida depois de observado que essa quantidade de usuários maliciosos já afeta significativamente o desempenho da grade. O gráfico da Figura 2 mostra o resultado de um cenário de grades p2p sem a utilização do mecanismo de segurança. Nesse cenário é variada a quantidade de usuários maliciosos modificadores, assim como a probabilidade desses usuários modificarem o resultado das tarefas. Desta forma, é possível apresentar o percentual de tarefas da grade com resultados alterados. Nota-se que, com 25% dos usuários maliciosos, a quantidade de tarefas com resultados corrompidos variam de $2,5\%$ a 25%, dependendo da probabilidade com que esse usuário modifica o resultado da tarefa. Isso comprova que, com essa porcentagem de usuários maliciosos, o índice de tarefas alteradas já é bem elevado e já prejudica consideravelmente o desempenho a grade.

Para a simulação, as constantes f_{max} e L foram definidas com os valores 1.5 e 0.7, respectivamente. Isso significa que o aumento máximo de Δt será de 50%. O limiar

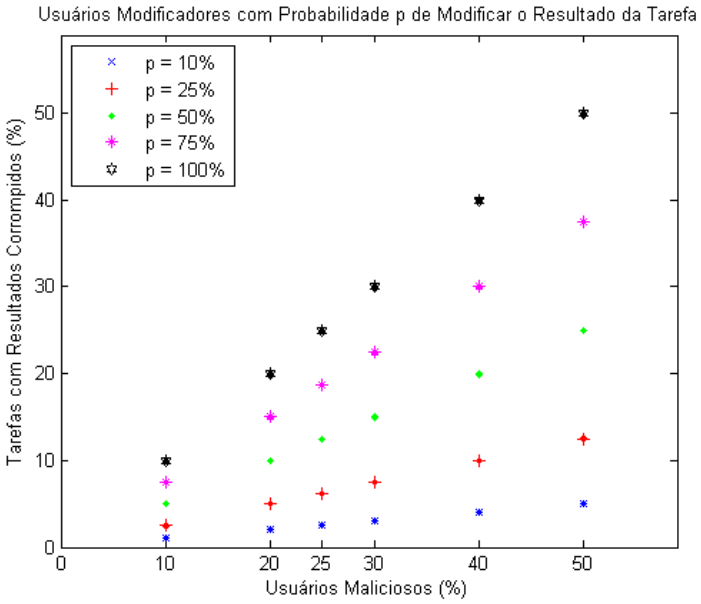

Figura 2. Percentual de tarefas corrompidas em um cenário sem mecanismos de segurança.

de confiança L escolhido deve-se ao seguinte fato: como a confiança local é baseada numa fração da quantidade de testes respondidos corretamente pela quantidade total de testes enviados, tem-se que essa confiança será, em média, a fração de testes respondidos corretamente. Dependendo dos pesos de cada um desses resultados, a confiança pode ser maior ou menor do que a fração. Ao considerar 25% das tarefas respondidas erroneamente uma porcentagem bem prejudicial para grade P2P, pode-se perceber que, se $L = 0.7$, o mecanismo ainda é tolerante. Essa tolerância é importante, pois caso aconteça algum erro, seja de *hardware* ou *software*, o usuário normal não será punido como malicioso.

Foram simulados cenários com cada um dos usuários maliciosos apresentados. Para os usuários maliciosos que modificam o resultado das tarefas, a probabilidade de eles modificarem o resultado das tarefas foi variada em 10%, 25%, 50%, 75% e 100%. Para os usuários maliciosos que difamam outros usuários, foi definido que eles escolheriam aleatoriamente 50% dos usuários normais da grade para difamar.

$$C_{grade,A} = \frac{\sum (C_{j,A} \cdot C_j)}{\sum C_j} \tag{7}$$

Para o cálculo de C_{grade} é utilizado um modelo de média ponderada apresentado em [Braga et al., 2009], pois esse foi um dos modelos que obteve melhores resultados nas análises realizadas. Nesse modelo, é obtida a média das informações de confiança passadas pelos usuários da grade, ponderada pela confiança nesse usuário que está passando essa informação. Por exemplo, para calcular o C_{grade} de um usuário A, como mostra a Equação 7, são utilizadas as informações de confiança passadas pelos usuários da grade em relação ao usuário A. Desta forma, são consideradas somente as informações passadas pelos usuários que são considerados localmente confiáveis, ou seja, quando $C_j \geq L$. Assim, nota-se que quanto maior for a confiança em um determinado usuário j, maior será a contribuição da informação passada por esse usuário no cálculo de C_{grade}.

4.2. Resultados

Os resultados são analisados a partir de gráficos que mostram a quantidade de usuários maliciosos detectados durante o tempo de simulação, utilizando o mecanismo de segurança proposto. A partir dos resultados, são discutidos os impactos que o mecanismo traz para a grade p2p.

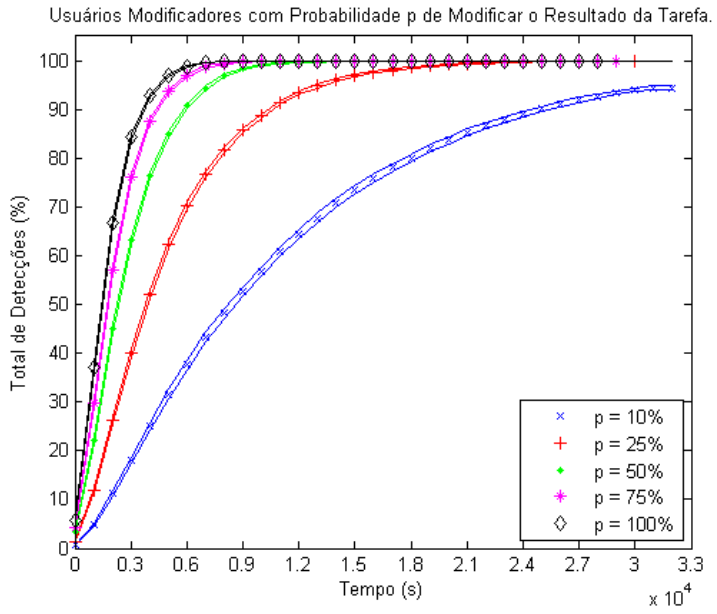

Figura 3. Total de detecções em um cenário com 25% de usuários modificadores.

Usuários Modificadores

De acordo com os resultados apresentados no gráfico da Figura 3, é possível observar que os usuários que mais prejudicam a grade são detectados mais rapidamente do que os usuários com uma menor probabilidade de corromper as tarefas. Nota-se que no instante de tempo igual a $9 \cdot 10^3$ segundos de simulação, praticamente todos os usuários com probabilidade $p = 100\%$, $p = 75\%$ e $p = 50\%$ são detectados e punidos. Já no instante de tempo igual a $18 \cdot 10^3$ segundos, são detectados quase todos os usuários com $p = 25\%$. Até mesmo os usuários com menor probabilidade de corromper o resultado das tarefas, $p = 10\%$, são detectados em até 94% dos casos, apesar de demandarem um tempo maior para que isso ocorra. Nota-se também que a detecção dos usuários com $p = 10\%$ poderia chegar a 100% se um tempo maior de simulação for considerado.

Usuários Inteligentes

No outro cenário, os usuários maliciosos executam normalmente as tarefas da grade até o instante de tempo igual a $15 \cdot 10^3$ segundos. A partir desse momento, eles passam a corromper o resultado das tarefas com uma probabilidade p. Ao observar a Figura 4 nota-se uma semelhança das curvas de detecções com os resultados da Figura 3.

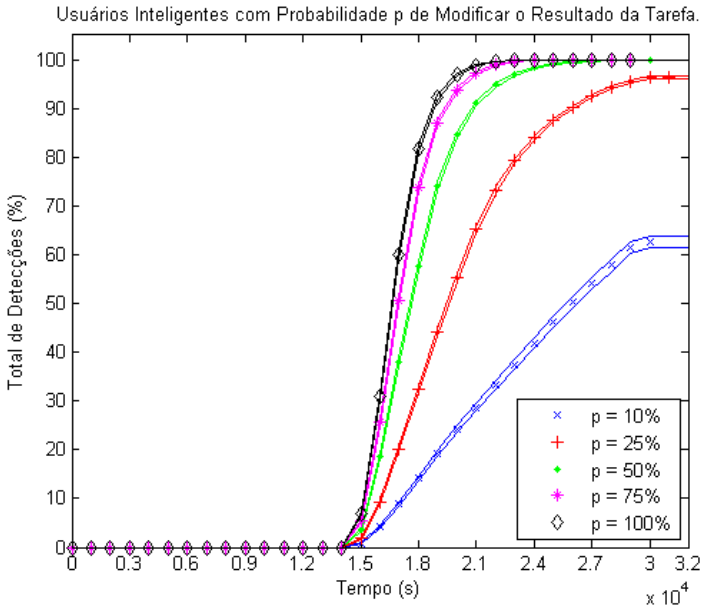

Figura 4. Total de detecções em um cenário com 25% de usuários inteligentes.

Isso significa que, apesar desse tipo de usuário tentar burlar o mecanismo de segurança, o tipo de comportamento é solucionado pelo mecanismo proposto. Isso acontece devido à utilização das janelas de tempo, sem memória, em conjunto com a ponderação utilizada para o cálculo da confiança local. Assim, mesmo que um usuário se comporte corretamente até um determinado instante, caso ele inicie algum comportamento malicioso, ele será detectado e punido.

Usuários Difamadores

Foram simulados cinco cenários diferentes com esse tipo de usuário, variando a quantidade em 10%, 20%, 30%, 40% e 50%. Em nenhum dos cenários analisados foram detectados falso-positivos. Usuários difamadores também não foram detectados. Isso ocorreu porque o mecanismo proposto não utiliza diretamente a confiança passada pelos usuários da grade para a detecção de usuários maliciosos. A confiança da grade é utilizada para calcular o intervalo de tempo no qual o usuário será analisado, definindo assim, certa tolerância ao usuário analisado. Isso significa que uma baixa confiança da grade em um usuário não causa, necessariamente, uma detecção, mas sim uma menor tolerância nesse usuário. O que denota que o mecanismo proposto é robusto para esse tipo de comportamento malicioso. Pode-se dizer, também, que o desempenho da grade não é afetado nesse cenário, pois o usuário difamador somente difama os usuários normais da grade, não alterando o resultado das tarefas executadas por ele.

Usuários Modificadores e Difamadores

Nos resultados apresentados na Figura 5, nota-se uma semelhança com os resultados da Figura 3. Apesar de os usuários maliciosos difamarem os usuários normais da grade, esse comportamento não terá um grande impacto no resultado final, pois no início da grade, o intervalo de tempo ainda é pequeno. Portanto, eles são detectados por serem

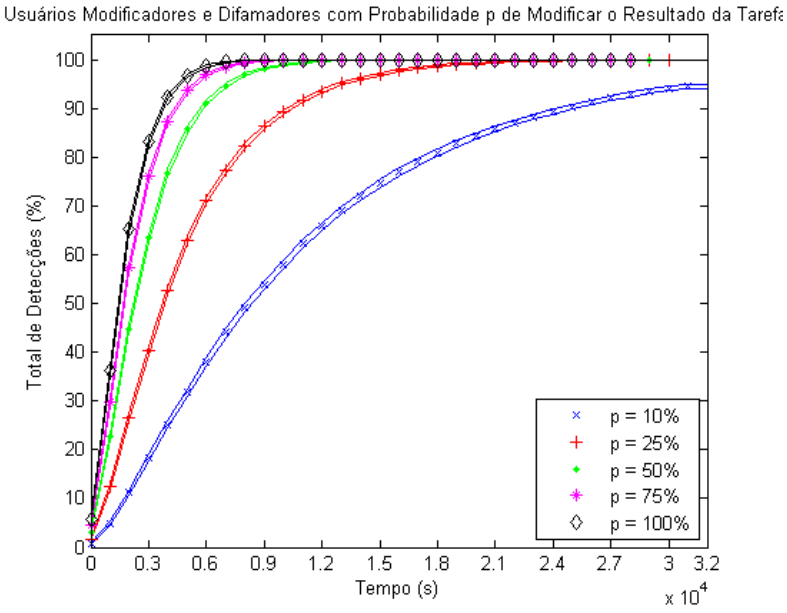

Figura 5. Total de detecções em um cenário com 25% de usuários modificadores e difamadores.

modificadores logo no início, dificultando a realização dos comportamentos de difamação por parte destes usuários.

Usuários Inteligentes e Difamadores

Os resultados desse cenário são apresentados na Figura 6. Apesar da semelhança com os resultados na Figura 4, nota-se que no cenário onde os usuários maliciosos difamam os usuários da grade, ocorre uma detecção mais rápida. Como esse tipo de usuário malicioso difama os usuários normais e, inicialmente, todos os usuários se comportam normalmente, então nota-se que o intervalo de tempo dos usuários não aumenta como no outro cenário. Consequentemente, a tolerância tende a ser menor. Por exemplo, ao observar o resultado no instante de tempo igual a $18 \cdot 10^3$ segundos, nota-se que aproximadamente 44% dos usuários maliciosos com $p = 25\%$ são detectados. No entanto, ao observar o mesmo ponto na Figura 4 nota-se que, aproximadamente, 32% dos usuários maliciosos com $p = 25\%$ são detectados. Isso ocorre porque a confiança passada pelos usuários da grade é utilizada para o cálculo do intervalo de tempo em que um usuário será analisado. Em outras palavras, o intervalo de tempo para a análise do usuário é menor, o que acarreta em uma menor tolerância aos erros e, consequentemente, em uma detecção mais rápida de usuários que corrompem o resultado das tarefas.

Ao observar os gráficos apresentados, pode-se perceber que a proposta é eficiente na detecção e punição de estações maliciosas em cenários de grades P2P. É importante destacar que esta proposta pode ser utilizada em outros cenários de grades computacionais, assim como em outros tipos de cenários que exigem uma relação de confiança entre as estações da rede, tais como as redes *ad hoc* sem fio.

5. Conclusão

Nesse artigo foi proposto e analisado um mecanismo de segurança baseado em confiança para grades P2P. Esse mecanismo tem como objetivo principal detectar e

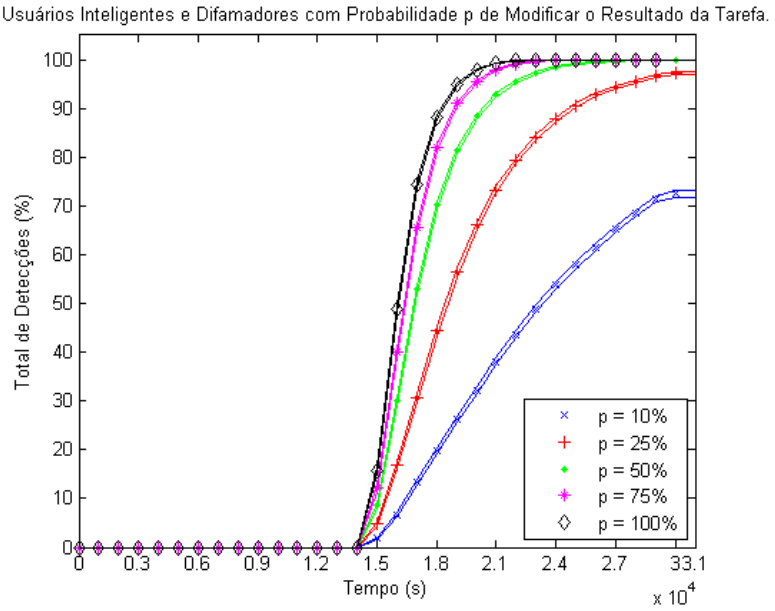

Figura 6. Total de detecções em um cenário com 25% de usuários inteligentes e difamadores.

punir usuários maliciosos que corrompem o resultado das tarefas submetidas para grade. Além disso, foi implementada uma funcionalidade que evita falsas punições na grade, provocadas pelos usuários difamadores. O mecanismo calcula a confiança a partir de informações obtidas localmente durante uma janela de tempo. Essa janela de tempo define a tolerância para cada usuário baseada nas informações de confiança passadas pelos usuários da grade.

A partir dos resultados obtidos pôde-se concluir que o mecanismo é eficiente contra os usuários que modificam o resultado das tarefas com uma determinada probabilidade, conseguindo detectar e punir 100% dos usuários com probabilidade $p \geq 25\%$ de corromper o resultado da tarefa e até 94% dos usuários com $p = 10\%$. O mecanismo se mostrou eficiente também contra os usuários maliciosos que agem normalmente por um período e, após certo instante de tempo, começam a corromper o resultado das tarefas. A partir dos resultados, pode-se perceber que são detectados até 100% desses usuários maliciosos, chamados de inteligentes. Em todos os cenários o mecanismo se mostrou robusto aos usuários maliciosos que difamam os usuários da grade, visto que em nenhum momento são gerados falso-positivos nesses cenários. Isso porque no mecanismo proposto não há a utilização direta das informações de confiança passadas pelos usuários da grade para a detecção de usuários maliciosos. Como essas informações não são diretamente utilizadas para decidir se o usuário é malicioso ou não, então não há detecção de usuários normais como maliciosos.

Apesar do conluio ser um problema existente e ainda sem solução para grupos formados por mais de 50% das estações da grade, esse artigo assume a não existência desse tipo de ataque. Portanto, as estações não agirão em conjunto para tentar enganar o mecanismo de detecção e punição da grade. Como trabalho futuro, a proposta está sendo implementada em um ambiente real de grades computacionais *peer-to-peer*, por meio dos recursos computacionais do projeto SIMEGRID [SIMEGRID, 2009]. Além disso, estão

sendo analisados outros métodos de punição para serem avaliados em conjunto com a proposta apresentada.

Referências

Azzedin, F. e Maheswaran, M. (2002). Evolving and managing trust in grid computing systems. *IEEE Canadian Conference on Electrical Computer Engeneering*.

Braga, R., Chaves, I., Andrade, R., Souza, J. e Schulze, B. (2009). Modelos probabilísticos de confiança para grades computacionais ad hoc. *Workshop on Grid Computing and Applications (WCGA)*.

Buyya, R. e Murshed, M. (2002). Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. *Journal of Concurrency and Computation: Practice and Experience (CCPE)*.

Foster, I. e Kesselman, C. (2004). *The Grid 2: Blueprint for a New Computing Infrastructure*. Elsevier, segunda edição.

Liu, J. e Issarny, V. (2004a). Enhanced reputation mechanism for mobile ad hoc networks. *Proceedings of iTrust*.

Liu, J. e Issarny, V. (2004b). A robust reputation system for p2p and mobile ad hoc networks. *Second Workshop on Economics of Peer-to-Peer Systems*.

Marsh, M., Kim, J., Nam, B., Lee, J. e Ratanasanya, S. (2008). Matchmarking and implementation issues for a p2p desktop grid. *Parallel and Distributed Processing*. IEEE International Symposium on.

Martins, F., Maia, M., Andrade, R., Santos, A. e Souza, J. (2006). Detecting malicious manipulation in grid environments. *IEEE International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*.

Sarmenta, L. (2001). Sabotage-tolerance mechanisms for volunteer computing systems. *First IEEE/ACM International Symposium on Cluster Computing and the Grid. Proceedings*.

Seti@Home (2009). http://setiathome.berkeley.edu/.

SIMEGRID (2009). Simulações médicas em grids. *http://comcidis.lncc.br/index.php/SIMEGRID*.

Uppuluri, P., Jabisetti, N., Joshi, U. e Lee, Y. (2005). P2p grid: service oriented framework for distributed resource management. *IEEE International Conf. on Services Computing*, p. 347–350.

Virendra, M., Jadliwala, M., Chandrasekaran, M. e Upadhyaya, S. (2005). Quantifying trust in mobile ad-hoc networks. *International Conference on Integration of Knowledge Intensive Multi-Agent Systems*.

Yu, B., Singh, M. e Sycara, K. (2004). Developing trust in large-scale peer-to-peer systems. *First IEEE Symposium on Multi-Agent Security and Survivability*.

Zhao, S., L. V. e GauthierDickey, C. (2005). Result verification and trust-based scheduling in peer-to-peer grids. *Fifth IEEE International Conference on Peer-to-Peer Computing*.