Mecanismos de segurança para grades de computação voluntária

Leonardo Laface de Almeida , Marco Aurélio Amaral Henriques (Orientador)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6101, 13083-970 – Campinas, SP, Brasil

{lalmeida,marco}@dca.fee.unicamp.br

Abstract – Grid computing environments are used to process algorithms that require a large amount of computing power. Some of these environments take advantage of computing power from personal computers connected to the internet. Several security services are required by this kind of grid computing. In this document, we propose three security mecanisms. The first provides data authenticity, integrity and confidentiality. The second provides computer identification and the last improves the application result reliability using trust models.

Keywords – Data security, Volunteer grid computing, Trust models

1. Introdução

Uma grade computacional (*grid computing*) é um tipo de sistema de processamento paralelo que faz uso do poder computacional de computadores geograficamente dispersos e interligados por redes de alto desempenho para resolver problemas que requerem grande volume de cálculo. Algumas grades, chamadas de grades de computação voluntária, fazem uso de milhares de computadores pessoais conectados à *internet*, que sozinhos tem baixo poder de cálculo, mas juntos agregam uma alta capacidade computacional.

Para se usar eficientemente um sistema de processamento paralelo é necessário dividir um problema que exige grande poder computacional (aplicação) em pequenos problemas que exigem baixo poder computacional (tarefas) e os distribuir pelos computadores pertencentes ao sistema (trabalhadores). Depois que as tarefas foram computadas pelos trabalhadores, os resultados são agrupados de forma a se obter a solução da aplicação.

Assim como outros sistemas, as grades computacionais podem ser alvo de vários tipos de ataques, tais como invasão de computadores, proliferação de vírus, falsificação de resultados entre outros. As grades de computação voluntária estão mais vulneráveis a estes ataques porque utilizam computadores pessoais que, normalmente, não são administrados pela grade e nem sempre podem ser vinculados aos seus donos. Este trabalho propõe mecanismos para aumentar a segurança em grades de computação voluntária.

2. Segurança em grades

Do ponto de vista dos provedores de serviços de grade, dois aspectos de segurança se destacam:

1. proteção dos trabalhadores contra ataques provenientes de aplicações da grade;
2. proteção das aplicações contra ataques provenientes de trabalhadores.

2.1. Proteção para trabalhadores

Uma forma de oferecer segurança aos trabalhadores é identificando e autenticando os computadores responsáveis pela infaestrutura da grade. Isto pode ser feito utilizando a infraestrutura de chaves públicas (PKI), que ainda oferece a vantagem de garantir integridade das conexões.

A infraestrutura de chaves públicas se baseia em um modelo de confiança que utiliza certificados digitais criados e assinados por autoridades certificadoras. O dono de um certificado recebe duas chaves, uma privada e outra pública. A privada deve ser mantida sob sigilo. Caso contrário, a autenticidade do dono do certificado não poderá ser comprovada. Portanto, cabe aos trabalhadores confiarem nas autoridades que assinaram os certificados utilizados pelos computadores na premissa de que as chaves privadas são mantidas sob sigilo pelos seus donos.

2.2. Proteção para aplicações

Um dos serviços de segurança requeridos para aplicações é a identificação de trabalhadores. Esta identificação é necessária para que se possa punir de alguma forma os trabalhadores nocivos. Não é viável

a identificação utilizando certificados digitais devido ao custo elevado para gerenciar tais certificados em uma grade com grande número de trabalhadores. Utilizar o endereço IP também não resolve o problema, porque trabalhadores podem estar conectados a NAT's, impedindo sua identificação. Outros identificadores, tais como o endereço MAC, também não atendem as necessidades porque é um identificador facilmente clonável que está restrito a uma rede local. Algumas grades utilizam arquivos de texto (similares a arquivos *cookies* em *browsers*) para identificar trabalhadores [1].

Um outro serviço de segurança necessário é a detecção de resultados incorretos vindo dos trabahadores, uma vez que é difícil garantir que todos os resultados retornados são corretos. Alguns trabalhos propõem a utilização de réplicas de tarefas para verificar os resultados, utilizando métodos de inspeção, de votação e de reputação [2].

3. Mecanismos de segurança para grades de computação voluntária

Propomos a implementação de três mecanismos de segurança neste trabalho. O primeiro deles visa proteger os trabalhadores contra ataques oriundos de computadores responsáveis pela infraestrutura da grade. O segundo e o terceiro visam proteger as aplicações contra ataques oriundos de trabalhadores.

3.1. Uso de PKI para computadores da infraestrutura da grade

Algumas grades já utilizam PKI, alternativa que reforçamos neste trabalho. Ao utilizar a infraestrutura de chaves públicas, é possível garantir aos trabalhadores a autenticidade dos computadores responsáveis pela infraestrutura da grade. Além disso, por meio do uso de certificados digitais nestes computadores principais, é possível ainda prover sigilo e integridade das mensagens trocadas entre eles e os trabalhadores.

3.2. Identificação de trabalhadores

Algumas grades criam para os trabalhadores arquivos do tipo *cookie*, que possuem uma identificação única criada pela grade. Diferentemente desses trabalhos, propomos que este arquivo seja assinado utilizando uma chave privada de algum computador da infraestrutura da grade para garantir a origem e a integridade do arquivo. Isto diminui bastante a chance dos arquivos serem alterados, possibilitando a detecção de trabalhadores nocivos ao sistema. A proposta de gerenciamento pela grade dos arquivos que identificam trabalhadores é ilustrada na Figura 1.

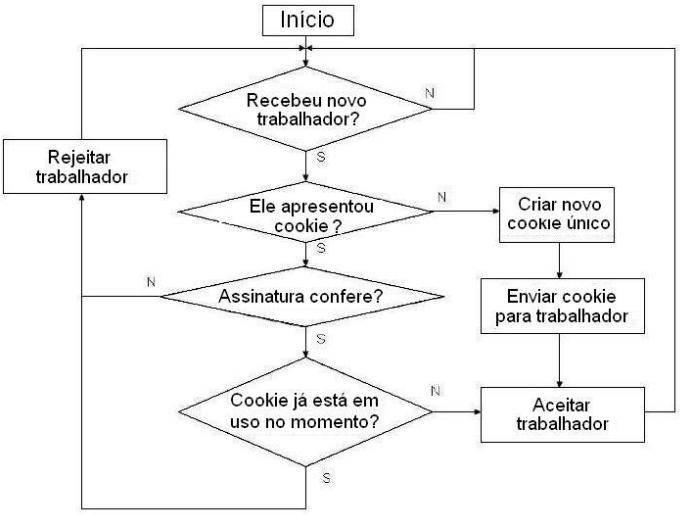

Figura 1. Gerenciamento de arquivos *cookies*

Com a utilização de *cookies*, é possível punir o trabalhador se a grade identificar qualquer ataque proveniente dele. Contudo, a identificação pode falhar porque o dono do trabalhador pode apagar o *cookie* depois de ser punido e inserir a mesma máquina como um novo trabalhador, recebendo uma nova identificação. Os donos dos trabalhadores devem restringir o acesso aos *cookies* para dificultar que terceiros os removam ou os copiem.

3.3. Verificação de resultados de tarefas

É proposto um mecanismo que utiliza os métodos de inspeção, de votação e de reputação, aproveitando a confiabilidade dos computadores responsáveis pela infraestrutura da grade. Nesta proposta, a grade deve classificar os trabalhadores, comparando os resultados retornados de tarefas e de suas réplicas.

Note que o número de réplicas por aplicação deve ser escolhido de forma a garantir maior confiabilidade sem comprometer significativamente o desempenho do sistema. É fundamental também evitar que os trabalhadores percebam que estão sendo testados. Para isso, propõe-se que:

- nenhum trabalhador execute a mesma tarefa mais de uma vez;
- as réplicas sejam criadas por amostragem a partir de tarefas convencionais e verificadas durante a execução de aplicações;
- a classificação (reputação) dos trabalhadores deve ser perene, isto é, mantida entre execuções de diferentes aplicações.

O mecanismo de verificação que segue o diagrama ilustrado na Figura 2. Trabalhadores classificados como desconhecidos são aqueles nunca testados pelo algoritmo. Trabalhadores honestos são aqueles aprovados no seu último teste. Trabalhadores suspeitos são aqueles reprovados no seu último teste. O trabalhador reprovado em dois testes consecutivos é testado isoladamente com um computador totalmente confiável, ou seja, pertencente à infraestrutura da grade. Se falhar também neste teste, ele será banido da grade. Certas precauções devem ser tomadas ao implementar esta proposta:

- todos os rótulos de trabalhadores honestos devem ter prazo de validade;
- trabalhadores suspeitos devem ser priorizados nos testes, mas nunca comparados entre si.

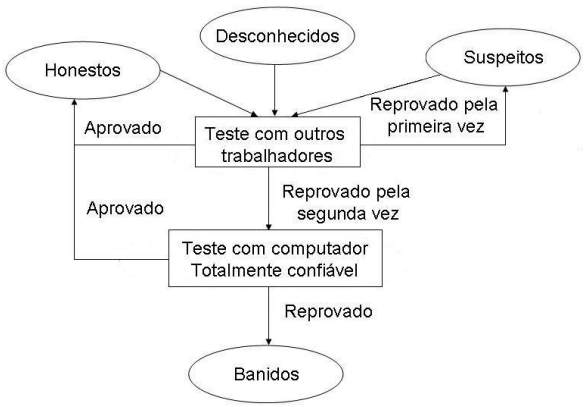

Figura 2. Definição de rótulos

Esta proposta propicia uma maior confiabilidade nos resultados de tarefas visto que alguns resultados coincidem em mais de um trabalhador. Além disso, os resultados de tarefas executadas pelos computadores totalmente confiáveis são considerados como corretos. Outra vantagem da proposta é que ela possui menor sensibilidade a erros transitórios, já que os trabalhadores não são banidos por retornar um ou outro resultado incorreto esporadicamente.

A proposta possui duas dificuldades. A primeira é detectar trabalhadores que retornam resultados corretos e incorretos alternadamente. A segunda é identificar grupos de trabalhadores que retonam resultados incorretos, porém idênticos entre si.

4. Simulações

Consideramos que um trabalhador que retorna ao menos um resultado incorreto é um trabalhador nocivo e que trabalhadores combinados são os que apresentam resultados incorretos e idênticos. Foram feitas simulações do algoritmo que seguem as seguintes condições:

- total de trabalhadores: 500 (constante);
- trabalhadores nocivos: entre 0% e 90% do total de trabalhadores;
- trabalhadores combinados: entre 0% e 100% do total de trabalhadores nocivos.

Uma maneira de avaliar o desempenho do algoritmo foi mantê-lo em loop até que todos os trabalhadores nocivos fossem banidos do sistema e cada trabalhador fosse testado ao menos uma vez. Estas condições são muito rigorosas e não são viáveis na prática, mas servem para avaliar o comportamento do mecanismo. Cada situação foi testada 500 vezes para cálculo de média e erro padrão e foi medido o volume de trabalho extra gasto pelo algoritmo até ele parar. Este volume é dado pelo número R de réplicas necessárias. Para efeito de comparação, foi utilizado um volume de trabalho relativo extra tendo como referência o volume de trabalho extra quando não há trabalhadores nocivos. Para rotular todos os trabalhadores, o volume de trabalho extra mínimo (usado como referência) é $R_{min} = \lceil W/2 \rceil$ réplicas, onde W é o total de trabalhadores.

5. Resultados

Em todas as figuras estão plotados também os erros padrão, que são muito pequenos e quase imperceptíveis. As Figuras 3 e 4 mostram o volume de trabalho relativo extra gasto pelo algoritmo quando não há trabalhadores nocivos combinados e quando todos os trabalhadores nocivos são combinados, respectivamente.

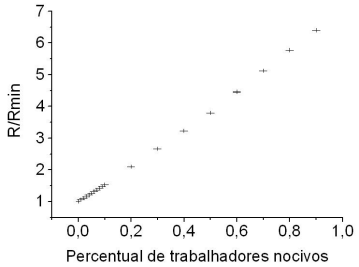

Figura 3. Volume de trabalho relativo extra sem trabalhadores combinados

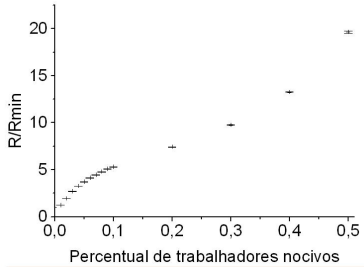

Figura 4. Volume de trabalho relativo extra sendo todos os trabalhadores nocivos combinados

É possível verificar que o mecanismo cria menos réplicas para situações mais comuns na prática, isto é, aquelas em que há mais de 90% de trabalhadores que retornam resultados corretos. Isso significa que o custo ao usar o algoritmo não aumenta muito em relação ao caso referência. Seja T o número total de tarefas de uma aplicação. Seja R/T o número relativo de réplicas necessárias para testar e banir todos os trabalhadores nocivos. Assim, é possível demonstrar que $R/T = 1/2 \times R/R_{min} \times (T/W)^{-1}$. As Figuras 5 e 6 mostram os valores de R/T necessários para que o algoritmo identifique todos os trabalhadores nocivos a medida que varia a relação T/W quando não há trabalhadores combinados e quando todos os trabalhadores nocivos são combinados, respectivamente.

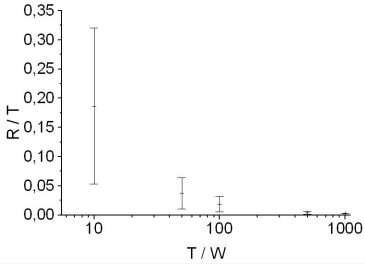

Figura 5. Valores de R/T sem trabalhadores combinados

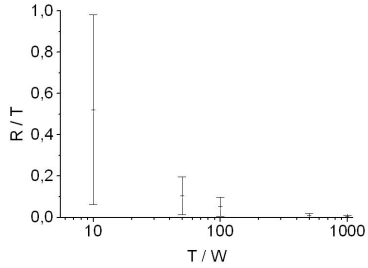

Figura 6. Valores de R/T sendo todos os trabalhadores nocivos combinados

É possível verificar que a medida que aumenta a relação T/W, o algoritmo se torna mais eficiente porque a relação R/T diminui. Em situações práticas, o valor de T/W costuma ser maior que 10 em grades de computação voluntária. Algumas aplicações possuem essa relação maior do que 100. Nestes casos, o mecanismo se mostrou eficiente, principalmente quando não há trabalhadores combinados.

6. Conclusões

Este trabalho propõe a implementação de três mecanismos para prover segurança para grades de computação voluntária. A primeira é a adoção de certificados digitais para identificar os computadores da infraestrutura da grade a fim de oferecer maior segurança aos trabalhadores. A segunda é a criação de arquivos *cookies* para identificar trabalhadores, viabilizando a proteção de aplicações contra ataques de trabalhadores. A última é a implementação de um algoritmo que utiliza réplicas de tarefas para classificar trabalhadores, oferecendo maior confiabilidade ao resultado da aplicação.

Simulações preliminares do algoritmo proposto mostraram que ele é eficaz para os casos em que o número total de trabalhadores é constante, sem causar grande impacto ao sistema. Testes em ambientes reais estão sendo feitos para comprovar a eficácia dos mecanismos propostos.

Referências

[1] Erik Elmroth, Mats Nylen, and Roger Oscarsson. A user-centric cluster and grid computing portal. *Int. J. Comput. Sci. Eng.*, 4(2):127–134, 2009.

[2] Luis F. G. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. In *CCGRID '01*, page 337, USA, 2001.