# Aumentando a confiabilidade dos resultados em grades computacionais com recursos públicos

**Leonardo Laface de Almeida**[1] , **Marco Aurélio Amaral Henriques**[1]

[1]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 with public resources are used to process large problems taking advantage of many personal computers connected to the Internet. Some extra security services are required in this kind of environment to improve the quality of the results obtained. This document presents a method, which is based on task replication, to verify the correctness of the results given by the personal computers.*

***Resumo.*** *Grades computacionais com recursos públicos são utilizadas para processar grandes problemas utilizando computadores pessoais conectados à internet. Alguns serviços de segurança adicionais são necessários neste tipo de ambiente para aumentar a qualidade dos resultados obtidos. Este documento apresenta um método, baseado em replicação de tarefas, para verificar a correção dos resultados fornecidos pelos computadores pessoais.*

## 1. Introdução

Grade computacional com recursos públicos (*Grid Computing with public resources*) é um tipo de sistema de processamento paralelo que faz uso do poder computacional de computadores pessoais conectados à internet. É desejável que estas grades ofereçam serviços de segurança para evitar ataques, tais como aqueles em que os resultados retornados pelos computadores pessoais são distorcidos ou falsificados. Apresentamos neste trabalho um mecanismo de segurança que verifica a correção dos resultados retornados por computadores pessoais utilizando réplicas que podem prover um maior grau de confiabilidade no resultado obtido. Alguns trabalhos utilizam réplicas para que a grade seja tolerante a falhas [Fechner et al. 2008]. Outros apresentam soluções contra ataques por sabotagem utilizando métodos de votação e de inspeção [Sarmenta 2001] . Diferentemente das propostas apresentadas nestas referências, o mecanismo proposto faz uso de pelo menos um computador totalmente confiável para comparar resultados. O uso de computadores confiáveis é vantajoso, já que os mesmos permitem uma classificação mais precisa dos resultados que são produzidos pelos computadores que formam a grade.

## 2. Mecanismo de verificação de resultados das tarefas

É proposto um mecanismo de criação de réplicas das tarefas enviadas aos computadores pessoais que fazem o processamento (trabalhadores). Um computador mestre classifica tais trabalhadores à medida que compara os resultados retornados pelas tarefas. Para evitar que os trabalhadores percebam que estão sendo testados, propõe-se que nenhum

trabalhador execute a mesma tarefa mais de uma vez e que as réplicas de teste sejam criadas a partir de tarefas convencionais durante a resolução das mesmas. O número de réplicas utilizadas em cada programa executado pela grade deve ser escolhido de forma a garantir uma maior confiabilidade no resultado e a não comprometer significativamente o desempenho do sistema, visto que a adoção de replicação aumenta o tempo de execução dos programas. É proposto que as réplicas sejam criadas e verificadas por amostragem, para diminuir o impacto da adoção das mesmas.

Este mecanismo de verificação de resultados de réplicas deve ser usado permanentemente no sistema enquanto os programas são executados. Ao final da execução de um programa, a classificação dos trabalhadores em níveis de confiabilidade deve ser preservada para uso posterior. Nesta proposta, é sugerido ainda o uso de computadores totalmente confiáveis para comparar resultados de tarefas. Esses computadores confiáveis não devem ser usados frequentemente para evitar que sejam sobrecarregados. Isto se consegue usando-os só em comparações finais, após terem se esgotado outras alternativas de comparação entre réplicas de tarefas. Os trabalhadores sob teste são classificados como confiáveis temporariamente ao retornarem um resultado que confere com o resultado apresentado por outro trabalhador. Eles são considerados suspeitos caso contrário. Os trabalhadores suspeitos são priorizados para os testes, mas nunca são comparados entre si. Quando um trabalhador suspeito retorna outro resultado não coincidente com outro trabalhador, ele é submetido a um terceiro teste, desta vez comparando com um computador totalmente confiável. O trabalhador será banido do sistema se falhar nesta terceira comparação e será considerado confiável temporariamente, caso contrário. Este esquema com múltiplas comparações é vantajoso porque trabalhadores de uma grade com recursos públicos estão mais sujeitos a erros transitórios. Entretanto, este mecanismo pode ser ineficaz se o sistema possuir trabalhadores não confiáveis que retornam resultados corretos ou incorretos deliberadamente e a comparação for feita entre eles mesmos.

## 3. Resultados e Trabalhos Futuros

O mecanismo foi simulado em várias situações, mostrando-se capaz de banir trabalhadores que retornam resultados incorretos e, assim, garantir maior confiabilidade aos resultados dos programas. O mecanismo se mostrou mais eficiente para situações práticas mais comuns, isto é, aquelas em que o percentual de trabalhadores suspeitos é inferior a $10\%$. Como trabalho futuro, pretende-se avaliar o mecanismo proposto em situações em que os trabalhadores suspeitos retornam resultados corretos com uma certa probabilidade, dificultando sua identificação. Além disso, serão feitas análises para o caso em que o número total de trabalhadores varia no sistema, situação que também é comum nas grades computacionais com recursos públicos.

## References

Fechner, B., Hoenig, U., Keller, J., and Schiffmann, W. (2008). Fault-tolerant static scheduling for grids. Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, Rome, Italy.

Sarmenta, L. F. G. (2001). Studying sabotage-tolerance mechanisms through web-based parallel parametric analysis and monte carlo simulation. ACM/IEEE Int'l Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia.