

Monitoramento de Aplicações Grid

Lucas Mello Schnorr, Philippe O. A. Navaux

PPGC/Instituto de Informática - Universidade Federal do Rio Grande do Sul
{lmschnorr,navaux}@inf.ufrgs.br

Introdução

Um *Grid* computacional é definido como uma infraestrutura de *hardware* e *software* que oferece acesso aos recursos computacionais de forma segura, consistente e barata [FOS 99]. Esta definição evoluiu para os conceitos de coordenação de compartilhamento de recursos e o encontro de soluções em organizações virtuais dinâmicas e multi-institucionais [FOS 2003].

Diversas aplicações para *Grid* já foram criadas. Exemplos dessas aplicações são a previsão de enchentes [HLU 2003] e a simulação dos efeitos da poluição do ar [MOU 2004]. Uma lista de aplicações para *Grid* foi analisada por Foster [FOS 2004]. Esta lista inclui aplicações cooperativas assim como aplicações para diagnósticos de motores de aeronaves e análise de dados de telescópios, por exemplo.

Uma classificação para aplicações *Grid* foi proposta por Foster [FOS 2003]. Esta classificação é dividida assim: Supercomputação distribuída, onde as aplicações utilizam o *Grid* para agregar recursos computacionais; Computação de alta produção, onde o *Grid* é utilizado para executar aplicações com tarefas independentes; Computação sob-demanda, onde as aplicações usam o *Grid* para melhorar a relação custo/benefício da utilização de recursos; Computação intensiva de dados, quando aplicações tem o foco na geração de novas informações baseadas em dados distribuídos geograficamente; e Computação colaborativa, que tem por objetivo melhorar as interações entre pessoas que estejam em diferentes localizações geográficas, possivelmente distantes.

De uma maneira geral, aplicações para *Grid* devem passar por um período de testes onde eventuais correções, otimizações e análises de desempenho devem ser realizadas [DER 2000]. Nas aplicações *Grid* classificadas como **Supercomputação distribuída**, esse período de testes é de extrema importância porque é através dele que a aplicação vai ser otimizada para melhor utilizar os recursos disponíveis em um *Grid* computacional. A forma como são feitos os testes de desempenho nessas aplicações *Grid* pode levar em conta dois tipos de dados: o monitoramento dos recursos utilizados pelas aplicações *Grid* e o monitoramento da própria aplicação. A idéia de realizar esses dois tipos de monitoramento tem como principal objetivo mostrar as influências do ambiente de execução sobre a execução de aplicações. Essa idéia surgiu no desenvolvimento de aplicações para *Cluster* de computadores [OTT 2001] e é especialmente importante em ambientes de execução do tipo *Grid*, onde os recursos que serão utilizados estão na maioria das vezes distantes geograficamente. Neste sentido, alguns trabalhos têm apresentado soluções que têm a intenção de realizar o monitoramento de aplicações *Grid*, tanto da parte de recursos que as aplicações utilizam, quanto de dados internos de uma determinada aplicação. Exemplos desse tipo de ferramenta de monitoramento são GridRM [BAK 2002], Mona-lisa [NEW 2003] e GRM/Prove [BAL 2001].

Objetivos

O objetivo geral deste trabalho é realizar uma análise da área de monitoramento de *Grid*. Os objetivos específicos deste trabalho incluem investigar as aplicações existentes para *Grid*, suas necessidades na parte de monitoramento da própria aplicação e dos recursos. Além disso, pretende-se analisar ferramentas de monitoramento de aplicações para *Grid*, incluindo suas funcionalidades, objetivos e principais características. Por fim, pretende-se identificar um conjunto de características importantes para monitoramento, de acordo com as necessidades de aplicações existentes, mas que ainda não se encontram presentes nas funcionalidades das ferramentas existentes.

Referências

- [BAK 2002] BAKER, M.; SMITH, G. Gridrm: a resource monitoring architecture for the grid. [S.l.: s.n.], 2002, London, UK, p.268–273, 2002. (Lecture Notes in Computer Science, v.2536).
- [BAL 2001] BALATON, Z.; KACSUK, P.; PODHORSZKI, N. Application monitoring in the grid with grm and prove. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCES-PART I, 2001, London, UK. **Proceedings...** Springer-Verlag, 2001. p.253–262.
- [DER 2000] DEROSE, L. A. et al. Performance issues in parallel processing systems. **Performance Evaluation: Origins and Directions**, v.1769, p.141, 2000.
- [FOS 99] FOSTER, I.; KESSELMAN, C. (Eds.). **The grid: blueprint for a new computing infrastructure**. San Francisco, CA: Morgan Kaufmann, 1999.
- [FOS 2003] FOSTER, I.; KESSELMAN, C. **The grid 2: blueprint for a new computing infrastructure**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.
- [FOS 2004] FOSTER, I.; KESSELMAN, C. The grid in a nutshell. **Grid resource management: state of the art and future trends**, Norwell, MA, USA, p.3–13, 2004.
- [HLU 2003] HLUCHY, L. et al. Flood forecasting in a grid computing environment. In: INTERNATIONAL CONFERENCE ON PARALLEL PROCESSING AND APPLIED MATHEMATICS, 5., 2003. **Proceedings...** Springer-Verlag GmbH, 2003. p.831 – 839. (Lecture Notes in Computer Science, v.3019).
- [MOU 2004] MOURIÑO, J. C. et al. Air pollution modeling in the crossgrid project. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE, 2004. **Proceedings...** Springer, 2004. p.132–139. (Lecture Notes in Computer Science, v.3036).
- [NEW 2003] NEWMAN, H. B. et al. Monalisa : a distributed monitoring service architecture. **ECONF**, La Jolla, California, v.C0303241, p.MOET001, 2003.
- [OTT 2001] OTTOGALLI, F.-G. et al. Visualisation of distributed applications for performance debugging. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL SCIENCE-PART II, ICCS, 2001. **Proceedings...** [S.l.]: Springer-Verlag, 2001. p.831–840.