

Integração do sistema AppMan de gerenciamento de aplicações para ambiente de grade com diferentes Sistemas de Gerenciamento de Recursos

(título provisório)

Tonismar Régis Bernardo

Curso de Ciências da Computação – Centro Universitário La Salle

Proposta para trabalho de conclusão

Orientador – Patrícia Kayser

tonismar.at.gmail.com

1.Introdução

Grades computacionais (*grid computing*) é uma das formas mais recentes de ambiente para processamento geograficamente distribuído, que conta com uma grande infra-estrutura de redes e pode ser empregada em troca de programas, dados e serviços. Segundo Dantas [1], pode-se dizer, também, que representa uma forma estendida dos serviços Web permitindo que recursos computacionais possam ser compartilhados. Podemos definir grades como uma plataforma computacional heterogênea distribuída geograficamente fornecendo serviços e recursos às organizações participantes da plataforma[1].

Um sistema de gerenciamento de recursos (*Resource Management System* - RMS) é a parte central de um sistema distribuído fornecendo um mecanismo de enfileiramento de tarefas, políticas de escalonamento, esquemas de prioridades e monitoramento de recursos proporcionando controles adicionais sobre inicialização, escalonamento e execução de tarefas. Também coordena a distribuição dessas tarefas entre as diferentes máquinas em uma rede [6][5][4].

Devido a heterogeneidade das grades alguns problemas são apresentados, tais como, a alocação dos nós para um grande número de tarefas, gerenciamento de dados e sobrecarga em nós de submissão. O modelo de gerenciamento de aplicações denominado GRAND (*Grid Robust Application Deployment*) [4] visa permitir um particionamento flexível e utilizar uma hierarquia de gerenciadores que realizam a submissão das tarefas. Baseado nesse modelo um protótipo, AppMan (*Application Manager*) [8], foi implementado objetivando garantir o escalonamento das tarefas bem como a

autorização e autenticação para tarefas executadas. Ele foi avaliado apresentando bons resultados referente ao gerenciamento de dados e serviços. Esse protótipo consiste em um gerenciador de aplicações que dispara e controla cada aplicação nos nós baseando-se nas informações indicadas pelo gerenciador de submissão que tem como função, além da já citada, criar e monitorar os gerenciadores de tarefas. Os gerenciadores de tarefas são responsáveis pela comunicação com o escalonador de um determinado domínio garantindo a execução remota e ordem das tarefas de acordo com a dependência de dados [4].

O modelo GRAND permitiria que qualquer sistema gerenciador de recurso fosse usado nos nós, porém o AppMan funciona unicamente com seu próprio sistema de gerenciamento de recursos. Nenhum estudo mais detalhado foi realizado até o momento de como possibilitar que o protótipo suporte a integração com diferentes RMSs.

Uma interface de aplicação (*Application Program Interface - API*) denominada DRMAA (*Distributed Resource Management Application API*) foi desenvolvida com o objetivo de facilitar a integração das aplicações para diferentes RMSs [3].

O trabalho proposto pretende avaliar se a especificação DRMAA atende as necessidades do AppMan bem como realizar a integração do AppMan ao menos com um RMS.

Na sequência deste texto serão apresentados os objetivos do trabalho separados em objetivos gerais e específicos, uma seção de justificativa abordando a importância do estudo proposto, uma explanação da motivação desse estudo seguido pela metodologia aplicada e um cronograma.

2. Objetivos

2.1. Gerais

Permitir que o AppMan exporte tarefas para nós da grade gerenciadas por diferentes RMSs e através disso proporcionar a sua integração e conseqüente utilização de recursos em outras instituições científicas.

2.2. Específicos

- Verificar se a especificação DRMAA atende todas necessidades esperadas pelo AppMan assim como confirmar a aptidão do AppMan na integração com a DRMAA;
- Integrar AppMan com pelo menos um RMS;
- Avaliar o sobrecusto (*overhead*) da solução implementada através da verificação do desempenho da integração

comparando com o escalonador atual;

3. Justificativa

O projeto GRAND, atualmente, possui uma série de cooperações informais com mais instituições de pesquisa como, UFRGS, UFRJ, UFLA e Universidade do Porto. Porém, devido a limitação atual do AppMan ao uso de RMSs, fica difícil sua utilização nos *clusters* [7] destas instituições.

Permitir que o AppMan interaja com ambientes onde um administrador tenha deixado um nó totalmente gerenciado por outro RMS através da especificação DRMAA contribuiria com o projeto de pesquisa em andamento GRAND [4]. Assim permitiria que este possa interagir e usufruir dos recursos dessas diferentes instituições de pesquisa.

Implementações existentes para Condor [2] comprovam sucesso mostrando viabilidade em prover uma interface unificada para submissão de tarefas e monitoramento através da DRMAA.

Isto, no entanto, não descarta a possibilidade de outras alternativas como as propostas nas integrações ditas multiclusters [7].

4. Motivação

Gerenciar aplicações consiste em preparar, submeter e monitorar o progresso de execução de todas as tarefas as quais compõem uma aplicação. Cada nó da grade possui seu gerenciador de recursos, os quais podem possuir inúmeros atributos. No modelo estudado foram considerados como atributos o controle exclusivo e a dependência de trabalhos [4]. Verificando esses atributos, determinadas ações podem ser agendadas para as aplicações. O GRAND respeita as políticas bem como as especificações administrativas do domínio escolhido considerando que a maioria dos *clusters* acadêmicos ou redes locais estarão prontos para serem integrados em uma grade, já possuindo usuários locais, submissão local de tarefas e um administrador de sistema, assim sendo o menos intrusivo possível nos diferentes RMSs existentes em uma grade [4].

A especificação DRMAA tem por objetivo facilitar a interface entre aplicações de RMSs com aplicações de diferentes desenvolvedores abstraindo as relações fundamentais da tarefa do RMS provendo um modelo fácil de usar (*easy-to-use*) para desenvolvedores tanto de aplicações como de RMSs encorajando, desse modo, adoção dos mesmos [3].

Como já dito anteriormente, qualquer RMS pode ser usado nos nós da grade de acordo com o modelo GRAND, porém o protótipo AppMan funciona apenas com seu próprio gerenciador de recursos [4].

Considerando as vantagens do DRMAA [2] e as questões citadas na seção 1, acredita-se motivador o estudo e desenvolvimento de uma integração com ao menos um RMS. Maiores detalhes serão esclarecidos na seção que segue.

5. Metodologia

Com o propósito de verificar a viabilidade da exportação de tarefas do AppMan para RMS diferentes, será realizado um estudo da especificação DRMAA visando o desenvolvimento desta integração. Também será feito aprofundamento no estudo do modelo GRAND onde foi desenvolvido o protótipo, bem como da implementação atual do AppMan.

O desenvolvimento deverá ser feito na linguagem Java que foi também a linguagem desenvolvida o AppMan facilitando a portabilidade.

As métricas de avaliação usadas na verificação das vantagens de um RMS diferente comparado com o escalonador do próprio AppMan serão definidas com base em estudo teórico mais aprofundado incluindo os itens citados na seção 7 presente nesta proposta.

Outros trabalhos que proporcionam integração entre diferentes RMSs serão estudados para averiguação das soluções adotadas.

Ao final deste estudo será apresentado um artigo servindo como base para o Trabalho de Conclusão de Curso (TCC) pretendido ser realizado em 2008/1. Neste momento a proposta será reformulada de acordo com os resultados obtidos no estudo aprofundado durante o processo de pesquisa, orientando assim as próximas atividades do TCC assim como os seminários que julgarem necessários.

Todo acompanhamento será feito em companhia do professor orientador através de reuniões periódicas previamente estabelecidas conforme o seguinte cronograma previsto.

6. Cronograma

1. Leituras para formulação da proposta.
2. Entrega da proposta.
3. Estudos aprofundados das aplicações e teorias envolvidas no trabalho.
4. Modelagem da solução de integração.
5. Desenvolvimento do artigo.
6. Apresentação do artigo.
7. Estudos aprofundados nas aplicações, aquisição de embasamento teórico e definição das métricas de

avaliação.

8. Desenvolvimento da monografia.
9. Palestra de andamento.
10. Implementação e testes.
11. Entrega da monografia.
12. Apresentação do TCC.
13. Reuniões com o orientador.

Ativ.	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											

7.Referências bibliográficas

[1] Dantas, Mario Antônio R., "Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais" 278pp, Axcel Books, 2005.

[2] Tröger, Peter; Gietzel, Becky. (Fevereiro de 2007) - "Open Grid Forum Documents, Condor DRMAA 1.0 Implementation - Experience Report" 9pp, <http://www.ogf.org/documents/GFD.103.pdf>. Acessado em Agosto de 2007.

[3] Rajic, Hrabir; Brobst, Roger; Chan, Waiman; Ferstl, Fritz; Gardine, Jeff; Hass, Andreas; Nitzber, Bill; Rajic, Hrabir; Tollefsrud, John. (Junho de 2004)- "Open Grid Forum Documents, Distributed Resource Management Application API Specification 1.0 (DRMAA) " 29pp, <http://www.ogf.org/documents/GFD.22.pdf>. Acessado em Agosto de 2007.

[4] Mangan, Patrícia Kayser Vargas, "GRAND: Um Modelo de Gerenciamento Hierárquico de Aplicações em Ambiente de Computação em Grade" - [Rio de Janeiro] 2006, 150pp. (COPPE - Engenharia de Sistemas e Computação/UFRJ - Universidade Federal do Rio de Janeiro, 2006. 150pp. Tese DSG.

[5] Bayucan, Albeaus; Henderson, Robert L.; Lesiak, Casimir; Mann, Bhroam; Proett, Tom; Tweten, Dave. "Portable Batch System (PBS), Externa Reference Specification", Numerical Aerospace Simulation Systems Division NASA Ames Research Center . 281pp <http://www.nas.nasa.gov/Software/PBS/pbsdocs/ers.ps>. Acessado em Agosto de 2007.

[6] High Throughput Computing (CONDOR), "An Overview of the Condor System", <http://www.cs.wisc.edu/condor/overview>. Acessado em Agosto de 2007.

[7] Navaux, Philippe Oliver Alexandre; Stringhini, Denise; Schlemer Elgio; Oliveira, Fábio A. D.; Barreto, Marcos E., "Projeto Multicluster, Distributed Execution and Communication Kernel (DECK)", 2pp. - <http://www.inf.ufrgs.br/~avila/download/Navaux:PM-ERAD01.ps.gz>. Acessado em Agosto de 2007.

[8] Vargas, Patrícia Kayser; Dutra, Inês de Castro; Geyer, Cláudio F. R., "Hierarchical Resource Management and Application Control in Grid Environments" - Relatório Técnico ES-608/03, COPPE/Sistemas - UFRJ, 2003, 8pp. - <http://www.cos.ufrj.br/grand/pdfs/es60803.pdf>. Acessado em Agosto de 2007.