

Uma ferramenta para execução de simulações em larga escala

André Detsch*, Luciano P. Gaspary,
Gerson Geraldo H. Cavalheiro

Programa Interdisciplinar de Pós-graduação em Computação Aplicada
Centro de Ciências Exatas e Tecnológicas
Universidade do Vale do Rio dos Sinos
{detsch, paschoal, gersonc}@exatas.unisinos.br

Palavras-chave: Simulação, *bag-of-tasks*, *grid* computacional.

Introdução

A execução de simulações computacionais é uma das mais importantes formas de avaliação de novas tecnologias. Entretanto, a quantidade de processamento necessário para realizar estas execuções em geral é significativa, tanto pelo fato de que uma simples execução pode levar várias horas, tanto pelo fato de que, muitas vezes, são necessárias milhares de execuções (variando-se parâmetros de entrada e sementes aleatórias) para que a avaliação seja conclusiva. Existe uma vasta gama de estudos que visam permitir que a simulação de um determinado caso (um conjunto de parâmetros e uma semente aleatória) possa ser executada de forma paralela (por exemplo, [2]). Entretanto, tais métodos envolvem, em geral, adaptar a forma de programação, adicionando complexidade.

Por outro, observa-se que o conjunto de execuções de experimentos que tipicamente envolve a realização de um estudo aprofundado é trivialmente paralelizável, podendo se enquadrado como uma aplicação *bag-of-tasks* ([1]). Este artigo apresenta uma ferramenta que atua neste nível, estabelecendo uma plataforma similar a um *grid* computacional ([1]) para execução de simulações em larga escala. A seguir, são apresentadas as duas camadas que compõem esta ferramenta: a linguagem e o executivo. Por fim, são tecidas algumas considerações finais.

Linguagem

Um primeiro nível da ferramenta se propõe a oferecer uma interface para descrição das simulações a serem realizadas. Com este objetivo, foi desenvolvida uma linguagem própria, estruturada de acordo com as características existentes neste tipo de aplicação.

A linguagem foi desenvolvida de forma a reduzir ao máximo a redundância na descrição dos experimentos. Para tanto, ela provê possibilidade de estruturação hierárquica de código (através de escopos) e capacidade de definição de macros. Outra característica importante é a existência de um *front-end* para geração de gráficos (desenvolvido em paralelo) que compartilha a mesma sintaxe, permitindo uma integração automática entre definição dos experimentos e observação dos resultados. A Figura 1 exemplifica o uso da linguagem na simulação do comportamento de uma rede com ns-2 (*Network Simulator*).

```

parameters protocol bytes packet_size seed
defaults TCP 20000 1000 0
macro SEMENTES 100 # definição de uma macro
number_of_bytes 10000 20000 # variação de parâmetro
seed range(0,!SEMENTES) # podem ser usados comandos python
comandbase ns experimento.tcl # comando que será chamado
{
  protocol TCP
  dtask } # gera as tarefas correspondentes
{
  protocol UDP
  packet_size 1000 2000 5000
  dtask }

```

Figura 1: Exemplo de especificação de experimentos

Executivo

O ambiente de execução se baseia em nodos independentes, com processador e memória próprios, interconectados através de uma rede. Atualmente, a implementação do ambiente se baseia no uso de NFS (*Network File System*): é criada uma estrutura de diretórios (compartilhada entre as máquinas) que armazena as tarefas. Cada tarefa, responsável pela execução de uma simulação, é descrita em um arquivo, cujo nome é um identificador único.

A adição de um nó ao ambiente se dá através do acionamento de um *daemon* que monitora o diretório **Waiting**, esperando que alguma tarefa seja gerada. Uma vez que alguma tarefa seja localizada, o arquivo referente a ela é movido para o diretório **Running** e o comando descrito dentro do arquivo da tarefa é executado, tendo suas saídas, padrão e de erro, redirecionadas para arquivos próprios dentro de **Output** e **Error**. Uma vez que a execução da tarefa esteja concluída, o arquivo é movido para o diretório **Done** e o *daemon* volta a monitorar o diretório **Waiting**, em busca de novas tarefas.

Considerações Finais

Apesar de ainda estar em fase de desenvolvimento, a ferramenta tem se mostrado bastante útil e prática na execução de simulações. Ela será amplamente utilizada no decorrer do mestrado, assim como já foi utilizada no desenvolvimento de outras dissertações.

Está sendo desenvolvido um conjunto de avanços que possibilitarão a especificação mais detalhada da forma de execução das tarefas, ampliando a aplicabilidade da ferramenta. Uma vez implementados, tais avanços permitirão, entre outros, especificar critérios de escalonamento de tarefas, tais como localidade de execução, inter-dependências e requisitos mínimos de hardware.

Referências

- [1] CIRNE, Walfredo. “Grids Computacionais: Arquiteturas, Tecnologias e Aplicações” em 3a Escola Regional de Alto Desempenho (ERAD 2003), Santa Maria, Janeiro 2003.
- [2] “UCLA Parsec Programming Language”, <http://may.cs.ucla.edu/projects/parsec/>