

Máquinas NOW heterogêneas*

Roni Antônio Dall Orsoletta**, Fábio Arezi, Marcelo Trindade Rebonatto,
Marcos José Brusso

Universidade de Passo Fundo

BR 285 Caixa Postal 611, Passo Fundo - RS, CEP 99001-970 - Fone:(54) 316.8354

{43945, 42246}@lci.upf.tche.br, {rebonatto,brusso}@upf.br

Introdução

A rápida evolução da computação e as transformações que proporcionaram contribuíram para o surgimento prático da computação paralela que possui uma vantagem de poder proporcionar um alto desempenho. Porém, existem fatores como o alto custo de aquisição e manutenção, e dependência do fabricante que diminuem esta vantagem [LIN 98], p. 1.

Segundo Hwang, processamento paralelo é uma forma eficiente de processar informações, a qual enfatiza a exploração de eventos concorrentes na computação de um processo [HWA 98]. Neste contexto, um computador paralelo é uma coleção de computadores, tipicamente do mesmo tipo, interconectados de certa forma que possibilite uma coordenação de suas atividades, atuando cooperativamente para resolver um problema [GOU 99], p. 14.

Um dos tipos de máquinas paralelas que nos últimos anos vem ganhando destaque são as máquinas NOW (Network of Workstation). Elas são geralmente programadas utilizando bibliotecas para a exploração do paralelismo que fornecem rotinas para a configuração de máquinas virtuais paralelas. As máquinas NOW são tipicamente constituídas de computadores de arquitetura homogênea, quer seja em relação ao hardware empregado (arquitetura e capacidade para multiprocessamento) quer seja em relação ao software (sistema operacional).

Um fato comum nos laboratórios dos cursos de Ciência da Computação é o uso de máquinas com diferentes arquiteturas e S.O. O emprego destas na composição de máquinas NOW pode contribuir para a utilização do processamento paralelo, uma vez que a área necessita muitas unidades processadoras. Na prática, nem sempre as instituições possuem um número razoável de máquinas homogêneas para serem dedicadas a uma NOW. Assim, pode-se utilizar as máquinas que em grande parte do tempo estão sem uso e contribuir para a execução de aplicações paralelas.

Máquinas empregadas

Como um dos objetivos deste trabalho era proporcionar um ambiente paralelo composto de máquinas heterogêneas, foram obtidas máquinas na instituição e também

* Apoio Universidade de Passo Fundo

** Bolsista PIBIC/UPF

através de convênios firmados com empresas privadas. A Tabela 1 mostra a configuração de máquinas disponíveis, com seus sistemas operacionais.

Tabela 1 – Máquinas utilizadas

Máquina	Arquitetura	No. de processadores	S.O.
HP-BRIO	Intel/CISC	1	Microsoft Windows 98
HP-BRIO	Intel/CISC	1	Red Hat Linux
IBM-PCSERVER	Intel/CISC	2	SUN Solaris
IBM-PCSERVER	Intel/CISC	2	Red Hat Linux
IBM-PCSERVER	Intel/CISC	2	Microsoft Windows NT
SUN-SPARC10	SPARC/RISC	1	SUN Solaris
ZYX-Axil 311	SPARC/RISC	1	Red Hat Linux
iMac	PowerPC/RISC	1	Yellow Dog Linux
HPD Class 9000	HP PA/RISC	1	HP-UX

A forma de interligação das máquinas é uma rede Ethernet, utilizando a estrutura de rede dos laboratórios de pesquisa do curso de Ciência da Computação. Sabe-se que a rede de comunicação pode influenciar decisivamente no tempo de execução de aplicações paralelas em ambientes NOW, porém nem todas as máquinas podem sofrer atualização da rede para uma de melhor desempenho.

Software utilizado

Para validar o ambiente paralelo heterogêneo foram desenvolvidas aplicações paralelas de teste utilizando as bibliotecas PVM e MPI. As aplicações tinham por objetivo apenas validar o ambiente, uma vez que não foram desenvolvidos algoritmos para resolver problemas específicos.

A execução das aplicações paralelas utilizando PVM e MPI necessitam que o serviço de rsh esteja presente e habilitado. Esta é uma tarefa direta nas máquinas com S.O. Unix-like, porém nas máquinas com sistemas operacionais Windows, foi necessária a instalação de programas a fim de permitir a execução remota de aplicações. Estes programas podem ser acessados através da internet no site <http://www.denicomp.com/>, com versões disponíveis para Windows NT e Windows 9x.

No desenvolvimento do trabalho, foi utilizado o compilador gcc nas máquinas Unix-like. Já nas máquinas com S.O. Windows, foi utilizado o pacote Visual C++, uma vez que as bibliotecas de funções disponíveis tanto para o PVM quanto para o MPI na versão para Windows, não disponibilizam arquivos compatíveis com o gcc.

Compilação de aplicações paralelas no ambiente heterogêneo

Dado que em um ambiente não homogêneo a portabilidade binária não é possível, é necessária a compilação individual dos softwares desenvolvidos para cada uma das combinações de processador/SO disponível. Esta tarefa se torna mais trabalhosa a medida em que novos computadores são adicionados à máquina paralela virtual. Para minimizar o tempo despendido na tarefa de compilação dos programas foi desenvolvido um software, denominado HetCompile, que automatiza este trabalho.

O HetCompile obtém a lista de máquinas para as quais o programa deve ser distribuído em um arquivo de configuração (*hetcompile.conf*). Para cada máquina

relacionada, este arquivo especifica também o diretório remoto para onde a aplicação deve ser enviada, dado que a estrutura da árvore de diretórios pode ser diferente em cada instalação. O HetCompile envia, então, o código-fonte da aplicação a ser distribuída, que é portátil, para todas estas máquinas, através de *rcp* (*remote copy*) e dispara a compilação remota através de *rsh* (*remote shell*). Desta forma, o programa é compilado localmente em cada uma das máquinas, gerando-se código binário compatível com a respectiva plataforma. A Figura 1 ilustra o funcionamento do software implementado.

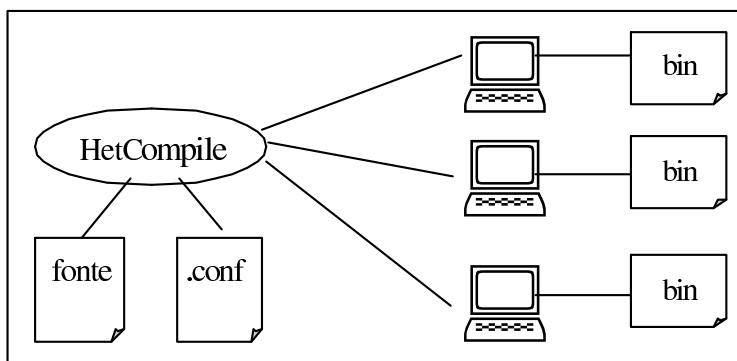


Figura 1 – Funcionamento do HetCompile

A versão atual (1.0.1)¹ aceita a compilação de aplicações MPI em sistemas operacionais Linux e Solaris. Versões futuras deverão suportar o S.O. Microsoft Windows.

Resultados obtidos

Para validar a utilização da máquina NOW heterogênea, foram utilizadas 3 aplicações experimentais, duas desenvolvidas utilizando MPI e outra utilizando PVM. Ambas as aplicações em MPI realizam cálculos de números primos por ser este cálculo oneroso em termos de tempo de processamento, sendo implementadas em C uma utilizando um algoritmo mestre-escravo e outra divisão e conquista. A aplicação escrita em PVM realiza um merge-sort utilizando um algoritmo divisão e conquista, também implementada em C.

Para a obtenção dos resultados, foram compostas diversas combinações de máquinas para compor a máquina virtual paralela, incluindo três, quatro, cinco e seis processadores. A composição das máquinas paralelas em relação aos processadores utilizados pode ser visualizada na Tabela 2, sendo priorizados os testes com combinações de arquiteturas/S.O. disponíveis.

Tabela 2 – Composição das máquinas utilizadas em testes

Legenda	Composição
Seq	HP BRIO c/ Linux
3	2 HP BRIO c/ Linux; 1 SUN SPARC10
4	2 HP BRIO c/ Linux; IBM PCSERVER c/ Solaris
5	2 HP BRIO c/ Linux; IBM PCSERVER c/ Solaris; 1 SUN SPARC10
6	2 HP BRIO c/ Linux; IBM PCSERVER; c/ Solaris; 1 iMac; 1. SUN SPARC10

Ressalta-se que a máquina HP-UX encontra-se em fase de instalação e as com S.O. Windows não foram incluídas nos testes de desempenho pela não disponibilização atual do HetCompile. Porém, as máquinas Windows

¹ Disponível, sob licença GNU GPL, em <http://inf.upf.br/~compad>

funcionaram corretamente nos testes de montagem de máquina virtual e execução de aplicação paralela que realiza um merge -sort.

A Figura 2 mostra os resultados do tempo de execução de um algoritmo mestre-escravo, escrito em MPI para cálculo de números primos, executado no ambiente descrito na Tabela 2. Ressalta-se que os resultados obtidos pelo algoritmo divisão e conquista para cálculos de números primos resultaram em tempos de execução superiores ao mestre-escravo, devido a estrutura de funcionamento do algoritmo que não consegue tirar proveito de máquinas mais rápidas, nivelando todas pela máquina mais lenta.

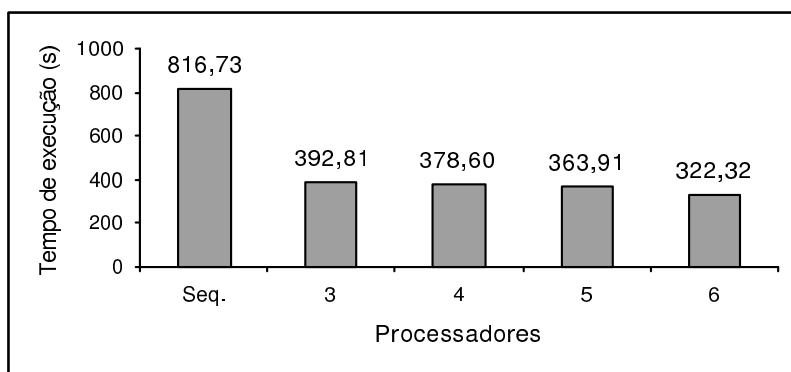


Figura 2 – Tempos de execução de algoritmo mestre-escravo

pequeno ganho pode ser devido a heterogeneidade ou devido ao desempenho diferenciado de cada máquina, uma vez que os modelos de máquinas utilizados são antigos.

Nota-se uma redução no tempo de execução do algoritmo paralelo em relação ao sequencial, porém uma análise entre as execuções paralelas mostra um pequeno ganho em relação ao tempo de execução com a adição de processadores. Este

Considerações Finais

O trabalho mostrou ser viável a utilização de máquinas não homogêneas em relação a arquitetura e S.O. na composição de máquinas paralelas NOW, possibilitando uma redução no tempo de execução das aplicações. A grau de dificuldade para a programação e configuração de máquinas paralelas heterogêneas é maior em relação as homogêneas, principalmente quando utiliza-se máquinas com S.O. Windows, onde programas específicos necessitam ser instalados e a tarefa de compilação ainda tem de ser automatizada. No futuro, deverão ser realizados testes avaliando também o desempenho com todas as arquiteturas/S.O. disponíveis.

Referências

- [LIN 98] LINHALIS, Flávia; FIATS, Mayb Iara. **PVM e MPI**. Capturado em 16 ago. 2001. Online. Disponível na Internet através de ftp anônimo em <ftp://ftp.cenapadne.br/pub/cenapad/mmpi>
- [HWA 98] HWANG, Kai; XU, Zhiwei. **Scalable Parallel Computing: Technology, Architecture, Programming**. Boston: McGraw-Hill, 1998. p.3-90, p.453-564 e p.318-326.
- [GOU 99] Peter Carvalhal et. al. **PARALELISMO: Algoritmos e complexidade**. Porto Alegre: UFRGS, Instituto de Informática, 1999. 106p. RP-306.