Softwares de Memória Compartilhada Distribuída

Rodrigo Cesar de Moura[1], Marcelo Trindade Rebonatto[2]

Universidade de Passo Fundo
Passo Fundo, RS, Brasil
E-mail: {rmoura@pas.matrix.com.br, rebonatto@upf.tche.br}

Introdução

A memória compartilhada distribuída fornece ao programador a sensação de um único espaço de endereçamento virtual, que é compartilhado entre uma rede de processadores que não compartilham memória física. Conforme a memória local é atualizada, as modificações são propagadas para os outros processadores, de forma que todos mantenham uma versão consistente [BER 91]. A estrutura de um sistema DSM é mostrada na figura a seguir.

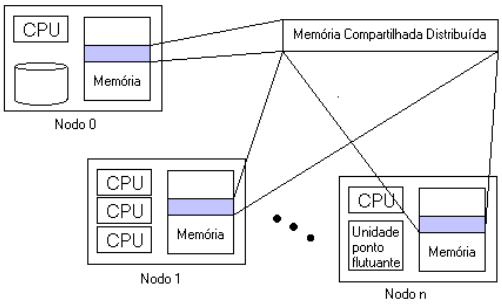

Figura 1 Estrutura de um sistema DSM

Conforme a fig. 1, a organização básica não é muito diferente de um multicomputador que utiliza troca de mensagens. Como se pode ver, esta geralmente envolve um conjunto de nodos ou clusters conectados por uma rede de interconexão escalável.

Esse texto descreve um estudo realizado sobre os softwares de memória compartilhada distribuída disponíveis na Internet. Foi dado ênfase às bibliotecas que proporcionam DSM e podem ser executadas sobre Linux.

[1] Acadêmico do curso de Ciência da Computação.
[2] Professor do curso de Ciência da Computação, UPF, Cx Postal 611, CEP 99001-970

Softwares DSM

Os dez softwares pesquisados foram os seguintes: TreadMarks, Millipede, Quarks, Shasta, Midway, Cilk, CVM, CRL, JIAJIA e Filaments, sendo que os dois últimos foram estudados profundamente, instalados e testados. Os cinco primeiros foram estudados sem aprofundamento devido à impossibilidade de instalação no ambiente montado.

Um programa criado em Cilk consiste de uma coleção de procedimentos onde cada um é quebrado em uma seqüência de processos não bloqueantes. Na terminologia usada pelo Cilk, um processo é a maior seqüência de instruções que termina com as expressões *spawn*, *sync* ou *result* [MIT 00]. O CVM suporta múltiplos protocolos e modelos de consistência. Roda na maioria dos sistemas tipo UNIX e foi criado especificamente como uma plataforma de experimentação de protocolos [KEL 01]. Os programas CRL compartilham dados através de regiões. Cada região é arbitrariamente dimensionada em áreas contíguas de memória. O programador define regiões e inclui anotações para delimitar o acesso a essas determinadas regiões. As regiões são cacheadas nas memórias locais dos processadores[JOH 01].

Os softwares JIAJIA e Filaments foram escolhidos para serem instalados e testados. Isso deve-se ao fato de atenderem aos requisitos do ambiente escolhido para a realização dos testes. Isto é, ambos rodam no sistema operacional Linux, estão disponíveis para *download* via Internet e a instalação ocorreu sem maiores problemas.

JIAJIA

É um software que utiliza protocolo de coerência de *cache* baseado em *lock* para consistência de entrada. O protocolo é baseado em *lock* porque ele elimina totalmente diretórios e todas as ações relacionadas com coerência através de notícias de escritas são mantidas no *lock*. Comparado ao protocolo baseado em diretório, o protocolo baseado em *lock* é mais simples e, conseqüentemente, mais eficiente e escalável [HU 99]. O software combina as memórias físicas de múltiplos processadores para formar um grande espaço compartilhado, onde cada página compartilhada tem seu nodo-inicial e os inicias das páginas compartilhadas são distribuídas através dos *hosts*. Com essa organização de memória, o tamanho do espaço compartilhado pode ser tão grande quanto a soma da memória local de cada máquina

Filaments

Apresenta-se sob duas diferentes implementações. Filamentos Compartilhados (SF) que roda em multiprocessadores de memória compartilhada e Filamentos Distribuídos (DF) que roda em multicomputadores de memória distribuída.

Permite a escrita de programas paralelos usando um processo para cada unidade independente de trabalho e uma memória compartilhada global. Suporta um espaço de endereçamento global através de um software de memória compartilhada distribuída, sendo similar a outros softwares DSM implementados sobre Unix utilizando chamadas de sistemas e *sockets*. Muitos filamentos podem executar em cada nodo, permitindo que a latência de falta de páginas seja aliviada [LOW 98].

Usa o modelo computacional "Múltiplos Dados Único Programa" (SPMD), em que cada nodo executa o mesmo código mas referencia um diferente subconjunto de elementos de dados, isso significa que todos os filamentos de um mesmo nodo executam o mesmo código.

Tabela 1 Comparativo entre os SDSM

Nome	Custo	Linguagem	S.O.	MP*
Cilk	Livre	C	Linux, Solaris, Irix e OSF.	Objetos compartilhados
Filaments	Livre	C	Linux, Solaris e Irix	Variáveis compartilhadas
CVM	Livre	C	SunOS 4.1 e 5.5, AIX 4.1,Digital Unix 3.2D-1	Variáveis compartilhadas
CRL	Livre	C	SunOS 4.1.3, Unix	Regiões
JIAJIA	Livre	C, Fortran	Unix, Solaris 2.4, AIX 4.1, Linux 2.0, SunOS 4.1e Windows NT.	Páginas
TreadMarks	$US1000	C, C++, Java e Fortran.	FreeBSD 2.1R e Linux.	Páginas
Millipede	Livre	C,C++	Windows NT 4.0 e Windows 2000.	Páginas
Quarks	Livre	C, C++	SunOs 4.1, HP-UX, Irix 5.2.	Regiões
Shasta	Livre	-	-	Variáveis compartilhadas
Midway	Livre	C	Mach 3.0	Páginas

*Modelo de programação

Testes e Resultados

Os softwares JIAJIA e Filaments foram instalados em um ambiente usando duas máquinas, sendo uma delas o servidor NFS. O sistema operacional utilizado foi o Conectiva Linux 6.0. possuindo *kernel* com versão 2.2.17-14cl e compilador *gcc* de versão 2.95.2-7cl. As máquinas possuem processador Intel Celeron 500 MHz com 64 MB de memória RAM.

Com a finalidade de realização de testes, tanto de funcionamento como de desempenho das duas ferramentas instaladas, uma aplicação de multiplicação de matrizes foi desenvolvida. Para garantir que os resultados processados em paralelo pelas aplicações retornassem corretamente, partes da matriz do programa seqüencial foram comparadas com as partes equivalentes dos programas paralelos. Os resultados obtidos apresentaram-se corretos, sendo que os valores dos tempos de execução mostrados são oriundos de 5 repetições.

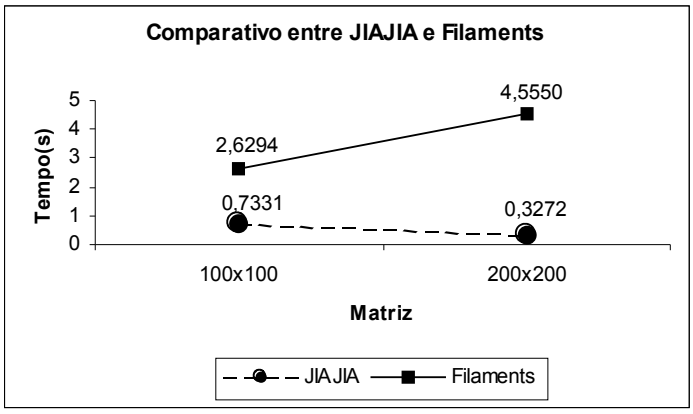

Figura 2 Gráfico comparativo entre JIAJIA e Filamens

Na fig. 2, pode-se ver que o Filaments não obteve uma aceleração na execução das matrizes em paralelo. Ou seja, houve um retardo de tempo em relação ao programa seqüencial. Já o JIAJIA, obteve um bom desempenho mostrando estabilidade em todas as repetições com um *speedup* e uma eficiência altos.

Conclusões

Com o estudo dos SDSM, percebeu-se que alguns deles ainda são muito instáveis, pois no mesmo ambiente, houve uma grande diferença de desempenho entre os dois softwares instalados. Nota-se, também, a diferença da facilidade de programação entre um e outro, sendo que o JIAJIA possui uma interface de programação mais enxuta e intuitiva que o Filaments. Como trabalhos futuros pode-se testar o funcionamento dos sistemas em um maior número de processadores, já que só foi possível instalá-los em um ambiente com duas máquinas. Além disso, é interessante um estudo comparativo entre os SDSM e os softwares de troca de mensagem, a fim de analisar a facilidade de instalação, programação e o desempenho.

Referências

[BER 91] BERSHAD, B. N.; ZEKAUSKAS, M. J. *Midway*: Shared Memory Parallel Programming with Entry Consistency for Distributed Memory Multiprocessors**.** CMU Report CMU-CS-91-170, Set. 1991.

[HU 99] HU, Weiwu; SHI, Weisong; TANG, Zhimin. *JIAJIA User's Manual*. Alberta: 1999.

[LOW 98] LOWENTHAL, David; FREEH, Vincent. *Filaments Programmer's Manual*. Georgia: 1998.

[MIT 00] MIT LCS SUPERTECH GROUP. *Cilk 5.3.1 – Reference Manual*. Massachusetts: 2000.

[KEL 01] KELEHER, Peter J. *Coherent Virtual Machine (CVM)*. Disponível em: < http://www.cs.umd.edu/projects/cvm/>. Acesso em: 28 abr. 2001

[JOH 01] JOHNSON, Kirk. *CRL version 1.0*. Disponível em: < http://www.pdos.lcs.mit.edu/crl/>. Acesso em: 29 abr. 2001