

PROJETO DE UMA ESTAÇÃO PROLOG

Claudio L. de Amorim
Valmir C. Barbosa
Edil S. T. Fernandes

Programa de Engenharia de Sistemas e Computação
COPPE/UFRJ
Caixa Postal 68511
21945 Rio de Janeiro - RJ

Sumário

Este trabalho contém a apresentação de um projeto cujo objetivo primário é a investigação da compilação e uso de Prolog Concorrente. O projeto apresenta problemas em diversas disciplinas, incluindo Programação Lógica e Processamento Distribuído. A intenção é que seja utilizada uma arquitetura em hipercubo baseada no microprocessador Transputer e na linguagem OCCAM. Problemas a serem investigados incluem a compilação eficiente de programas em Prolog Concorrente e o balanceamento de carga em sistemas distribuídos.

1. Introdução

O Projeto Estação Prolog é resultante da aproximação que vem ocorrendo entre os membros dos grupos de interesse em processamento paralelo/distribuído e Programação Lógica na COPPE.

A idéia central é construir um protótipo de um computador paralelo de alto desempenho com topologia hipercúbica, que servirá de veículo para diversas atividades de pesquisa em andamento. Em particular, contempla-se o desenvolvimento de um compilador Prolog Concorrente.

A questão de alto desempenho torna-se fundamental para o grupo de processamento paralelo, considerando que as implementações convencionais de interpretadores Prolog atingem cerca de 2 KLIPS (milhares de inferências lógicas por segundo) num VAX-780, o que está bem distante das GLIPS (bilhões de LIPS), segundo as estimativas do FGCS [1].

Neste contexto, o trabalho de Warren [2] representa um marco referencial. O conjunto virtual de instruções, quando microprogramado, permite que programas Prolog atinjam um

desempenho da ordem de 400 KLIPS. A partir desses resultados, vários projetos de pesquisa começaram a investigar versões paralelas de programação lógica *lato sensu*, embora versões concorrentes de Prolog continuem proeminentes [3].

Pretendemos neste trabalho fazer uma introdução ao projeto, de modo a compartilhar nossas idéias e apresentar alguns resultados preliminares.

2. Arquitetura Básica do Projeto

O trabalho pioneiro do Caltech [4] mostrou que a arquitetura hipercúbica com 32 a 128 nodos exibe uma eficiência média de 85% na execução de uma variada classe de problemas científicos. Há atualmente uma intensa atividade de pesquisa na área, demonstrada pelos recentes simpósios sobre computadores hipercúbicos.

Por outro lado, computadores paralelos baseados em múltiplos barramentos compartilhados por vários processadores com memórias local e global apresentam-se como opção a computadores hipercúbicos [5]. Queremos afirmar com isso que a nossa escolha pela arquitetura hipercúbica é devida à maior facilidade de construção do protótipo (exige menos desenvolvimento de hardware), e também à conveniência da tecnologia Transputer, que permite o desenvolvimento de programas concorrentes através da linguagem OCCAM em um IBM PC. Torna-se com isso possível uma maior concentração na pesquisa em processamento distribuído.

A arquitetura hipercúbica a ser implementada é apresentada na Figura 1. Suas características principais são:

- 16 nós, onde cada nó contém um Transputer.
- Cada nó é conectado a 4 outros nós.
- A comunicação entre nós é realizada por troca de mensagens.
- Não existe memória global. Cada nó dispõe de 2 Mbytes de memória local.

O Microprocessador Transputer (IMS T414-20)

O Transputer [6] é um chip VLSI contendo memória, processador e elos de comunicação para conexão direta com outros Transputers.

- Processador: tempo de ciclo = 50 ns, 10 MIPS (RISC), 32 bits, 2 Kbytes SRAM, taxa de transferência = 80 Mbytes/s, 4 elos de comunicação, memórias multiporta.
- Memória Externa: 4 Gbytes, taxa de transferência = 25 Mbytes/s, 32 bits.
- Elos: 4, full-duplex, seriais até 20 Mbits/s.

- Tecnologia: 1,5 μ CMOS, 150.000 dispositivos.

3. A Linguagem OCCAM

OCCAM [7] é uma linguagem que permite que sistemas sejam descritos como uma coleção de processos concorrentes que se comunicam através de canais pré-definidos. O principal objetivo da linguagem é permitir sua implementação direta numa rede de Transputers, podendo expressar algoritmos concorrentes. Existe uma relação direta entre processos concorrentes e Transputers e entre canais em OCCAM e elos nos Transputers. OCCAM pode ser considerada como uma linguagem de montagem para esses sistemas.

A definição e uso dos procedimentos em OCCAM segue regras semelhantes às regras de escopo em Algol, com canais, variáveis e constantes como parâmetros. O corpo de um procedimento pode ser um processo seqüencial ou paralelo. Indentação é usada para indicar a estrutura do programa.

Existem três processos primitivos que permitem a formação de construções: atribuição de uma expressão e a uma variável v ($v := e$), escrita de uma expressão e no canal c ($c!e$) e leitura de uma expressão do canal c para a variável v ($c?v$).

As construções são seqüenciais (SEQ), condicionais (IF), paralelas (PAR) e alternativas (ALT). Estes são também processos que podem ser usados como componentes de outras construções. Segue um exemplo de uma construção paralela:

```
CHAN c:
PAR
  WHILE TRUE
    VAR ch:
    SEQ
      in?ch
      c!ch
  WHILE TRUE
    VAR ch:
    SEQ
      c?ch
      out!ch
```

Este programa corresponde ao esquema da Figura 2.

4. Compiladores Prolog

Numa fase preliminar do projeto, foi feito um estudo sobre a implementação de compiladores Prolog, levando à elaboração da tese apresentada em [8]. A partir desse trabalho

e com a recente instalação do sistema de desenvolvimento do Transputer no laboratório, iniciamos a implementação da máquina de Warren.

5. Balanceamento de Carga em Sistemas Distribuídos

Um dos problemas gerados pela paralelização do processo de inferência lógica no hipercubo é o problema de balanceamento de carga entre os diversos nós. Há basicamente dois níveis em que este balanceamento pode ser atacado. O primeiro faz com que o próprio compilador se encarregue da distribuição de tarefas entre os nós. O segundo consiste de uma abordagem do problema de forma dinâmica, em que o balanceamento é realizado em tempo de execução. O estudo deste problema faz parte do projeto.

6. Pesquisas Relacionadas

Muitos problemas relacionados com o projeto estão sendo investigados, entre eles:

- Processamento de Linguagem Natural: processamento de um subconjunto da língua portuguesa.
- Sistemas Especialistas: sistema de apoio à interpretação de leis jurídicas.
- Compilador ACTUS: geração de código para a arquitetura hipercúbica da linguagem ACTUS para processamento numérico em grande escala.

7. Conclusões

Com a aquisição e instalação em março deste ano de um sistema de desenvolvimento do Transputer, o projeto da estação Prolog inicia uma fase de "laboratório", ou seja as idéias começam a ser postas em prática com a participação dos alunos da pós-graduação. Através de seminários regulares, os participantes das pesquisas em arquiteturas paralelas, programação lógica, processamento distribuído, linguagem natural e sistemas especialistas começam a apresentar trabalhos relacionados ao projeto Prolog Concorrente.

Referências

- [1] T. Moto-okã, "Overview to the fifth generation computer system project", Proc. Tenth Int. Symp. on Computer Architecture, pp. 417-422, Junho de 1983.
- [2] D. H. D. Warren, "Implementing Prolog — Compiling Predicate Logic Programs", Research Reports nos. 39 e 40, Department of Artificial Intelligence, University of Edinburgh, 1977.

- [3] P. C. Treleaven *et al.*, "Computer Architectures for Artificial Intelligence", TR 119, Department of Computer Science, University College, Março de 1986.
- [4a] C. L. Seitz, "The Cosmic Cube", Communications of the ACM, Vol. 28, No. 1, pp. 22-33, Janeiro de 1985.
- [4b] G. Fox, "The Performance of the Caltech Hypercube in Scientific Calculations — A Preliminary Analysis", CALT-68-1298, Abril de 1985.
- [5] D. B. Davis, "Parallel Computers Diverge", High Technology, Fevereiro de 1987.
- [6] INMOS Ltd., "The Transputer Family", Março de 1986.
- [7] D. May, "OCCAM", IFIP Working Group 2.4 (Systems Implementation Languages), Conference Proceedings, Junho de 1984.
- [8] P. M. V. Lima, "Implementação de Compiladores Prolog", Tese de Mestrado, Programa de Sistemas, COPPE/UFRJ, 1987.

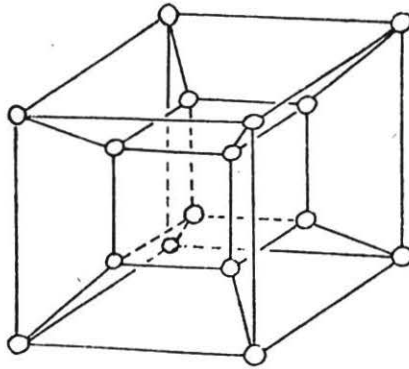


Figura 1. Um Hiper cubo de grau $n = 4$

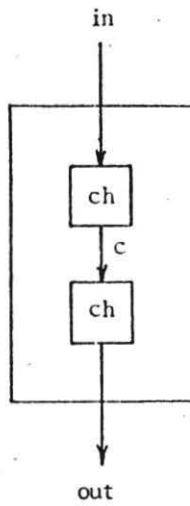


Figura 2. Exemplo de uma construção paralela