

Projeto de Implementação dos Protocolos de Transporte no Modelo TCP/IP em Placas FPGAs

Alisson Roggia Ceolin¹, Cristiano Bonato Both¹

¹Departamento de Informática - Universidade de Santa Cruz do Sul (UNISC) – RS -
Brasil

{alissonc, cboth}@unisc.br

Abstract: *This article describes a hardware model project implementation TCP/IP (Transport Control Protocol/Internet Protocol) transport protocol, making possible the exchange messages between FPGAs (Field Programmable Gate Arrays). These plates make possible the hardware programming for specifies application, normally aiming high performance. It is credited that with a set of FPGAs linked can be configured a new model of aggregates and to acquire an equal or superior performance the conventional aggregates.*

Resumo: Este artigo descreve um projeto de implementação dos protocolos de transporte do Modelo TCP/IP (*Transport Control Protocol/Internet Protocol*) em hardware, possibilitando a troca de mensagens entre FPGAs (*Field Programmable Gate Arrays*). Estas placas possibilitam a programação do hardware para uma aplicação específica, normalmente visando alto desempenho. Acredita-se que com um conjunto de FPGAs interligadas pode-se configurar um novo modelo de agregados e adquirir um desempenho igual ou superior a agregados convencionais.

1. Introdução

Agregados são conjuntos de computadores pessoais (PCs), dedicados ou não, e interligados por uma rede de alta velocidade utilizados para fins específicos. Esta arquitetura apresenta uma boa relação custo/benefício, embora existam sub-utilizações de alguns dispositivos de entrada e saída e algumas camadas de softwares ocasionando perda de desempenho.

Uma arquitetura de computadores dedicada para extrair o máximo desempenho, deve possuir a menor quantidade de camadas de softwares e a aplicação deve ser voltada para aquele hardware [Dehon 1999]. Um modelo que se enquadra nesta arquitetura são as placas FPGAs, que podem ser uma alternativa para problemas que necessitam um alto processamento.

Podem ser desenvolvidos agregados de FPGAs reconfiguráveis para adquirir desempenho em aplicações distribuídas, mas para que isso ocorra, estas FPGAs devem possuir meios de transmissão de dados, isto é, os protocolos de comunicação devem ser implementados em hardware.

Este artigo aborda a implementação dos protocolos de transporte em um agregado de FPGAs, sendo organizado com as seguintes seções: na seção 2 descreve-se o projeto ACR para agregados de computadores reconfiguráveis, na seção 3 ocorre uma breve descrição sobre os protocolos de transporte, na seção 4 é relatado o estágio atual

da implementação dos protocolos de transporte na placa FPGA e finalmente na seção 5 encontram-se as conclusões deste projeto.

2. O projeto ACR - Agregados de Computadores Reconfiguráveis

O projeto ACR visa utilizar a lógica encontrada em dispositivos reconfiguráveis para a obtenção de desempenho em aplicações distribuídas específicas. Sua motivação provém da possibilidade de construir agregados de processadores dedicados à resolução de cálculos que necessitam de alto poder de processamento. Enviando instruções diretamente para os processadores não há necessidade de tratamentos de interrupções adicionais de dispositivos de E/S, assim como, há redução de *overhead* em camadas de softwares, como por exemplo, pode-se citar o próprio sistema operacional, encontrados em agregados convencionais [Rose 2003].

Este modelo de agregados com FPGAs reduz o espaço físico a ser ocupado (salas com mecanismos de controle de refrigeração), simplifica o hardware a ser utilizado e com isso, espera-se adquirir um desempenho igual ou superior aos encontrados em arquiteturas convencionais. O problema encontrado no desenvolvimento de aplicações distribuídas em agregados de computadores tradicionais deverá ser encontrado também neste novo paradigma, pois estas aplicações são implementadas com um forte acoplamento ao hardware [Rose 2003].

Um exemplo de uma aplicação que necessita um grande poder de processamento foi desenvolvida no projeto ACR, para uma arquitetura específica. Este programa procura a melhor alternativa entre espectros de produtos farmacológicos utilizando algoritmos genéticos [Aguilar 2005].

Esta aplicação possui várias versões para diferentes arquiteturas de computadores, entre elas, está sendo desenvolvida uma versão sequencial para placas reconfiguráveis. A implementação deste programa está sendo feita na linguagem Java e a sintetização para o microcontrolador (FentoJava) é realizado pela ferramenta Sashimi [Carro 2001] [Freitas 2004], que transpõem dinamicamente as instruções de uma linguagem de alto nível para as instruções interpretadas pelo microprocessador.

Para a utilização da aplicação distribuída em um conjunto de placas FPGAs deverá existir um mecanismo de troca de mensagens. Diversas são as arquiteturas para a comunicação de dados entre dispositivos, entretanto o padrão utilizado em redes de computadores é o modelo TCP/IP, porque são protocolos simples, funcionais e interoperáveis. Na Figura 1, pode-se visualizar um agregado de dispositivos reconfiguráveis contendo um nó servidor e “n” nós clientes.

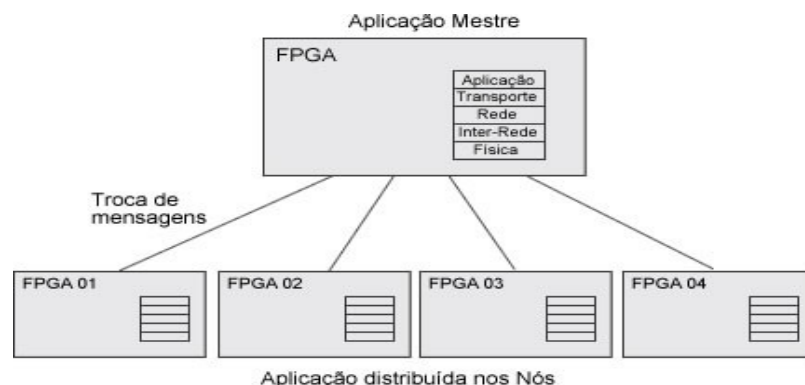


Figura 1. Estrutura de comunicação entre as FPGAs.

Pode-se verificar na figura apresentada, um nó mestre interligado com diversos outros nós. A proposta é executar um programa em modo paralelo sendo iniciado no mestre auxiliado pelos seus nós. Este auxílio no processamento deverá ser realizado de acordo com a aplicação a paralelizada, normalmente são rotinas bem definidas que recebem e retornam parâmetros para o nó mestre, ou mesmo, um outro nó. Para esta troca de mensagens tornar-se possível neste modelo, toda a pilha TCP/IP, exceto a aplicação, deve ser implementada diretamente no hardware da placa.

Não encontrando-se uma versão completa do TCP/IP para ser utilizada, atualmente, estão sendo desenvolvidos novas funções a pilha encontrada, como o envio de pacotes do protocolo UDP (*User Datagram Protocol*) [Freitas 2004]. O objetivo deste projeto é implementar o restante dos protocolos de transporte da pilha TCP/IP, possibilitando o desenvolvimento de aplicações distribuídas em FPGAs, o que atualmente só é possível de forma sequencial.

3. Protocolos de Transporte

A função dos protocolos de transporte é prover a comunicação entre aplicações instanciadas em computadores distintos mediante a interligação de uma rede de dados. Dois protocolos são definidos no modelo TCP/IP, UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*) [Comer 2001]

O UDP é um protocolo que possui sua implementação simplificada, na qual, pode-se adquirir alta performance. As suas principais desvantagens são a não garantia de entrega, ordenação e controle de duplicação de pacotes. Este protocolo é normalmente utilizado em meios de transmissão físicos que garantam a entrega de seus quadros.

O TCP é um protocolo de implementação mais complexa, é estabelecida uma conexão entre as máquinas origem e destino (conexão fim-a-fim), incluindo métodos que garantam a confiabilidade na entrega de pacotes, de forma ordenada, sem duplicação e com controle de fluxo.

Este protocolo originalmente não determina como a camada TCP deve ser implementada em cada equipamento (Darpa, 1981). Deste modo, é possível implementá-lo de maneiras distintas desde que o protocolo seja respeitado. As suas principais implementações são TCP Tahoe, TCP Reno e TCP Vegas.

De acordo com [Fall 1996], o TCP Tahoe opera com as técnicas *Slow Start*, *Congestion Avoidance* e *Fast Retransmit*. O principal problema desse algoritmo ocorre quando muitos segmentos são perdidos fazendo acionar a técnica *Slow Start* diversas vezes e provocando, dessa forma, uma baixa eficiência na rede.

O TCP Reno é um dos mais utilizados na Internet, atualmente, difere do TCP Tahoe, pois o *ACK* (número de confirmação) é realizado para cada segmento enviado e, além disso, a técnica *Fast Recovery* é incluída fazendo com que a janela de congestionamento seja aumentada pelo número de *ACKs* duplicados que o TCP transmissor recebe antes do recebimento de um novo *ACK*. A baixa eficiência do TCP Reno ocorre quando múltiplos segmentos são perdidos de uma *CWND* (janela de congestionamento), porque este algoritmo não retorna para a fase *Slow Start* e sim aciona a fase *Congestion Avoidance*, reduzindo a *CWND* para a metade do seu valor diversas vezes. Entretanto, se o timeout expirar, o TCP Reno começa a transmissão pela fase *Slow Start* [Fall 1996].

O TCP Vegas [Brakmo 1994] é uma implementação introduzida no TCP por meio de algumas modificações realizadas no mecanismo de controle de congestionamento no lado do transmissor. Nele o transmissor antecipa o princípio de congestionamento monitorando a diferença entre a taxa esperada e a taxa real. O Vegas usa a estratégia de ajuste da taxa de envio de dados da origem (janela de congestionamento) na tentativa de manter um pequeno número de pacotes armazenados nos switches durante uma transmissão. Ele é considerado eficiente referente a perda e atraso médio dos pacotes, porém deve ser atribuído tanto para o transmissor quanto o receptor para estas vantagens serem percebidas.

Os protocolos UDP e TCP são utilizados em diversas aplicações e deverão ser implementados em placas FPGAs, deste modo, o programador poderá optar por um protocolo otimizado, visando performance na entrega de pacotes sem conexão, ou um protocolo confiável (com conexão), entretanto com maior *overhead*. A versão do protocolo TCP a ser utilizada no projeto é a TCP Reno por ser utilizado em larga escala mundialmente e possuir um bom desempenho, porém não é estinta a idéia de implementar-se futuramente outro modelo TCP visando maior desempenho entre as FPGAs.

4. Protocolos de Transporte na placa FPGA

Para a programação de placas FPGAs devem ser utilizadas linguagens de descrição de hardware como o VHDL (*Very High Speed Integrated Circuit Hardware Description Language*), na qual todo o hardware é projetado de acordo com a necessidade da aplicação, permitindo a descrição do hardware em entidades interoperáveis.

Existem implementações da pilha TCP/IP desenvolvidas em VHDL, para sua utilização em hardware. Normalmente essas implementações possuem protocolos proprietários, não *open source* e/ou seu desenvolvimento é parcial, como por exemplo o projeto XSV Board [Sutton 2001], que possui implementados os protocolos: ARP (*Address Resolution Protocol*), IP e ICMP (*Internet Control Message Protocol*).

Para a utilização desta pilha TCP/IP com uma aplicação, por exemplo, a procura a melhor alternativa entre espectros de produtos farmacológicos [Aguiar 2005], devem existir componentes que interligam as interfaces do microcontrolador fentoJava com a pilha TCP/IP. Estes componentes de interligação foram desenvolvidos pelo [Freitas 2004] no projeto ACR.

Para o seu completo funcionamento deverão ser descritos em hardware os protocolos de transporte citados na seção 3.

Como descrito anteriormente, o mecanismo de recebimento de mensagens UDP já está completo e descrito na entidade “udp.vhd”. Esta entidade será expandida convertendo-se o código já existente formando o mecanismo de envio de pacotes UDP.

O protocolo TCP necessitará uma boa abstração de seu funcionamento e também uma adaptação para ser possível sua execução em FPGAs. Para a sua implementação, poderão ser utilizadas premissas encontradas em “icmp.vhd”, originalmente desenvolvido para enviar “echo” e responder “replay”, porém, possui um mecanismo semelhante ao necessário pelo protocolo TCP, no instante do envio e recebimento dos pacotes.

Todas as características como: conexão fim-a-fim, ordenamento, controle de fluxo e duplicação de pacotes deverão ser estudadas amplamente possibilitando uma implementação fiel deste protocolo em um modelo FPGA. Neste momento, o interesse é implementar o protocolo TCP com o máximo de suas características possíveis, porém em trabalhos futuros poderão surgir versões mais otimizadas visando um desempenho ainda maior na execução de aplicativos distribuídos.

5. Conclusão

Com a realização deste projeto espera-se viabilizar a integração entre FPGAs possibilitando a utilização deste agregado para a execução distribuída de aplicações específicas. Acredita-se que, mesmo com *clocks* baixos (100MHz) comparados aos computadores convencionais (2GHz e 3GHz), pode-se executar aplicações distribuídas alcançando desempenhos satisfatórios, pois são reduzidas as complexidades dos hardwares e softwares.

No projeto ACR foi implementado uma aplicação que necessita poder computacional, para servir de caso de uso, também foi desenvolvido componentes que interligam as interfaces do microcontrolador fentoJava com uma pilha TCP/IP incompleta.

Para a implementação distribuída de uma aplicação em um agregado de placas FPGAs deve-se implementar os protocolos de transporte, possibilitando a troca de mensagens entre essas placas. Esta implementação está sendo realizada atualmente e após essa etapa ser concluída deverá ser realizada uma grande quantidade de testes para comparar o desempenho entre os diferentes tipos de arquiteturas.

References

- Aguiar, Alexandra de (2005) “Aplicação de Algoritmos Genéticos em Cluster Reconfigurável”. Universidade de Santa Cruz do Sul.
- Brakmo, L., O'Malley, S.W., Peteron L.L., (1994) “TCP Vegas: New Techniques for Congestion Detection and Avoidance”. In Proceedings of ACM SIGCOMM'94.
- Carro, Luigi (2001) “Sashimi: Manual do usuário” Universidade Federal do Rio Grande do Sul.
- Comer, Douglas (2000) “Internetworking with TCP/IP: principles, protocols, and architectures. 4ª ed. New Jersey: Prentice Hall.
- Darpa (1981) “Transmission Control protocol”, DARPA, RFC 793.
- Dehon, A. and Wawrzynek, J. (1999) “reconfigurable Computing: What, Why, and Design Automation Requirements?” in proceedings of the 1999 Design Automation Conference, pp. 610-615.
- Fall, K., Floyd, S., (1996) “Simulation-based Comparisons of Tahoe, Reno and SACK TCP”, Computer Communication, vol 26.
- Freitas, Josué de (2004) “Integração em FPGA de um microcontrolador FentoJava com uma Pilha TCP/IP”. Universidade de Santa Cruz do Sul – Trabalho de Conclusão de Curso.

Rose, Cesar A. F. de and Navaux, Philippe O. A. (2003) “Arquiteturas Paralelas”, Sagra Luzzatto, Brasil.

Sutton, Peter (2001) “VHDL XSV Board Interface Projects”, <http://www.itee.uq.edu.au/~peters/xsvboard>, March.