Avaliação de abordagens de comunicação com FPGA no supercomputador Cray XD1

Vitor C. F. Gomes[1], Andrea S. Charão[1], Haroldo F. C. Velho[2]

[1]Universidade Federal de Santa Maria (UFSM)
[2]Instituto Nacional de Pesquisas Espaciais (INPE)
{vconrado,andrea}@inf.ufsm.br, haroldo@lac.inpe.br

Resumo. *Este artigo apresenta uma avaliação de duas abordagens de comunicação com FPGA no supercomputador Cray XD1. Essa avaliação é importante para aplicações que exigem transferências frequentes de dados a serem processados pelo FPGA. Os resultados mostram que uma abordagem é bastante superior à outra em determinados casos.*

1. Introdução

A computação reconfigurável vem sendo utilizada para o aumento de desempenho de aplicações através do uso de *Field Gate Programable Arrays* (FPGAs) [Gokhale and Graham 2005]. Tais dispositivos podem ser configurados para realizar tarefas específicas, permitindo o aumento de desempenho pela ampliação da vazão de dados ou pela paralelização de computações [Chamberlain et al. 2008, Gokhale and Graham 2005]. Atualmente, existem sistemas que incorporam dispositivos reconfiguráveis em sua arquitetura, possibilitando sua utilização conjunta com processadores de propósito geral [Cray Inc. 2005a]. Esta abordagem, conhecida por computação híbrida, visa a utilização de distintos recursos computacionais para o aumento do desempenho de aplicações [Chamberlain et al. 2008].

Uma técnica utilizada em computação híbrida consiste em executar em FPGAs as operações computacionalmente intensivas de um programa, mantendo as demais operações executando na CPU [Gokhale and Graham 2005, Wain et al. 2006, Chamberlain et al. 2008]. Neste caso, podem ocorrer múltiplas transferências de dados entre os dispositivos para efetuar-se a computação, sendo que o tempo gasto na comunicação pode tornar-se um fator limitante do desempenho. Neste contexto, este trabalho apresenta uma avaliação das abordagens de comunicação com FPGA no supercomputador Cray XD1, baseada em medições de taxas de transferências de dados em diferentes situações. No restante deste artigo, expõe-se inicialmente o ambiente híbrido utilizado e suas formas de comunicação com o FPGA. Em seguida, apresenta-se a metodologia de avaliação e os resultados obtidos.

2. Comunicação no Cray XD1

O Cray XD1 é um sistema híbrido composto por 12 *blades*. Cada *blade* contém dois processadores Opteron de 64 bits, até 8 GB de memória DDR SDRAM e um processador de comunicação RapidArray. Este processador permite a comunicação com alta largura de banda e baixa latência entre os processadores. Cada *blade* pode conter ainda um módulo de expansão que conta com um FPGA, o qual é conectado ao *blade* através de outro processador RapidArray [Cray Inc. 2005a]. A Cray disponibiliza a API RapidArray Transport Core para a comunicação dos processadores com o FPGA. Esta API é

composta por dois blocos denominados Fabric Request e User Request [Cray Inc. 2005b]. Cada bloco realiza a comunicação utilizando abordagens distintas que serão detalhadas nas subseções seguintes.

2.1. Bloco Fabric Request

Nesta abordagem de comunicação, conhecida como *push*, o programa executado nos processadores é responsável pela comunicação com o FPGA. Para isso, a Cray disponibiliza a biblioteca *enlib*, a qual abstrai ao programa o FPGA como um arquivo. A transferência de dados entre os processadores e o FPGA é realizada através de leituras e gravações pelo programa em C neste arquivo. Com a realização de uma leitura ou escrita, o FPGA recebe, através do RapidArray Transport Core, uma requisição que deve ser tratada pela aplicação do FPGA. Em caso de leitura, deverá ser retornado um valor ao RapidArray Transport Core para que ocorra o retorno da função chamada pelo programa em linguagem de alto nível. Somente é permitida a manipulação de um *quadword* (64 bits) por requisição utilizando o bloco Fabric Request [Cray Inc. 2005b].

2.2. Bloco User Request

A abordagem que utiliza o bloco User Request é conhecida como *pull* e, diferentemente da Fabric Request, mantém os processadores livres durante a transferência de dados entre o programa em C e o FPGA. Para isso, este bloco permite que o FPGA realize leituras e escritas em um espaço de memória compartilhado do programa. O endereço para a região de memória, que é compartilhada utilizando a *einlib*, é enviado ao FPGA através do bloco Fabric Request. Estando disponível o endereço, a aplicação descrita para o FPGA é capaz de fazer até 32 requisições seqüenciais ao RapidArray Transport Core. Cada requisição pode solicitar até 8 posições contíguas da memória do programa. Os retornos das solicitações ao RapidArray Transporte Core não são necessariamente na ordem em que foram realizadas. O *core* garante somente a ordem das 8 posições contíguas de cada requisição [Cray Inc. 2005b]. O sistema descrito para o FPGA deve ordenar os dados através do auxílio de *tags* disponibilizadas durante a requisição e o retorno. Outra solução é aguardar o retorno de uma solicitação antes de realizar outra.

3. Avaliação

Para medir a taxa de transferência de dados entre os processadores e o FPGA, implementou-se duas descrições em VHDL e um programa em linguagem C. Cada descrição em VHDL utiliza uma das duas abordagens de comunicação.

Para a abordagem *push*, a descrição para FPGA faz uso do bloco Fabric Request e realiza a manipulação das requisições de leitura e escrita através de uma máquina de estados. Para a abordagem *pull*, a aplicação descrita utiliza os dois blocos disponíveis no RapidArray Transport Core. Isto é necessário pois o envio do endereço da região de memória compartilhada é feito através do bloco Fabric Request, e a transferência de dados utiliza o bloco User Request. Esta implementação utiliza uma máquina de estados para tratar o recebimento do endereço de memória e outra que faz as requisições de dados e manipula as respostas do *RapidArray Transport Core*. Esta máquina aguarda o retorno de uma requisição antes de realizar outra. Este método foi utilizado visando simplificar a descrição do sistema e evitar que uma maior área do FPGA seja utilizada para o tratamento de comunicação.

A função principal do programa em linguagem de alto nível é realizar a carga dos arquivos de configuração no FPGA e calcular o tempo de execução dos testes de transferência de cada abordagem. Para cada forma de comunicação implementou-se uma função que realiza a transferência dos dados. Na abordagem *push* a função envia os dados para o FPGA através de sucessivas gravações. Para a abordagem *pull* o programa compartilha a região de memória, envia o endereço para o FPGA, envia o total de requisições que o FPGA deverá realizar e aguarda o fim da transferência dos dados. O tempo de compartilhamento de memória foi considerado na obtenção dos tempos.

A avaliação da comunicação entre os processadores do Cray XD1 e os FPGAs é baseada na tomada de tempos de execução das transferências de dados utilizando as duas abordagens. Nesta avaliação não são considerados os tempos de carga dos binários de descrição de hardware no FPGA. Para avaliar diferentes situações de transferência de dados aos FPGAs, realizou-se experimentos que variam a quantidade de dados de 8B até 2MB. Em outro estudo que avalia a comunicação com FPGAs no XD1 [Fernando et al. 2005], trabalha-se com quantidades de dados maiores, que variam de 100MB a 500MB. Tal estudo é direcionado a uma aplicação que executa inteiramente em FPGA. No caso de aplicações híbridas, no entanto, a quantidade de dados transferidos pode ser menor, o que justifica a delimitação dos experimentos no presente trabalho.

O tempo de cada experimento foi obtido pela média de 10 execuções. A quantidade de dados transferidos, os tempos médios das execuções e os desvios padrão (σ) são apresentados na tabela 1. Na figura 1 podem ser vistas as curvas formadas entre o tempo e o total de dados transferidos das duas abordagens implementadas. A abordagem *push* foi mais rápida para transferências com até 1KB. Nos demais experimentos a abordagem *pull* foi mais rápida em relação à *push*. Pode-se observar um crescente distanciamento das curvas, fato este que pode estar ligado ao tempo constante no envio do endereço para o FPGA e ao tempo de compartilhamento de memória, que em média foi de 939,1 μs. Este tempo fixo torna-se cada vez menos significativo com o aumento dos dados transferidos.

A maior taxa de transferência foi encontrada no experimento que utiliza a abordagem *pull* com 2MB e é de 87,5MB/s . Essa taxa está muito abaixo da máxima teórica especificada pela Cray - 1,6GB/s. Em experimentos [Fernando et al. 2005] de gravação e leitura realizados em um Cray XD1 utilizando o bloco User Request foi possível transferir 500MB em 1s. Esta maior taxa pode ser justificada pela utilização de 32 requisições seqüenciais de 8 posições contíguas de memória e a realização de leituras e escritas simultaneamente.

4. Conclusão

Com a comparação da transferência de dados usando as abordagens *push* e *pull*, verificou-se que o segundo método possui melhor desempenho, em termos de tempo, em relação ao primeiro, para transferências superiores a 16KB. A transferência de dados no sentido processador-FPGA chega a ser 7 vezes mais rápida usando a interface User Request em relação à Fabric Request. Entretanto, para a transferência de poucos dados, é preferível o uso da abordagem *push*. Mesmo assim, a taxa de transferência está muito aquém da máxima teórica, sugerindo que o estudo deva ser continuado, buscando-se formas mais eficientes de implementar a comunicação usando os recursos disponíveis.

Tabela 1. Tempos de comunicação

	Tempo (μs)		σ		Taxa (MB/s)	
Dados	Push	Pull	Push	Pull	Push	Pull
8B	$0,9$	$1104,8$	$0,32$	$8,28$	$8,5$	$< 0,1$
16B	$1,6$	$1119,7$	$0,97$	$6,78$	$9,5$	$< 0,1$
128B	$11,7$	$1122,1$	$1,89$	$8,02$	$10,4$	$0,1$
1KB	$94,3$	$1130,2$	$12,96$	$5,47$	$10,4$	$0,9$
16KB	$1440,3$	$1284,6$	$79,53$	$9,19$	$10,9$	$12,2$
128KB	$10986,1$	$2421,0$	$959,73$	$17,62$	$11,4$	$51,6$
1MB	$83017,2$	$11726,5$	$2896,20$	$70,75$	$12,0$	$85,3$
2MB	$166725,0$	$22844,4$	$4940,44$	$71,91$	$12,0$	$87,6$

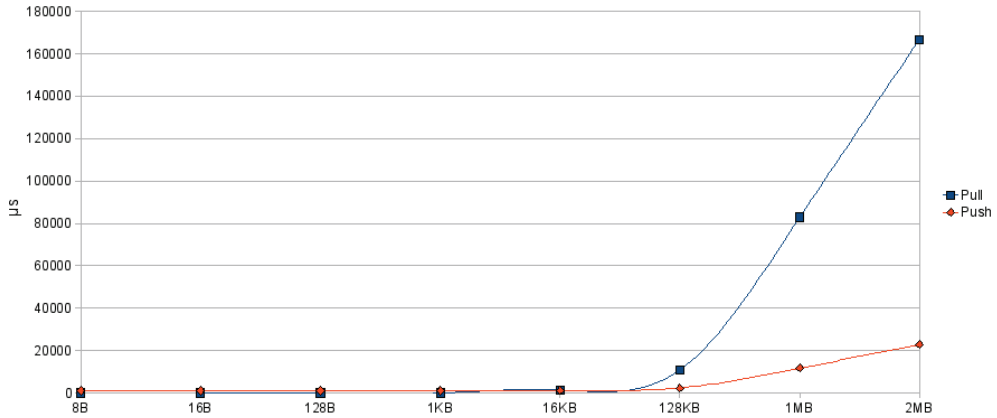

Figura 1. Tempos de comunicação

Referências

Chamberlain, R. D., Lancaster, J. M., and Cytron, R. K. (2008). Visions for application development on hybrid computing systems. *Parallel Comput.*, 34(4-5):201–216.

Cray Inc. (2005a). *Cray XD1 System Overview*. Mendota Heights, MN, USA.

Cray Inc. (2005b). *Design of Cray XD1 RapidArray Transport Core*. Mendota Heights, MN, USA.

Fernando, J., Dalessandro, D., Devulapalli, A., and Wohlever, K. (2005). Accelerated FPGA based encryption. New Mexico, USA. The Cray User Group.

Gokhale, M. and Graham, P. S. (2005). Reconfigurable computing: Accelerating computation with field-programmable gate arrays. In *Reconfigurable Computing*, New Mexico, USA. Springer.

Wain, R., Bush, I., Guest, M., Deegan, M., Kozin, I., and Kitchen, C. (2006). An overview of FPGAs and FPGA programming; initial experiences at Daresbury. pages 2–4, Daresbury, Cheshire, UK. Council for the Central Laboratory of the Research Councils.