Modelagem e Avaliação de Desempenho de Agregados Conectados por Tecnologia SCI

Rafael B. Ávila[1]*, César A. F. De Rose[2]†, Philippe. O. A. Navaux[1]‡, Roberto A. Hexsel[3]§ e Hans-Ulrich Heiß[4]¶

[1] Instituto de Informática
Universidade Federal do Rio Grande do Sul
Av. Bento Gonçalves, 9500 – Bloco IV
Cx. Postal 15064 CEP 91501-970 Porto Alegre
Tel.: (051) 316-6165 Fax: (051) 319-1576
[2] Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre
[3] Departamento de Informática
Universidade Federal do Paraná
Curitiba
[4] Centro de Computação Paralela de Paderborn
Universität GH Paderborn
Paderborn, Alemanha

Resumo—

SCI é uma das mais promissoras tecnologias de comunicação da atualidade na construção de agregados de estações para processamento de alto desempenho, principalmente pelo oferecimento do modelo de memória compartilhada. Sua utilização na prática, entretanto, é ainda cercada de questões técnicas como topologias de construção de agregados, expectativa de desempenho e padrões de referências a dados compartilhados. Este artigo apresenta um projeto que visa a resolução de tais questões por meio de modelos de simulação de hardware e de programas sobre agregados SCI. O texto expõe os objetivos e principais atividades previstas para a pesquisa. Ao final são apresentados os resultados obtidos na primeira fase do projeto, com a avaliação de desempenho de um agregado baseado em máquinas PC multiprocessadas utilizando o sistema Linux.

Palavras-chave— **SCI, processamento de alto desempenho, agregados de estações, memória compartilhada.**

I. Introdução

O uso de agregados (*clusters*) de estações é atualmente a forma mais freqüentemente empregada na prática da programação paralela [7], principalmente pelo baixo custo de implementação e pela disponibilidade de software gratuito como PVM [9] e MPI [8] na Internet [1]. Uma das mais promissoras tecnologias de interconexão de agregados na atualidade é SCI—*Scalable Coherent Interface*. SCI tem chamado de forma significativa a atenção de diversos grupos de pesquisa na academia e na indústria, porque apresenta desempenho comparável ao de outras tecnologias atuais como Myrinet [2] e mesmo de arquiteturas MPP (com vazão da ordem de centenas de MBytes/s e latência de poucos microsegundos) e ainda assim é de relativo baixo custo de implementação. Porém o grande atrativo de SCI consiste na possibilidade de programação com memória compartilhada ao invés de troca de mensagens. Um agregado conectado por SCI torna-se um sistema com *cache* coerente global entre todos os nodos, permitindo o uso do modelo de memória compartilhada. Diferentemente de modelos de emulação de memória compartilhada distribuída [16], o controle de *caches* e acesso ao barramento interno de um nodo é feito diretamente pelo hardware de comunicação, o que proporciona desempenho aprimorado e menor complexidade no nível de software.

Por ser uma tecnologia recente, entretanto, algumas questões relativas à programação paralela com SCI encontram-se ainda em aberto. Em especial, não existe uma maneira óbvia, em termos de topologia, de se construírem agregados com muitos nodos (da ordem de centenas), nem se tem uma idéia precisa do tipo de desempenho, ou quais fatores podem afetar o desempenho de uma determinada aplicação paralela sobre tal arquitetura. Uma alternativa para avaliação *a priori* de tais questões é o desenvolvimento de modelos de simulação para o hardware e para aplicações a serem executadas sobre agregados SCI, de modo que questões como as colocadas anteriormente possam ser resolvidas, ao

*Pesquisador RHAE/CNPq do PPGC/UFRGS; Mestre em Ciência da Computação (UFRGS, 1999)

†Professor da Faculdade de Informática/PUC-RS; Doutor em Informática (Universidade Fridericiana de Karlsruhe, 1998)

‡Professor do PPGC/UFRGS; Dr. Eng. em Informática (INPG, 1979)

§Professor do Depto. de Informática/UFPR; Ph.D. em Informática (Universidade de Edinburgo, 1994)

¶Professor da Faculdade de Matemática e Informática/Universidade de Paderborn; Doutor em Informática (Universidade Fridericiana de Karlsruhe, 1993)

menos parcialmente, antes da implementação propriamente dita.

O interesse na utilização de SCI no processamento paralelo motivou o estabelecimento de um projeto que visa o desenvolvimento de modelos de desempenho de agregados de estações baseados nessa tecnologia, os quais possam ser usados para responder as questões abordadas anteriormente. Além do desenvolvimento dos modelos, o projeto, dentro de um programa de cooperação entre a UFRGS, PUC-RS e UFPR, no Brasil, e a Universität Paderborn, na Alemanha, prevê sua validação com a implementação na prática de aplicações paralelas sobre uma máquina paralela baseada em SCI. Este artigo apresenta os resultados obtidos com o hardware construído na primeira fase do projeto, enfocando aspectos de desempenho da máquina, os quais serão utilizados como base para a implementação dos modelos.

O restante do documento está organizado como segue. A seção II descreve em mais detalhes a tecnologia SCI e sua arquitetura operacional. Na seqüência a seção III faz uma breve apresentação do projeto de pesquisa e das atividades sendo desenvolvidas. A seção IV descreve a máquina paralela construída, e a seção V apresenta os resultados obtidos com a avaliação de desempenho. Por fim a seção VI apresenta as conclusões dos autores.

II. A TECNOLOGIA SCI

SCI é uma tecnologia de comunicação de alto desempenho especificada na norma IEEE 1596-1992 [12]. Sua principal diferença em relação a outras tecnologias como Fast Ethernet e Myrinet consiste no modelo de programação empregado. Ao invés de troca explícita de mensagens entre nodos de processamento, SCI permite que um processo tenha mapeado em seu espaço de endereçamento um segmento de memória fisicamente localizado em outro nodo, de modo que as aplicações executadas sobre tais máquinas podem seguir o modelo de memória compartilhada.

A arquitetura básica de conexão com SCI é em anel. Cada interface (de hardware) possui dois canais de comunicação, um de entrada e um de saída, com performance nominal de 500 MBytes/s de vazão e 3μs de latência. Atualmente as interfaces SCI são fabricadas pelas firmas Dolphin [5], na Noruega, e Scali [15], nos EUA, sendo que esta última produz uma versão modificada das interfaces com conexões para dois anéis, o que permite facilmente a implementação de agregados com topologia toróide. Além do desempenho e baixo custo, uma das características marcantes de SCI é sua alta capacidade de escalabilidade (da ordem de centenas de nodos).

A programação com SCI baseia-se na criação e mapeamento remoto de segmentos de memória a serem compartilhados. O princípio de funcionamento dessa característica é implementado em hardware, pela utilização da janela de endereçamento do barramento PCI, com largura de 64 bits, e pela manipulação das tabelas de páginas relativas ao espaço de endereçamento de um processo. Uma falha de acesso a um endereço de memória indica uma referência a um segmento remoto. A falha é tratada pelo gerenciador de dispositivo (*device driver*) da interface SCI, que então executa a operação remotamente, através do meio de comunicação.

Note-se que a implementação de gerenciadores de dispositivo para interfaces SCI deve ser fortemente integrada ao sistema operacional, normalmente exigindo alterações ou extensões nos módulos de gerência de memória. Além do mapeamento de segmentos locais, os gerenciadores devem manipular o mapeamento de segmentos remotos em dois níveis, (*i*) do espaço de endereçamento do processo para a interface, e (*ii*) da interface para o segmento remoto, através do meio de comunicação. Os três tipos de mapeamento são representados na figura 1.

A padronização SCI inclui a implementação de todos os protocolos de comunicação e controle de *cache* nas próprias interfaces de interconexão. O protocolo de comunicação é baseado na transferência de pacotes de no máximo 256 bytes de dados[1], com confirmações (*echoes*) de 8 bytes. Já o controle de coerência de *cache* não pode ser inteiramente implementado em interfaces PCI. Mais especificamente, o *caching* de segmentos remotos não pode ser implementado porque transações entre o processador e a memória física de um nodo não são visíveis ao barramento PCI, e em conseqüência à interface SCI. Esta limitação acarreta em diferenças de performance que devem ser levadas em conta quando do projeto de aplicações para agregados SCI [4].

SCI é uma tecnologia relativamente recente, por isso muitos grupos de pesquisa ainda estão estabelecendo sua infraestrutura e conduzindo experimentos em análise de desempenho. As atividades são mais concentradas na pesquisa em diferentes topologias e interfaces de programação de baixo nível. Estas últimas, especialmente bibliotecas de comunicação, estão atraindo a atenção de diversos grupos de pesquisa, principalmente na Alemanha e nos Estados Unidos. O Laboratório HCSR [10], da Universidade da Flórida, está conduzindo experimentos e comparações com software de baixo nível portado para SCI, como Active Messages [17], e também com uma implementação proprietária de MPI. O HCSR também realiza experimentos em mais alto nível portando HPF para SCI. A Alemanha reúne uma comunidade de universidades muito ativas em diversas áreas relacionadas à programação com SCI, desenvolvendo pesquisas tanto em hardware [11] como em software de baixo nível [6, 11].

[1] A interface produzida pela firma Dolphin, atualmente a mais utilizada na prática, limita o tamanho dos pacotes a 64 bytes de dados [3].

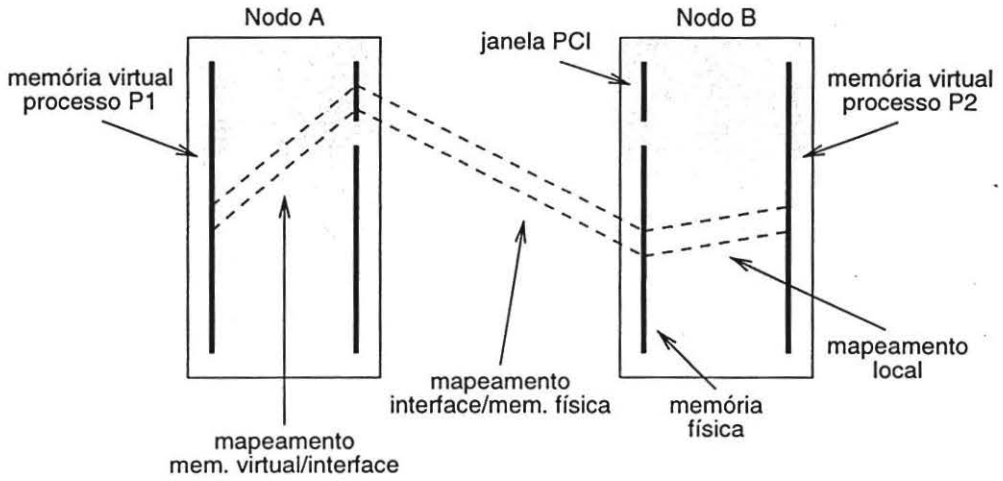

Fig. 1: Tipos de mapeamento de segmentos compartilhados.

III. Atividades do projeto

Mesmo oferecendo características muito atraentes, a utilização de agregados baseados em SCI ainda apresenta diversas questões em aberto:

- qual a topologia mais adequada para a construção de agregados com centenas de nodos?
- qual o comportamento típico de referência a dados compartilhados?
- se o padrão de referência a dados de um determinado programa é conhecido, como os dados podem ser distribuídos no agregado?
- existe uma maneira automática de avaliar o comportamento de referências de aplicações paralelas, para então, de forma também automática, mapear os dados em processadores?

O estabelecimento deste projeto visa unir recursos técnicos e humanos das universidades participantes com um objetivo geral de desenvolver modelos de desempenho e ferramentas de modelagem de aplicações paralelas para agregados SCI, os quais possam ser usados para prover respostas às perguntas colocadas. A expectativa é que se possa ter uma idéia do desempenho de aplicações paralelas, através de prototipação rápida, antes que seja dispendido tempo em sua implementação completa sobre o agregado. Devido à complexidade de tais modelos, porém, o compromisso entre complexidade e precisão deve ser cuidadosamente analisado.

A primeira etapa do projeto consistiu na definição e implementação da arquitetura de hardware e software a ser modelada e na obtenção de parâmetros de desempenho, como latência e vazão, por meio de aplicações típicas (*benchmarks*). Estas tarefas foram executadas em sua maior parte em Paderborn, onde continuam a ser realizados experimentos de avaliação de desempenho e ampliação da arquitetura. A etapa atual concentra os principais objetivos da pesquisa pois define as ferramentas de simulação e modelagem que servirão de base para a execução das atividades finais. O simulador do hardware é desenvolvido pelo grupo da UFPR, com base nos parâmetros de desempenho obtidos anteriormente. A PUC-RS e a UFRGS desenvolvem as ferramentas de modelagem que produzirão a carga para o simulador.

Os resultados esperados para o projeto incluem a documentação e validação dos modelos de avaliação de desempenho, um análise profunda do comportamento de tais sistemas e a identificação de estratégias quase-ótimas para mapeamento de código e dados de aplicações paralelas sobre agregados SCI, bem como o reforço da interação entre os grupos envolvidos, formação de pesquisadores e publicações através de artigos e relatórios técnicos.

IV. As máquinas construídas

A tarefa inicial do projeto foi a modelagem e a construção de máquinas paralelas basedas em agregados interconectadas por tecnologia SCI. O objetivo aqui era explorar ao máximo o potencial desta tecnologia com o objetivo de obter o maior desempenho possível em uma máquina paralela de porte médio (em relação ao número e ao poder computacional dos nós).

Como o padrão SCI ainda é relativamente novo, as placas de interconexão encontradas no mercado que implementam este protocolo ainda não podem ser considerados produtos consolidados, tanto em termos de hardware como principalmente de software. Isto resultou no fato desta tarefa não ter se resumido na escolha dos nós e da topologia de interconexão entre eles, mas ter acabado envolvendo uma pesquisa de mercado para verificar a funcionalidade dos produtos disponíveis, que nem sempre implementam a totalidade dos serviços especificados no padrão. Após a constatação que não havia suporte disponível para o sistema operacional que

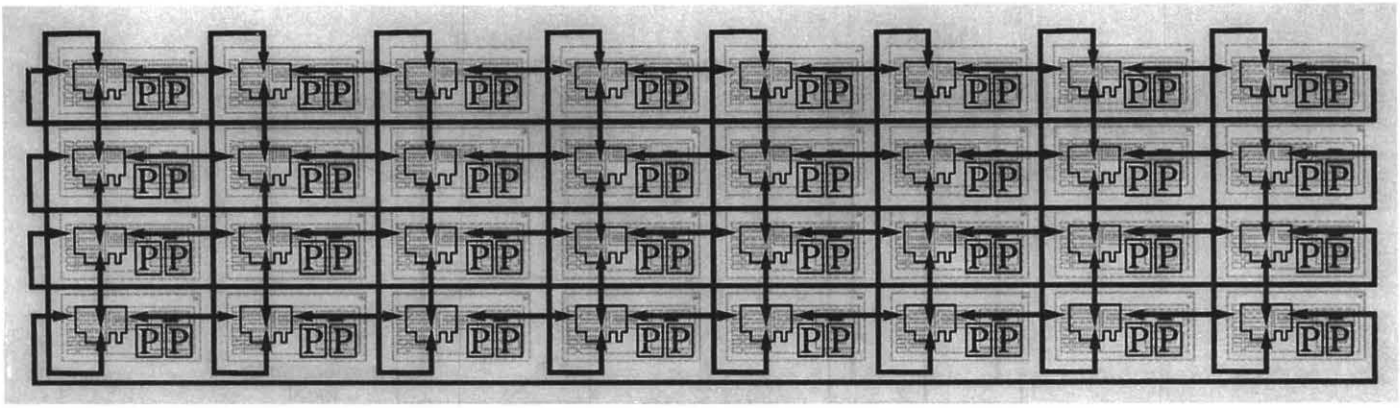

Fig. 2: Padeborn SCI Cluster (PSC-64) com 32 nós de 2 processadores interligados em um torus de dimensão 8×4.

se desejava utilizar foi consolidada até uma cooperação com os fabricantes no desenvolvimento de gerenciadores de dispositivos para estas placas. A placa da firma Scali foi a escolhida para o projeto por ser o produto mais desenvolvido até o momento, e o grupo de Paderborn, no comando do prof. Heiß, participou do desenvolvimento dos gerenciadores de dispositivo para o sistema Linux. Os principais fatores que levaram à escolha do Linux foram a possibilidade de alteração no código do sistema, necessário devido à integração dos serviços de memória compartilhada fornecidos pela placa no gerenciador de memória do sistema, e ao fato do sistema ser gratuito.

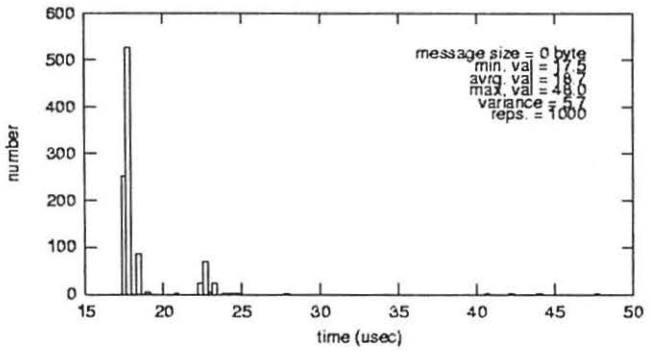

Fig. 3. Histograma da latência na transferência de mensagens.

A figura 2 apresenta a arquitetura da primeira máquina desenvolvida com as placas SCI da firma Scali, que já se encontra em funcionamento no Centro de Computação Paralela de Paderborn (PC^2). A topologia escolhida foi a toróide por ter uma ótima relação entre a escalabilidade e o diâmetro da rede. Para esta primeira máquina foram interconectados 32 nós em um torus de dimensão 8×4. Como cada placa possui conexões para dois anéis, cada nó está ligado com os vizinhos da mesma linha e da mesma coluna por anéis. A placa se encarrega de rotear as mensagens, ou acessos a memórias remotas, de nós em outras linhas ou colunas. Este roteamento é feito em hardware nas placas de forma distribuída, ou seja, sem necessidade de um chaveador (*switch*) adicional. Para os nós foram escolhidas máquinas com dois processadores Pentium II de 300 Mhz e 256 Mbytes de memória principal SDRAM. Experimentos com máquinas de 4 processadores demonstraram problemas no seu *chipset* o que fez com que se optasse por máquinas duais. Os nós rodam o sistema operacional Linux com alterações no gerenciador de memória e os gerenciadores de dispositivo desenvolvidos para a placa SCI da firma Scali. Sobre o Linux está disponível uma versão de MPI adaptada a estas placas, fornecida pelo fabricante.

V. Resultados obtidos

Após a definição da arquitetura e da confecção da máquina paralela conectada por tecnologia SCI iniciaram-se os testes de medição do desempenho da comunicação. Estes valores serão alimentados nos simuladores que, a partir de uma descrição em alto nível de aplicações paralelas, estimarão o desempenho obtido na arquitetura. Esta estimativa será então comparada com medições de desempenho na máquina paralela com o objetivo de ajustar as ferramentas de avaliação de desempenho desenvolvidas.

Para expressar o desempenho da comunicação foram medidos os seguintes indicadores: latência na transmissão de mensagens, vazão na transferência de dados com mensagens, custo da sincronização, vazão de transferências de dados para memórias remotas. Como não se encontravam disponíveis aplicações que estivessem sido totalmente adaptadas para este sistema foram utilizados *benchmarks* sintéticos para a medição destes resultados que rodam sob o ScaMPI, versão da biblioteca MPI adaptada pela firma Scali para as placas SCI. A única exceção foi a medição da vazão de transferência de dados para memórias remotas, que por não ser suportada no modelo de programação do MPI (troca de mensagens), foi implementada diretamente na placa. No final desta seção, após a apresentação das medições do desempenho da comunicação, são apresentados resultados do

desempenho global desta máquina, obtidos com o programa de avaliação de desempenho para sistemas de troca de mensagem MP Linpack [14].

A figura 3 apresenta um histograma de latência na transmissão de mil mensagens para uma máquina remota. Os valores resultam da metade do tempo necessário para que uma mensagem vazia seja enviada a uma máquina remota e recebida de volta. O tempo médio obtido foi de 18,7 microssegundos, para uma variância de 5,7.

O tempo de sincronização foi medido através da utilização de um mecanismo de barreira. A figura 4 apresenta os resultados obtidos para a variação no número de processos sincronizados. Estes resultados já foram obtidos com uma segunda versão da máquina paralela que foi expandida para 96 nós. Não foram efetuadas alterações na arquitetura da máquina (figura 2), sendo a topologia toróide de dimensões 8×12.

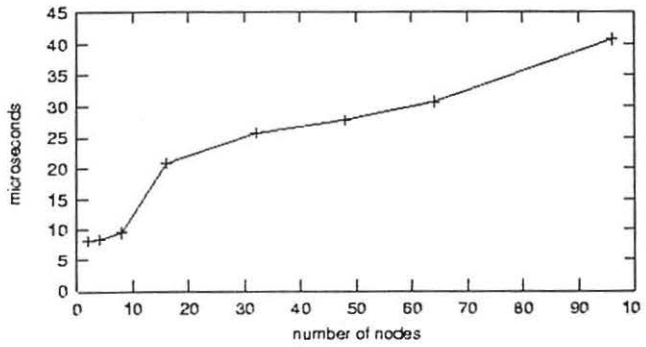

Fig. 4. Tempos de sincronização pelo mecanismo de barreira.

A figura 5 apresenta os resultados obtidos com a medição da vazão no envio de mensagens entre máquinas. São apresentados os valores máximos e médios para transferências bidirecionais (1) e unidirecionais (2). O tamanho das mensagens variou de 10 bytes a 1 Mbyte. Em média foram obtidas vazões de 65 MB/s no caso bidirecional e de 45 MB/s no caso unidirecional.

Outro dado importante para as ferramentas que avaliarão o desempenho de programas rodando nesta máquina é o custo de uma transferência de dados para uma memória remota. Neste caso não é possível a utilização de um programa MPI, pois esta biblioteca é baseada no modelo de troca de mensagens e não suporta diretamente a operação de escrita e leitura em uma memória remota. Para estas medições foram escritos programas em C que se utilizam da rotina `memcpy()` da biblioteca C padrão do Linux. Os resultados obtidos mostraram uma vazão média de 70 MB/s na escrita e 10 MB/s na leitura. A diferença de desempenho entre a escrita e a leitura remota é originada em parte pelo próprio protocolo de baixo nível de SCI. Em uma operação de escrita o dado a ser escrito pode ser incluído no mesmo pacote que faz a requisição

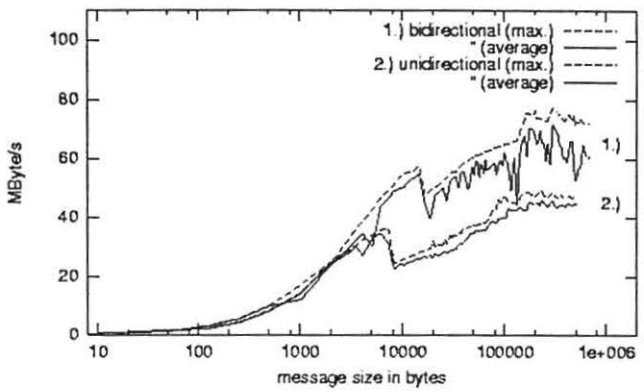

Fig. 5. Vazão da transferência de dados em relação ao tamanho da mensagem.

do serviço à interface remota, bastando que o nodo receptor emita um *echo* para confirmar a requisição. Já uma operação de leitura requer que o nodo receptor envie, após o *echo*, um pacote adicional, o qual contém a informação desejada. Um estudo detalhado das operações que ocorrem no barramento PCI das máquinas durante estas operações está sendo realizado, e talvez revele características que provoquem o acentuamento desta discrepância.

Para dar uma idéia do desempenho da máquina paralela como um todo são apresentados na tabela I os resultados obtidos com o pacote MP Linpack, um conjunto de programas de avaliação de desempenho para sistemas de troca de mensagem. Os resultados mostram o desempenho total, medido em MFlops/s, para a execução do algoritmo com diferentes números de nodos. Além do excelente desempenho (atingindo 7811 MFlops/s com 32 nodos) a tabela comprova a alta capacidade de escalabilidade de SCI, visto que a performance aumenta proporcionalmente ao números de nodos utilizados.

TABELA I
RESULTADOS DA AVALIAÇÃO DO DESEMPENHO DA MÁQUINA PARALELA COM O MP LINPACK

Nós	Rmax (Mflops/s)	Nmax order	N1/2 order	Rpeak (Mflops/s)
32	7811	28000	8000	19200
24	5685	24000	—	14400
16	4009	19500	—	9600
8	2042	14000	—	4800

VI. CONCLUSÕES

Os resultados obtidos nos experimentos realizados sobre o agregado confirmam na prática o potencial de desempe-

nho oferecido pela tecnologia SCI. As medidas de latência e vazão implementadas em MPI apresentam performance comparável, por exemplo, à do pacote MPI-FM [13], uma implementação dedicada de MPI para Myrinet. Enquanto que a latência mínima fica em torno de 19μs para ambas as implementações, a vazão atingida por SCI é significativamente maior (33 MB/s na Myrinet e 45 MB/s em SCI). Também com a implementação do *benchmark* MP Linpack o agregado mostrou sua capacidade para processamento de alto desempenho, atingindo performance de 7811 MFlops/s quando da utilização de 32 nodos.

Além do alto desempenho, a possibilidade de programação com memória compartilhada já representa por si só um enorme atrativo. Estes dados levam a crer que SCI é uma das mais promissoras tecnologias de comunicação em rede da atualidade.

O projeto aqui apresentado busca resolver questões ainda em aberto quanto à utilização de agregados baseados em tecnologia SCI. Em sua primeira etapa a pesquisa foi dedicada à definição da topologia da máquina a ser construída e posterior avaliação do desempenho apresentado pelo agregado. Atualmente os integrantes do projeto trabalham na elaboração de modelos analíticos de desempenho da máquina (simulador) e ferramentas de modelagem de aplicações paralelas.

É importante ressaltar que a tecnologia para interconexão de agregados ainda está em evolução, uma vez que vários fabricantes estão produzindo placas de baixa latência, e novos padrões para estas placas estão sendo desenvolvidos. Cabe à comunidade científica fazer a sua parte pesquisando e desenvolvendo arquiteturas e ambientes de execução para estas placas, traçando assim o caminho a ser seguido por esta nova classe de máquinas paralelas.

REFERÊNCIAS

[1] Louis Baker and Bradley J. Smith. *Parallel Programming*. McGraw-Hill, New York, NY, 1996.

[2] N. Boden et al. Myrinet: A gigabit-per-second local-area network. *IEEE Micro*, 15(1):29–36, February 1995.

[3] Roger Butenuth and Hans-Ulrich Heiß. Leistungsvergleich SCI-gekoppelter PC's. In Wolfgang Rehm, editor, *1. Workshop Cluster-Computing*, Chemnitz, Alemanha, 1997.

[4] Roger Butenuth and Hans-Ulrich Heiß. Shared memory programming on PC-based SCI. In *SCI Europe*, Bordeaux, França, 1998. Promovido como Conference Stream no EMMSEC'98—European Multimedia, Microprocessor Systems and Electronic Commerce Conference and Exposition.

[5] Dolphin interconnect solutions home page. http://www.dolphinics.no/dolphin2/interconnect/adapters/pci-32/pci32adapter.htm.

[6] Marcus Dormanns, W. Sprangers, H. Ertl, and T. Bemmerl. A programming interface for NUMA shared-memory clusters. In *Proceedings of High Performance Computing and Networking '97*, number 1225 in Lecture Notes in Computer Science, Vienna, Austria, 1997. Springer.

[7] Ian East. *Parallel Processing with Communicating Process Architecture*. UCL Press, London, 1995.

[8] MPI Forum. The MPI message passing interface standard. Technical report, University of Tennessee, Knoxville, April 1994.

[9] Al Geist et al. *PVM: Parallel Virtual Machine*. MIT Press, Cambridge, MA, 1994.

[10] HCS Reserach Laboratory home page. http://www.hcs.ufl.edu/.

[11] Hans-Ulrich Heiß. AG Heiß—project Arminius. http://www.uni-paderborn.de/cs/heiss/arminius.

[12] IEEE. IEEE standard for scalable coherent interface (SCI). IEEE 1596-1992, 1992.

[13] Mario Lauria and Andrew Chien. MPI-FM: High performance MPI on workstation clusters. *Journal of Parallel and Distributed Computing*, 40(1):4–18, January 1997.

[14] Linpack library. http://www.netlib.org/linpack.

[15] SCALI—high performance affordable supercomputing. http://www.scali.com.

[16] The TreadMarks distributed shared memory (DSM) system. Available by WWW at http://www.cs.rice.edu/~willy/TreadMarks/overview.html, December 1998.

[17] Thorsten von Eicken, David E. Culler, Seth Copen Goldstein, and Klaus Erik Schauser. Active messages: a mechanism for integrated communiaction and computation. In *Proc. of the 19th Annual International Symposium on Computer Architecture*, pages 256–266, Gold Coast, Australia, 1992.