BlueGene/L

Douglas Gielo Quinellato
Unicamp
RA: 057615
ra057615@students.ic.unicamp.br

ABSTRACT
Este artigo descreve a arquitetura do supercomputador BlueGene/L. São descritos a sua estrutura de máquinas, sua rede de interconexão e o modelo de programação utilizado para o desevolvimento de aplicativos.

Categories and Subject Descriptors
B.0 [**Hardware**]: general

General Terms
Design

Keywords
BlueGene/L, supercomputer, MPI, torus network

1. INTRODUÇÃO
BlueGene/L é um supercomputador desenvolvido por uma parceria entre a IBM e o Lawrence Livermore National Laboratory. Ele é um supercomputador massivamente paralelo, composto de até 65.536 nós de processamento, com pico de performance de 360TeraFLOPS. Seu desenvolvimento foi feito tendo como meta um bom relacionamento cuto/benefício, e visando uma redução no consumo de energia. Para atingir estes objetivos, foi utilizado a tecnologia *System on a chip*, que integra processador, memória, cache, controlador de comunicações num único ASIC (Application Specific Integrated Circuit), permitindo a utilização de uma quantidade enorme de nós com uma capacidade relativamente modesta. É utilizado o modelo de programação por trocas de mensagens, através do padrão MPI.

Seu projeto tem como filosofia a utilização de processadores de baixo custo e de baixo consumo, com uma freqüência de operação próxima à de acesso à memória. Dessa forma, obtém-se uma economia de energia (aprox. 1MW) e de espaço (menos que 232 m²).

O BlueGene/L ocupa as duas primeiras posições na lista top500[5] de Novembro/2005. A primeira posição foi conseguida com o computador instalado no Lawrence Livermore National Laboratory, com 131072 processadores; a segunda posição foi conquistada com o BlueGene/L instalado no IBM Thomas J. Watson Research Center, com 40960 processadores.

2. ARQUITETURA[1]
O BlueGene/L é formado por vários nós, interconectados por 5 redes de interconexão, uma para comunicação geral em formato de toro e as outras com finalidades específicas. Estes nós são compostos de dois processadores IBM PowerPC, com memória de até 2GB e controladora de rede gigabit ethernet. Utilizando a tecnologia *System on a chip*, os nós possuem um *die size* de 11.1 mm². Cada processador é capaz de realizar 4 operações de ponto-flutuante, na forma de duas operações de multiplicação-adição. Geralmente um dos processadores é dedicado à recepção e envio de mensagens entre os nós. Neste caso, o desempenho máximo é de 180 TeraFLOPS.

O *packing* do sistema é mostrado na Figura 1. Cada *computing card* é composto de dois nós. 16 computing card podem ser colocados numa *board*, 16 boards podem ser colocados num *midplane* e até dois midplanes formam um *rack*, num total de até 1024 nós por rack.

Na figura também é mostrado a quantidade de operações em ponto flutuante (FLOPS) e a capacidade máxima de memória nos diferentes componentes.

Cada nó executa um pequeno kernel, responsável por tarefas básicas de comunicação e por prover funções de alto nível para código científico de alto desempenho. Para a compilação, análise de depuração, é necessário uma máquina externa. A comunicação com esta máquina externa é feita através de nós de I/O, que são nós dedicados a execução de funções de suporte, executadas pelo sistema operacional. Deve-se haver nós de I/O na proporção máxima de 1 para cada 8 de processamento, ficando esta normalmente na razão de 1 para 64.

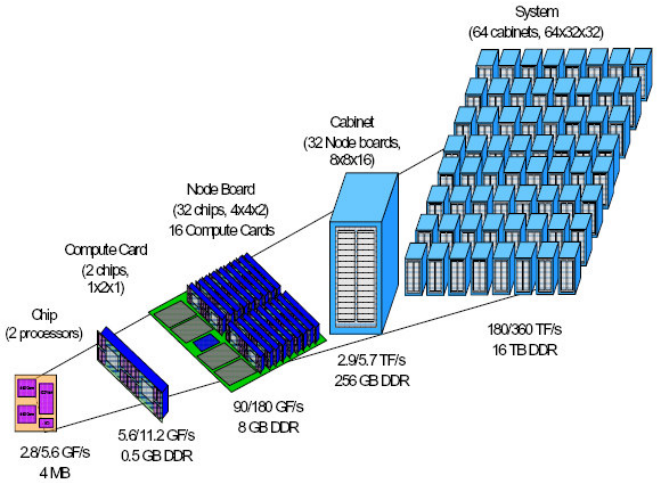

Figura 1. Packing do sistema

2.1 Nós de processamento

Um nó é formado por dois processadores PowerPC 440, com co-processadores PowerPC FP2. Este processador é um processador superscalar de 32 bits padrão. Por este motivo, as chaches L1 dos processadores do mesmo nó não mantém coerência.

Cada nó possui uma chache L2 de 2KB controlada por um mecanismo de prefetch de dados, uma memória SRAM rápida pra comunicação entre os nós, uma *L3 chache directory* e uma cache L3 compartilhada de 4MB. Além disso, também compõe um nó um controlador de ethernet gigabit, uma interface JTAG(Joint Test Action Group), uma interface para acesso a memória DDR e buffers de link de rede. A Figura 2 mostra o diagrama de blocos de um nó. Fecha a formação do nó uma *lockbox*, que provê mecanismos de *test and set* atômicos. As caches do nó são coerentes a partir da L2.

Uma forma comum de uso do nó é utilizar um núcleo para os cálculos do algoritmo e o outro para processamento de mensagens. Porém, caso o programa a ser executado necessite de pouca troca de mensagens, pode-se utilizar ambos para a execução do programa.

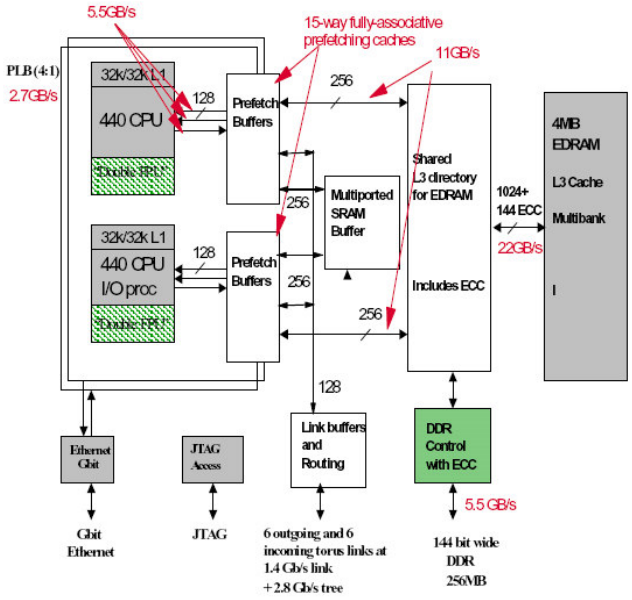

Figura 2. Diagrama de blocos de um nó

O núcleo FP2 é formada por duas unidades de ponto flutuante, cada uma com 32 registradores de 64-bit. As duas unidades são bastantes parecidas, mas há algumas diferenças. A primeria unidade age como uma unidade de ponto flutuante padrão; a segunda possui um conjunto de instruções SIMD. Algumas dessas instruções rodam paralelamente nas duas unidades. Algumas instruções rodam instruções diferentes nas unidades diferentes do FP2. Estes tipos de instruções foram denominadas de SIMOMD (Single Instruction Multiple Operation on Multiple Data).

O pipeline do processador possui latência de 5 ciclos para cada instrução, com exceção das instruçoes de divisão e *denormal*, entregando uma instrução por ciclo. As instruções de divisão

entregam o resultado iterativamente, 2 bits do quociente por ciclo.

O acesso à memória é feito de forma a maximizar a banda. O cache L2 retorna um dado em 6 a 10 ciclos, o L3 em aproximadamente 25 ciclos, e miss do L3 retorna o dado em 75 ciclos. Até 3 loads podem ser feitos em seqüência.

2.2 Redes de interconexão

O BlueGene/L possui 5 redes de interconexão, cada uma otimizada para um determinado tipo de conexão. Os tipos de redes são:

- Uma rede no formato de toro 3D, de dimensões 64x32x32;
- uma árvore de combinação/broadcast, para otimização de operações do tipo AllReduce();
- uma rede para barrier e interrupt;
- uma rede gigabit para conexão JTAG, para transmitir informações de controle e testes;
- uma rede gigabit para conexão com outros hosts.

A rede de toro é utilizada para comunicação geral entre os nós, e para operações de multicast. Ela é formada por links seriais ponto a ponto, entre roteadores embutidos nos nós. Cada roteador possui conexões com 6 vizinhos. A troca de mensagens é feita colocando as mensagens nas filas de saída. A mensagem então é enviada para a fila de entrada do próximo roteador, sendo tarefa de um dos processadores do nó o envio e a recepção destas mensagens. É implementado mecanismos de detecção de falha e de retransmissão, fornecendo guarantia de entrega.

O roteamento é feito de forma dinâmica, com *virtual cut-through* buffering[3]. Cada link possui 4 canais virtuais para troca de mensagens, mais alguns circuitos virtuais dinâmicos para troca de mensagens de roteamento e mensagens para garantir que o canal seja *deadlock free*. O roteamento é feito através do protocolo *Bubble router*[4], desenvolvido para determinar o roteamento em redes toro evitando deadlocks.

A rede em árvore é utilizada para comunicação de broadcast e de reduces. ALU's nesta rede povêem operaçoes aritméticas e bitwise para as operaçòes de reduçào em cada nó. Esta rede também é utilizada para comunicação com os nós de I/O.

3. Arquitetura de software[3]

A arquitetura de software do BG/L é composta port três entidades: um *núcleo computacional*, uma *infraestrutura de controle* e uma *infraestrutura de serviço*. Os nós de I/O rodam um kernel linux, fazendo parte da infraestutura de controle. O código de usuário é rodado nos nós que formam o núcleo computacional. Estes nós rodam um kernel linux miminalista, monoprocessado e monousuário, dedicado totalmente a aplicação sendo executada. A infraestutura de serviços é feita por programas comerciais aparte do núcleo, conectadas ao resto da arquitetura por uma rede ethernet.

O núcleo de computação pode ser particionado em conjuntos de nós isolados e idependentes, cada partição rodando um único processo.

O kernel do linux utilizado nos nós de controle necessitou de algumas modificações, para se adaptar ao mecanismos de acesso à memória, interrupção, boot e drives de dispositivo espec

3.1 Sistema operacional

O sistema operacional do BlueGene/L consiste de um kernel linux distribuído nos nós de I/O, e de pequenos kernels rodando nos nós de processamento. Este último é denominado *Blue Gene/L Run Time Supervisor* (BLRTS), e provê um espaço de endereçamento simples, sem paginaçao. Os recursos físicos são particionados entre o BLRTS e o processo do usuário. O acesso a rede fica no espaço de usuário, para aumentar a eficiência. A aplicação escreve diretamente neste espaço de memória a mensagem a ser enviada, através das funções de envio de pacotes.

O BLRTS implementa a interface POSIX. Foi portado a biblioteca GNU Glibc para prover suporte a chamadas básicas de sistema. As

chamadas de criação de processo e outras chamadas multiprocessos não foram implementadas, pois o kernel dos nós são monoprocessados. As funções de início e término de programas comunicam-se com os seus nós de I/O através da rede em árvore, usando modo de endereçamento ponto a ponto para as mensagens. Este kernel suporta a execução de processos com duas threads, sendo que cada thread é alocada para um processador, rodando exclusivamente nele.

O kernel linux rodando nos nós de I/O é um kernel padrão linux, com suporte a multiplos processos. A função destes processos é prover serviços adicionais aos nós, funcionando como o sistema operacional onde os processos estão rodando, provendo serviços como sockets de comunicação e acesso a sistemas de arquivos.

Além destes dois kernels, responsáveis pela execução das tarefas, existe um sistema operacional "global", responsável pela aplicação das políticas de funcionamento e pelo funcionamento geral do sistema, como inicialização, monitoração e início de processos. Ele roda como um serviço, numa parte em separado do sistema.

O estado do sistema é mantido em um banco de dados. Esta decisão foi feita porque banco de dados provêem confiabilidade, robustez e segurança, entre outros serviços. No banco de dados ão armazenados configurações estáticas, como conexões físicas entre nós, e dinâmicas, como o particionamento dos nós. Estas informações podem ser alteradas por processos externos, através de *stored procedures* e *triggers*, sendo um mecanismo de configuração.

3.1.1 Inicialização do sistema

Os nós não possuem disco rígido. A inicialização, portanto, deve ser feita pela rede. Ela é feita em dois passos: primeiro, um *boot loader* é escrito diretamente na memória através do protocolo JTAG. Este loader então carrega a imagem de boot efetiva do nó através do protocolo de mailbox, também pelo chip do JTAG.

É utilizado uma imagem para os nós de processamento e uma para os nós de I/O. A imagem dos nós de processamento possui tamanho de aproximadamente 64KB. A imagem dos nós de I/O é uma imagem linux de 2MB e um ramdisk com o sistema de arquivos raiz do nó. Sistemas de arquivos adicionais podem ser montados posteriormente. Como as imagens carregadas nos nós são iguais (comparadas com nós do mesmo tipo), é necessário carregar informações específicas para cada nó, como posição nas redes. Estas informações são chamadas de *personalidade(personality)* do nó.

3.1.2 Execução de jobs

A execução dos processos é iniciada pelos serviços externos. O usuário especifica o *job*(processo) a ser executado, e especifica a partição para a execução. O *scheduler* seleciona um grupo de nós e os configura para formar a partição desejada. Uma vez criada a partição, o processo é carregado nos nós de computação através dos seus respectivos nós de I/O.

3.2 Programação

O BlueGene/L utiliza a estrutura de programação do linux, utilizando o conjunto de ferramentas de compilação da GNU(binutils, gcc, gdb, etc). A compilação é feita externamente ao BlueGene/L, fazendo compilação cruzada. Existem dois alvos possíveis para a compilação: Linux para os nós de I/O, ou BLRTS para os nós de processamento.

Os programas feitos para o BlueGene/L utilizam o modelo de passagem de mensagens. Este modelo é implementado utilizando o padrão MPI. Este padrão provê um conjunto de funções para troca de mensagens entre processos.

A arquitetura de comunicação é dividida em três camadas: pacotes, mensagens e MPI.

A camada de pacotes provê mecanismos básicos para envio e recepção de pacotes. Este nível possui apenas três operações, *inicializar*, *enviar* e

receber. A inserção do pacote na fila de envio é abstraída, funcionando para as redes em toro e a árvore. Porém, fica a cargo do programador enviar pacotes com o tamanho máximo de 256 Bytes, e alinhar os dados em 16bytes. O envio e a recepção são não-bloqueantes.

A camada de mensagens provê mecanismos para envio de mensagens de formato arbritário pelo toro. Este nível possui as seguintes características:

- Sem mecanismos de retransmissão, pois a rede já possui mecanismos de retransmissão;
- Empacotamento e alinhamento dos dados, requerido pela camanda inferior. Os dados da mensagem são divididos em pacotes e enviados;
- Ordenação dos pacotes: uma mensagem pode ser dividida em vários pacotes, e estes pacotes podem chegar fora de ordem; esta camada reordena os pacotes antes de repassar à camada superior.

Para garantir uma melhor eficiência neste nível, é necessário utlizar um dos processadores do nó apenas com a função de lidar com as mensagens. Dessa forma obtém-se paralelismo entre a execução do código e tratamento de mensagens.

A camada de MPI é a camada utilizada em nível de usuário. Para suporte a essa camada foi portado uma implementação do MPICH2, com a adição de alguns módulos adicionais. Envio de Mensagens de broadcast é otimizado para envio pelo toro, enquanto mensagens de redução são enviadas pela árvore.

4. Conclusões

O supercomputador BlueGene/L é um computador massivamente paralelo formado por uma quantidade enorme de processadores de baixo custo/consumo, interligados por uma rede de interconexão em formato de toro. Programas desenvolvidos devem utilizar o modelo de passagem de mensagens, atravéz do padrão MPI.

5. Referências

[1] N.R. Adiga et al. An overview of the BlueGene/L supercomputer. *Proceeding of the IEEE/ACM SC2002 Conference* (2002).

[2] Almasi, Geroge et al. An overview of the BlueGene/L System Software Organization. *Proceedings of the Euro-Par – Springer*(2003) , 243-255.

[3] Blumrich, M. et al. Design and Analysis of the BlueGene/L Torus Interconnection Network. *IBM Research Report RC23025*. (Dez. 2003)

[4] V. Puente, C. Izu, R. Beivide, J.A. Gregorio, F. Vallejo et al. Adaptive bubble router. *Journal of Parallel and Distributed Computing* (2001). 58-67.

[5] November 2005 – Top 500 Supercomputing sites - http://www.top500.org/lists/2005/11 acesso em 22 jun. 2006