

SIMIO<sup>+</sup> - Um simulador de interconexões regulares ortogonais

Denise Stringhini<sup>1</sup>  
Lisiane Pioner Ramos<sup>2</sup>  
Roberta Jungblut<sup>3</sup>  
Paulo Fernandes<sup>4</sup>  
Philippe Navaux<sup>5</sup>

Universidade Federal do Rio Grande do Sul - UFRGS  
Instituto de Informática - II  
Curso de Pós-Graduação em Ciência da Computação - CPGCC  
Caixa Postal 15064, Porto Alegre, RS, 91501-970, BRAZIL

## RESUMO

A rede de interconexões é um fator de influência determinante no desempenho de máquinas paralelas. Este trabalho descreve a construção de uma ferramenta de auxílio na análise do grupo de interconexões regulares ortogonais, para a qual foi utilizado o método de simulação discreta orientada a eventos. Além disso, um exemplo de utilização através da avaliação de desempenho para um caso particular, é apresentado.

## ABSTRACT

The interconnection network is a determinant factor of influence in the parallel machine performance. This work describes the construction of a tool which helps the analysis of the group of orthogonal regular interconnection networks, for which was used the method named discret event simulation. Furthermore, an utilization example through the performance analysis for a particular case, is presented.

<sup>1</sup>Aluna do Curso de Pós-Graduação em Ciência da Computação - CPGCC-UFRGS. Área de interesse: Processamento Paralelo. e-mail: string@inf.ufrgs.br

<sup>2</sup>Aluna do Curso de Pós-Graduação em Ciência da Computação - CPGCC-UFRGS. Área de interesse: Processamento Paralelo. e-mail: pioner@inf.ufrgs.br

<sup>3</sup>Aluna do Curso de Pós-Graduação em Ciência da Computação - CPGCC-UFRGS. Área de interesse: Processamento Paralelo. e-mail: jungblut@inf.ufrgs.br

<sup>4</sup>Mestre em Ciência da Computação (UFRGS, 1990), Colaborador - CPGCC-UFRGS. Área de interesse: Avaliação de Desempenho.

<sup>5</sup>Doutor em Informática (Grenoble, 1979), Professor Pesquisador - CPGCC-UFRGS. Área de interesse: Processamento Paralelo. e-mail: navaux@inf.ufrgs.br

## 1 Introdução

O desenvolvimento de computadores compostos por vários processadores funcionando em conjunto surgiu da necessidade de aumentar o desempenho, já que máquinas seriais são limitadas. Esses computadores, denominados máquinas paralelas, têm seus processadores trabalhando na resolução de uma mesma tarefa. Uma parte da tarefa é atribuída a cada um dos processadores que podem cooperar entre si através da troca de mensagens.

A troca de mensagens pode ocorrer através de uma memória global ou diretamente entre os processadores. Quando existe uma memória de acesso comum a todos os processadores, toda a troca de dados, via de regra, é feita através desta, e cada processador possui uma memória para armazenar os dados e instruções mais utilizados. No outro caso, quando a troca de mensagens é feita diretamente entre os processadores, eles estão ligados através de canais de comunicação e cada processador possui dispositivos de entrada e saída e uma memória local para armazenar dados e instruções [HWA 84, KIT 90].

Em ambos os casos, a forma como os processadores são conectados define a rede de interconexão. Esta estrutura define o caminho que a mensagem deverá percorrer, o que influencia diretamente no tempo de comunicação entre os processadores e, conseqüentemente, no desempenho global. Desta forma, os computadores com processamento paralelo necessitam de uma eficiente rede de interconexão para que não haja desperdício de tempo na comunicação entre os processadores.

Na medida em que a eficiência da rede de interconexão é importante, se torna necessário que seu desempenho seja avaliado. Existem diversos métodos de avaliação de desempenho que apresentam resultados bastante confiáveis. Neste trabalho foi adotado o método de simulação [FER 90] para avaliar o desempenho de interconexões regulares ortogonais.

Simulação é uma técnica de avaliação de desempenho que representa, através de um modelo, o comportamento de um sistema real em um determinado período de tempo [FER 78, SHA 75]. Como modelo entende-se a representação formal, simbólica da realidade, isto é, representação por alguma outra forma que não seja a da realidade em si. Para uma avaliação deve-se planejar um modelo que considere os principais fatores que determinam o desempenho da realidade, definir as medidas de desempenho do modelo e utilizar essas medidas como estimativas do comportamento do sistema real [SAU 81].

Neste método, a realidade é representada através de um programa de computador que imita o comportamento do sistema real e fornece as medidas de desempenho baseado no que foi observado durante a simulação [JAI 91, SMI 79]. Este é um método que, por sua generalidade, permite a modelagem de realidades complexas e, por esse motivo, foi adotado neste trabalho.

Com base neste método, foi implementado, em linguagem C, o SIMIO<sup>+</sup> [STR 93], um simulador discreto orientado a eventos. Isto foi feito em estações de trabalho Sun, cujo sistema operacional é o SunOS R.4.1.1, no ambiente de janelas Open Windows V.3.

No decorrer deste trabalho, são abordados primeiramente as interconexões estudadas, posteriormente é apresentado o simulador para, finalmente, exemplificarmos a avaliação de uma interconexão.

## 2 Interconexões Regulares Ortogonais

As redes de interconexões regulares consistem de  $n$  processadores interconectados de maneira idêntica, por caminhos dedicados, segundo um padrão de ligação [AND 75, VAN 89, BHU 87]. Mensagens são enviadas através da rede, da origem para o destino, com os processadores intermediários determinando qual dos processadores vizinhos será o próximo a receber a mensagem. Entre as mais conhecidas cita-se: árvore, malha, sistólica, cubo.

As interconexões regulares ortogonais, que são o escopo deste trabalho, são um subgrupo das regulares que caracterizam-se pela simetria de sua estrutura e pela disposição ortogonal das ligações entre os processadores. Para uma melhor descrição dos tipos que serão abordados, foi elaborada uma taxonomia própria que considera o número de dimensões e a presença de ligações entre os processadores dos extremos de uma mesma dimensão. De acordo com esta taxonomia, apresentam-se os seguintes tipos:

- **Bidimensional:** nesta estrutura os processadores estão conectados aos seus vizinhos mais próximos nos sentidos horizontal e vertical. Isto a caracteriza como sendo uma matriz de processadores de  $m \times n$  elementos, onde  $m$  e  $n$  são o número de processadores na primeira e segunda dimensões, respectivamente [CAR 88, FER 88]. De acordo com a presença ou não de ligações entre suas extremidades, pode ser dividida em dois tipos (fig. 1):

- *aberta:* sem ligações nos extremos;
- *fechada:* com ligações nos extremos.

As estruturas bidimensionais aberta e fechada são comumente conhecidas como Mesh [MAT 90, HWA 84] e Torus [FER 88], respectivamente.

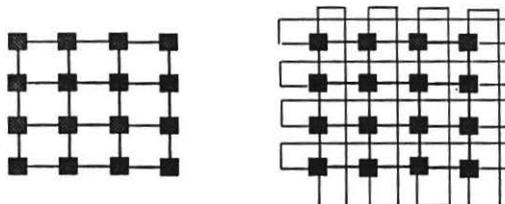


Figura 1: Estruturas Bidimensionais: Aberta e Fechada

- **Tridimensional:** são estruturas em forma de cubo, onde cada face é composta por, no mínimo, quatro processadores. Podem ser classificadas, de acordo com as ligações nas extremidades, em abertas e fechadas (fig. 2). A estrutura tridimensional que possui apenas quatro processadores em cada face é conhecida por cubo ou hipercubo 3D e as demais por supercubo 3D; entretanto, cabe salientar que alguns autores divergem quanto a esta nomenclatura. Informações sobre cubo e hipercubo em geral, podem ser encontradas em [DUN 91, MAT 90, REE 87, ROS 85, WEL 85].

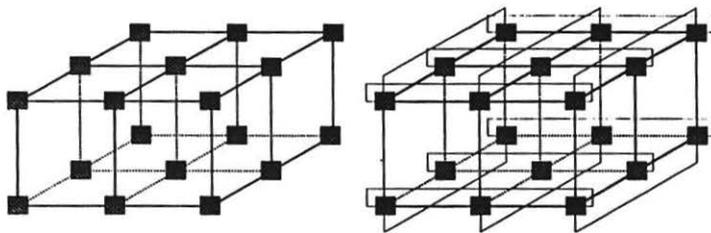


Figura 2: Estruturas Tridimensionais: Aberta e Fechada

- **Quadridimensional:** é uma extensão da estrutura anterior, pois, a partir da conexão de todos os processadores de duas tridimensionais, obtém-se uma quadridimensional (fig. 3). A exemplo das anteriores, podem ser classificadas em abertas e fechadas. Quando possuem quatro processadores em cada face são conhecidas como hipercubo 4D [FER 88, HWA 89] e, as demais, como supercubo 4D.

Considera-se somente estruturas com até quatro dimensões, por serem estas, as mais frequentemente encontradas. Entretanto, a taxonomia aqui apresentada é facilmente expandível, visto que pode englobar estruturas com maior número de dimensões.

### 3 SIMIO<sup>+</sup> - Características do Simulador

A modelagem de uma realidade, através da simulação discreta orientada a eventos, implica em transpor suas características para um programa de computador. Para este simulador, optou-se pela linguagem de programação C.

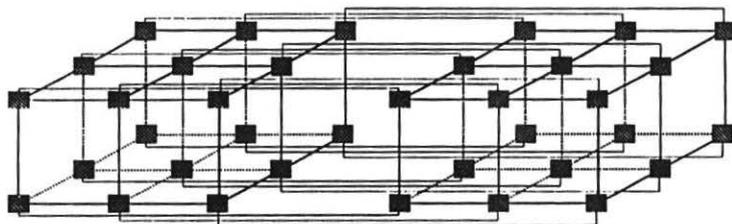


Figura 3: Estrutura Quadridimensional

que é uma linguagem genérica, ao invés de uma linguagem específica para simulação. As linguagens genéricas têm a vantagem de serem mais conhecidas e, em geral, produzem simuladores mais rápidos do que uma linguagem de simulação, porém estas, quando utilizadas, facilitam a programação, na medida em que possuem rotinas específicas para simulação. Além disso, optou-se pela implementação em estações de trabalho *Sun*, devido às facilidades de hardware e software disponíveis.

Esta seção descreve como foi realizada a implementação do simulador, salientando-se que o objetivo foi simular o comportamento das interconexões regulares ortogonais em relação à troca de mensagens entre os processadores, e fornecer resultados que possibilitem a avaliação de desempenho.

### 3.1 Definição dos Eventos

Cada evento é um acontecimento no simulador que corresponde a uma determinada característica de funcionamento do sistema real. Para a modelagem das interconexões regulares ortogonais, foram definidos os seguintes eventos: *início de processamento local*, *fim de processamento local*, *tentativa de transmissão*, *início de transmissão/recepção*, *fim de transmissão* e *fim de recepção*.

A escolha do evento a ser simulado é feita através do primeiro nó da fila de eventos. Na figura 4 pode ser visualizado o ciclo básico dos eventos de um processador<sup>1</sup>. Estes eventos serão descritos a seguir:

- **Início de Processamento Local:** marca o momento em que o processador inicia o processamento de informações internamente. Neste momento, o simulador deve definir a duração deste processamento e gerar um evento *Fim de Processamento Local*.
- **Fim de Processamento Local:** marca o momento do término do processamento interno de informações. Para simular a necessidade da troca de mensagens em um sistema real, foi definido que, a cada *Fim de Processamento Local*, será gerada uma mensagem a um destinatário definido aleatoriamente. Para que a mensagem chegue ao seu destino, este evento deve escolher um intermediário, se houver, e criar imediatamente um evento *Tentativa de Transmissão*.
- **Tentativa de Transmissão:** é responsável por verificar se o processador requisitado (intermediário ou destino) está ou não ocupado. Se estiver, deve ser gerada uma nova tentativa, que poderá ser para algum outro processador. Esta nova tentativa será colocada na fila de eventos de acordo com o tempo de espera entre tentativas definido pelo usuário. Se o processador requisitado estiver desocupado, deve ser criado imediatamente um evento *Início de Transmissão/Recepção*.
- **Início de Transmissão/Recepção:** simula o momento em que dois processadores iniciam o processamento global, ou seja, o momento em que a mensagem começa a ser transmitida de um para outro. A duração deste processamento global é a mesma para os dois processadores, devendo ser criados, ao final deste processamento, os eventos *Fim de Transmissão* e *Fim de Recepção* para o transmissor e receptor, respectivamente.

<sup>1</sup> Note-se que foram desconsiderados diversos fatores que podem desviar o processador atual deste ciclo, assim como desconsiderou-se a hipótese de o processador receptor da mensagem não ser o processador destino. O objetivo foi fornecer uma visão simplificada do ciclo de eventos, para facilitar a compreensão do funcionamento do simulador.

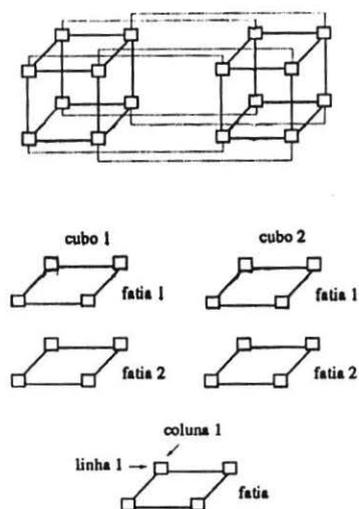


Figura 5: Representação CFLC

números, da esquerda para a direita, correspondem à identificação do cubo ao qual pertence o processador, seguidos por dois números para a fatia, linha e coluna. A figura 6 exemplifica o CFLC, demonstrando a representação de cada processador em uma estrutura quadridimensional  $2 \times 2 \times 2 \times 2$ .

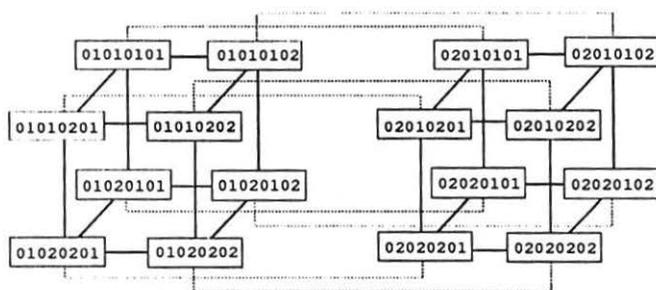


Figura 6: Exemplo da representação CFLC

### 3.3 Parâmetros de Entrada

O número de opções oferecidas ao usuário determina a flexibilidade do simulador a nível de modelagem. Essas opções são os parâmetros de entrada, que irão definir as características do modelo a ser simulado, tanto em relação aos aspectos físicos quanto aos estruturais e comportamentais. Os parâmetros de entrada do SIMIO<sup>+</sup> são descritos a seguir.

### 3.3.1 Topologia

Segundo a classificação vista na seção 2, as interconexões regulares ortogonais podem ser divididas em três tipos básicos: bidimensional, tridimensional e quadridimensional. Além disso, cada uma destas interconexões pode ser classificada em aberta ou fechada. Portanto, existem seis tipos de interconexões diferentes que podem ter seus desempenhos avaliados e comparados através do SIMIO<sup>+</sup>.

Além de poder escolher um desses seis tipos de interconexões, o usuário poderá, também, definir o número de processadores em cada dimensão. Estes fatores permitem que se possa simular inúmeras interconexões diferentes a nível de estrutura. Isso quer dizer que, apenas através da variação da topologia, inúmeros modelos de redes de interconexões podem ser simulados e ter seus resultados comparados.

Note-se que os algoritmos utilizados neste simulador suportam estruturas com um número de dimensões superior. A restrição de até quatro dimensões foi feita por ter-se constatado que estruturas com um número maior de dimensões não são muito utilizadas e sua simulação, devido ao hardware disponível, se tornaria muito demorada.

### 3.3.2 Modo de Recepção

A maneira como um processador se comporta quando outro tenta lhe enviar uma mensagem influencia no desempenho da máquina em relação a troca de mensagens. Por este motivo, este simulador possibilita que se opte por dois tipos de funcionamento para a recepção de mensagens, que são:

- **Sem preempção:** quando se simula este modo de recepção, dá-se a mesma prioridade a todas atividades do processador, não se permitindo, portanto, que uma atividade seja interrompida por outra. O processador só recebe mensagens quando está inativo, isto é, esperando para transmitir ou receber uma mensagem.
- **Com preempção:** neste modo, a troca de mensagens recebe uma prioridade maior, sendo assim, permite-se que o processamento local seja interrompido para recepcionar uma mensagem. Portanto, a recepção de mensagens pode acontecer quando o processador está inativo ou em processamento local.

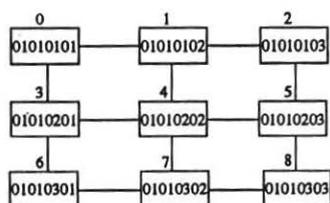
### 3.3.3 Forma de Roteamento

O caminho que uma mensagem percorre do processador origem até o destino é definido pelo roteamento. Para garantir a eficiência de uma rede de interconexões é necessário que esta mensagem chegue ao seu destino no período mais curto possível. Normalmente, isso é conseguido fazendo com que a mensagem percorra o menor caminho até o destino, ou seja, passe pelo menor número de intermediários. Neste simulador, o roteamento é realizado levando em consideração as seguintes informações: processador portador da mensagem, processador destino e forma de roteamento, que pode ser **não inteligente** ou **inteligente**.

Caso a forma de roteamento seja do tipo **não inteligente**, os processadores intermediários são obtidos da seguinte maneira: escolhe-se, como primeiro intermediário, o vizinho do processador portador da mensagem que mais se aproxime da coluna do processador destino; para os demais intermediários, repete-se este procedimento até que o processador portador da mensagem pertença a mesma coluna do processador destino. Em seguida faz-se o mesmo para a linha, a fatia e o cubo, nesta ordem. Com este procedimento, o menor número possível de intermediários é percorrido até que a mensagem chegue ao processador destino.

Se a forma de roteamento for **inteligente** é possível, utilizando o mesmo conceito de procurar um intermediário que se aproxime da coluna, linha, fatia e cubo aos quais pertence o destino, utilizar uma outra ordem de aproximação que a definida no roteamento **não inteligente**. Isto quer dizer que, se após um determinado número de tentativas, o processador intermediário escolhido não estiver disponível, um processador de outra dimensão pode ser requisitado. A vantagem deste método é permitir que o caminho da mensagem seja alterado sem que se aumente o número de processadores intermediários. O número de tentativas por um intermediário é determinado antes de se iniciar a simulação.

A representação **CFLC** para o identificador dos processadores permite que se saiba quais são os vizinhos de



Bidimensional Aberta 3x3

ORIGEM ..... 01010301  
 DESTINO ..... 01010103  
 INTERMEDIÁRIO 1 ..... 01010302  
 INTERMEDIÁRIO 2 ..... 01010303  
 INTERMEDIÁRIO 3 ..... 01010203

Figura 7: Exemplo de roteamento Não Inteligente

cada processador. Por exemplo, para que uma mensagem se aproxime da coluna do seu processador destino, basta que se varie em uma unidade, o valor do identificador correspondente à coluna. Assim, vão sendo definidos todos os processadores intermediários pelos quais a mensagem deve passar. A figura 7 apresenta um exemplo de roteamento para uma estrutura bidimensional aberta 3x3.

### 3.3.4 Duração das atividades do processador

Os tempos em que os processadores permanecem em processamento local, processamento global e espera por recepção, assim como o tempo que deve transcorrer entre uma tentativa de transmissão mal sucedida e uma nova tentativa, são importantes na modelagem do sistema. Por este motivo, este simulador permite que o usuário entre com valores médios para os três primeiros, e, a partir destes valores, os tempos são gerados aleatoriamente, com base em uma distribuição estatística. Dentre as distribuições existentes, o usuário poderá optar por:

- constante: estabelece um valor fixo para a duração do evento;
- uniforme: a duração do evento poderá assumir qualquer valor dentro do raio de 1 u.t. em torno do valor fornecido pelo usuário;
- exponencial: o usuário fornece um tempo médio em torno do qual será gerada a duração do evento.

Já o tempo entre as tentativas, assume um valor fixo, fornecido pelo usuário. Isto porque correspondem a uma atividade determinística do processador, ou seja, a cada tentativa mal sucedida, é possível que se conheça o tempo da próxima.

### 3.4 Parâmetros de Saída

Os parâmetros de saída são os resultados da simulação, que refletem o comportamento do modelo e devem ser relevantes e suficientes para que, através deles, seja realizada a análise de desempenho. Para que se chegue a esses resultados, são realizadas diversas contabilizações no decorrer da simulação, que são as observações feitas sobre seu andamento. Ao final da simulação, são feitos cálculos, utilizando as contabilizações, e é gerado o relatório correspondente aos seus resultados.

- **Número médio de mensagens geradas por u.t.:** reflete a carga de mensagens geradas pelos processadores em cada unidade de tempo e que circularam pela rede no decorrer da simulação. Note-se que as mensagens retransmitidas por processadores intermediários não são contabilizadas.

$$n^{\circ} \text{ medio de msg geradas por u.t.} = \frac{img}{Rf}$$

Onde: *img* é o total de mensagens geradas e *Rf* é o valor final do relógio de simulação.

- **Número médio de mensagens recebidas por u.t.:** em contrapartida ao anterior, informa o número médio de mensagens efetivamente recebidas pelos processadores destinos.

$$n^{\circ} \text{ medio de msg recebidas por u.t.} = \frac{tmr}{Rf}$$

Onde: *tmr* é o total de mensagens recebidas por seus destinos.

Utilizando-se esses dois parâmetros em conjunto, pode-se chegar a algumas conclusões, por exemplo, se a diferença entre o número médio de mensagens geradas por u.t. e o número médio de mensagens recebidas por u.t. for estatisticamente relevante, pode-se concluir que o tempo de simulação foi insuficiente, ou, após várias execuções do simulador, que a rede modelada é ineficiente para a troca de mensagens, etc.

- **Número médio de mensagens geradas por u.t. por processador:** reflete a responsabilidade dos processadores pela carga de mensagens que circularam pela rede.

$$n^{\circ} \text{ medio de msg geradas por u.t. por proc} = \frac{n^{\circ} \text{ medio de msg geradas por u.t.}}{nprocs}$$

Onde: *nprocs* é o número total de processadores da topologia.

- **Número médio de mensagens recebidas por u.t. por processador:** informa a quantidade média de mensagens que foram efetivamente recebidas por seus respectivos processadores destinos.

$$n^{\circ} \text{ medio de msg recebidas por u.t. por proc} = \frac{n^{\circ} \text{ medio de msg recebidas por u.t.}}{nprocs}$$

- **Utilização média dos processadores:** os processadores estão em utilização, ou seja, em atividade, quando estão em um dos seguintes estados: processamento local ou processamento global (transmissão/recepção)<sup>2</sup>. A utilização média corresponde ao tempo de atividade de cada processador em cada unidade de tempo. Desta forma, pode-se avaliar, entre outras coisas, se há um número excessivo de processadores na rede, conclusão à qual se pode chegar se, por várias execuções, a utilização for muito baixa, por exemplo.

$$\text{utilização média dos processadores} = 1 - \frac{aep}{Rf}$$

Onde: *aep* é o acumulador dos tempos de espera dos processadores.

- **Tempo médio de espera para enviar mensagem:** estima o tempo médio que cada processador teve que esperar desde a primeira tentativa até conseguir iniciar uma transmissão. Este parâmetro é importante na medida em que é um dos fatores utilizados para medir o desempenho do modelo principalmente em relação ao tipo de funcionamento e comportamento adotados, como o modo de recepção, a forma de roteamento, o tempo médio de processamento local, etc.

$$\text{tempo medio espera para enviar mensagem} = \frac{aem}{tmt}$$

Onde: *aem* é o acumulador dos tempos de espera das mensagens e *tmt* é o total de mensagens transmitidas.

- **Número médio de tentativas de transmissão:** informa o número de tentativas que um processador, em média, teve que realizar até conseguir transmitir uma mensagem. Da mesma forma que o anterior, reflete principalmente o desempenho do sistema em relação às características de funcionamento e comportamento escolhidos na modelagem.

<sup>2</sup> São consideradas somente transmissões/recepções nas quais o processador é o origem ou o destino da mensagem.

$$n^{\circ} \text{ médio de tentativas} = \frac{t_{\text{tent}}}{t_{\text{mt}}}$$

Onde:  $t_{\text{tent}}$  é o total de tentativas de transmissão.

- **População média em processamento local:** indica quantos processadores, em média, permaneceram em processamento local em cada unidade de tempo. O processamento local é o tempo em que os processadores estão efetivamente utilizando seus recursos internos para processar informações.
- **População média em processamento global:** indica quantos processadores, em média, permaneceram em processamento global em cada unidade de tempo. Entende-se como processamento global o tempo em que dois processadores permanecem interligados, um transmitindo e outro recebendo a mensagem.
- **População média em espera por transmissão:** indica o número médio de processadores que permaneceram em espera por transmissão em cada unidade de tempo.
- **População média em espera por recepção:** indica o número médio de processadores que permaneceram em espera por recepção em cada unidade de tempo. A espera por recepção é um tempo de inatividade do processador que significa que ele está esperando por alguma mensagem ou terminou sua tarefa e ainda não foi requisitado para alguma outra.

$$\text{população média} = \frac{A}{\pi f}$$

Onde:  $A$  é a área sob a curva de processadores em espera por recepção.

#### 4 Exemplo de uma Avaliação de Desempenho

Com o objetivo de comparar o desempenho de algumas interconexões regulares ortogonais foi realizada uma avaliação onde buscou-se tomar esta comparação dependente apenas dos tipos de topologias escolhidos: bidimensional aberta, tridimensional aberta e quadridimensional aberta.

Nesta experiência, verificou-se o comportamento destas interconexões variando-se o número de processadores e a forma pela qual eles estão interconectados. Além disto, considerou-se que o tempo total de execução de um problema (aplicação) seria dividido pelo número de processadores, ficando, cada processador, responsável por uma parte da tarefa. Com isto, desejou-se saber até que ponto é interessante aumentar o número de dimensões, assim como verificar qual a influência da comunicação interprocessadores na execução da tarefa.

Para isso, os parâmetros de entrada do *SIMIO*<sup>+</sup> permitem que sejam definidas algumas características da aplicação e do funcionamento da arquitetura. Estas características são a carga de processamento para cada processador, o tempo de comunicação entre eles, o modo de recepção, a forma de roteamento, etc, os quais foram descritos na seção 3. As definições adotadas para estes parâmetros são apresentadas a seguir.

Os tempos médios de processamento local foram definidos tomando como base uma aplicação com tempo computacional total de 2048 u.t., considerando sua execução em uma máquina uniprocessadora. Definiu-se o tempo de processamento local para cada processador (tabela 1) como  $1/N$  deste tempo computacional total, onde  $N$  é o número de processadores da topologia a ser simulada. A fim de considerar possíveis variações no tempo de processamento local de cada processador, considerou-se a distribuição exponencial como a mais adequada.

Num. Proc.	Proc. Local (u.t.)
8	256
16	128
32	64

Tabela 1: Tempos Médios de Processamento Local

Já o tempo médio de processamento global foi fixado com um mesmo valor para todas as topologias, pois as mensagens foram consideradas como sendo do mesmo tamanho para todos os casos. Como o objetivo da análise

era demonstrar o efeito da variação da topologia no número de comunicações entre os processadores, considerou-se como satisfatório um valor que representasse um tempo de comunicação mínimo, ou seja, não afetasse o desempenho; assim, definiu-se o processamento global como sendo aproximadamente 1% (20 u.t./exponencial) do tempo computacional total.

Da mesma forma, o tempo médio de espera por recepção de mensagens e o tempo de espera entre tentativas foram fixados com valores mínimos em, respectivamente, 5 u.t./constante e 1 u.t. para todos os casos.

Optou-se pelo modo de recepção Com Preempção para priorizar a comunicação entre os processadores. Já a forma de roteamento, foi definida como Não Inteligente por ser eficiente e, com frequência, encontrada em arquiteturas existentes.

Os resultados que serão apresentados a seguir se referem aos parâmetros de saída população média em processamento local e população média em processamento global. Estes resultados são médias obtidas a partir de várias simulações.

Os valores da tabela 2 para 8 processadores refletem o efeito de uma aplicação com granularidade alta, visto que os processadores se comunicam pouco, passando a maior parte do tempo em processamento local. Pode-se notar que o aproveitamento da bidimensional foi bom, em média 72,86% dos processadores permaneceram em processamento local em cada unidade de tempo. Já a tridimensional se mostrou levemente superior (cerca de 2%). Neste caso particular, não haveria necessidade de se utilizar uma topologia tridimensional, já que sua vantagem sobre a outra foi muito pequena.

Tipo de Atividade	BIDI		TRIDI	
	População	%	População	%
Local	5.8284	72,86	5.9913	74,89
Global	1.8980	23,73	1.7469	21,83
Outras	0.2734	3,38	0.2579	3,28

Tabela 2: Resultados para 8 processadores

A tabela 3 apresenta os resultados obtidos com 16 processadores. Pode-se observar que na bidimensional apenas 46% destes processadores permaneceram, em média, em processamento local, sendo que o restante se encontrava em outras atividades. Acrescentando-se uma dimensão, obteve-se uma melhora mínima (cerca de 3%) o que não se considerou como satisfatório, já que apenas 50% dos processadores estavam em processamento local a cada unidade de tempo. Já com a quadridimensional, superou-se, ainda que não muito, esta marca de 50%, o que fez com que esta topologia fosse considerada a melhor opção dentre as existentes. Isto deve-se ao fato da quadridimensional possuir um menor número de processadores intermediários para as comunicações mais distantes.

Tipo de Atividade	BIDI		TRIDI		QUADRI	
	População	%	População	%	População	%
Local	7.4883	46.80	7.9464	49.66	8.6423	54.00
Global	7.1953	44.97	6.8421	42.76	6.2393	39.00
Outras	1.3204	8.23	1.2209	7.58	1.1051	7.00

Tabela 3: Resultados para 16 processadores

Na tabela 4 pode-se visualizar os resultados obtidos com 32 processadores. Analisando-a, percebe-se que, para as três interconexões, o número médio de processadores em processamento local é bem menor que o número médio de processadores em outras atividades, ou seja, estas interconexões não se mostraram eficientes para este caso. Isto acontece porque uma mensagem, em qualquer uma destas interconexões, deve percorrer muitos intermediários. Com isso, muitos processadores acabam fazendo o papel de canal de comunicação, ao invés de executar processamento útil. Também foram realizadas simulações para 64 e 128 processadores, onde verificou-se que este comportamento se acentuou.

Tipo de Atividade	BIDI		TRIDI		QUADRI	
	População	%	População	%	População	%
Local	6.6170	20.68	8.2032	25.63	8.9835	28.07
Global	21.1971	66.24	19.6783	61.49	19.0368	59.49
Outras	4.2199	13.08	4.1255	12.88	3.9518	12.44

Tabela 4: Resultados para 32 processadores

Concluiu-se que, para esta aplicação, caso se deseje uma menor influência da comunicação, é melhor que se opte pelos casos onde o número de processadores é menor e o tempo de processamento local maior. Isto acontece porque, nos casos analisados, determinou-se que um processador tem a mesma probabilidade de referenciar qualquer processador da rede, mesmo sendo distante. Este tipo de comportamento é adequado para aplicações com granularidade alta, já que aplicações com granularidade fina exigem que haja uma certa localidade na comunicação entre os processadores. Com isso, a medida em que a granularidade foi diminuindo, as interconexões simuladas foram perdendo a sua eficiência.

## 5 Conclusão

No decorrer deste trabalho foi demonstrada a construção de uma ferramenta para a avaliação de desempenho das redes de interconexão regulares ortogonais. A partir daí, mostramos ser a simulação um método bastante adequado para isto, visto que os objetivos de se obter um simulador que abrangesse vários aspectos da realidade foram alcançados.

Cabe salientar que a flexibilidade do simulador deve-se ao usuário poder decidir vários detalhes da interconexão, os quais representam seu comportamento, estrutura e carga de processamento. Outro indicativo da flexibilidade do simulador é que este apresenta vários parâmetros de saída, os quais permitem que se avalie a interconexão sob diversos pontos de vista.

A partir de análises aqui realizadas, pôde-se verificar que é possível chegar a várias conclusões, mesmo variando-se apenas dois parâmetros de entrada e analisando-se apenas alguns dos parâmetros de saída. Portanto, apesar da tarefa de avaliação ser muito complexa, e conclusões definitivas só poderem ser tomadas a partir da análise de todos os aspectos relevantes, através do simulador pode-se ter noções de comportamento muito interessantes das topologias abordadas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [AND 75] Anderson, G. A. & Jensen, E. D. *Computer Interconnection Structures: Taxonomy, Characteristics and Examples* ACM Computing Surveys, Vol. 7, No. 4, Dez 1975.
- [BHU 87] Bhuyan, Laxmi N. *Interconnection Networks for Parallel and Distributed Processing* Computer, Jun 1987.
- [CAR 88] Carlson, David A. *Modified Mesh-Connected Parallel Computers* IEEE Transactions on Computers, Vol. 37, No. 10, Out 1988.
- [DUN 91] Dunigan, T. H. *Performance of the Intel iPSC/860 and NCube 6400 Hypercubes Parallel Computing* Parallel Computing, No. 17, 1991.
- [FER 78] Ferrari, Domenico *Computer Systems Performance Evaluation* Prentice-Hall, 1978.
- [FER 88] Fernandes, P. *Análise Quantitativa de Interconexões Regulares* Relatório Interno, CPGCC - UFRGS, Jun 1988.
- [FER 90] Fernandes, P. *Modelos Analíticos para Interconexão de Processadores: Avaliação de Desempenho de Alocação Simultânea de Recursos* Dissertação de Mestrado, CPGCC - UFRGS, 1990.

- [HWA 84] Hwang, Kay & Briggs, Fayé A. *Computers Architectures and Parallel Processing* McGraw-Hill, 1984.
- [HWA 89] Hwang, Kay & Degroot, Douglas *Parallel Processing for Supercomputers & Artificial Intelligence* McGraw-Hill, 1989.
- [JAI 91] Jain, Raj *The Art of Computer Systems Performance Analysis* John Wiley & Sons, Inc., 1991.
- [KIT 90] Kitajima, João Paulo F. W. *Tempos de Comunicação em Multiprocessadores* Dissertação de Mestrado, CPGCC - UFRGS, 1990.
- [MAT 90] Matic, Spira *Emulation of Hypercube Architecture on Nearest Neighbor Mesh-Connected Processing Elements* IEEE Transactions on Computers, Vol. 39, No. 5, Mai 1990.
- [RFE 87] Reed, Daniel A. & Grunwald, Dirk C. *The Performance of Multicomputer Interconnection Networks* Computer, Jun 1987.
- [ROS 85] Rosenberg, Robert *Super Cube* Eletronics Week, Fev 1985.
- [SAU 81] Sauer, Charles H. & Chandy, K. *Computer Systems Performance Modeling* Prentice-Hall, 1981.
- [SHA 75] Shannon, Robert E. *Systems Simulation, The Art and Science* Prentice-Hall, 1975.
- [SMI 79] Smith, B. S. & Lewis, T. G. *Computer Principles of Modeling and Simulation* Houghton Mifflin Company, 1979.
- [STR 93] Stringhini, D.; Ramos, L.P.; Jungblut, R.; Fernandes, P.H.L. & Navaux, P.O.A. *SIMIO+ - Simulador de Interconexões Regulares Ortogonais* Relatório de Pesquisa, CPGCC - UFRGS, Nov 1993.
- [VAN 89] Van de Goor, A. J. *Computer Architecture and Design* Addison-Wesley Publishers, 1989.
- [WEL 85] Welty, Lyle L. & Patton, Peter C. *Hypercubes Architectures* National Conference, 1985.