Acta Scientiarum 22(5):1321-1325, 2000.
ISSN 1415-6814.

Performance analysis of interconnection networks for the Wolf architecture

Marcos Antônio Cavenaghi[1], João Angelo Martini[2]* e Álvaro Garcia Neto[3]

*[1]Departamento de Computação, Universidade Estadual Paulista, Bauru-São Paulo, Brazil, [2]Departamento de Informática, Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá-Paraná, Brazil. [3]Departamento de Física e Informática, Universidade de São Paulo, São Carlos-São Paulo, Brazil. *Author for correspondence. e-mail: jangelo@din.uem.br*

ABSTRACT. A performance analysis of interconnection networks for the Wolf architecture is provided. Based on communication requirements, interconnection networks for simulation using the SAW simulator were selected. SAW is a time-driven simulator that implements the Wolf characteristics. The parameters execution time, average memory occupation and cost are analyzed to investigate the architecture cost-efficiency relationship. Simulations performed led to non-conclusive results concerning the most cost-effective interconnection network for the Wolf architecture.

Key words: interconnection networks, parallel architectures, performance analysis.

RESUMO. Análise de desempenho de redes de interconexão para a arquitetura Wolf. Este trabalho apresenta uma análise do desempenho das redes de interconexão da arquitetura Wolf. Com base nos requisitos de comunicação, selecionamos algumas redes de interconexão para simulação usando o simulador SAW. SAW é um simulador dirigido a tempo que implementa as características do Wolf. Os parâmetros tempo de execução, ocupação média de memória e custo, foram usados para investigar a relação custo-desempenho da arquitetura. As simulações realizadas levaram a resultados não conclusivos a respeito da rede de interconexão com melhor relação custo-desempenho para a arquitetura Wolf.

Palavras-chave: redes de interconexão, arquiteturas paralelas, análise de desempenho.

This work presents a complete analysis of interconnection networks for the Wolf architecture. In Martini (Martini *et al.*, 1998) only preliminary results concerning execution time were given. Besides execution time, present research investigates too average memory occupation and cost.

Interconnection networks play an important role in parallel architecture context. They constitute the basic structure of the communication system for the parallel architecture and influence the performance and reliability of such systems. The design of interconnection networks is a tradeoff between cost, performance and complexity of control. The ideal solution allows all interconnected units to communicate simultaneously without any conflict, although such network is expensive and complex. Searching for an alternative combining cost and performance is the main goal of this research work.

Besides the choice of the most adequate topology, other decisions also play an important role in designing interconnection networks, such as operation mode, control strategy and switching methods. The operation mode deals with the way data are transmitted. There are two basic types: synchronous and asynchronous. A global clock that transmits signals to all components of the system in order to synchronize them characterizes the synchronous mode. On the other hand, the asynchronous mode operates without a global clock, allowing better expandability with higher complexity.

The control strategy corresponds to the way control signals are generated. They coordinate the routing functions of a network. This control can be centralized or distributed. In the centralized mode, a unique control unit generates signals that may compromise the reliability of the system and require a complex controller in order to obtain a good performance. In distributed mode, signals are originated locally by system components, eliminating the need for a complex global controller.

The switching method deals with the physical utilization of switching elements in order to obtain a routing function. There are two principal switching methodologies, or rather, circuit-switching and packet-switching. In circuit-switching, a physical path is set between source and destination before the starting of communication and maintenance of the transmission path. In packet-switching, messages are divided into smaller packets to be transmitted through the network by switching elements.

Wolf architecture

The Wolf architecture (Cavenaghi *et al*., 1996, Cavenaghi *et al*., 1998, Cavenaghi *et al*., 2000) is shown in Figure 1. Communication between its units is implemented by exchange of tokens. Each token represents a datum in the context of the architecture. The communication requirements are briefly described below.

Tokens are inserted into the machine via the Input stream and answers are sent to the host computer via the Output stream. From the Collecting Network (CN), tokens pass through the Token Queue (TQ) to the Data Memory (DM). The DM is responsible for matching operands and thus both operands of a dyadic operator are made available to subsequent units. It forms a new token named PA (Pair) composed by a pair of tokens directed to the same node in the data-dependency graph and with the same label. The PA token is distributed through the Distributed Network (DN) to one of the available parallel Instruction Memory (IM).

The IM holds the details of the node the pair of operands has reached. It builds a token named EX (Executable) containing all the information required to execute the node and forwards it to its associated Functional Unit (FU). The results produced by the FU are passed to CN. If the CN cannot forward the received tokens, TQ stores them to form a pool of available data for later distribution.

The Wolf simulator (SAW) is a time-driven simulator. It was developed in the object-oriented language C++. The first version of SAW used two ideal interconnection networks named Cross1 and Cross2 for the Distributing Network and Collecting Network respectively. The communication requirements for the Wolf architecture have been investigated in order to identify the set of interconnection networks that are more suitable for the architecture. Based on these investigations, three interconnection networks have been simulated derived from Crossbar (Bhuyan *et al*., 1989), Delta (Hwang, 1984, Ibbett, 1989) and Generalized Shuffle Network (GSN) (Bhuyan, 1983).

Implementation of the collecting network

The Collecting Network was implemented through the employment of the following interconnection structures: Crossbar, GSN and Delta interconnection networks. The Crossbar can address tokens to TQ, Output stream and DM. An Arbitrator coordinates the transmission of tokens addressed to DM, simplifying the network. This implies a simplified control and avoids the excessive re-routing as in Cross2.

The Arbitrator coordinates the tokens addressed to DM. It verifies the availability of that unit to send tokens or not. It can also send tokens to TQ if DM is not available. If the Arbitrator is not able to send a token either to DM or to TQ, it blocks the token and waits until one of the two units becomes available. GSN and Delta networks have a similar behavior.

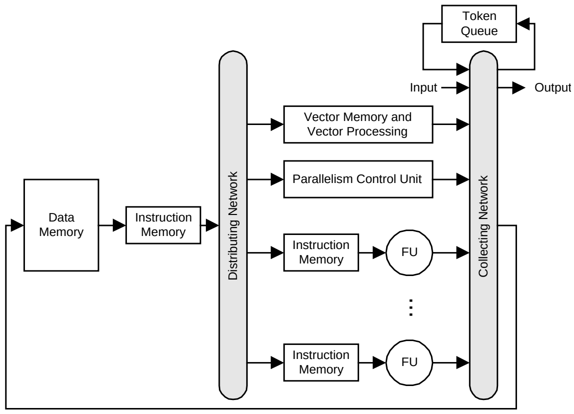

Figure 1. The Wolf architecture

Implementation of the Distributing Network

The Distributing Network (DN) deals with tokens of PA and EX types. The EX type must be sent to the VMVP (Vector Memory and Vector Processing) and the PA type must be sent to the rings (internal IM-FU). Therefore, the Distributing Network (named SED) may be separated into two blocks:

- One Switching Element (SE) similar to the Arbitrator;
- A Distributor (D) to detect which ring is available to receive the PA token.

The SE identifies the token type and sends it to the correspondent output.

Evaluation of the collecting and distributing networks

To compare network costs a simplified cost model has been used (Bhuyan, 1983). This model takes into account that a crossbar switch **mxn** (m inputs and n outputs) has a cost of **m*n** units. This corresponds to the number of interconnection crosspoints shown in Figure 2. The cost of an interconnection network with **r** stages and **K mxn** switches per stage is the sum of all crosspoints in each stage: **r*K*m*n**.

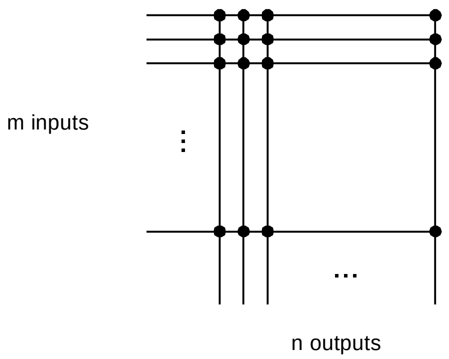

Figure 2. Cost model

Evaluation of system behavior after the insertion of these networks is based on the results obtained with four benchmarks, which explore distinct characteristics of the machine:

- Increment (Inc.): the benchmark is the simplest of all benchmarks and is strictly sequential. It calculates the increment of all terms from 1 to n (1+2+...+n).
- Ackermann (Ack.): the Ackermann function is defined by Ack(0,n) = n+1; Ack(m,0) = Ack(m-1,1); Ack(m,n) = Ack(m-1,Ack(m,n-1)); and generates a recursive tree of processes.
- Binary Integration (B.I.): the benchmark explores recursive calls to a process and calculates the area below a curve using the trapeze rule.
- Matrix Multiplication (M.M.): the algorithm creates and multiplies a quadratic matrix. It explores the characteristics of the VMVP.

Results obtained with simulations and discussion

Table 1 shows the results obtained for the benchmarks in all possible combinations of DN-CN pairs. Execution time (Time), cost of networks (Cost) and average occupation of TQ (Occ) were analyzed. Execution time and average occupation of TQ indicate network efficiency, since they either improve or degrade the machine performance. The first column indicates all possible combinations of interconnection networks used in the simulations. Increase in Time indicates that the network introduces undesirable delays in token transmissions. Increase in Occ reflects increase in memory utilization due to network inefficiencies.

Table 1. Results for the benchmarks

Networks		Inc		Ack		BI		MM	
	Cost	Time	Occ	Time	Occ	Time	Occ	Time	Occ
Cross1-Crossbar	19	1502	0.07	218202	0.18	136001	587.29	89002	1209.24
Cross1-Cross2	40	1178	0.09	137403	0.12	57000	802.67	69002	1409.87
Cross1-Delta	19	1716	0	260604	0.09	116001	438.26	84539	1209.91
Cross1-GSN	18	1715	0	273400	0	180000	534.93	82008	1361.09
SED-Crossbar	20	1715	0.07	246802	0.45	136000	638.07	93004	1123.95
SED-Cross2	41	1390	0.17	154802	0.23	56004	831.91	70002	1725.13
SED-Delta	20	1925	0	275605	0	114004	490.25	87002	1151.94
SED-GSN	19	1936	0.18	284804	0.01	184003	590.21	84011	1176.04

Table 2 shows each interconnection network cost in ascendant order. Cross1-GSN network pair is the least expensive of all network pairs and SED-Cross2 network pair is the most expensive.

Table 2. Interconnection network costs

Networks	Cost
Cross1-GSN	1
Cross1-Crossbar	2
Cross1-Delta	2
SED-GSN	2
SED-Crossbar	3
SED-Delta	3
Cross1-Cross2	4
SED-Cross2	5

Table 3 shows the results for Increment (Time Ascendant). First column shows Time classification in ascendant order. Cross1-Cross2 network pair is better than any other pair. SED-Delta and SED-GSN network pairs are the worst combinations. There is no apparent relationship between the two other parameters (Occ and Cost) and Time.

Table 3. Increment (time ascendant)

Networks	Inc		
	Time	Occ	Cost
Cross1-Cross2	1	3	4
SED-Cross2	2	4	5
Cross1-Crossbar	3	2	2
Cross1-Delta	4	1	2
Cross1-GSN	4	1	1
SED-Crossbar	4	2	3
SED-Delta	5	1	3
SED-GSN	5	5	2

Table 4 shows the results for Increment (Occ Ascendant). First column shows Occ classification in ascendant order. Cross1-Delta, Cross1-GSN and SED-Delta network pairs are better than any other pairs. SED-GSN network pair is the worst combination.

Table 4. Increment (occupation ascendant)

Networks	Inc		
	Occ	Time	Cost
Cross1-Delta	1	4	2
Cross1-GSN	1	4	1
SED-Delta	1	5	3
Cross1-Crossbar	2	3	2
SED-Crossbar	2	4	3
Cross1-Cross2	3	1	4
SED-Cross2	4	2	5
SED-GSN	5	5	2

Table 5 shows the results for Ackermann (Time Ascendant). First column shows Time classification in ascendant order. Cross1-Cross2 network pair is better than any other pair. SED-GSN network pair is the worst combination. There is no apparent relationship between the two other parameters (Occ and Cost) and Time.

Table 6 shows the results for Ackermann (Occ Ascendant). First column shows Occ classification in ascendant order. Cross1-GSN and SED-Delta network pairs are better than any other pair. SED-Crossbar network pair is the worst combination.

Table 7 shows the results for Binary Integration (Time Ascendant). First column shows Time classification in ascendant order. SED-Cross2 network pair is better than any other pair. SED-GSN network pair is the worst combination. There is no apparent relationship among the two other parameters (Occ and Cost) and Time.

Table 5. Ackermann (time ascendant)

Networks	Ack		
	Time	Occ.	Cost
Cross1-Cross2	1	4	4
SED-Cross2	2	6	5
Cross1-Crossbar	3	5	2
SED-Crossbar	4	7	3
Cross1-Delta	5	3	2
Cross1-GSN	6	1	1
SED-Delta	6	1	3
SED-GSN	7	2	2

Table 6. Ackermann (occupation ascendant)

Networks	Ack		
	Occ	Time	Cost
Cross1-GSN	1	6	1
SED-Delta	1	6	3
SED-GSN	2	7	2
Cross1-Delta	3	5	2
Cross1-Cross2	4	1	4
Cross1-Crossbar	5	3	2
SED-Cross2	6	2	5
SED-Crossbar	7	4	3

Table 8 shows the results for Binary Integration (Occ Ascendant). First column shows Occ classification in ascendant order. Cross1-Delta network pair is better than any other pair. SED-Cross2 network pair is the worst combination.

Table 7. Binary integration (time ascendant)

Networks	BI		
	Time	Occ	Cost
SED-Cross2	1	7	5
Cross1-Cross2	2	6	4
SED-Delta	3	2	3
Cross1-Delta	4	1	2
Cross1-Crossbar	5	4	2
SED-Crossbar	5	5	3
Cross1-GSN	6	3	1
SED-GSN	7	4	2

Table 8. Binary integration (occupation ascendant)

Networks	BI		
	Occ	Time	Cost
Cross1-Delta	1	4	2
SED-Delta	2	3	3
Cross1-GSN	3	6	1
Cross1-Crossbar	4	5	2
SED-GSN	4	7	2
SED-Crossbar	5	5	3
Cross1-Cross2	6	2	4
SED-Cross2	7	1	5

Table 9 shows the results for Matrix Multiplication (Time Ascendant). First column shows Time classification in ascendant order. Cross1-Cross2 network pair is better than any other pair. SED-Crossbar network pair is the worst combination. There is no apparent relationship between the two other parameters (Occ and Cost) and Time.

Table 10 shows the results for Matrix Multiplication (Occ Ascendant). First column shows Occ classification in ascendant order. SED-Crossbar network pair is better than any other pair. SED-Cross2 network pair is the worst combination.

Table 9. Matrix multiplication (time ascendant)

Networks	MM		
	Time	Occ	Cost
Cross1-Cross2	1	6	4
SED-Cross2	2	7	5
Cross1-GSN	3	5	1
Cross1-Delta	4	4	2
SED-GSN	4	3	2
SED-Delta	5	2	3
Cross1-Crossbar	6	4	2
SED-Crossbar	7	1	3

The results obtained with simulations show that there is no satisfactory interconnection network pair that matches Time, Cost and Occ simultaneously for all benchmarks. The best result obtained with Increment for Time was Cross1-Cross2 interconnection network pair and the worst SED-GSN and SED-Delta. A different result was obtained with Cost and Occ. Similar results were obtained for all other benchmarks, with variation only of the interconnection network pair.

Table 10. Matrix multiplication (occupation ascendant)

Networks	MM		
	Occ	Time	Cost
SED-Crossbar	1	7	3
SED-Delta	2	5	3
SED-GSN	3	4	2
Cross1-Crossbar	4	6	2
Cross1-Delta	4	4	2
Cross1-GSN	5	3	1
Cross1-Cross2	6	1	4
SED-Cross2	7	2	5

The identification of the best interconnection network pair in all situations is a complex exercise leading to the use of some analytic method. One of the aims of this work was to elaborate a ranking of interconnection network pairs and relate them to each benchmark, taking into account the cost of the network, time to execute a benchmark and the performance of each network in the architecture (TQ occupation).

Simulations performed led to non-conclusive results concerning the most cost-effective interconnection network for the Wolf architecture. A new approach based on optical interconnection networks has been adopted in order to investigate whether the electronic approach has something to do with these non-conclusive results.

References

Bhuyan, L.N.; Agrawal, D.P. Design and performance of generalized interconnection Networks. *IEEE Transact. Comput.*, *32*(12):1081-1090, 1983.

Bhuyan, L.N.; Yang, Q.; Agrawal, D.P. Performance of multiprocessor interconnection networks. *IEEE Computer*, *22*(2):25-37, 1989.

Cavenaghi, M.A.; Spolon, R.; Martins, J.E.M.P; Domingues, S.G.; Garcia Neto, A. Using PVM in the simulation of a hybrid dataflow architecture. In: *Proceedings of the Third European PVM Conference*. Munich, Germany: Technische Universitaet Müenchen, 1996. p.343-346.

Cavenaghi, M.A.; Travieso, G.; Garcia Neto, A. A simple mechanism to deal with sequential code in dataflow architectures. In: _______. *Proceedings of International Conference on High Performance Computing (HiPC)*. Madras, India: [s.n.], 1998. p. 188-193.

Cavenaghi, M.A.; Martini, J.A.; Garcia Neto, A. Proposal to solve workload in the simulator of the wolf architecture. In: *International Conference/Exhibition on High Performance Computing in Asia-Pacific (HPC-Asia)*, 4.ed., 2000, Beijing. *Proceedings...* Beijing, China: [s.n.], 2000. p.315-318.

Hwang, K.; Briggs, F.A. *Computer architecture and parallel processing*. USA: McGraw-Hill, 1984. (McGraw-Hill Series in Computer Organization and Architecture).

Ibbett, R.N.; Topham, N.P. *Architecture of high performance computers*. London: Macmillan Education, 1989. (Macmillan Computer Science Series, v.2).

Martini, J.A.; Garcia Neto, A.; Cavenaghi, M.A. Influência das redes de interconexão sobre os tempos de execução na arquitetura Wolf. *Acta Scientiarum*, *20*(4):477-481, 1998.

Received on April 24, 2000.

Accepted on November 28, 2000.