Dr. Sudarson Jena*
Prof. (Dr.) C R Tripathy**

An Artificial Neural Network Approach to Performability of Multiprocessor Interconnection Networks

Abstract

Performability of an interconnection system depends upon the failure characteristics of its components. There is the need of a technique to predict the performability of a multiprocessing network from the existing available input/output data. In an interconnection network, the processors are connected with each other through links. There may be imperfection at the links or at the nodes, which affects the system performance. Hence a general and flexible prediction model needs to be developed to compute the reliability and performance of the multiprocessor interconnection networks. In this paper we presents an artificial neural network model based on principle of back propagation algorithm to compute the performability of crossed-cube and star graph multiprocessor interconnection networks.

Introduction:

An important factor of a multiprocessor interconnection network is the system topology. The system topology defines the interprocessor communication architecture [1,5, 9]. Therefore the suitability of a multiprocessor for various scientific and engineering applications can be assessed by careful analyses of the performability indices, viz; performance and reliability characteristics of its interconnection structure. Performance evaluation aspects of these networks have been investigated by several researchers [8,10]. While evaluating the performance of a multiprocessor, generally the network is assumed to be fault-free. However, in a practical situation, the components may fail at random for various reasons. Therefore, there is a utter need to investigate its performability incorporating the failure characteristics of the nodes and links in detail.

The design of reliable multiprocessor interconnection systems is one of the most important issues facing system engineers today. In large multiprocessor systems, the probability of increase in faults and errors commensurate with their increased complexity. It is desirable to be able to model the performance and reliability of such systems as their functionality degrades due to failed components or recovery actions.

*Asst. Professor, Department of Computer Science, HDF School of Management, Bhubaneswar, Odisha, India, E-mail:-sjena2k5@rediffmail.com. **Principal,University College of Engineering , Burla, Orissa, India

Many researchers have attempted evaluation of performability of multiprocessor interconnection system in the past [7,10]. However, as the system size increases the reliability/performance prediction using the above approaches becomes extremely difficult and some times becomes intractable.

There are various approaches to compute the reliability of a system. They include SDP(sum-of-disjoint product) techniques[8], Markov modeling[9] and Combinatorial method[7].The SDP technique utilizes Boolean concepts to convert a sum-of-products expression for minpaths or mincuts into an equivalent SDP expression of exclusive and mutually disjoint terms. In this form, a logical success (failure) state of a component x can be replaced by component reliability (unreliability), and the Boolean sum (product) can be replaced by the arithmetic sum (product) to obtain the reliability. But disjointing for large network becomes cumbersome.

In Markov modeling approach [9] the system probability of success is expressed in terms of the probabilities of success of its components. A Markov discrete time model consists of n states. State 1 represents the initial state when the system is completely operational (the probability of failure = 0), while state n represents the final state. Every two states are connected to the upper layer implying the upgrade (restoration or repair) or degrade (partial or total failure) of hardware. If there is no connection between two states in a forward, backward or both directions, then simply that particular weight is equated to zero. Simulating any interconnection network through this model requires solution of differential equations corresponding to each state and it is difficult to solve them.

In the combinatorial approach [8], the reliability of different interconnection networks is calculated in different ways. This method is not flexible and requires very complex mathematical calculation. There is the need of a technique to predict the performability of a multiprocessing network from the existing available data.

Recent advances in artificial neural networks show that they can be used in applications that involve predictions. The neural network models have a significant advantage over analytical models because they require only failure history as input and no assumptions using that input. This model automatically develops its own internal model of the failure process and predicts future failures. It doesn't require a prior knowledge of a mathematical function that maps input to the output value and so complexity of the system does not create any problem.

Artificial Neural Networks [3,4] are computational metaphor inspired by studies of the brain and nervous system in biological organisms. They are highly idealized mathematical models of how we understand the essence of these simple neuron systems. The advantages of using neural network for computation of the performability is that (i) it is flexible i.e. this method can be used for any type of multiprocessor interconnection network; (ii) it doesn't require a prior knowledge of a mathematical function that maps input to the output value and so complexity of the system does not create any problem.

In this paper, the back propagation algorithm has been used to compute the performability of crossed cube and star graph multiprocessor interconnection networks.

The rest of the paper is organized as follows. Section 2 of this paper presents a general description of the multiprocessor interconnection network. Performability analysis of proposed approaches are present in section 3 & 4 of this paper. Numerical results of both crossed cube and star graph networks are presented in section 5 of this paper. Concluding remarks are presented in section 6.

Multiprocessor Interconnection Networks

Based on interconnection parallel-distributed computer architecture is broadly classified into two categories, namely: *tightly coupled* and *loosely coupled* machines[1]. In a tightly coupled system, all processors in addition to their own local memory have access

to a global memory and interprocessor communication is achieved through the shared memory. On the other hand, in a loosely coupled system each processor has its own private memory and message/packet switching is used for communication among processors.

Loosely coupled multiprocessors are now-a-days widely used in various fields. The prominent candidates of this category include Crossed cube and Star graph. Performability evaluation of these networks needs to be carried out through some techniques so that one can make a choice between various alternative topologies. Multiprocessing is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. Concurrency implies parallelism, simultaneity, and pipelining. There are different types of interconnection networks. Which are used in multiprocessing system some of them are briefly outlined below.

Crossed cube Topology

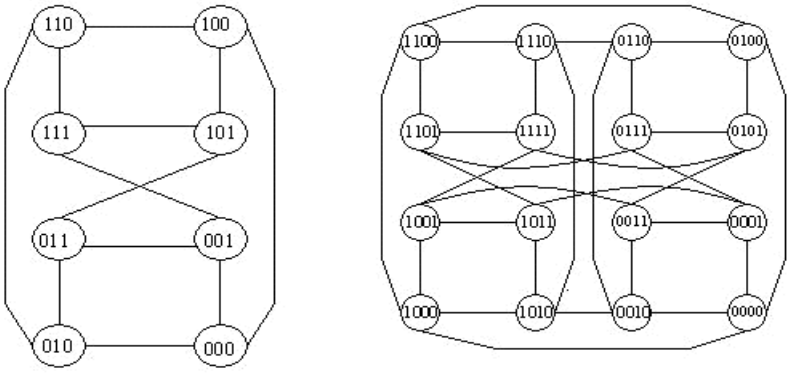
An n-dimensional "crossed" cube denoted by CQ_n is a parallel interconnection network which has the same node and link complexity as the hypercube [11]; and has most of its desirable properties including regularity, recursive structure, partionability and ability to simulate other architectures. The diameter of the CQ_n network is almost half of that of the hypercube or more precisely that is [(n+1)/2].

The n-dimensional crossed cube is the labeled graph defined inductively as follows [5].CQ_1 is K_2 the complete vertices with labels 0 & 1 , for n>1, CQn contains CQ^0_{n-1} and CQ^1_{n-1}joined according to the following rule :

the vertex $u=0u_{n-2}$ ……u_0 from CQ0n-1 and the vertex $v= 1v_{n-2}$ ……v_0 from CQ^1_{n-1} are adjacent in CQ_n if and only if (a)$u_{n-2} = v_{n-2}$ if n is even , and

(b) for $0£ i £ [(n-1)/2]$, $u_{2i+1}\ u_{2i} \sim v_{2i+1}\ v_{2i}$

[Two binary strings $x = x_1x_0$ and $y = y_1y_0$ are pair-related, denoted by x~y, if and only if (x, y) Î{(00,00), (10,10),(01,11),(11,01)} ; if x & y are not pair related, we write x c y .]



CQ_3 CQ_4

Fig. 1 CQ_n for n = 3,4

It follows from the above definition that every vertex in CQ_n with a leading 0 bit has exactly one neighbour with a leading 1 bit and vice versa. An n dimensional crossed cube contains a variety of subgroup, which is isomorphic copy of crossed cubes of lower dimension.

Star graph Topology

Star graph has more complex structure rather than binary n-cube. The n-dimensional star graph S_n is an edge and node symmetric graph containing n! nodes and *(n-1)n! " 2* links. Each node is labeled by a distinct permutation set of integers{1——n} .Two joints are linked with a link i if and only if the label of the one can be obtained from the label of the other by swapping the first digit

layer of the units is the input layer-the only units in the network that receive external input. The layer above is the hidden layer, in which the processing units are interconnected to layers above and below. The top layer is the output layer. A detailed discussion on the back propagation can be found in [3].

Proposed Approach for computation of the performability using Artificial Neural Network.

The proposed approach for computation of the performability using artificial neural network consists of the following steps.

Given a set of input output vector pairs

Input = failure rate x time, Output = Performability

Compute: The performability for a new value of input

l_n = Processor failure rate

l_i = Processor failure rate

C = Coverage factor

n = Dimension of the network

N = No. of Processors

Step 1

Let A= no. of units in input layer, as determined by the length of the training input vectors

B = no. of units in the hidden layer

C = no. of units in the output layer

Referring to the above figure, the input and hidden layers each have an extra unit used for thresholding ; therefore, the units in these layers will sometimes be indexed by the ranges (0.....A) and (0.....B)

Activation levels of the units in input layers = x_i

Activation levels of the units in hidden layers = h_j Activation levels of the units in output layers = o_j Weights connecting the input layer and hidden layer are W_{ijj}

Where i indexes the input units and j indexes the hidden units.

W2ij denotes weight connecting the hidden layer and output layer

Where i indexes the hidden units and j indexes the output units.

Step 2:

Initialize the weights in the networks. Each weight should be set randomly to a number between –0.1 and 0.1 .

W_{ijj} = random (-0.1 , 0.1) for all i = 0 ....A , j = 1.....B

W_{2ij} = random (-0.1 ,. 0.1) for all i= 0....B, j=1.....C

Step 3:

Initialize the activations of the thresholding units.

The value of these thresholding units should never change.

x[0] = 1.0 h[0] = 1.0

Step 4:

Determine the number of input-output training pairs (= T). Choose an input-output pair. Suppose the input is x_in and the corresponding output is y_in.

Step 5:

Find the binary equivalent of the values x_in and y_in . Now the input vector corresponding to x_in is bin_x[] and the out put vector corresponding to y_in is bin_y[].

Step 6:

Propagate the activation from input layer to hidden layer using activation function

$$h_j = 1/ (1 + \exp [-\text{å}w_{ijj} x_i]) \quad (1)$$

i=0.....A ,for all j = 1.....B

w_{ijj} is the threshold weight for hidden unit j and x[0] is always 1.0.

Step 7:

Propagate the activations from the units in the hidden layer to the units in the output layer.

$$_{Oj} = 1/ (1 + \exp [-\text{å}w_{2ij} h_i]) \quad (2)$$

i=0.....B

for all j = 1.....C

The threshold weight w_{2ij} for output unit j plays a role in the weight summation ; h[0] is always 1.0

Step 8:

Compute the errors d_{2j} of the units in the output layer , errors are based on the network 's actual output (o_j) and the target output (y_j).

$$d_{2j} = o_j (1- o_j) (y_j – o_j) , \quad (3)$$

for all j = 1,2,.....,C

Step 9:

Compute the errors of the units in the hidden layer denoted by d_{ij} .

$$d_{ij} = h_j (1-h_j) \text{å}d_{2j} w_{2ij} ,$$

for all j = 1.....B (4)

Step 10:

Compute the error term given by

Err = 0.5 ($\sum d_{2j}$ * d_{2j}) ,

for all j =1.....C (5))

Step 11:

Adjust the weights between the hidden layer and the output layer—

(Assume the learning rate = 0.35)

Dw_{2ij} = h d_{2j} h_i , for all i=0.....B (6)

Step 12:

Adjust the weights between the input layer and the hidden layer –

Dw_{iij} = h d_{ij} x_i for all j = 1.....C (7)

Continue adjustment of weights until the error term "err" is less than or equal to 0.0001

Step 13:

Go to step 4 and repeat. When all the input output pairs have been presented to network, training of the network is completed.

Step 14:

Once the network is trained, it is able to predict the value of the output (performabilityy) for a new value of the the input (failure rate x time). When an input is given, first it is converted to the binary number. Then the output values of each unit in the output layer is approximated to a binary value. So the matrix corresponding to these binary values are converted to a floating point number which is the required predicted value. This algorithm is written for one input layer, One hidden layer and one output layer. However it can be used for more than one hidden layer with a little modification.

Results and Discussions

The paper considers the performability evaluation of some important tightly coupled multiprocessor interconnection networks viz; Crossed cube and Star graph using the models proposed in previous section. The proposed algorithm has been applied to find the reliability of crossed-cube network and star graph based network using back propagation method. Analytical results for higher dimensional star graph and crossed cubes have been presented in Table-1 & 2.

Table-1 Training data for Reliability of Star network

(λ_p=0.0001, λ_l=0.00001)

λ_pt	Dim. (n =5)	Dim. (n=6)	Dim.(n=7)
0	1.000000	1.000000	1.000000
0.04	0.998500	0.999224	0.999555
0.08	0.984265	0.986957	0.988413
0.12	0.946862	0.946238	0.943372
0.16	0.886059	0.873095	0.856263
0.20	0.808118	0.776325	0.739711
0.24	0.721075	0.668931	0.613194
0.28	0.632035	0.562115	0.492400
0.32	0.546193	0.463438	0.386214
0.36	0.466801	0.376576	0.297849
0.40	0.395538	0.302749	0.226985

From the tables it is shown that the value of $l_n t$ is increased then performability of star graph and crossed cube networks drops down quickly to Zero level. The observations for the star and crossed cube networks using the back propagation algorithm were found to have a tolerable error.

Conclusion

The use of artificial neural network for performability prediction for different multiprocessor interconnection networks has been presented. The results with actual value of the data suggest that the proposed neural network model predict the performability very near to exact value. The major advantages of the proposed neural network based approach are that the user need not know much about the underlying failure process. Further, It is flexible and can be used for any type of multiprocessor interconnection network. From this study, it is concluded that artificial neural network approaches are quite powerful tools for effectively predicting performability of multiprocessor systems.

References

1 Hwing K, Briggs F.A, *Computer Architecture and Parallel Processing*, Ny McGraw Hill Book company,1985.

2. Trivedi K. S, *Probability and Statistics with Reliability, Queueing and Computer Science Applications*, Englewood Cliffs, Prentice-Hall, 1982

3 Troudet T. P, Walters S. M, Neural network architecture for crossbar switch control. *IEEE Trans. Circuits and Systems,* 1991, 42-56,

4 N Karunanithi et al, Using neural Networks in reliability Prediction, *IEEE Trans. Software Engg,* 1992. vol (4), pp.53-58,.

5 Efe K., The crossed cube architecture for parallel Computation ,*IEEE Trans. Parallel and Dist. systems* , 1992., Vol 3(5), PP. 513-524

6 Day K et al, A Comparative study of topological properties of hypercube and star graphs, *IEEE Trans. Parallel and Dist. Systems*, 1994, Vol5(1), PP.31-38,.

7 Chang Yeimkuan et al ,A Combinatorial analysis of subcube reliability in hypercubes, *IEEE Trans. computers* , 1995., Vol 44(7), 952-956,

8 Tripathy C. R,. Mohapatra R.N. Misra R. B, Reliability analysis of Hypercube Multicomputers, *Microelectronics and Reliability: An International Journal*, 1997., Vol37(6), PP.885-891,

9 Mishra K.B, *Reliabilty analysis and prediction, A. methodology oriented* ,Elsevier science Amsterdam, 1992.

10.Saad Y, .Schultz M.H,Topological properties of Hypercubes, *IEEE Trans. computers*, 1998, Vol(37),PP. 86-88.