IBM Technical Disclosure Bulletin Vol. 28 No. 7 December 1985

PARALLEL COMPUTERS COUPLING A PERMUTATION NETWORK

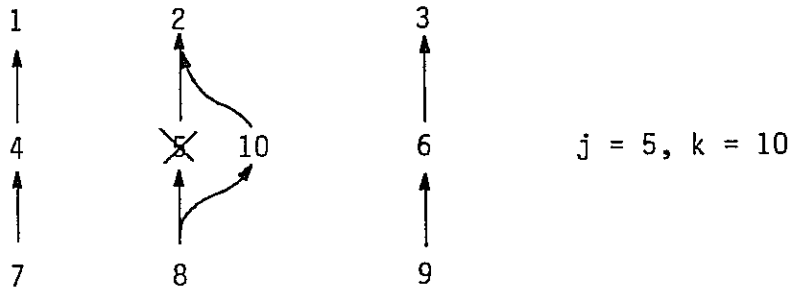

FIG. 1

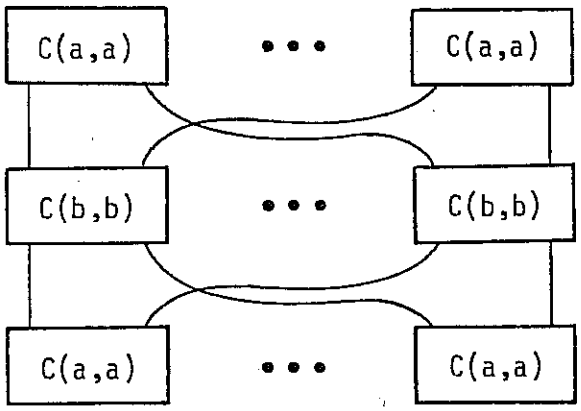

FIG. 2

This invention relates to the reliability of parallel computers interconnected by a permutation network. In this regard, an n-permutation network is a network with n inputs $I_1,\ldots,I_n$ and n outputs $O_1,\ldots,O_n$ such that for all permutations p, one can send data from I_j to $O_{p(j)}$ simultaneously for all j. Connecting n processors $P_1,\ldots,P_n$ to a permutation network, such that for all j processors P_j can send data to I_j and receive data from O_j, has been shown to be a good way to build massively parallel computers. If n is large (say, 1000), then processor and network failure become problems.

The inventive solution contemplates:

1. Processor failure. Under normal operation not all processors are used. Whenever a processor, say P_j, fails during a computation, one of the unused processors, say P_k, is "switched in for P_j" in the following sense: for each permutation p that is used in the computation set p(x) = k if previously p(x) was j. Set p(k) to the old value of p(j). See Fig. 1. The computation is restarted from the last checkpoint, which must be provided by the user.

2. Network construction and reliability. For all i and j, let C(i,j) denote an ixj crossbar switch. Let n = axb. It is well known that an n-permutation network can be built from 2b copies of C(a,a) and a copies of C(b,b), as indicated in Fig. 2: for each i the ith copy of C(b,b) is connected to the ith output of each of the copies of C(a,a) in the upper row and the ith input of each of the copies of C(a,a) in the lower row. It is also known that even if one of the copies of C(a,a) can only realize one arbitrary fixed permutation, the remaining network is still a permutation network. Thus, adding little extra circuitry which duplicates for each of the C(a,a)'s the data path corresponding to the identity permutation among a number, say, will protect one against one failure among the C(a,a)'s.

Here is how single failures among the C(b,b)'s are managed. Replace the C(a,a)'s in the first row by C(a,a+1)'s. Use a+1 C(b,b)'s in the middle row, but under normal operation use only a of them. Replace the C(a,a)'s in the bottom row by C(a+1,a)'s. Use the same interconnection as described above. Now, if one of the C(b,b)'s fails, route all paths that went through it through the spare a+1'st copy.

If one has, say, b-bits wide data paths through the communication network and one realizes this as b-permutation networks, each 1-bit wide, and if one realizes the crossbars as chips, then one can continue operations with one chip failure in each of the b 1-bit wide permutation networks.

All the above changes of the network configuration can be performed by software, and it is not necessary to shut the machine down.