

OBJECTIVE OF PRESENT WORK

Parallel Processing has had a major impact on the development of computer science. There is extensive literature on parallel architectures, parallel programming languages and parallel algorithms which has achieved insight into the capabilities and limitations of parallel machines. However, the diversity and complexity of parallel architectures have created a fundamental challenge for programmers, viz. the efficient mapping on algorithms to the parallel environment. Although efficient practical algorithms for problems on particular parallel architectures are already known, still, the active area of current research is the development of a general methodology for the design of parallel algorithms that are practical and efficient on a wide variety of architectures.

This thesis is a further step in that direction. We propose a paradigm for structuring and designing programs for parallel computing. This paradigm is described informally, and advocated by means of a collection of case studies on parallel architecture which are massive.

The inspirational motivation for our paradigm is the drive to initiate a uniform problem-solving method for eliciting the underlying algebraic structure. The existence of some algebraic structure is particularly important in the context of parallel computing, insofar as parallel computers typically perform more efficiently on highly structured computations than they do on unstructured computations.

In this thesis we also identified and then discussed the issues faced in relation to the efficient operations of the multi-dimensional arrays. It was found that the most of the proposed methods do not perform well for extended form of tensors although these methods show good performance when applied to two-dimensional arrays. We discussed the flaws of the traditional matrix representation (TMR) and then proposed the Extended Karnaugh Map Representation (EKMR) as a new scheme which ruled out the drawbacks of the TMR scheme. EKMR is based on the Karnaugh Map. The basic concept of the EKMR technique is to represent the multi-dimensional array in to the form of a set of two-dimensional arrays. Thus, the extended Karnaugh map representation made it easier to design the efficient data parallel algorithms for multi-dimensional arrays having more than two dimensions. We analyzed the data parallel algorithms for multi-dimensional matrix multiplication using the Karnaugh map that is EKMR

and concluded that EKMR is better than TMR in all aspects. The concepts given by O' Boyle to design the loop re-permutation have been applied in this report to design the data parallel algorithms for multi-dimensional array multiplication operation using the EKMR scheme [135, 136]. This report focused on the application of the EKMR on the dense multi-dimensional array, however we have discussed that EKMR is equally effective in case of sparse multi-dimensional arrays.

With the help of the parallel algorithms for multi-dimensional matrix multiplication operation using the Karnaugh map, it was proved that the cost of computing index of elements with EKMR scheme is less than that of TMR scheme and the number of lines cached which the dense array operations have accessed for EKMR scheme is less than that of TMR scheme. These were the flaws of the TMR scheme which previously caused the inefficient performance when the dimensions of the arrays exceeded the value of 2. Thanks to the EKMR scheme which optimized the performance even to the n th dimension of the tensors. This thesis described techniques for the design of parallel programs that solve well structured problems with inherent symmetry.

Part first demonstrated the reduction of such problems to generalized matrix multiplication by a group-equivariant matrix. Fast techniques for this multiplication were described, including factorization, orbit decomposition, and Fourier transforms over finite groups. Our algorithms entailed interaction between two symmetry groups: one arising at the software level from the problem's symmetry and the other arising at the hardware level from the processors' communication network.

Part second illustrated the applicability of our symmetry-exploitation techniques by presenting a series of case studies of the design and implementation of parallel programs.

(i) A parallel program that solved matrix multiplication by factorization of an associated dihedral group-equivariant matrix was described. This code ran faster than previous serial programs and discovered a number of results in its domain.

(ii) Parallel algorithms for Fourier transforms for finite groups were developed and preliminary parallel implementations for group transforms of dihedral and of symmetric groups were described. Applications in learning, vision, pattern recognition and statistics were proposed.

(iii) Parallel implementations solving several computational science problems were described, including the direct N-body problem, convolutions arising from molecular biology, and some communication primitives such as broadcast and reduce. Some of our implementations ran orders of magnitude faster than previous techniques, and were used in the investigation of various physical phenomena.