Transactions of the **13th International Conference on Structural Mechanics in Reactor Technology** (SMiRT 13), Escola de Engenharia - Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, August 13-18, 1995

Role of Barc parallel processing system in structural analysis of nuclear components

Prasad, P.S., Dutta, B.K., Kushwaha, H.S., Mahajan, S.C., Kakodkar, A.
Bhabha Atomic Research Centre, Reactor Desing & Development Group, Bombay, India

ABSTRACT: Computational structural analysis of nuclear reactor components requires high speed computing systems. Parallel processing systems, such as ANUPAM developed at Bhabha Atomic Research Centre, have helped immensely in meeting this requirement. The implementation of structural integrity analysis codes on ANUPAM is illustrated here by commissioning two in house codes TABS and FAIR. The parallelisation strategies and the efficiency of ANUPAM are highlighted.

1 INTRODUCTION

Demonstrating the safety of nuclear reactors under both normal and accident conditions of reactors involves a rigorous structural analysis of various reactor in core and out of core components. This analysis is often carried out with the help of numerical techniques, such as the finite element method. The computational requirements in using this method for various types of analyses, such as nonlinear, dynamic analysis etc, necessitate high speed computational power. The analysis of large size problems, which require high memory and computational power, and the non availability of high speed sequential computers capable enough to solve such problems, have paved the way for the emergence of parallel processing systems.

The advantages of parallel processing systems and the restricted availability of high speed sequential computers have forced indigenous development of parallel processing systems. Such an effort in the parent organisation of the authors has yielded ANUPAM series of parallel processing systems. The advantages of ANUPAM systems in reducing computational time during structural integrity analysis of nuclear reactor core components is studied by commissioning two in house nonlinear structural analysis codes TABS (Dutta et al, 1995) and FAIR (Swami Prasad et al, 1994).

2 ANUPAM PARALLEL PROCESSING SYSTEM

The ANUPAM parallel processing system developed at the computer division of Bhabha Atomic Research centre is a loosely coupled MIMD (Multiple Instruction Multiple Data) type of computer system that

makes use of RISC (Reduced Instruction Set Computer) based CPUs. Each processor of a parallel processing system is colloquially called as node. The architecture of the ANUPAM is based on 2-D mesh topology for intra cluster communication and bus topology for inter cluster communication. A cluster is a collection of maximum of 8 nodes. The system is highly scalable and field upgradable from 2 nodes to 64 nodes. It uses a standard 32 bit high performance Multibus II back plane bus with message passing co-processor VLSI chip to transfer the message within the cluster and the clusters are inter linked by two 16 bit wide SCSI buses one in x-direction and the other in y-direction, thus forming the 2-D mesh topology. An 8 node version of the ANUPAM gave a Linpack performance of 52 MFlops . More details of this system and its applications can be found in the reference of Mahajan et al (1994).

Out of all the processors that constitute the ANUPAM parallel processing system, one is called the master or host processor and all the other processors are called slave processors. The master processor can perform the input/output operations independently, whereas the slave processors necessarily have to depend on the master processor for performing their input/output operations. The master processor initiates all the slave processors, sends the necessary data to the slave processors. The master and slave processors manipulate the data and the slave processors send the results to the master processor.

The code development on the ANUPAM parallel processing system requires two different versions of the program to be written by the code developers. The first one called as master program runs on the master processor. The second one called the slave program runs on all the slave processors. Both the codes are similar algorithm wise, but for the data communication statements (available as library routines) which should be put in the proper place for data communication required among the processors.

3 SCOPE OF PARALLELISATION OF STRUCTURAL INTEGRITY ANALYSIS CODES:

The data and algorithms structure of computer codes is the decisive factor in exploiting the parallel processing system capabilities to the best advantage. The advantages of a parallel processing system can be best realised for codes which guarantee proper work load balance among different processors, large computation time and minimum data communication time. With respect to these criteria, finite element based structural analysis codes can be classified into two categories. The first category of codes require parallelisation of individual algorithms, such as, stiffness calculation, equation solution and stress calculation. In this category, the parallelisation of equation solution algorithm poses major challenges and schemes such as domain decomposition method, parallel active column solver have been in vogue. In the second category, the entire algorithm of code can be run in parallel, as in the case of harmonic analysis of structures, where each harmonic can be analysed in parallel.

A similar class of codes belonging to the second category are fuel performance analysis codes. These codes are used to demonstrate the integrity of reactor fuel pins. Modern fuel performance analysis codes perform finite element based thermomechanical calculations and

employ mechanistic models for modeling various physical and chemical phenomena occurring in the fuel during irradiation. The behaviour of fuel pin is affected by the power generated across the fuel pin length. Depending on the variation of power across the length of fuel pin, it becomes necessary to perform analysis at various axial sections of fuel pin. The analysis is distinctly similar, but for data input variation in the form of power generated at each cross section. So each axial location can be analysed independently in each processor. But the fission gas released at each section readily mixes with the fission gas released at other sections. This provides a coupling between different sections of the fuel rod. Thus, the data communication is restricted only to the exchange of fission gas release values among different processors. The computation time required between two successive exchanges of fission gas release data is very high because of mechanistic thermo mechanical and fission gas release analyses and repetitive thermal and mechanical analyses required for gap conductance convergence due to coupling between the two analyses. Thus the fuel performance analysis codes stand as the best candidates for parallel processing system applications. The number of processors required for the analysis depends on the number of axial sections of fuel rod to be analysed.

The codes TABS and FAIR belong to each of these categories respectively. The parallelisation details of these codes on ANUPAM parallel processing system are described in the following sections.

4 PARALLELISATION OF 'TABS'

The code TABS (Thermoplastic Analysis of Bending Structures) is a finite element based code for the analysis of bending structures under thermal viscoplastic loading. nine noded heterosis quadratic degenerate shell bending elements are implemented in this code. Plastic flow along the thickness of the element is considered using layered approach.

A typical finite element code, such as TABS consists of six basic modules. These modules perform computation of element stiffness matrix, assembly of element stiffness matrices to obtain global stiffness matrix, the substitution of boundary conditions in global stiffness matrix, computation of load vector, solution of simultaneous equations and computation of element stresses. Among these modules, computation of element stiffness matrix , solution of equations and computation of element stresses are computationally intensive and significant time reductions can be accrued by parallelising these modules. This is depicted in Fig.1. The stiffness and stress computations for each element can be carried out independently and hence in a parallel processing system, it is only necessary to assign group of elements to each of the processors. Unlike these two modules, the equation solution algorithms are not implicitly amenable to parallelisation. Many parallel algorithms for equation solution have appeared in the literature and research in this area is still actively pursued. In implementing the TABS code on ANUPAM, two different solvers have been tried. These are active parallel column solver (Farhat et al, 1988) and domain decomposition method (Yagawa et al, 1991). The domain decomposition method, though requires less data communication, is iterative in nature even for a linear set of equations. The number of iterations increase with number of sub

domains or processors. A proper preconditioning algorithm is required to reduce the number of iterations. On the other hand, the parallel active column solver, though is a direct equation solver, involves comparatively heavy interprocessor data communication. In fact in all the three subroutines of this algorithm, namely factorisation, forward substitution and backward substitution, data communication is required during the processing of every equation. This makes granularity of tasks very small and results in lesser efficiency.

5 PARALLELISATION OF 'FAIR'

The code FAIR (Fuel Analysis of Indian Reactors) is a fuel performance analysis code for assessing the integrity of fuel pins. The development of this code is based on modern mechanistic models. The analysis of fuel pin involves extreme nonlinear situations because of complex material behaviour and interdependence of various parameters affecting the fuel performance. The parallelisation strategy for this code is explained below.

The two versions of the code FAIR i.e, master version and slave version are developed as per the requirements of the ANUPAM parallel processing system. The master version of the code FAIR reads all the geometry and power history data of a fuel rod and sends the geometry and power history data corresponding to each axial location to each of the slave processors, where the slave version of the code FAIR is executed. The calculations corresponding to a particular time step are performed simultaneously on all the processors. These calculations include gap conductance convergence loop over thermal and mechanical routines and fission gas release calculations for each node. The fission gas release data and output data of interest such as the centre line temperature, pellet-clad gap etc, for each axial location are communicated to master processor by all the slave processors for further treatment. The fission gas mixing calculation is performed in the master processor assuming instantaneous mixing of the gases and the corresponding gas composition information is sent to all the slave processors for subsequent calculations. Thus, the entire power history of a fuel rod is analysed and the output data is sent for post processing. A schematic diagram of parallelisation strategy adopted for the code FAIR is given in Fig.2.

6 CASE STUDIES

The code TABS has been used to analyse the sagging behaviour of a PHWR (Pressurised Heavy Water Reactor) coolant channel following postulated severe accident. The analysis is carried out using active column solver. The stiffness and stress modules gave a linear speedup where as the equation solver module could yield a 50% efficiency of parallelisation.

The code FAIR has participated in an IAEA (International Atomic Energy Agency) project on Fuel modeling at extended burnups (IAEA/IND/7348, 1993). As a part of this project six cases consisting ten different fuel pins are analysed. The fuel pin corresponding to case-2 is chosen for the illustration of parallelised version of FAIR. This fuel pin is divided into five segments denoted as LHR1, LHR2, LHR3, LHR4, LHR5. One pellet each from each of these sections is

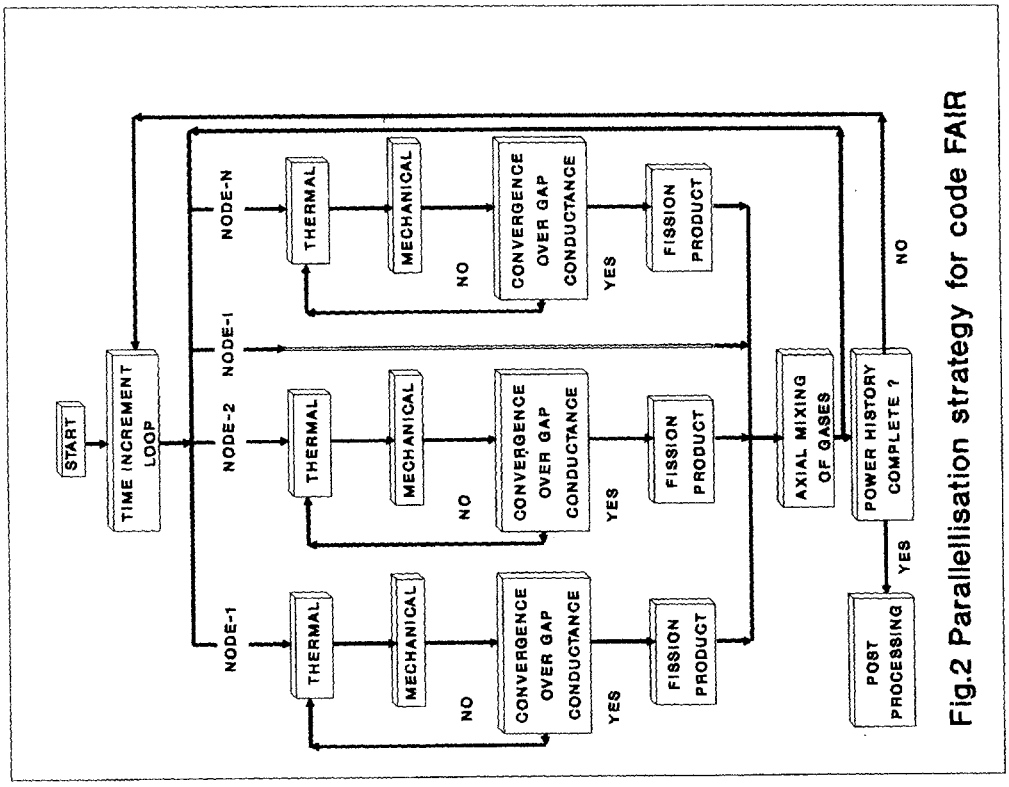

Fig.2 Parallellisation strategy for code FAIR

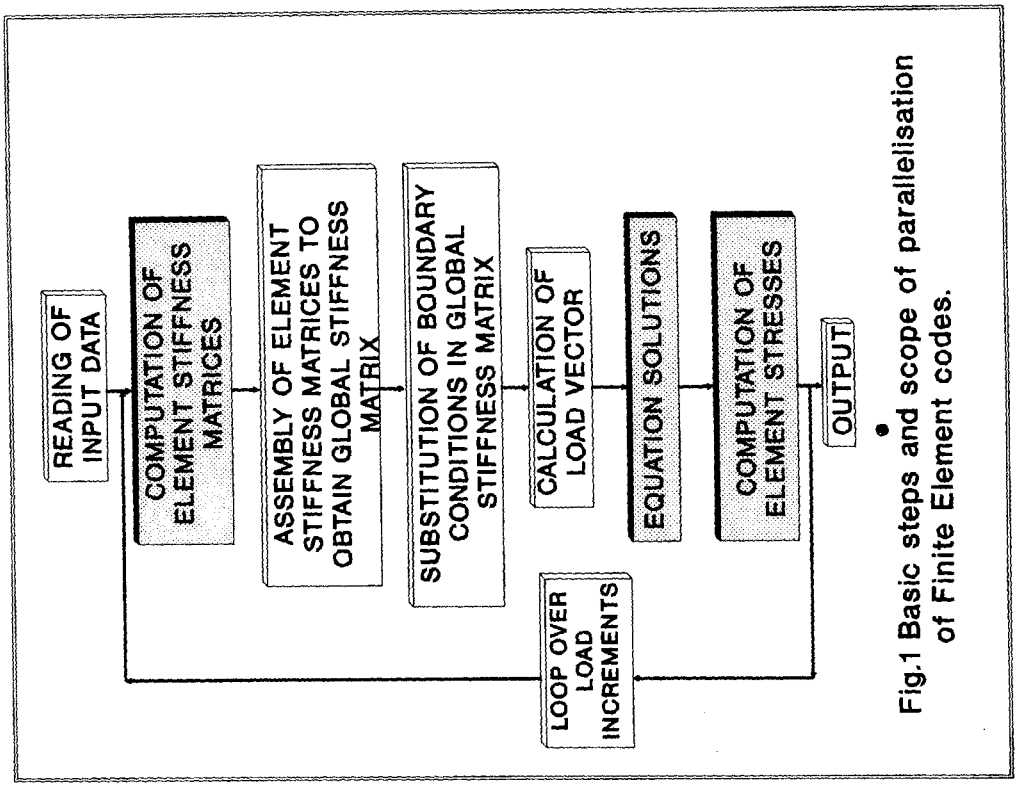

Fig.1 Basic steps and scope of parallelisation of Finite Element codes.

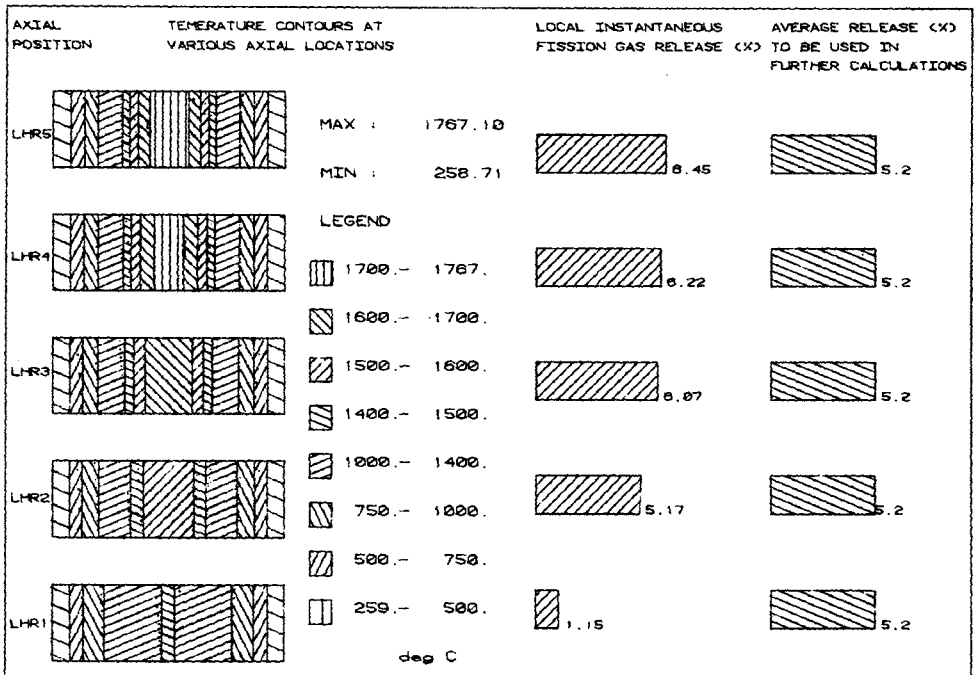

FIG.3 RESULTS COMPUTED BY CODE 'FAIR' USING 'ANUPAM' FOR CASE-2 OF IAEA CRP ON FUEL MODELING AT EXTENDED BURNUP

analysed in different processors of ANUPAM. The temperature contours in pellets at various axial locations are shown in Fig.3. The fission gas release based on these local temperatures at each axial section are also shown in this Figure. But the fission gas generated by all the pellets will mix and after mixing there will be an average fission gas value, which will be used for further calculations. Thus ANUPAM has been effectively used in this work and the speedup obtained for this analysis is almost linear.

REFERENCES

Dutta, B.K. et al, 1995. Analysis of sagging behaviour of PHWR coolant channels following postulated severe accidents, SMiRT 13, Division-F , Paper no:425.

Swami Prasad, P. et al, 1994. Fuel performance analysis code FAIR, BARC report BARC/E/013/1994.

Mahajan, S. et al, 1994. Proceedings of seminar on Super computing for scientific visualisation, Bombay India, Feb 15-17 1994.

Farhat, C. et al, 1988. A parallel active column equation solver, Computers & structures Vol.28 pp.289-304.

Yagawa, G. et al, 1991. A large scale finite element analysis using domain decomposition method on parallel computer, Computers & structures Vol.38 pp.615-625.

IAEA/IND/7348, 1993. Agreement on FUMEX cases between IAEA and India.