CONF 930372--5

Paper to be presented at the High Performance Computing Symposium, Computer Simulation Multiconference of The Society for Computer Simulation, March 29 - April 1, 1993, Arlington, VA

ANL/RA/CP--78622

DE93 008659

ADVANCES IN THERMAL HYDRAULIC AND NEUTRONIC SIMULATION FOR REACTOR ANALYSIS AND SAFETY*

A. M. Tentner, R. N. Blomquist, T. R. Canfield, T. F. Ewing, P. L. Garner, E. M. Gelbard, K. C. Gross, M. Minkoff, R. A. Valentin

Reactor Analysis Division
Argonne National Laboratory
9700 S. Cass Avenue
Argonne, IL 60439

*Work supported by the U. S. Department of Energy, Nuclear Energy Programs under Contract W-31-109-ENG-38.

ADVANCES IN THERMAL HYDRAULIC AND NEUTRONIC SIMULATION FOR REACTOR ANALYSIS AND SAFETY

A.M.Tentner, R.N.Blomquist, T.R.Canfield, T.F.Ewing, P.L.Garner, E.M.Gelbard, K.C.Gross, M.Minkoff, R.A.Valentin

Introduction.

Computer Simulation is recognized today as a critical technology, playing a major role in many areas of national interest, such as increasing the efficiency and safety of nuclear reactors, improving the efficiency of car engines or predicting weather patterns and global climate changes. In recent years, rapid changes in computing technology have made available to the computer simulation professionals more and more computing power. With the advent of massively parallel computers and distributed computing networks this trend has accelerated and is sure to continue in the foreseeable future. The development of large scale computational applications needed for the solution of practical problems is beginning, relatively slowly, to take advantage of the available parallel computing technology. The reasons for this relatively slow change in applications include the existence of a considerable investment in software developed for single-processor mainframe computers and a yet limited understanding, among the application developers, of the potential advantages of parallel computing. Many large scale computer codes, such as those used in reactor thermal-hydraulic and neutronic reactor analysis must be reviewed and partially rewritten to take advantage of the new hardware capabilities. In the process, the capabilities of these codes can be significantly extended and their performance can improve to the point where parametric studies and real time simulation using advanced thermal-hydraulic and neutronic computer models become possible.

This paper describes several large-scale computational models developed at Argonne National Laboratory for the simulation and analysis of thermal-hydraulic and neutronic events in nuclear reactors and nuclear power plants. The impact of advanced parallel computing technologies on these computational models is emphasized.

Reactor Analysis and Advanced Computing at Argonne National Laboratory.

The early involvement of Argonne National Laboratory in the design and safety analysis of advanced metal-cooled reactors has led to the development of large-scale thermal-hydraulic and neutronic computational capabilities needed to carry out its mission. Over decades, guided by the results of many in-pile and out-of-pile experiments and using the most up-to-date numerical methods, some of these thermal-hydraulics and neutronics codes have become increasingly sophisticated and have achieved both national and international recognition. Codes such as SAS [1,2], VIM [3] or COMMIX [4] are used in many research institutions, both in the U.S. and abroad for the analysis of thermal-hydraulic and neutronic events in nuclear reactors and nuclear power plants of various designs. The current development of these codes is driven by the needs of the current programs, such as the

Integral Fast Reactor [5]. This development is strongly influenced by the current trends in advanced computing, in particular the advent of distributed computing and multi-processor computers. The improved performance allowed by these computing platforms has stimulated the development of new capabilities in the areas of Artificial Intelligence and System Simulation. High-performance versions of the thermal-hydraulics and neutronics codes, coupled with these new capabilities, allow the real-time simulation and monitoring of integrated nuclear power plant systems of increasing complexity. The current status of several large scale computational capabilities used in the analysis and simulation of nuclear power plant systems is described below.

Reactor System Simulation using Distributed Parallel Computing.

The SAS system of reactor analysis codes developed at Argonne National Laboratory has played an important role in the computer simulation and assessment of energetics potential for both operational transients and unprotected accidents in advanced Liquid Metal Reactors. Two major code systems have been developed, the SASSYS [1] code system and the SAS4A [2] code system. The SASSYS systems analysis code is capable of analyzing a wide range of transients, from mild operational transients through more severe transients leading to coolant boiling in the core. In addition to a point kinetics neutronics treatment, the code provides a detailed thermal hydraulic analysis of the reactor core, inlet and outlet coolant plena, primary and intermediate heat transport systems, steam generators, and emergency heat removal systems. The code can handle any Liquid Metal Reactor design, loop or pool, with an arbitrary arrangement of components. The main applications of SASSYS have been in the simulation and analysis of operational transients and protected accidents and in the analysis of shut-down heat removal. The SAS4A code system has been designed to simulate the initiating phase of hypothetical severe unprotected accidents, which can lead to core disruption and fuel relocation. During such postulated accident scenarios as the unprotected Loss of Flow and Transient Overpower events, a large number of interrelated physical phenomena occur during a relatively short time interval. These phenomena include transient heat transfer and hydrodynamic events, coolant boiling, cladding melting, relocation and freezing and fuel melting, disruption, relocation and freezing. SAS4A includes detailed phenomenological models of the core, such as the LEVITATE [6] model, which can simulate these complex multiphase, multicomponent hydrodynamic events. Due to the neutronic feedback present in a nuclear reactor, these events can significantly influence the reactor power. The SAS4A code provides an integrated and quantitative framework for examining the behavior of various reactor designs during unprotected severe accident conditions. The SAS4A code can be used in conjunction with the SASSYS code, with which it is fully compatible.

Both SASSYS and SAS4A are large, complex FORTRAN codes which have been developed for single-processor mainframe computers. The running time was always a concern, requiring the use of CRAY mainframes and imposing limitations on the complexity of the physical models. The advent of high-performance parallel computing systems promises to relieve these problems. Both SASSYS and SAS4A have a structure that contains both high-grain and low-grain parallelism and can benefit considerably from the use of parallel,

multi-processor computing. In order to describe the behavior of the different core regions the reactor core is subdivided into channels, each channel containing a number of fuel assemblies. All the fuel assemblies in a channel are assumed to behave in an identical manner and the channels interact mainly through the upper and lower plena and through the neutronics feedback. Thus the channels provide a high level parallel structure which is a natural target for parallelization.

In order to take advantage of the opportunities for parallelization of large system codes such as SASSYS and SAS4A efforts have been undertaken to develop a high- performance network computing system that can serve as a platform for research into simulation and control of reactor systems, with emphasis on the Integral Fast Reactor (IFR). A Reactor Simulation and Control Laboratory (RSCL) has been established at ANL, comprising a network of SUN stations dedicated to four areas of reactor analysis: development of faster-than-real-time reactor simulation for design and safety analysis; development and testing of control algorithms; validation of control schemes prepared by universities and other laboratories for testing with the IFR; and development of advanced pattern recognition techniques to detect the onset of degradation of sensors or reactor components.

Central to the goal of designing and evaluating control systems for IFR and other reactors is the ability to simulate the plant behavior with real-time realistic models in an interactive computer laboratory environment. Towards this goal we have installed a newly available innovative message-passing software package known as P4 [7]. P4 allows large Fortran codes to be parallelized across a network of Sun workstations, allowing fuller utilization of existing desktop workstations and achieving mainframe performance at a fraction of the cost. The SASSYS code is used as the simulation engine for the IFR simulator. In addition to the reactor core, SASSYS simulates the primary and secondary coolant loops, and can model either once-through or recirculating steam generators plus components in the balance of the plant. A control system model based on classical control theory is also available. The code is not limited to just the IFR design; rather, it is applicable to a wide range of reactor designs and types of transients. Although SASSYS is the engine for the full-scope plant simulator, the code used for experimentation with distributed parallel computing was the LSAS (or Little SAS) code, which is a scaled down version of SASSYS. The LSAS code includes modelling of the reactor core, reactor kinetics, and the heat transport in the primary and intermediate sodium loops. Unlike SASSYS, LSAS does not model a control system or the balance of plant, nor does it include a detailed steam generator waterside model.

Many aspects of the thermal-hydraulic calculations in LSAS and SASSYS are inherently parallel, and the computationally intensive portions of these calculations can be effectively implemented on a loosely-coupled distributed-memory network computing system using the P4 message-passing paradigm. For our initial parallelization experimentation, a 19-channel EBR-II version of LSAS was employed. Domain decomposition was effected by simply dividing core radial channel calculations across the available processors with load balancing for overhead reduction.

Initial parallelization experimentation focused on making a detailed evaluation of the separate components of both the computation and communication costs. It was quickly learned that the dominant component of the communication overhead cost was the

store-and-forward delay associated with packetization. Use of standard Unix virtual-timer mechanisms enabled us to isolate and quantitatively evaluate the SEND-RECEIVE time as a function of message size between 1 byte to 1 Mbyte. It was learned that with a standard TCPIP-based Ethernet, this communication component is a constant 200 msec for message sizes between 1 and 1536 bytes. For this reason, our early experimentation with parallelization of LSAS produced the frustrating results that the parallelized versions ran slower than the uni-processor version and, moreover, the slowdown factor increased with the number of processors used. At 1537 bytes, the SEND-RECEIVE time drops to only 8.09 msec, thereby effectively reducing the communication overhead by a factor of 25. As the message size increases from 1.5 kbytes to 4 kbytes, the message-passing time increases linearly from 8.09 msec to 11.9 msec. At 4.001 kbytes, the time jumps back up to 200 msec, where it remains constant to 5.5 kbytes.

Although the foregoing behavior was unexpected, it is completely explained by the store-and-forward mechanism within TCPIP. To enhance overall network throughput, TCPIP holds small messages for a maximum of 200 msec, attempting to "pack" multiple short messages for a single transmission. Messages greater than 1.5 kbytes are transmitted immediately. When the message size exceeds 4 kbytes, however, the message is broken into two segments, and the shorter segment suffers the 200 msec store-and-forward delay. For LSAS parallelization, and for our expert system parallelization research (following section), all message sizes were less than 1.5 kbytes. A brief script was written and added to the MASTER and SLAVE P4 Proc Groups to simply pad all messages with zeros, making them 1537 bytes. This reduces the message-passing overhead to only 8.1 msec, which is quite satisfactory for the coarse-grain parallelization objectives of the research reported here.

The LSAS code was successfully parallelized, installed on the network computing system, and run in faster than real time. Use of synchronized interrupts and memory swapping techniques enabled us to tie together a dynamic, color-graphic user interface (running on a stand-alone Sun SparcStation) with the parallelized version of LSAS, so that the integrated system dynamically responds to real-time operator inputs as an operating reactor would. The overall control structure for LSAS as implemented on a five-processor configuration is illustrated schematically in Fig. 1. One of the 23 color-graphic screens that can be brought to the foreground of the operator interface is illustrated in Fig. 2. This simulator technology will be a powerful design and safety analysis tool of EBR-II/IFR for areas where human interactions are critical to safety and operability.

Distributed Parallel Computing with AI-Based Expert System

Research and development in the area of expert systems used in the operation of nuclear power plants is an important component of the computer simulation and analysis conducted at Argonne National Laboratory. Expert system can play an important role in improving the efficiency and safety of nuclear power plants by advising the operators about actions required to correct inefficiencies and about developing abnormal situations. The expert systems must monitor a large number of parameters and provide timely advice to the

control room operator, whose response time must be minimized. The structure of expert systems using artificial intelligence and neural networks makes these applications natural candidates for the use of parallel distributed computing.

An AI based expert system has been developed to be used as an operator decision aid during exposed-fuel operation in LMRs. This expert system is embodied in a device we call a Failed Fuel Surveillance and Diagnosis (FFSD) apparatus [8]. This device monitors, processes, and interprets information from the Delayed Neutron (DN) system and from other key plant variables and displays for the operator the diagnostic information needed to make proper decisions about technical-specification conformance during breached-fuel operation. This operator decision aid enhances plant availability and economics by minimizing unnecessary reactor trips caused by events having no safety significance. By combining and evaluating information from several plant sensors, the FFSD system makes it possible to significantly relax the conservatism in DN shutdown limits in place at operating LMRs without compromising plant-safety assurance. At the same time, the FFSD apparatus reduces complexity and mitigates confusion in the reactor control room. It minimizes the possibility of human error or oversight by providing automatic annunciation of discrepant signals or the incipience of initiating faults. More importantly, even when the FFSD expert system is in its "passive" surveillance mode, the interactive capability is provided for the operator to manually query the status of any component of the system for operability validation. This symbiosis of automatic and manual systems reduces challenges to plant availability while allowing incorporation of the role of the operator in a manner which most effectively augments the achievement of overall plant operability goals. The expert system has been validated using archive data from actual breached-fuel tests in EBR-II. A detailed parametric sensitivity analysis has been performed using Monte Carlo simulation to assess propagation-of-uncertainty characteristics for the system's algorithms as a function of both the DN noise level and the levels of various background components that are present in an LMR. Results from this simulation portion of our investigation, analyzed with standard 3-D response-surface methodology, show that the output uncertainty associated with the computed diagnostic parameters is +/- 3% during full-power operation with EBR-II's present fission-chamber DN detectors. Expert system surveillance of diagnostic parameters from the DN monitoring system will enhance plant safety by providing the operator with rapid identification of off-normal operation, thereby enabling him/her to terminate or avoid any events that might challenge safety or radiological performance guidelines.

The completed expert system has been parallelized on a loosely coupled network, distributed-memory, computing system that enables the computationally intensive kernel of the expert system to run in parallel on a group of SUN workstations. This expert system provided an ideal Fortran test code for distributed computing experimentation insofar as it cannot achieve real-time performance on a single workstation, and the algorithmic structure of its computationally intensive kernel - iterative solution of a large system of coupled differential equations - is readily amenable to standard parallel decomposition techniques. Our goal has been accomplished with the use of the portable distributed parallel-computing system P4 [7]. For the present project the P4 package was exploited to implement a distributed computational environment on a network of SUN workstations that is suitable for testing and validation of computationally intensive reactor-surveillance expert-system

software. Timing studies have shown that the FFSD software parallelizes very well. We have successfully parallelized the system on rings of up to 10 Sun Sparc-2 processors connected by standard ethernet. Speedup factors attained ranged linearly from 99% (of theoretical maximum) for two processors to 95% for ten processors. This approach to distributed-memory parallel processing has proven extremely useful and flexible, and will permit the failed-fuel surveillance expert system to operate on low-cost hardware with a higher input sampling rate (1/s) than would have been possible on EBR-II's current data-acquisition system computer.

Neutronics calculations and parallel computing.

The reactor system simulation codes described in the previous section account for the very important neutronic feedback effects by using simplified neutronics models and neutronic cross-sections appropriate for various reactor types. In order to quantify the effects of the approximations used in these simplified models and to calculate the appropriate neutronic cross-sections, rigorous neutronics models have been developed which require very extensive computations.

The most cpu-intensive reactor physics computations are neutron transport calculations, required in situations where the reactor materials are too heterogeneous and the neutron flux is too anisotropic to permit accurate diffusion theory calculations. The Monte Carlo, nodal transport, or collision probability methods which are applied to these problems often require large amounts of cpu time.

The Monte Carlo code VIM [3] has been used to quantify or understand the effects of the approximations used in other codes. Complex geometries with curved surfaces can be represented very accurately with the combinatorial geometry package. Perhaps more importantly, finely detailed continuous-energy cross sections have been derived directly from the ENDF/B data and include secondary angular and energy distributions. The spectral influence of neutronically different neighboring materials, e.g., coolant, structural, or absorbing elements, is then accounted for in the flux spectra and spatial distributions which develop naturally during the random walk procedure. This obviates geometric and spectral approximations of uncertain validity.

For example, Argonne has applied deterministic transport methods to a number of fast and thermal reactors, including the Experimental Breeder Reactor II (EBR-II). Because EBR-II has a very small core and a core neutron leakage factor of 50%, diffusion calculations underpredict criticality by more than 4%. To make matters more difficult, interposed between the core and the radial blanket is a steel reflector with highly anisotropic scattering, which introduces a diffusion theory error of about 15% in blanket power. The Monte Carlo VIM code treats the transport effect and the anisotropic scattering exactly, but requires about one Sun SPARC-2 cpu-month to estimate accurately reaction rate distributions or for straightforward Monte Carlo perturbation calculations. This has motivated the efforts in parallel Monte Carlo described later in this section.

For two decades, VIM has undergone validation against both critical reactors and assemblies, and against Argonne's other high-precision computational methods. The benchmark experiments include a dozen or more plate-lattice criticals covering a wide range

of compositions characteristic of Liquid Metal Reactors, Light Water Reactors, Heavy Water Reactors, Gas Cooled Reactors, etc. As computing costs decline, VIM has been used increasingly to estimate worths and reaction rates by region or pin, serving as a benchmark for simpler but faster running models. For example, pin powers produced by CASMO [7] were benchmarked against VIM, and advanced BWR pin-bundle designs are being analyzed by utilities using VIM.

To enable the practical use of Monte Carlo estimation of localized quantities in a reactor, e.g., depletion or small worths, work is underway to exploit the statistical independence of individual neutron histories on massively parallel architectures or workstation networks. For code systems like VIM, which have been exhaustively validated over many years, the comprehensive restructuring necessary for implementation on SIMD (and vector) architectures has deterred most Monte Carlo code developers from exploiting these architectures. The arrival of workstation networks and massively parallel MIMD machines with larger distributed memories now permits straightforward parallelization, obviating clever but complicated partitioning by space, event, or energy. Work on parallel VIM has proceeded in two steps. In an early version independent multi-generation simulations, including input processing, were assigned by the master processor, using the P4 message-passing system, to nodes on a workstation network. Such an approach suffers from inefficiency due to unbalanced computing loads on the worker nodes. Furthermore, this method should not be applied to criticality calculations on massively parallel machines because the eigenvalue bias becomes important if the size of a generation on one node becomes small. Recently, VIM has been parallelized for criticality calculations using a finer grain technique, in which the master assigns tasks (histories) to the slave processors and performs all overhead tasks. Because the source data is integrated for each generation, the generation sizes are kept large, so the eigenvalue bias is reduced to practical insignificance.

In addition to Monte Carlo neutronics codes, other large neutronics codes using rigorous deterministic methods have been developed, which can benefit considerably from the use of high-performance computing platforms. GTRAN2 [10], for example, a high performance general geometry collision probability transport code is being developed for integral neutron transport calculations in thermal reactor lattices where the geometry is irregular. The code performs several tasks: ray-tracing, calculation of the collision probabilities, and solution of the equations for the flux. The calculations of chord length by ray-tracing is performed only once for a specific geometry and reused for each energy group. This calculation has not been parallelized. The collision probability calculations are designed for parallel execution on MIMD shared-memory machines, and the flux solver is vectorized. Fig. 3 illustrates a 2-D Flux Distribution in a BWR pin bundle calculated by GTRAN2.

Collaboration with utilities and universities is now underway to perform physics analyses with Argonne codes, such as VIM and GTRAN2, in Monte Carlo depletion, transient and safety analysis, and fuel management. The use of high performance parallel computing will allow these rigorous codes to provide useful results within the limited time constraints imposed by the plant operation.

3-D Thermal Hydraulic Simulation of Reactor Components and Systems.

Argonne has had a long history of developing general purpose 3-D thermal-hydraulic codes to provide detailed simulations of a wide variety of nuclear reactor components and systems. These activities have resulted in the development of the COMMIX [4] family of codes which have evolved in response to the needs of the new reactor technologies and as the capabilities of advanced computer architectures and processing speeds have increased. The COMMIX codes, in continuous development since the mid-70s, have been designed to answer thermal-hydraulics issues arising in the safety analysis of both commercial and advanced reactor designs.

The codes in the COMMIX series are three-dimensional, transient, global analysis models that allow the prediction of temperature, velocity, pressure and phase distribution in separate reactor components (e.g. heat exchangers or reactor plena) or in a group of coupled components such as a reactor primary system. They are based on a porous media formulation that allows the modeling of flow effects caused by complex internal geometry through the use of distributed and directional porosity, distributed flow resistance and local sources and sinks. The full conservation equations of mass, momentum and energy are solved as a boundary value problem in space and as an initial value problem in time. Although all the COMMIX versions have the capacity of solving most thermal hydraulic problems that could arise in the design and safety analysis of reactor systems, they have been individually tailored to some subclass of problems for the sake of general efficiency. For example, COMMIX-1AR/P [11] is an extension of COMMIX which allows the treatment of multiple fluids, includes pumps, allows radiation heat transfer between submerged solids and provides a more efficient numerical treatment. COMMIX-1D, now under development, contains many special features relating to the tracking of air, steam, and hydrogen in a reactor containment vessel. Another specialized version, COMMIX-PPC, is being developed specifically for the analysis of power plant condensers and shell-and-tube heat exchangers.

The use of parallel computational platforms is of significant interest for the COMMIX codes, which have always placed extreme demands on computer resources. This trend is expected to continue as detailed studies of the transient behavior of the next generation of LWRs become necessary. By their very nature, such evolutionary and revolutionary designs have greatly extended the length of reactor transients and, in consequence, placed severe demands on computational resources. Passive safety features that involve transition to natural convection and subsequent decay heat removal by natural circulation will place extreme demands on thermal hydraulic simulation. This is an area where the COMMIX codes should prove invaluable since many of the calculations done in the past have involved buoyancy-driven flows, the study of flow stability and stratification, and other situations where the driving forces are created by quite small temperature variations. In addition, such calculations have been extensively validated by comparison with experiments such as those performed by Kasza et al.[12].

In order to study the advantages of multi-processor computing a shared memory parallel implementation of the COMMIX-1AR/P thermal-hydraulics computer program was developed as part of the Advanced Software Development and Commercialization (ASDAC) project, which involved computer scientists and engineers at the University of Illinois at

Urbana-Champaign and Argonne National Laboratory [13].

The goal of the project was develop a production version of the program that would run on a shared memory multiprocessor Cray X-MP/Y-MP class computer while preserving the functionality of the production program. It was recognized that many man-years of effort had been invested in the development of COMMIX and that commercial benefit would be best achieved by preserving the basic functionality and feel of the code. For this reason efforts concentrated on modifying existing FORTRAN code and did not attempt develop a new prototype for COMMIX.

The code was ported to the various target computer systems and several data sets that would exercise critical sections of the COMMIX program were selected. These data sets were then employed to establish baseline performance of the original ported code and in subsequent evaluation of restructured code. Performance analysis tools were used to identify critical codes segments that would be likely candidates for automatic, manual restructuring and/or algorithmic revision. The process was repeated several times during the optimization process. Initial studies were performed with several small data sets. This type of data set provides detailed timing information which is used to locate critical code segments and measure performance. An industrial strength data set was obtained from Commonwealth Edison which was used in the latter stages of the project. This data set was used to determine the temperature distribution within a nuclear reactor vessel following a PWR steam line break accident and is described in detail below.

The PWR steam line break accident is initiated by the rupture of one of the main steam lines, typically at hot zero power conditions. This results in a rapid cool down of all steam generators via cross connect steam lines until main steam isolation valve closure. This is followed by an extended blowdown of a single steam generator producing an asymmetric temperature distribution in the primary system. COMMIX is used to predict this temperature distribution which is then used to derive the power distributions for use in Departure from Nucleate Boiling calculations. The COMMIX model included the cold leg entrance into the downcomer, the downcomer, the lower plenum, and the core. These were modeled using a cylindrical geometry with 13,104 computational cells. Internal cells were used in the model to simulate the core walls and the curvature associated with the bottom of the vessel. Fig. 4 shows an isothermal surface of the cool plume of water entering one of the four legs of the reactor, as calculated by COMMIX. The colder water flows into the inlet, down through the downcomers, and rises in the core where it will eventually exit at the top. Although COMMIX is a mature code that has been ported to a variety of compute architectures it was necessary to develop a portable version of the code which would compile on SUN workstations, the Alliant FX/80, and the Cray Y-MP. A set of benchmark calculations was run with the original code on each of the machines to establish baseline performance and to locate critical code segments. The results of our baseline benchmarks were similar for the Cray and Alliant. The bulk of the time was spent in routines that were constructing algebraic equations for momentum equations and energy equations (62%) or in routines solving these algebraic equations (30%).

Early experience compiling the original code to run concurrently on the Cray X-MP/48 and the Alliant FX/80 was not startling. Both the Cray and Alliant compilers have built in features that recognized parallel constructs and interpret compiler directives to

generate concurrency but a penalty has to be paid for communication between processors. The original code slowed down by a factor of 0.4 on 4 processors of the Cray X-MP/48 and was sped up by a factor of 1.2 on 8 processors of the Alliant FX/80. There are many reasons for this poor performance. On the Cray much of the time was spent waiting for the processes to synchronize. The performance a single processor of the Cray Y-MP is about 24 times faster than that of the Alliant FX/80. For that reason the penalty in terms of time waiting for system resources and communications between processes is a much larger percentage of the compute time on Cray.

Most of the success at optimizing COMMIX has been the result of rewriting and reorganizing the most computation-intensive subroutines. This hand restructuring is intensive and time consuming. Examination of the routines that construct algebraic equations for the momentum and energy equations revealed complex logic and indirection that impeded vectorization and autotasking. A single integer array was employed to encode several types of boundary conditions and its values was used as a pointer if its sign was negative. The use of arrays as pointers is called indirection. Indirection is often effective, but current vectorizing compilers can only optimize one level of straight forward indirection. Rewriting these routines with separate loops, flags and pointers for the various types of boundary conditions resulted in speed up of a factor of 7.5 on the Cray Y-MP for the construction of the energy equation.

Studies of concurrent performance were made with partially optimized code. The curves labeled with the superposed asterisk in Fig. 5 shows the concurrent performance of the code on the Cray Y-MP/48 and the Alliant FX/80 after the restructuring of the routines that construct the energy equations. The performance improves only slightly on the Cray and there is a speed up of about 2.2 on the Alliant. The performance was improved on both the Cray and Alliant when the X-, Y-, and Z-momentum routines were restructured to run concurrently. The speed ups for 4 processors on the Cray and 8 processors on the Alliant were 2.0 and 3.0 respectively. It is worth noting that the curves for the Alliant tend to flatten with increased number of processors.

In the concurrent mode, the restructured COMMIX spends 30% of the time constructing equations on the Cray and 10% of the time constructing equations on the Alliant. The bulk of the time is spent solving algebraic equations. This is in contrast to the situation before restructuring, when the bulk of the time was spent constructing the equations.

Much of the effort on COMMIX has concentrated on improving the subroutines which construct equations. These algorithms involve local calculations and nearest neighbor communications. For that reason they lend themselves well to vectorization and should be scalable on massively parallel machines. In our studies we have refrained from changing the overall program organization of COMMIX and the equation solving routines. Reorganization of the program and underlying data structures to enable domain decomposition would provide a mechanism to achieve further improvements in performance. New scalable methods are becoming available for solving algebraic systems on massively parallel computer systems (14). These solvers have exhibited gflops (billion mathematical operations per second) on the 528 processor Intel Delta. The performance of COMMIX running the Commonwealth Edison Problem on the Cray Y-MP is about 72 mflops (million mathematical operations per second). For this reason it is reasonable to expect a variant of COMMIX to run several

hundred times faster on the current generation of massively parallel computer systems.

Conclusions.

Over the years, several families of large thermal-hydraulic and neutronic codes have been developed at Argonne National Laboratory, providing a detailed computer simulation of the complex phenomena that occur in nuclear reactors and nuclear power plant systems. These codes have achieved national and international recognition and have been used extensively in the design and analysis of many types of reactors.

Work is currently underway to restructure these large computational tools to allow them to take advantage of the capabilities of distributed computing networks and massively parallel computers. The advent of high-performance parallel computing holds the promise of dramatically increasing the performance of computer simulations, allowing the use of complex thermal-hydraulics and neutronic models in real-time simulations and parametric studies for the design and safety analysis of nuclear power plants.

The adaptation of the existing, detailed thermal-hydraulic and neutronic codes on high-performance parallel computational platforms makes possible the ambitious goal of a Numerical Nuclear Power Plant, which would simulate in detail the response of the plant, including the reactor, to specified transients. The Numerical Nuclear Power Plant is a central element of the Nuclear Power Analysis Center currently being organized by a consortium which includes Argonne National Laboratory and a group of Universities and Utilities.

References:

1. F.E.Dunn, F.B.Prohammer, D.P.Weber, R.B.Vilim, "The SASSYS-1 LMFBR Systems Analysis Code," Proc. Int. Topical. Meeting on Fast Reactor Safety, Knoxville, Tennessee, (April 1985).

2. A.M.Tentner et.al., "The SAS4A LMFBR Whole Core Accident Analysis Code", Proc. Int. Topical Meeting on Fast Reactor Safety, Knoxville, TN (April 1985).

3. R.N.Blomquist, "VIM", Proceedings of the International Topical Meeting on Advances in Mathematics, Computations and Reactor Physics, April 28 - May 2, 1991, Pittsburgh, PA.

4. W.T.Sha, H.M.Domanus, R.C.Schmitt, J.J.Oras, E.I.Lin, "COMMIX-1: A Three-Dimensional Transient Single-Phase Component Computer Program for Thermal-Hydraulic Analysis", NUREG/CR-0785, Argonne National Laboratory, Argonne, IL (Sept. 1978).

5. C.E.Till, "The IFR Status and Prospects", Trans. American Nuclear Society, 66, p.351, Chicago, IL (Nov. 1992).

6. A.M.Tentner and H.U.Wider, "LEVITATE-A Mechanistic Model for the analysis of Fuel and Cladding Dynamics under LOF conditions", Proc. Intl. Meeting on Fast Reactor Safety Technology, Seattle, WA (1979).

7. D. W. Leibfritz, K. C. Gross, and W. K. Nickless, "P4--A Portable Message Passing System for Distributed Parallel Computing in Fortran," Intnl. Topical Mtg. on Advances in Math., Computations, and Reactor Physics, Vol. 4, pp. 20.2 3-1--20.2 3-4 (April 1991).

8. K. C. Gross, R. E. Hawkins, and W. K. Nickless, "Operator Decision Aid for Breached Fuel Operation in LMRs," Trans. Am. Nucl. Soc., 63, 118-119 (1991).

9. M. Edenius, et al, "The Physics Model of CASMO-4", Advances in Mathematics, Computations, and Reactor Physics, Proceedings of the Intenational Topical Meeting, Pittsburgh, April 28 - May 2, 1991.

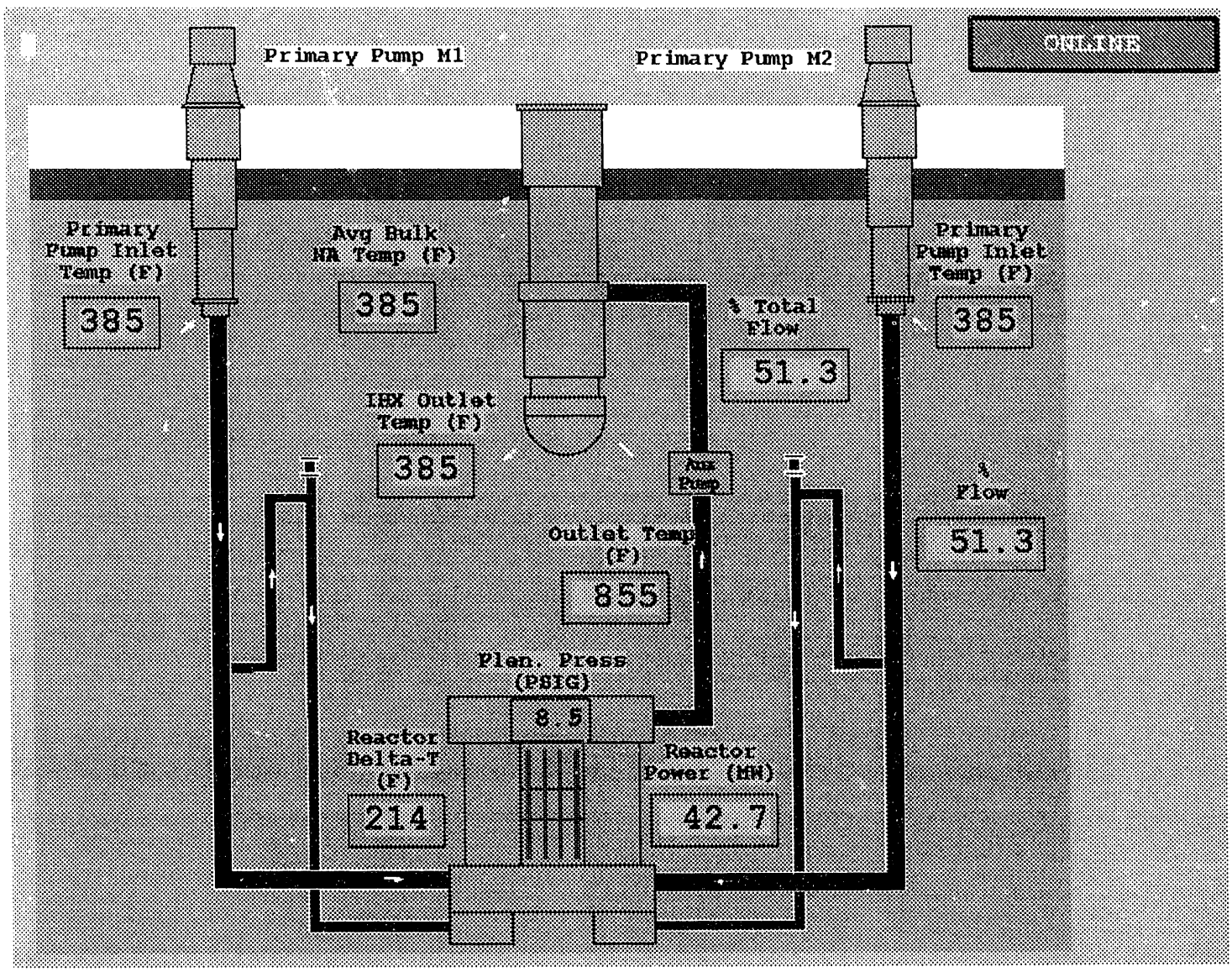
10. J. L. Vujic , "GTRAN2: AGeneral Geometry Transport Theory Code in 2D" , Proceedings of the International Topical Meeting on Advances in Mathematics, Computations, and Reactor Physics, , Pittsburgh, PA., April 28-May 2, 1991.

11. R.N.Blomquist, P.L.Garner, E.M.Gelbard, "COMMIX-1AR/P: A Three-Dimensional Transient Single-Phase Computer Program for Thermal-Hydraulic Analysis of Single and Multicomponent Systems", ANL-90/45, Argonne National Laboratory, Argonne, IL (July 1991).

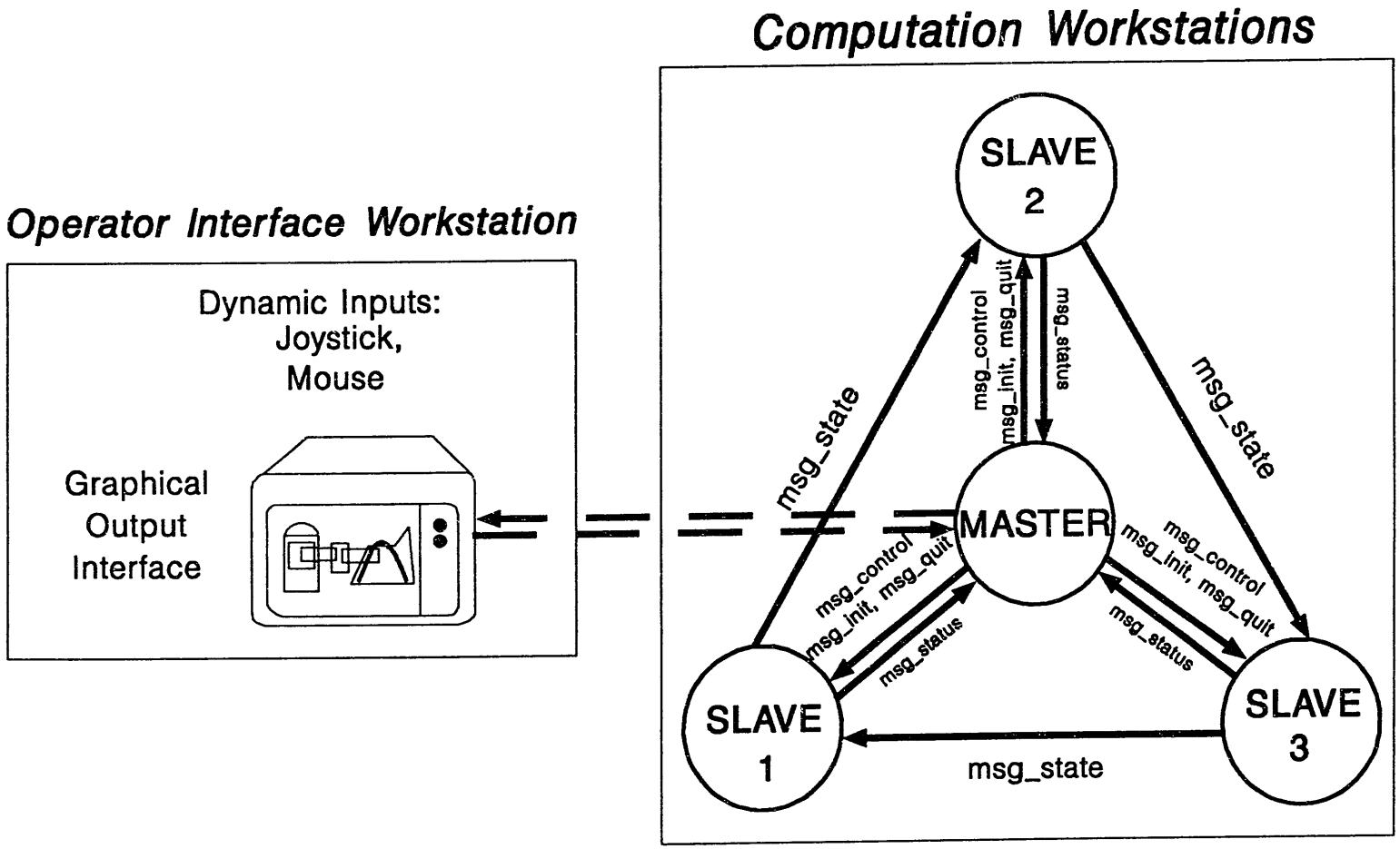
12. K.E.Kasza, T.M.Kuzay, J.J.Oras, "Overview of Thermal Buoyancy Induced Phenomena in Reactor Plant Components", Third Intl. Conf. on Liquid Metal Eng. and Technology in Energy Production, Oxford, England, April 1984.

13. S.Aslam, E.Gallopoulos, M.Ham, T.Canfield, M. Minkoff, R.Blomquist, "Experiments in thermal hydraulics simulation: multiprocessing COMMIX", Proc. of the Fifth SIAM Conference on Parallel Processing for Scientific Computing, 1992.

14. M. T. Jones and P. E. Plassmann, "An Efficient Parallel Iterative Solution for Large Sparse Linear Systems", Proc. of the IMA Workshop on Sparse Matrix Computations: Graph Theory and Algorithms, Publshed by Univ. of Minnesota, Minneapolis, MN 1991.


Primary Pump M1
Primary Pump M2
Primary Pump Inlet Temp (F)
385
Avg Bulk NA Temp (F)
385
% Total Flow
51.3
Primary Pump Inlet Temp (F)
385
IHX Outlet Temp (F)
385
Aux Pump
% Flow
51.3
Outlet Temp (F)
855
Plen. Press (PSIG)
8.5
Reactor Delta-T (F)
214
Reactor Power (MW)
42.7

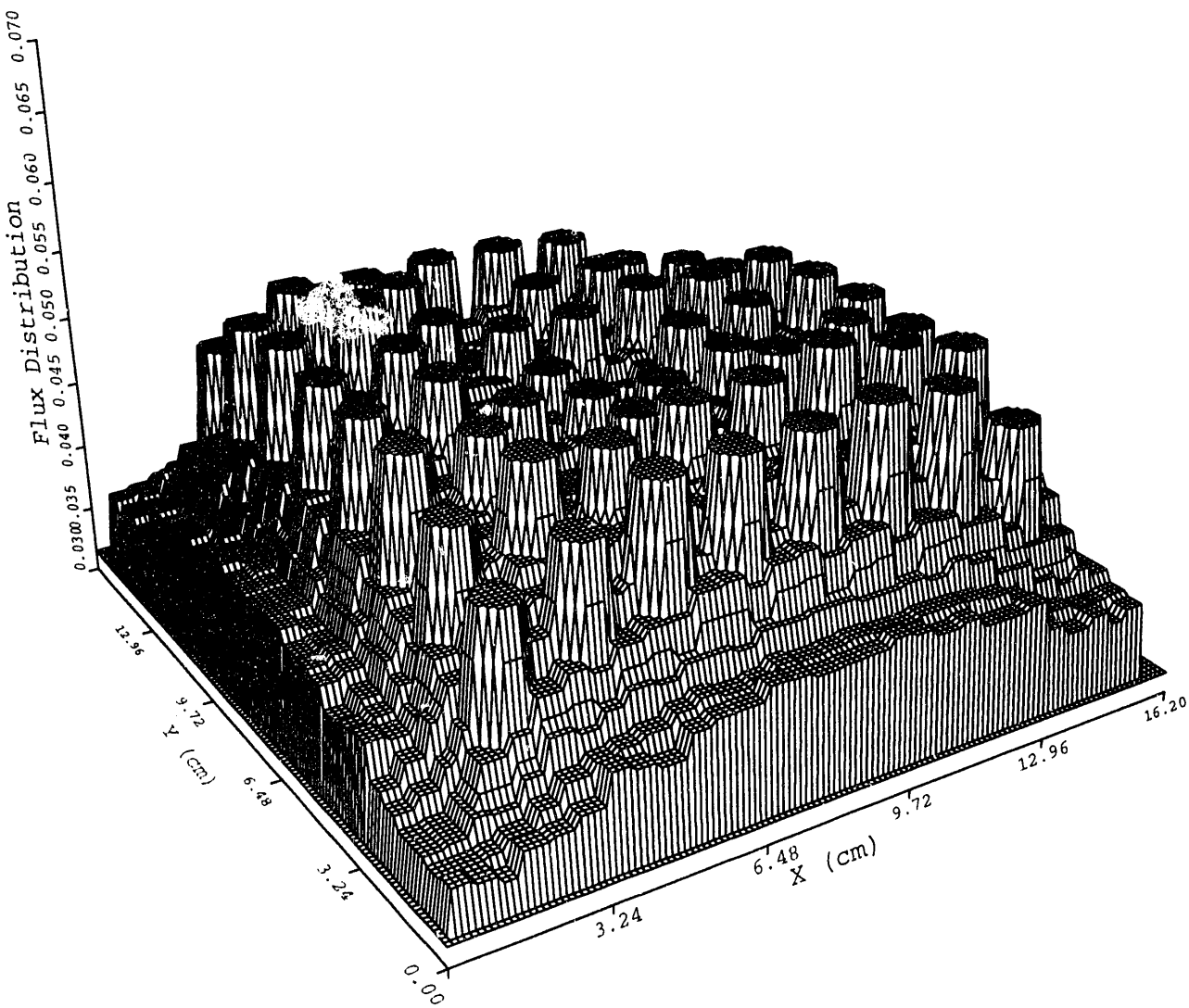
Interprocess communication structure for a four-node implementation of P4.

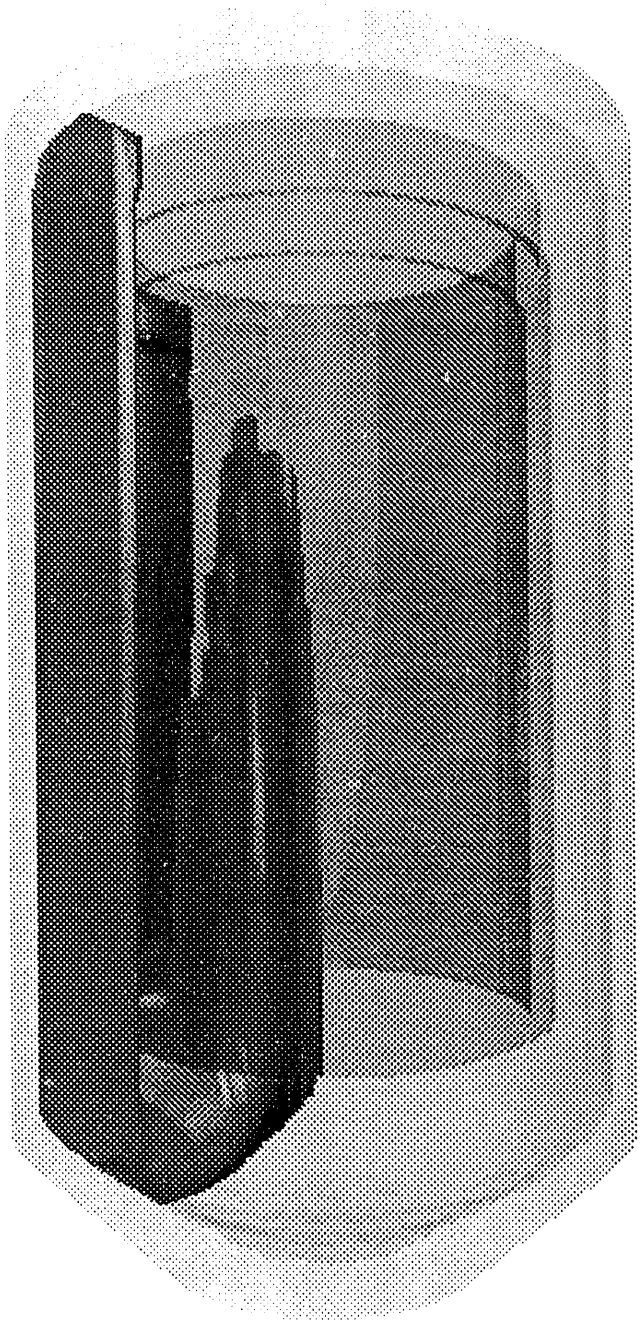


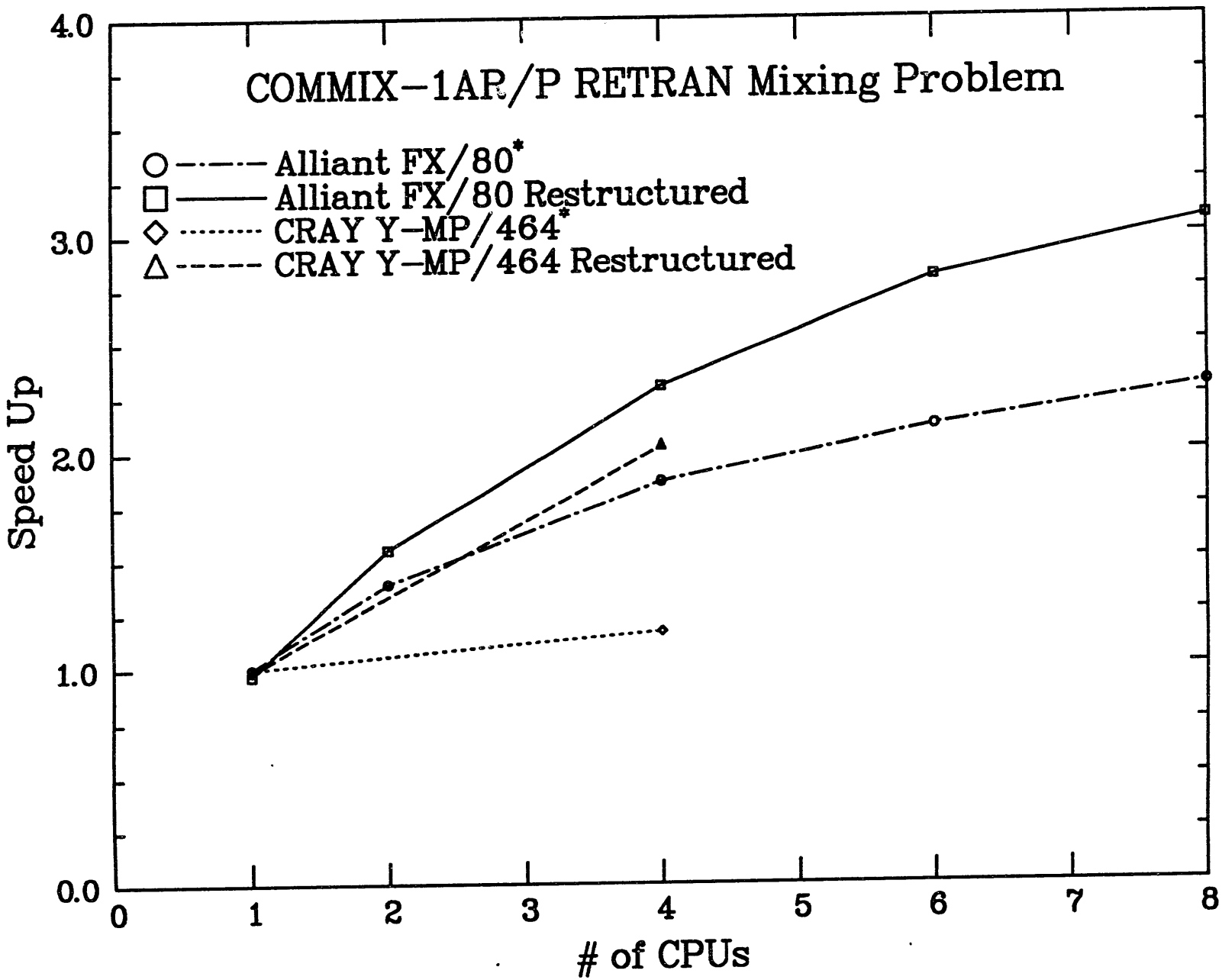
GTRAN2: FLUX DISTRIBUTION IN BWR ASSEMBLY

CASE WITH Gd RODS: 996 ZONES, 10 ANGLES, 0.1 CM STEP

ENERGY GROUP 1 OF 7






COMMIX–1AR/P RETRAN Mixing Problem
Alliant FX/80*
Alliant FX/80 Restructured
CRAY Y–MP/464*
CRAY Y–MP/464 Restructured
Speed Up
# of CPUs
4.0
3.0
2.0
1.0
0.0
0
1
2
3
4
5
6
7
8

END

DATE
FILMED
5/20/93