Exploiting Parallelism in Chemical Engineering Computations

B. Keith Harrison
Department of Chemical Engineering
University of South Alabama
Mobile, AL 36688

There is substantial interest in exploring techniques to take advantage of advanced computer architectures which are expected to stress parallel processing to an ever greater degree. One aspect of parallel processing that holds great promise as a means of speeding up calculations, is the breaking down of computational tasks into independent sections so that several can be executed simultaneously on multiple processors.

To date, the main approach for exploiting this type of parallelism (also termed "multitasking" and "multiprocessing") has centered on manual intervention in the computer code to restructure parallelism into an application (e.g., Cera, 1989). This requires considerable analysis and often requires the complete rewriting of existing application programs. Some thought is also being given to the strategy of developing algorithms which can analyze particular classes of applications for exploiting parallelism (e.g., Chimowitz and Bielinis, 1987). Both of these approaches, of course, hold promise but suffer from the disadvantage of considerable development cost for each application or class of applications. Also, development efforts for applications computed on one set of computer hardware are not likely to result in code that is portable to a different manufacturer's computer hardware.

Procedure and Results

As an alternative to the above methods, another approach is investigated here. The strategy is to take advantage of a class of utility programs being developed by several sources, that modify code automatically to take advantage of the parallel architecture available in a particular piece of computer hardware. This approach does not require any code intervention and would let one take immediate advantage of advanced computer architectures with existing code.

Some experiments have been accomplished using the sequential modular flowsheet simulator FLOWTRAN (Rosen and Pauls, 1977). The flowsheet simulator processes the user's flowsheet description into a FORTRAN program, which is then executed for a solution to the user's stated problem. For this study, a code analyzing utility program (CFT77 Reference Manual, 1988) was applied to the generated FORTRAN program before execution. The utility modified the FORTRAN code such that directives were inserted automatically at appropriate places to cause portions of the code to execute in parallel. The code was then executed on a two-processor machine, the Cray XMP/24. Performance statistics were collected for four different but typical flowsheet simulation problems.

The results shown in Table 1 indicate that large fractions of the code were converted for parallel processing. In the best case, 96% of the code after conversion can run in a parallel mode. This means that only 4% of the computation time for this particular test case need be spent in sequential calculations. The maximum speed-up that can be achieved due to executing a stated fraction of the code in parallel can be readily calculated by applying Amdahl's Law (Amdahl, 1967) indicated as Eq. 1 below:

$$\text{maximum speed-up} = \frac{P}{P - fP + f} \tag{1}$$

where

P = number of processors

f = fraction of code that can be performed in parallel on P processors

Maximum speed-up amounts calculated for each of the four test cases are shown in Table 1. Actual speed-ups experienced showed great variability but were always less than the maximums indicated. This is due to the problem of attaching and maintaining the availability of both processors in a heavily loaded multiple user environment for the duration of solution of a typical case. Time needed for systems coordination tasks, etc., is also a source of a reduction in speed-ups achieved. There is evidence however, that close to maximum speed-up values can be achieved on dedicated multiprocessor machines (Chevrin, 1989).

More than two processors would, of course, offer a much

Table 1. **Parallel Performance for Test Cases**

Test Case	% of Code Made Parallel	Max. Speed-Up Amdahl's Law
1	96	1.92
2	88	1.78
3	94	1.88
4	75	1.60

greater potential speed-up, as can be readily calculated by Amdahl's Law. However, the success of the utility program would be expected to be more modest as it would have to keep a greater number of processors concurrently occupied.

Investigation of the actual changes made in the code by the utility program reveals exploiting parallelism on the do-loop level. For example, 105 out of 232 do loops were executed in parallel in 52 routines comprising a major portion of case 4. However, opportunities for parallel operations, readily seen by an analyst examining the chemical process flowsheet, looking for parallel unit operations, were often not discovered by the utility. Even with these lost opportunities, the focus of the utility on lower level program segments brought considerable success, as demonstrated in the percentage of the code made parallel. This is true in part because much of the calculation time is spent at a very low level in the structure of the flowsheet simulation. For instance, it was found that approximately 50 to 65% of the execution time in the case study simulations were spent in the low-level physical property routines.

It should also be mentioned that there is a trade-off between exploiting multiprocessor parallelism and making use of other advanced architectural features such as pipeline vector operations.

The experiments reported here offer encouragement to the possibility of taking substantial advantage of the advanced computer architecture feature of multiprocessor parallelism without rewriting existing computer codes. In view of the enormous investment in existing chemical engineering computer codes, approaches that do not require code intervention are attractive.

Acknowledgment

This work was supported by a grant from Cray Research Inc. The Monsanto Company furnished its simulator FLOWTRAN for this study.

Literature Cited

Amdahl, G. M., *AFIPS Conf. Proc.*, **30,** 483 (1967).

Cera, G. D., "Parallel Dynamic Process Simulation of a Distillation Column on the BBN Butterfly Parallel Processor Computer," *Comp. Chem. Eng.*, **13**(7), 737 (1989).

CFT77 Reference Manual, Publication SR-0018C, Cray Research, Inc., Mendota Heights, MN (1988).

Chervin, R. M., "Microtasking Climate Models at NCAR," *Cray Channels,* **11**(2), 6 (1989).

Chimowitz, E. H., and R. Z. Bielinis, "Analysis of Parallelism in Modular Flowsheet Calculations," AIChE J., **33,** 976 (June, 1987).

Rosen, C. M., and A. C. Pauls, "Computer-Aided Chemical Process Design: The FLOWTRAN System," Comp. Chem. Eng., **1,** 11 (1977).

Manuscript received Aug. 28, 1989, and revision received Nov. 27, 1989.