LA-UR -81-3528

CONF-820202--3

MASTER

LA-UR--81-3528

DE82 004360

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE IMPLEMENTING ASPEN ON THE CRAY COMPUTER

AUTHOR(S) Kenneth H. Duerre and Amar C. Bumb

SUBMITTED TO Computer-Aided Process Design and Simulation with ASPEN, AIChE National Meeting, Orlando, FL, February 28 - March 4, 1982

DISCLAIMER

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

IMPLEMENTING ASPEN ON THE CRAY COMPUTER

by

Kenneth H. Duerre
Los Alamos National Laboratory
Los Alamos, NM 87545

Amar C. Bumb
Department of Chemical Engineering
University of Wyoming
Laramie, WY 82071

ABSTRACT

This paper describes our experience in converting the ASPEN program for use on our CRAY computers at the Los Alamos National Laboratory. The CRAY computer is two-to-five times faster than a CDC-7600 for scalar operations, is equipped with up to two million words of high-speed storage, and has vector processing capability. Thus, the CRAY is a natural candidate for programs that are the size and complexity of ASPEN. Our approach to converting ASPEN and the conversion problems are discussed, including our plans for optimizing the program. Comparisons of run times for test problems between the CRAY and IBM 370 computer versions are presented.

I. INTRODUCTION

The Los Alamos National Laboratory provides process engineering design studies, analyses, and economic studies of fossil fuel conversion processes for the Laramie Energy Technology Center (LETC), as well as for the Laboratory. Other areas of interest appropriate for ASPEN study are nuclear fuel cycle and nuclear waste processes. ASPEN will make it possible to do mass and energy balances with much greater speed and accuracy, allowing more and better comparisons of alternative flow sheets.

The computing power at Los Alamos currently consists of four CRAY computers with up to two million words of fast memory per machine, four Control Data Corporation (CDC) 7600 computers, and a host of other less powerful machines, all with a supporting common file system of 4.7 trillion bits of disk and archival storage. The operating system for our CRAY machines is the Cray Time Sharing System (CTSS), which differs from the CRAY Operating System (COS) because of the Laboratory's special requirements and to provide continuity for users between different machines (the CDC machines have a similar system, the Livermore Time Sharing System (LTSS)).

The CDC machines have much smaller fast memories than the CRAY and would require extensive overlaying for a program the size of ASPEN. This was the primary reason for our decision to implement ASPEN on the CRAY rather than on the CDC 7600 computer. There are secure and unclassified partitions in the Los Alamos computer network. The ASPEN source file storage and maintenance are in the unclassified partition, which permits noncleared personnel to perform ASPEN maintenance and ASPEN run preparation. A more detailed discussion of the computing power at Los Alamos may be obtained from Ref. 1.

II. THE IMPLEMENTATION PLAN

The ASPEN program was written by the ASPEN group at the Massachusetts Institute of Technology (MIT) for an IBM 370 computer. Double precision was used in calculations to retain accuracy because the word size for the IBM machine was only 32 bits. The statement by the ASPEN group that the double-precision version of ASPEN was more thoroughly debugged than the single precision CDC version contributed to our decision to implement ASPEN in three stages.

1. The double-precision version would be installed first to obtain an inefficient, but early, version of ASPEN for our use in programmatic work and to provide checks against the single-precision version, as some incompatibilities were anticipated.
2. The single-precision version would then be installed to provide greater efficiency in running speed and, hence, lower run costs.
3. The latest release of ASPEN would be incorporated into the single precision version. We would then perform timing analysis of ASPEN runs and, if necessary, optimize the critical programs by vectorization and improved input/output schemes.

The ASPEN program is composed of approximately 300,000 lines* of FORTRAN code with 1500 subroutines. That this could create a logistics problem in

*This number increases almost daily and includes comment cards.

terms of maintenance and editing was apparent from the start of the project. This logistics problem caused conversion updates of ASPEN to be made as much as possible with the UPDATE[2] utility to provide documentation of the changes. Other changes such as changing IBM REAL*8 declarations to DOUBLE declarations required use of a text editor. Early conversion effort was devoted to translating the EBCDIC (IBM) character set files to ASCII for the CRAY and to creating back-ups of these files on the common file system.

The primary areas requiring conversion work were the input/output operations (which require character string and byte manipulation) and the random access reads and writes associated with the ASPEN data bases.

III. CONVERSION EXPERIENCE

A manual[3] for installing ASPEN on the IBM computer and notes on the areas of code requiring conversion for non IBM computers was included with the Universal ASPEN tapes.

The notes that flagged machine-dependent code in the lexical processor were especially helpful, although not all affected code was flagged. These corrections consisted primarily of rewriting the MVBYTE routine and aligning the characters in the 8-byte word (vs IBM's 4-byte word) before and after calls to MVBYTE.

The random (direct) input/output (I/O) associated with the ASPEN data bases caused more problems than the lexical analyzer. The CTSS and COS operating systems require keeping track of the disk record addresses. The direct I/O in ASPEN mostly (but not always) consists of reads and writes of fixed length records.* Because the record number and length are passed to the I/O routines, it is relatively easy to calculate the address for CTSS; namely,

$$LDA = (NREC-1) \cdot LREC, \qquad (1)$$

where LDA is the disk address, NREC is the record number, and LREC is the length of the record in computer words.

*ANSI FORTRAN 77 requires random I/O records of fixed length.

This formula holds if the records are of equal length for both input and output. Cases where this was not true in the original code caused the memory to be overwritten and a complex bit of detective work was required to find the cause of this problem.

The formatted direct I/O required the use of FORTRAN compiler routines ENCODE and DECODE. This led to the discovery of an ASPEN routine named DECODE. Subsequently, other FORTRAN in-line functions such as INT and SIN were found to have been used as variable names in ASPEN routines. The names of other ASPEN routines such as COPY, CENTER, and CONVI were the same as names found in our system libraries. These ASPEN routine names were changed to avoid conflicts.

The CRAY FORTRAN compiler (CFT version 1.09) is more restrictive than IBM compilers in mathematical operations because it does not permit division by zero, raising the value zero to a nonintegral power, and so on. It also requires that the number of arguments in a subroutine call not be greater than the arguments defined for that routine; that is, if a call is made to a subroutine with three arguments in the calling list, the subroutine definition must also contain three arguments. The subroutine WRMAIN was modified to make the number of calling arguments agree for simulator program calls. The CFT compiler treats the declaration REAL*8 as single precision; therefore, we replaced each occurrence of this declaration with the DOUBLE declaration.

Some tape transcription errors occurred, and these usually caused compiler errors. Machine language routines such as TIMLIM and RESET were simply converted to FORTRAN dummy routines. Substitutions were made for other routines such as CDATE. Other routines such as DTAN and DARCOS had no analogs in our system so DTAN was changed to DSIN(x)/DCOS(x), and DARCOS was replaced with the single-precision function ARCOS(x).

The ASPEN source code proved to be an excellent vehicle for finding bugs in the CFT compiler. These errors were difficult to find and, in one instance, required extensive consultation with the ASPEN staff to isolate the problem. Other bugs in CFT were uncovered because of the extensive use of double precision functions and variables not normally required by most CRAY users.

The conversion problems ranked in order of difficulty to implement or difficulty to debug were as follow.

1. CFT compiler errors
2. Direct input/output with partial record reads
3. Incompatibilities in the single-precision version of the data management routines
4. Logistics problems* caused by the size of the source code and data
5. All other types of problems

IV. STATUS OF THE CRAY VERSION OF ASPEN

The double-precision version of ASPEN is now running on our CRAY computers. Sample problems have been run successfully for the following five systems, which must precede the input translator and system simulator runs.

1. Insert file management system (to create and maintain insert files).
2. Data file management system (to create and maintain physical property data bases).
3. Data management system (to check the routines used by the simulator in data allocation and retrieval).
4. Table building system (builds system definition tables and files for use by the simulator).
5. The system that reports the contents of these system tables.

Input processor and process simulator input test problems have also been run successfully. These problems covered tests of

a. all the unit operation models,
b. the data regression system,
c. user creation of physical property data bases,
d. convergence algorithms,
e. creation of user insert files,
f. user FORTRAN blocks, and
g. use of parts of a. through f. in a simulation run.

An ASPEN program has been written to model a simplified eastern-shale-burning power plant as part of a LETC-funded study. It appears to be close to successful execution using the CRAY ASPEN code. A more complex ASPEN program to model all the major components of a complete eastern-shale power plant has

*By logistics problems, we mean the extra effort required to operate on and handle that volume of source code and data.

also been written, and a program to model the Paraho DH oil-shale-retorting process is being planned. These programs represent a logical progression of increasing complexity for ASPEN applications.

The single-precision version of ASPEN is currently being installed on the CRAY and, as expected, some incompatibilities have been found.

V. RUN TIME COMPARISONS

The use of double-precision variables and functions on the CRAY computer imposes severe timing penalties[4] and restricts the use of its vector architecture. One normally need not use double-precision on the CRAY because its word size is twice that of an IBM 370 computer. The approximate penalties one must pay are shown in Table I.

With these caveats, we now present in Table II some relative times for sample problems run on the CRAY and MIT IBM 370 computer in the double-precision mode. The sample problems were selected from the test problems in the installation manual and universal tape.

TABLE I

APPROXIMATE FACTORS OF GREATER EXECUTION TIME
FOR DOUBLE-PRECISION VARIABLE OPERATIONS
WHEN SINGLE-PRECISION VARIABLES WOULD SUFFICE

Operation	Factor*
+ (add)	25
- (subtract)	25
x (multiply)	33
÷ (divide)	33
** (exponentiate)	37

*Multiply single-precision execution time by this factor to obtain the double precision execution time.

TABLE II

COMPARISON OF CENTRAL PROCESSOR TIME
FOR ASPEN TEST PROBLEMS

Problem	Time (s) IBM 370*	CRAY**
UOSA	58	18
UOSB	38	16
PP	53	16
CC	47	12
DRS	38	19

These test problems were

a. UOSA, all unit operations models except staged separations, solids handling and separation models;
b. UOSB, those unit operations models not included in a., plus loop control and a FORTRAN Block;
c. PP, physical property test model;
d. CC, convergence and control test problem; and
e. DRS, a data regression test problem.

VI. PLANS FOR PROGRAM OPTIMIZATION

The gross statistics for ASPEN runs thus far show a tendency toward being input/output-bound relative to central processor time. Successful implementation of the single-precision version will almost certainly increase this tendency.

The CFT compiler option, flow trace,[5] and local routines will be used to sample runs in a more detailed manner to determine which routines are consuming the most time. We will then study these routines to see which offer

* Times obtained from MIT.
**Double precision version.

the best opportunities for optimization. The optimization hints in Refs. 4 and 6 will be used to complete the task.

Computing charges at Los Alamos are not simply for central processor time used, so detailed attempts to vectorize may be less prudent than choosing to optimize the input/output operations as well as to reduce memory requirements.

VII. CONCLUSIONS

We now have a useful, albeit inefficient, CRAY version of ASPEN running at Los Alamos. The reasons for further conversion of ASPEN to single precision are compelling, and we will pursue this activity.

The lexical analyzer and direct input/output features of ASPEN were highly nonportable; however the notes supplied by MIT were helpful in the CRAY implementation.

More testing of the CRAY ASPEN version is warranted, particularly on more complex problems.

The ASPEN program is extremely flexible; this is responsible for its complexity and problems in implementation, but it is also a measure of its usefulness.

ACKNOWLEDGMENTS

The Los Alamos National Laboratory computer user services group, especially Chester S. Kazek, Jr., was most helpful in bringing ASPEN on-line on the CRAY computer. Thanks also are due Tony Warnock of CRAY Research Incorporated.

REFERENCES

1. Carolyn Robinson, "Computing at Los Alamos," Los Alamos National Laboratory brochure LAL81-17 (September 1981).

2. Update Reference Manual, Control Data Corporation Publication No. 60449900 (1975).

3. ASPEN IBM/OS Installation Manual, Massachusetts Institute of Technology Energy Laboratory Publication (December 16, 1980).

4. Tony Warnock, "Computing Education Program CFT-Dependent FORTRAN Class," Cray Research Institute unpublished class notes (June 23-24, 1981).

5. FORTRAN (CFT) Reference Manual, Cray Research Incorporated Publication No. 2240009 (May 1980).

6. Lee Higbie, "Vectorization and Conversion of FORTRAN Programs for the CRAY-1 (CFT) Compiler," Cray Research Incorporated Technical Note No. 2240207 (June 1979).