Next Generation Oil Reservoir Simulations

Wayne Joubert, Los Alamos National Laboratory,
Los Alamos, NM USA

ABSTRACT: *This paper describes a collaborative effort between Amoco Production Company, Los Alamos National Laboratory and Cray Research Inc. to develop a next-generation massively parallel oil reservoir simulation code. The simulator, code-named Falcon, enables highly detailed simulations to be performed on a range of platforms such as the Cray T3D and T3E. The code is currently being used by Amoco to perform a sophisticated field study using multiple geostatistical realizations on a scale of 2-5 million grid blocks and 1000-2000 wells. In this paper we discuss the nature of this collaborative effort, the software design and engineering aspects of the code, parallelization experiences, and performance studies. The code will be marketed to the oil industry by a third-party independent software vendor in mid-1996.*

Introduction

In 1994 Los Alamos National Laboratory initiated a cooperative research and development agreement (CRADA) with Amoco Production Company and Cray Research Inc. to develop a commercial-quality parallel oil reservoir simulation code. The parallel reservoir simulator, code-named Falcon, is based on the Young/Stephenson compositional reservoir simulation model as embodied in Amoco's proprietary GCOMP simulator, a production-quality reservoir simulation code for serial and vector processors. Beginning from a Connection Machine CM Fortran prototype, the code has been ported to a variety of architectures including the Cray T3D and other message passing environments. This paper discusses some of the technical challenges involved with developing the code. Amoco intends to commercialize the Falcon code in mid-1996 through a third-party independent software vendor.

The Parallel Model

The Falcon code is based on a three-dimensional finite difference discretization of the reservoir grid. The code currently implements the IMPES formulation for solving the equations of fluid flow; fully implicit compositional capabilities are currently being added to the code. We plan later in this CRADA effort to incorporate local grid refinement near wells and faults. Falcon has capabilities for multiple rock types and rock compressibilities, vertical and deviated wells, well rate constraints and restart capabilities.

The topologically regular 3-D Cartesian grid generated by Falcon admits to a decomposition into subgrids across processors. A 2-D grid decomposition in x and y is employed, keeping the z axis on-processor to facilitate the use of z-line-based linear solvers. In the message passing version, well perforations are stored locally to the processor which owns the subgrid which contains that well's perforations, reducing communication overhead for well operations.

Software Design

The development of a commercial production-quality parallel code has a variety of challenges that are not always present in the development of more research-oriented codes. Besides the need for efficient parallel performance, the code must be well-designed and maintainable for the long term. The code must also be based on language standards that are likely to persist for the foreseeable future. It is difficult to achieve all of these goals optimally, and some tradeoffs are inevitable.

The base language of the Falcon code is Fortran 90. The source code is maintained with C-preprocessor directives which make it possible to generate either single-processor Fortran 90 code, High Performance Fortran (HPF) code or node Fortran 90 with message passing. Using this approach, most of the code is kept free of machine-dependent constructs. The message passing layer of the code containing point-to-point messaging, global operations, etc. is kept isolated in a single source file which can be easily changed to port to other environments, e.g. MPI, etc.

For the purposes of code development, maintenance, and rapid prototyping, the relative simplicity of the HPF is key. On the other hand, message passing allows greater user control of the hardware and allows the implementation of more intricate algorithms, e.g. complex linear solver strategies. To strike a balance between these two extremes and to address the full range of target architectures, the development process for new code features begins with an HPF development that is then "mimd-ized" to run also in message passing environments.

For certain compute-intensive parts of this code, such as the linear solver and the matrix coefficient generation, it is possible to locally "break" this programming model by writing highly-optimized machine-specific code which is kept alongside the generic code.

Linear Solver

The linear system solves are typically the most time-consuming part of the reservoir simulation process. Because of this, considerable effort was aimed toward optimizing the linear solver.

The current solver for the IMPES model is a red/black line SOR solver. It is fairly straightforward to modify a CM Fortran version of this code to perform in a message passing environment, by adding several communication calls to the code. However, the resulting code runs very slow on the T3D. It is possible to improve the performance by separating the red and black grid points into separate arrays, resulting in better use of cache. A third version of the solver was developed which was heavily optimized using SHMEM communication calls, system BLAS calls for arithmetic, and careful alignment of vectors to minimize cache interference. This highly optimized version was able to attain in excess of 17 Mflops/processor on the T3D. Some timing results for this linear solver are shown in Table 1.

We are currently in the process of developing robust parallel solvers for the more difficult matrices arising from the fully implicit compositional model.

Performance

The performance of Falcon has been measured primarily on two problem sets. The first set is a basic field simulation with two wells over a period of 2800 days including primary and secondary recovery. The second problem set is a very large field study problem with 2-1/4 million grid blocks and 1039 wells.

Figure 1 summarizes performance results for various versions of the code on the basic problem. On the T3D we tested two message passing libraries: the T3D PVM library, and the ACLMPL library developed at Los Alamos. We also tested the three versions of the linear solver. The seventh bar denotes the result from using an optimized version of COEF, the matrix coefficient generation routine. It is clear from these results that the linear solver is the primary cost of the simulation, and significant benefit can be gained from optimizing that routine. The COEF routine was second largest in cost. We believe it would be possible to optimize COEF and other routines more fully to make better use of cache; however, the effort to do so would be substantial.

Subgrid size	Processor grid	Mflop rate/PE
8x8x64	1x2x1	12.194
8x8x64	2x2x1	10.781
8x8x64	2x4x1	9.716
8x8x64	4x4x1	8.747
8x8x64	4x8x1	8.694
8x8x64	8x8x1	8.674
16x16x64	1x2x1	15.314
16x16x64	2x2x1	14.064
16x16x64	2x4x1	13.138
16x16x64	4x4x1	12.211
16x16x64	4x8x1	12.145
16x16x64	8x8x1	12.113
32x32x64	1x2x1	17.265
32x32x64	2x2x1	16.466
32x32x64	2x4x1	15.837
32x32x64	4x4x1	15.138
32x32x64	4x8x1	15.042
32x32x64	8x8x1	15.016
65x65x64	1x2x1	16.268
65x65x64	2x2x1	15.965
65x65x64	2x4x1	15.689
65x65x64	4x4x1	15.378
65x65x64	4x8x1	15.322
65x65x64	8x8x1	15.271

Table 1. Mflop rates per processor for optimized solver.

Figure 2 shows the scaling behavior of the basic dataset. All parts of the code scale well, even for a relatively small dataset on a large number of processors.

The Amoco Field Study

In late 1995 Amoco conducted a major field study, believed to be the largest of its kind ever performed, which involved 50 simulations on geostatistically generated datasets of 2-1/4 million grid cells with over 1000 wells. The typical simulation time required for a 30-year simulation was 4 hours.

Table 2 shows some representative timings required to perform a small number of days of simulation for one of these datasets.

	64 PE's	128 PE's	256 PE's
CM5	-	61	37
T3D/PVM	97	57	39
T3D/ACL	89	50	34

Table 2. Falcon runtime (secs.) for field study dataset.

Conclusions

Falcon's ability to effectively exploit parallel computing technology offers the possibility of redefining the nature of reservoir simulations, due to the tremendous increase in speed and problem size afforded by these machines.

Our experience with this code showed the difficulty in developing a code that is both high-performance and maintainable. We are hopeful that future software and compiler technologies will make this task easier.

Acknowledgements

The Falcon development team is composed of Olaf Lubeck, Wayne Joubert, Ken Koch (Los Alamos National Laboratory), Robert Stephenson, Gautam Shiralkar (Amoco Production Co.), and Bart van Bloemen Waanders (Cray Research Inc.).

This work was supported in part by the Department of Energy through grant W-7405-ENG-36, with Los Alamos National Laboratory. This research was performed in part using the resources located at the Advanced Computing Laboratory of Los Alamos National Laboratory, Los Alamos, NM 87545.

References

[1] Wayne Joubert, "Node Performance of the BLAS-1 Routines on the Cray T3D," CIC-19 Performance Brief, Los Alamos National Laboratory, September 1994.

[2] Wayne Joubert, "Port of Amoco Linear Solver Code to the Cray T3D," CIC-19 Performance Brief, Los Alamos National Laboratory, 1994.

[3] Richard Volz, Manny Vale, Bob Stephenson, Kirk Hird and Gautam Shiralkar, "Showcase Reservoir Study Using FALCON (Redefining Reservoir Simulation)," Amoco internal document, December 1995.

http://www.c3.lanl.gov/~wdj/amoco.html

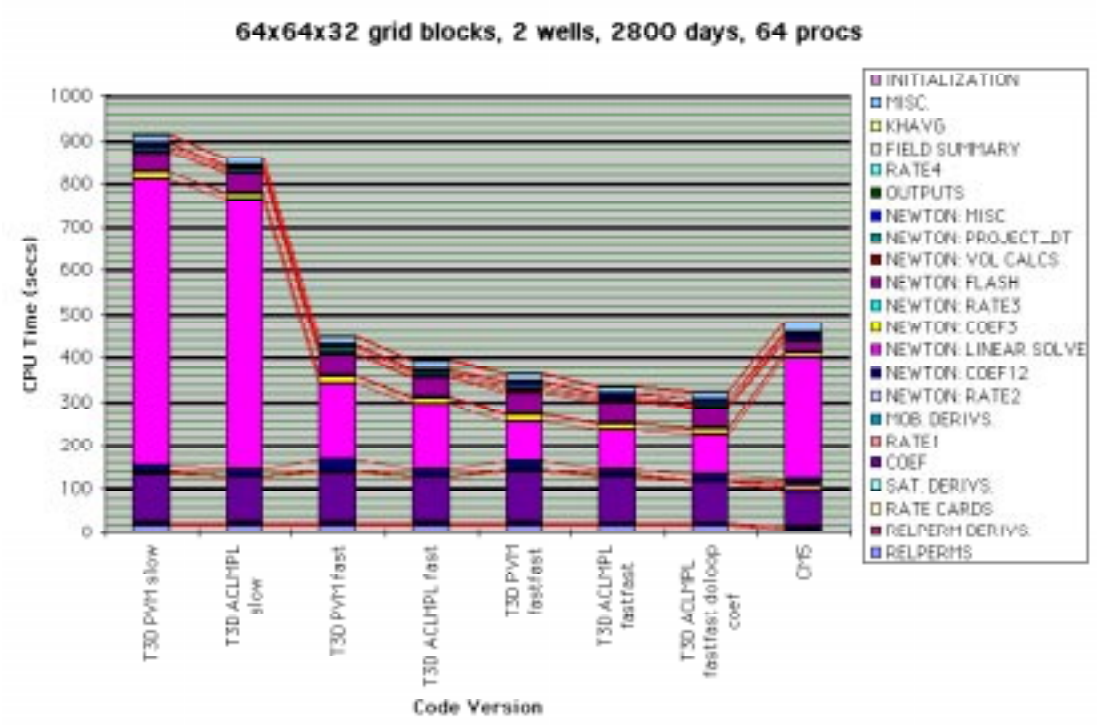

Figure 1: Performance of different code versions of Falcon on the T3D and CM5.