A High-Performance Computing Cluster for Parallel Simulation of Petroleum Reservoirs

This article examines compute-intensive benchmark tests performed on an Intel® Xeon™ processor–based Dell™ PowerEdge™ server cluster running the Red Hat® Linux® operating system. To determine how configuration factors affect a standards-based high-performance computing (HPC) cluster, the study explores the scalability of a parallel petroleum reservoir simulator when varying the network interconnect type, test case size, and configuration. Each test was run twice to compare the results of single-processor versus dual-processor nodes.

BY KAMY SEPEHRNOORI, PH.D.; BARIS GULER; TAU LENG, PH.D.; VICTOR MASHAYEKHI, PH.D.; AND REZA ROOHOLAMINI, PH.D.

Petroleum reservoir simulation requires detailed—and computationally intensive—geological and physical models. Traditionally, this work has been performed on supercomputers, mainframes, and powerful workstations. Now developers are experimenting with reservoir simulation on loosely coupled parallel systems known as clusters, enabled by recent hardware advances in the design of standardized processors, memory and I/O subsystems, storage, and network interconnects. Software developments bolstering the use of clusters include the rapid adoption of the Linux® operating system (OS), new compiler and math libraries, and the middleware Message Passing Interface (MPI) specification. Altogether, these advances are helping to make standards-based clusters a legitimate and cost-effective alternative to proprietary high-performance computing (HPC) systems for reservoir simulations.

This article is an update of a previous study in the ongoing research collaboration between the Center for Petroleum and Geosystems Engineering at The University of Texas at Austin and the Scalable Systems Group at Dell.1 Building upon the previous study, small and large simulation cases were developed to benchmark different cluster configurations using various network interconnects. The goal for the testing reported in this article was to understand how problem size affects simulator performance and to determine which

---

1 See “Parallel Simulation of Petroleum Reservoirs on High-Performance Clusters” by the Center for Petroleum and Geosystems Engineering, The University of Texas at Austin, in collaboration with Dell, in Dell Power Solutions, Issue 2, 2001.
network connections might be optimal for each simulation case. These simulations were carried out using both single-processor and dual-processor nodes to analyze symmetric multiprocessing (SMP) cluster performance.

Introducing reservoir simulation
Predicting the performance of petroleum reservoirs under a variety of operations is essential for reservoir management. Making oil recovery more productive through the injection of substances such as carbon dioxide, nitrogen, surfactant, and polymers is called Improved Oil Recovery (IOR). By conducting a series of reservoir simulations that address different IOR processes, researchers can assess the risk involved in each process to help determine the best possible recovery method for a given reservoir before it is implemented in the field.

A reservoir simulator consists of a coupled set of nonlinear partial differential equations and constitutive relations that describe the physical process occurring in a petroleum reservoir. The governing partial differential equations are solved using numerical methods. Black oil and compositional simulators are the most commonly used simulators. Black oil simulators use water, oil, and gas phases for modeling fluid flow in a reservoir, whereas compositional simulators use phases with different chemical species for modeling physical processes that occur in a reservoir. Compositional simulators are required in simulations that account for mixing fluids that have drastically different properties and for displacing oil by miscible fluids. Traditionally, black oil simulators have been used more often than compositional simulators because compositional simulators are more complicated and have intensive memory and CPU requirements.

However, advances in memory and CPU technologies in the last decade have enabled researchers to increase the use of compositional simulations. Measurement methods also have evolved, allowing researchers to use more data to characterize reservoirs. Current simulations require hundreds of thousands of cells or millions of unknowns; as simulation complexity increases, future simulations will call for millions of cells with multimillions of unknowns.

Parallel reservoir simulators have the potential to solve larger, more realistic problems than previously possible. Researchers at The University of Texas at Austin have been developing a new-generation code for parallel computers that is designed to perform fast, accurate, and efficient high-resolution simulations of fluid flow in permeable media. Their research scope involves development of new, complex physical and chemical models and accurate numerical methods implemented in a parallel-processing environment.

Predicting the performance of petroleum reservoirs under a variety of operations is essential for reservoir management.

The simulator
The governing equations for fluid flow in a permeable medium are implemented in a new, parallel, fully implicit compositional simulator called the General Purpose Adaptive Simulator (GPAS). A finite-difference method, which divides a continuous domain into small cells, is used to solve the governing partial differential equations. As the number of cells increases, more accurate results can be obtained; however, a greater number of cells increases the computation time. A fully implicit solution results in a system of nonlinear equations that are solved using Newton’s method. Numerical solution of nonlinear equations requires large, sparse linear systems of equations. The linear systems are handled with solvers from the Portable Extensible Toolkit for Scientific Computation (PETSc).

To handle the complicated tasks associated with parallel processing, researchers at The University of Texas have developed an Integrated Parallel Accurate Reservoir Simulator (IPARS) framework. The goal is to separate the physical model development from parallel processing. Communications between the simulator framework and a physical model are carried out through hooks provided within the IPARS. These hooks are composed of FORTRAN subroutine calls, and all the communications among processors for a physical model are performed in these routines. The physical model developers insert these calls into the codes that perform the corresponding tasks. IPARS provides input and output, memory management, domain decomposition, and message passing among processors to update overlapping regions.

Establishing the computational environment
The computational environment for the benchmarks in this study comprised a cluster of 64 Dell PowerEdge 2650 servers using Fast Ethernet, Gigabit Ethernet, and Myricom Myrinet interconnects.

---


4 This term indicates compliance with IEEE standard 802.3ab for Gigabit Ethernet, and does not connote actual operating speed of 1 Gbps. For high-speed transmission, connection to a Gigabit Ethernet server and network infrastructure is required.
Each PowerEdge 2650 had two Intel® Xeon™ processors running at 2.4 GHz with 512 KB of Level 2 (L2) cache and 2 GB of double data rate (DDR) RAM operating on a 400 MHz frontside bus (FSB).

The PowerEdge 2650 was equipped with Peripheral Component Interconnect Extended (PCI-X) slots capable of supporting the peak network traffic generated by the Myrinet network interface card (NIC). The operating system installed in the cluster was Red Hat® Linux 7.3 (kernel version 2.4.18-4smp). The GPAS was compiled using the PETSc library and the PGI® CDK™ (Cluster Development Kit) C/C++ and FORTRAN 77/90 compilers. Figure 1 shows the architectural stack of the computational environment.

**Simulation cases**

To evaluate the scalability of the GPAS using different interconnects on the same cluster, two simulation cases were developed. The first, a small simulation case with 3-D grid $16 \times 224 \times 8$ used approximately 350 MB of memory. The case comprised 28,672 grid blocks, requiring 229,376 unknowns to be solved at each time step in the process. A total of 10 time steps was taken, and each time step (ranging from 0.1 to 20 days) consisted of 2 to 5 Newton iterations. The first case simulated 100-day gas injection with one injection well and one production well. The reservoir had homogeneous permeability and porosity fields.

The second simulation case built upon the first, adding complexity by enlarging the reservoir, introducing heterogeneity, and increasing the number of injection and production wells. The reservoir in the larger simulation was divided into 197,120 ($77 \times 256 \times 10$) grid blocks, requiring 1.5 million unknowns to be solved simultaneously at each time step. To run the case, the GPAS required approximately 1.7 GB of memory.

**Performance metric**

In parallel applications, efficiency is usually measured by speedup. In this study, the speedup of a cluster with $N$ processors was defined as $speedup = \frac{t_1}{t_N}$, where $t_1$ is the execution time running on one processor and $t_N$ is the execution time running on $N$ processors. The ideal speedup of a parallel simulation with $N$ processors is $N$; that is, the program runs $N$ times faster. However, as the number of processors becomes larger, researchers usually observe a speedup of less than $N$. The performance reduction can be attributed to increased interprocessor communication (known as memory contention) or to network contention arising from a cluster whose nodes are SMP machines, or both. Such overhead is not encountered using only one processor. Sometimes the performance reduction can be caused by an inefficient program that does not decompose the application evenly.

**Analyzing the interconnect test results**

The study tested three different interconnects: Fast Ethernet, Gigabit Ethernet, and Myrinet. Researchers executed both simulation cases—small ($16 \times 224 \times 8$) and large ($77 \times 256 \times 10$)—to analyze the performance of the cluster using different interconnects. The initial test used only one processor per compute node. Figure 2 shows the execution times (bars) and speedups (lines) of the small case from 1 node to 32 nodes. The left y-axis in the figure indicates the execution time (in seconds), and the right y-axis indicates the performance speedup of the simulator. As shown in Figure 2, the
simulator performed and scaled best on the low-latency, high-bandwidth Myrinet cluster. In fact, the Myrinet cluster performance scaled linearly from 1 to 32 nodes regardless of the simulation size—the result of cache and memory aggregations when more than one node is used.

For configurations using few nodes, the clusters with high-latency Fast Ethernet or Gigabit Ethernet interconnects offered similar performance to the Myrinet cluster. As node count increased, communication among processes increased—while computation performed by each node decreased. The performance differences among the three types of interconnects became increasingly significant after scaling to as few as four nodes. As Figure 2 shows, the most efficient Gigabit Ethernet cluster was the 16-processor configuration, because this interconnect did not scale when the processor count went beyond 16.

Similar observations were made in the large simulation case, as shown in Figure 3. However, performance differences became significant after 16 nodes instead of 4 nodes as in the small simulation case. Because the communication-to-computation ratio of the large case was lower than the communication-to-computation ratio of the small case, the simulator showed better scalability for all interconnects in the large case. In the 64-node configuration, Myrinet provided the best performance and super linear speedup. In addition, both Fast Ethernet and Gigabit Ethernet clusters could be scaled beyond 32 nodes efficiently. This indicates that for reservoir simulation, the importance of the interconnect is truly case dependent; even Fast Ethernet might be a viable solution for large simulations.

**Single-processor and dual-processor node performance**

In addition to testing the three types of interconnects at two different problem sizes, researchers examined simulator performance using one processor per node and then using two processors per node. Two processes running in a node will compete for resources such as memory and I/O. In particular, the shared memory bus will be a performance bottleneck when the memory is accessed at the same time by both processes. In addition, the communication traffic generated by the two processes may create another potential bottleneck on I/O resources, such as the PCI bus or the NIC, or both.

Researchers could calculate the ratio of time spent in each section of the large simulation case based on the recorded data, using a dual-processor node configuration as compared to a single-processor node configuration. The runs were performed on the Myrinet cluster. Figure 4 shows the ratios of the plotted execution times. A ratio larger than 1.0 indicates some contention inherent to SMPs. For example, highly computational- and data-dependent sections of the simulation such as “Update viscosity and relperm” and “Update dependent variables” suffered the most from shared memory architecture.

All the ratios, except one, decrease as the number of processors increases, which indicates that memory contention diminished as more processors were used. Because the amount of data per processor was reduced as the number of processors increased, less tendency for contention existed when retrieving the data. Conversely, increased communication among processes resulted in network contention on the NIC. For the communication-sensitive sections such as the “Total linear solver time,” the ratio
is considerably larger—greater than 1.2—and increases with the processor count as shown in Figure 4.

Figures 5 and 6 show the performance speedup curves using dual-processor nodes for the small and large simulation cases and varying the interconnect. Overall, single-processor configurations (see Figures 2 and 3) outperformed dual-processor configurations anywhere from 2 percent to 27 percent. As seen in both Myrinet runs, the aggregated cache and memory improvements were offset by both memory contention and network contention; this performance degradation was caused by routing communication traffic for two processors through a single NIC. In the Myrinet cluster, the performance degradation was 18 percent and 27 percent for small and large cases, respectively. The Gigabit Ethernet cluster was the least affected by the use of SMPs as compute nodes, showing 9 percent performance degradation for the small case and 2 percent for the large. In fact, using 64 processors with 49.8 speedup, the Gigabit Ethernet cluster performed and scaled slightly better than the Myrinet cluster. Using SMPs affected Fast Ethernet cluster performance similarly for both small and large cases, exhibiting 13 percent performance degradation.

Configuring standards-based clusters to scale optimally
Examine both small and large simulation cases using Fast Ethernet, Gigabit Ethernet, and Myrinet cluster interconnects enabled conclusions about the scalability of the simulator. Additionally, comparing single-processor node to dual-processor node cluster performance led to the following observations:

• A low-latency, high-bandwidth interconnect, Myrinet performed best in all simulations.
• For the small case simulation, when relatively few computations were performed on each processor, low-latency interconnects provided better scalability.
• As the problem size increased, a high-bandwidth interconnect became more important than a low-latency interconnect to produce consistent scalability.
• The aggregations of cache and memory helped computational performance.
• I/O resource contention may introduce additional communication overhead in an SMP cluster.
• Memory contention that can arise in SMP nodes may contribute to overall performance degradation ranging from 2 percent to 27 percent when comparing dual-processor nodes to single-processor nodes.
• The Myrinet and Gigabit Ethernet clusters performed almost the same when using 32 dual-processor nodes (64 processors).

The results of this study and others like it can help system administrators make wise infrastructure choices. By determining the most effective network interconnects and number of processors per node for their specific computational problem sets, administrators can improve the scalability of clusters built using Dell PowerEdge servers running Linux—thereby leveraging inexpensive, standards-based HPC clusters as a cost-effective alternative to proprietary supercomputers, mainframes, and workstations for many computational purposes.
Acknowledgments
This research was supported in part by the U.S. Department of Energy, the sponsors of the Joint Industry Project for Reservoir Simulation at the Center for Petroleum and Geosystems Engineering at The University of Texas at Austin, and Dell Inc.

References


Baris Guler (baris_guler@dell.com) is a systems engineer and advisor in the Scalable Systems Group at Dell. His current research interests are parallel processing, diskless HPC clusters, performance benchmarking, reservoir engineering and simulation, and numerical methods. Baris has a B.S. in Petroleum and Natural Gas Engineering (PNGE) from the Middle East Technical University in Turkey and an M.S. in PNGE from Pennsylvania State University. He is currently a Ph.D. candidate in Petroleum and Geosystems Engineering at The University of Texas at Austin.

Tao Long, Ph.D. (tao_long@dell.com) is an engineer manager of the Scalable Systems Group at Dell. His current research interests are parallel processing, distributed computing systems, compiler optimization, and performance benchmarking. Tao has a B.S. in Mathematics from the Fu Jen Catholic University in Taiwan, an M.S. in Computer Science from Utah State University, and a Ph.D. in Computer Science from the University of Houston.

Victor Mashayekhi, Ph.D. (victor_mashayekhi@dell.com) is a senior technical manager of the Scalable Systems Group at Dell. His current research interests are distributed systems, database management systems, computer-supported cooperative work, concurrent software engineering, multimedia systems, clustering, and interconnect technologies. Victor has a B.A., M.S., and Ph.D. in Computer Science from the University of Minnesota.

Reza Rooholamini, Ph.D. (reza_rooholamini@dell.com) is the director of the Enterprise Solutions Engineering Group at Dell, which develops Linux and cluster products. He has a B.S. in Electrical Engineering from the University of Illinois at Urbana, an M.S. in Electrical Engineering and an M.S. in Computer Science from the University of Wisconsin, and a Ph.D. in Computer Science/Engineering from the University of Minnesota. Reza has over thirty publications in areas of his research interest, including distributed systems, multimedia systems, HPC computing, storage systems, high-availability clustering, and interconnects.

FOR MORE INFORMATION
Center for Petroleum and Geosystems Engineering at The University of Texas at Austin: http://www.cpge.utexas.edu
Myricom Myrinet products: http://www.myri.com
Dell HPCC products: http://www.dell.com/hpcc

Kamy Sepehrnoori, Ph.D. (kamy@mail.utexas.edu) is the Bank of America Centennial Professor in the Department of Petroleum and Geosystems Engineering at The University of Texas at Austin. His research interests include computational methods, reservoir simulation, parallel computations, applied mathematics, and enhanced oil recovery modeling. He has a B.S. in Mechanical Engineering, an M.S. in Aerospace Engineering, and a Ph.D. in Petroleum Engineering from The University of Texas at Austin.