Simulating 3D HF in a Parallel Computing Environment

Bruce J. Carter, Anthony R. Ingraffea, Gerd Heber
Cornell Fracture Group/Computational Materials Institute
Cornell Theory Center, Cornell University, Ithaca, NY

Three Dimensional and Advanced Hydraulic Fracture Workshop
Pacific Rocks 2000: Rock Around the Rim
Fourth North American Rock Mechanics Symposium
Seattle, Washington, USA
July 29, 2000

Summary

Three-dimensional hydraulic fracture (HF) models have been under development since the late 1970's. The Cornell Fracture Group (CFG) in collaberation with Schlumberger-Dowell and Schlumberger Cambridge Research worked on the development of a fully-3D HF simulator from 1988 through 1996. This software, called HYFRANC3D, has seen limited use in the petroleum industry primarily because of the significant time and effort required to simulate the growth of a fully-3D HF. However, increasing computer speeds and decreasing computer costs have made the use of 3D models more practical. Parallel computing platforms are aiding this process by drastically reducing the time-to-solution. The Cornell Theory Center (CTC) now has two industry standard Dell/Intel PC clusters running the Microsoft™ Windows 2000 operating system. The authors are part of a group of researchers from computer science, physics, and engineering who are taking advantage of these clusters to perform large scale 3D finite element based fracture simulations. The goal of the project is to perform a thousand steps of crack growth, involving a million degrees of freedon (DOFS) per step, in an hour. The long term goal is to extend the general framework to specific applications such as HF.

Introduction

Parallel computers are not required for 3D hydraulic fracture (HF) simulation, but they sure are nice to have. Multiple processors, with their own memory, allow much larger problems to be solved much faster by avoiding disk access and by simultaneously solving many small parts of the whole model. Parallel computers, in the past, were specially designed, large, expensive machines. Today, however, a relatively inexpensive parallel computer can be built from a cluster of industry standard PC's. The Cornell Theory

Center (CTC) has shifted from an IBM SP2 to a cluster of Dell/Intel PC's and the Cornell Fracture Group (CFG) software is being ported to this new environment.

The CFG[19] in collaberation with Schlumberger[30] worked on a serial version of a fully-3D HF simulator from 1988 through 1996. This software, called HYFRANC3D and available from the CFG, is based on a general purpose 3D fracture analysis software package that includes OSM, FRANC3D, and BES. OSM is a program for building geometric models. FRANC3D is a pre- and post-processor for simulating crack growth. BES is a linear elastic boundary element code. HYFRANC3D builds upon FRANC3D by adding a module for simulating fluid flow in the 3D fractures.

The software has seen limited use in the petroleum industry primarily because of the significant time and effort required to model a fully-3D HF. Most HF simulators used in industry are either 2D or psuedo-3D (planar 3D cracks). This is because the time-to-solution for these simulators is seconds or minutes, which allows for active control of a HF stimulation in the field. A fully-3D, linear elastic, boundary element solution of a complex geometry with one or more abitrarily shaped cracks can take hours (or days) on a single computer processor using BES. A BES analysis is required for each step of crack growth, meaning that a reasonable number of crack growth steps (20 to 30) could take weeks. That is just too much time, even in a research environment.

To decrease the time-to-solution, BES was ported to the CTC[10] IBM SP2 in 1996[1]. The time-to-solution decreased significantly, as a single analysis could be obtained in an hour or less using 32 to 64 processors. However, very few companies have a large IBM SP2 in their engineering field offices meaning that the HYFRANC3D software was still only useful as a research tool. BES now has been ported to the CTC's new Dell/Intel PC cluster, which makes parallel computations more accessible. To further reduce computing time, parallel FE simulations are being explored.

In 1998, the NSF funded the CCISE: Crack Propagation on Teraflop Computers[9] project at Cornell. This project combines researchers from computer science, physics, and engineering with a common goal of simulating 3D fracture growth very quickly. The project is based on the development of several 3D finite element mesh generators and several iterative solvers in a parallel computing environment. The initial software was developed on the CTC IBM SP2. However, it now has been ported to the new cluster of PC's. Although this software is not yet ready to solve 3D HF simulations, it has the potential to make fully 3D HF simulation widely accessible.

The HYFRANC3D software is briefly described along with the past and present computing environments at the CTC. The recent progress in 3D finite element based fracture simulation is then described along with comparisons of the past and present parallel computing platforms and software. Extension to HF simulation lies ahead.

HYFRANC3D and Accompanying Software

The FRANC3D software is described by Carter et al[2]. The HYFRANC3D software is described by Carter et al[3]. A brief overview of the numerical algorithm is provided here. Although, the fluid flow equations are not the central theme, the flow is tightly coupled to the elastic solution and the manner in which the elastic solution is obtained. The parallel implementation of the fluid flow equations should follow the parallelization of the more time consuming elastic solution.

The HF governing equations consist of linear elasticity, lubrication theory for flow, and mass conservation. The boundary conditions include far-field stresses, fluid flux in and out of the crack, zero crack aperature at the crack front, and the requirement that the fluid velocity at the crack front equals the crack speed. The elasticity equation relates the fracture width to the fluid pressure:

$$w(x) = \frac{4}{\pi E'} \int_0^L \log_e \left[\frac{(L^2 - x^2)^{1/2} + (L^2 - y^2)^{1/2}}{(L^2 - x^2)^{1/2} - (L^2 - y^2)^{1/2}} \right] [p(y) - \sigma_3] dy$$

The lubrication equation relates the pressure gradient and fracture width with the fluid flux:

$$q = -\frac{n}{2n+1} \left(\frac{1}{2^{n+1} K'} \right)^{1/n} \left(\frac{\partial p}{\partial x} \right)^{1/n} w^{(2n+1)/n}$$

The continuity equation imposes mass conservation and implies that the fluid volume in the fracture should equal the fracture volume:

$$\frac{\partial w}{\partial t} + \frac{\partial q}{\partial x} = 0$$

q is the fluid flux
w is the width of the fracture
L is the fracture half length
p is the fluid pressure in the fracture
n and K are the fluid viscosity parameters
$E' = E/(1-\nu)$ is the effective Young's modulus
E is the Young's modulus
ν is Poisson's ratio
x is the distance along the fracture from the fluid inlet
σ_3 is the far-field stress normal to the fracture
V is the crack front speed

Combining these equations gives:

$$p-\sigma_3 = p_h\left[\left(\frac{2\sqrt{2}(2+n')}{\pi(2-n')}\right)\left(\frac{L_h}{L}\right)^{n'/(2+n')} - \left(\frac{L_h}{\xi}\right)^{n'/(2+n')}\right]$$

$$w = \xi^{2/(2+n')} L_h^{n'/(2+n')}\left[c_1(n') - c_2(n')\left(\frac{\xi}{L}\right)^{\frac{2+3n'}{4+2n'}}\right]$$

where

$$L_h = V\left(\frac{K'}{E'}\right)^{1/n'}$$

$$p_h = E'\left[\left(\frac{\cos((1-\alpha)\pi)}{\sin(\alpha\pi)}\right)^{1+n'}\left(\frac{2n'+1}{n'^2(2+n')}\right)^{n'}\right]^{1/(2+n')}$$

$$\alpha(n') = 2/(2+n')$$

$$\xi = L - x$$

For a Newtonian fluid, n=1 and the pressure has a 1/3 singularity. The stress ahead of the crack has the same order of singularity. The solution has been dubbed LEHF to differentiate it from standard linear elastic fracture mechanics (LEFM). Physically, the pressure cannot be singular, which implies that a fluid lag exists at the crack front.

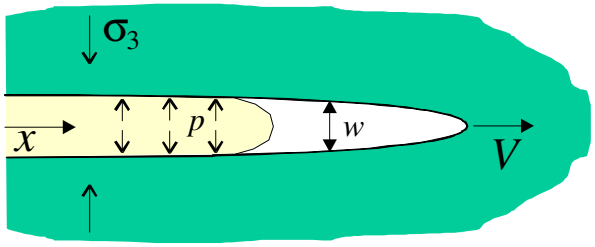

Figure 1. Crack front region with fluid lag.

For a finite element formulation, the near-crack front behavior is captured by special crack front elements where the analytical equations hold. The crack front element captures the zone, which includes the fluid lag zone, between the zone where the lubrication equation holds and the immediate crack tip zone where LEFM holds. This avoids having a large number of elements near the crack front and assumes that the behavior is dominated by the "intermediate" region. The assumptions are that there is no sizeable effect of fracture toughness, there is a small fluid lag, there is a sharp pressure drop at the crack front, and there is no effect of pore pressure. The bulk of the fracture is

modeled using collapsed solid elements with the lubrication approximation. The fluid flow elements correspond exactly to the BE elements. Thus, limiting the number of elements near the crack front also reduces the elastic solution time.

The finite element formulation leads to a set of n fluid flow equations and n structural equations with n unknown nodal widths and n unknown nodal fluid pressures. The resulting system of equations is given by:

$$\sum_{j=1}^{j=n} \frac{\hat{w}_j}{\Delta t} \int_{\Omega_i} N_i N_j \, d\Omega + \sum_{j=1}^{j=n} \hat{p}_j \int_{\Omega_i} \frac{1}{12\mu} \left(\sum_{k=1}^{k=n} N_k w_k \right)^3 \text{grad } N_i \cdot \text{grad } N_j \, d\Omega =$$

$$\sum_{j=1}^{j=n} \frac{\hat{w}_j(t_n)}{\Delta t} \int_{\Omega_i} N_i N_j \, d\Omega - \int_{\Gamma_i} N_i \left(\sum_{k=1}^{l} M_k \hat{\beta}_k \hat{V}_k^{4/3} \right) d\Gamma + Q(t) N_i(O) +$$

$$\int_{\Omega_i} \frac{1}{12\mu} \left(\sum_k N_k \hat{w}_k \right)^3 \text{grad } N_i \cdot \text{grad} \left(\sum_{j=m+1}^{n} N_j \hat{B}_j \hat{V}_j^{1/3} \right) d\Omega$$

where

$$\hat{B}_j = E'^{2/3} \left(\frac{\mu}{3\hat{\rho}_j} \right)^{1/3} \quad \text{and} \quad \beta = (2)\left(3^{7/6}\right)\left(\frac{\mu}{E'} \right)^{1/3} \rho^{2/3}$$

These latter terms account for the fluid flow in the special crack front elements. The relationship between fluid pressure and crack aperature is provided by an elastic influence matrix that is generated by BES. The final solution is a set of crack surface displacements that can be corellated to the stress intensity factors to determine the crack advance and resulting new crack shape.

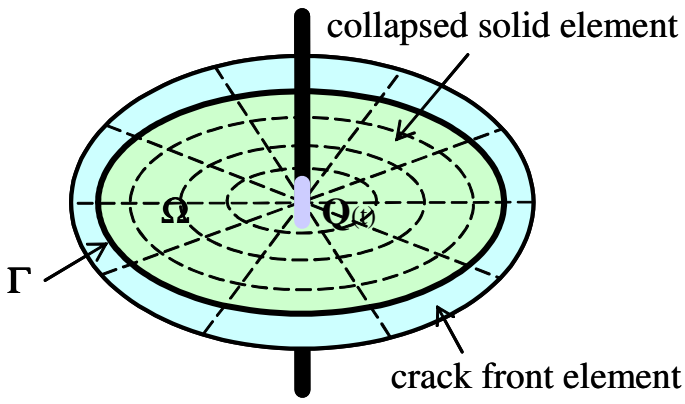

Figure 2. Finite element mesh of a radial crack from a wellbore.

The fluid flow analysis relies upon an elastic influence matrix that relates the displacements and tractions everywhere in the model to an applied unit pressure at each of the nodes on the crack surface. Historically, the CFG has used BES to generate this matrix by solving for multiple right hand sides. The elastic analysis and the generation of the influence matrix is the most time consuming and computationally expensive portion of a HF simulation. Therefore, this is the part of the code that has been parallelized first.

Parallel Computing Environment at the Cornell Theory Center

The CTC was a NSF National Supercomputer Center from 1985-1998. The parallel computing platform during that era was a massively parallel IBM Scalable POWERparallel Systems SP2. The current SP2 configuration has only 32, 120MHz POWER2 Super Chips (P2SC) thin nodes, with 256 MB RAM per node and TB3 switching fabric (150 MB/s peak hardware bandwidth), each running AIX 4.2.1. This system will be phased out later this year.

In August, 1999, the CTC installed a cluster of 64 Dell PowerEdge servers[5], each with four Intel[28] Pentium® III Xeon 500 Mhz processors and running the Microsoft[26] Windows® NT operating system. Dubbed AC3 Velocity (V1), each server has 2 MB of L2 cache per processor, 4 GB of RAM and 54 GB of hard disk space with TCP and Giganet interconnects. Due to its success, and due to increasing demands, a second cluster was installed in April, 2000. Dubbed Velocity+ (V+), the new cluster consists of 64 dual Pentium III 733 Mhz processors, with 256 KB L2 cache, 2 GB of RAM per node, 27 GB of hard disk space, and full 64-way Giganet[6] interconnect, running Microsoft Windows® 2000 Advanced Server.

The move to cluster computing is a dramatic shift from the IBM SP2. The AC3 cluster is designed to illustrate the feasibilty and performance of a cluster of industry standard computer parts. Processors and network switches have evolved to a point where it is possible to build inexpensive, high speed, low latency computer clusters that rival the much more expensive, specially designed, parallel architectures. Finally, new software, including the release of the new Microsoft Windows® 2000 operating system and robust middleware such MPI Pro[7] for message passing has led to a stable software environment that is uniform from desktop to cluster. This enables software development on an inexpensive desktop PC and immediate scalability on the cluster. Of course, one of the drawbacks is that large amounts of software developed on the AIX operating system of the SP2 had to be ported to the Microsoft Windows operating system. This has proven to be less difficult than first imagined, but is not completed yet.

Parallel BEM Software

The boundary element code, BES, was initially ported to the IBM SP2 in 1996 and has recently been ported to the Velocity cluster. The software is a mixture of C and Fortran

and uses MPI for message passing. There are two phases to a BE solution, integration and solution. The model is discretized and integration is performed at collocation points (usually at nodes). The collocations can be evenly distributed amongst a set of processors as the integrations can be performed independently making this phase embarassingly parallel. The result of the integrations is a dense unsymmetric matrix. A QR solver is used to solve the system of equations. Although, the solver does operate in parallel it is not as efficient as the integration phase. Figure 3 shows the parallel speed up of both components on the IBM SP2. A 97% efficiency is achieved with an almost linear scaling for the integrations, but the solver does not scale nearly as well, especially for large problems.

It is possible to use other solvers; in fact, the serial version of BES employs a Gauss elimination direct solver and an iterative solver with a variety of preconditioners. The iterative solver is more efficient for most problems than either of the direct solvers. However, the iterative solver has not been extended to the parallel version. The reason is that iterative solvers are not well suited for the multiple right hand sides that HF simulations require. This issue needs to be addressed, especially if the FE software described next is extended to HF.

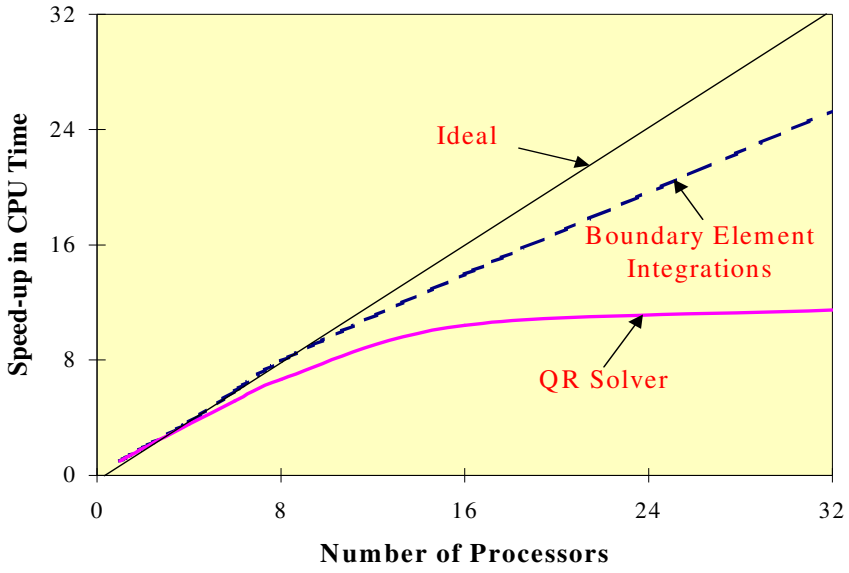

Figure 3. Parallel speed up of BES on the IBM SP2.

Parallel FEM Software

The CFG and a team of computer scientists and physiscists are working on the design and implementation of a highly parallel finite element code for 3D fracture analysis. The main challenges in implementing such a code are the following.

1) The simulation requires the solution of many finite-element problems (each crack step) involving on the order of a million degrees of freedom (DOF). In contrast to

matrices arising from the use of boundary elements (BE), finite-element (FE) matrices are sparse.

2) As cracks grow, the problem geometry and sometimes the topology changes.
3) Because of evolving geometry, repeated volume meshing is necessary. Since it is not practical to manipulate large numbers of meshes by hand, the meshes must come with certain quality guarantees.

Figure 4 shows a flow diagram of the code. During pre-processing, a solid model is created, problem-specific boundary conditions (displacements, tractions, etc.) are imposed, and flaws (cracks) are introduced. In the next step, a volume mesh is created, and linear elasticity equations for the displacements are formulated and solved. An error estimator is used to determine whether the desired accuracy has been reached, or whether refinement is necessary. Once the solution has converged, stress intensities are extracted, and the crack is advanced for one time step. The entire process must be repeated for a number of time (crack growth) steps. Finally, the results are fed back into a fracture analysis tool for post-processing and life prediction.

The most time consuming module in this code is the sparse linear system solver. One can choose between iterative and direct methods. For the iterative solvers, one can select from a variety of preconditioners (ICC, EBE, SPAI[32] etc.). The modules for mesh generation (QMG[25], DMESH[31], and JMESH[19]), finite-element formulation, preconditioning, and h/p-adaptation are implemented almost exclusively in C and C++ with MPI for message passing. A variety of third party packages for solving and mesh partitioning are used, including BLAS/LAPACK[12], PETSc[13], BlockSolve95[13], ParMETIS[15], PSPASES[14], and JANUS[33].

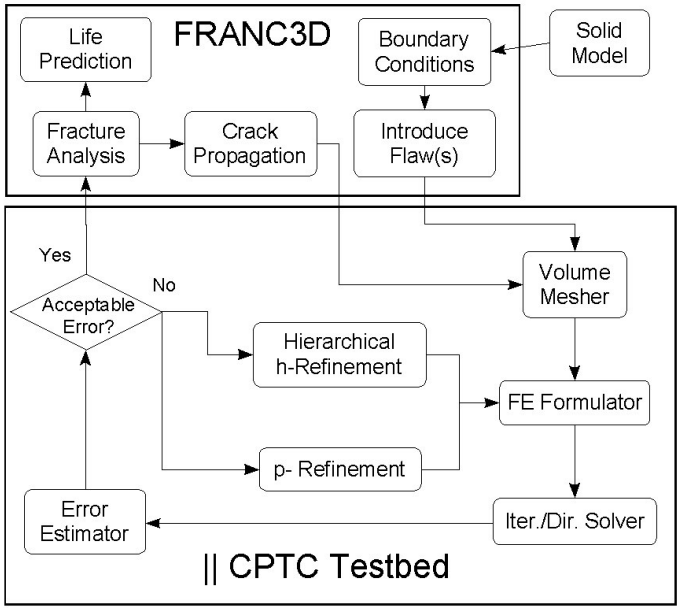

Figure 4: Block Diagram of a finite-element crack growth simulation.

Example Simulations and Parallel Performance

Two example simulations illustrate the parallel performance of both the BE and FE solution techniques. The first example shows the decreased time-to-solution for parallel compared to serial BE simulations. It also shows a further decrease in solution time for FE compared to BE simulations. The second example better illustrates the performance of the Velocity cluster compared to the IBM SP2 using only the FE solver.

Example 1

The first example is a half-symmetry model of a vertical wellbore with a crack aligned with the borehole axis. This is a typical tutorial HF example; the model is shown in Figure 5. A BE surface mesh is shown in Figure 6; this mesh is rather coarse, but is suitable for a tutorial.

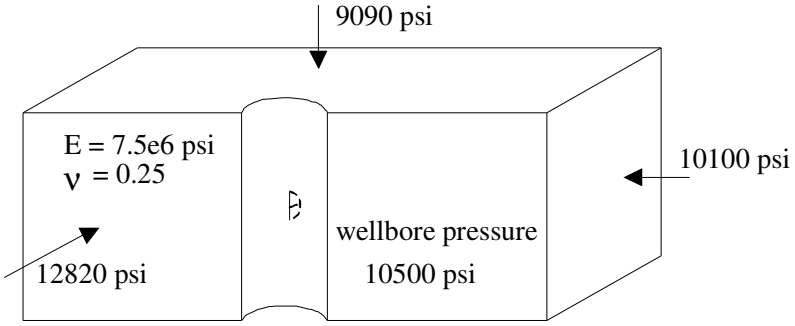

Figure 5. Wellbore model and boundary conditions.

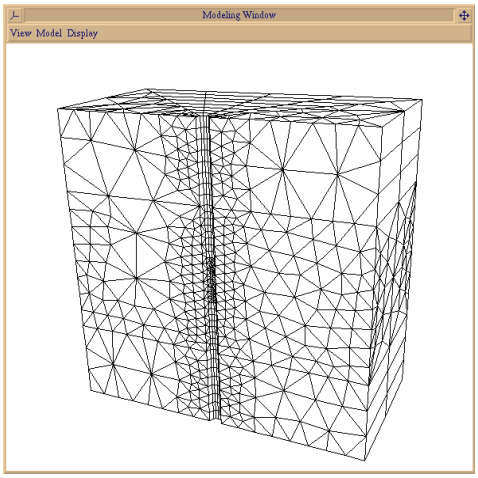

Figure 6. BE surface mesh of wellbore model.

A HF simulation normally involves both an elastic BE analysis and a fluid flow analysis in HYFRANC3D. The results from HYFRANC3D include crack opening displacements (COD), fluid pressures, and flow rates. Figure 7 shows the COD and flow rate arrows for the first and second flow analyses from the tutorial. Since, the elastic solution time is generally much greater than the flow analysis time, the examples concentrate on the elastic solution times rather than the flow simulations.

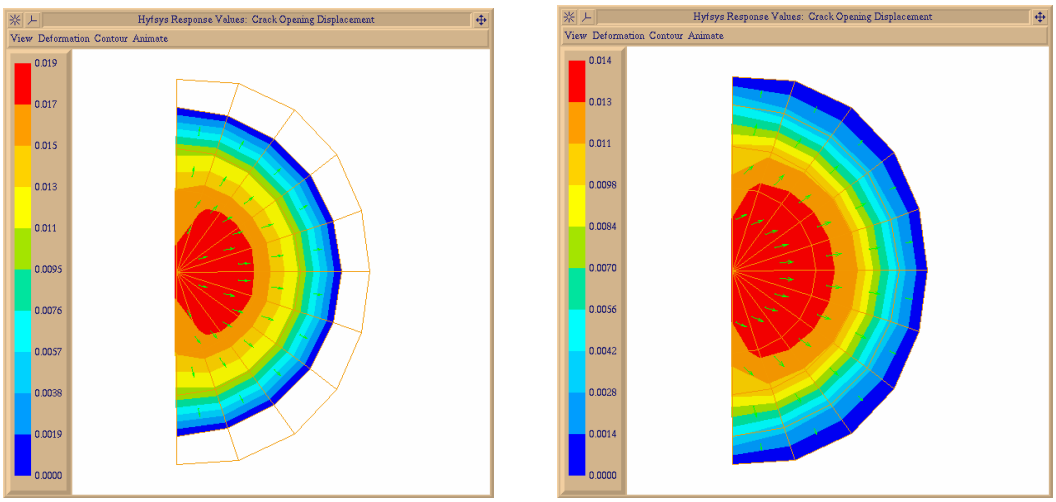

Figure 7. COD contours with flow rate arrows for the first two flow analyses.

Elastic solution times are provided in Table 1 for two mesh densities for the initial crack model only. The solution times for serial and parallel BE analyses, including the solution of multiple right hand sides are compared. The parallel solution times on both the SP2 and the V+ cluster are reported. Only one processor per node of the V+ cluster is used and TCP (100 MB/s Fast Ethernet) is used for communication. The FE analyses do not include multiple rhs and their contribution to the FE time-to-solution is unknown.

Table 1: Time-to-solution for the initial crack model of a wellbore.

Wellbore Model	1658 BE 2541 dofs	1658 BE + 26 rhs 2567 dofs	4038 BE 6171 dofs	4038 BE + 120 rhs 6291 dofs	14762 FE 62622 dofs	35656 FE 151312 dofs
V+ (1x1)	14 min	14 min	126 min	141 min	4 min	12 min
V+ (2x1)	7 min	7 min	68 min	77 min	----	6 min
V+ (4x1)	----	5 min	41 min	48 min	----	3 min
V+ (8x1)	----	4 min	26 min	32 min	----	1.65 min
SP2 (2)	----	13 min	----	NEM*	----	----
SP2 (4)	----	8 min	----	66 min	----	----
SP2 (8)	----	6 min	----	44 min	----	----

*NEM: not enough memory

The results show a reasonable speedup from 1 to 8 processors for the BE analyses on both the SP2 and the V+ cluster. The cost of solving for multiple right hand sides is significant only for larger models where the number of rhs is large. The V+ cluster is clearly faster than the relatively old SP2. A further decrease in time-to-solution is expected for the cluster when using the Giganet switch for communication, especially for larger models. The FE meshes are created using JMESH, an advancing front tetrahedral mesher, starting from the BE surface meshes. The FE solution time is significantly less than the BE solution time for similar accuracy in displacements. To compare the solution accuracy, the stress intensity factors (SIF) along the crack front are compared for the refined BE and FE meshes under static pressure loading. The average mode I SIF for the middle portion of the crack is 1390 and 1470 psi$\sqrt{\text{in}}$ for the BE and FE models respectively, a difference of about 5%. The value at the exact center differs by only 0.8%.

Example 2

The second example is based on grout pressure control in a cracked concrete dam[4]. A model of the doubly-curved concrete arch dam with cracks on the up-stream face is shown in Figure 8. Cement grout is to be pumped into the crack to repair it without causing further crack growth by over-pressurization. This model is used here as a test case for the parallel FE solver; thus, the loading is static and no rhs are required.

On the V1 cluster, there are 4 processors in each node. It is possible to use only a subset of the processors in a given node. In the illustrations below, the notations V1 (#x1), V1 (#x2), and V1 (#x4) refer to experiments where 1, 2, and 4 processors per node are used, respectively.

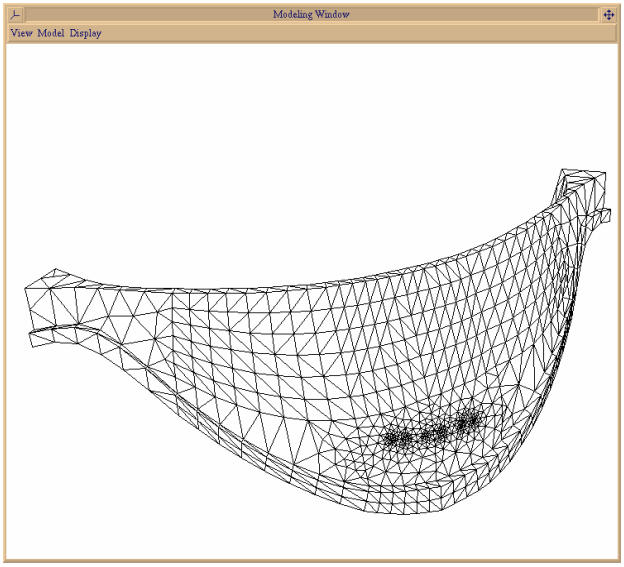

Figure 8. Doubly-curved concrete arch dam.

Table 2 shows the timing results for the FE solution of the initial crack model on the SP2 and the V1 cluster. All execution times are in seconds; note that the total time includes the time for formulation and assembly of the matrix, which is not shown here, but is generally less than 10% of the total time. Figure 9 shows the relative speed up on both computers as the number of procesors increases. The V1 cluster delivers better scalability than the SP2 for this FE solver. Furthermore, for large processor numbers, the V1 cluster delivers better absolute performance. There are two reasons for this: (i) the processors on the V1 are a generation ahead of the processors on the relatively old SP2, and (ii) the Giganet switch provides lower latency than the SP2's TB3 fabric. When the number of processors is large, these factors become important because the cost of performing global reductions in the conjugate gradient code dominates performance.

Table 2: FE execution times for the initial crack model of a dam, with 86,325 tetrahedral elements and 401,124 dofs.

Concrete Dam Model	**# Processors**	**# Iterations**	**Time/Iteration (s)**	**Total Time (s)**
SP2	16	3549	0.48	1772.67
V1 (4x4)	16	3372	0.53	1821.11
V1 (8x2)	16	3485	0.43	1534.14
V1 (16x1)	16	3719	0.39	1483.42
SP2	32	3374	0.27	948.54
V1 (8x4)	32	3548	0.27	975.42
V1 (16x2)	32	3549	0.22	797.52
V1 (32x1)	32	3556	0.20	728.03
SP2	64	3548	0.16	594.11
V1 (16x4)	64	3539	0.15	540.89
V1 (32x2)	64	3548	0.12	437.71
V1 (64x1)	64	3540	0.12	433.80

Conclusions

Parallel computing is becoming more widely accessable due to the development of industry standard PC-based clusters that now rival specially designed parallel architectures in terms of speed and performance while remaining relatively inexpensive.

HF simulation in 3D requires parallel computations to decrease the time-to-solution. The CFG has a long history in the use of boundary elements for simulating crack growth. The parallel implementation of the 3D BE software dramatically decreases the time-to-solution. However, the 3D FE technology has more potential in terms of modeling a wider class of problems with faster solution techniques. FE based fracture simulation

using iterative solvers shows almost linear scaling in parallel and holds great promise for further decreasing the time-to-solution for HF simulation.

A simple tutorial HF example takes about 3 hours per step of crack growth using the BE code for stress analysis on a single 400 MHz processor. This time-to-solution is reduced to 32 minutes on 8 processors of the V+ cluster. This time is expected to decrease once the solver is tuned for the new architecture. The parallel FE solution time is an order of magnitude less, requiring less than 2 minutes. The FE solution does not include multiple right hand sides, however, and is an issue that still needs to be adressed.

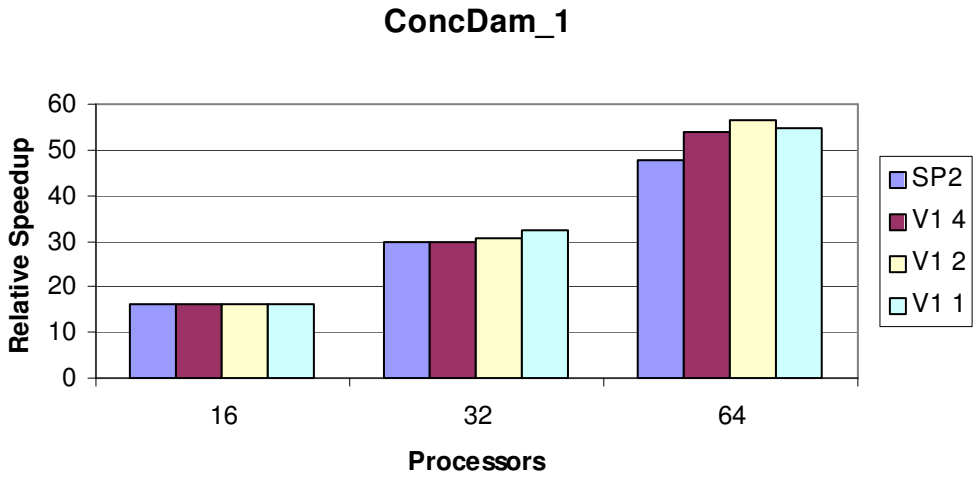

Figure 9: Relative speedup of the FE solver on the SP2 and V1 cluster.

Acknowledgments

The authors would like to acknowledge the past and present financial support of NSF (EIA-9972853; EIA-9726388; CMS-9625406) and Schlumberger. This research was conducted using the resources of the Cornell Theory Center, which receives funding from Cornell University, New York State, federal agencies, and corporate partners. The intellectual support of the other members of the CCISE/CPTC and CFG teams is also gratefully acknowledged.

References

1. Shah, K.R., B.J. Carter and A.R. Ingraffea,(1997) Hydraulic fracturing simulation in parallel computing environment, Int. J. Rock Mech. & Min. Sci., Vol 34, No. 3-4, Paper No. 282, presented at the 36th US Rock Mechanics Symposium, Columbia University, New York, July, 1997.
2. Carter, B.J., Wawrzynek, P.A., Ingraffea, A.R., (2000), Automated 3D Crack Growth Simulation, Gallagher Special Issue of Int. J. Num. Meth. Engng., Vol 47, pp. 229-253.
3. Carter, B.J, Desroches, J., Ingraffea, A.R. and Wawrzynek, P.A. (2000) Simulating Fully 3D Hydraulic Fracturing, in "Modeling in Geomechanics" Ed. Zaman, Booker, and Gioda, Wiley Publishers, 730p.
4. Ingraffea, A.R., Carter, B.J. and Wawrzynek, P.A., 1995, Application of computational fracture mechanics to repair of large concrete structures, FRAMCOS Vol 3, Proc. 2nd Int. Conf. Fracture Mechanics of Concrete Strucutres, Zurich, Switzerland, p.1721-1734.
5. http://www.dell.com/us/en/hied/topics/vectors_1998-wpvia.htm
6. http://www.giganet.com/
7. http://www.mpi-softtech.com/
8. http://www.mpi-softtech.com/performance/perf-win-lin.html
9. http://www.tc.cornell.edu/Research/CrackProp/
10. http://www.tc.cornell.edu/
11. http://www.tc.cornell.edu/UserDoc/Cluster
12. http://www.netlib.org/
13. http://www-fp.mcs.anl.gov/division/software/
14. http://www-users.cs.umn.edu/~mjoshi/pspases/
15. http://www-users.cs.umn.edu/~karypis/metis/
16. http://www.mingw.org/
17. http://www.tc.cornell.edu/~glc5/mpidc/
18. http://www.tc.cornell.edu/
19. http://www.cfg.cornell.edu/
20. http://wwwwissrech.iam.uni-bonn.de/research/projects/parnass2/
21. http://www.cs.sandia.gov/cplant/
22. http://www.cacr.caltech.edu/beowulf/index.html
23. http://www.python.org/

24. http://sourceware.cygnus.com/cygwin/
25. http://www.cs.cornell.edu/home/vavasis/qmg-home.html
26. http://www.microsoft.com
27. http://www.compaq.com/fortran/
28. http://www.intel.com
29. http://www.intel.com/vtune/compilers/fortran/
30. http://www.slb.com
31. http://www.cs.cornell.edu/Info/People/chew/chew.html
32. http://www.sam.math.ethz.ch/~grote/spai/
33. http://www.first.gmd.de/~jens/janus/