

# **SAND REPORT**

SAND2002-1371  
Unlimited Release  
Printed May 2002

## **Experiences with FETI-DP in a Production Level Finite Element Application**

Kendall H. Pierson, Garth M. Reese, Manoj K. Bhardwaj, Timothy F. Walsh and David M. Day

Prepared by  
Sandia National Laboratories  
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

**NOTICE:** This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from  
U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831

Telephone: (865)576-8401  
Facsimile: (865)576-5728  
E-Mail: [reports@adonis.osti.gov](mailto:reports@adonis.osti.gov)  
Online ordering: <http://www.doe.gov/bridge>

Available to the public from  
U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Rd  
Springfield, VA 22161

Telephone: (800)553-6847  
Facsimile: (703)605-6900  
E-Mail: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
Online order: <http://www.ntis.gov/ordering.htm>



SAND2002-1371  
Unlimited Release  
Printed May 2002

# Experiences with FETI-DP in a Production Level Finite Element Application

Kendall H. Pierson, Garth M. Reese, Manoj K. Bhardwaj and Timothy F. Walsh  
Computational Solid Mechanics & Structural Dynamics Dept.

David M. Day  
Computational Math and Algorithms Dept.

Sandia National Laboratories  
P. O. Box 5800  
Albuquerque, New Mexico 87185-0847

Padma Raghavan  
The Pennsylvania State University  
Pennsylvania State University  
0308 Pond Laboratories  
University Park, PA 16802

## Abstract

We discuss application of the FETI-DP linear solver within the Salinas finite element application. An overview of Salinas and of the FETI-DP solver is presented. We discuss scalability of the software on ASCI-red, Cplant and ASCI-white. Options for solution of the coarse grid problem that results from the FETI problem are evaluated. The finite element software and solver are seen to be numerically and cpu scalable on each of these platforms. In addition, the software is very robust and can be used on a large variety of finite element models.

# 1 Introduction

The need for predictive, qualified models of very complex structures drives the requirements for large scale structural analysis. Reduced testing in the nuclear weapons program is a driving factor at the DOE labs. In addition, more detailed models reduce the need for engineering approximation, improve accuracy and often simplify model construction. Uncertainty in model parameters (for example, variations in joint preloads) can require multiple analyses for evaluation of a structure. Salinas was designed to meet the needs of a very large scale, general purpose, structural dynamics analysis ([1] and [14]).

Salinas is implemented with the goal of providing predictive modeling of complex structural systems. This necessitates a full suite of structural elements and solution methods that must perform reliably on serial and distributed memory machines. Robust solution methods and platform portability are critical. Sensitivity analysis and optimization capabilities are also required for application to the design and uncertainty quantification communities.

Salinas is implemented on a variety of Unix<sup>(TM)</sup> and Unix-like platforms. The core libraries are written in C++ using MPI communications. This facilitates extensibility to a full range of solvers, solution methods and element libraries. Scalability to thousands of processors is achieved through application of Finite Element Tearing and Interconnecting (FETI) methods ([2], [7], [6]). Recently, FETI-DP, the Dual-Primal Finite Element Tearing and Interconnecting method has been implemented as the replacement to the one-level FETI method previously used (see discussion below). High performance over a range of platforms is obtained through effective use of optimized BLAS routines. The BLAS routines are the building blocks for the sparse serial and parallel direct solvers used within Salinas/FETI-DP ([11], [13]).

Salinas has been used for production solutions of linear and nonlinear statics and implicit transient dynamics, and for eigen analysis and modal superposition solutions (such as frequency response, modal transient and random vibration). Extremely complex models have been analyzed utilizing combinations of beams, shells and solids. The models contain hundreds of different materials which may differ in modulus by ratios greater than  $10^6$ . Models larger than 100 million degrees of freedom have been solved with demonstrated scalability above 3000 processors. Salinas is limited to small deformation analysis, but several nonlinear elements have been added, and others are under development.

## 2 FETI-DP Overview

We present an overview of the FETI-DP method to keep this paper self-contained. Let the global domain  $\Omega$  be partitioned into a set of  $N_s$ , non-overlapping subdomains  $\Omega^s$ . Select a set of corner points for each subdomain such that all zero energy modes are suppressed if dirichlet boundary boundary conditions are applied to the set of corner points. The selected corner points remain primal unknowns which are used to define a sparse coarse grid matrix for FETI-DP. See [9] and more recent work by Lesoinne

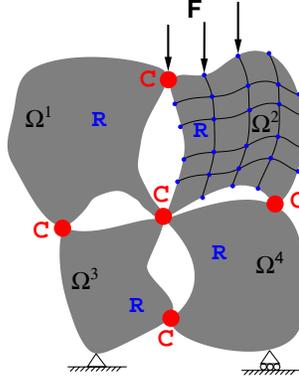


Figure 1: Dual-Primal Mesh Partitions

appearing in these proceedings about optimal corner point selection. Define  $u^s$  as the unknown solution vector associated with subdomain  $s$ . Split the global solution vector,  $u$ , into two sub-vectors such that:

$$u = \begin{bmatrix} u_r \\ u_c \end{bmatrix} = \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} \quad (1)$$

where  $u_c$  is a primal unknown vector over all selected corner degrees of freedom and  $u_r^s$  is the unknown vector for all remaining subdomain degrees of freedom on subdomain  $s$ . The subdomain operator can be partitioned into the following 2x2 block matrix.

$$K^s = \begin{bmatrix} K_{rr}^s & K_{rc}^s \\ K_{rc}^{sT} & K_{cc}^s \end{bmatrix} \quad (2)$$

Global equilibrium can be written by introducing unknown Lagrange multipliers exactly like the classical one-level FETI method.

$$\begin{bmatrix} K_{rr}^1 & \dots & 0 & K_{rc}^1 B_c^1 & B_r^{1T} \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & K_{rr}^{N_s} & K_{rc}^{N_s} B_c^{N_s} & B_r^{N_s T} \\ B_c^{1T} K_{rc}^{1T} & \dots & B_c^{N_s T} K_{rc}^{N_s T} & \sum_{s=1}^{N_s} B_c^{sT} K_{cc}^s B_c^s & 0 \\ B_r^1 & \dots & B_r^{N_s} & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r^1 \\ \vdots \\ f_r^{N_s} \\ \sum_{s=1}^{N_s} B_c^{sT} f_c^s \\ 0 \end{bmatrix} \quad (3)$$

Where  $B_c^s$  maps the local corner equation numbering to global corner equation numbering,  $f_c^s$  is the external force applied on the corner degrees of freedom,  $f_r^s$  is the external force applied on the remaining degrees of freedom,  $B_r^{sT}$  is a boolean matrix that extracts the interface of a subdomain, and  $\lambda$  are the Lagrange multipliers. Let  $K_{rr}$  denote the block

diagonal matrix of subdomain operators restricted to the remaining,  $r$ , points,  $K_{rc}$  the block column vector of subdomain coupling operator matrices and  $f_r$  the block column vector of subdomain force vectors. Using the same corner/remaining degrees of freedom matrix partitioning, we can rewrite the equilibrium equations compactly.

$$\begin{bmatrix} K_{rr} & K_{rc} & B_r^T \\ K_{rc}^T & K_{cc} & 0 \\ B_r & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \end{bmatrix} \quad (4)$$

The first equation can be solved for  $u_r$  since  $K_{rr}$  is a symmetric positive definite matrix if the selected corner points remove all of the local singularities. Then substitute the result into the compatibility equation (last equation in 4). The FETI-DP interface problem can be derived with some algebraic manipulation where the unknowns are  $\lambda$ , the Lagrange multipliers and  $u_c$ , the global corner degrees of freedom.

$$\begin{bmatrix} F_{rr} & F_{rc} \\ F_{rc}^T & -K_{cc}^* \end{bmatrix} \begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ -f_c^* \end{bmatrix} \quad (5)$$

where  $F_{rr}$ ,  $F_{rc}$ ,  $d_r$  and  $f_c^*$  are defined below.

$$F_{rr} = \sum_{s=1}^{N_s} B_r^s K_{rr}^{s-1} B_r^{sT} \quad (6)$$

$$F_{rc} = \sum_{s=1}^{N_s} B_r^s K_{rr}^{s-1} K_{rc}^s B_c^s \quad (7)$$

$$d_r = \sum_{s=1}^{N_s} B_r^s K_{rr}^{s-1} f_r^s \quad (8)$$

$$f_c^* = \sum_{s=1}^{N_s} [B_c^{sT} (f_c^s - K_{rc}^{sT} K_{rr}^{s-1} f_r^s)] \quad (9)$$

The corner degrees of freedom,  $u_c$ , are condensed out to form the following symmetric positive definite Dual-Primal FETI interface problem which we solve using a preconditioned conjugate gradient method. For a detailed derivation of this equation, please see [3]. Because of the preconditioning, the number of cg iterations (or FETI iterations) required for the solution is independent of model size. This scaling is demonstrated in the following sections.

$$\left[ F_{rr} + F_{rc} K_{cc}^{*-1} F_{rc}^T \right] \lambda = d_r - F_{rc} K_{cc}^{*-1} f_c^* \quad (10)$$

The FETI operator defined above has an embedded coarse grid problem which can be written in the following form.

$$K_{cc}^* = \sum_{s=1}^{N_s} \left[ B_c^{sT} (K_{cc}^s - K_{cr}^s K_{rr}^{s-1} K_{rc}^s) B_c^s \right] \quad (11)$$

This new coarse problem has some highly beneficial properties over the previously defined two-level FETI coarse problem ([5]). First, this new coarse problem is symmetric positive definite sparse matrix. Second, only one forward/backward substitution has to be performed per FETI iteration. The original FETI algorithms required two forward/backward substitution operations per iteration. For a detailed derivation of FETI-DP see [9], [3], [4] and [12]. For a detailed mathematical analysis of the dual-primal FETI method one can review [8] and [10].

### 3 Scaled Problem Size Scalability

We generate a series of model cube problems to assess scalability of Salinas and the underlying FETI-DP linear solver. The target platforms for assessing the scaled problem scalability of Salinas and FETI-DP are ASCI-Red, ASCI-Cplant and ASCI-White. The model cube problem is 13x13x13 hex elements per subdomain on ASCI-Red and ASCI-Cplant. On ASCI-White, we increased the model cube problem to 18x18x18 hex elements per subdomain to utilize the additional memory available. We scale the model cube problem with the number of processors keeping the size of the subdomains fixed. The number of subdomains is equal to the number of processors for all of our scalability experiments. The eight processor model cube problem is shown in Figure 2.

For each of the platforms we evaluate the number of FETI iterations, the solver time, and the total time. The solver time (or FETI-DP time) represents the total time spent in the solver. This includes setup, factorization and solve time. The total time represents the time it takes to read the input geometry files, generate the subdomain matrices, solve a single  $Ax = b$  problem and output the solution. The right hand side vector in all cases was a pressure load applied to the face opposite of the face where the dirichlet boundary conditions were applied. The convergence tolerance was 0.001 for all platforms.

#### 3.1 ASCI-Red

The ASCI Red supercomputer, also known as the Intel Teraflops machine, is the first large-scale supercomputer built mostly of commodity, commercial, off-the-shelf (COTS) components. It has 4,536 compute and 72 service nodes each with 2 Pentium Pro processors. The system was delivered with 128 Mbytes of memory per node, but has been upgraded to 256 Mbytes of memory per node. The Pentium Pro processor runs at 333 MHz and has a peak floating-point rate of 333 Megaflops. The system has over a Terabyte of real memory, and two independent 1-Terabyte disk systems. The system's 9216 Pentium Pro processors are connected by a 38x32x2 custom interconnect mesh with a bandwidth of 800 MB/s.

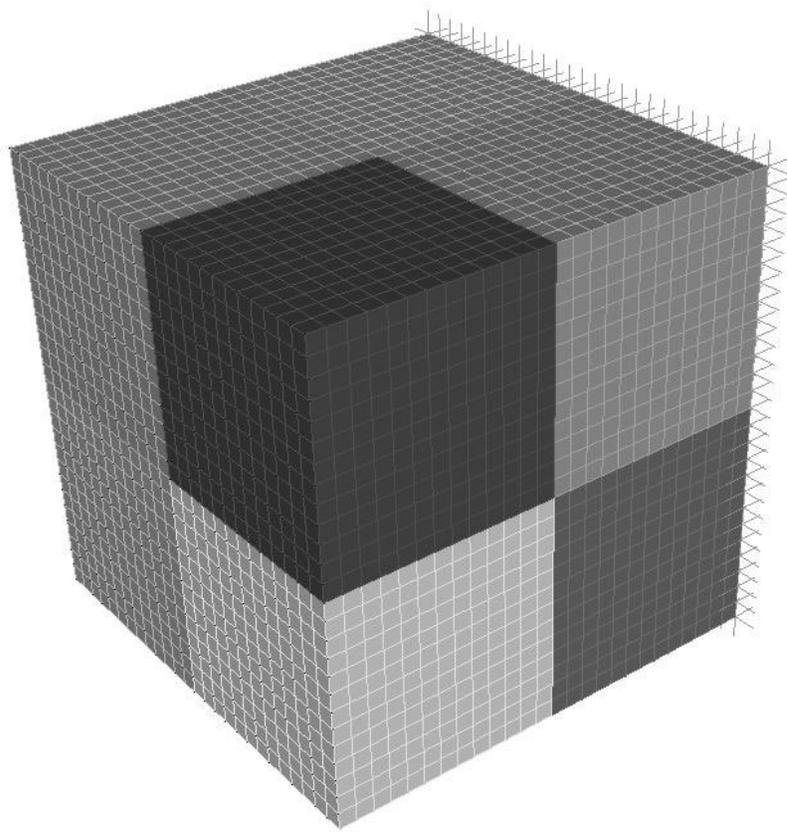


Figure 2: Model cube for 8 processors scaling study.

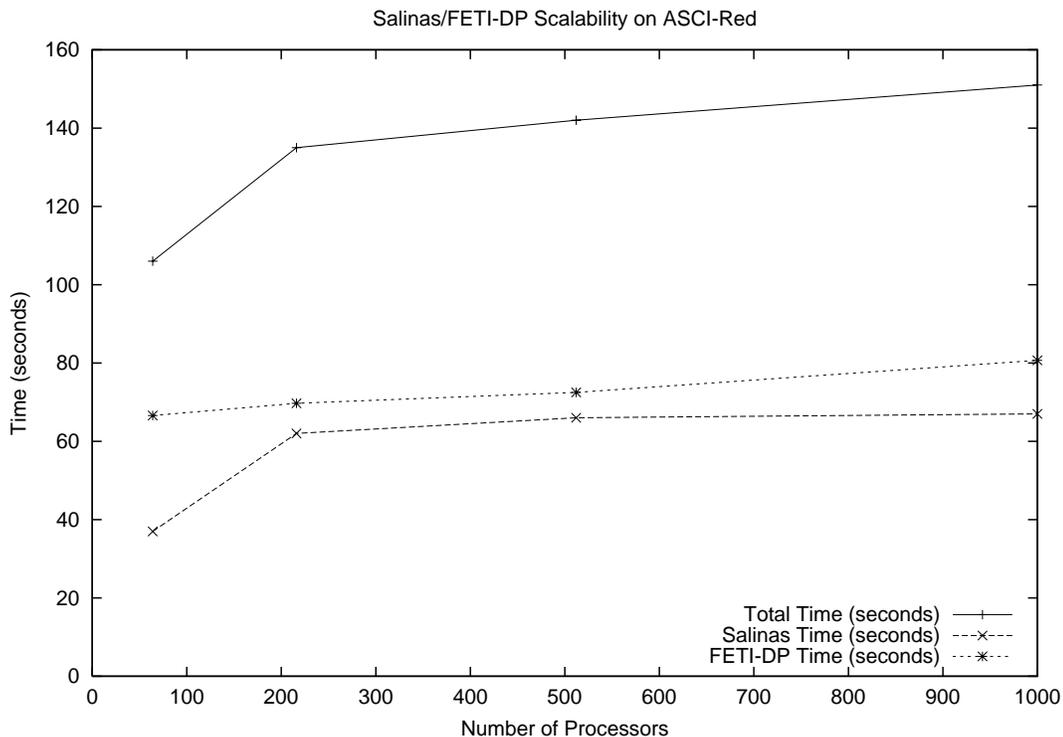


Figure 3: Performance on ASCI-Red for 13x13x13 cube/subdomain problem

We show scalability results for up to 1000 nodes on ASCI-Red. Scaling the problem from sixty-four processors to one-thousand processors increased the number of iterations from 38 to 58. In Figure 3, the total execution time is plotted for Salinas and FETI-DP running on ASCI-Red. The scalability as shown by iteration count is shown in Figure 4.

### 3.2 ASCI-Cplant

CPlant is a large-scale massively parallel computer built from commodity computing and network components with a theoretical peak performance of 1.5 Tflops. The project goal of CPlant is to develop an architecture similar to the ASCI Red machine with entirely off-the-shelf commodity parts. Cplant uses the same partition model as the ASCI Red machine where pools of nodes can be divided into different categories, such as service, compute, and I/O nodes. The compute processors are Compaq XP1000 workstations each containing a 500 MHz EV6 21264 microprocessor with 256 MB ECC SDRAM. The memory subsystem includes a 64KB instruction L1 cache and 4 MB L2 cache. The EV6 can issue four instructions per clock cycle and has two floating point units which amounts to a theoretical peak performance of 1 Gigaflops. The compute nodes are connected using Myrinet gigabit network.

We show scalability results for up to 1000 nodes on ASCI-Cplant. Scaling the problem from sixty-four processors to one-thousand processors increased the number of iterations from 48 to 58, identical to the ASCI-Red results (see Figure 4). In Figure 5 the to-

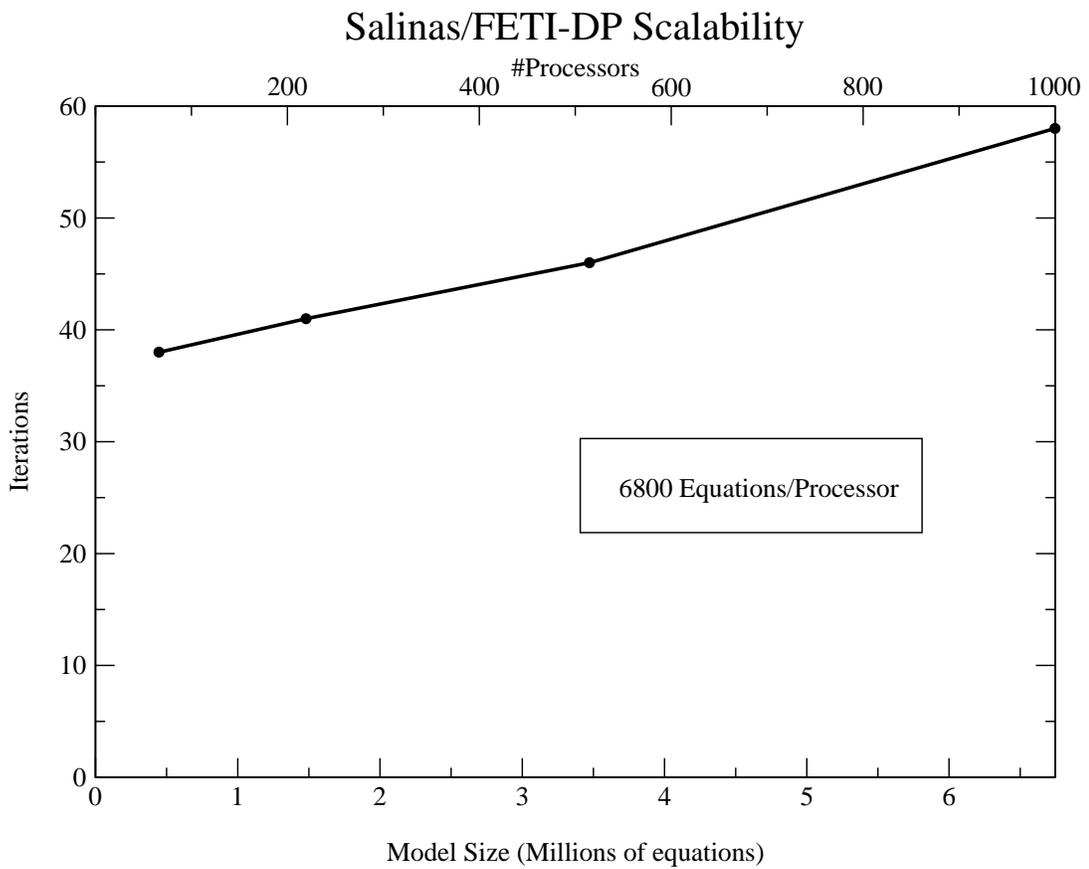


Figure 4: Iteration scalability on ASCI-Red and CPlant for 13x13x13 cube/subdomain problem

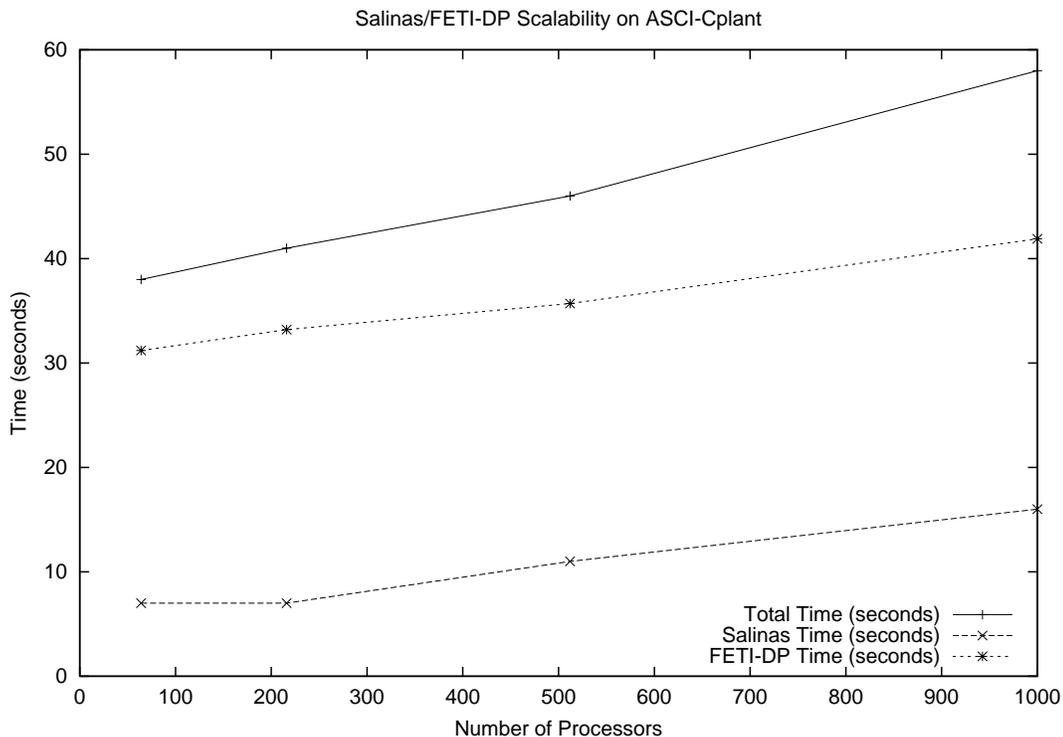


Figure 5: Performance on ASCI-CPlant for 13x13x13 cube/subdomain problem

tal ASCI-Cplant execution time is plotted for Salinas and FETI-DP. The same model cube problem was tested on ASCI-Red and ASCI-Cplant. Therefore, one can directly compare the scalability results for ASCI-Red in Figure 3 with those for ASCI-Cplant shown in Figure 5. Based on these results, one can conclude that the total analysis time is approximately three times faster on ASCI-Cplant compared to ASCI-Red. This can be readily explained by the faster processors available on ASCI-Cplant. As expected, communication affects the overall wall-clock scalability for large number of processors, but it does not affect the iteration count.

### 3.3 ASCI-White

ASCI White, the third step in a multi-step computational platform ladder, is currently the world’s fastest computer with a peak speed slightly greater than 12 Tflops. ASCI White is based upon IBM’s latest SP technology. It comprises 512 symmetric multi-processor (SMP) machines, each possessing 16 processors, for a total of 8192 processing units. Each Nighthawk-2 SMP node consists of IBM’s RS/6000 POWER3 symmetric multiprocessor (64 bit) architecture establishing a stand-alone system with its own memory, operating system, local disk and 16 CPUs. POWER3-II processors are super-scalar, pipelined, 64 bit RISC chips with two floating point units and three integer units, capable of executing up to eight instructions per clock cycle and up to two floating point operations per cycle. At 375 Megahertz, this processor is capable of producing a peak performance of 1500

Megaflops. The one cycle latency L1 cache is 128-way set associative and consists of a 64KB data cache and a 32 KB instructions cache. The 4 MB L2 cache runs typically at half the processor speed and uses a direct mapped approach. Each processor has 1 Gigabyte of available memory. All nodes are interconnected by the internal SP switch network, which has a bidirectional bandwidth of 300MB/second.

We show scalability results for up to 3375 processors on ASCI-White. Scaling the problem from sixty-four processors to 3375 processors increased the number of iterations from 58 to 71. In Figure 6 the total execution time is plotted for Salinas and FETI-DP running on ASCI-White.

## 4 Coarse Grid Solution Options

We describe two FETI coarse grid solver technologies which are implemented in Salinas. The FETI-DP coarse grid as described above is a sparse matrix that couples all of the subdomains. This coarse sparse matrix has to be factored during the FETI-DP initialization step. During the FETI-DP solve step, one coarse matrix forward/backward solve is performed per iteration. The two coarse grid solver technologies are listed below.

- Redundant storage, factorization and forward/backward solves on each Salinas processor. In this option, an inverse is computed and stored on each processor. Solves are accomplished with local matrix-vector products.
- Distributed storage, parallel factorization and parallel forward/backward solves on a subset of the Salinas processors.

For the redundant storage case, we choose a sparse matrix solver based on a multiple minimum degree ordering ([11]). The parallel factorization case is accomplished by the Domain Separator Cholesky package (DSCpack) ([13]).

### 4.1 Parallel Sparse Solver Experiments

We experiment with solving large scale problems using large numbers of processors which results in increasingly larger FETI-DP coarse grid problems. A comparison is done between the redundant factorization and subsequent coarse grid matrix inverse technique versus using a parallel distributed memory sparse solver. At each iteration of FETI-DP, the parallel sparse solver does forward/backward solves in parallel. In the future, additional studies will be conducted to determine if the parallel sparse solver in conjunction with the coarse grid matrix inverse technique will result in reduced CPU time.

### 4.2 Coarse Grid Scalability

A finite element model of an engine block was chosen for coarse grid scalability studies. A picture of the engine block finite element model is shown in Figure 7. This model is available in three increasing larger sizes. We choose the smallest model to illustrate the affect of increasing the number of subdomains for a fixed size problem on the FETI-DP

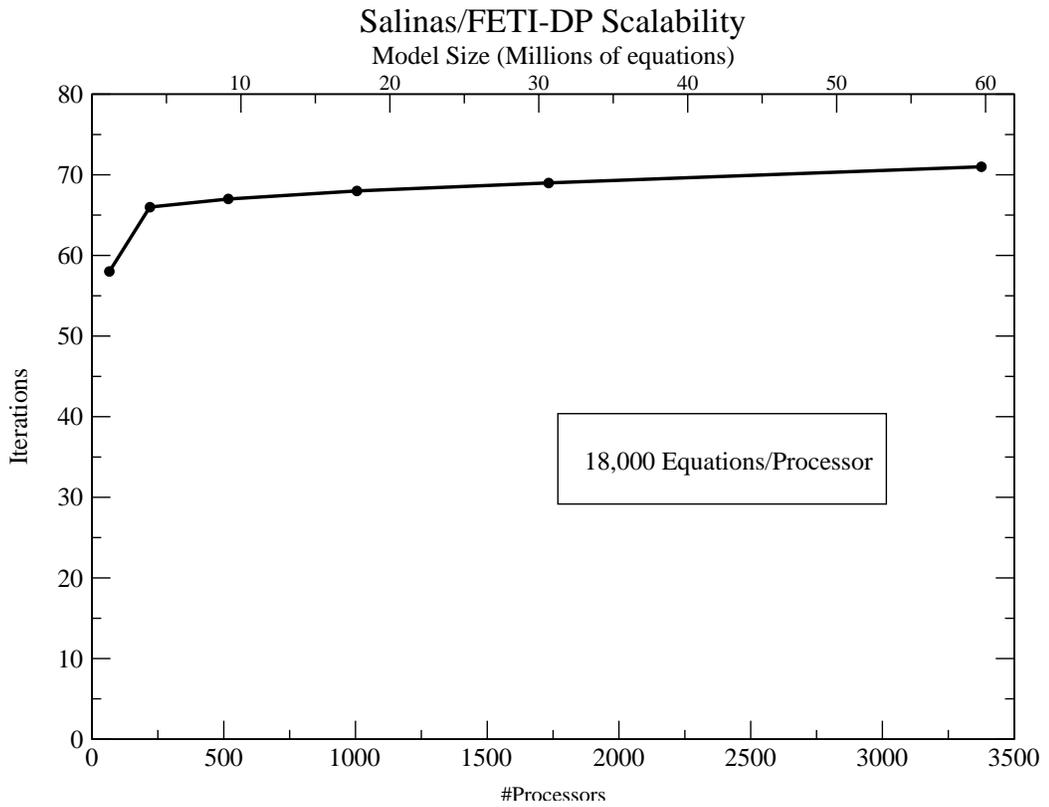
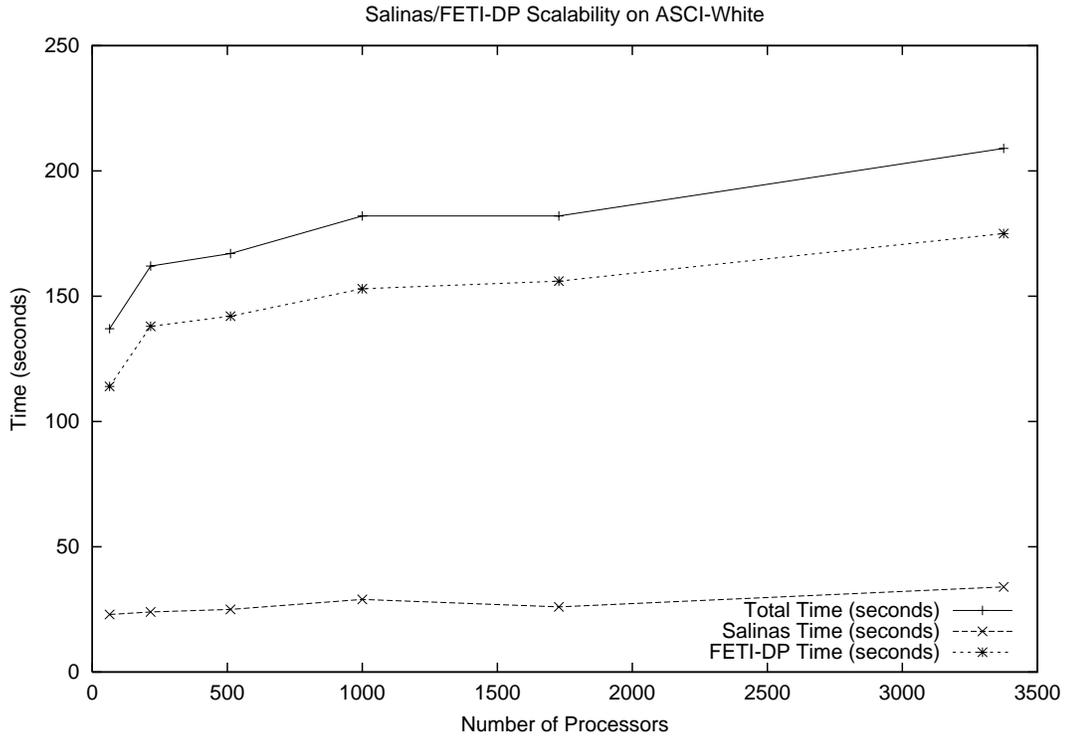


Figure 6: Performance on ASCI-White for 18x18x18 cube/subdomain problem

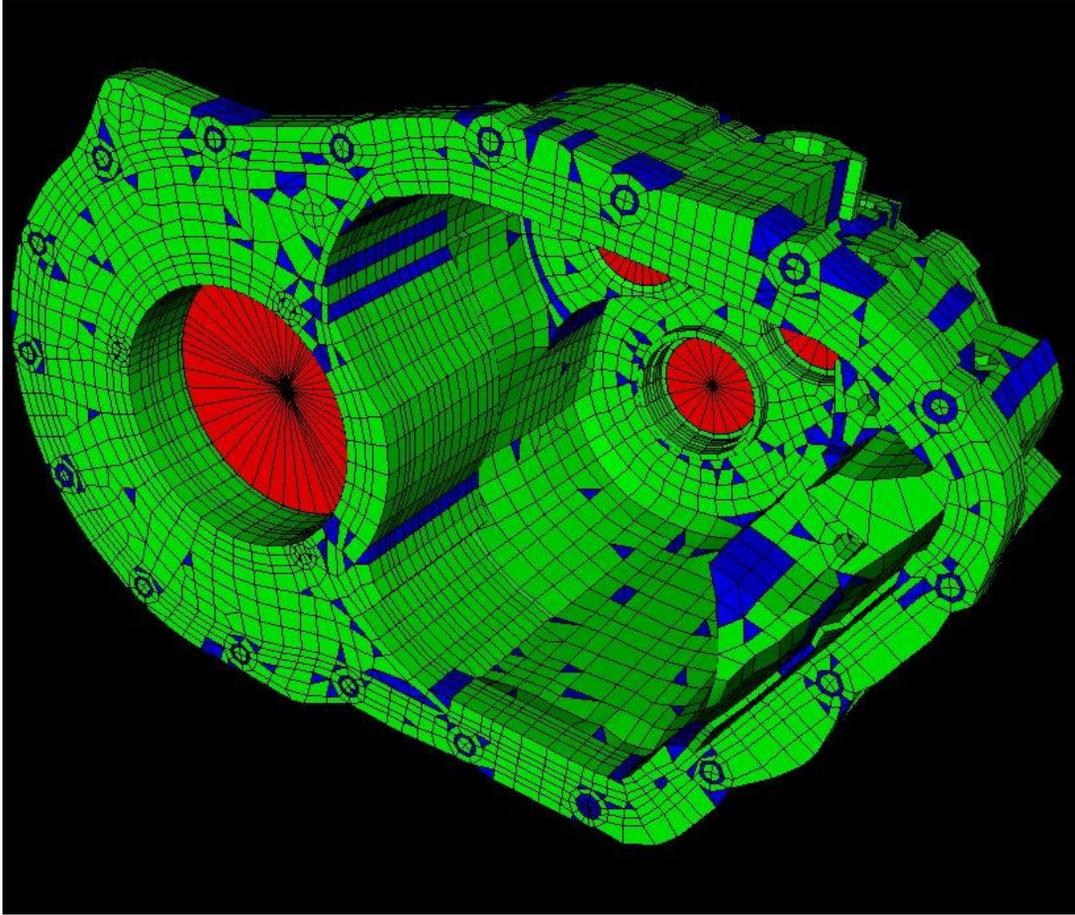


Figure 7: Engine block finite element model.

coarse grid matrix. The small engine block model has 28498 nodes, 24363 elements and approximately 75000 degrees of freedom. This problem contains hex, wedge and triangular shell elements. We partition this model into 25, 115, 137, 222 and 276 subdomains respectively. We then solve an eigenvalue system using Salinas and FETI-DP. In this Salinas solution method, FETI-DP is employed as the linear solver inside of a Lanczos based parallel eigensolver. Table 1 shows the factorization times of the two coarse grid options on an increasing number of processors. Table 1 also shows the memory requirements of the serial sparse solver. The parallel sparse solver distributes the coarse grid  $LDL^T$  factor over the number of coarse solver processors. This effectively reduces the per processor memory requirements. For small coarse grid sizes, the redundant sparse direct method outperforms the parallel sparse solver. Note that this is mainly due to the calculation of the coarse grid inverse (a detailed description can be found in [6]) and subsequent replacement of sparse forward/backward substitution by matrix-vector multiplication during the FETI-DP iterations. For a sufficiently large problem, the factorization of the coarse grid becomes the dominant time and the parallel sparse solver begins to out-perform the serial sparse direct method. More importantly, the coarse grid eventually becomes too large to store on every processor and the only option is to use the

$N_p = N_s$	$N_{eqn}^{coarse}$	Serial Sparse	Parallel Sparse	Memory
25	831	8.3 sec.	7.9 sec. (8 processors)	1.7 MB
115	3999	9.7 sec.	5.9 sec. (16 processors)	14.6 MB
137	4815	14.0 sec.	8.4 sec. (32 processors)	19.9 MB
222	7425	25.8 sec.	11.4 sec. (32 processors)	34.0 MB
276	9339	36.4 sec.	12.8 sec. (32 processors)	45.6 MB

Table 1: Coarse grid setup and factorization time for engine block on ASCI-Red

parallel sparse distributed solver. Future studies will investigate tiling the parallel sparse solver on  $N_p/N_{cs} = N_{tiles}$  while  $N_{cs}$  equals the number of processors solving the coarse grid matrix. This approach leaves an integer number of processors,  $N_{rem} = mod(N_p/N_{cs})$  idle while the current parallel sparse solver implementation leaves  $N_p - N_{cs}$  processors idle during coarse grid factorization and subsequent forward/backward solves required during FETI-DP iterations.

## 5 Conclusion

We have shown that FETI-DP performs well in a production finite element application on a variety of massively parallel platforms. Scalability was demonstrated on ASCI-Red, ASCI-Cplant and ASCI-White. We are actively pursuing parallel factorization of the FETI-DP coarse grid to enable further improvements in scalability.

## 6 Acknowledgments

The authors would like to thank Charbel Farhat and Michel Lesoinne, from the University of Colorado at Boulder for all of their time, energy, and dedication in researching and implementing FETI methods into Salinas. The authors would also like to thank Ken Alvin, David Martinez, James Peery, Dan Segalman, and Sam Key for their support, contributions and insightful discussions. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

## References

- [1] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems. *Int. J. Numer. Meth. Engrg.*, 47:513–535, 2000.
- [2] C. Farhat. A Lagrange multiplier based on divide and conquer finite element algorithm. *J. Comput. System Engrg*, 2:149–156, 1991.
- [3] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP: A dual-primal unified FETI method – part I: A faster alternative to the two-level FETI method. *Int. J. Numer. Meth. Engrg.*, 50:1523–1544, 2001.
- [4] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.
- [5] C. Farhat and J. Mandel. The two-level feti method for static and dynamic plate problems - part i: an optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–152, 1998.
- [6] C. Farhat, K. H. Pierson, and M. Lesoinne. The second generation of feti methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.
- [7] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.
- [8] A. Klawonn and O. B. Widlund. FETI-DP Methods for Three-Dimensional Elliptic Problems with Heterogeneous Coefficients. Technical report, Courant Institute of Mathematical Sciences, 2000. In Preparation.
- [9] M. Lesoinne and K. Pierson. Feti-dp: An efficient, scalable, and unified dual-primal feti method. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 421–428. DDM.org, 1999.
- [10] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. Technical report, University of Colorado at Denver, Department of Mathematics, January 2000. To appear in *Numer. Math.* URL: <http://www-math.cudenver.edu/~jmandel/papers/dp.ps.gz>.
- [11] E. G. Ng and B. W. Peyton. Block sparse cholesky algorithms on. *SIAM J. Sci. Stat. Comput.*, 14:1034–1056, 1993.

- [12] K. H. Pierson. *A family of domain decomposition methods for the massively parallel solution of computational mechanics problems*. PhD thesis, University of Colorado at Boulder, Aerospace Engineering, 2000.
- [13] P. Raghavan. Efficient parallel triangular solution with selective inversion. *Parallel Processing Letters*, 8:29–40, 1998.
- [14] G. Reese, M. Bhardwaj, D. Segalman, K. Alvin, and B. Driessen. *Salinas User's Manual*. Sandia National Laboratories.

**Distribution List:**

MS0841 T. C. Bickel, 9100  
MS0835 J. M. McGlaun, 9140  
MS0847 H. S. Morgan, 9120  
MS0835 J. S. Peery, 9142  
MS0847 K. H. Pierson, 9142 (10)  
MS0847 M. J. Bhardwaj, 9142 (5)  
MS0847 G. M. Reese, 9142 (5)  
MS0847 T. F. Walsh, 9142 (5)  
MS0847 K. F. Alvin, 9124  
MS0321 W. J. Camp, 9200  
MS1110 D. M. Day, 9214 (5)  
MS1110 D. E. Womble, 9214  
MS9217 R. S. Tuminaro, 9214

MS9018 Central Technical File, 8945-1 (1)  
MS0899 Technical Library, 9612 (2)  
MS0612 Review & Approval Desk, 9612  
for DOE/OSTI (1)

Padma Raghavan  
Pennsylvania State University  
0308 Pond Laboratories  
University Park, PA 16802

Charbel Farhat  
University of Colorado at Boulder  
Center for Aerospace Structures  
Boulder, CO 80309-0429

Michel Lesoinne  
University of Colorado at Boulder  
Center for Aerospace Structures  
Boulder, CO 80309-0429