Conf-9405109--9

IMPACT ANALYSIS ON A MASSIVELY PARALLEL COMPUTER

Thomas Zacharia
Metals and Ceramics
Oak Ridge National Laboratory
Oak Ridge, TN 37831
(615) 574-4897 (FAX: 615-574-7721)

Gustavo A. Aramayo
Central Engineering Services
Oak Ridge National Laboratory
Oak Ridge, TN 37831
(615) 574-6503 (FAX: 615-574-7986)

ABSTRACT

Advanced mathematical techniques and computer simulation play a major role in evaluating and enhancing the design of beverage cans, industrial, and transportation containers for improved performance. Numerical models are used to evaluate the impact requirements of containers used by the Department of Energy (DOE) for transporting radioactive materials. Many of these models are highly compute-intensive. An analysis may require several hours of computational time on current supercomputers despite the simplicity of the models being studied. As computer simulations and materials databases grow in complexity, massively parallel computers have become important tools. Massively parallel computational research at the Oak Ridge National Laboratory (ORNL) and its application to the impact analysis of shipping containers is briefly described in this paper.

I. INTRODUCTION

High performance computing and simulation are expected to play a key role in supporting the analysis requirements for design, manufacturing and certification of containers for transport of radioactive material components by lessening the dependence on expensive prototyping and re-tooling. Computer models can incorporate all significant features of the container, thereby providing a convenient tool to support qualification and licensing efforts. The cost of analysis is, in general, significantly less than that of experimental tests. Modifications to the container design and testing conditions can be effectively performed using computer modeling in an economical and timely manner. Depending on the complexity of the problem, these analysis may require an inordinate amount of computational time. Even on current supercomputers, some of these relatively small problems can be intractable. In studies of impact and penetration, it is not unusual for an analysis to require many hours of computational time on current supercomputers despite the simplicity of the models being studied. The computational requirements for a single Belytschko-Tsay shell element with five integration points through the thickness are on the order of 3500 floating point operations per time-step and 602 words of memory. Impact analysis models currently use 50,000 elements and run for about 100,000 time steps. To meet the need for greater processing speed, massively parallel computer designs have emerged that make multiple processors available to the user. However, they are limited in use due to the lack of availability of applications software. In order to exploit the emerging high performance computers, existing and future codes must be restructured. In this paper, ORNL's early experience with these emerging massively parallel computers for impact/crash analysis are discussed.

II. HIGH PERFORMANCE COMPUTING

Today's leading supercomputers are capable of performing over one billion floating point operations per second (Gflops). The current generation of conventional supercomputers attain such speeds by combining a small number of very powerful vector processors having a cycle time of about 4 to 10 nanoseconds. Massively parallel computers, which combine several thousand processors, are able to operate concurrently on a problem. These computers provide exciting new opportunities for process modeling by allowing the implementation of more complex databases and more realistic simulations.

At ORNL, researchers have the choice of several parallel computing platforms ranging from Kendall Square Research KSR-1 to several Intel Paragons and iPSC/860. These parallel computers are one of two general types: shared memory (KSR-1) and distributed memory (Intel) multi-processors. Shared memory multi-processors allow all processors equal access to memory that has been declared global, usually through a common communication channel. This means that all of the processors can access any portion of the computer's memory. As the number of processors sharing the memory and the communication channel increases, the potential for contention between the processors for control of the memory increases. Distributed memory multiprocessors are characterized by a network of communication channels, each of which connects a processor to other processors and memory units. Typically, each processor has a local memory of limited size. When a processor needs data stored in memory other than its own, the computer program must explicitly request transmission of those data, as needed. The processors would send and receive messages containing the results of calculation in a processor. Extra programming effort is required for coordinating data movement. Because communication channels are not shared in distributed-memory-architectures, they are more efficient than bus architectures in the use of processors and memory. The lack of equal access to memory by all processors makes the use of distributed-memory computers more difficult and more limited. Although not all problems lend themselves to a parallel solution, our experience shows that a number of computationally intensive problems can be efficiently solved on parallel computers. For example, ORNL's Intel Paragon, a distributed memory multi-processor, and KSR-1 shared memory multi-processor, have both been used for the numerical solution of problems ranging from metal-forming to crash analysis.[1-3]

III. FINITE ELEMENT MODEL

The mathematical model of the shipping container was generated using the program LS-INGRID,[4] and the post-processing of the results was performed using AVS[5]. The symmetrical half model of the shipping container had 22,539 nodes, 10,452 8-node solid elements, 6562 4-node shells, and 36 sliding surfaces.[6] The number of sliding surfaces were reduced by connecting several surfaces together to form a single surface. A large number of sliding surfaces were determined to be inefficient for massively parallel processors. The model used 26 different material components. Figure 1 shows the finite element model used in the present analysis. Most of the metallic materials were represented using a piecewise linear isotropic plasticity model. The bolts in the flange interface between the external lid and the cylindrical shell have been modeled using a failure model that uses the effective strain at failure as an indicator that allows the automatic deletion of the element where it exceeds a critical strain. The numerical analysis was performed using a massively parallel version of the LS-DYNA3D.[7]

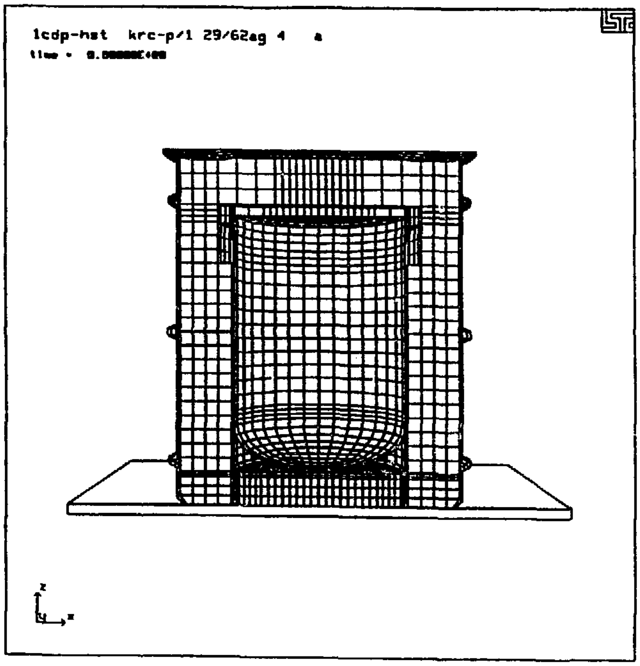

Figure 1. Finite element model using LS-DYNA3D.

IV. DOMAIN DECOMPOSITION

If large problems such as the impact analysis of containers are to be solved on distributed-memory parallel computers, their data structures must be partitioned and distributed across processors. The partitioning process must minimize interprocessor communication and maximize load balancing.

The allocation of finite elements among multiprocessors introduces communication costs which must be controlled to achieve maximum speedup. Greedy algorithms are simple and balance processor loads for a constant mesh topology. They neither minimize the number of subdomain interface nodes (message length) nor the number of neighboring subdomains (number of messages). Both of these problems are aggravated in an impact-contact simulation.

Recently, the recursive spectral bisection (RSB) algorithm[8] has received significant attention from the parallel computing community. RSB is derived from a graph bi-section strategy and recursively divides the computational domain, as specified, into powers of 2 (2^1, 2^2, 2^3, 2^4, ...). The RSB algorithm has been modified[9] to decompose the computational domain into any specified number of subdomains. This algorithm was used to automatically decompose the finite element mesh of the container, shown in Figure 1, into N submeshes (where N is the number of processors), such that the subproblems associated with each processor had approximately the same complexity and the amount of communication between the processors was minimized.

Figure 2-4 shows partitioning of the container problem using the modified RSB algorithm. Figure 2 shows partitioning of the container problem for two processors. In this case, the decomposition algorithm assigned most of the outer steel shell containing fine shell elements to one processor and the rest of the problem to the second processor. Figure 3 shows the same problem decomposed for 4 processors. In this case, the algorithm assigned a large part of the outer shell, containing coarse mesh, to one processor and the remaining outer shell, containing finer mesh, to a second processor. The primary and the secondary containment vessels are each assigned to the third and the fourth processor, respectively. Figure 4 shows the partitioning of the same container problem to

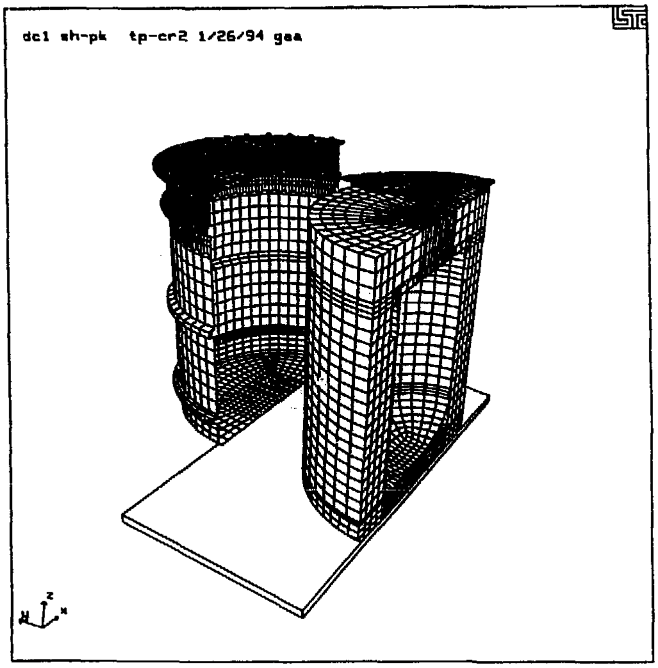

Figure 2. Finite element model decomposed for calculation on two processors.

a larger number of processors. In each of the cases considered here, the partitioning of the mesh is performed in order to maintain computational load balance among the various processors.

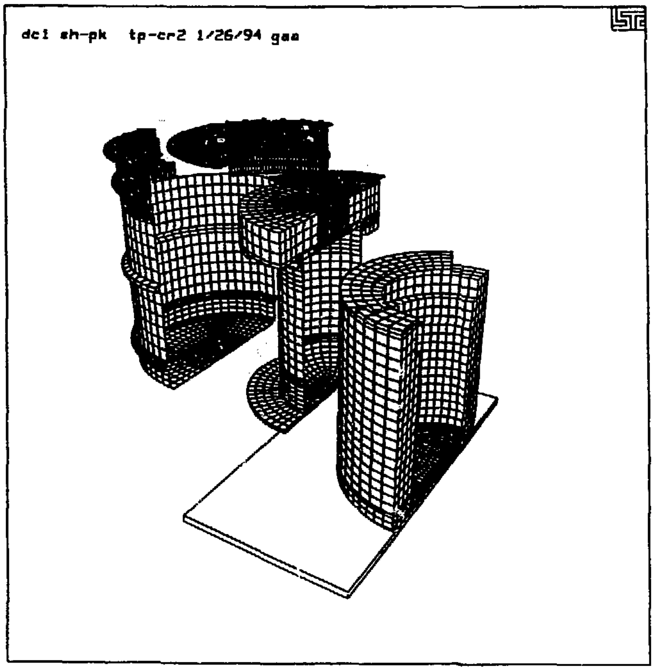

Figure 3. Finite element model decomposed for calculation on four processors.

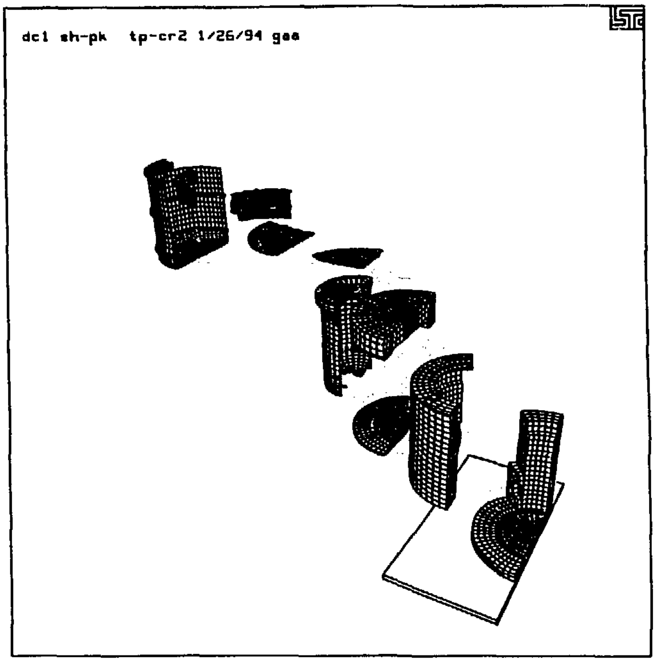

Figure 4. Finite element model decomposed for calculation on eight processors.

V. RESULTS

Results of the impact analysis of the container are presented in detail elsewhere in this proceedings.[6] As stated earlier, the objective of this study was to assess the applicability of massively parallel computers for impact analysis of shipping containers. Initially, vertical impact on base assembly of a simple container problem containing 5260 shell elements and 9198 nodes were analyzed using the Intel Paragon. Scalability studies from 16 to 256 processors indicated that the problem scaled very well. Initially, as shown in Figure 5, the CPU time per iteration decreased with increasing number of processors up to 64, beyond which the time increased with increasing number of processors. This, however, was not surprising due to the small size of the problem.

The efficiency of massively parallel computation for impact analysis was influenced by two factors: the computer's "load balance" and communications costs. If work is distributed unevenly, then most of the processors may be idle during much of the computation time - a waste of the computer's power. Even if work is well distributed, processors will be idle while they are waiting to receive the data from other processors needed to perform the next task. Typically, the computational time (t) for a particular multi-processor (p) application can be expressed as,

$$t = A + B*p + C/p \tag{1}$$

where A is a constant related to initial setup (usually negligible), B is a constant that related to interprocessor communication, and C is the computation time for a single processor. From Equation 1, it can be seen that the overall performance is determined by the communication cost that increase with number of processors and computational costs that decrease with number of processors. Depending on the relative value of B and C, it can be seen that the overall computation time, t, may initially decrease, as shown in Figure 5, with increasing number of processors up to a point beyond which t would increase with increasing number of processors. For large problems containing ~20,000 elements or more, our analysis shows (Figure 6) good scalability and efficiency. The results clearly show a reduction in computation time from 31 days to 6 hours as the number of processors used in the analysis increased from 1 to 128. In this case, C (computation performed in a single processor), is sufficiently large such that B is relatively small.

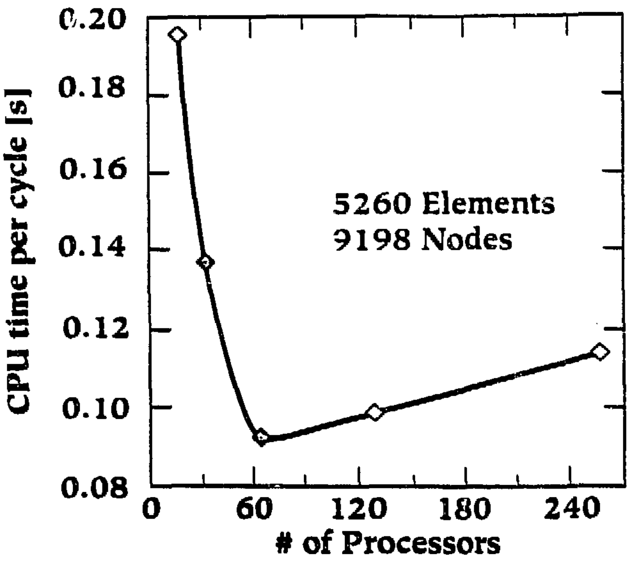

Figure 5. Performance of LS-DYNA3D on the Intel Paragon for a problem containing ~5000 elements.

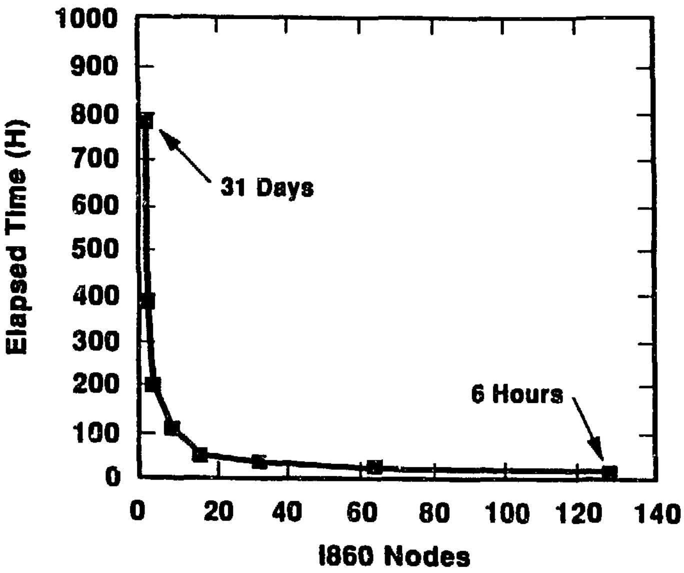

Figure 6. Performance of LS-DYNA3D on the Intel Paragon for a problem containing ~20000 elements.

Following these analyses, the container shown in Figure 1 was analyzed for a vertical impact on base assembly. Figure 7 shows the configuration of the container after the impact. The results of the analysis was compared with previous validated calculations performed on a scientific workstation.[6]

VI. CONCLUSIONS

Modeling, simulation, and high performance computing have evolved as a powerful technology for industrial design and manufacturing. Effectively coupling unique state-of-the-art massively parallel computers and large deformation models for impact analysis is an important milestone. Problems that took several weeks on a workstation were solved in a few hours on the massively parallel machine. Apart from serving as the basis for improvements in fundamental understanding, these technologies will play an important role in container transport safety from design to manufacturing. Improved computational modeling capability can contribute to more effective approaches to problem solving, new products and enhanced national competitiveness.

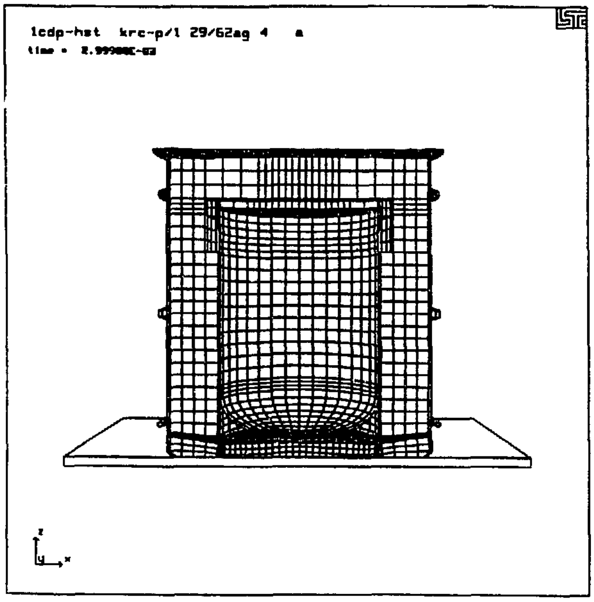

Figure 7. Simulation results of vertical impact on base.

VII. ACKNOWLEDGEMENTS

The authors would like to thank S. Šimunović for reviewing the manuscript and D. McCoy for technical editing. The authors acknowledge use of the Kendall Square KSR1-64 and Intel Paragon XP/S5 computers located in the Oak Ridge National Laboratory Center for Computational Sciences (CCS), funded by the Department of Energy's Office of Scientific Computing. The research was sponsored by the U.S. Department of Energy, Assistant Secretary for Energy Efficiency and Renewable Energy, Office of Industrial Technologies, Advanced Industrial Materials Program, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

VIII. REFERENCES

1. T. ZACHARIA, M. J. BJERKE and S. ŠIMUNOVIĆ, International Conference on Modeling and Control of Joining Processes, Dec. 8-10, 1993.

2. P. SU, R. FULTON and T. ZACHARIA, ORNL/TM-12732.

3. P. SU, R. FULTON and T. ZACHARIA, ORNL/TM-12733.

4. D. W. STILLMAN, and J. O. HALLQUIST, "LS-INGRID: A pre-processor and Three Dimensional Mesh Generator for the Programs LS-DYNA3D, LS-NIKE3D, and TOPAZ3D," LSTC Report 1019, Livermore Technology Corporation (1992).

5. AVS USER'S GUIDE, May 1992.

6. G. A. ARAMAYO, "Impact Analysis of the DC-1 Shipping Container," proceedings of this conference.

7. J. O. HALLQUIST, D. W. STILLMAN and T. LIN, "LS-DYNA3D Users Manual," LSTC Report 1007, Rev. 2, Livermore Technology Corporation (1992).

8. H. D. SIMON, Computing Systems in Engineering 2, (2/3), 135 (1991).

9. B. WAINSCOTT, private communication, 1993.