Experiences Running LS-DYNA3D in Production on the Cray T3D

John W. Gregory, Cray Research, Inc., Eagan, Minnesota

ABSTRACT: *This paper describes LS-DYNA3D, a three dimensional nonlinear industrial simulation software package, and aspects of its implementation on the massively parallel CRAY T3D. LS-DYNA3D solves models in such areas as automobile crash simulation, metal forming and stamping, and impact/penetration analysis. The version on the CRAY T3D uses the message passing paradigm, and results so far are very promising, giving speeds on various models well in excess of what is possible on a CRAY C90, up to a measured Gigaflop on 512 processing elements and above. Current status and directions for future work will be given.*

1 Introduction

LS-DYNA3D, developed by Livermore Software Technology Corp (LSTC), is a sophisticated large-scale Finite Element Analysis (FEA) code used to analyze the deformation response of inelastic three dimensional solids. This paper describes the massively-parallel (MPP) CRAY T3D version of LS-DYNA3D, and in some instances contrasts it to the parallel-vector (PVP) version that runs on Cray Research's vector supercomputers.

Notable applications of LS-DYNA3D are in the automotive industry (crash, airbag modeling, occupant safety, and roof crush), the aerospace industry (bird strike, turbine blade containment), general manufacturing (sheet metal stamping, forging, rolling, plastics technology), and many others.

FEA codes are among the most computationally intense applications used in industry. Demands on both the hardware and the software are growing, due to steadily increasing use and more detailed models. There is thus a natural interest in running LS-DYNA3D on the CRAY T3D, because of the promise of high scalability as processing elements (PEs) are added.

The benefits expected to arise from using an MPP version come from increased capability, and also better usage of the computing capacity. For the largest models, solution becomes possible in a more timely manner than before; while for a mix of smaller production jobs, the MPP architecture allows a new level of flexibility in matching resources to tasks. For an industrial customer, this translates into higher levels of safety, productivity, and competitiveness, in less time.

2 Porting Methodology

LS-DYNA3D has been used successfully on Cray PVP platforms for many years. It is reasonable to start from this version, for the port to the CRAY T3D. However, the approach taken with the MPP version differs from the vector version in significant ways. Because of the distributed memory of the CRAY T3D architecture, it was decided that a Domain Decomposition approach would fit this problem class most naturally, as opposed to a data-sharing/work-sharing approach such as provided by the CRAFT programming model.

There are at least two major advantages of this decision. First, all of the logic to handle the parallelism is placed at a high level in the program; the subroutines that perform the actual computations are, in most cases, completely untouched. This means, for instance, that if enhancements are made to the PVP code, the changes can be immediately plugged into the MPP version. The second advantage is that good parallel speedups are quickly achieved because individual computation loops need not be parallelized; for some codes, where the bulk of the work is done in a handful of loops, this would not be viewed as an advantage, but for a code like LS-DYNA3D where individual loops never account for as much as 10% of the total run time, it is a major benefit.

LS-DYNA3D is written mostly in Fortran with a small amount of C. The CFT77 and CC compilers accept the same source code on Cray's PVP and MPP platforms, easing the porting effort.

The MPP version of LS-DYNA3D uses the MPI message passing library. In order to achieve an early release of LS-DYNA3D for the CRAY T3D, it was decided to write a library of translation routines to permit the use of PVM. Originally, the expectation was that this would be a porting tool only, and that in-line calls to PVM would be mandatory (or, in places where performance was crucial, the SHMEM library could be substituted). To our surprise, it has turned out that communications does not appear to be a bottleneck on the CRAY T3D in the solution algorithms, for any models of commercial importance. This is presumably due to the careful attention paid by

---

LSTC developers to keeping communications minimal and efficient. Use of the CRAY T3D's more sophisticated message-passing techniques has therefore been postponed to a later time.

Several Cray Research tools have been used during this project. The Emulator and Simulator were used early in the porting effort, before native hardware was available. The Apprentice helped locate where the time was being spent; the RTC (run time clock) was used to identify the places where load balance was hurting performance. And the Total view debugger has been helpful in investigating the movement of data within the code.

During the development of the MPP version of LS-DYNA3D, features were phased in gradually. Generally, algorithms related to metal forming applications were installed first, and those supporting crash analysis were added afterward. All the main LS-DYNA3D methods are now in place to solve industrial sized structures problems on the CRAY T3D.

3 Supported Features

At this writing, input format 920 is supported. Format 930 and keyword format support are planned for the near future.

The MPP version assumes arbitrarily numbered input. This affects the format of some of the input options. For example, if initial velocity option 1 is used, ALL nodes in the problem must appear in the initial velocity section.

It is intended that the MPP version of LS-DYNA3D will support, eventually, all features found in the vector version. There are still some features that remain unimplemented; these will be added based on customer demand.

It is shorter to list, as given in the “MPP-DYNA3D Users Guide”, the features that currently are entirely unsupported, rather that enumerate the multitude of supported LS-DYNA3D features:

- All ALE related options
- Interface Definitions Component Analysis
- DYNA3D/JOY Interface Definition
- Traction Boundary Cards for Beam Elements
- Nodal Force Groups
- Tie-Breaking Shell Definitions
- Tied Node Sets with Failure
- Nodes Moved Via Section 11 Interface File
- Generalized Body Force Load Input
- Detonation Point Data
- Shell-Brick Interfaces
- Nonreflecting Boundary Segments
- Temperature Data
- 1D Slideline Definitions
- Lumped Parameter Control Volumes
- Rigid Body Stoppers
- AVS Database
- MPGS Database
- MOVIE Database
- Input Data For User Interface Subroutines
- Cyclic Symmetry
- Superplastic Forming Option
- Material Repositioning Section
- Displacement Termination
- Tracer Particles
- Shell Parts Tied to Solid Parts
- USA Surface Boundary Condition Cards
- Temperature Boundary Condition Cards
- Flux Boundary Condition Cards
- Convection Boundary Condition Cards
- Radiation Boundary Condition Cards
- Rigid/Deformable Material Switching

Here is the list of partially supported features. These features are fully supported except as noted.

- Element Time History Blocks. Only thin shells are currently supported.
- Generalized Stonewall Cards. 4 node force segments are not supported.
- Sliding Interface Definitions. Types 3, 4, 5, 10, 13 are supported. See Users Guide for other important information about Sliding Interfaces.
- Seat Belts. Accelerometers are not yet implemented.
- Restart capabilities. Only the termination time, plot interval, time step control, and restart dump frequency may be changed when restarting.

4 Unique Features in the MPP Version

Certain aspects of the MPP version of LS-DYNA3D do not have an analogue in the PVP version. The user has control over how these features work. This is accomplished through the use of a parameters file, named “pfile” by default. A script is provided, which sets up a pfile if the user does not provide one.

Before a structure can be solved on multiple processors, it must be decomposed and the data distributed to the processors. MPP LS-DYNA3D offers three algorithms for decomposition. The greedy algorithm is quick, but in practice does not seem to give especially good decompositions. The Recursive Spectral Bisection (RSB) method is based on a coloring algorithm and generally gives decompositions that are much more even across all PEs. The Recursive Coordinate Bisection (RCB) performs a decomposition along the coordinate axes, and for some models it can be the best. There are a few parameters in the pfile that control the action of these decomposition methods.

Compared to the total time of a typical run (several hours) the time to perform the decomposition is not very long (a minute or two). Nevertheless, because it is an inherently scalar operation, an optional pre-decomposition program is provided. This pre-decomp runs on the vector front-end of the CRAY T3D, and leaves a reusable file containing the decomposed model, which the MPP program then reads and distributes to the appropriate PEs. The main advantage of the pre-decomp is speed; however it also allows the execution of models that are too large to fit (for the decomposition operation) on one PE, and does not tie up the MPP for this scalar operation.

The contact algorithm was extensively rewritten for the MPP version, and the user has control over several options, such as the bucket size for the periodic sort, the sorting frequency, memory usage, and the penetration thickness used for solid brick elements in the contact surface.

The user can also specify, in the pfile, the name of auxiliary files and directories to be used in the course of the solution algorithm. Since each PE works independently (except when communicating with other PEs), more files are created by the MPP version than with the PVP version, and a scratch directory is created to house these files; they can generally be discarded at the end of the run.

Because the MPP version is somewhat more complex than the PVP version, a script called "t3ddyna" is provided to automate the steps of running it. The script takes all the parameters normally accepted by LS-DYNA3D, and passes them to the MPP version. An additional parameter, the number of PEs, may be specified (optionally) as the first parameter; a suitable default value is provided. If the user provides a pfile (specified with a p=filename parameter), then that file is used; otherwise, a default pfile is generated and used. At the end of the run, certain conversion steps are performed on the plot files and ASCII files to make them usable, and then the script removes the scratch files in the temporary directory. The intention is to make the CRAY T3D version look and operate much like the user is accustomed to; since it is run by a script, the user is at liberty to alter the procedure as desired, or even discard the script entirely and run the executable file directly.

5 Typical Performance

All comments regarding the performance of the MPP version of LS-DYNA3D need to be prefaced by the observation that model structure has avery large effect on scalability. Models that contain a large number of contact interfaces tend to be very difficult to decompose in an effective manner. We also do not yet have much experience with the performance on crash models, since most of the production work to date has been aimed at metal forming problems.

With that in mind, it is our experience that in using LS-DYNA3D for metal forming, one can gain approximate parity with a single CRAY C90 CPU by using 32 PEs on a CRAY T3D. This would be for dedicated runs with each machine; in practice, in a heavily loaded production environment, the CRAY C90 times might be affected greatly (upward) when measured as elapsed wall-clock time, whereas of course the elapsed time on the CRAY T3D is what is always reported. So parity with a CRAY C90 can actually translate into an advantage for the CRAY T3D in time-to-completion for a run.

Larger numbers of PEs can result in still better run times. On average, most models seem to get a speed benefit up to 128 PEs, and some continue to benefit beyond that.

At these high numbers of PEs, it is important to note that a "benefit" does not necessarily imply an "optimal choice". Amdahl's Law tells us that as the number of processors is increased, the relative benefit declines unless the work is 100% parallel. It is difficult, on many models, to measure the level of parallelism in LS-DYNA3D directly, because large models do not fit in memory on 1 CRAY T3D processor. Therefore, some extrapolation is required, to guess what the speed would have been on 1 processor. Comparing execution times on runs with different numbers of PEs, the implied parallelism is almost always at least 95%, and generally in excess of 98%. While these are good results for general asymmetrical structures, they imply that doubling the PEs from 64 to 128 may give only a modest boost in speed; if two jobs of equal priority need to be run, better throughput and aggregate elapsed time would be had by running them each with 64 PEs at the same time, instead of at 128 PEs one after the other. Increasing this economic "sweet spot" remains a challenge for the MPP version of LS-DYNA3D.

It should be noted that the choice of decomposition algorithm can have a very large effect on parallel speedup. The default decomposition in the T3D script is a RCB compressed in the Z direction; this seems in keeping with many of the models we see, oriented mainly along the X and Y axes, but can have disastrous effects on models oriented a different direction. Thus, when solving a family of related models, a bit of experimentation will usually pay off in better performance of the code.

A final note concerns the writing of disk files. The area of I/O is currently relatively weak, compared to the scalable performance we see in the rest of the code. When many processors are in use, frequent writing of files can become a major bottleneck in the code, as the time to do these steps can actually increase with more PEs. The best advice is to do as much I/O as is needed, but no more than that. Extra writing of restart/dump files, in particular, should be avoided whenever possible. Frequent writing of plot states will also be noticeable in slowing the code down.

6 Study of Two Models

Performance results are presented here for two models of interest. Results include the *total* time to read, decompose and solve the model, and do not represent merely a kernel of computation. Please note that timings like these tend to become out of date very quickly, on both the CRAY MPP and the CRAY PVP hardware, due to improvements in the application software and in the compiler/OS suite; contact your local Cray representative for updated values.

The first model was developed as the crash benchmark for the VDI (Society of German Engineers) conference at Würzburg 1994. It is a simulation of a vehicle impacting a rigid barrier head on. The model has 26949 nodes, 28007 shell elements, 216 beam elements, and 222 materials. There are 5 contact interfaces.

Performance is shown in Table 1, measured in Megaflops. LS-DYNA3D on a CRAY C90 ran at 237 Megaflops, taking 3 hours and 41 minutes. This level of performance was reached with 32 PEs on the CRAY T3D. Going to 128 PEs approximately doubled the speed, implying an effective level of parallelism of about 98.5%. A further small increase in speed, to the 500 Megaflop level, was obtained with 256 PEs.

The second model was developed as the sheet metal forming benchmark for the VDI conference in Zürich in 1991. The original problem was rather small by current standards, and so in this case the mesh of the blank material was refined, resulting in a total of 19014 elements. There are 3 rigid tooling parts: a lower die, a binder, and an upper die.

Table 2 presents the measured performance for this model. The observed CRAY C90 performance is comparatively low (110 Megaflops, taking 145.2 minutes), and since the CRAY T3D version uses many of the same routines for computation, the MPP version of LS-DYNA3D also has lower single-PE performance. However, the effective level of parallelism is quite high, 99.5%, giving very good scaling, so that CRAY C90-level performance is achieved with only 16 PEs. An increase to 128 PEs gives almost a six-fold increase over that. And at 512 PEs, the total elapsed time is 15.8 minutes, resulting in performance just above 1000 Megaflops (1 Gigaflop).

Table 1. Megaflop performance for VDI Crash benchmark (Oct. 1994).

Computer	Number of Processors							
	1	4	8	16	32	64	128	256
Cray C90 Vector version	237							
Cray T3D		45	80	140	240	350	460	525

This model was also run on 1024 PEs, and absolute performance decreased. An analysis of the way the time was spent indicates the main computation of the finite element algorithm was faster than at 512 PEs, consistent with a 99.5% parallelism; but the startup time (to perform the distribution of the data to the PEs) and the ending time (to write a restart file, notably) had come to dominate the total time. This is an area for future study and refinement.

7 Plans and Conclusion

It may seem trite to say that much work remains, but it is true. LS-DYNA3D accepts a wide variety of model formulations, and the MPP version must be tuned for high performance on all of them. Additional features, not yet supported in the MPP version, must be added. I/O implementation is a concern, for high numbers of PEs. And work must begin now to prepare for the CRAY T3D's follow-on product.

Nonetheless, LS-DYNA3D on the CRAY T3D already is a powerful and effective tool for solving industrial sized problems in structural simulation.

Table 2. Megaflop performance for VDI Metalforming benchmark (Oct. 1994).

Computer	Number of Processors										
	1	2	4	8	16	32	64	128	256	512	1024
Cray C90 Vector version	112										
Cray T3D	9	17	33	60	110	210	370	630	870	1025	770