SPH Performance Enhancement in LS-DYNA

Gregg Skinner and Dennis Lam
Advanced Technical Computing Center
NEC Solutions (America), Inc.

Masataka Koishi
Yokohama Rubber Corporation

Hiroki Shimamoto
NEC Corporation

Abstract

The Smoothed Particle Hydrodynamic (SPH) method had been implemented in LS-DYNA for some time. However, SPH had not been used extensively; therefore, performance issues were never highlighted and never addressed. Recent efforts to run SPH on NEC SX systems revealed substantial performance problems. NEC, Yokohama Rubber Corporation and LSTC collaborated to enhance the performance of SPH. As a result, the performance of SPH function in LS-DYNA has been improved on NEC SX-6 vector-parallel supercomputer by a factor of four. This article provides some background information about the code tuning effort, the SX series vector-parallel supercomputers, and the performance improvement achieved

Introduction

The implementation of smooth particle hydrodynamics (SPH) in LS-DYNA expanded further the capability of LS-DYNA to solve highly nonlinear problems. SPH method was developed in late 70's to solve astrophysical problems. Because of the distinct advantage to handle very large deformation, SPH method has been adopted to solve other applied mechanics problems. The complicated automotive tire behavior such as hydroplaning is difficult to simulate with traditional finite element method. SPH in LS-DYNA stands out as the best tool to solve this type of problem. However, the initial performance of SPH was rather poor on SX vector-parallel supercomputer. A joint project among NEC, Yokohama Rubber Corporation, and LSTC was set up to address performance issues.

SX Systems

SX vector-parallel supercomputers were developed and manufactured by NEC. In 1983 NEC launched SX-2. It was the beginning of NEC's commitment to provide the most powerful computer for scientific computing. The current generation of SX series vector supercomputers, SX-6, is the most powerful and competitive vector-parallel supercomputer that NEC has ever built. Although the technology behind SX-6 evolved from its predecessors, SX-6 is remarkably different from all previous generations.

An SX-6 node, which has been upgraded in forth quarter of year 2003, is a complete parallel-vector systems consisting of 2 to 8 vector processors each with 9 GFLOPS of peak performance. The processors are coupled to a uniform shared main memory of up to 128 GB capacity. The SX series solution for large scalability is a hybrid system. A powerful SMP single node with uniform shared memory provides high performance for running both LS-DYNA/MPP and LS-DYNA/SMP. Multi-node, distributed-memory, configurations are available for running LS-DYNA/MPP at higher performance. In a multi-node configuration, up to 128 nodes can be connected through an Inter-node Crossbar Switch (IXS). SX-6 series multi-node models can be scaled up to 1024 CPUs and a peak performance of 9 TFLOPS. Although they are built on distributed memory architecture, multi-node systems still provide a single system image.

SPH Performance Enhancement

Although SX series vector-parallel supercomputers have the most powerful processors to run LS-DYNA, the performance can suffer significantly if the software code cannot be vectorized. At the onset of this project, certain subroutines in SPH implementation were not vectorized. Based on two SPH input data from tire simulations at Yokohama Rubber, we set performance goals to increase by four times the original performance and to reduce the turn around time of the larger simulation to less than two days. The resulting performance improvements benefit not only tire simulations, but also any LS-DYNA simulation that uses SPH method.

Using SUPER-UX profiling tools on SX-6, we identified the most time-consuming SPH subroutines. In these subroutines we found core SPH calculations that were not executing in vector mode. We modified these loops so that they could be vectorized by the FORTRAN90/SX compiler. In particular, we examined closely a calculation that is expensive in both finite element and SPH, which is a bucket sort followed by a search for neighboring segments or particles.

In finite element method, the bucket sort can be performed infrequently, perhaps every 200 program cycles. SPH particles must be sorted more frequently, often once every program cycle. In the original LS-DYNA SPH, the bucket sorting algorithm was not vectorized. We modified the bucket sorting operations so that they could be vectorized by FORTRAN90/SX compiler. The ordering produced by the vectorized sort is identical to the original ordering.

The search for neighboring segments or particles is time-consuming for both finite element method and SPH method. In LS-DYNA the finite element neighbor search is highly tuned for vector-parallel processing. The original SPH neighbor search was parallelized, but not yet vectorized. We modified the SPH neighbor search to match the highly tuned vector-parallel algorithms already used by finite element method.

Finally, we vectorized some SPH subroutines using simple transformations such as unrolling an inner loop and moving subroutine calls (which inhibit vectorization) to a separate loop.

The optimized subroutines greatly improved the performance of SPH. Figure 1 and 2 show the performance of SPH before and after the optimization effort. For the smaller model (Figure 1), the elapsed time required was reduced from 40.6 to 11.0 hours with four processors. With eight processors, the small model ran in 6.7 hours. For the larger model (Figure 2), the elapsed time was 43.7 hours using three processors, and 25 hours using six processors.

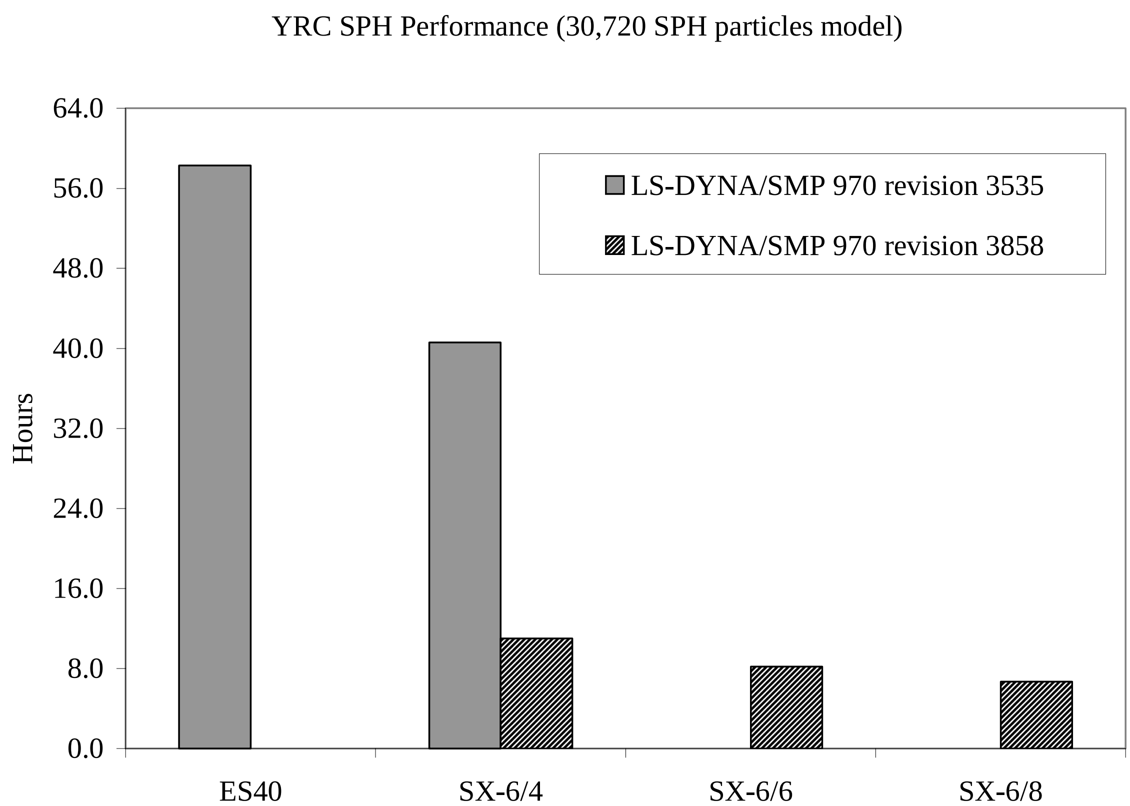

Figure 1

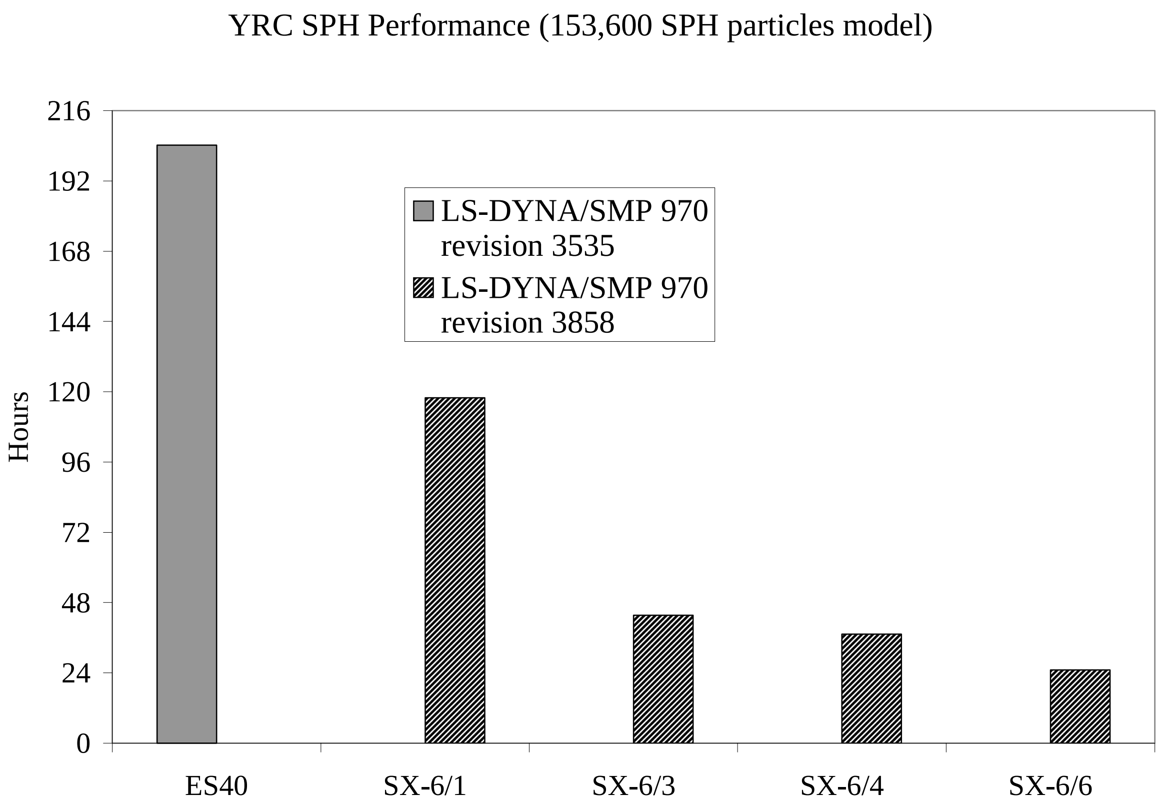

Figure 2

At the end of this SPH performance tuning project, we checked that the forces computed by the original and tuned SPH subroutines were not significantly changed.

Conclusion

The LS-DYNA code modifications from this joint project resulted in greatly reduced execution times for all simulations using SPH method on SX-6 vector-parallel supercomputer. The improvement goals based on two tire simulation models were achieved. The optimizations have been incorporated into newly released versions of LS-DYNA, revision 3858 and later.