Using a Rolls-Royce representative engine model to evaluate scalability of LS-DYNA thermal solvers

Gunther Blankenhorn[1], James Ong[2], Jason Wang[1], Roger Grimes[1], Francois-Henry Rouet[1]

[1] Livermore Software Technology Cooperation
[2] Rolls-Royce

1 Introduction

In the Finite Element Modeling community there is a trend to use models with increasing modeling details which raises the numbers of elements and solution variables. The increase in solution variables has a big impact on the run time of the analysis. Reducing wall clock time is an important item in using numerical analysis in production.

The wall clock time can be reduced by using improved CPU technology and hardware with a higher throughput and lower latency for memory, storage and interconnect.
On the software side, the use of parallel models to utilize more cores in an analysis reduces the wall clock time. Key measure for reducing wall clock time is scalability, which is in general expressed as the reduction of the run time due to an increase of cores used for the analysis.

LSTC is currently offering LS-DYNA in three different parallel models, namely shared memory parallel (SMP), massive parallel processor (MPP) and the combination of both models (HYBRID). The focus on these developments is scalability for all three parallel models.

Scalability is influenced by several factors. Beside the already mentioned hardware environment, main contributors are the decomposition (MPP and HYBRID) of the model, the model size and application type.
Scalability can not only be evaluated on a global implementation level. It needs to be evaluated on the application at hand and the features utilized in this analysis.

This contribution discusses the scalability of thermal solvers offered by LS-DYNA MPP using a surrogate engine model from Rolls-Royce. Three thermal solver types are used with three different MPP rank count (4, 8 and 16). The scalability is measured using the wall clock time summary of the LS-DYNA runs found in the d3hsp files.

2 The Rolls-Royce Representative Engine Models

The Rolls-Royce Representative Engines are a family of LS-DYNA models used in a joint project of Rolls-Royce, Cray, NCSA and LSTC to evaluate scalability with state of the art hardware and implicit solver technology.
The original models are surrogate engines, hence the name "Representative" and are used for structural (implicit dynamic) and eigenvalue analysis.
The underlying CAD geometry is the same for all models. The models are differing in the spatial discretization. They consist of solid elements and have node counts of up to 67 million nodes [1].

These Rolls-Royce Representative Engine Models are heavily utilized in the development processes at LSTC. They are used in this contribution to evaluate scalability of thermal implicit analysis. Fig. 1 depicts a cross section of the model.

The model used for the thermal scalability analysis is the model with approx. 67 million nodes. This model has temperature boundary conditions applied on the surface and initial temperatures are assigned to the internal nodes. A total for 21 time steps are calculated to evaluate scalability.

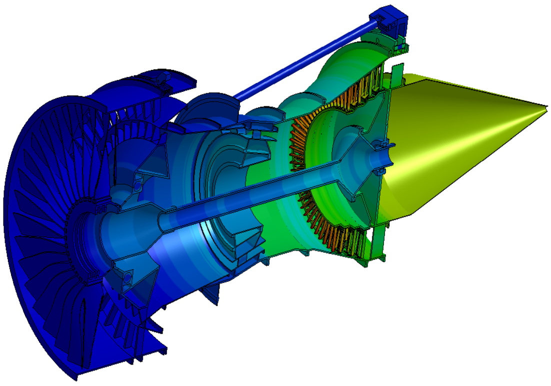

Fig. 1: Rolls Royce Representative Engine Model

3 Thermal solvers

A substantial cost for the thermal solution in LS-DYNA is the cost of solving the associated linear system.

Compared to a structural analysis in three dimensions, where each node has three degrees of freedom, the thermal solution only has on degree of freedom per node, namely the temperature. This leads to a significant reduction of the number of unknowns to solve for in a thermal analysis.
Furthermore the matrices associated with the thermal problem are usually having lower condition numbers compared to matrices resulting from structural problems. This makes them suitable for different kind of numerical solvers.

Currently LS-DYNA is offering one direct solver and an iterative solver, Conjugate Gradients, for the thermal feature. The iterative solver is usually the solver of choice for thermal problems. It usually converges fast enough to outperform the direct solver, and requires less memory.

LS-DYNA offers five different preconditioners that can be used to accelerate convergence of the iterative solver, thermal solver options 12 through 16. We recall that preconditioning means transforming the original linear system A x = b into M A x = M b. If M is a good approximation to the inverse of A, then the latter system is easier to solve and Conjugate Gradients will converge faster. Preconditioners can range from simple techniques like diagonal scaling (M is the inverse of the diagonal of A) to sophisticated approaches likes incomplete factorizations and algebraic multigrid. The goal is to find a good trade-off between the cost of computing M and the number of iterations needed to solve the preconditioned system.

In this study three preconditioners are utilized:
- Type 12: Diagonal scaling
- Type 13: Symmetric Gauss-Seidel
- Type 14: Symmetric Successive Over-Relaxation

In the MPP implementation, these preconditioners are communication-free. They are applied to the local subproblem that a process owns. This means that iteration counts might increase with the number of processors, but it allows for parallel efficiency. A global preconditioner (where processors communicate to compute a preconditioner for the whole matrix) would maintain constant numbers of iterations but would increase communication volume and might limit scalability.

4 Scalability Study

4.1 Setup

The study was performed with an LS-DYNA MPP development version (revision 133389). The three thermal solver types 12, 13 and 14 were tested. A total of three runs where made per thermal solver type, utilizing 4, 8 and 16 MPP ranks. The Platform MPI product was used for compute node communication. The compute nodes were equipped with 2 Intel® Xeon® E5-2690 v4 CPU's (14 cores each) and have 512 GB of memory.

The timings were taken from the d3hsp file. The runs were performed with a pfile setting (see [2], Appendix O), which uses the local scratch discs. The standard LS-DYNA decomposition method was used.

4.2 Results

4.2.1 Wall clock time and iteration counts

The wall clock time for the three solvers and the three MPP rank counts are depicted in Fig. 2. This timing is extracted from the d3hsp files and includes the time spent from the start of LS-DYNA until the termination.

All solvers scale, meaning the wall clock time reduces with an increase of MPP ranks.

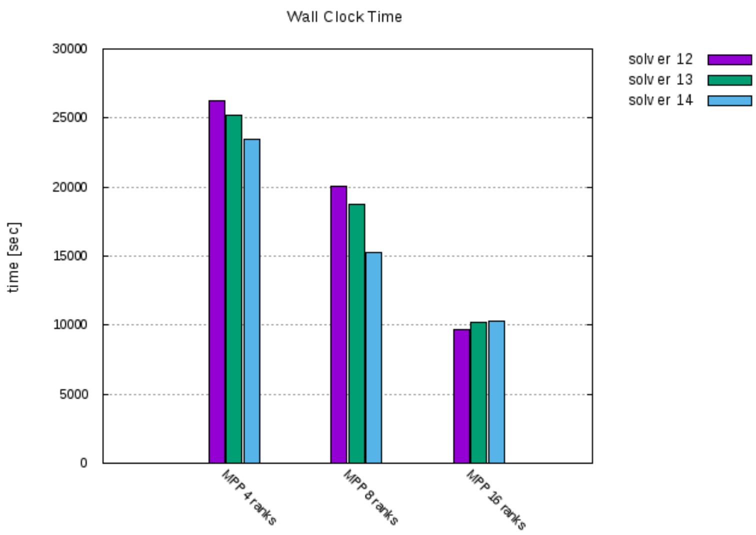

Fig. 2: Wall clock time

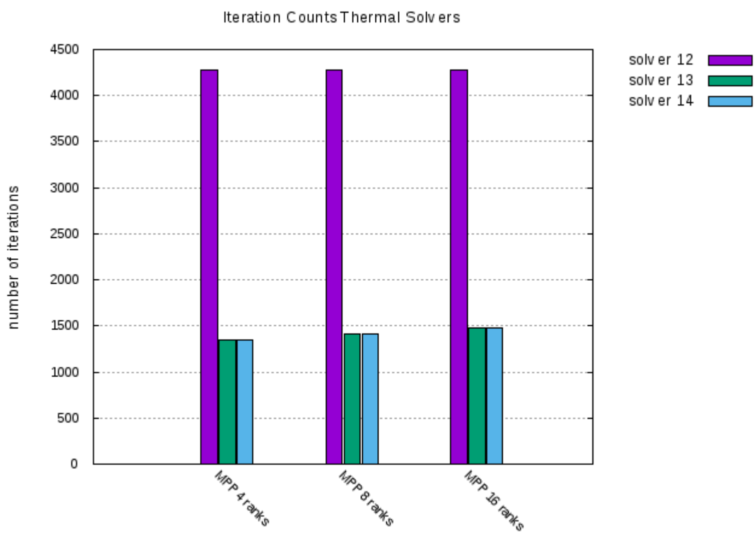

Fig. 3: Iteration counts for thermal solvers

For 4 and 8 MPP ranks, the analysis using thermal solver type 12 is slower than the analysis using thermal solver type 13 and 14. The analysis using thermal solver type 14 is faster than the analysis done with thermal solver type 12 and 13.
For 16 MPP ranks the timings are inconclusive. The analysis with thermal solver type 12 is faster than the analysis with the two other thermal solver types.

To a certain extent, the speed up of the analyses can be explained by the total number of iterations done for the different analyses. The iteration counts are summarized in Fig. 3. Thermal solver type 12 needs overall 4281 iterations, whereas thermal solver type 13 and 14 needs between 1342 and 1472 total iterations.

4.2.2 Wall clock time contributors

The computational cost contributors to the wall clock time are displayed in Fig. 4. The overall wall clock time is separated in five main groups and their subgroups:

- Group “Keyword Processing” summarizes the time spent to process the keyword input including:
  - “KW Reading” is the time spent to read in the keyword input file(s).
  - “KW Writing” is the time spent to write the structured input file and structured lsda files.
- Group “MPP Decomposition” summarizes the time spent for the decomposition of the input including:
  - “MPP Init Proc” is the time spent to read structured input and lsda files and to write decomposition database.
  - “MPP Decomposition” is the time spent to decompose the model and to setup ownership information for nodes and elements.
  - “MPP translation” is the time spent to write processor’s structured input and lsda files.
- Group “Initialization” summarizes the time spent to initialize the decomposed input including:
  - “Init Proc Phase 1” is the time spent to read processor’s structured input and lsda files.
  - “Init Proc Phase 2” is the time spent to set the initial conditions, i.e. initial velocities, initial stress, etc.
- Group “Thermal” summarize wall clock time in the thermal feature including:
  - “Thermal solver”: is time spent to solve the thermal problem.
  - “Thermal Element”: is time spent to calculate the contribution of each element to the global matrix and assemble this matrix.
  - “Thermal Other”: is time spent in thermal routines not associated with the thermal solver or the thermal element calculations.
- Group “Other” summarize the time which is spent in various routine and do not contribute to the above four groups.

The wall clock time for the group “Keyword Processing” and “MPP Decomposition” should be constant. The keyword processing is a serial operation and the decomposition operates serial or in the later stage in parallel, but on the same amount of data (complete input) for each MPP rank.
The discrepancy between the three runs where not closely investigated, but the assumption was that disk I/O from external processes can easily pollute this measurement.

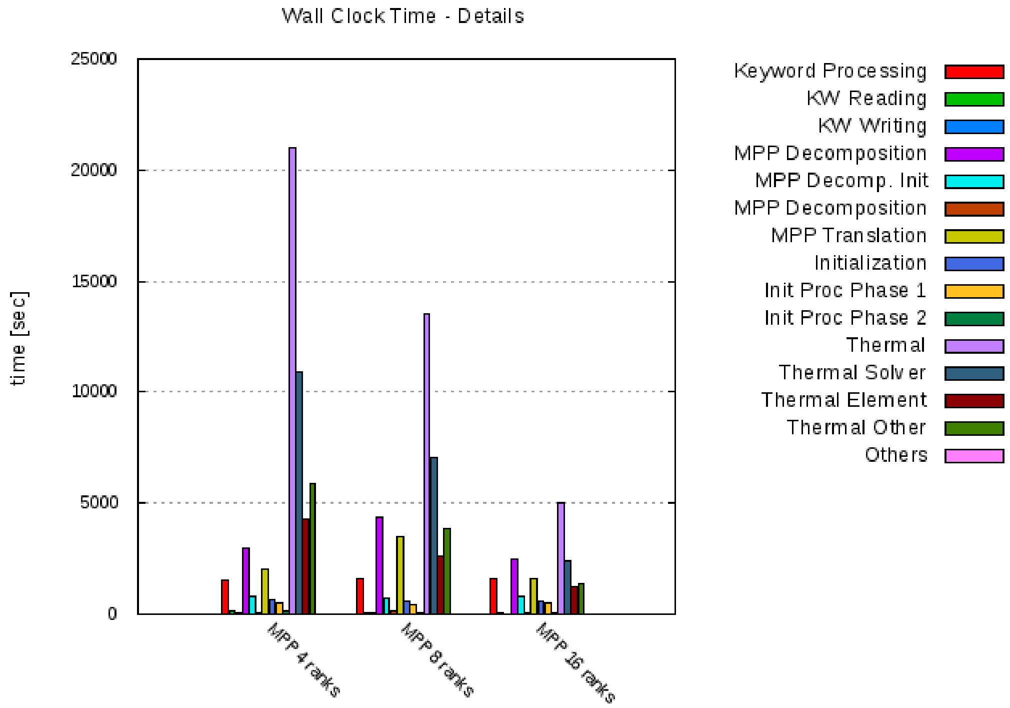

Fig. 4: wall clock time details – thermal solver type 12

4.2.3 Wall clock time thermal feature

Main focus of this study is on the scalability of the thermal feature. In the following, the contribution of processing the keyword, the MPP decomposition and the initialization are not further investigated.

Fig. 5 shows the overall time spend in routines for the thermal feature. Scaling can be easily observed. Again, the analysis with the thermal solver type 12 slightly outperforms the analysis with thermal solvers type 13 and 14 when used with 16 MPP ranks.

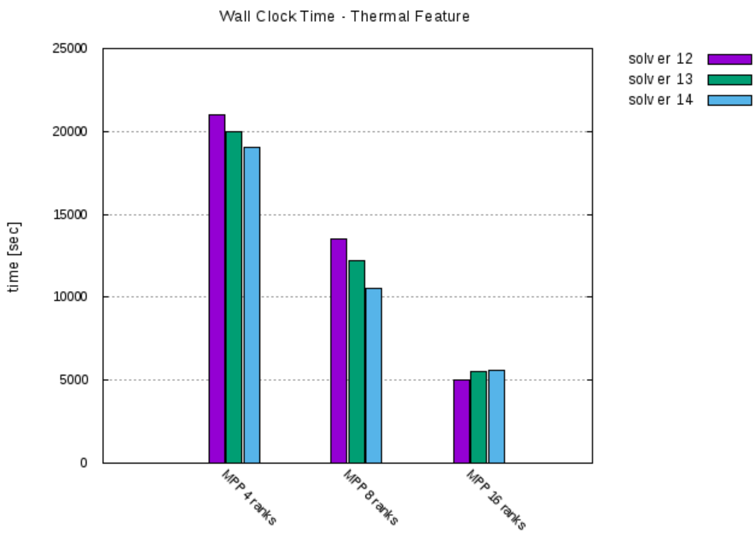

Fig. 5: Wall clock time - overall time spend in thermal feature

The timings for the thermal solver are depicted in Fig. 6. This is the time spent in the Preconditioned Conjugated Gradient algorithms. Thermal solvers 13 and 14 have an advantage here. Thermal solver type 14 is the fastest overall in this study.

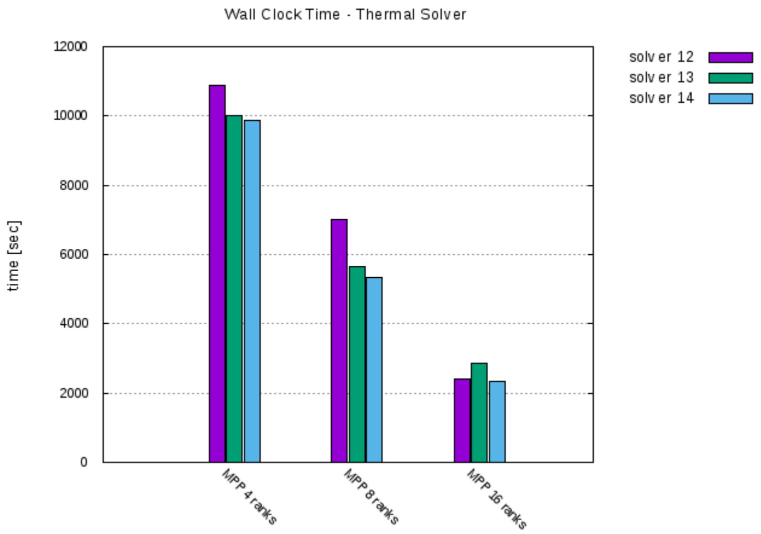

Fig. 6: Wall clock time details – time spent in thermal solver algorithm

The time spent in the thermal element routines are shown in Fig. 7. Scaling for this component can be seen as well. Analyses using 16 MPP ranks show the same wall clock time for thermal solvers types 12, 13 and 14.

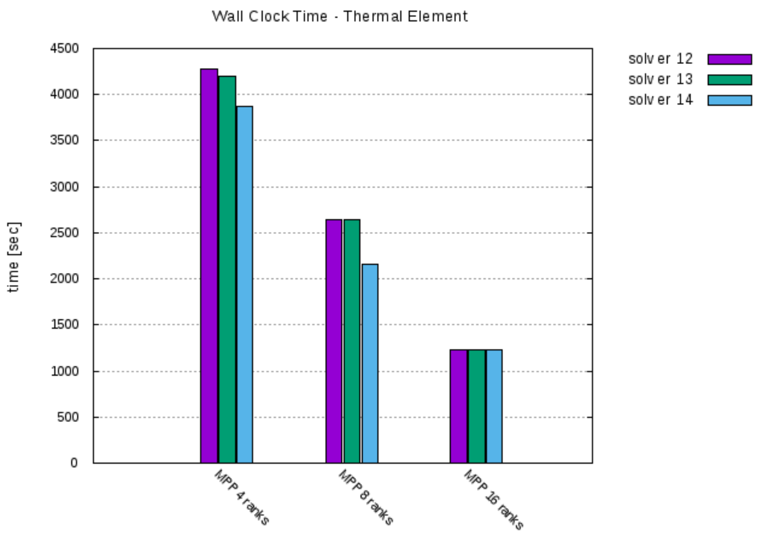

Fig. 7: Wall clock time details – time spent in thermal element routines

5 Summary

We reviewed the performance of the linear equation solvers based on Preconditioned Conjugate Gradients with three different preconditioners applied to the Thermal Solution Module in LS-DYNA. We used the Rolls-Royce Representative Engine model with 67 Million nodes as the basis for our testing and tested on 4, 8, and 16 MPI ranks. We were able to demonstrate good scalability for the linear equation solvers.

The thermal analysis compared to the structural implicit dynamic analysis is less mathematical challenging [1]. Thermal solver type 12, which is the most basic solver in this line up, converges in 4281 iterations for 21 load steps. These iteration counts are fairly low for a problem this size and complexity in geometry. Thermal solvers type 13 and 14 are more evolved and converging in 1342 - 1472 iterations for 21 load steps.

The results are also promising regarding a coupled thermal mechanical analysis of the Rolls-Royce Representative Engine model. The thermal calculation costs are low compared to the computational costs of the implicit dynamic calculation. Therefore the thermal calculation comes with a small overhead for the coupled analysis.

Finally, we should mention, even when thermal solver type 14 is the fastest option in this study, it might not be necessarily the best option for other thermal problems. Iterative solvers and preconditioners are in nature somewhat unpredictable. Thermal solver type 13 and 14 use preconditioners that are less accurate when one increases the number of MPI ranks; the goal is to maintain something as fast as possible. If there are concerns about the solution obtained with these solvers, the direct solver (thermal solver type 11) should be used to check against the results.

6 Literature

[1] Ashcraft C. et al., "Increasing the Scale of LS-DYNA® Implicit Analysis", 15th International LS-DYNA® User Conference, 2018

[2] "LS-DYNA® Keyword User's Manual", Volume I, LS-DYNA R11, 2018, Livermore Software Technology Cooperation