Get Your HPC Experience Even “Faster”

Maple SUN，Shawn LIU

Bosch (China) Investment Ltd., 215000

Abstract Nowadays, high performance computing (HPC) is getting more and more popular in engineering simulation. In some urgent project or tasks, simulation engineers would like to get the simulation result as soon as possible, in order to reach that, some of simulation engineers just increase the number of the CPUs in the parallel computing. As a HPC engineer I would say that, increase the number the CPUs would definitely decrease the wall time (physical time) used to finish the simulation task, but on the other hand, with the increase of the number of CPUs, the time consumed on communication between each compute nodes will also increase. To the balance the time saved with more CPUs and time consumed on communication between compute nodes, we conduct the test. The result shows that, compared the original compute resource distribute mode, with the optimized distribution mode, we could finish the task with only half compute resources as it’s used before, in some distribution mode, the compute resource saving rates are more than 50%.

Keyword High performance computing, Speedup ratio optimize, Parallel computing, Ansys mechanical

1 Introduction

Imagine you got a simulation task that has to be computed repeatedly, dozens of times per week, with various conditions or inputs. Would it be joyful if it is possible to make each of the computation more efficient, and in total save much more time, (as we all know time is money). Then please check out what we worked together with our AE HPC customers, on how to make their HPC experience even "faster".

A common understanding about using parallel computing (e.g. HPC) techniques for simulation tasks is like, when allocating more compute resources (CPUs), simulation job gets parallelized into more parts, and thus speed up solving processes. However, the fact is, for most of cases, the speed won't scale out linearly, which means even at some certain points, more compute resources lead to slower computations.

This "turning point" varies from case to case, with different hardware, simulation applications (solvers), and simulation jobs setups. Yet still, one will definitely benefit from figuring out these features with frequently repeated cases like the one we show in the following.

2 Methodology

In this section, we will give detail information about the model and the method we used in the test.

2.1 Introduction of the case

a. Simulation tool: ANSYS Mechanical v18.2

b. Module: electronic package with high level of details, using workbench mechanical module.

c. Material: both linear and non-linear

d. Loads: both thermal and mechanical, the temperature cycling with 26 steps and more than 5 sub-steps for each step

e. Element: about 1.71 Million nodes in total and over 1.2 Million nodes are non-linear material

The simulation engineer normally uses 200 CPU cores with 20 cores per node for this simulation job on Bosch HPC system in Germany, which takes about 125 minutes ("baseline time") for a single job, and at least 15 similar jobs per month.

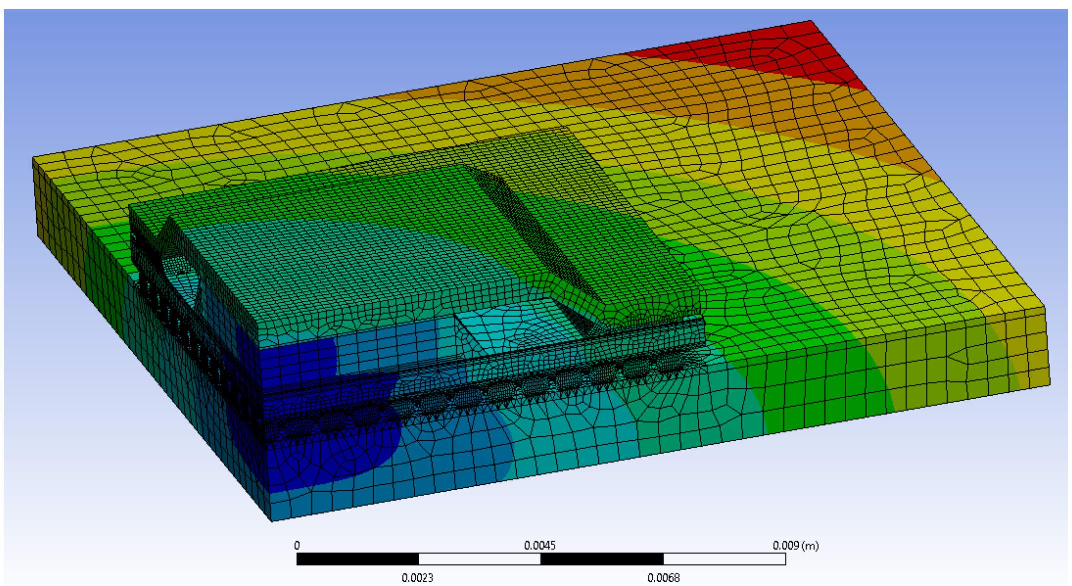

Figure1 Test case model

2.2 The test conditions

With the current setup of our HPC system, 28 CPU cores on each compute node, 2 sets of test conditions were designed.

a. CPU cores ranges [20 ~ 200], with 20 cores per node, on ANSYS v18.2

b. CPU cores ranges [28 ~ 364], with 28 cores per node, on ANSYS v18.2

And following analyses were conducted to the test results.

a. Parallel scalability analysis measured with "speedup ratio".

b. Total compute resource consummation analysis

c. "Resource saving rate" analysis

* Speedup ratio = Single node job run time / Multiple nodes job run time

* Resource saving rate = ("Baseline" CPU time - test CPU time) / ("Baseline" CPU time)*100%

3 Results and Discussion

In the following, we will discuss some of the main findings from the view of speedup ratio, total CPU time consumed and compute resource rate. (Here we only list out the results of using 28 CPUs on each node.)

3.1 Speedup ratio

Observed from the speedup ratio curve, the speedup ratio increases with the increase of the cores, but meanwhile, the gap between the speedup ratio of test results and the linear scaling is becoming larger. Therefore, the number of 84 and 112 cores are considered efficient already, comparing with the user "baseline" (in green).

Figure2 Speedup Ratio Curve with 28 cores per node

3.2 Total CPU time

While the "speedup ration" only mainly focus on the computation speed efficiency, the users also care about the total "CPU time" (number of CPU cores multiplied by job run time), which directly relates to the money paid for the HPC resources, a.k.a. User Based

Resource (Unit). Therefore, when putting the best test condition (28 cores per node) together with the user baseline, it's easily noticed that the HPC resource can be saved up to 50% (halved) with "core distribution as 28 cores per node and total number of cores between 56 to 112".

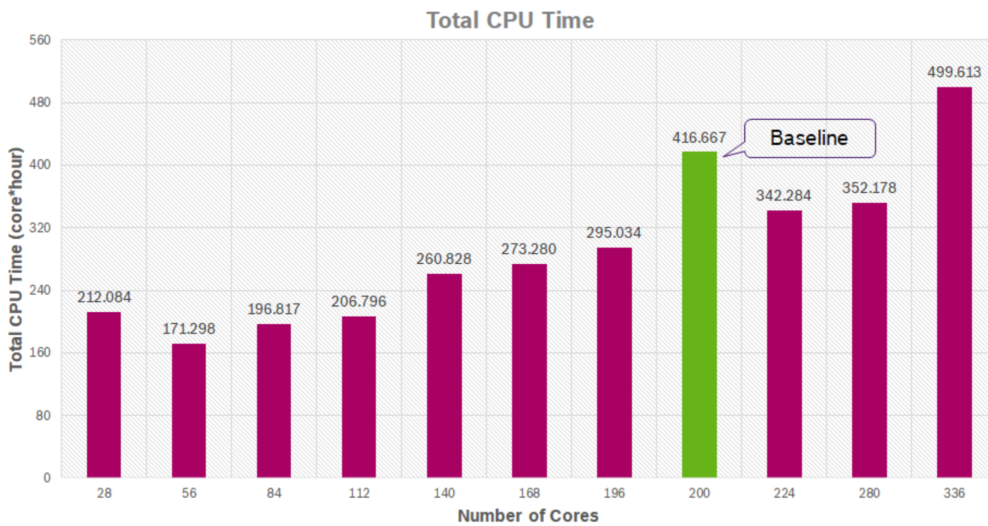

Figure3 Total CPU Time with 28 cores per node

3.3 Resource saving rate

In the first two parts, we discuss the compute efficient and compute resource consummation, and here we'll have a deep view on the resource saving rate, which is the most important part of this research. In the fig. 4, with the increase of the CPUs, the resource saving increase slightly and then decline with the increase of the CPUs. It's easily noticed that the HPC resource save rate can up to 50%. This is due to with the increase of the CPUs, the time consumed on the communication between each CPUs increase, that why we do not recommend to increase the CPUs to shorten the wall time (physical) for some simulation tasks.

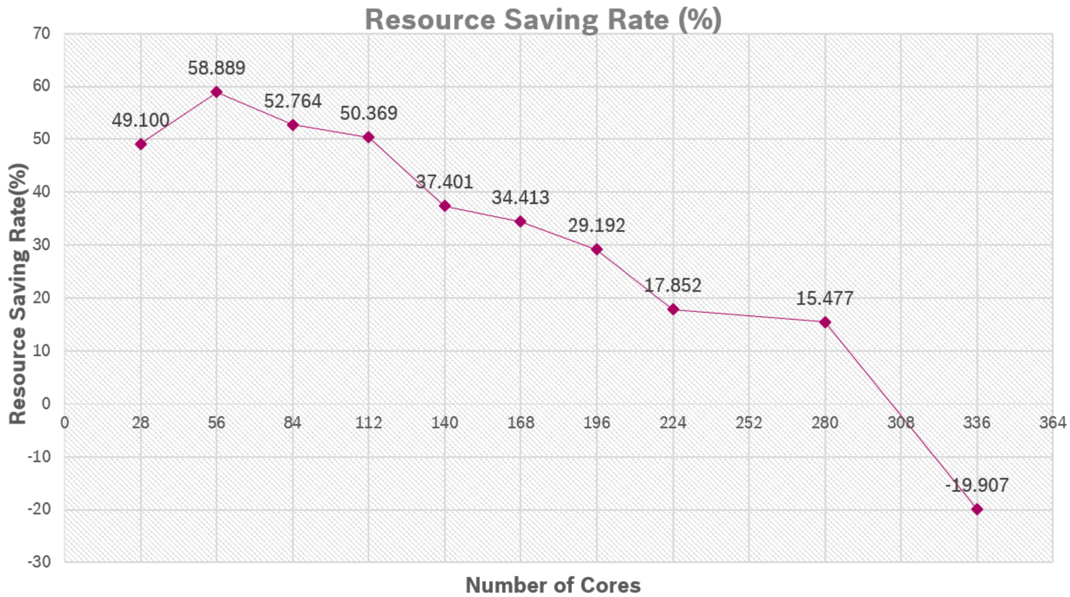

Figure4 Resource saving rate with 28 cores per node

4 Conclusion

Based on the data we get from the test and the comparison with the resource distribute mode the simulation engineer used, here is the main conclusions:

a. Properly distributed resource gets higher computation speed with less CPU time. With suitable compute resource distribute mode, the resource saving rate can more than 50%.

b. For this case, the recommended change is to use 84 or 112 cores, with 28 cores per node (use whole compute node instead of part of the resource on each node).

Please note, since different simulation problems use different software and different function modules. Or even for the same problem, different cases have different geometry, material, element and so on. The most frequently repeated simulation tasks run on HPC cluster would benefit the most for this kind of investigations/tests.

Reference

[1] Workbench mechanical , https://www.ansys.com/zh-cn

[2] Cell Phone Impacting Floor, https://www.simulia.com/