

# Usage of multiple CPUs and cores, and influence on solution time

Article ID: x385

Solution ID: S-018633

Product(s): SOLIDWORKS Simulation

Version: All Versions

Category: Analysis

Created: 06/21/2007

Last Revised: 11/06/2017

## Discussion

This article presents the advantages of using several CPU cores versus one, for different versions of SOLIDWORKS Simulation, different types of studies, and different areas of the program.

The following formula was used to compute percent improvement in time to solve when comparing the use of x CPUs versus only 1:

$$\% \text{ improvement} = (\text{Solution\_Time\_with\_1\_CPU\_Core} - \text{Solution\_Time\_with\_x\_CPU\_Cores}) / (\text{Solution\_Time\_with\_1\_CPU\_Core})$$

### Note:

- To change the number of CPU cores associated with the SOLIDWORKS process, you don't need to close and reopen SOLIDWORKS. Simply change the number of desired processor for sldworks.exe in **Task Manager, Processes, Set Affinity**. The model can stay open.
- Changing the **Affinity** of SOLIDWORKS also changes the **Affinity** of the solvers that will be launched.

## Use of multiple CPU cores in different parts of the program

### Overall behavior

Area of the program	Hyper-Threading	Multi core/processor
User Interface	✗	✗ <sup>(1)</sup>
Mesher (Standard)	✗	✗
Mesher (Curvature based)	✗	✓ <sup>(2)</sup>
Solver – Direct Sparse	✗	✓ <sup>(3)</sup>
Solver – FFE+	✗	✓ <sup>(3)</sup>

(1) Animation of plots is multi threaded

(2) Before version 2011: Only parallelization of volume filling for solid bodies - surface mesh still single threaded. In 2011 the surface meshing now supports multi-threading. It has the most benefit for models with multiple parts. This is where the most time reduction will occur. One can expect a 60% mesh time reduction on average with 4 CPU cores compared to a single CPU core.

(3) Depends on the program's version and the study type. See below.

### Solver behavior

Stage of the FFEPlus solver	#of CPU cores used	Comment
-----------------------------	--------------------	---------

Forming surface to surface bonding	1	
Establishing stiffness matrix	Some portions of this stage 1 Core, others multi-core	40% to 60% with 4 CPU cores
Solving contact constraints	Multi core, but not 100%	40% to 60% with 4 CPU cores
Iterations	Multi core, but not 100%	40% to 60% with 4 CPU cores
Stress calculation	1	

Stage of the Sparse solver	#of CPU cores used	Comment
Forming surface to surface bonding	1	
Establishing element connectivity	1	
Establishing stiffness matrix	1	
Decomposition of Stiffness matrix	Multi-core	100%
Solving contact constraints	Depends	Some portions of this stage use one core, others are multi-core at 100%
Stress calculation	1	

Stage of the LPDS solver	#of CPU cores used	Comment
Forming surface to surface bonding	1	
Establishing element connectivity	1	
Establishing stiffness matrix	1	
Decomposition of Stiffness matrix	Multi-core	100%
Solving contact constraints	Depends	Some portions of this stage use one core, others are multi-core at 100%
Stress calculation	1	

The most computationally intensive stages of the analysis using the Direct Sparse and LPDS solvers are generally **Decomposition of Stiffness matrix** and **Solving contact constraints**. These are the stages which support multi-core, hence making them less time consuming.

### **Influence of multiple CPUs on solution time**

#### **Version 2007**

Here is a table that shows the time savings obtained by using 2 CPU cores for an analysis instead of 1 on a Dell Precision 650 machine, for each solver. It was performed using version 2007 on a model with following statistics:

Study Type	% Improvement in Time to Solve		DOF	Model Used/Notes
	Direct Sparse	FFEPlus		
Static	52%	13.5%	91812	Solid mesh assembly with bolt connectors and node-surface contact conditions

Table 1: Solution time comparison in version 2007

As you can see, the use of multiple CPU cores yields a greater relative time gain with the Sparse solver than with the FFEPlus solver. With the Sparse solver, solution time is divided by 2 when using 2 CPU cores instead of 1. For the FFEPlus solver, the time gain is marginal.

In this version, multiple CPU cores are only supported for Static analysis. The other types of analysis (frequency, thermal, nonlinear, advanced dynamic, etc. only use a single CPU core)

### Version 2009

The table below shows the Percentage improvement in time taken to complete solving when using 4 CPU cores compared to using only one CPU core. The data was obtained using test cases in Simulation 2009 SP2.1 on a Windows XP x64 Dell Precision T7400 with Quad Core Intel Xeon X5472 @ 3.0 GHz, 16 GB RAM.

Study Type	% Improvement in Time to Solve		DOF	Model Used/Notes
	Direct Sparse	FFEPlus		
Static (Test 1)	58%	13%	373437	Simple part, one load and restraint
Static (Test 2)	74%	22%	66057	Assembly with bolt connectors and contacts
Static (Test 3)	71%	22%	66057	Same as Test 2 but with large displacement enabled
Frequency	0%	25%	48855	Frequency analysis of a shaft, 20 frequencies
Buckling	0%	14%	122940	3 plates-surfaces model, 10 modes
Thermal (Test 1)	81%	0%	15345	Computer chip example, steady state
Thermal (Test 2)	82%	0%	15345	Same as above model but transient thermal
Nonlinear (Test 1)	58%	0%	48360	Elasto-plastic nonlinear static analysis, single part, no contacts
Nonlinear (Test 2)	62%	0%	48360	Same as above but nonlinear dynamic with base excitation

Table 2: Solution time comparison in version 2009

### Version 2012

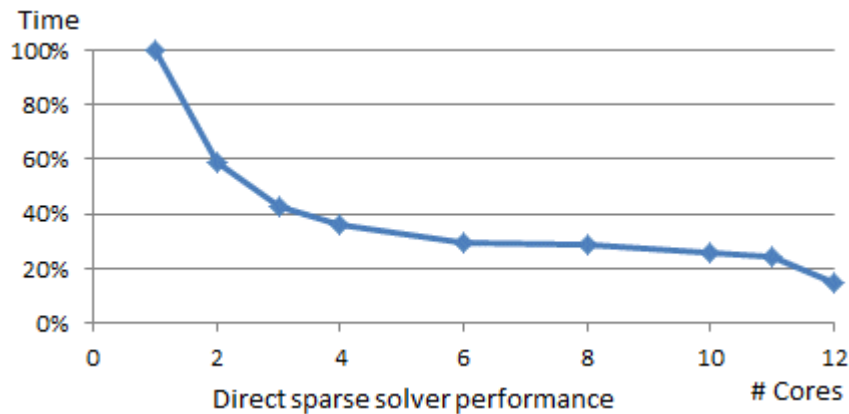
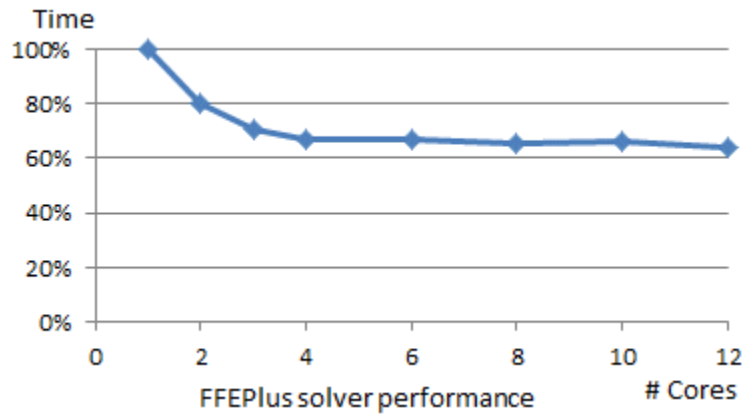
The table below shows the Percentage improvement in time taken to complete solving for several models when using 4 CPU cores compared to using only one CPU core. The data was obtained using

test cases in Simulation 2012 SP4.0 on a Windows 7 x64 Dell Precision T5500 with Quad Core Intel Xeon X5550 @ 2.66 GHz, 12 GB RAM.

Study Type	% Improvement in Time to Solve	
	Direct Sparse	FFEPlus
Static	72% ± 5%	35% ± 10%
Frequency	6% ± 1%	46% ± 1%
Buckling	6% ± 1%	53% ± 3%
Nonlinear	none	10% ± 10%
Dynamic	none	none
Thermal	70% ± 10%	none

Table 3: Solution time comparison in version 2012

Tests of static analyses on several machines with 12 CPU cores (dual 6-core Xeon) show increased performance when the number of used cores increases:



On some such machines, though, a decrease in performance has sometimes been seen when using 11 or 12 cores for the Direct Sparse solver only. In that case, the best performance is obtained when using 10 CPU cores. The process **Affinity** can be used to set the number of CPU cores. To automate this, a shortcut can be used:

1. Right click on the Desktop and select **New, Shortcut**

2. Browse to select C:\Windows\System32\cmd.exe and validate
3. Right click the shortcut and select **Properties**
4. In the **Target** field, enter this:

```
C:\Windows\System32\cmd.exe /c start /affinity FFC sldworks.exe
```

**Note:**

- FFC is the affinity code in hexadecimal that corresponds to the activation of the cores in binary. Imagine a 12 digit binary number (because we are assuming here a 12-core machine), where the usage of cores is represented by the status of each digit. The 1st core is the unit digit (right), the 2nd core is the digit of tens, etc. Notice FFC (hex) = 111111111100 (bin). This corresponds to not using the first two CPU cores, and using the last CPU 10 cores. This rule can be generalized to any number of CPU cores.

5. In the **Start In** location, enter within quotation marks the path to the folder where the sldworks.exe file is located. For instance:

```
"C:\Program Files\SOLIDWORKS 2012\SOLIDWORKS"
```

6. When you launch SOLIDWORKS using the shortcut, it will automatically have the affinity you selected, and the affinity will be transferred to the solvers for this session.

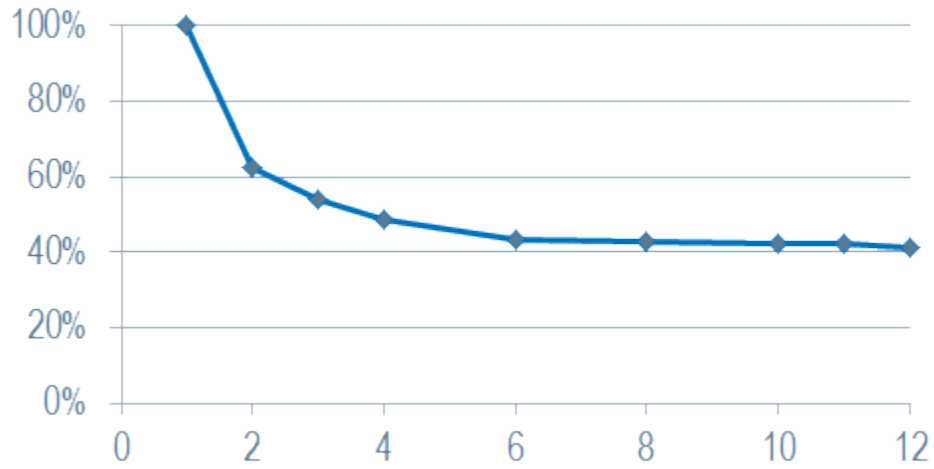
**Version 2014**

The table below shows the Percentage improvement in time taken to complete solving for several models when using 4 CPU cores compared to using only one CPU core. The data was obtained using test cases in Simulation 2014 SP0.0 on a Windows 7 x64 Dell Precision T5500 with Quad Core Intel Xeon X5550 @ 2.66 GHz, 12 GB RAM.

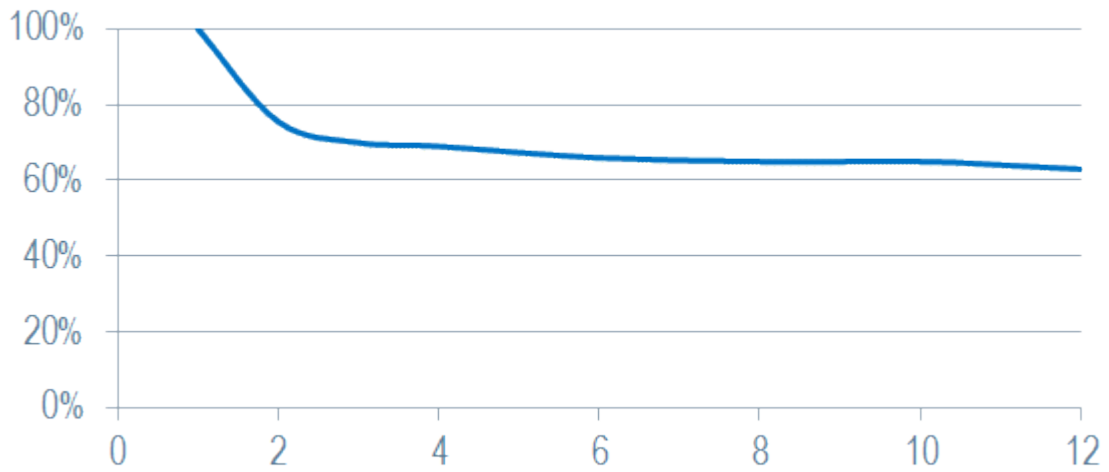
Study Type	% Improvement in Time to Solve		
	Direct Sparse	FFEPlus	LPDS
Static	40% ± 25%	42% ± 10%	38% ± 10%
Frequency	6% ± 1%	58% ± 2%	N/A
Buckling	6% ± 1%	35% ± 3%	N/A
Nonlinear	10% ± 10%	26% ± 15%	26% ± 10%
Dynamic	0% ± 5%	0% ± 5%	N/A
Thermal	30% ± 20%	14% ± 2%	N/A

Table 4: Solution time comparison in version 2014

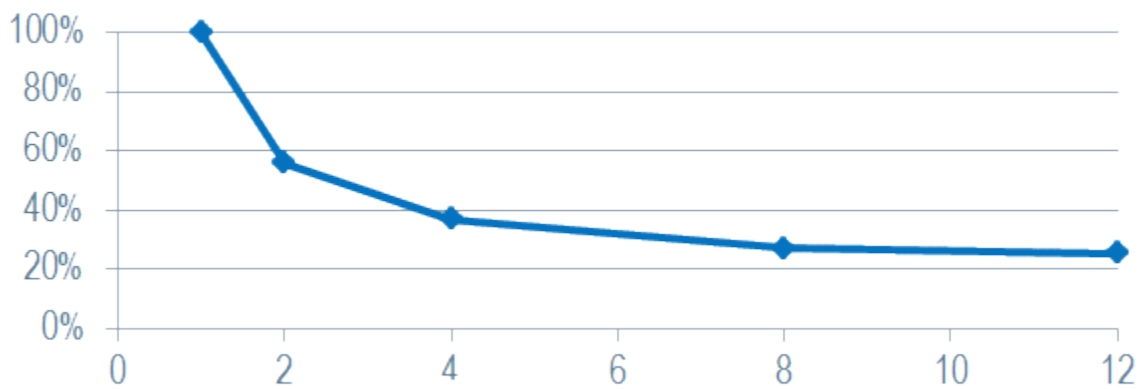
Tests of static analyses on several machines with 12 CPU cores (dual 6-core Xeon) show increased performance when the number of used CPU cores increases:



FFEPlus Solver performance vs number of cores



Direct Sparse Solver performance vs number of cores



LPDS Solver performance vs number of cores

Other Study Types:

**Drop Test**

Only one solver type available, which only uses one CPU core.

## **Fatigue**

Only one solver type available. Fatigue solver itself uses only one CPU core but preparing to run a fatigue study involves setting up and running one or more static studies. Since static studies do benefit from multiple CPU cores, users doing this type of analysis would see an overall improvement in time to perform a fatigue analysis on a multi-core machine.

## **Optimization**

Most of the time spent solving an optimization analysis is taken up by running loops of designs iterations of the studies defined for constraints. In the current release, these constraints can be based on static, buckling, frequency, and thermal studies. Since the user can specify which solver type they wish to use for each of the constraint studies, performing an optimization analysis on a multi-core machine would show improvement in performance over using just a single CPU core machine.

## **Linear Dynamic**

The actual post dynamic analysis and stress calculations use special solvers which used only one CPU core in testing. However, performing a linear dynamic analysis involves first finding resonant frequencies, which did show usage of more than one CPU core when using the FFEPlus solver.

## **Pressure Vessel Design**

The majority of the time taken to complete a pressure vessel analysis is running the respective static studies that you wish to combine. The actual calculations for combination of results used only one CPU core during testing but it made up a small percentage of the total time perform the analysis. As a result, a user with a multi-core machine would see an overall improvement in time to perform a pressure vessel design analysis when compared to a single-core machine.

## **Conclusion**

Every analysis type with the exception of Drop Test is capable of showing an improvement in overall time taken to complete solving when a multi-core/CPU machine is used.

## **Notes**

- When looking at CPU usage in Task Manager, and for a machine with 4 CPUs, 100% indicates that all four CPUs are being used at maximum capacity. 25% indicates that one CPU is being used at full capacity.
- In general, the Iterative solver (FFEPlus) did not show usage beyond 60% which could imply full usage of two CPU cores and partial usage of a third.
- Direct Sparse showed the greatest improvement in performance during the matrix decomposition stage, where for most testing done the total CPU usage was 99%.
- Even for solver stages or study types which use only one CPU core, there is still a minor benefit to having more than one CPU core since it makes it possible to allocate an entire CPU core to the solver while leaving background applications and system processes to use their own separate CPU cores.
- Nonlinear test models showed usage of more than one CPU core only for certain stages of the Direct Sparse solver – during contact iterations for test model, total CPU usage stayed below 25%.
- Multistep analysis such as the nonlinear one can have significant differences in multi-core performance between models depending on geometry, setup, number of contacts, etc.