Maximizing Memory Performance for ANSYS Simulations

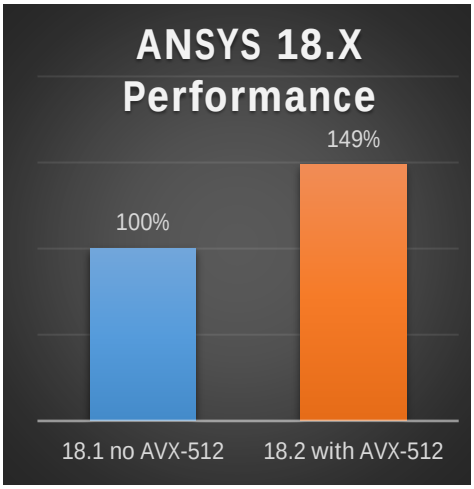
By Alex Pickard, 2018-11-19

Memory or RAM is an important aspect of configuring computers for high performance computing (HPC) simulation work. The performance of the computer is not only affected by the amount of RAM but also by the amount of bandwidth, or transfer speed, between the processor and the memory. The bandwidth is affected by the speed of the memory stick, but what few people know is that the way memory sticks are configured on the motherboard is even more important. CPUs are capable of accessing information on multiple memory sticks in parallel in what are referred to as memory channels. Modern CPUs can access between 1 and 8 memory channels simultaneously (model specific). We have found however, that many computers are not configured to make use of all their memory channels and this is creating a bottleneck in their performance. All ANSYS solvers require significant memory bandwidth to sufficiently feed data to the multiple CPU cores available on modern processors

This guide will review a recent test on how memory channels can impact your simulation performance and show how to best configure a new or existing system for maximum performance.

CPU Data Starvation

CPU "data starvation" is becoming an ever more common obstacle in HPC applications, where the number of cores on a CPU is constantly increasing, and the amount of work each core is able to perform each cycle is also increasing (See AVX, and AVX-512). This ability to process massive amounts of data requires the ability to provide said data to the CPU cores in a timely fashion, or else the CPU will sit idle waiting for work.



Intel and AMD have both been working to increase the memory bandwidth available regularly. For example, Intel has upgraded their highly successful Xeon-EP platform (our most commonly recommended solution) as follows:

- 3 channel DDR3-1333 in 2010
- 4 channel DDR3-1600 in 2012
- 4 channel DDR3-1866 in 2013
- 4 channel DDR4-2133 in 2014
- 4 channel DDR4-2400 in 2016
- 6 channel DDR4-2666 in 2017

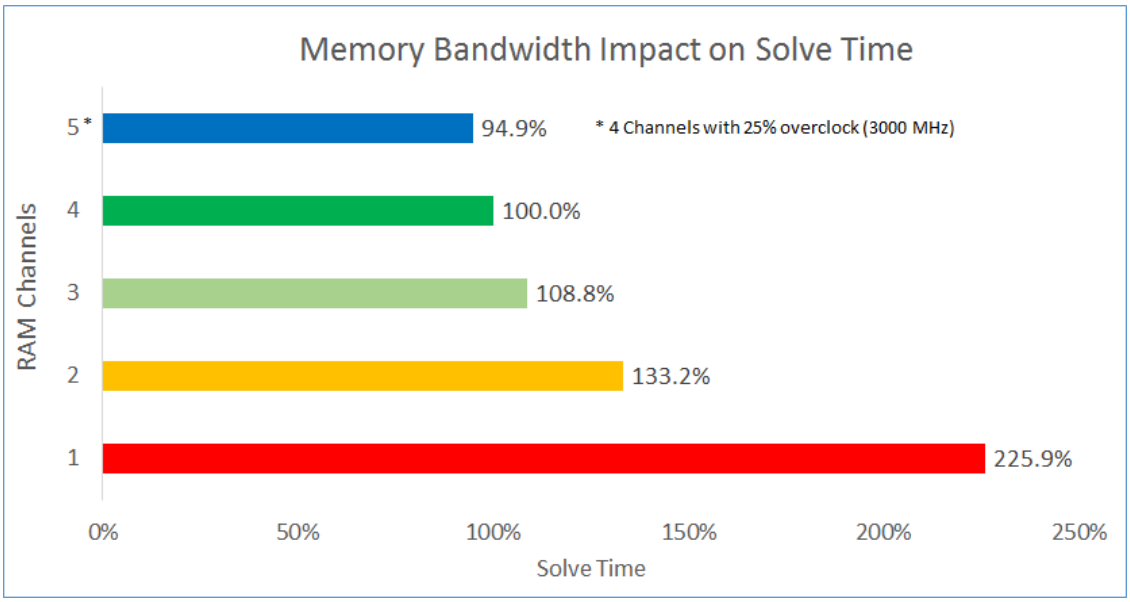
The increase is substantial, but in the same period of time the platform has moved from a maximum of 6 cores per CPU to now 28 cores per CPU, and also increased in both core frequency and operations per core cycle (AVX). The total memory bandwidth available per theoretical gigaflop of performance is clearly falling. There are also other system platforms available, such as the dual memory channel intel consumer CPUs (i5, i7 series, Xeon E) and the quad memory channel Intel "HEDT" platform (i7, i9 and Xeon W). Likewise, AMD has CPU models and platforms ranging from 1 to 8! memory channels.

Memory Bandwidth Impact on Constant CPU Performance

To test the effect of memory bandwidth on solver performance, a test case is setup where an identical CFX model is solved on the same system, but each successive test removes a stick of RAM, and thus a memory channel, from the system. The test model is small so RAM quantity is not an issue, and all other factors are kept constant (CPU cores, frequency , software version, CPU cache). The setup is as follows.

CFX Version	18.0
Model	perf_Pump (1.3M Nodes, 5.3M elements, see end of report)
CPU	i7-5960X @ 4.0 GHz
CPU Cores	8
RAM	DDR-2400 CAS14
Memory Ch.	4
OS	CentOS Linux 7
Solver	Intel MPI Distributed Single Precision

This system would be typically classified as having ample memory bandwidth, having only 2 CPU cores per DDR4 memory channel. The effect of varying number of memory channels on solve time is as follows.



Simply removing 1 stick of RAM makes the solver 9% slower, this would also be comparable to running 10-12 cores on 4 Channels. Removing 2 sticks, giving 4 CPU cores per memory channel, gives a

performance degradation of 33%. Going to the extreme, and running this computer with only a single RAM stick, shows the severe effects of CPU bandwidth starvation, the solve time is 226% of normal. This shows how critical it is that your computer be setup with optimal memory layout to take advantage of available memory channels. Buying 1 stick and "getting another one later" is not a plan for success.

Note that the 2 channel result, which is comparable to 16 cores on all 4 channels, does not necessarily make a 16 core CPU a poor choice. The 16 core CPUs that are available today run at lower frequency than 4.0 GHz, and have more memory channels, and are thus comparable to the 4 channel result above in terms of efficiency if configured properly. A minor bandwidth loss, but with many more cores, provides a better platform value than a cluster of comparable i7 machines. The performance per core will be lower, but due to system complexity, for users requiring up to 36 cores the recommended platform is a dual socket Xeon-EP system.

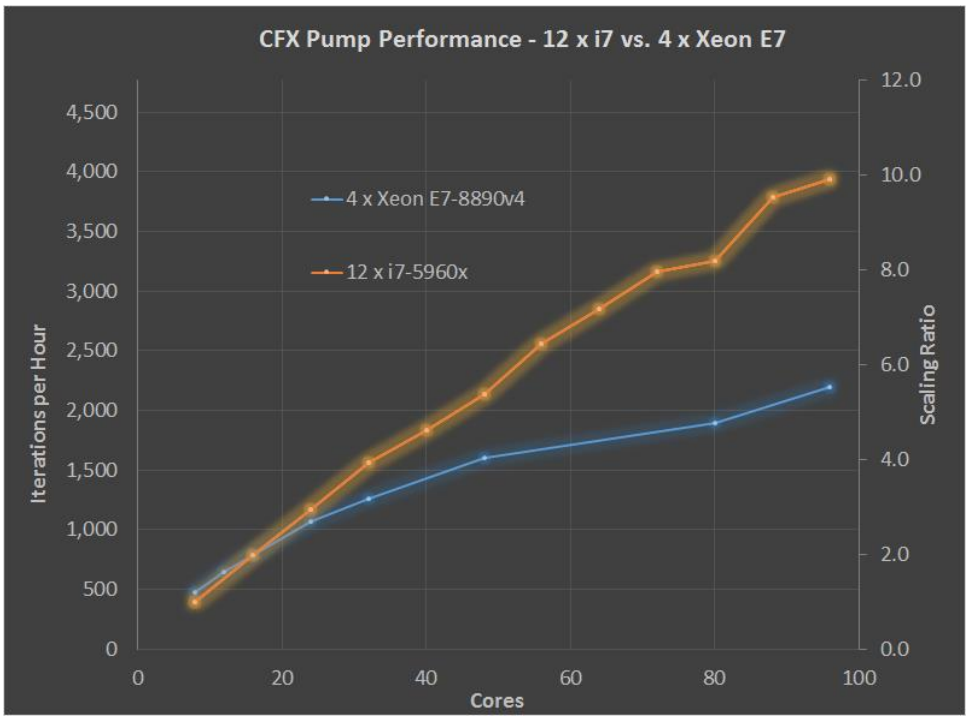
CPU Scaling in Constant vs. Increasing Memory Bandwidth Scenario

To demonstrate the effect of memory bandwidth on high core count scenarios, a very powerful, quad CPU, Xeon E7 system is compared against a cluster of 8 core Intel i7 machines.

CFX Version	17.2	17.2
Model	perf_Pump (1.3M Nodes)	perf_Pump (1.3M Nodes, 5.3M elements)
CPU	4 x Xeon E7-8890v4	12 x i7-5960X or 6900k (mixed)
Cores	4 x 24 cores (96 total)	12 x 8 cores (96 total)
RAM	4 x 4 channels DDR4	12 x 4 channels DDR4-2400
CPU Cache	4 x 60 MB (240 MB total)	12 x 20 MB (240 MB total)
Interconnect	Single Monolithic System	QDR Infiniband

The testing consists of ramping up the number of CPU cores being used. The Xeon E7 system will have all memory bandwidth and CPU cache available at any quantity of cores in use, and will be subdividing those finite resources as more CPU cores are assigned. By comparison, the i7 cluster will ramp up by adding more topologically identical nodes to the job. Thus, every time a CPU is added, so does an equal amount of memory bandwidth and core cache. This should show the true scaling potential of the solver, and will only be penalized by communication overhead and mesh overlap.

Please note that core frequency is not constant on the Quad E7 system. It is 3.4 GHz max at low core counts, but has a base frequency of 2.2 GHz. Turbo frequency decreases as more Xeon cores are added.



The quad Xeon E7 system initially starts out faster than the single i7 machine (both using 8 cores). This is despite the i7 having a faster core frequency (4.0 GHz vs 3.4 GHz max turbo). This is because the Xeon system has 4 times the memory bandwidth and 3 times the core cache available. In fact, it is using 16 memory channels to feed only 8 CPU cores. By the time 16 cores are in use the systems are effectively matched in performance, and from there on the i7 cluster is faster. Scaling is still good in the up to 48 core range on the quad Xeon system, which is effectively 3 CPU cores per memory channel, and then the performance beings to taper off more severely. In the end, the Xeon system ends up at only 56% of the i7 cluster's speed.

Another result of significant importance is that when adding additional nodes to a cluster, specifically ones that are topologically identical and add as much non-CPU resources as they do CPU power, the scaling power of the CFX solver is very impressive. CFX was able to solve this model 9.91 times faster on 12 machines than it could on 1, and that was with substantial mesh overlap (18.6% on average) and only 13.6 thousand nodes per CPU core, which is much lower than the recommended range for good scaling performance (30-50k). For 64 cores, where there were 20k nodes per core, results were even better at 90% scaling efficiency.

What can I do about it?

Memory channels are not necessarily the number of sticks in the computer, because it is possible to install 2 or even 3 sticks into a single memory channel. It is also possible to accidently install 2 sticks in channel #1 and no sticks in channel #2, thus having only channel #1 active and restricting bandwidth. Optimal memory setup is very important. Every CPU has its own dedicated memory, so a computer with 2 CPUs has twice the number of memory channels available as one with only a single CPU. To determine how many memory channels are available on your computer, search for your CPU model on the Intel ARK database.

Memory Specifications

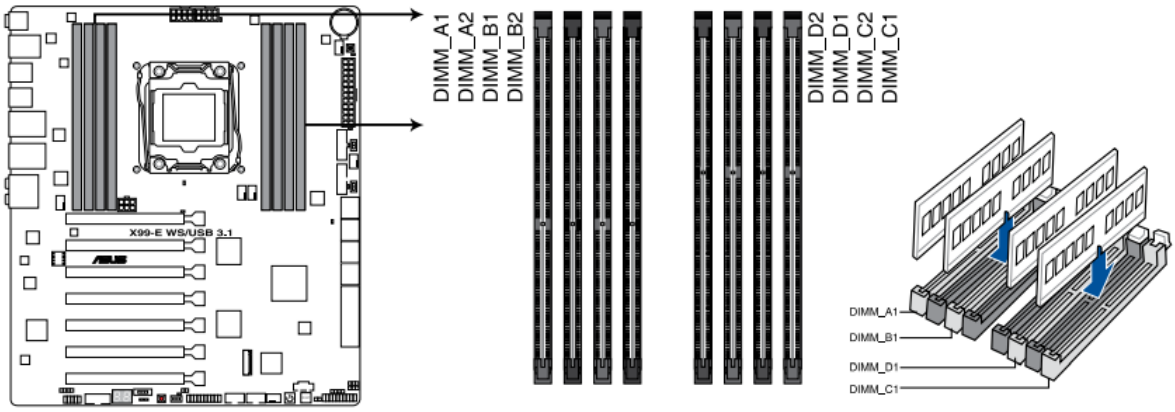
Max Memory Size (dependent on memory type) ?	128 GB	Memory Types ?	DDR4 2400/2133
Max # of Memory Channels ?	4	ECC Memory Supported ‡ ?	No

As you can see above, an example computer shown here has a maximum of 4 memory channels available. Make sure to put an identical stick in every channel, before moving on to putting a second stick in every channel. It is not necessary to have the first batch of 4 sticks match the second batch, but there should be one of each type in every channel to have a balanced configuration. Your motherboard manual will describe what positions to install the RAM in. A typical example of the relevant entry in the manual is shown below:

1.2.4 System memory

The motherboard comes with eight DDR 4 (Double Data Rate 4) Quad Inline Memory Modules (DIMM) slots.

A DDR4 module is notched differently from a DDR, DDR2, or DDR3 module. DO NOT install a DDR, DDR2, or DDR3 memory module to the DDR4 slot.



X99-E WS/USB 3.1 288-pin DDR4 DIMM socket

If you are unable to open your computer and check that the memory is balanced properly, there is a command you can run in the windows command prompt to check your memory layout.

```
wmic MEMORYCHIP get BankLabel,DeviceLocator,Capacity,ConfiguredClockSpeed
```

Output:

```
BankLabel  Capacity    ConfiguredClockSpeed  DeviceLocator
ChannelA   8589934592  1600                  ChannelA_Dimm1
ChannelA   4294967296  1600                  ChannelA_Dimm2
ChannelB   8589934592  1600                  ChannelB_Dimm1
ChannelB   4294967296  1600                  ChannelB_Dimm2
ChannelC   8589934592  1600                  ChannelC_Dimm1
ChannelC   4294967296  1600                  ChannelC_Dimm2
ChannelD   8589934592  1600                  ChannelD_Dimm1
```

```
ChannelD   4294967296  1600                ChannelD_Dimm2
```

This computer has an 8GB stick and a 4GB stick of 1600 MHz RAM in every channel, and is thus properly balanced.

Buying a New Computer

When selecting a new computer for simulation, we typically recommend a "High End Desktop" (HEDT) platform or Xeon EP (Xeon scalable, i.e. Xeon Gold etc.). The first is a 4 memory channel single CPU platform that accommodates 6 to 18 CPU cores (12 for use with 1 ANSYS HPC pack). The second is a 6 memory channel per CPU platform, which can accommodate up to 8 CPU's, but for ANSYS is frequently used in dual CPU, 12 total memory channel configurations targeting the 36-ish core range. In both cases it is 1 memory channel per 3 cores typical.

Besides choosing the right platform it is also important to make sure all memory channels are equally populated, even if it means buying smaller sticks. 4 channels with 8 GB sticks to get 32 GB total is better than 2 channels with 16 GB sticks and 2 channels left empty.

Conclusion

Memory bandwidth has a quite noticeable impact on solver performance, even for modern machines with optimal layout and apparently ample bandwidth. It is very important to consider and plan for an optimal system memory layout.

Leaving memory channels unpopulated, or unevenly balanced with different RAM sticks, is not good idea. Get memory in matched sets that are appropriate for the platform being used (at least 1 identical stick per channel).

One of the main reasons people witness disappointing performance gains when adding more cores to their simulations is not due to inefficiency of the solver code, but instead due to a combination of decreasing core frequency (turbo speed) and subdivision of out-of-core resources, especially memory bandwidth. The CFX solver demonstrated near perfect scaling at quite low mesh nodes per core, but this is frequently not witnessed when testing a single computer due to operating in a resource constrained environment.

CFX Problem Description

Pump

Case Details

- Automotive pump with rotating and stationary components (2 domains)
  - Turbulent k-e, incompressible, isothermal, multiple frames of reference
  - Advection scheme: specified blend factor 0.75
- Global mesh size: 1,305,718 nodes, 5,362,055 elements
  - 4,509,881 tetrahedra, 850,617 prisms, 1557 pyramids

Benchmark Information

- Suitable for up to ~16 cores
- Currently set to 10 iterations
- Partitioning memory requirement ~450 MB
- Partitioning time <20 sec on Intel i7-2820qm (Sandy Bridge)
- Solver memory requirement (total) ~3 GB