AMD Rome review

Martin Cuma, CHPC

In this article we look at the performance of the the AMD second generation EPYC CPU, code named Rome, released in August 2019 and compare it to the current Intel competitor, Cascade Lake Xeon, released in April 2019, along with the previous generations of the AMD and Intel CPUs.

The 2nd gen EPYC is a refresh of the older EPYC architecture introduced in 2017. While many of the CPU core specifics are similar or the same, there have been significant modifications to both the core and the whole CPU design which have fixed the deficiencies of the 1st gen EPYC and provide considerable speed and memory bandwidth improvements for technical computing. The Rome chips, similar to the previous generation Naples, consist of several multi-core chiplets, as opposed to traditional monolithic CPU designs, as shown in Figure 1. The Rome design consists of 7 nm process CPU chiplets and 14 nm process I/O die, which connects to all the chiplets and creates a more uniform core hierarchy as compared to the Naples (Figure 2). The smaller chiplet production is easier and cheaper than larger monolithic CPU.

Figure 1. Monolithic, Naples and Rome CPU layouts, from https://www.nextplatform.com/2019/08/07/amd-doubles-down-and-up-with-rome-epyc-server-chips/

The chiplets include eight CPU cores and are called Core Complex Dies (CCDs). The CCDs communicate with the I/O Die (IOD) via high speed Infinity Fabric links (see Fig 1). The IOD connects to DRAM, PCIe or other CCDs. Each CCD consists of two four-core Core Complexes (CCX), each of which has 16 MB L3 cache. There are two possible NUMA modes on this CPU, most likely enabled by a BIOS change. One considers all cores in each CPU as monolithic (single level NUMA, Figure 3a), while the other has an extra NUMA level on the CCx (dual level NUMA, Figure 3b). On a 32 core CPU (4 CCDs, 8 CCXs), the “numactl -H” output for the first case would then look like:

```
node distances:
node   0   1
  0:  10  32
  1:  32  10
```

while for the other case it would be:

```
node distances:
node   0   1   2   3   4   5   6   7
```

```
0:  10  12  12  12  32  32  32  32
1:  12  10  12  12  32  32  32  32
2:  12  12  10  12  32  32  32  32
3:  12  12  12  10  32  32  32  32
4:  32  32  32  32  10  12  12  12
5:  32  32  32  32  12  10  12  12
6:  32  32  32  32  12  12  10  12
7:  32  32  32  32  12  12  12  10
```

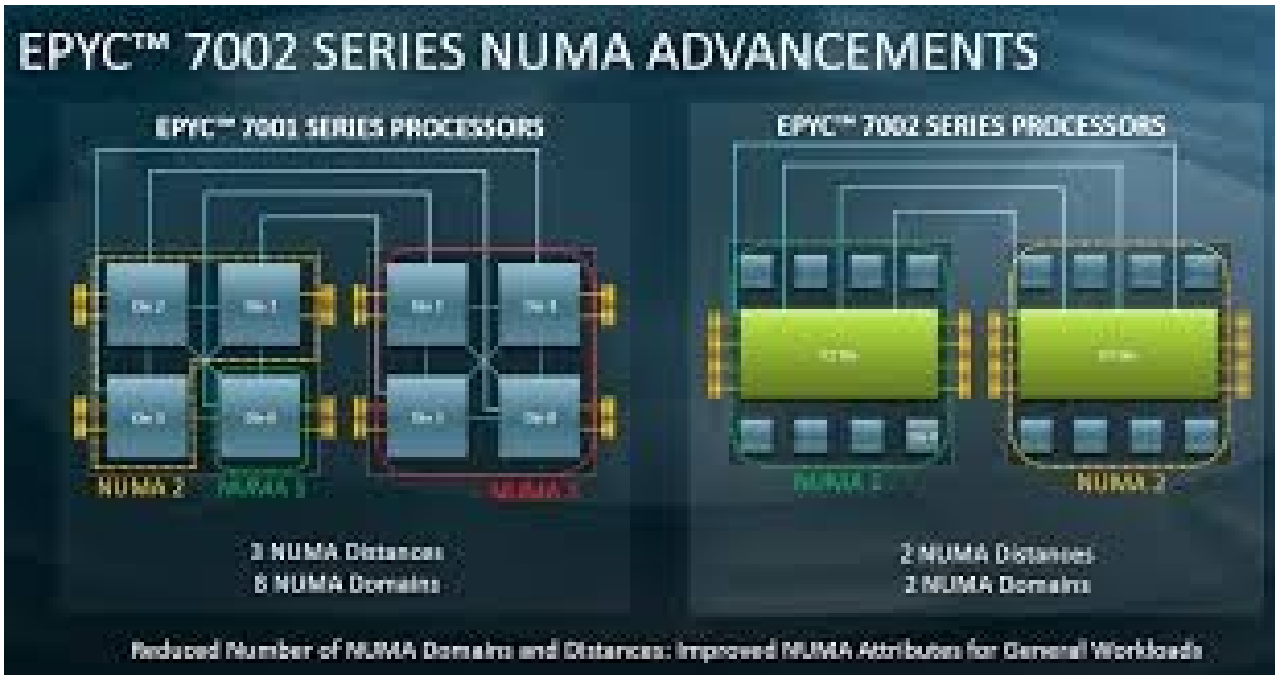

Figure 2, Naples and Rome NUMA layout, previously at https://www.anandtech.com/show/14694/amd-rome-epyc-2nd-gen/2

With respect to the CPU core architecture improvements, the most significant in for our purposes is the inclusion of two AVX2 vector units, making it capable of up to 16 double precision FLOPS per cycle, double of what was on Naples. Other architectural changes are well described at https://www.nextplatform.com/2019/08/15/a-deep-dive-into-amds-rome-epyc-architecture/.

Other features of note inculde PCI-Express generation 4 support, up to 128 lanes, eight-channel memory controller on CPU each socket, and DDR4 memory speed up to 3200 MHz.

Looking at other notable differences from the Intel Cascade Lake architecture, the AMD Rome includes 8 memory channels (compared to 6 in Cascade Lake) and faster memory (3200 MHz vs. 2933 MHz), which results in higher memory bandwidth. The Rome also integrates 128 PCI Gen 3 lanes, while the Skylake CPUs have 48 lanes, which should be beneficial for connection of peripherals like GPUs or network cards. The Intel CPU has two AVX512 vector units per core, as compared to two AVX2 vector units in the AMD CPU. However, the Intel CPU scales down the core frequency considerably when more cores with vector units are being used, while AMD is claiming not to do so, or, at least, not so aggressively. Microway has a great article describing Cascade Lake and its frequency scaling at https://www.microway.com/knowledge-center-articles/detailed-specifications-of-the-cascade-lake-sp-intel-xeon-processor-scalable-family-cpus/. Similar article describing the AMD Rome is at https://www.microway.com/knowledge-center-articles/detailed-specifications-of-the-amd-epyc-rome-

cpus/. Our testing below confirms that the AMD CPUs do not scale down the clock speed as much when running at full core utilization, though some frequency scaling is apparent for the higher core count CPU (7702 with 64 cores).

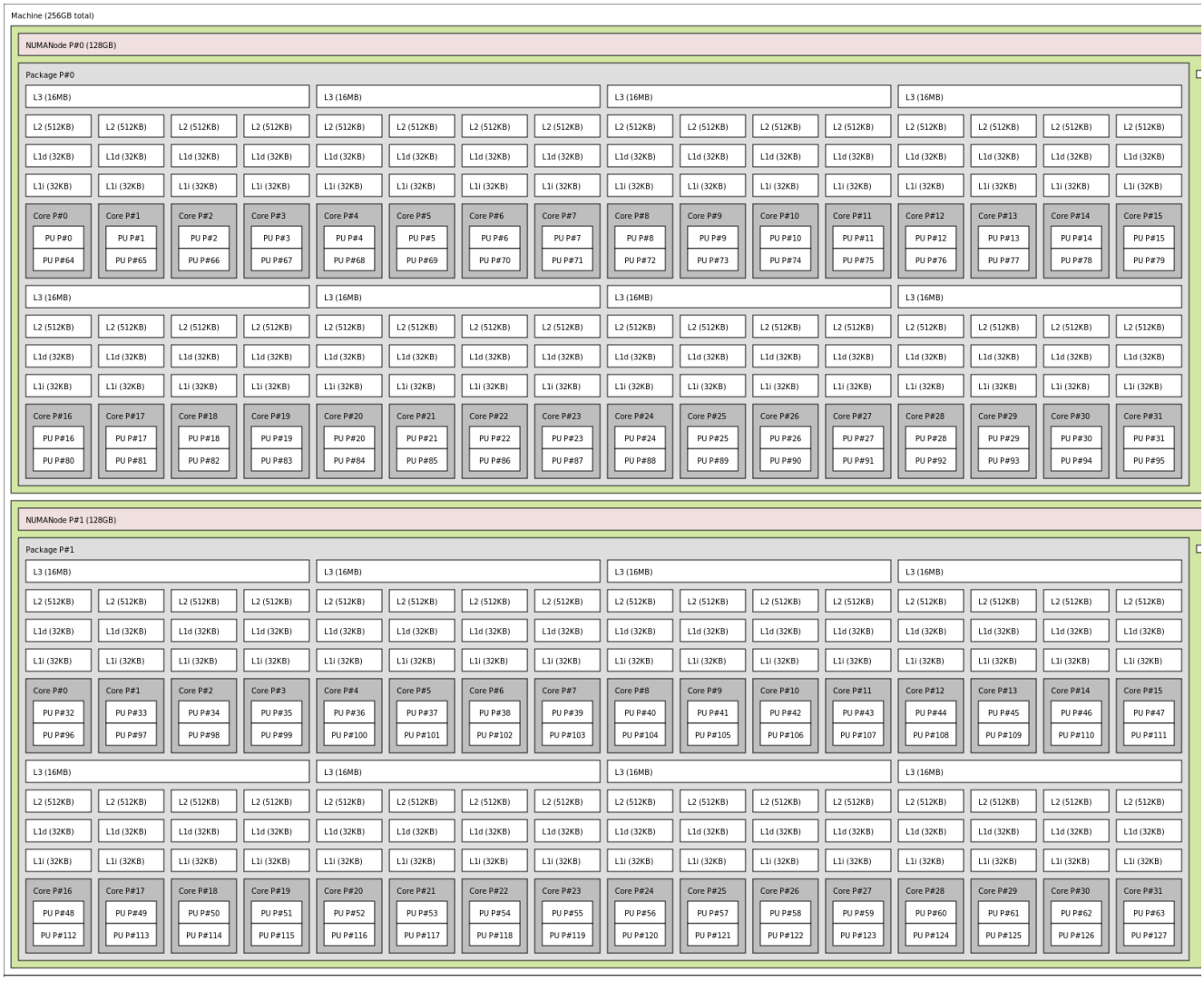

Figure 3a. NUMA layout of a 2x32 core (2x EPYC 7452) Rome server in the single NUMA level mode. The two CPUs are on the top of each other. Each L3 cache is shared by 4 cores marking the CCX.

Figure 3b. NUMA layout of a 2x32 core (2x EPYC 7452) Rome server in the dual NUMA level mode. Package is the CPU, NUMANode are the CCDs. Each L3 cache is shared by 4 cores marking the CCX.

We got access to a Dell test cluster that included several different Rome CPUs on the PowerEdge C6525 platform. We focused on the 7452 32 core, 2.3 GHz CPUs in a dual socket configuration, and on the 7702 64 core, 2.0 GHz CPU, that was also in a dual socket configuration, but, only looking at a single socket performance. The MSRP of the 7452 is $2025 and of 7702 is $6450, however, a single socket node is supposed to cost considerably less than a dual socket node. Each AMD node had 256 GB of RAM.
Intel Cascade Lake nodes are current standard nodes at CHPC, which are dual socket 20 core Xeon Gold 6230 running at 2.1 GHz, at a list price of $1894.

In the comparisons, we also include older timings from the AMD Rome and Skylake Xeon Gold 6130 CPUs.

External benchmarks

Of the benchmark resources on the Internet, good ones include https://www.anandtech.com/show/14694/amd-rome-epyc-2nd-gen, and https://www.phoronix.com/scan.php?page=article&item=amd-epyc-7642, but each only has a handful of HPC like applications.
Dell has published their initial benchmarks at https://www.dell.com/support/article/us/en/04/sln319015/amd-rome-is-it-for-real-architecture-and-

initial-hpc-performance?lang=en. This document gives the best performance estimates for our purposes, and we have worked with Dell engineers during our tests to match their performance in the tests that we both ran.

Raw and synthetic performance benchmarks

STREAM benchmark

The STREAM benchmark tests the bandwidth from CPU to the main memory by performing four different operations on large sequential data arrays. We have compiled STREAM using the Intel 2019.5 on the Rome and Cascade Lake and use older data for Skylake and Naples that was built with Intel 2017.4 and gcc 6.3.0, respectively. STREAM is thread parallelized using OpenMP and we look at the memory throughput from one thread to the number of threads equal to the number of the physical cores. As all the machines have NUMA CPUs, we also look at the effect of the thread locality to the CPU core.

The chiplet design of the AMD CPUs allows for several different thread placements, as compared to two on the Intel monolithic CPUS: *sequential* (called *compact* by Intel OpenMP) - where first all the cores on CPU 0 get filled, followed by CPU 1, and *spread* (called *scatter* by Intel OpenMP), where the threads get packed on the two CPU sockets in a round robin fashion.

On the Rome CPU, we have looked at five different CPU placements listed below, with the core placements corresponding to dual socket 7702 CPUs:
sequential, core placement 0,1,2,3,…, Intel OpenMP options *KMP_TOPOLOGY_METHOD=hwloc KMP_AFFINITY=verbose,granularity=fine,compact*
L3sequential, core placement 0,4,8,12,… - distributing threads across the 4-core shared L3 cache CCXs , Intel OpenMP options *KMP_TOPOLOGY_METHOD=hwloc KMP_AFFINITY=verbose,granularity=fine,compact,1*
chipletsequential, core placement 0,16,32,48,… - distributing threads across the CCD chiplets, Intel OpenMP options *KMP_TOPOLOGY_METHOD=hwloc KMP_AFFINITY=verbose,granularity=fine,compact,2*
socketspread, core placement 0,64,1,65,… - distributing threads across the sockets, but sequentially on each socket, Intel OpenMP options *KMP_TOPOLOGY_METHOD=cpuinfo KMP_AFFINITY=verbose,granularity=fine,scatter*
socksequentchipspread – core placement 0,64,16,90,… - distributing threads across sockets and the CCD chiplets – Intel OpenMP options *KMP_TOPOLOGY_METHOD=hwloc KMP_AFFINITY=verbose,granularity=fine,compact,3*

The Intel OpenMP compiler environment variables that control the thread placement are also listed above. More details on these options is at https://software.intel.com/en-us/cpp-compiler-developer-guide-and-reference-thread-affinity-interface-linux-and-windows.

In Figure 4 we look at these five different thread placements, along with using no thread placement at all (= allowing the threads migrate across all the cores on the dual socket machine – the OpenMP runtime default), for the Copy benchmark, which is one of the four that the STREAM contains. They all show similar trends. From this figure, we can see that the thread distribution over the CCXes and chiplets yields the best memory bandwidth on the undersubscribed system. On the opposite side, no thread placement, and sequential placement does not achieve the top bandwidth on the undersubscribed

system. Interestingly, also the socket spread (round robin) placement is not optimal – because it packs the threads tightly on the chiplets rather than distributing them over the chiplets and utilizing each chiplets' memory controller. Dell is reporting the same observation for the Triad benchmark, achieving about 3% better bandwidth that we have observed.

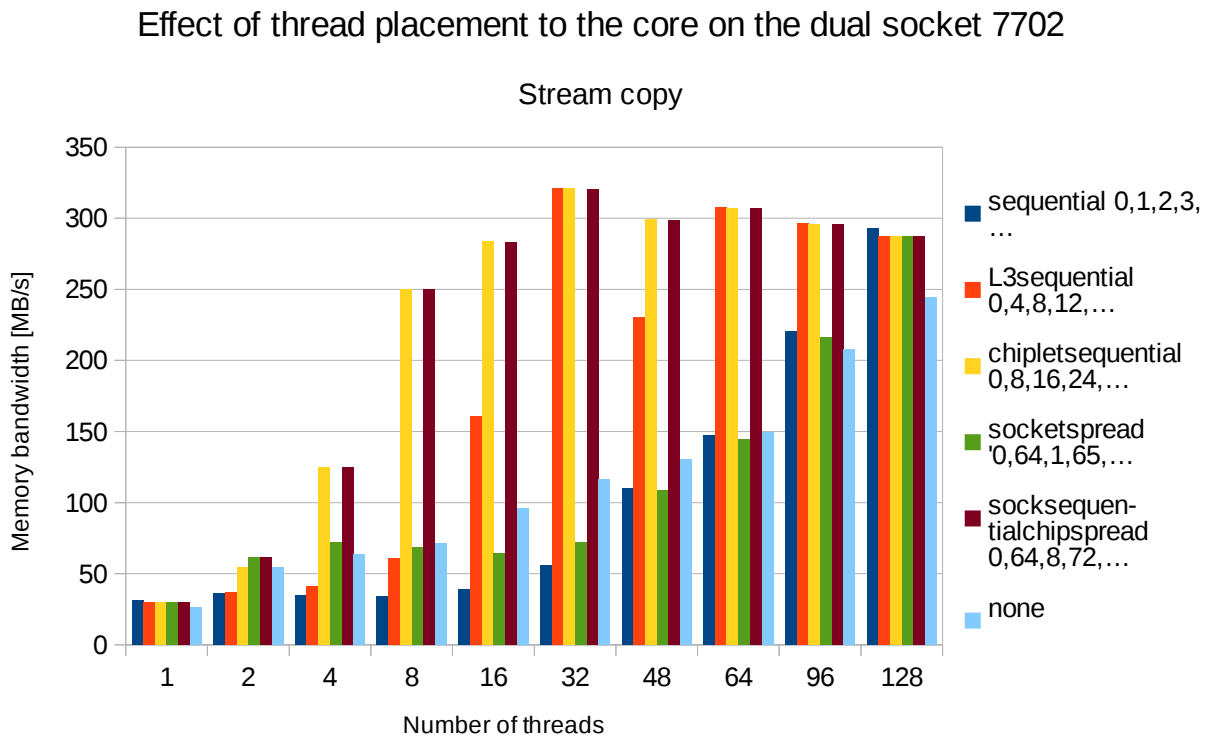

Figure 4. Stream Copy on the dual socket Rome 7702 on different thread to core mapping

Next, in Figure 5, we compare the STREAM Copy memory bandwidth between the different AMD and Intel CPUs. There are a few things to note. First, both older and new AMD CPUs have higher bandwidth than any of the Intel offerings. The Cascade Lake only has a nominal bandwidth increase over the Skylake at the full subscription (40 threads). And, the Rome 7702's undersubscribed peak bandwidth is higher than that of the 7452, presumably due to more chiplet spread over the memory controllers.

Rome's peak bandwidth is 204.8 GB/s per socket, that is 409.6 GB/s per node, though we are only achieving about 330 GB/s on the 7702 and 290 GB/s on the 7452. Supposedly the NUMA clustering has some effect on this but there seem to be a dependence on the CPU as well.

The Cascade Lake maximum of ~190 GB/s is also lower than the the per socket 128 GB/s that is 256 GB/node, and also lower than a few other published numbers that went up to 220 GB/s (https://www.dell.com/support/article/us/en/04/sln316864/bios-characterization-for-hpc-with-intel-cascade-lake-processors). We have investigated this further and concluded that this difference is due to a drop in the STREAM bandwidth with increasing array size. The Dell published benchmark used about 5 GB of RAM, while ours used about 96 GB. In Figure 6. we show the dependence of the STREAM on the problem array size on four dual socket nodes, each with a different processor. Notice that all the shown processors, except for the Cascade Lake, have a flat bandwidth curve with increasing array size. High bandwidth values below ca. 100 MB are due to the CPU caching.

Lastly, we simulate the single socket 7702 bandwidth based on the thread placements in the different pinnings we did and observe half the bandwidth of the 2 socket node, as expected, roughly equivalent

to the Skylake bandwidth.

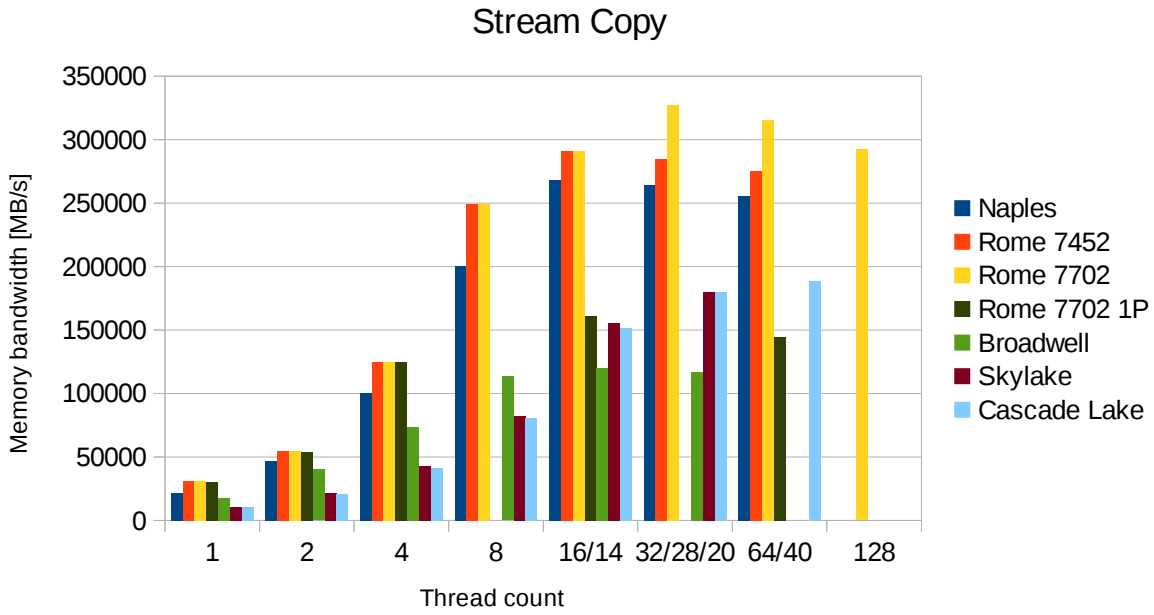

Figure 5. STREAM Copy maximum bandwidth per thread at optimal thread to core distribution

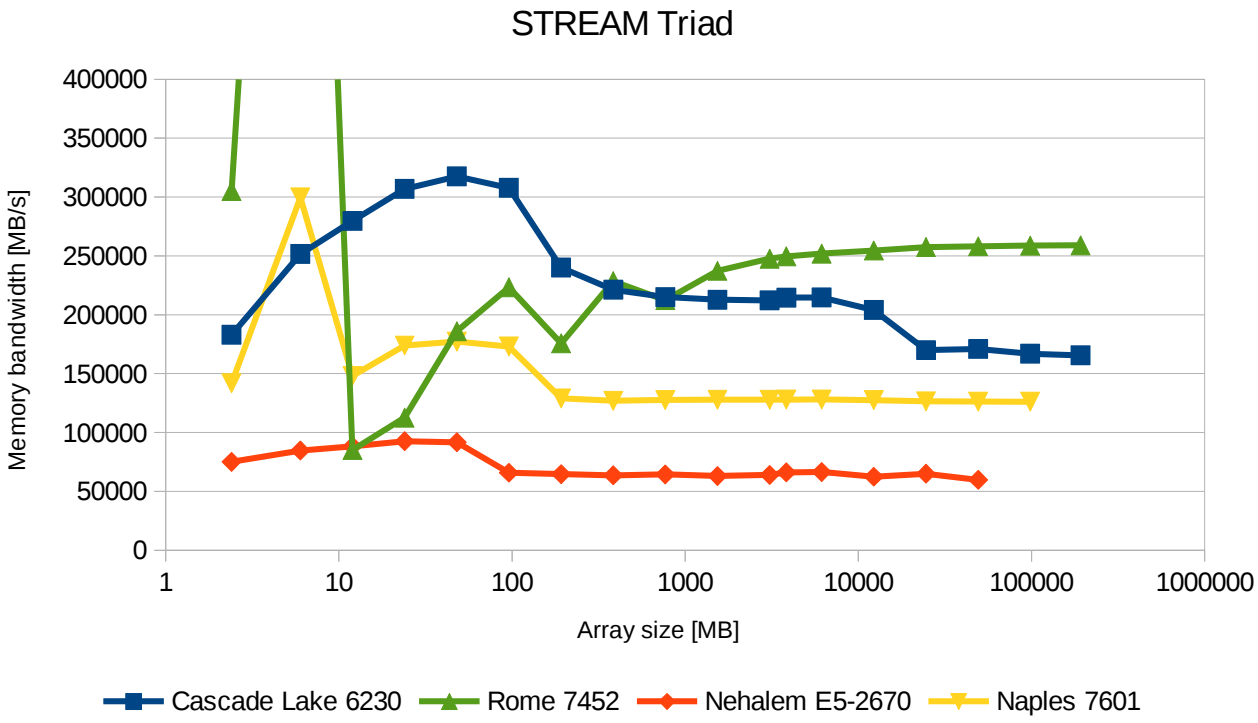

Figure 6. Dependence of STREAM Triad on the array size.

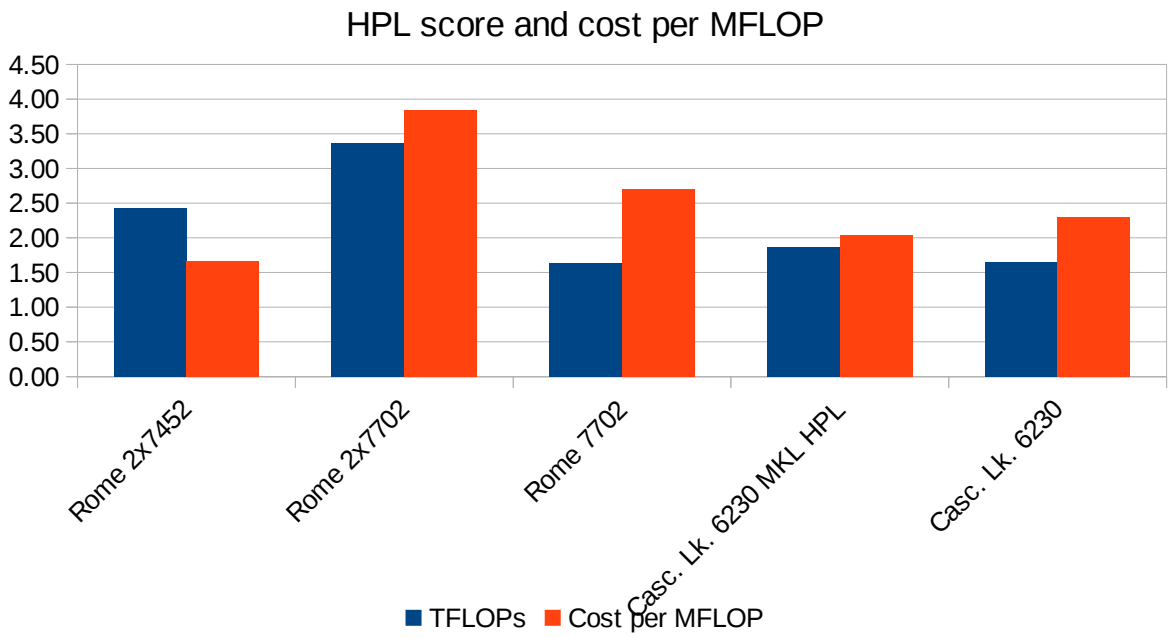

Figure 7. HPL score and cost pet MFLOP

High Performance Linpack (HPL) benchmarks

HPL is a part of the HPCC detailed below, but, we also used it separately since we have observed lower than expected HPL value for the Cascade Lake in the HPCC benchmark. After communication with Dell about their HPL score of about 2 TFLOPs, we have nearly replicated this value by running the Intel optimized HPL binary that ships with the Intel MKL library, obtaining **1.862** TFLOPs running 4 MPI tasks, 10 threads each on the 6230's 40 cores. This is about **73%** of the **2.56** TFLOPs theoretical peak. On the AMDs, the 7452s really shine getting **2.43** TFLOPs running 64 tasks per node, a **103%** of the **2.355** TFLOPs theoretical. Two 7702s get **3.355** TFLOPs running 32 tasks, 4 threads each, a **82%** of the **4.096** TFLOPs theoretical. Based on these numbers it seems that the clock is getting throttled more due to the heat on the higher core count 7702. Similarly, single 7702 achieved **1.633** TFLOPs, **80%** of its peak

Based on these observations, we conclude that the Intel CPU scales its frequencies due to the heat issues the most, followed by the high core count AMD 7702. The AMD 7452 achieves effective cooling even when all the cores are being used. This supports the AMD unwritten claim that the CPU clock speed is not throttled significantly when using vectorization at high core count. The explanation for the 7702 slower performance may be twofold – more CPU clock speed throttling as the 64 core CPU cooling must be more difficult than the 32 core one, and also limited memory access bandwidth since the 64 7702 cores have the same memory bandwidth as the 32 7452 cores. The Cascade Lake continues the trend set by previous Intel processor generation, where the CPU clock speed gets throttled significantly with the use of vectorization at high core count.

In Figure 7 we summarize the HPL scores and normalize them to cost per MFLOP, based on the list price of the CPU. Note that the price does not include other node infrastructure cost, which would possibly favor more the single socket AMD 7702P. So, this chart needs to be taken with a grain of salt, or adjusted for the real node costs. In either case, we can see that the AMD 2x7452 comes out as the best, followed by the Intel 2x6230, followed by the AMD 7702P. The AMD 7702 dual socket processor is much more expensive, and better comparable to the high end Intel SKUs which are overpriced with

respect to the performance as well.

High Performance Computing Challenge (HPCC) benchmark

HPCC benchmark is a synthetic benchmark suite geared at assessing HPC performance from different angles. It consists of seven main benchmarks that stress various computer subsystems, such as raw performance, memory access and communication. For detailed description of the benchmark see http://icl.cs.utk.edu/hpcc/. We use version 1.5.0.

On the latest AMD and Intel CPUs, we have built HPCC with Intel 2019.5 compiler and the corresponding MKL and Intel MPI. On the Intel platforms, we used flags -O3 -ansi-alias -ip -axCORE-AVX512,CORE-AVX2,AVX -restrict and on the AMD Rome flags -O3 -ansi-alias -ip -march=core-avx2 -restrict. For the older Skylake and Broadwell, we have built HPCC 1.5.0 with Intel 2017.4 compilers and the corresponding Intel MKL and MPI using the same compiler flags as on the Cascade Lake. On the Epyc, we used gcc 6.3.0 with BLIS and -O3 -fomit-frame-pointer -funroll-loops -march=native.
Also of note is that we had to use undocumented MKL environment variable MKL_DEBUG_CPU_TYPE=5, which turns on specific vectorization instructions in the MKL, and results in about 20% speedup in HPL and other codes that use MKL BLAS.

Year	2019	2019	2019	2017	2017	2016	2014	2012	2010
CPU generation	Rome 2x32	Rome 64	Casc. Lk.	Naples 64	Skylake	Broadwell	Haswell	SandyBr.	Westmere
Core count	2x32	1x64	2x20	2x32	2x16	2x14	2x12	2x8	2x6
Frequency_GHz	2.3	2	2.1	2.0	2.1	2.4	2.5	2.2	2.8
HPL_Tflops	**2.43**	1.63	1.66	1.03	1.64	0.85	0.73	0.27	0.12
StarDGEMM_Gflops	40.91	27.80	48.36	17.57	**54.04**	31.98	31.83	17.08	10.46
SingleDGEMM_Gflops	51.11	49.20	**60.11**	18.48	56.09	41.41	41.72	20.30	10.71
PTRANS_GBs	14.79	11.56	**14.92**	11.72	13.94	10.84	7.39	4.62	3.05
MPIRandomAccess_GUPs	**0.34**	0.23	0.15	0.092	0.0026	0.0037	0.0266	0.0171	0.0427
StarRandomAccess_GUPs	0.01	0.00	**0.02**	0.028	0.0397	0.0304	0.0256	0.0292	0.0196
SingleRandomAccess_GUPs	0.12	0.12	0.04	0.093	0.0787	0.0825	0.0778	0.0611	0.0366
StarSTREAM_Triad	2.84	1.62	3.98	2.90	**4.55**	3.26	2.55	3.42	2.48
SingleSTREAM_Triad	**20.62**	20.27	14.80	19.65	12.57	10.55	12.93	12.50	10.25
StarFFT_Gflops	1.39	0.95	**1.79**	0.96	2.06	1.67	1.53	1.51	1.22
SingleFFT_Gflops	1.69	1.59	2.49	1.23	**2.75**	2.31	2.38	2.03	1.95
MPIFFT_Gflops	**37.79**	24.97	28.79	20.38	29.88	11.93	8.53	7.90	4.64

Table 1. HPCC results, the higher the value the better. The best values shown in bold.

In Table 1 we show the result of select HPCC metrics for select fully loaded nodes Intel Xeon CPUs since 2010 and the two generations of the AMD EPYC CPUs. The dual socket Rome 7452 node has taken the lead in the HPL, at a fairly impressive ratio over the Cascade Lake node. A single socket 7702 node has roughly the same performance as the Cascade Lake. Note that the Cascade Lake performance is using our binary, not the optimized one from Intel, running one task per CPU core – which is why we are getting roughly 10% lower score.

HPL_Tflops	High Performance Linpack benchmark - the one that's used for Top500 - measures the floating point rate of execution for solving a linear system of equations.

StarDGEMM_Gflops	Parallel DGEMM - measures the floating point rate of execution of double precision real matrix-matrix multiplication.
SingleDGEMM_Gflops	Serial DGEMM - on single processor
PTRANS_GBs	Parallel Matrix Transpose - exercises the communications where pairs of processors communicate with each other simultaneously. It is a useful test of the total communications capacity of the network.
MPIRandomAccess_GUPs	MPI Parallel Random Access
StarRandomAccess_GUPs	UPC Parallel Random Access - measures the rate of integer random updates of memory (GUPS).
SingleRandomAccess_GUPs	Serial Random Access
StarSTREAM_Triad	Parallel STREAM - a simple synthetic benchmark program that measures sustainable memory bandwidth (in GB/s) and the corresponding computation rate for simple vector kernel.
SingleSTREAM_Triad	Serial STREAM
StarFFT_Gflops	Parallel FFT - measures the floating point rate of execution of double precision complex one-dimensional Discrete Fourier Transform (DFT).
SingleFFT_Gflops	Serial FFT
MPIFFT_Gflops	MPI FFT

Table 2. HPCC benchmarks explanations.

To visualize the improvement in floating point performance, in Figure 8 we show the High Performance Linpack (HPL) performance of the current and previous AMD and Intel CPU generations, which is exemplifies the change in the floating point (FP) vectorization units. The 2010 Westmere CPU had SSE4.2 vectorization set capable of doing 2 double precision operations (DPO) per cycle. This has doubled to 4 DPO/cycle in 2012 SandyBridge with the AVX instruction set. The 2014 Haswell's AVX2 added Fused Multiply Add (FMA) instruction, which, along with the increase in core count and clock speed as compared to our benchmarked SandyBridge more than doubled the floating point output. Broadwell CPU was a process shrink of Haswell so the extra performance was added mainly by the increased core count. Going to Skylake, we are seeing another doubling of FP performance with the 8 DP long AVX512 instruction set. The Cascade Lake performance improvement over Skylake is minimal, likely due to the similar memory bandwidth and CPU clock speed throttling. Again this is building the HPL from the source using Skylake/Cascade Lake compiler optimization. Dell Labs published results on the same Cascade Lake processors reaching up to 2 TFLOPs, https://www.dell.com/support/article/us/en/04/sln316864/bios-characterization-for-hpc-with-intel-cascade-lake-processors?lang=en, we achieved 7.5% less with Intel optimized HPL binary running in multi-threaded mode, but still 10% more than what we are showing in Figure 8. Based on discussion with Dell the 7.5% lower perfomance with the Intel optimized HPL binary is within a range of roughly 15% performance difference they have noticed during their tests.
Also, the Skylake performs better than the Cascade Lake in a few tests. My guess here would be potential effects of the power governors in the system or the BIOS – in the Cascade Lake we saw about 8% HPL difference between them and the HPCC runs used the less aggressive ones.

On the AMD side, the first generation Epyc had only one AVX2 unit capable of 8 FLOPs per cycle, and its 1.03 GFLOPs is close to the theoretical peak. Both Rome chips pulled ahead significantly, with the

2x7452 more than doubling the HPL throughput as compared to the first CPU generation.

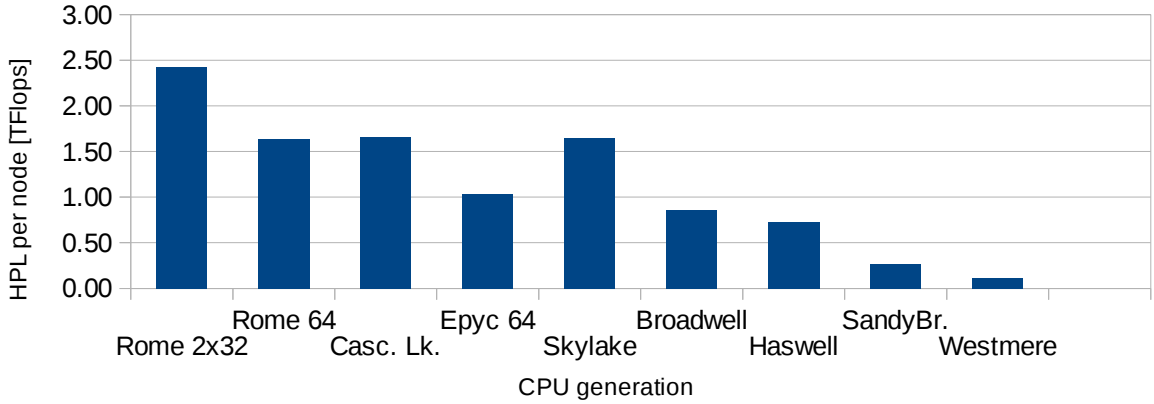

Figure 8. Top HPL performance for the Epyc and select Intel CPU generations. Higher value is better.

The other HPCC benchmarks paint further interesting points on the AMD vs Intel performance. Single core dense linear algebra is a strong point of the Intel thanks to its wider vector unit – as evident from the DGEMM values. The same goes, somewhat surprisingly, to the FFT. Memory bandwidth is the strong point of the AMD, seen from the STREAM numbers. The MPI benchmarks probably benefit the most from the higher AMD core count, so the AMD is better. The single socket 7702 is not a winner anywhere but its performance is within the range.

NAS Parallel Benchmarks

NAS Parallel Benchmarks are a set of programs derived from computational fluid dynamics (CFD) applications. Some basic information about the benchmarks is here: https://en.wikipedia.org/wiki/NAS_Parallel_Benchmarks. Each of these benchmarks can be run with different problem sizes. Class A is a small problem, Class B is medium size, Class C is a large problem, and Class D is a very large problem (needing about 12 GB of RAM). There are also even larger classes E and F. We have ran Classes A-D and present results for Class C. We have compiled the codes with Intel 2019 and 2017 compilers, using "-O3 -ipo -axCORE-AVX512 -qopenmp" option on the Cascade Lake and Skylake, respectively, and "-O3 -ipo -axCORE-AVX2 -qopenmp" option on the Haswell. On the AMD for the older Naples chip, we used gcc 6.3.0 with “-O3 -fopenmp -mcmodel=medium -mavx2 -mfma4” flags. On the Rome, we used Intel compiler 2019.5 with options “-O3 -ipo -march=core-avx2 -qopenmp”.

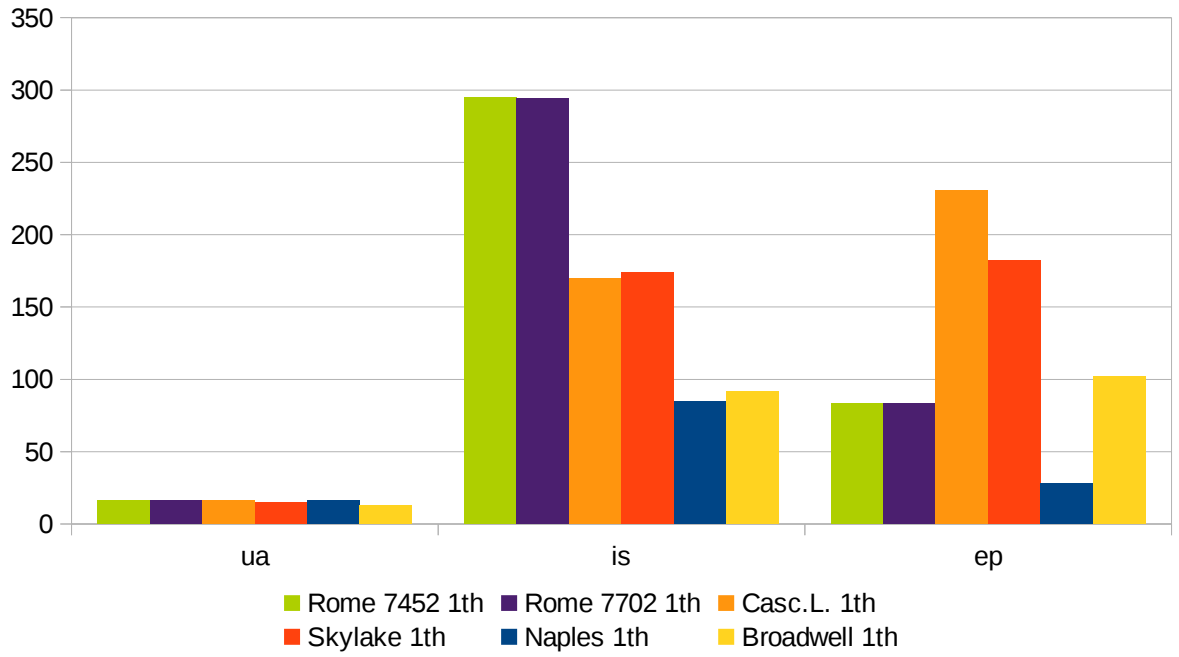

Figure 9a. Single core (one thread) NAS UA, IS and EP benchmarks for size C

All the NAS benchmark plots compare the performance in Mops/sec or Mops/sec/thread. As we are looking at comparing maximum performance on the whole multi-core machine, and also evaluating the SMP capabilities, below we look at the Mops/sec. The higher is the Mops/sec count, the better. We present the benchmarks in four graphs broken by the single thread and whole node performance, and by similar values of Mops/sec, for better comparison.

The NAS parallel benchmarks cover a wide variety of algorithms and as such their performance varies both with the CPU generations and across the different CPU manufacturers. Benchmarks like the UA (Unstructured Adaptive) or MG (MultiGrid) do not vectorize as much and therefore their single core performance stays similar across the CPU generations. Other benchmarks, such as the FT (Fast Fourier Transform), EP (Embarrassingly Parallel random numbers), or even IS (Integer Sort) improve significantly with newer CPU generations, benefiting either from increased memory bandwidth, or from vectorization.

On the single core basis – the two AMD Rome CPUs are more or less comparable. They also beat the Cascade Lake CPU on all but 3 benchmarks (EP, CG, BT)

Moving to the whole node graphs, in most cases we can see the effects of the increasing core count. Comparing the AMD Rome to Intel Cascade Lake, the AMD is a winner in eight of the benchmarks whileIntel wins two (BT, EP). The single Rome 7702 CPU is better than the dual CPU Cascade Lake node in 6 out of the 9 benchmarks. The 7702 takes a large performance hit in memory intense benchmarks like the FT or IS, while for the more computational like LU or SP the difference from the dual socket 7452 system is smaller.

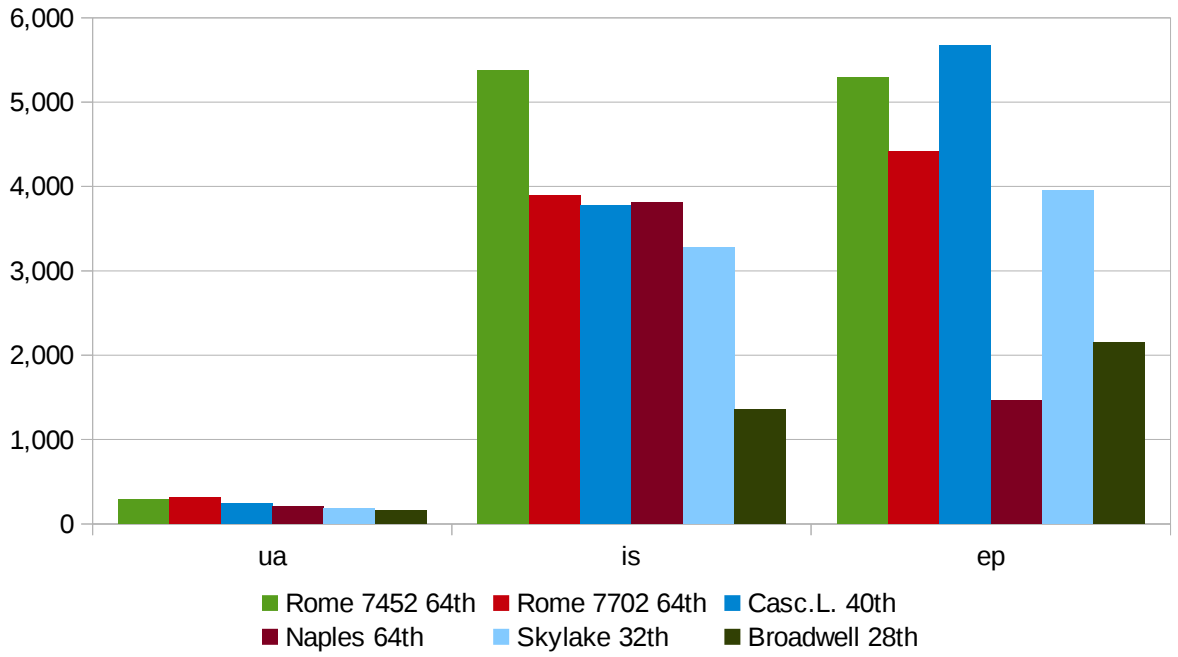

Figure 9b. Whole node NAS UA, IS and EP benchmarks for size C

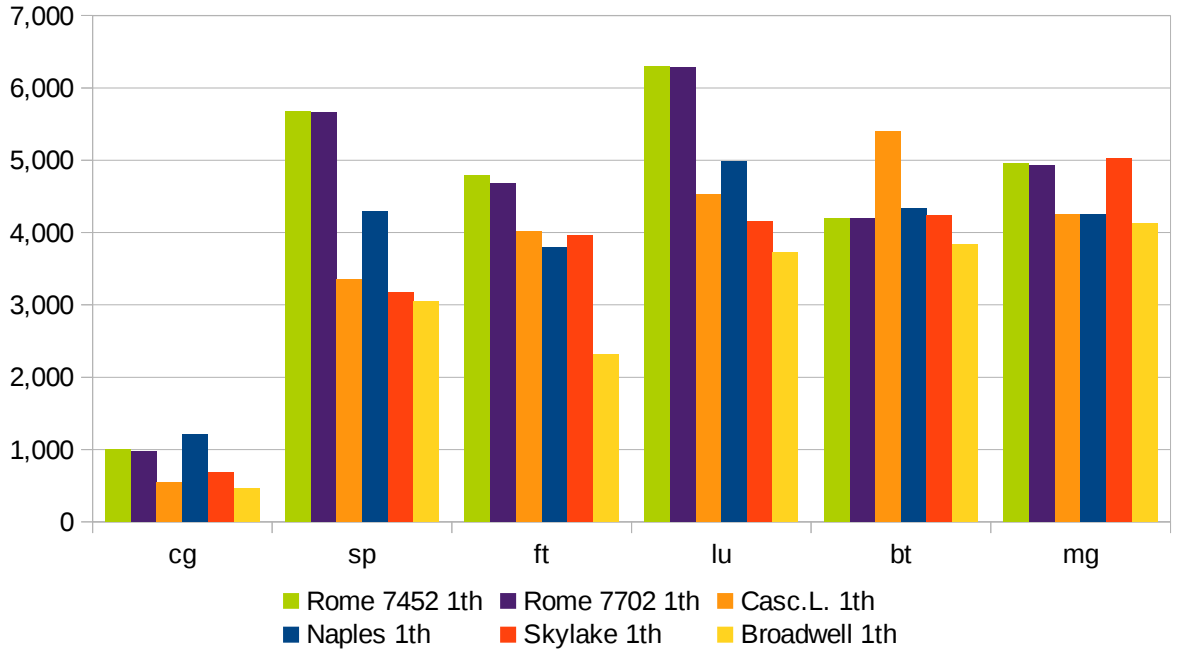

Figure 10a. Single core (one thread) NAS CG, FT, SP, LU, BT and MG benchmarks for size C

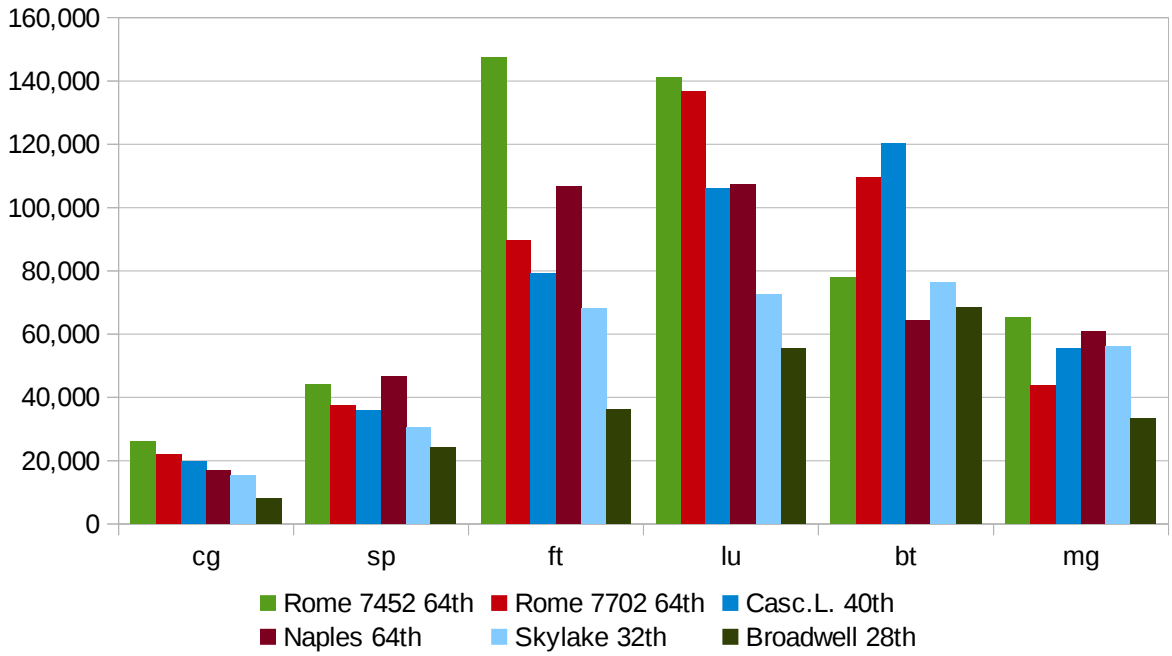

Figure 10b. Whole node NAS CG, FT, SP, LU, BT and MG benchmarks for size C

Real applications benchmarks

LAMMPS

LAMMPS is a popular molecular dynamics simulation program developed at Sandia National Laboratory. It is a good representative for multi-body like simulations that use internally coded computational kernels, not relying so much on vendor accelerated libraries.

We have built the 31Mar17 version using Intel 2019.5 or 2017 compilers, MPI and MKL (using MKL's FFTW wrappers) and with optimization flags "-axCORE-AVX512,CORE-AVX2,AVX,SSE4.2 -O3 -prec-div -fp-model precise". On the AMD Naples, we used the fat Intel built binary as used on the Skylake. On the AMD Rome, we used Intel 2019.5 with flags “-march=core-avx2 -ip -prec-div -fp-model precise”. The rest of the flags were taken from the USER-INTEL package makefile.

We have run three LAMMPS benchmarks from http://lammps.sandia.gov/bench.html:
LJ = atomic fluid, Lennard-Jones potential with 2.5 sigma cutoff (55 neighbors per atom), NVE integration
Chain = bead-spring polymer melt of 100-mer chains, FENE bonds and LJ pairwise interactions with a 2^(1/6) sigma cutoff (5 neighbors per atom), NVE integration
EAM = metallic solid, Cu EAM potential with 4.95 Angstrom cutoff (45 neighbors per atom), NVE integration
Each problem was scaled 2x in each dimension resulting in 256,000 atoms and was run for 1,000 time steps.

In Table 2 we show the benchmark results for the last two generations of the AMD and Intel CPUs, with the bold number being the whole node runtime in seconds – that is what an user would typically

run. All runs were run with MPI tasks only, single OpenMP thread. LAMMPS benefits from high core count and as such the Rome 7452 gives almost 2x advantage over the Intel Cascade Lake node. Comparing the dual socket 7452 with single socket 7702P, the performance hit is 12-15%.

One thing to keep in mind is that we built the LAMMPS fairly standardly without additional packages. We have looked at the USER-OPENMP and KOKKOS packages, which provide thread based parallelism in LAMMPS on the top of the default MPI parallelism, but, we did not get better performance as compared to the pure MPI runs. This is reasonable as molecular dynamics codes generally are not as communication heavy, so the thread based overhead sticks out more.

Procs	NP	RM1 7452	RM2 7702	SKL	CL	RM1/CL	RM2/CL
1	95.24	80.50	78.71	74.26	71.21	1.13	1.11
2	48.49	36.55	42.08	35.04	33.23	1.10	1.27
4	23.07	17.91	23.93	17.74	17.38	1.03	1.38
8	10.91	8.83	10.63	9.37	9.02	0.98	1.18
16	5.45	4.64	5.19	4.80	4.65	1.00	1.11
32/20	3.32	2.38	2.45	**2.89**	3.92	0.61	0.62
64/40	**1.82**	**1.30**	**1.47**		**2.32**	**0.56**	**0.64**

Table 2a. LAMMPS chain benchmark performance (in seconds, lower is better) and speedup ratio of the Rome 7452 node with respect to the Cascade Lake node and the 7702P single socket node.

	NP	RM1 7452	RM2 7702	SKL	CL	RM1/CL	RM2/CL
1	437.37	357.19	354.38	305.00	294.20	1.21	1.20
2	223.54	182.78	181.91	155.34	149.19	1.23	1.22
4	113.86	92.34	95.47	80.93	80.49	1.15	1.19
8	58.13	46.90	49.37	43.73	42.79	1.10	1.15
16	29.35	24.06	25.56	22.75	23.08	1.04	1.11
32/20	17.33	12.22	12.57	**13.87**	19.81	0.62	0.63
64/40	**9.00**	**6.32**	**7.25**		**11.44**	**0.55**	**0.63**

Table 2b. LAMMPS eam benchmark performance (in seconds, lower is better) and speedup ratio of the Rome 7452 node with respect to the Cascade Lake node and the 7702P single socket node.

	NP	RM1 7452	RM2 7702	SKL	CL	RM1/CL	RM2/CL
1	166.38	135.34	134.81	117.24	115.33	1.17	1.17
2	85.96	69.12	69.91	59.08	58.78	1.18	1.19
4	44.05	34.47	37.43	31.01	31.54	1.09	1.19
8	22.22	17.59	19.11	16.76	16.75	1.05	1.14
16	11.03	8.93	9.78	8.66	8.77	1.02	1.12
32/20	6.48	4.57	4.84	**5.22**	7.31	0.62	0.66
64/40	**3.36**	**2.36**	**2.82**		**4.39**	**0.54**	**0.64**

Table 2c. LAMMPS lj benchmark performance (in seconds, lower is better) and speedup ratio of the Rome 7452 node with respect to the Cascade Lake node and the 7702P single socket node.

VASP

VASP is a plane wave electronic structure program that is widely used in solid state physics and materials science. As with many quantum simulation codes, VASP uses dense linear algebra heavily

through the BLAS – LAPACK – ScaLAPACK libraries. It thus provides a convenient benchmarking tool for vendor supplied accelerated libraries like the MKL.

We have compiled VASP 5.4.4 with Intel 2017 or 2019 (for the Cascade Lake) compilers, MKL and MPI, and "-O2 -axCORE-AVX512,CORE-AVX2,AVX,SSE4.2" compiler flags on the Intel machines and with “-O2 -march=core-avx2” on the AMD machines. In all cases we started with the VASP supplied makefile.include.linux_intel make flags, and used MKL for all the external libraries, including FFTW.

We present two benchmarks of semiconductor based systems, Si and SiO, the SiO being several times larger. The smallest system is slowly becoming less relevant as both the hardware and the software improve, so, in our explanations we focus on the larger problem. As with the HPCC, we include results we obtained on previous generation of processors in Table 3, though, beware that the older CPUs were run with older VASP version which was potentially less optimized. The results are runtime in seconds, the smaller the number the better.

(Si 12 layer, 24 at., 16 kpts, 60 bnds)

CPUs	1	2	4	8/12	16	24	28/32	40/64
Westmere-EP 2.8 12c	233.49	123.05	68.79	47.13				
Sandybridge 2.2 16c	195.83	102.24	56.15	36.17	36.71			
Haswell 2.5 20c	118.02	56.70	34.58	22.13	15.74	27.06		
Broadwell 2.4 28c	108.46	55.31	30.06	19.25	12.84	13.52	13.85	
Skylake 2.1 32c	80.41	41.60	22.78	15.50	11.33	11.08	9.30	
Cascade Lake 2.1 40c	46.87	23.64	14.09	10.10	7.33		7.90	8.54
Naples 2.0 64c	118.58	61.88	32.35	18.62	11.03	12.19	8.99	8.22
Rome 7452 2.3 64c	60.89	32.53	17.25	10.01	6.30		4.94	5.08
Rome 7702 2.0 64c	61.22	36.42	25.00	19.56	16.56		9.81	6.69
Rome 7452 vs. CasL	**1.30**	**1.38**	**1.22**	**0.99**	**0.86**		**0.63**	**0.59**
Rome 7702 vs. CasL	**1.47**	**1.49**	**1.42**	**1.20**	**0.97**		**0.97**	**0.88**

(Si192+O, 4 kpts, 484 bnds)

CPUs	1	2	4	8/12	16	24	28/32	40/64
Westmere-EP 2.8 12c	999.36	514.66	330.20	175.22				
Sandybridge 2.2 16c	771.53	396.33	215.07	128.79	120.68			
Haswell 2.5 20c	424.72	187.93	116.83	76.69	57.79	41.52		
Broadwell 2.4 28c	395.01	163.62	91.65	55.61	41.63	34.36	35.09	
Skylake 2.1 32c	278.25	144.49	75.63	45.29	32.17	26.25	27.92	
Cascade Lake 2.1 40c	266.72	148.63	78.75	44.37	30.85		27.26	23.76
Naples 2.0 64c	549.13	269.80	138.39	75.16	42.80	38.54	35.23	33.55
Rome 7452 2.3 64c	277.34	147.55	73.21	36.78	26.34		22.08	20.94
Rome 7702 2.0 64c	275.89	176.02	128.42	104.09	58.39		23.80	21.11
Rome 7452 vs. CasL	**1.04**	**0.99**	**0.93**	**0.83**	**0.85**		**0.81**	**0.88**
Rome 7702 vs. CasL	**1.03**	**1.18**	**1.63**	**2.35**	**1.89**		**0.87**	**0.89**

Table 3. VASP performance in seconds (lower is better)

With respect to the per core performance, the larger system benchmark runtime is comparable between the Rome and Cascade Lake CPUs, while for the smaller one the Cascade Lake runs faster. This suggests that more computation in the larger system helps the AMD processor – as VASP is heavy in dense linear algebra, provided by the MKL library. Looking at the whole system performance, both Rome CPUs get 11-12% advantage over the Cascade Lake – though note that even the larger system is

getting small for the 64 CPUs to scale. We should note that we have used the undocumented MKL_DEBUG_CPU_TYPE=5 environment variable to get better MKL performance – without it the performance was 10-20% worse.
We also tried multi-threaded mode on the dual socket 7452 but the performance using more than one thread per MPI task, filling up the node with tasks/threads, was slightly worse than using 64 single threaded MPI tasks.

Conclusions

The AMD Rome CPU was promising to deliver a shake to the CPU industry and it did. Comparably priced two Rome 7452 CPUs perform better than their Intel 6230 counterpart in many benchmarks, sometimes, like in the LAMMPS example by almost 2x.

The situation is not as clear-cut with the single socket 64 core Rome 7702P, which we have tried to simulate by running on a single socket of a dual socket 7702 node. Some benchmarks like the HPL get hit by the lower memory bandwidth that the single socket solution provides, and likely also more frequency throttling due to the lower heat dissipation. Nevertheless, in both real applications we tested the single AMD 7702 processor is faster than a Intel equipped machine with two 6230 CPUs.

Finally, the price point at which we can obtain machines with these CPUs will be an important factor as well. At the time of the writing, CHPC can obtain roughly the same price for the single CPU AMD 7702P node with 256 GB of RAM as for the dual CPU Intel 6230 node with 192 GB of RAM, while the cost of the dual CPU AMD 7452 node with 256 GB RAM is 50% higher. With this pricing in mind, the single CPU AMD 7702P node comes out as the best choice.