

Experimental analysis of vectorized instructions impact on energy and power consumption under thermal design power constraints

Amina Guermouche
Télécom SudParis, Evry, France
amina.guermouche@telecom-sudparis.eu

Anne-Cécile Orgerie
Univ. of Rennes, Inria, CNRS, IRISA, Rennes, France
anne-cecile.orgerie@irisa.fr

Abstract—Vectorized instructions were introduced to improve the performance of applications. However, they come with an increase in the power consumption cost. As a consequence, processors are designed to limit the frequency of the processors when such instructions are used in order to maintain the thermal design power.

In this paper, we study and compare the impact of thermal design power and SIMD instructions on performance, power and energy consumption of processors and memory. The study is performed on three different architectures providing different characteristics and four applications with different profiles (including one application with different phases, each phase having a different profile).

The study shows that, because of processor frequency, performance and power consumption are strongly related under thermal design power. It also shows that AVX512 has unexpected behavior regarding processor power consumption, while DRAM power consumption is impacted by SIMD instructions because of the generated memory throughput.

Index Terms—Power consumption; energy efficiency; SIMD instructions; TDP; memory.

I. INTRODUCTION

The race for computing performance has led the computer vendors to introduce many features and new techniques to run computations as fast as possible. Such hardware improvements allow the supercomputers in the TOP500 to gain 6 orders of magnitude in terms of performance in the last 25 years [1].

Turboboost allows the processor to run at higher frequencies than the base one in order to increase performance. Simple Instruction Multiple Data (SIMD) model is another of these hardware techniques. In the SIMD model, the same operation is executed simultaneously on different elements of a vector or different data points. For instance, several iterations of the same loop for a vector/vector addition can be processed at once. Figure 1 shows the difference between an SIMD processor and a scalar processor.

This performance race comes with side effects in terms of power consumption and heat dissipation. Indeed, the power efficiency of supercomputers in the TOP500 has gained 3 orders of magnitude in the last 15 years – this metric being collected since fewer years than performance [1]. It means that the power consumption of supercomputers keeps increasing resulting in larger and larger heat dissipation at the processor

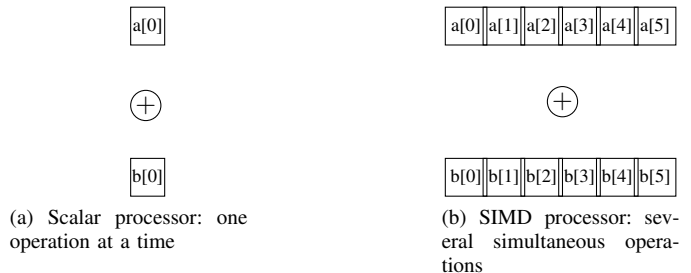


Fig. 1: Difference between SIMD and scalar operations. Example: addition of two vectors

level, and consequently, because of thermal limits, to a growing fraction of dark silicon [2]. To prevent physical damages due to heat, processors are designed to respect a thermal design power (TDP). TDP is the average power that the processor dissipates when operating at the base frequency with all cores active [3]. The base frequency is defined as the frequency when turboboost is disabled.

The performance of hardware improvements can be limited by the TDP. Indeed, when the processor detects SIMD instructions, additional voltage is applied to the core. With the additional voltage applied, the processor could run hotter, requiring the operating frequency to be reduced to maintain operations within the TDP limits. This is also the case for turboboost frequencies [4].

The TDP limit enforcement impacts the execution of applications in a non trivial manner. As an example, we run HPL, a CPU-intensive HPC benchmark, with and without turboboost on server chifflet. This experiment will be described in details later in the paper. Here, we only provide the execution time in Table I as an example. It illustrates that, unexpectedly, turboboost does not significantly increase the performance (0.26% difference) on this server for this application, which is optimized for vectorized instructions.

	with Turboboost	without Turboboost
Execution time (s.)	799.068	801.166

TABLE I: Execution time of HPL on chifflet with AVX2 while enabling and disabling turboboost (average on 5 runs).

The goal of this paper is to experimentally study such behaviors and provide scenarios where unexpected results may be obtained. The idea consists in identifying the impact of thermal design power on SIMD instructions. In order to do so, we will examine the behavior of different application profiles, when using different SIMD instructions on different hardware platforms. Note that we will only focus on Intel architectures in the remainder of the paper. This study aims at analyzing and comparing the power and energy consumption of each type of instruction and presenting cases of abnormal and unexpected behaviors.

The paper is organized as follows. Section II describes the experimental testbed. The used applications are detailed in Section III and experimental results are analyzed in Section IV. Related work is presented in Section V. Section VI concludes this experimental analysis.

II. ARCHITECTURE AND EXPERIMENTAL TESTBED

This section briefly presents SIMD instructions characteristics. Then it describes the processors used for the experiments and the power measurement methodology.

A. SIMD instructions

As stated before, SIMD instructions allow the same operation to be executed simultaneously on different elements of a vector or different data points. The number of simultaneous operations depends on the registers' size provided by the processors. Intel implements floating point SIMD instructions since the late 90's with the introduction of Streaming SIMD Extensions (SSE). Registers of 128 bits (16 Bytes) were used to hold 2 double-precision floats or 4 single-precision floats. Advanced Vector Extensions (AVX) appeared in 2010. The registers' size was doubled (256 bits) for floating point operations. However, the 128-bits integer SIMD instructions were not expanded. Finally, since the Haswell processor (2013), AVX2 extensions were introduced. They expand most 128-bits SIMD integer instructions to 256 bits. Moreover, AVX2 extension adds fused multiply-add instructions (FMA) [5]. AVX-512 is a 512-bits extension of the 256 bits operations (for floating point and integer operations). They are implemented in the Intel Xeon Phi and Skylake CPUs (2015).

B. Target platform

For the experiments, we used three servers: two from the Grid'5000 [6] platform (nova and chifflet) and a Skylake processor located in our lab (skylake), at Telecom SudParis. We chose these nodes because they do not provide the same characteristics: nova runs at the same frequency regardless of the SIMD instructions being used. chifflet and skylake frequencies are impacted by SIMD instructions. Besides, skylake provides AVX512 instruction set. The nodes are described below and their characteristics in terms of TDP and frequencies are summarized in Table II:

- nova: The Nova cluster, located in Grid'5000 Lyon site, is equipped with 23 Intel(R) Xeon(R) CPU E5-2620 v4. All experiments were run on nova-11. It is equipped with 2

CPUs, 8 cores per CPU and 64GB of memory. It provides SSE, AVX and AVX2 instruction sets. Note that for this processor, turboboost frequency does not depend on the SIMD instruction being used.

- chifflet: Chifflet is a cluster located in Grid'5000 Lille site and is equipped with 8 Intel Xeon E5-2680 v4. We used chifflet-1 for our experiments. It is equipped with 2 CPU, 14 cores per CPU and 768 GB of memory. It provides SSE, AVX and AVX2 instruction sets. For this processor, the frequency varies according to the SIMD instruction used (as shown in Table II).
- skylake: skylake is a server located in our lab at Telecom SudParis. It has 4 Intel(R) Xeon(R) Gold 6130, each equipped with 16 cores. Each NUMA node has 63 GB of memory. Thus, skylake has 64 cores and 252GB of memory in total. It provides SSE, AVX, AVX2 and AVX512 instruction sets. Like chifflet, SIMD instructions have an impact on the turboboost frequency. Moreover, AVX512 has also an impact on the base frequency as shown in Table II.

All platforms run under Intel Pstate with powersave governor. Table II provides the idle, base frequency, and characteristics for each server. It also shows the turboboost frequency depending on the SIMD instruction. Note that the turboboost frequency is not the frequency of the processor during the whole execution, but rather the frequency when SIMD instructions are used. Thus, higher frequencies may be observed during the execution time. On nova, the turboboost frequency is independent from the SIMD instructions being used while skylake shows the same base and turboboost frequencies when using AVX512. Moreover, when reaching TDP, processor may run at lower frequencies on chifflet and skylake as we will observe in Section IV. skylake data were extracted from [7] while nova and chifflet data can be found in [8]. One should note that while TDP constitutes a physical hard limit, a processor can briefly goes beyond it due to thermal inertia.

	nova	chifflet	skylake
number of cores	16	28	64
idle frequency (GHz)	1.2	1.2	1
base frequency (GHz)	2.1	2.4	2.1
TDP (W) per socket	85	120	125
AVX512 Base frequency (GHz)	-	-	1.9
Turboboost SSE frequency (GHz)	2.3	2.9	2.8
Turboboost AVX frequency (GHz)	2.3	2.8	2.5
Turboboost AVX2 frequency (GHz)	2.3	2.8	2.4
Turboboost AVX512 frequency (GHz)	-	-	1.9

TABLE II: Target platforms characteristics extracted from processors documentation

C. Power measurements methodology

In our experiments, we measure the power consumption and the execution time. Execution time is provided by the applications themselves.

In our study, power measurements rely on LIKWID [9] (version 4.3.0¹). LIKWID is a set of command line tools that

¹LIKWID commit: 233ab943543480cd46058b34616c174198ba0459

are able to read performance counters, pin threads, In order to measure the power consumed during the execution of an application, we use `likwid-perfctr` which reads the corresponding hardware counters. We measure both sockets and DRAM power consumption of the target platforms. Note that the words package, socket or processor will be used in the document. `likwid-perfctr` relies on Running Average Power Limit (RAPL) counters. RAPL was introduced by Intel in order to stay within the power limit. It uses Dynamic Voltage and Frequency Scaling (DVFS) to guarantee the desired power limit. RAPL interface describes Model Specific Registers (MSRs). These registers provide energy consumption information for components such as the processor and the DRAM. On processors like Intel Sandy Bridge, RAPL was based on a modeling approach. Since the Intel Haswell generation, the processors have fully integrated voltage regulators (FIVR) that provide actual measurements and allow for more accuracy [10]. Note that the literature provides many studies on RAPL accuracy for both DRAM [10], [11] and processor [12].

In all our experiments, the measurements are performed every second, with no overhead for all the applications. The mean of the power consumption is computed over the whole execution. The results presented in Section IV represent the average over 5 runs. Regarding measurements errors, all configurations show a small standard deviation. The maximum package power difference observed was on skylake with SVD_Bulge using SSE (4W over 121W). The variation in DRAM measurements are very low (< 1%).

III. APPLICATIONS

This section describes the applications and the configuration parameters that we used. It also provides a characterization of these applications.

A. Applications description and configuration

In order to study AVX impact on power and energy consumption, we target 4 different HPC applications, which have automatic vectorization (by setting a compilation flag or an environment variable). We used applications with different CPU behavior and/or available options: HPL [13] and Plasma svd [14] which use the Math Kernel Library (MKL), AFF3CT [15] and STORM [16]. Note that AFF3CT and STORM only use integer. The following paragraphs present a short description of the applications and their configurations.

- High Performance Linpack (HPL) [13]: HPL is a software package that solves dense linear algebra systems. It is used as a reference benchmark to compute the performance of the supercomputers in the TOP500 [1]. All the parameters are presented in Table IIIa. We used HPL version 2.2 compiled with OpenMPI 3.3.1 and with MKL library version 1_mkl_2018.3.22. MKL allows choosing the right SIMD instructions by setting the environment variable `MKL_ENABLE_INSTRUCTIONS` to `SSE4_2`, `AVX`, `AVX2` or `AVX512` (for skylake only). Note that HPL is a CPU-intensive application.

	N	NB	(PxQ)
nova	58912	224	(4x4)
chifflet	100996	224	(4x7)
skylake	91840	224	(8x8)

(a) HPL

	N	NB
nova	20000	200
chifflet	28000	200
skylake	48000	200

(b) SVD

TABLE III: HPL and SVD setup for nova, chifflet and skylake

- PLASMA Singular Value Decomposition (SVD) [14]: The SVD decomposition computes the singular values of a matrix. It is performed in three steps. Readers can refer to [14] for more details on the algorithm. The first step is referred to as SVD_Band in the remainder of the paper while the second is referred to as SVD_Bulge. In our configuration, we used the SVD version implemented in PLASMA [17]. PLASMA is a software package for solving problems in dense linear algebra using multicore processors and Xeon Phi coprocessors. We used PLASMA version 2.8.0 compiled with MKL library version 1_mkl_2018.3.22. Thus, setting the desired SIMD instruction is done the same way as HPL. Table IIIb details the parameters values used for the target nodes. We fixed the size such that the first two phases last long enough to have several power measurements. The first phase is CPU-intensive while the second phase is memory-intensive. Note that the third phase is very short (few seconds) for these configurations. For this reason, we will not present an analysis for this phase.
- Seed-based Read Mapping tool for SOLiD or Illumina sequencing data (STORM) [16]: STORM is a read mapping tool based on mapping data between reads and a reference genome. It runs several phases. We present the results for the search algorithm used in the application. As STORM uses integers in the SIMD parts, only SSE, AVX2 and AVX512 results will be presented. Note that STORM is a CPU-intensive application. We will no further detail the software. The user can refer to [16] for more details. Section ?? is an appendix detailing how we generated the input reads for STORM. Note that the SIMD instructions are handled within the code. Setting the desired SIMD instruction set is done using the right compilation flag.
- A Fast Forward Error Correction Toolbox (AFF3CT) [15]: AFF3CT is a library used for forward error correction. Forward Error Correction (FEC) is used to control errors during data transmission in order to enable efficient communications over noisy channels. It is done through encoding (by the sender) the data frame and decoding (by the receiver). We will no further detail Forward Error Correction, but we will present how the authors used vectorized instructions in their decoding solution. The decoder takes a set of frames as input. The decoding of the frames is vectorized. In order to do so, the frames are first buffered and then the vectorized algorithm is applied on the frames. Thus, depending on the SIMD instruction being used, the number of loaded frames differs. For instance, when using AVX2 instructions, twice the number of frames are loaded

simultaneously compared to using SSE instructions. Just like SToRM, the vectorization is handled within AFF3CT code and a compilation flags allows setting the desired SIMD instruction. Note that this application is CPU-intensive and uses integers which are not supported by AVX instructions. As a consequence, there will be not results for AVX with AFF3CT.

On nova and chifflet, all applications were compiled against gcc 4.9.2 and the machines were running a Linux Debian 3.16.43-2+deb8u5 (2017-09-19) x86_64 GNU/Linux with a kernel version 3.16.0-4-amd64. On skylake, applications were compiled against gcc 8.2.1 and the machine is running an Arch Linux 4.18.9-arch1-1-ARCH x86_64 GNU/Linux. On all architectures, hyperthreading was disabled.

B. Applications characterization

In order to better understand these applications, we start by characterizing them as CPU or memory-intensive. Note that since AFF3CT and SToRM use only integer operations, we cannot rely on the usual FLOPS and the roofline model to characterize the applications. For this reason, in order to characterize the applications, we vary the CPU frequency and observe the impact on the execution duration of the different applications. This does not determine if one application is more CPU-intensive than another, but rather whether it is CPU-intensive or memory-intensive. Note that HPL mainly computes matrix-matrix multiplications which are CPU-intensive. SVD_Band and SVD_Bulge have been characterized in the literature [14].

Figure 2 shows the execution time according to the CPU frequency on chifflet. We used the same configurations as the ones described in Section III-A for all applications. The impact on SVD_Bulge is as expected. Since it is a memory-intensive phase, CPU frequency should have little impact on its performance. SVD_Band and HPL behaviors perform also as expected: the lower the frequency, the slower the application. AFF3CT and SToRM seem to have a similar behavior. This indicates that SToRM and AFF3CT are also CPU-intensive.

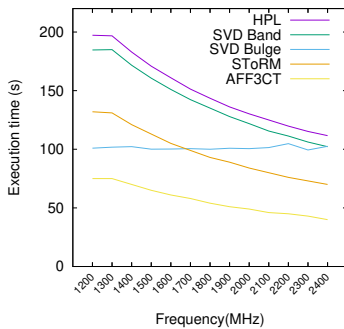


Fig. 2: Frequency impact on execution time on chifflet

IV. EXPERIMENTAL RESULTS

This section presents SIMD instructions behavior when the power consumption reaches TDP and when it does not.

	application	SSE	AVX	AVX2	AVX512	AVX512 noTB
chifflet	HPL	2.9	2.6	2.4	-	-
	SVD_Band	2.9	2.8	2.6	-	-
	SVD_Bulge	2.6	2.6	2.5	-	-
	SToRM	2.9	-	2.9	-	-
	AFF3CT	2.9	-	2.8	-	-
skylake	HPL	2.7	2.4	2.4	2.0	1.9
	SVD_Band	2.7	2.4	2.4	2.1	1.9
	SVD_Bulge	2.8	2.4	2.4	1.9	1.9
	SToRM	2.8	-	2.8	2.4	2.1
	AFF3CT	2.7	-	2.7	2.4	2.1

TABLE IV: Average observed frequency, in GHz, on chifflet and skylake over all cores.

To do so, for each application and each platform, we use the available SIMD instructions on each machine. For each configuration, we will explain the observed behavior of SIMD instructions and if TDP is impacting this behavior. This study is on applications socket power (Section IV-A) consumption, performance (Section IV-B), energy consumption (Section IV-C) and DRAM power (Section IV-D) and energy (Section IV-E) consumption. The goal here is not to compare the different platforms, but rather to study the behavior of SIMD instructions under different configurations.

Figure 3 shows the execution time ratio (Figure 3a), socket power ratio (Figure 3b), socket energy consumption ratio (Figure 3c), DRAM power ratio (Figure 3e) and DRAM energy consumption ratio (Figure 3f) of the different applications on nova while using the different SIMD instructions when turboboost is activated and not activated. Figures 4 and 5 show the same ratios for chifflet and skylake. For each application, the ratio is computed over the default SIMD instruction which is AVX2 for nova and chifflet and AVX512 on skylake.

A. Impact on applications socket power consumption

In this section, we present how SIMD instructions impact power consumption. We describe the results for each platform separately since they are not similar.

1) *nova: SIMD at the same frequency:* On nova (Figure 3b), for all CPU-intensive applications except AFF3CT, the larger the SIMD instruction vector size, the larger the power consumption. This is the typical behavior described in the literature [5]. Regarding AFF3CT, SSE and AVX2 have the same power consumption. This is because even if AVX2 is known to consume more power (as shown for the other applications), the performance of SSE are better (as shown in Figure 3a and will be discussed in Section IV-B) and this balances the power consumption with AVX2. We will observe a similar behavior on chifflet.

Regarding SVD_Bulge, which is memory-intensive, one can see that SIMD instructions have a very small impact on power consumption (an increase of 2.8% from SSE to AVX2 and 2.07% from AVX to AVX2). This slight impact is explained by the small impact on the performance as shown in Figure 3a which shows the use of SIMD instructions in SVD_Bulge.

Finally, disabling turboboost shows the same behavior. As the frequency is lower, the power consumption when disabling

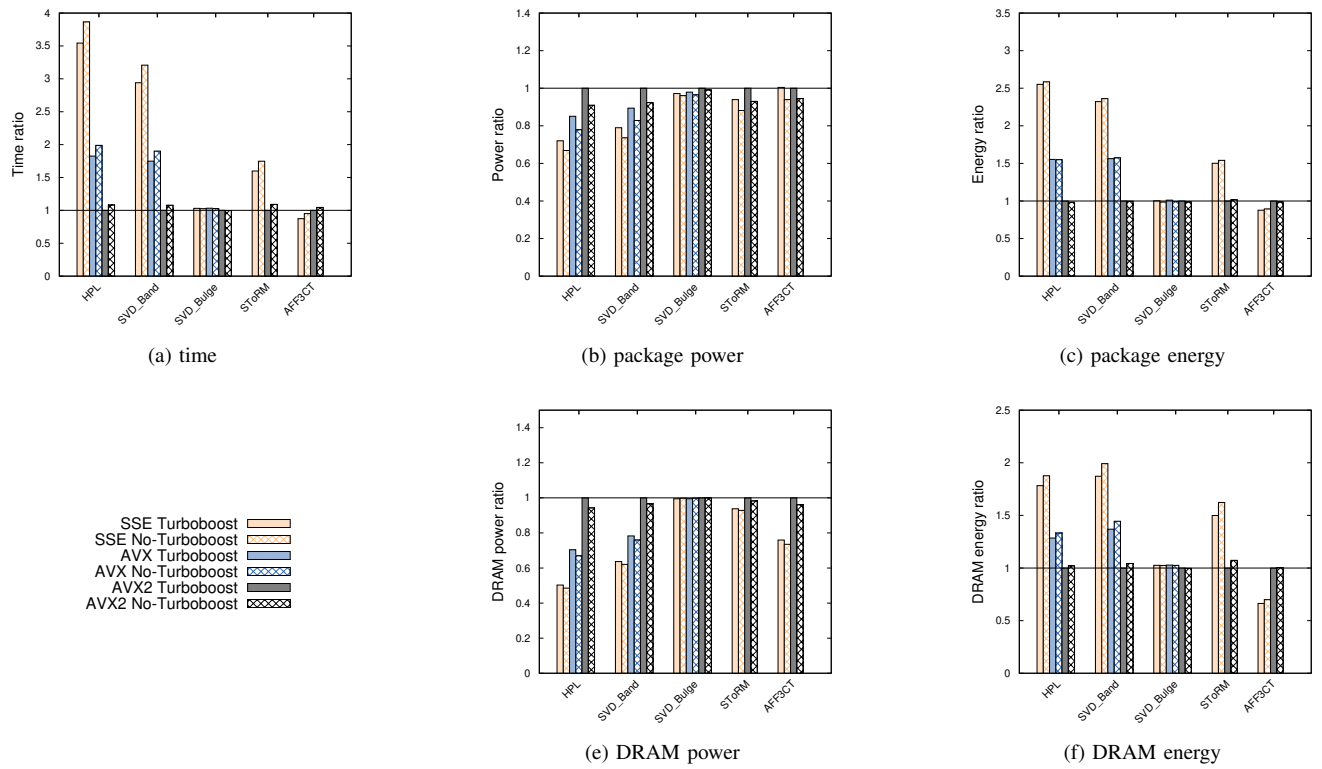


Fig. 3: Comparison of SIMD instructions with and without turboboost on nova

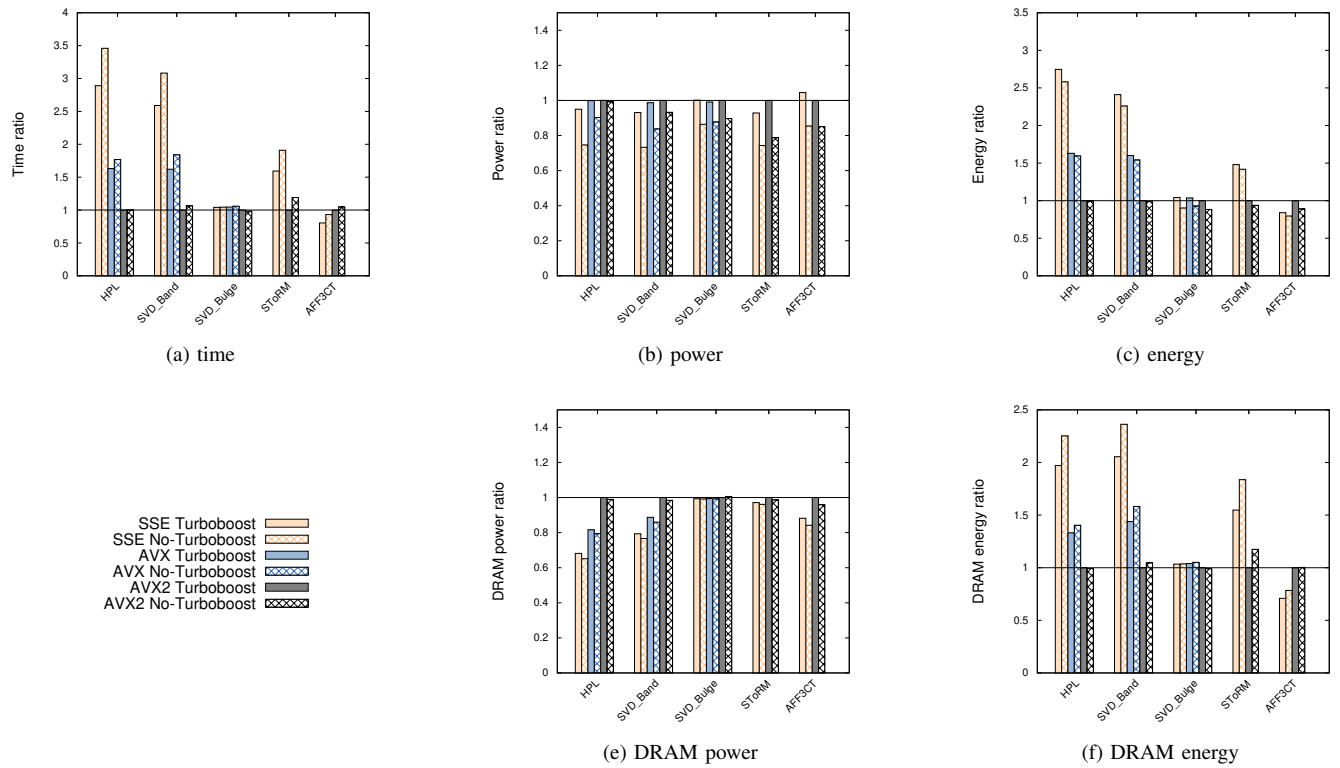


Fig. 4: Comparison of SIMD instructions with and without turboboost for on chifflet

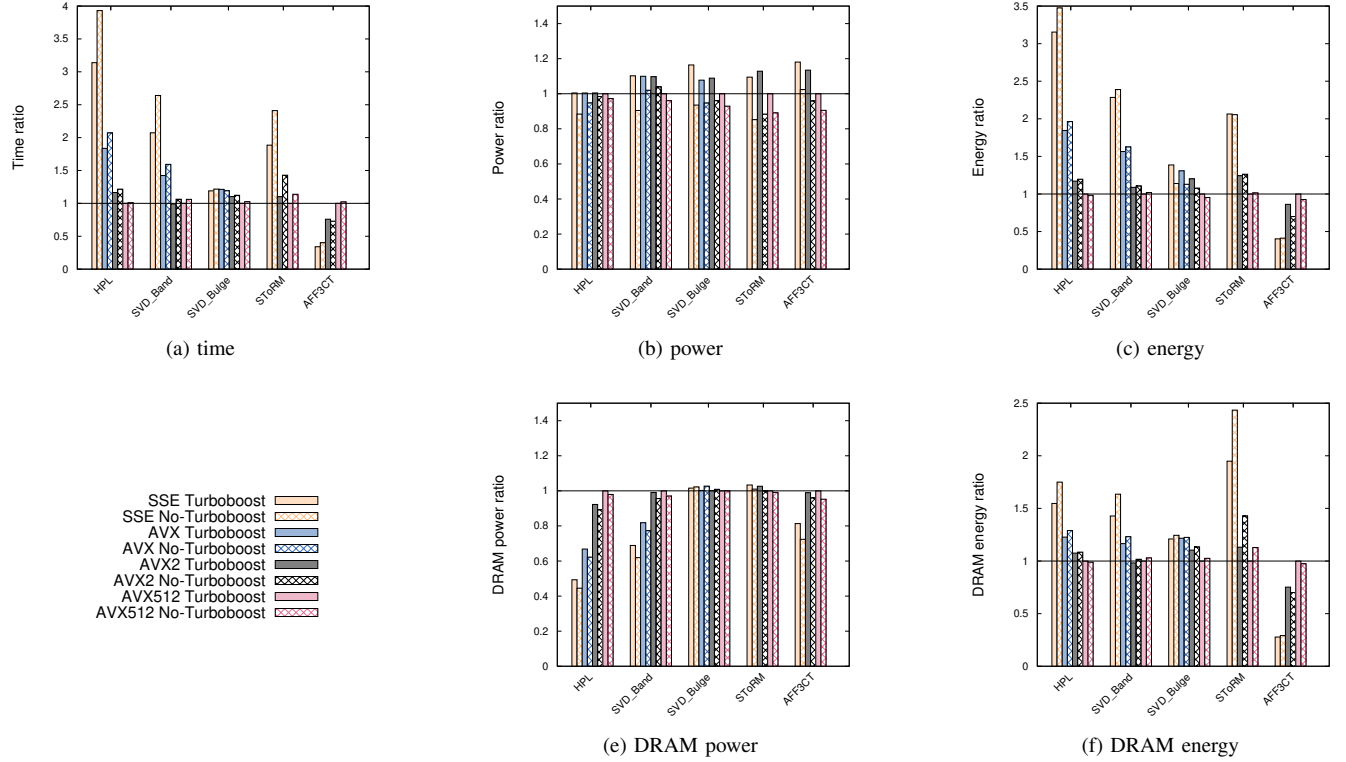


Fig. 5: Comparison of SIMD instructions with and without turboboost for on skylake

turboboost is lower for all applications which is the normal behavior. As expected, for all applications, the power ratios of all instructions over AVX2 when enabling turboboost are the same as the ratios when turboboost is disabled. This indicates that for nova, since all instructions run at the same frequency, the type of used SIMD instructions is the major factor impacting power consumption and shows that, as expected, as the buffer size increases, the power consumption increases.

2) *chifflet: SIMD at different frequencies:* On chifflet (Figure 4b), the behavior is similar to nova: the larger the SIMD buffer size, the larger the power consumption for all CPU-intensive applications except AFF3CT. However, the difference between the power consumption of the different instructions is less important on chifflet compared to nova for HPL and SVD_Band. As a matter of fact, on nova using AVX2 consumes 28% more than using SSE for HPL and 21% more for SVD_BAND. On the other hand, on chifflet, using AVX2 only consumes 5% more than using SSE for HPL and 7% more for SVD_BAND.

For these two applications, AVX is almost at the thermal design power while AVX2 reaches it. Because of that, for HPL and SVD_Band, the frequency is reduced for AVX and AVX2. This explains why they have the same power consumption. Table IV shows the observed frequencies for all applications on chifflet and skylake. For HPL, AVX2 frequency is equal to the base frequency (2.4GHz) while AVX frequency is 2.6GHz. This explains why disabling turboboost shows the same values

for performance and power consumption for AVX2 (as hinted in Table I). For SVD_Band, the frequency is 2.6GHz for AVX2 and 2.8GHz for AVX. Thus, unlike HPL, AVX2 does not run at the base frequency, which explains the difference in performance and power consumption between AVX and AVX2 for SVD_Band compared to HPL. This explains why disabling turboboost shows a greater impact on power consumption for SVD_Band compared to HPL (9.76% of power consumption increase when using AVX2 compared to AVX for HPL and 16.24% for SVD_Band).

Regarding SVD_Bulge, SIMD instructions have no impact on the power consumption. Figure 4b shows that AVX consumes slightly less power than to AVX2 or SSE, but this difference is 1.6%, which is in the error measurement range. When setting all processors to the same frequency (No-Turboboost plot), one can see a slight difference between the instructions (6.5W for SSE and 4W for AVX compared to AVX2). This is due to the use of SIMD instructions in SVD_Bulge (as will be shown in section IV-B). However, the variation is small (at most 3.5%).

STORM shows an interesting behavior. According to Table IV, AVX2 reaches the same frequency as SSE. Thus, the ratio SSE/AVX2 is the same regardless of turboboost. Moreover, just like AFF3CT, it uses only integers, which means that for all instructions, only half of the buffer is used. This impacts power consumption. As a matter of fact, STORM consumes 101W per socket while HPL consumes 124W for

AVX2. Note that we do not compare the applications since they may not have the same computation intensity, we only state that using half of the buffer most likely leads to less power consumption and thus higher frequencies can be used.

Finally, regarding AFF3CT, unlike the other applications, the larger the SIMD buffer size, the lower the power consumption. For instance, using SSE consumes 4.3% more power than using AVX2. This is due to the fact that SSE outperforms AVX2 for this application. Moreover, when we look at the power consumption when turboboost is disabled, it is the same. This means that for AFF3CT on chifflet, the frequency is the parameter influencing the power consumption because SSE outperforms AVX2.

3) *skylake AVX512*: Applications on skylake exhibit a completely different behavior compared to the other platforms: larger SIMD buffer does not mean more power consumption. This means that AVX512 shows better power consumption compared to other instructions. This is different from the behavior observed on nova and chifflet.

On Figure 5b, for all applications, the power consumed when using AVX512 (thus SIMD instruction with the largest buffer) is lower or equal (equal for HPL) compared to using other buffers. The reason why SSE, AVX and AVX2 consume more power than AVX512 is that they reach higher frequencies as shown in Table IV. This can be already be observed for AFF3CT and STORM when disabling turboboost. We will start by describing the behavior for these two applications before moving to HPL and SVD.

AFF3CT and STORM have a specific behavior on skylake: AVX2 and AVX512 do not run at the same frequency as specified in Table II. As a matter of fact, AVX2 frequency is the same as SSE whereas AVX512 frequency is 2.4GHz when turboboost is enabled and is the base frequency when turboboost is disabled. Thus, when turboboost is disabled, SSE, AVX2 and AVX512 run at the same frequency.

AFF3CT shows a similar behavior to its behavior on chifflet: SSE has a larger power consumption compared to AVX2 despite running at the same frequency (an increase of 3.8%). When running at the same frequency (when turboboost is disabled), we observe a similar behavior where SIMD instruction (SSE) with the best performance consumes more than the others (using SSE consumes 6.4% more than using AVX2 and 11.6% more than using AVX512).

Regarding STORM, AVX2 reaches the thermal design power and consumes more energy than SSE (by 3%). Besides, SSE consumes 8.65% more than AVX512 because of AVX512 frequency. This can be verified when observing the results when turboboost is disabled. Since all instructions run at the same frequency, AVX512 consumes more power than SSE. AVX2 and AVX512 have roughly the same power consumption when disabling turboboost.

For HPL, all SIMD instructions reach the TDP which is why they have the same power consumption. Even when turboboost is disabled, AVX2 is almost at TDP.

Regarding SVD_Band, SSE, AVX and AVX2 are at the thermal design power. One interesting observation is that, as

shown in Table IV, on average, SSE and AVX run at lower frequencies compared to the values presented in Table II for HPL and SVD_Band. This is the behavior described by Intel when processors reach the thermal design power [5].

Finally SVD_Bulge also shows a similar behavior where SSE is the most power consuming compared to AVX and AVX2 since it consumes 7.45% more power compared to AVX and 6.5% compared to AVX2. This is due to frequency which is higher for SSE as shown in Table IV. AVX512 consumes less than the other instructions. The frequency impact can also be verified when observing the plots where turboboost is disabled. In this case, the power consumption of SSE, AVX and AVX2 is the same, while AVX512 is still below because its frequency is lower when turboboost is disabled.

For HPL, SVD_Band and SVD_Bulge, most of the time AVX2 consumes more than AVX512 even when turboboost is disabled. We believe that this is because of the frequency. Recall that AVX2 frequency is 2.4GHz while AVX512 frequency is 1.9GHz. In order to validate our assumption, we ran these three applications at 1.9GHz. Note that we focus only on these three applications since on STORM and AFF3CT, all SIMD instructions already run at the same frequency when turboboost is disabled. Note also that differences will be observed since running applications with AVX512 instructions will have the frequency vary depending on the CPU load and performed instructions, whereas we fixed the frequency to 1.9GHz so it cannot vary. Figure 6 shows the power consumption of HPL and SVD applications when running on skylake at 1.9GHz. Regarding power consumption, HPL and SVD_Bulge have the expected behavior observed on nova. For SVD_Band however, AVX2 and AVX512 shows the same power consumption. We believe that this is because they have the same performance.

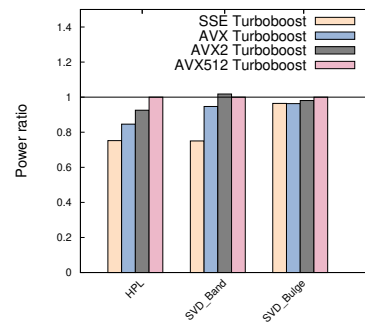


Fig. 6: Comparison of SIMD instructions power consumption when running at AVX512 frequency (1.9GHz) on skylake for HPL and SVD

As a conclusion, for CPU-intensive applications, both SIMD instructions and frequency seem to impact power consumption. On the other hand, for Memory-intensive applications, frequency seems to be the most impacting factor.

B. Impact on performance

In this section, we study the performance behavior of applications. Note that in this section, we will provide explanations on why applications do not have the expected behavior. We will however not explain why an application running using AVX2 is not at least half as slow as running AVX512 for instance. This is because sometimes this problem is due to the algorithm and how the application is designed, which is out of our expertise since we did not design any of these applications.

As expected, on all platforms, the behavior of CPU-intensive applications is similar: the larger the buffer size, the better the performance, except for AFF3CT. We will explain AFF3CT’s behavior in the end of this section.

On nova and chifflet, all CPU-intensive applications show great performance gain when increasing the vector size. For instance, HPL shows a $\times 3.54$ improvement on nova and a $\times 2.89$ on chifflet when using AVX2 compared to SSE. On chifflet, HPL is the only application which power consumption reaches the thermal design power. As such, its frequency is lowered as shown in Table IV. The average frequency observed with HPL is the same as the base frequency. This is why the performance when disabling turboboost for AVX2 are so close to the performance with turboboost. Using AVX2 over SSE when disabling turboboost reduces HPL execution time by 71%. This is mainly due to SSE performance which are reduced by a factor of $\times 1.19$ when disabling turboboost.

For SVD_Band and SToRM, disabling turboboost shows an even larger impact when increasing the SIMD buffer size. As a matter of fact, using AVX2 over SSE reduces SVD_Band execution time by 67% and SToRM execution time by 48%.

On skylake however, using AVX512 does not seem to improve performance as much as using AVX2 compared to SSE for all CPU-intensive applications (except AFF3CT). For HPL, one of reasons lies behind the memory bandwidth. As a matter of fact, the configuration HPL + AVX512 reaches the maximum sustainable memory bandwidth observed using STREAM [18] (approximately 30GB/s) which explains why its performance are limited compared to AVX2. Moreover, because of TDP, HPL frequency when using AVX512 is lower than when using AVX2 as stated in Section IV-A. Note that we tried running HPL on skylake while forcing the frequency to 1.9GHz. This showed a small improvement of the performance ratios (1.16 with the normal behavior and 1.21 when forcing the frequency to 1.9GHz), but not as great as expected. The frequency has also an impact on SToRM performance since disabling turboboost shows a better improvement of AVX512 over AVX2 (9.25% using the default configuration and 30% when disabling turboboost). This is because for SToRM, as shown in Table IV, all instructions run at the same frequency when disabling turboboost on skylake. Finally, SVD_Band shows similar performance results with the configuration that we used. Even when forcing AVX2 to run at 1.9GHz, it still provides the same performance as AVX512. We could not find a reason behind this behavior. At 1.9GHz, SVD_Band

does not reach the thermal design power with AVX512, the frequency does not have an impact and neither does the memory bandwidth or the cache misses. We will no further investigate this behavior.

For SVD_Bulge, AVX has little to no impact on the performance on nova and chifflet. On skylake however, one can see that there is an impact. We studied the performance (in flop/s) of SVD_Bulge using LIKWID and FLOPS_DP group. The group provides double precision floating point performance in addition to vectorization performance and ratios. Vectorization shows an impact of few GFlops/s on SVD_Bulges while the total performance does not exceed 44 GFlops/s. This means that despite being memory_intensive, this application still computes floating point operations and still uses vectorization which explains why SIMD instructions have an impact on its performance.

AFF3CT exhibits a behavior that is completely opposite to the other applications: the smaller the SIMD instruction buffer size, the better the performance. In other words, for this application, with the parameters that we used, it is better to use SSE rather than AVX2 or AVX512. As stated in Section III-A, AFF3CT handles frames which are loaded in the memory. As a consequence, the larger the buffer (AVX2 or AVX512), the greater the number of loaded frames. Therefore, for the configuration that we used for AFF3CT, for AVX2 and AVX512, the load exceeds the cache. We compared the ratio of cache misses of the different applications when using SSE, AVX2 and AVX512 on skylake (recall that AVX is not supported by AFF3CT). These results are shown in Table V and represent an average over all sockets. Note also that the results show only CPU-intensive applications (thus we exclude SVD_Bulge). Although all applications show a difference when comparing their cache miss ratios between AVX512 and SSE or AVX2, AFF3CT shows the highest difference, especially between SSE and AVX512. As a matter of fact, with AFF3CT, using AVX512 generates more than 16.5 times more L3 cache misses than using SSE while this ratio is at most 1.80 for the other applications. Further details are presented in [15].

Application	AVX512/SSE	AVX512/AVX2
HPL	1.05	1.03
SVD_Band	1	0.83
SToRM	1.80	1.06
AFF3CT	16.55	4.82

TABLE V: Applications average L3 cache misses over all socket on skylake. The results are presented as a ratio of AVX512 cache misses over SSE or AVX2 cache misses.

C. Impact on applications socket energy consumption

Regarding energy consumption, on all platforms, the performance seems to be the major factor impacting energy consumption. This is because SIMD instructions have a large impact on performance compared to power consumption. Thus, except for AFF3CT, for which SSE is better, it is better to use the SIMD instruction with the largest buffer.

Regarding frequency, it has more impact on chifflet and skylake since each SIMD instruction runs at a different frequency. For these two platforms, disabling turboboost shows little to no impact on energy consumption.

D. Impact on applications DRAM power consumption

Figures 3e, 4e and 5e present the DRAM power on nova, chifflet and skylake.

For all applications, on all platforms (except SToRM on skylake), the larger the buffer size, the higher the DRAM power consumption. In order to explain this behavior, we studied the impact of vectorized instructions on the applications memory throughput. Table VI shows the mean memory throughput of all applications on all platforms over all sockets. Note that not all applications show the same memory throughput over all socket. For AFF3CT, the sockets may have different memory throughput but they remain in the same order of magnitude (a difference of 4000 MByte/s over 30000 MByte/s between the sockets on skylake). For SToRM however, the difference between the sockets is noticeable. It is especially the case when using SSE since it varies from 330 to 1900 MByte/s between socket 0 and socket 4 on skylake. For SVD_Bulge, only socket 0 has a large memory throughput compared to the other sockets. For SVD_Bulge, we present only the memory throughput for socket 0.

The results show a strong correlation between the DRAM power consumption and the memory throughput. First of all, one can notice that, for CPU-intensive applications, as the size of the instruction buffer increases, the memory throughput increases as well [19] which also leads to larger power consumption. SToRM seems to exhibit different behavior on skylake where SSE has a larger DRAM power consumption. This behavior is still correlated to memory throughput. Note that for SToRM, the power consumption is low compared to the other applications: it is roughly half the power consumed by the other CPU-intensive applications for AVX2 and AVX512. SToRM throughput (Table VI) also shows the small memory movement of SToRM compared to the other applications on all platforms. For instance, on skylake using AVX512, SToRM memory throughput is $\times 0.022$ HPL throughput and $\times 0.037$ SVD_Band throughput. Finally, as stated before, SToRM memory throughput is different between the sockets. However, there is still a correlation between each socket DRAM power consumption and memory throughput. In other words, for each socket, the larger the memory throughput the larger the DRAM power consumption.

Memory-intensive application SVD_Bulge shows almost no impact of SIMD instructions on DRAM power consumption (with a the highest ratio is of 1.01 between SSE and AVX512). Moreover, only the first socket has an impact on power consumption. The other sockets have a very low power consumption (15.9W for socket 0 and between 4.3W and 4.9W for the other three sockets).

Finally, turboboost has also an impact on the DRAM power consumption. This is also due to memory throughput which is

	application	SSE	AVX	AVX2	AVX512
nova	HPL	3656.98	7929.83	15328.66	-
	SVD_Band	2943.61	4812.55	8285	-
	SVD_Bulge	24444.18	24407.1	24911.74	-
	SToRM	325.63	-	467.55	-
	AFF3CT	7993.41	-	12355.53	-
chifflet	HPL	7962.2	15471.27	27203.38	-
	SVD_Band	6394.85	9721.01	15405.46	-
	SVD_Bulge	38899.47	38706.24	38823.91	-
	SToRM	578.61	-	769.03	-
	AFF3CT	16754.13	-	25663.37	-
skylake	HPL	8489.71	15441.3	26352.54	30375.21
	SVD_Band	9099.95	12685.94	17936.55	18070.59
	SVD_Bulge	28417.24	28687.84	27831.89	26832.16
	SToRM	1173.75	-	938.33	670.16
	AFF3CT	20523.98	-	29127	27403.43

TABLE VI: Mean memory throughput over all socket, in MByte/s, for each SIMD instruction on nova, chifflet and skylake. For SVD_Bulge, only socket 0 memory throughput is presented.

lower when turboboost is disabled (as there are less requests per unit of time since the processor is slower).

E. Impact on applications DRAM energy consumption

Figures 3f, 4f and 5f present the DRAM energy consumption on nova, chifflet and skylake.

In order to measure the DRAM energy consumption, we just multiplied the total execution time by the DRAM power consumption. The results show that on all platforms and for all applications, the most performing application is the least energy consuming. Thus, for all applications except AFF3CT, using the SIMD instruction with the largest buffer provides the best energy consumption.

F. Key findings

This study provided an insight on how thermal design power and SIMD instructions impact performance, power and energy consumption for both processor and memory. From our observations, one can conclude that :

- Performance and power consumption are more and more related. Because of TDP and SIMD instructions, core frequency may be lowered which directly impacts performance.
- For most CPU-intensive applications (except applications with special design like AFF3CT), the larger the buffer size, the better the performance and the energy consumption. Moreover, turboboost has no impact on energy consumption when using the instruction with the largest buffer. Thus, if power is more important than performance, one can start with disabling turboboost.
- Frequency has more impact on power and performance of memory-intensive applications than SIMD instructions.
- The larger the SIMD buffer size, the larger is the power consumption, except for AVX512 since it runs at a lower frequency.
- DRAM power consumption is strongly correlated to memory throughput. Thus, larger SIMD buffer sizes lead to larger DRAM power consumption.

V. RELATED WORK

Many studies focused on the energy consumption of vectorization. [10], presents recent changes in Intel Haswell processors such as AVX frequencies and voltage regulators. In [20], the authors compare the time, power consumption and energy consumption of an erasure code running on traditional x86 and more recent SIMD platforms. They showed that using SIMD instructions on such applications reduces execution time and energy consumption. In [21], the authors present a comparison of Sandy Bridge Processors and Haswell-EP processors in order to show how the new changes, like the TDP, challenges the performance analysis. In [2], the authors compared multithreading and vectorization on different processors. They showed the benefit of vectorization over multithreading for energy consumption. Special instruction set impact was also studied, like load and store AVX instruction set [22]. In [23], the authors study the energy consumption of different implementations of Gaussian elimination on different architectures. These studies present a comparison of an execution with and without SIMD instructions, while we present a comparison of the instructions.

Studies like [24] and [19] can be considered complementary to our work. In [24], the authors studied the impact of SSE and AVX on applications performance. This work was done on older architectures. In [19], the authors compared the execution time, power and energy consumption of an AVX and SSE implementation of a sorting algorithm. They also showed the impact of varying the memory bandwidth on the performance and energy consumption of SIMD instructions. This work is complementary to our work since it studies the impact of SIMD instructions on another algorithm and studies other parameters such as memory bandwidth, while we studied the DRAM power consumption. Moreover, we focused on different applications with different profiles. [25] presents an energy model using codelets. They also study the energy consumption of scalar and vectorized instructions (using SSE and AVX2). They showed the effect of different profiles on the energy consumption of SSE and AVX2 instructions. They also showed the effect of data located on L2 and L3 cache on the energy consumption of the codelets. Our work is complementary since we study the power and energy consumption of applications for both socket and memory consumption, and study the impact of TDP on the performance and power consumption. In [26], the authors compared different vectorized instructions (SSE and AVX) while varying the number of threads using different Intel and ARM platforms. They also evaluated when turboboost improves energy consumption. This work is the closest to our work since the authors compared vectorized instructions and studied the impact of turboboost. However, the machines they used (Intel Ivy and Sandy Bridge) were such that using SIMD instructions had almost no impact on power. Moreover, we presented a study on the impact of SIMD instructions on DRAM power consumption.

VI. CONCLUSION

In this work, we studied the impact of vectorization and thermal design power on processor and DRAM power consumption. For that purpose, we used 3 different architectures and 5 applications with different behavior. Our conclusions showed that because of thermal design power, performance and power become less and less independent. As a consequence, when trying to understand an application performance, studying its power consumption and frequency can help understanding its behavior. Moreover, our study showed that although using SIMD instructions with larger buffer size improves performance and energy consumption, it has a negative impact on both DRAM and processor power consumption. However, AVX512 seem to have a different behavior where its power consumption is lower than the other instructions despite providing better performance.

Since power consumption is becoming a major problem, using power capping techniques may provide a good leverage to reduce power consumption. As a consequence, in the future, we plan to study the behavior of the processors when a power cap is applied. This will be especially interesting for application like AFF3CT and STORM which do not reach TDP. We also plan to study how using hyperthreading in addition to vectorized instructions can impact application behavior.

ACKNOWLEDGMENT

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see <https://www.grid5000.fr>).

We would like to thank Mathieu Faverge for his help on PLASMA. We would also like to thank Laurent Noé for his help on STORM. Finally, we would like to thank Adrien Cassagne for his help on AFF3CT.

REFERENCES

- [1] "TOP500 Supercomputer Site," <http://www.top500.org>.
- [2] J. M. Cebrian, L. Natvig, and J. C. Meyer, "Improving Energy Efficiency through Parallelization and Vectorization on Intel Core i5 and i7 Processors," in *Int. Conf. for High Performance Computing, Networking Storage and Analysis (SC)*, 2012, pp. 675–684.
- [3] "Intel® xeon® processor e5 v4 product family. thermal mechanical specification and design guide," 2017.
- [4] "Intel® xeon® processor e5 v3 product family. specification update," 2017.
- [5] G. Lento, "Optimizing performance with intel® advanced vector extensions. intel white paper," 2014.
- [6] D. Balouek *et al.*, "Adding virtualization capabilities to the Grid'5000 testbed," in *Cloud Computing and Services Science*, ser. Communications in Computer and Information Science, I. Ivanov, M. van Sinderen, F. Leymann, and T. Shan, Eds. Springer, 2013, vol. 367, pp. 3–20.
- [7] "Intel® xeon® processor scalable family. specification update," 2018.
- [8] "Intel® xeon® processor e5 v4 product family. specification update," 2016.
- [9] J. Treibig, G. Hager, and G. Wellein, "LIKWID: A Lightweight Performance-Oriented Tool Suite for x86 Multicore Environments," in *Int. Conf. on Parallel Processing (ICPP Workshops)*, 2010, pp. 207–216.
- [10] D. Hackenberg, R. Schne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, "An Energy Efficiency Feature Survey of the Intel Haswell Processor," in *IEEE Int. Parallel and Distributed Processing Symposium (IPDPS Workshops)*, 2015, pp. 896–904.

- [11] S. Desrochers, C. Paradis, and V. M. Weaver, "A Validation of DRAM RAPL Power Measurements," in *International Symposium on Memory Systems (MEMSYS)*, 2016, pp. 455–470.
- [12] K. Khan, M. Hirki, T. Niemi, J. Nurminen, and Z. Ou, "RAPL in Action: Experiences in Using RAPL for Power Measurements," *ACM Trans. on Modeling and Performance Evaluation of Computing Systems (TOMPECS)*, vol. 3, 2018.
- [13] A. Petit, R. C. Whaley, J. Dongarra, and A. Cleary, "Hpl - a portable implementation of the high-performance linpack benchmark for distributed-memory computers," *Innovative Computing Laboratory*, p. Available at <http://icl.utk.edu/hpl/>, September 2000.
- [14] A. Haidar, J. Kurzak, and P. Luszczek, "An improved parallel singular value algorithm and its implementation for multicore hardware," in *Int. Conf. for High Performance Computing, Networking, Storage and Analysis (SC)*, 2013.
- [15] A. Cassagne, T. Tonnellier, C. Leroux, B. L. Gal, O. Aumage, and D. Barhou, "Beyond Gbps Turbo decoder on multi-core CPUs," in *Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC)*, 2016, pp. 136–140.
- [16] L. Noé, M. Gírdea, and G. Kucherov, "Seed design framework for mapping solid reads," in *Research in Computational Molecular Biology*, B. Berger, Ed. Springer, 2010, pp. 384–396.
- [17] A. Buttari, J. Langou, J. Kurzak, and J. Dongarra, "A class of parallel tiled linear algebra algorithms for multicore architectures," *Parallel Computing*, vol. 35, no. 1, pp. 38–53, 2009.
- [18] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, 1995.
- [19] H. Inoue, "How SIMD width affects energy efficiency: A case study on sorting," in *IEEE Symp. in Low-Power and High-Speed Chips (COOL CHIPS)*, 2016.
- [20] H. Chen, G. Grider, J. Inman, P. Fields, and J. A. Kuehn, "An empirical study of performance, power consumption, and energy cost of erasure code computing for HPC cloud storage systems," in *2015 IEEE International Conference on Networking, Architecture and Storage (NAS)*, Aug 2015, pp. 71–80.
- [21] J. Schuchart, D. Hackenberg, R. Schöne, T. Ilsche, R. Nagappan, and M. K. Patterson, "The Shift from Processor Power Consumption to Performance Variations: Fundamental Implications at Scale," *Comput. Sci.*, vol. 31, no. 4, pp. 197–205, Nov. 2016.
- [22] T. Jakobs and G. Rüniger, "On the energy consumption of load/store AVX instructions," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2018, pp. 319–327.
- [23] T. Jakobs and G. Runger, "Examining energy efficiency of vectorization techniques using a gaussian elimination," in *Int. Conference on High Performance Computing Simulation (HPCS)*, July 2018, pp. 268–275.
- [24] P. Gepner, V. Gamayunov, and D. L. Fraser, "Early performance evaluation of AVX for HPC," in *International Conference on Computational Science (ICCS)*, vol. 4, 2011, pp. 452 – 460.
- [25] A. Mazouz, D. C. Wong, D. Kuck, and W. Jalby, "An incremental methodology for energy measurement and modeling," in *ACM/SPEC Int. Conf. on Performance Engineering (ICPE)*, 2017, pp. 15–26.
- [26] J. M. Cebrián, L. Natvig, and J. C. Meyer, "Performance and energy impact of parallelization and vectorization techniques in modern microprocessors," *Computing*, vol. 96, no. 12, pp. 1179–1193, 2014.

VII. APPENDIX

In this appendix, we provide, for each application, how it was compiled to use the desired SIMD instruction. We also provide the command line that we used for our experiments. If input files are required, we describe how we obtained them (this is especially the case for STORM). Section III-A provides some settings (sur as matrix size for HPL).

A. HPL

HPL is compiled against MKL which allows choosing the right SIMD instruction through the environment variable `MKL_ENABLE_INSTRUCTIONS`. It can be set to `SSE4_2`, `AVX`, `AVX2` and `AVX512` (for skylake only).

HPL uses a configuration file (the values that we changed are described in Table IIIa and is simply launched with `mpirun`: `mpirun -n $nb_cores xhpl`

B. SVD

SVD is also compiled against MKL. Thus setting the desired SIMD instruction is done the same way as HPL.

The command line that we used is the following:
`./time_dgesvd_tile --threads=$cores --n_range=$N:$N:1 --nb=$NB --nowarmup --nodyn --nocheck` where \$cores is the number of cores that we want to use, and \$N and \$NB are described in Table IIIb.

C. STORM data generation and execution command line

STORM can be compiled using the provided Makefile with the desired SIMD compilation flags (`msse4.2`, `maxx2` and `maxx512bw`). We chose to run the storm-nucleotide program (and not the storm-color). The command line that we used to run the program is:
`./storm-nucleotide-sse42-x-gcc -g data/Homo_sapiens.GRCh38.dna_rm.chromosome.22.fa -r data/test_1000000.fq -N $nb_cores -i 15 -z 180 -t 200 -o /dev/null`

where `-g` takes a genome file. We downloaded the Homo_sapiens file from http://bioinfo.lille.inria.fr/yass/data/Homo_sapiens/GRCh38.dna_rm.chromosome.22.fa

`-r` is the reads file to map against the genome. In order to generate the `test_1000000.fq` file, we used the ncbi toolkit (<https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>). It comes with three separate tools (NGC, NCBI-VDB and SRA Toolkit). The tool provides a command `fastq_dump`. It generates the `n` first reads (`-X` option) from an input file (SRR7764388) and writes it to output (`-Z`). SRR7764388 is a file available at the ncbi database (<https://www.ncbi.nlm.nih.gov/sra/SRR7764388>).

Note that we used 1000000 on nova and chifflet and 1000000 on skylake.

D. AFF3CT command line

For AFF3CT, we used the compilation guideline provided in the documentation. Just like STORM, compiling with the desired SIMD instructions is done through compilation flag. Launching the execution is done with `./bin/aff3ct --sim-type BFER --sim-cde-type TURBO -m 1.0 -M 1.0 -K 6144 --dec-type TURBO --dec-implement FAST --dec-sub-simd INTER -i 6 --itl-type LTE --dec-sf-type LTE_VEC --dec-sub-max MAX`