EAR contact: julita.corbalan@bsc.es, luigi.brochard@gmail.com, ckim1@lenovo.com

EAR benefits in energy evaluation and optimization

For EAR evaluation we have used three applications: SATURNE, GROMACS and POP.

SATURNE is an open source fluid dynamics code developed by EDF (www.code-saturne.org). GROMACS is an open source molecular dynamics code originally developed by Groeningen university (www.gromacs.org). POP is the open source Parallel Ocean Model version 2 developed by Los Alamos National Lab (www.cesm.educar.edu). GROMACS and POP have been compiled from sources. These three applications have been executed on Lennox, a Lenovo cluster located in Stuttgart. The Infiniband network has been used and each node has the following characteristics:

Linux Version: CentOS Linux release 7.6.1810 (Core)

Kernel Version: 3.10.0-957.27.2.el7.x86_64

Memory: 96296MB total

CPU Type: Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz

- 2.2GHz maximum frequency for AVX512 instructions
- 2.6GHx maximum frequency for AVX2 instructions

Figure 1 Compute node main characteristics

For each of the applications we executed it with fixed frequencies and with min_time_to_solution power policy. EAR library reports performance and power metrics helping to understand energy variations and power policy decisions. Intel MPI has been used in all the experiments (EAR also support OpenMPI) and icc compiler for GROMACS and POP compilations.

After an application has been executed with EAR, getting its performance and energy metrics is easy with the following ear command (cf. Figure 2).

SATURNE baseline evaluation

SATURNE has been executed on 8 nodes, 320 processes. Figure 2 shows the basic metrics describing SATURNE characteristics having an impact on frequency selection (the *application signature*).

[xjcorbalan@login2302]$ **eacct -j 99383.10 -l**

JOB ID-STEP ID	NODE ID	USER ID	APP ID	**FREQ**	**TIME**	**POWER**	**GBS**	**CPI**	**ENERGY**
99383-10	cmp2531	xjcorbalan	cs_02	2.01	289.85	309.26	36.88	0.77	89640.71
99383-10	cmp2532	xjcorbalan	cs_02	2.02	289.85	304.75	36.84	0.77	88331.02
99383-10	cmp2533	xjcorbalan	cs_02	2.02	289.85	299.57	36.86	0.76	86831.13
99383-10	cmp2534	xjcorbalan	cs_02	2.02	289.85	309.77	36.84	0.77	89787.11
99383-10	cmp2535	xjcorbalan	cs_02	2.02	289.85	291.89	36.88	0.77	84603.28
99383-10	cmp2536	xjcorbalan	cs_02	2.02	289.85	307.57	36.82	0.78	89149.13
99383-10	cmp2543	xjcorbalan	cs_02	2.02	289.85	294.13	36.89	0.78	85255.36
99383-10	cmp2544	xjcorbalan	cs_02	2.02	289.85	283.95	36.83	0.76	82303.12

Figure 2 Application signature + accounting information reported by EAR commands

Figure 3 shows SATURNE characterization. Application signature includes some other metrics but two main characteristics describing application behavior and influencing frequency selection are Cycles per Instructions (CPI), describing CPU utilization intensity, and Transactions per Instruction (TPI), describing memory utilization intensity. In the graphs we will present GBs rather than TPI since it is simple to compare with other profiling tools. The graph shows SATURNE reports CPI and GBs for three test cases cs_01, cs_02 and cs_03 which simulate a small, medium and large problem. We see that in all three cases, CPI is < 1 which indicate a kind of CPU intensive application getting more CPU intensive as the problem size increases. For cases 2 and 4, we note a medium high GBs value increasing with the problem size (STREAM reports more than 100GBs in these nodes). From these metrics, we can deduct that SATURNE is both medium CPU and memory intensive, characteristic of an application doing a good use of the data cache.

Since use case cs_01 was consuming less than 100 seconds, we decided to focus on cs_02 and cs_04 for the following experiments.

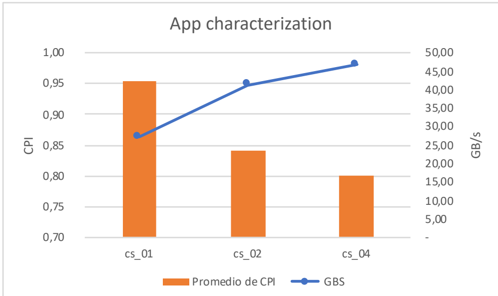

Figure 3 SATURNE characterisatión

Figure 4 show the Time variation, Power variation and Energy savings for two SATURNE use cases when statically varying the CPU frequency. Given cs_04 was consuming a lot of time and presented similar characteristics than cs_02, we selected cs_02 for policy evaluation (execution time was near to 300 seconds).

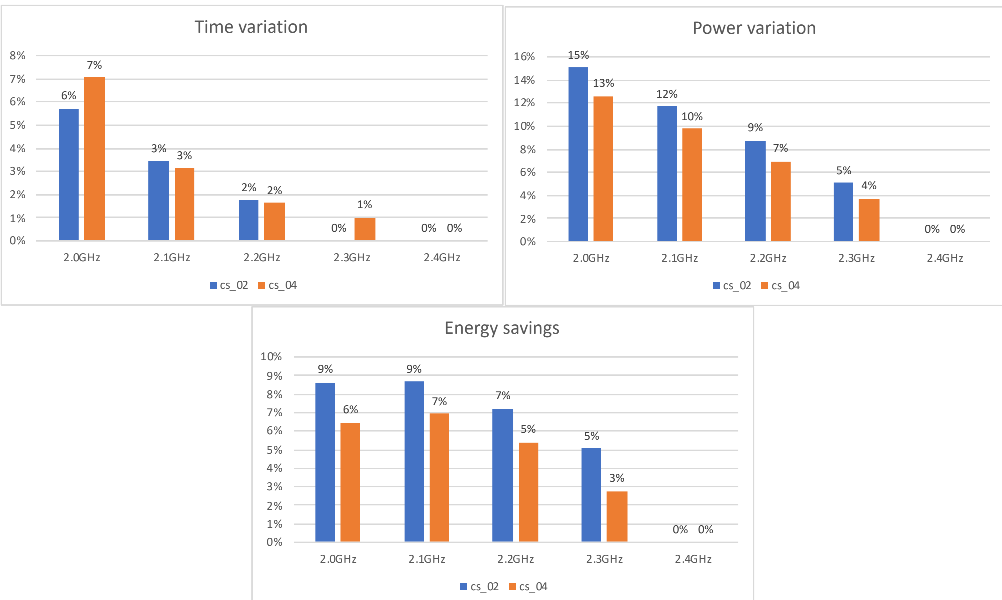

Figure 4 Time, power and energy variation when statically changing the CPU frequency

Based on these results, we can see SATURNE has a maximum potential of up to 7%-9% in energy savings and the optimal case is reached when it is executed at 2.1Ghz since lower CPU frequencies reports significant time variations.

It should be noted that these experiments are not needed when EAR is used with an energy policy, since the application monitoring and frequency selection are 100% automatic and transparent. We do these manual experiments only to compare with what EAR will select automatically.

Next set of experiments include execution of SATURNE with energy policies. For simplicity we have evaluated only one energy policy: min_time_to_solution.

Min_time_to_solution_description

Min_time_to_solution policy starts application at a default frequency F specified by the sysadmin and typically lower than maximum. On Lennox default frequency was set to F_{def}=2.0Ghz whereas the nominal frequency is F_1=2.4 GHz. Once the application signature is computed (at runtime) for one loop iteration, EAR estimates iteration time at F_{def-1},F_{def-2}…F_1. Min_time policy computes, the frequency variation (F_n/F_{n+1}) vs the time variation ($T(F_n)/T(F_{n+1})$). If the Time_variation >= Frequency_variation x policy_threshold (0.7 on Lennox) , then the next pstate (next frequency) is considered. Min_time_to_solution iterates through the possible frequencies until Time_variation is smaller than the criteria meaning the the application does not scale anymore with frequency as requested by the threshold.

Figure 5 shows results EAR decisions and results for SATURNE when executed with EAR and min_time_to_solution as energy policy.

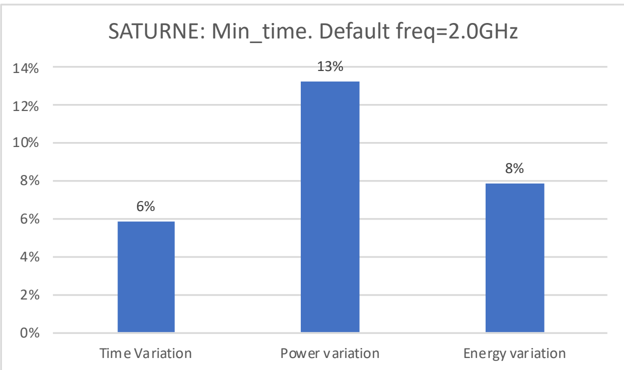

Figure 5 SATURNE results executed with min_time_to_solution

With this policy, EAR selected 2.1 GHz as optimal frequency leading to 6% time variation, 13% power variation and 8% energy savings. Comparing these results with Figure 4, we see that EAR made the right decision.

GROMACS and POP

GROMACS is a package to perform molecular dynamics. The lignocellulose-rf.tpr input case has been used with 16 nodes and 640 mpi processes.

POP is an application for ocean circulation model. It has been executed with 384 mpi processes (10 nodes). GROMACS is a very interesting use case since it does and intensive use of AVX512 instructions. These instructions report very good ratios of Gflops/Watt. They are also complicated to manage since AVX512 instructions have their own frequency dynamically decided by the processor itself based on its characteristics and the power, temperature and number of cores used. Such applications usually run at frequencies lower than nominal when all cores aure used.

Table 1 shows CPI, GBs and GFlops/Watt average values measured during the all execution time of GROMACS and POP. Table 2 shows the same metrics computed only for the main iteration loop that EAR detected.

Table 1 GROMACS and POP characterization

	GBs	CPI	Gflops/Watt
GROMACS	9,04	0,66	**2,98**
POP	63,90	1,51	0,08

It is important to remark how they different compared with global metrics. This is especially relevant and enforces the idea metrics must be collected at relevant parts of the application and at runtime as EAR is doing.

Table 2 Per loop metrics for POP and GROMACS

	GBs	CPI	Iteration time
GROMACS	~11	0,57	14 sec.
POP	~100	1,9-2,3	20 sec.

On Table 1, we see that GROMACS and POP have a very different behavior, with GROMACS being a CPU intensive application (CPI = 0.57) with very little memory use (GBS ~11) while POP is memory bound (GBs ~100) with no CPU intensity (CPI 2). The Gflops/Watt vaue is reflecting this difference with GF/W ~ 3 for GROMACS and ~ 0.10 for POP.

Figure 6 presents the time, power and energy variation when running GROMACS and POP at different CPU frequencies. GROMACS is a CPU intensive application using AVX512 instructions. Usually a CPU intensive application leads to little energy savings. But as AVX512 instructions are power hungry we note that GROMACS has a significant energy savings potential (~10%) at all frequencies but with very different power and time variations. As POP is a memory intensive application, we see potential energy savings of 15% with very low impact on performance.

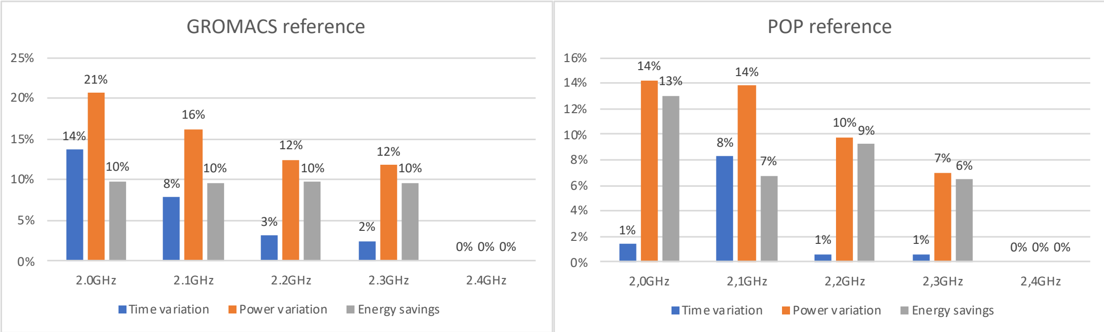

Figure 6 Time, power and energy variation for GROMACS and POP when changing the CPU frequency

Figure 7 presents the time, power and energy variations when running EAR with min_time_to_solution policy on Lennox.

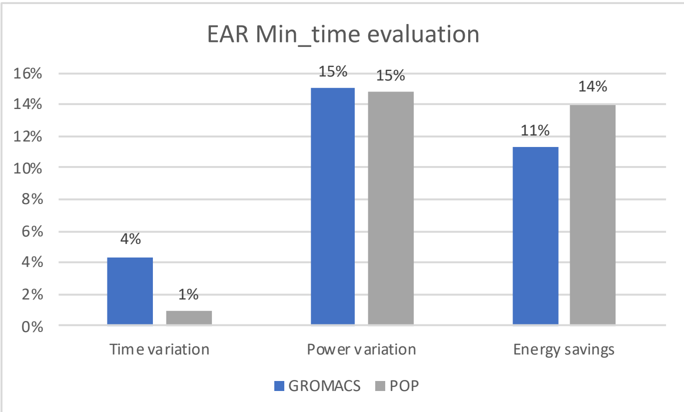

Figure 7 GROMACS and POP evaluation when using EAR and Min_time_to_solution as power policy

As we can see, EAR has been able to identify the optimal frequency for both GROMACS and POP to save the maximum energy with lowest impact on performance according to the min_time_to_solution policy and its threshold criteria. For GROMACS, EAR selected the frequency 2.2GHz and for POP EAR selected 2.0GHz which is in agreement with the results shown in Figure 6.