

Energy consumption optimization of the Total-FETI solver and BLAS routines by changing the CPU

Extended abstract

David Horak, Lubomir Riha, Radim Sojka, Jakub Kruzik, Martin Beseda
 IT4Innovations National Supercomputing Center
 VSB-Technical University of Ostrava
 17. listopadu 15, Ostrava 70833 Czech Republic
 Email: david.horak@vsb.cz

Abstract — The energy consumption of supercomputers is one of the critical problems for the upcoming Exascale supercomputing era. The awareness of power and energy consumption is required on both software and hardware side. This poster deals with the energy consumption evaluation of the Finite Element Tearing and Interconnect (FETI) based solvers of linear systems, which is an established method for solving real-world engineering problems, and with the energy consumption evaluation of the BLAS routines. The experiments performed in the poster deal with CPU frequency.

This work is performed in the scope of the READEX project (Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing).

The measurements were performed on the Intel Xeon E5-2680 (Intel Haswell micro-architecture) based Taurus system installed at TU Dresden. The system contains over 1400 nodes that have an FPGA-based power instrumentation called HDEEM (High Definition Energy Efficiency Monitoring), that allows for fine-grained and more accurate power and energy measurements. The measurements can be accessed through the HDEEM library, allowing developers to take energy measurements before and after the region of interest.

We have evaluated the effect of the CPU frequency on the energy consumption of the FETI solver for a linear elasticity 3D cube synthetic benchmark. On the dualized problem $MPF\lambda=MPd$, we have evaluated the effect of frequency tuning on the energy consumption of the essential processing kernels of the FETI method. There are two main phases in FETI – preprocessing and solve. In preprocessing it is necessary to regularize the stiffness matrix K and factorize it and to assemble the G and GG^T matrices and the second one to factorize. Both operations belong to the most time and also energy consuming operations. The solve employs the Preconditioned Conjugate

Gradient (PCG) algorithm, which consists of sparse matrix-vector multiplications (by F , P , M_L , M_D matrices) and vector dot products and AXPY functions. In each iteration, we need to apply the direct solver twice, i.e., for forward and backward solves for the pseudoinverse K^+ action and for the coarse problem solution, the $(GG^T)^{-1}$ action. The multiplication by the dense Schur complement matrix adds an additional operator with different computational characteristics, potentially increasing the exploitable dynamism.

The poster provides results for two types of frequency tuning: (1) static tuning and (2) dynamic tuning. For static tuning experiments, the frequency is set before execution and kept constant during the runtime. For dynamic tuning, the frequency is changed during the program execution to adapt the system to the actual needs of the application. The poster shows that static tuning brings up 11.84% energy savings when compared to default CPU settings (the highest clock rate). The dynamic tuning improves this further by up to 2.68%. In total, the approach presented in this paper shows the potential to save up to 14.52% of energy for FETI based solvers, see Table 1.

Another energy consumption evaluations were done with selected Sparse and Dense BLAS Level 1, 2 and 3 routines. For benchmarking we have used a set of matrices from University Florida collection. We have employed AXPY, Sparse Matrix-Vector, Sparse Matrix-Matrix, Dense Matrix-Vector, Dense Matrix-Matrix and Sparse Matrix-Dense Matrix multiplication routines from Intel Math Kernel Library (MKL). The measured characteristics illustrate the different energy consumption of BLAS routines, as some operations are memory-bounded and others are compute-bounded. Based on our recommendations one can explore dynamic frequency switching to achieve significant energy savings up to 23%, for more details see Table 2.

Action	Run-time	Static Tuning Savings	Overall Static Savings	Dynamic Tuning Savings	Optimal Frequency	Overall Dynamic Savings
$M_L v$	10%	3.2%	0.32%	3.1%	2.1 GHz	0.31%
Fv with M_L	80%	12%	9.6%	1%	1.6 GHz	0.8%
$M_D v$	40%	13%	5.2%	4%	1.2 GHz	1.6%
Fv with M_D	50%	12%	6%	1%	1.6 GHz	0.5%
Pv	6%	2%	0.12%	1%	2.3 GHz	0.06%
AXPY	4%	13%	0.52%	6%	1.2 GHz	0.52%
Total savings:						
with M_L			10.56%			1.69%
with M_D			11.84%			2.68%

Table 1: EFFICIENCY OF STATIC AND DYNAMIC TUNING FOR THE FETI SOLVE STAGE AND BOTH PRECONDITIONERS

Action	AXPY	SpMV CSR	SpMV COO	SpMM CSR	SpMM COO	DMV FLOAT
Opt.freq. [GHz]	1.2	1.3-1.7	1.3-1.8	1.8-2.4	2.1	1.2
Savings [%]	19	11-20	8-17	5-8	2.1	21-23
	DMV DOUBLE	DMM FLOAT	DMM DOUBLE	SpMDM CSR-DOUBLE	SpMDM COO-DOUBLE	
	1.2	1.7-1.9	1.8-2.5	1.2-1.7	1.2-1.8	
	17-22	2-14	0-10	12-20	13-20	

Table 2: EFFICIENCY OF FREQUENCY TUNING FOR BLAS 1,2,3 ROUTINES