Performance Prediction of Stencil Applications based on Machine Learning

Víctor Martínez and Philippe Navaux
Informatics Institute (INF),
Federal University of Rio Grande do Sul (UFRGS)
Porto Alegre, Brazil
{victor.martinez, navaux}@inf.ufrgs.br

Abstract

Stencil computations are the basis to solve many problems related to Partial Differential Equations (PDEs). Performance prediction for such numerical kernels is a major issue as many critical parameters (architectural features, compiler flags, memory policies, multithreading strategies) are involved. In this context, fast and accurate seismic processing workflow is a critical example for this kind of computations. In order to understand complex geological structures, the numerical kernels used mainly arise from the discretization of Partial Differential Equations (PDEs) and High Performance Computing methods play a major in seismic modeling. This paper focuses on the use of Machine Learning to predict the performance of stencil kernels on multi-core and many-core architectures. Low-level hardware counters (e.g. cache-misses and TLB misses) on a limited number of executions are used to build our predictive model. We have considered three different kernels (7-point Jacobi, acoustic and seismic wave modeling) to demonstrate the effectiveness of our approach. Our results show that performance can be predicted by simulations of hardware counters with high accuracy.

1. Introduction

Stencil computations lie at the heart of many problems in areas as diverse as electromagnetics, fluid dynamics or geophysics. The trend for High Performance Computing (HPC) applications is to pay a higher cost in order to optimize the overall performance. This comes from the complexity of many interdependent factors (non-uniform memory access, vectorization, compiler optimizations, memory policies) at an architectural level that may severely influence the application's behavior.

On the one hand, wave propagation modeling is the current backbone for several seismic tools. It has been extensively applied for imaging potential oil and gas reservoirs beneath salt domes for the last five years. Such acoustic propagation engines should be continuously ported to the newest HPC hardware available to maintain competitiveness. At the same time, on the HPC hardware front, the days of faster single core CPUs are over, and the solutions adopted are being replaced by many-core technologies [5, 4]. In this context, several heuristics or frameworks have been proposed to speed up the process of finding the best configuration for stencil applications [3, 14, 17, 12].

On the other hand, Machine Learning (ML) is a comprehensive methodology for optimization that could be applied to find patterns on a large set of input parameters. Recently, ML algorithms have been used on HPC systems under different situations. In [19] the authors used ML algorithms to select the best job scheduling algorithm on heterogeneous platforms whereas in [2] the authors proposed an ML-based scheme to select the best I/O scheduling algorithm for different applications and input parameters. And recently, in [13] the authors used ML algorithms to predict the performance of stencil computations on multicore architectures.

In this paper, we describe the general procedure to build a suitable ML-based performance model for classical numerical kernels: 7-point Jacobi and acoustic and seismic wave modeling. This model allows us to simulate the performance behavior of stencil computations on multi-core architectures. The paper is organized as follows. Section 2 provides the fundamentals of stencils under study. Section 3 describes the methodology of our ML-based approach. Section 4 presents experiment configuration and model accuracy. Section 5 describes related works, and finally Section 6 concludes this paper.

2. Stencil models

From the numerical analysis point of view, stencil-based computations often arise when discretizing Partial Differential Equations (PDEs). For instance, the Finite-Difference Methods (FDMs) computational procedure consists in us-

ing the neighboring points in the north-south, east-west and forward-backward directions to evaluate the current grid point in the case of a three-dimensional Cartesian grid. The algorithm then moves to the next point applying the same stencil computation until the entire spatial grid has been traversed. In this work, we study two well-known stencil kernels:

1. **7-point Jacobi**: The seven-point Jacobi stencil is a reference example of numerical kernel used in various context in order to evaluate the impact of advanced reformulation or the impact of the underlying architecture. A review can be found in [7] for instance.
2. **Seismic Wave**: Evaluation of damages occurred during strong ground motion is critical for urban planning. The numerical kernel under study relies on the classical 4-th order in space and second-order in time approximation and was detailed in [18, 15, 10].
3. **Acoustic wave**: We consider the isotropic acoustic wave propagation under Dirichlet boundary conditions over a finite 3D rectangular domain. The operator is discretized by a 12^{th} order finite differences approximation and the time derivatives are approximated by a 2^{nd} order finite differences operator. This kernel represents the cornerstone of the classical Reverse Time Migration imaging procedure. For sake of clarity of this paper focused on the impact of machine learning methodology, we will not go into too many details regarding the implementation and interesting readers can refer to [1].

3. Machine Learning Methodology

In this section we describe our ML methodology which relies on support vector machines (SVM). First, we present the feature vectors considered in our study. Finally, we describe our ML model.

3.1. Feature vectors

We considered three sets of vectors, which are described below:

1. **Input Vector**: We considered different parameters available in OpenMP as vector input, such as the number of threads, the loop scheduling policy (static or dynamic), and the chunk size (which defines how many loop iterations will be assigned to each thread at a time).
2. **Hardware Counters Vector:** We used the PAPI library to collect information from hardware counters. We considered the following metrics as the most relevant ones: Last level cache misses, L3 and L2 total cache misses for multicore and manycore respectively (**PAPI_L3_TCM, PAPI_L2_TCM events**), data translation lookaside buffer misses (**PAPI_TLB_DM event**), and total cycles (**PAPI_TOT_CYC event**). For acoustic wave model we use only two hardware counters.
3. **Performance Vector:** The output vector, which uses billions execution time as performance characterization metric.

3.2. Machine Learning Model

The proposed ML model is based on SVM, which is a supervised ML approach introduced in [6] and extended to regression problems where support vectors are represented by kernel functions [9]. Our ML model was built on top of three consecutive layers, where output values of a layer are used as input values of the next layer (Figure 1). The input layer contains the configuration values from the input vector. The hidden layer contains the SVMs that take values from the input vector to simulate the behavior of hardware counters. Finally, the output layer contains takes each simulated value from the hidden layer to obtain the corresponding GFLOPS and execution time values.

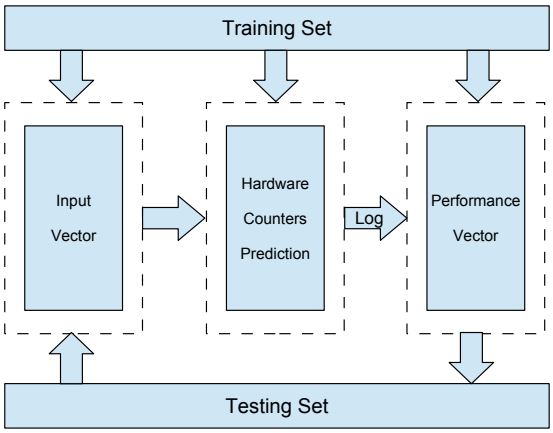

Figure 1: Flowchart of the proposed ML-based model.

4. Experiments

In this section we describe our experimental testbed and present the data analysis and results.

4.1. Experimental Testbed

We used two multi-core and one many-core platforms to carry out the experiments. Their hardware configurations are shown in Table 1.

	Multicore 1	**Multicore 2**	**Manycore**
Processor	Xeon E5-2650	Xeon E5-4650	Xeon Phi 7520
Clock(GHz)	2.0	2.7	1.40
Cores	8	8	68
Sockets	2	4	1
Threads	16	32	272
Last level cache size (MB)	20 (L3)	20 (L3)	34 (L2)

Table 1: Experimental testbed configurations.

4.1.1. Training and validation sets We created a training set by randomly selecting a subset from the configuration set presented in Table 2 and details all the configurations available for our optimization categories. As it can be noted, a brute force approach would be unfeasible.

	Parameters	**Multicore 1**	**Multicore 2**	**Manycore**
Number of threads	1	8	12	272
Scheduling policy	1	2	2	2
Chunk size	1	32	32	272
Total of configurations	3	512	768	147,968

Table 2: Configurations available for our optimization procedure

A random testing set was used since all SVMs in both the hidden and the output layers are trained to calculate new GFLOPS and execution time values through simulation. Table 3 presents the total number of experiments that were performed to obtain the training and validation sets.

		Multicore 1	**Multicore 2**	**Manycore**
7-point	Training	44	38	-
	Testing	11	10	-
	Total	55	48	-
Seismic	Training	211	237	-
	Testing	53	60	-
	Total	264	297	-
Acoustic	*Training*	-	-	808
	Testing	-	-	203
	Total	-	-	1,011

Table 3: Number of experiments in the training and the testing sets.

4.2. Results

We use the validation set to evaluate the model with two statistical estimators: root mean square error (RMSE) and the coefficient of determination (R-square). The former represents the standard deviation of the differences between predicted values and real values whereas the latter represents how close the regression approximates the real data (R-square equal to 1 indicates a perfect fit of data regression).

As it can be noted in Table 4, the RMSE value confirms that deviation of time value is high, but the approximation of R-square is close to 99%, then we get a highly accurate regression.

		Multicore 1	**Multicore 2**	**Manycore**
7-point	*RMSE*	0.66	2.51	-
	R-Square	0.98	0.86	-
Seismic	*RMSE*	13.53	212.28	-
	R-Square	0.99	0.62	-
Acoustic	*RMSE*	-	-	154.04
	R-Square	-	-	0.94

Table 4: Estimators for predicted values of the numerical kernels.

5. Related Works

In [1], the authors focused on acoustic wave propagation equations, choosing the optimization techniques from systematically tuning the algorithm. The usage of collaborative thread blocking, cache blocking, register re-use, vectorization and loop redistribution.

Other works investigated the accuracy of regression models and ML algorithms in different contexts. In [16] the authors compared ML algorithms for characterizing the shared L2 cache behavior of programs on multi-core processors. The results showed that regression models trained on a given L2 cache architecture are reasonably transferable to other L2 cache architectures.

Finally, in [11] the authors applied ML techniques to explore stencil configurations (code transformations, compiler flags, architectural features and optimization parameters). Their approach is able to select a suitable configuration that gives the best execution time and energy consumption. In [8], the authors improved performance of stencil computations by using a model based on cache misses. In [13], the authors proposed a ML model to predict performance of stencil computations on multicore architectures.

6. Conclusion

In this paper, we introduced a general predictive performance modeling strategy for geophysical numerical kernel on multi and many-core architectures. We showed that performance of the common numerical kernels can be predicted with a high accuracy (99%) based on hardware counters and Machine Learning.

Firstly, we expect to extend our methodology in order to capture complex behaviors (vectorization capabilities, data mapping). Secondly, we intend to design a model based on unsupervised ML algorithms to further improve our results.

Finally, we believe that a general model can be integrated into an auto-tuning framework to find the best performance configuration for a given stencil kernel.

7. Acknowledgments

For computer time, this research partly used the resources of Colfax Research. This work has been granted by CAPES, CNPq, FAPERGS and *PETROBRAS* company. The authors thank Jairo Panetta from Aeronautics Institute Of Technology (ITA) for providing the acoustic wave numerical kernel code. It was also supported by Intel Corporation under the Modern Code Project. Research has received funding from the EU H2020 Programme and from MCTI/RNP-Brazil under the HPC4E Project, grant agreement n° 689772.

References

[1] C. Andreolli, P. Thierry, L. Borges, G. Skinner, and C. Yount. Chapter 23 - Characterization and Optimization Methodology Applied to Stencil Computations. In J. Reinders and J. Jeffers, editors, *High Performance Parallelism Pearls*, pages 377 – 396. Morgan Kaufmann, Boston, 2015.

[2] F. Z. Boito, R. V. Kassick, P. O. A. Navaux, and Y. Denneulin. Automatic I/O scheduling algorithm selection for parallel file systems. *Concurrency and Computation: Practice and Experience*, 28(8):2457–2472, 2016. cpe.3606.

[3] M. Christen, O. Schenk, and H. Burkhart. Automatic code generation and tuning for stencil kernels on modern shared memory architectures. *Comput. Sci.*, 26(3-4):205–210, June 2011.

[4] R. G. Clapp. Seismic Processing and the Computer Revolution(s). In *SEG Technical Program Expanded Abstracts 2015*, pages 4832–4837, 2015.

[5] R. G. Clapp, H. Fu, and O. Lindtjorn. Selecting the right hardware for reverse time migration. *The Leading Edge*, 29(1):48–58, 2010.

[6] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[7] K. Datta, S. W. Williams, V. Volkov, J. Carter, L. Oliker, J. Shalf, and K. Yelick. *Scientific Computing with Multicore and Accelerators*, chapter Auto-Tuning Stencil Computations on Multicore and Accelerators. CRC Press, Taylor & Francis Group, 2010.

[8] R. de la Cruz and M. Araya-Polo. *Modeling Stencil Computations on Modern HPC Architectures*, pages 149–171. Springer International Publishing, Cham, 2015.

[9] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems 9*, pages 155–161. MIT Press, 1997.

[10] F. Dupros, H. Do, and H. Aochi. On scalability issues of the elastodynamics equations on multicore platforms. In *Proceedings of the International Conference on Computational Science, ICCS 2013, Barcelona, Spain, 5-7 June, 2013*, pages 1226–1234, 2013.

[11] A. S. Ganapathi. *Predicting and Optimizing System Utilization and Performance via Statistical Machine Learning*. PhD thesis, EECS Department, University of California, Berkeley, Dec 2009.

[12] V. Martinez, F. Dupros, M. Castro, H. Aochi, and P. O. A. Navaux. Stencil-based applications tuning for multi-core. In *Latin American High Performance Computing Conference (CARLA 2016)*, pages 1–15, Sep 2016. Oral presentation.

[13] V. Martínez, F. Dupros, M. Castro, and P. Navaux. Performance improvement of stencil computations for multi-core architectures based on machine learning. *Procedia Computer Science*, 108:305 – 314, 2017. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.

[14] R. Mijakovic, M. Firbach, and M. Gerndt. An architecture for flexible auto-tuning: The periscope tuning framework 2.0. In *International Conference on Green High Performance Computing (ICGHPC)*, pages 1–9, Feb 2016.

[15] P. Moczo, J. Robertsson, and L. Eisner. The finite-difference time-domain method for modeling of seismic wave propagation. In *Advances in Wave Propagation in Heterogeneous Media*, volume 48 of *Advances in Geophysics*, chapter 8, pages 421–516. Elsevier - Academic Press, 2007.

[16] J. K. Rai, A. Negi, R. Wankar, and K. D. Nayak. On prediction accuracy of machine learning algorithms for characterizing shared l2 cache behavior of programs on multicore processors. In *Computational Intelligence, Communication Systems and Networks, 2009. CICSYN '09. First International Conference on*, pages 213–219, July 2009.

[17] Y. Tang, R. A. Chowdhury, B. C. Kuszmaul, C.-K. Luk, and C. E. Leiserson. The pochoir stencil compiler. In *ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '11, pages 117–128, New York, NY, USA, 2011. ACM.

[18] J. Virieux. P-SV wave propagation in heterogeneous media; velocity-stress finite-difference method. *Geophysics*, 51(4):889–901, 1986.

[19] D. Vladuic, A. Cernivec, and B. Slivnik. Improving job scheduling in grid environments with use of simple machine learning methods. In *International Conference on Information Technology: New Generations*, pages 177–182, April 2009.