SANDIA REPORT
SAND2011-7049
Unlimited Release
Printed September 2011

LDRD Final Report: Autotuning for Scalable Linear Algebra

Michael A. Heroux
Bryan Marker

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2011-7049
Unlimited Release
Printed September 2011

LDRD Final Report: Autotuning for Scalable Linear Algebra

Michael A. Heroux
Scalable Algorithms Department
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1320

Bryan Marker
Department of Computer Science
The University of Texas at Austin
Austin, TX 78712

Abstract

This report summarizes the progress made as part of a one year lab-directed research and development (LDRD) project to fund the research efforts of Bryan Marker at the University of Texas at Austin. The goal of the project was to develop new techniques for automatically tuning the performance of dense linear algebra kernels. These kernels often represent the majority of computational time in an application. The primary outcome from this work is a demonstration of the value of model driven engineering as an approach to accurately predict and study performance trade-offs for dense linear algebra computations.

ACKNOWLEDGMENTS

The authors thank the Sandia LDRD program that funded this work.

CONTENTS

FIGURES

1. INTRODUCTION

Dense linear algebra kernels such as matrix-vector and matrix-matrix multiplications represent a large portion of the computational time in many applications, so optimal performance is very important. Modern computer architectures present a challenge to custom optimization techniques since there are many variations in architecture design and minor changes in programming strategy can have a tremendous impact on performance. As a result, automatic tuning for performance is becoming increasingly important.

The goal of this research was to develop a way to automatically parallelize and optimize dense linear algebra (DLA) algorithms and codes. We initially focused on distributed-memory implementations but future work will target sequential and shared-memory implementations as well. The goal is to mimic the process an expert goes through since this is often very mechanical and regular. In the past year, we have laid the foundation for two major aspects of this research: First, we have studied distributed-memory DLA algorithms and software and the efforts of experts to optimize such software. Second, we have used this study to develop a prototype system to automatically generate optimized code just like an expert does.

2. MODEL DRIVEN ENGINEERING FOR AUTOTUNING

Parallelizing and optimizing dense linear algebra (DLA) algorithms for distributed memory machines has historically been done by domain experts who are very familiar with both linear algebra and the oddities of a target class of machines. When a DLA expert has no experience with a new architecture and wants to implement an algorithm, he must live with an existing library, learn a lot about that architecture, or find an experienced developer. This is inefficient and unnecessary because the work of an expert can be very mechanical and systematic, and therefore automated.

In this project we applied Model Driven Engineering (MDE), which fosters the codification of fundamental algorithms and domain-specific expertise, to this domain. MDE enables automation of the activities of experts: selecting algorithms, composing algorithms, and applying optimizations to achieve customized and high-performance implementations in code. In this project, we showed how expert-tuned, high-performance code for a Cholesky factorization for distributed memory architectures can be automatically produced by a tool. Furthermore, we showed how layering code in a way that is amenable to mechanical transformations not only makes a library more maintainable but also writable by other software. Since Cholesky factorization is a prototypical example of a broad class of dense linear algebra operations and communication on a distributed memory architecture is an example of data movement between memory layers, we believe our approach can be extended to target other algorithms and other architectures (such as multi-core processors, GPGPUs, and many-core processors) to create the DLA libraries of the future. Since DLA library development has often introduced new software engineering techniques to the broader scientific computing community, our approach may influence the broader scientific computing programmer community.

3. SUMMARY OF EFFORTS

For the first part of the past academic year we studied how to apply software engineering principles and methods to DLA. The most interesting outcome from this is how Model Driven Engineering (MDE) can be successfully used to model DLA algorithms. MDE enables us to encode how an expert parallelizes DLA operations for various architectures such as distributed-memory computers. Furthermore, we can encode how experts optimize DLA codes as patterns of "bad" codes to be transformed into "good" codes. Essentially, we can modularize the efforts of experts similar to how code is modularized in functions.

With this method for encoding and modularizing expert knowledge of DLA algorithms and codes, we developed a prototype system to explore how this knowledge could be applied. This system takes as input a MDE-encoded DLA algorithm. It then applies encoded knowledge of how to parallelize and optimize code fragments to generate a search space of hundred to tens of thousands of implementations codes. For each implementation, the system generates a simple cost function to take into consideration communication, memory, and computation costs. These cost functions are then used to choose which implementations are "best," i.e., the implementations that are likely the fastest. Our approach has been successfully tested in this prototype for a number of prototypical DLA algorithms. Figure 1 (taken from [1]) shows results for Cholesky decomposition. The system is able to generate the same or better implementations as an expert for these algorithms. A first paper [1] has been submitted to PPoPP'12, a top conference in the field.

This project has allowed us to make significant progress. We now have a proven way to encode DLA algorithms and codes as well as the transformations an expert applies to parallelize and optimize codes. Furthermore, this encoding enabled construction of a prototype system to explore the possibility to automate the parallelization of DLA codes. So far we have attained good success with this approach. Future work includes exploring how this approach can be applied to optimize sequential and share-memory codes.

In addition to the paper mentioned in [1], articles [2], [3] and [4] acknowledge support from this project.

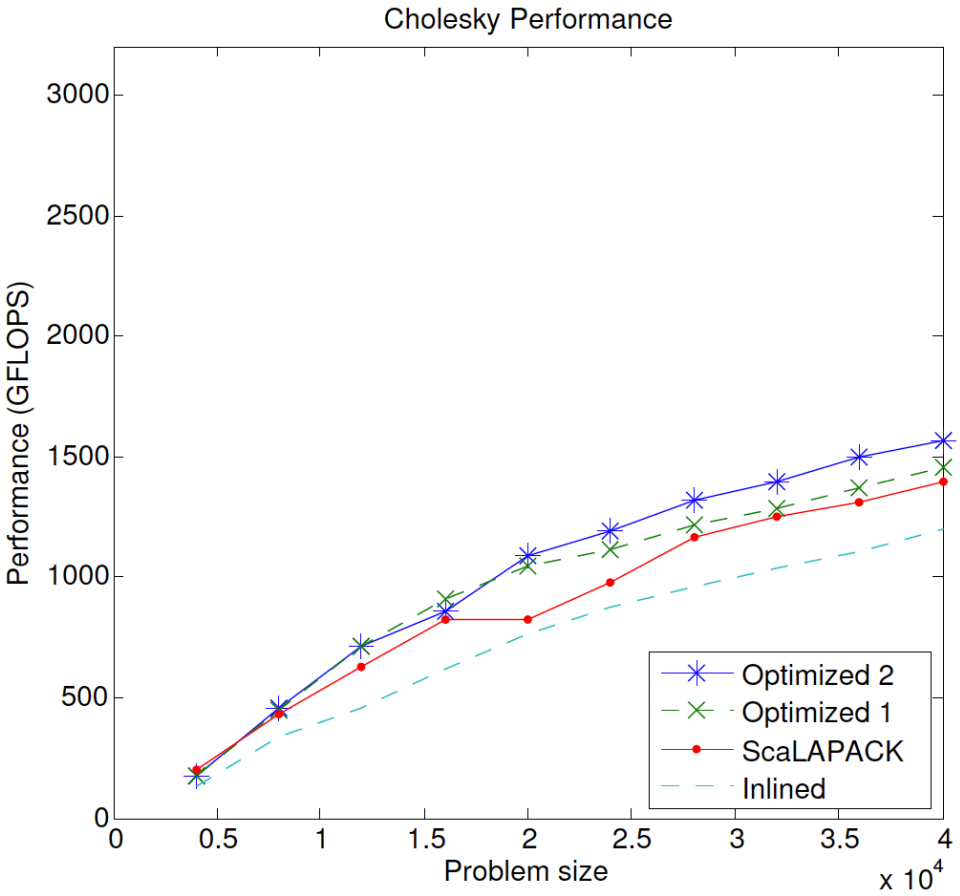

Figure 1: *Performance of Cholesky decomposition on 20 node system with 2 Intel six-core processors per node for a total of 240 cores. Results compare the initial code (Inlined) vs. ScaLAPACK (production library) and two autotuned versions generated by our approach. Results show that our automatically generated optimized versions outperform the expert-coded ScaLAPACK version.*

3. CONCLUSIONS

We have shown that Model Driven Engineering can be successfully applied to automating performance optimization of kernels in the problem domain of dense linear algebra on distributed memory systems. The results show competitive and even superior performance over a human expert in the design of optimal computational methods.

We expect the insights from this project to have a significant impact on the FLAME project. This project encompasses a formalism for deriving DLA algorithms, notation for expressing these as algorithms, and APIs for implementation in code. Two library instantiations of FLAME exist: the libflame library that targets sequential, multicore, and (multi-)GPU architectures, and Elemental, which targets distributed memory architectures. The proposed approach would allow us to instead support a single encoding of algorithms and knowledge, with libraries like libflame and Elemental being the products (outputs) of applying our methodology. Because of the breadth of potential impact, we expect this project to be the first of many to explore this area research.

4. REFERENCES

1. Bryan Marker, Andy Terrel, Jack Poulson, Don Batory, and Robert van de Geijn. *Mechanizing the Expert Dense Linear Algebra Developer, FLAME Working Note #58,* The University of Texas at Austin, Department of Computer Science. Technical Report TR-11-18, April 2011. (Submitted to PPoPP'12.)
2. Bryan Marker, Ernie Chan, Jack Poulson, Robert van de Geijn, Rob F. Van der Wijngaart, Timothy G. Mattson, and Theodore E. Kubaska. *Programming Many-Core Architectures - A Case Study: Dense Matrix Computations on the Intel SCC Processor,* Concurrency and Computation: Practice and Experience. To Appear.
3. Jack Poulson, Bryan Marker, Jeff R. Hammond, Nichols A. Romero, and Robert van de Geijn. *Elemental: A New Framework for Distributed Memory Dense Matrix Computations,* ACM Transactions on Mathematical Software. Submitted.
4. Taylor L. Riché, Don Batory, Rui Gonçalves, Bryan Marker. *Architecture Design by Transformation, FLAME Working Note #54,* The University of Texas at Austin, Department of Computer Science. Technical Report TR-10-39. Dec. 14, 2010.

DISTRIBUTION

1	MS0899	Technical Library	9536 (electronic copy)
1	MS0359	D. Chavez, LDRD Office	1911


Sandia National Laboratories