Final Project Report

DOE Award: SC0016260
Sponsoring Program Office: Advanced Scientific Computing Research
Project Title: Using Verified Lifting to Optimize Legacy Stencil Codes
Principal Investigator: Alvin Cheung
Report Period: 7/15/16-7/14/19

Executive Summary

This project investigated new techniques for compiling stencil and stencil-like computations. Stencils computations are commonly found in applications such as image processing, physical simulations, image processing, and machine learning. In recent years, many high-performance domain-specific languages (DSLs) have been proposed to optimize stencil computations. To leverage such DSLs, however, existing codes often need to be rewritten. Such rewriting is manual, labor intensive, and error-prone.

To alleviate such issues, this project investigated the application of program synthesis and artificial learning techniques to enable stencil computations to automatically leverage new high-performance DSLs. Rather than constructing syntax driven rules, verified lifting uses program synthesis to search for a target code fragment to compile the given input code into. In addition, it also searches for a proof that validates how the found target code fragment preserves the semantics of the original input. Thus, the target code fragment is guaranteed to be semantically equivalent to the input.

Project Activities

Over the course of this project, we have constructed a new compiler called STNG based on verified lifting to automatically lift stencil computations embedded in real-world applications. We designed a new high-level specification language, based on the theory of arrays, to express stencil computations. STNG takes in stencil codes written in FORTRAN, lifts such codes to our specification language using program synthesis, and compiles the resulting specification to Halide, a high-performance DSL for GPU computation. Using verified lifting, we have also implemented Casper, a new compiler that enables sequential Java programs to leverage large-scale data parallel processing frameworks based on the map reduce programming model, such as Spark and Hadoop. We have used Casper to speed up real-world image processing benchmarks and have demonstrated up to an order of magnitude improvement in program execution time after compiling the input to Spark, and our compiler also supports generating code to

other map reduce backends as well, including Hadoop and Flink with similar performance improvement.

In the context of Casper, we investigated new cost models that can be used to both speed up the lifting process and choose among different implementations when multiple semantically equivalent versions are available. Unlike typical cost models used in compilers that focus on aspects such as generated code size or complexity of the instructions used, our cost model instead focuses on estimating the amount of data that will be transferred among different nodes, as data transfer costs often dominate execution time for data-intensive code. We have applied this new cost model in Casper and have demonstrated its ability in efficiently pruning the search space during lifting and generating resulting programs that are performant when deployed on a cluster of machines.

Our results on STNG was published in PLDI 2016 and was presented by Shoaib Kamil, PI Cheung's collaborator at Adobe. Casper was published in the SYNT workshop in 2016 and we received the best student paper award for the work. Maaz Ahmad, PI Cheung's student at the University of Washington, presented the work at the workshop. The work also subsequently appeared as a demo paper in SIGMOD 2017 and as a full paper in SIGMOD 2018.

Given the previous instances of compilers that have been constructed using verified lifting, we then decided to build a framework that allows compiler developers to leverage the technique themselves. The target users of the framework are those who implement compilers for domain-specific languages (DSLs), which are prevalent across many disciplines. Such languages let developers easily express computations using high-level abstractions and execute the resulting applications in a highly efficient manner. To leverage DSLs, however, application developers need to learn the syntax of the language and manually rewrite existing code. To address these issues, we are currently building MetaLift, a new compiler generator for transforming general-purpose code into DSLs. MetaLift comes with a high-level language that DSL designers can use to specify the semantics of their language. It uses this specification to generate a compiler that leverages program synthesis techniques to automatically rewrite input code into the target DSL. The framework is currently under joint-development with Adobe.

On the other hand, we also focused on automatically translating image processing kernels written in C to Halide, a DSL for image processing operations. In particular, we developed a new algorithm to both speed up the search for target Halide codes and also scale up the size of input kernels that can be processed. Our new algorithm breaks down code search into three stages: given an input kernel, we first synthesize the range of points that are processed by the it (by ignoring the actual computation performed in the

input code), then we determine the traversal order of the processed points (still ignoring the computation performed), and finally synthesize the computation given the outputs from the first two stages. We subsequently implemented our algorithm in a new compiler called Dexter. The work was published in SIGGRAPH ASIA 2019, with Dexter-compiled code now shipping with Adobe Photoshop since 2020.

Project Publications and Outreach Activities

Funding from this project has led to the following publications:

- Maaz Bin Safeer Ahmad, Alvin Cheung. Leveraging Parallel Data Processing Frameworks with Verified Lifting, SYNT 2016.

- Shoaib Kamil, Alvin Cheung, Shachar Itzhaky, Armando Solar-Lezama. Verified Lifting of Stencil Computations, PLDI 2016.

- Alvin Cheung, Maaz Bin Safeer Ahmad. Optimizing Data-Intensive Applications Automatically By Leveraging Parallel Data Processing Frameworks. Proceedings of the SIGMOD 2017 conference.

- Alvin Cheung, Maaz Bin Safeer Ahmad. Automatically Leveraging MapReduce Frameworks for Data-Intensive Applications. Proceedings of the SIGMOD 2018 conference.

- Maaz Bin Safeer Ahmad, Jonathan Ragan-Kelley, Alvin Cheung, Shoaib Kamil: Automatically translating image processing libraries to halide. ACM Transactions on Graphics 38(6): 204:1-204:13 (2019).

Along with the following outreach activities:

- PI Cheung's collaborator, Shoaib Kamil from Adobe, presented the STNG work in PLDI 2016 to researchers and practitioners.
- Maaz Ahmad, PI Cheung's student, presented the Casper work in the SYNT workshop in 2016 to researchers and practitioners. In addition, Maaz has also presented the work at the Northwest Programming Languages Research Day at the University of Washington, and at the research retreats of the Programming Languages and Database research groups in the department.
- PI Cheung has presented verified lifting in invited talks at the Database research groups at Stanford University and the University of California, Berkeley, Intel Research, and Huawei Research in 2016.

- Maaz Ahmad presented a demo of Casper at the SIGMOD 2017 conference.
- PI Cheung's collaborator, Shoaib Kamil from Adobe, presented the STNG work at a talk in the Seventh International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing (WOLFHPC), held during the Supercomputing conference in 2017.
- Maaz Ahmad, PI Cheung's student, completed an internship under the guidance of Dr Shoaib Kamil, senior research scientist at Adobe Research, in the summer of 2017.
- Maaz recently presented the Casper compiler annual Architecture, Systems, Programming Languages at the University of Washington retreat in 2017.
- Maaz Ahmad will present the Casper work at the upcoming SIGMOD conference in June.
- PI Cheung has presented verified lifting in invited talks at Microsoft Research, Redmond; the Pacific Northwest National Laboratory (hosted by Dr Sriram Krishnamoorthy), the University of California, Berkeley; and Stanford University in 2018.
- PI Cheung has also given an invited talk on verified lifting at the StrangeLoop developers' conference in 2017.
- The Casper compiler has been released to the public and has been used in PI Cheung's class to teach students about parallel and distributed computing.
- Maaz Ahmad presented the Casper system (a compiler for translating sequential Java code into Spark using verified lifting) at the SIGMOD conference in 2018.
- Maaz Ahmad completed a summer internship at Intel in 2018, with the goal to deploy his prior work on verified lifting within the company.
- PI Cheung has presented verified lifting in invited talks at the University of California, Berkeley and Stanford University in 2018, and Intel Research in early 2019.
- Maaz Ahmad presented the Dexter compiler at the SIGGRAPH ASIA 2019 conference.

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0016260.