Automatic Context-sensitive Input-aware Acceleration of Sparse Applications

Abstract

Applications with huge sparse data structures have emerged as a new challenge for high-performance computing, and become increasingly important in big-data analytics, computational biology, web search, and knowledge discovery, etc. Unlike the traditional dense linear algebra, these applications' performance is bound by memory bandwidth, and their parallelism is dependent on specific input data. The most widely following approach to speed them up is by manual optiomizations with "ninja" experts, which is extremely time-consuming.

This paper proposes a co-designed compiler and library approach to automatically analyze and optimize sparse applications to achieve close-to-ninja performance. A compiler automatically analyzes an application, and passes context information to a library so that a library function can adjust its behavior optimally accross multiple loop iterations and multiple matrices, and collaborate with multiple functions, even though they are still uncoupled in design and implementation.

We shows how to automatically reorder data. While there are many reordering algorithms, it has been an open question for more than 5 decades how to correctly insert the minimum amount of reordering and reverse reordering into an arbitrary piece of code. We prove this is an NP-complete problem, and propose a heuritisc algorithm to solve it. We show the key concepts of distributivity and inter-dependence, and their analyses.

With ..., performance ...

1. Introduction
2. A Motivating Example
3. Reordering
4. Context-sensitive Acceleration
5. Experiments
6. Related Works
7. Acknowledgements

References