Procedia Computer Science

Volume 80, 2016, Pages 178–189

ICCS 2016. The International Conference on Computational Science

A Performance Prediction and Analysis Integrated Framework for SpMV on GPUs

Ping Guo and Chung-wei Lee

Department of Computer Science, University of Illinois at Springfield, Springfield, Illinois, USA
Email: {pguo6, clee84}@uis.edu

Abstract

This paper presents unique modeling algorithms of performance prediction for sparse matrix-vector multiplication on GPUs. Based on the algorithms, we develop a framework that is able to predict SpMV kernel performance and to analyze the reported prediction results. We make the following contributions: (1) We provide theoretical basis for the generation of benchmark matrices according to the hardware features of a given specific GPU. (2) Given a sparse matrix, we propose a quantitative method to collect some features representing its matrix settings. (3) We propose four performance modeling algorithms to accurately predict kernel performance for SpMV computing using CSR, ELL, COO, and HYB SpMV kernels. We evaluate the accuracy of our framework with 8 widely-used sparse matrices (totally 32 test cases) on NVIDIA Tesla K80 GPU. In our experiments, the average performance differences between the predicted and measured SpMV kernel execution times for CSR, ELL, COO, and HYB SpMV kernels are 5.1%, 5.3%, 1.7%, and 6.1%, respectively.

Keywords: GPU, performance modeling, sparse matrix-vector multiplication

1 Introduction

Sparse matrix-vector multiplication (SpMV) is an essential operation in solving linear systems and partial differential equations. For many scientific and engineering applications, the matrices are naturally large and sparse with various sparsity characteristics. It is a challenging issue to accurately predict SpMV performance. This paper addresses this challenge by presenting performance modeling algorithms to predict SpMV performance on GPUs.

A sparse matrix is a matrix in which most of the elements are zeros and a few other elements are non-zeros. Bell and Garland [1] proposed and implemented some widely-used formats to store non-zero elements in a sparse matrix, including CSR (Compressed Sparse Row), ELL (ELLPACK), COO (Coordinate), and HYB (Hybrid). They also designed and implemented CUDA-based SpMV computational kernels to perform SpMV operations with each sparse matrix format. From our experiments, we observed that different matrices may have their own most appropriate storage formats to achieve the best performance. This observation motivates

Selection and peer-review under responsibility of the Scientific Programme Committee of ICCS 2016

doi:10.1016/j.procs.2016.05.308

us to design a framework to provide performance prediction for SpMV computing using multiple SpMV kernels. The prediction results reported by our framework can be used to effectively assist researchers in foreseeing SpMV performance before some specific programs are actually to be run. In addition, our framework can further assist researchers in making choice of appropriate matrix storage formats and test matrices. Specifically, given a target sparse matrix and a specific GPU architecture, the most appropriate matrix storage format among CSR, ELL, COO, and HYB formats can be selected by comparing the predicted kernel performance of four SpMV computational kernels which are proposed and implemented for these four sparse matrix storage formats. In another aspect, given a specific GPU architecture, researchers can select an appropriate small set of test matrices for experiments from a large collection of spare matrices by evaluating the predicted performance of SpMV computing using specific kernels.

We solve the performance approximation problem by offline benchmarking. This approach has wide adaptability for sparse matrices with different sparsity characteristics and it supports any NVIDIA GPU platform. For each GPU platform supported, we only need to generate benchmarks for that platform once. The approach to generate benchmark matrices is easy to follow. In addition, our platform-based benchmarking approach has competive advantage in accuracy of performance prediction compared with traditional analytical modeling approach without using benchmarks.

Our performance modeling approach consists of 5 steps in 2 stages, *i.e.*, benchmark generation, performance measurement, relationship establishment, matrix analysis, and performance prediction. In the offline stage, a series of unique benchmark matrices are generated and an SpMV computation is performed by generating a random vector for each benchmark matrix. Some features representing the matrix settings and performance measured for SpMV computations are collected for establishing some linear relationships for performance prediction. In the online stage, the features of a given target sparse matrix are collected as inputs to instantiate our parameterized relationships for performance estimation.

We make the following contributions: (1) We provide theoretical basis for the generation of benchmark matrices according to the hardware features of a given specific GPU. (2) Given a sparse matrix, we propose a quantitative method to collect some features representing its matrix settings. (3) We propose four performance modeling algorithms to accurately predict kernel performance for SpMV computing using CSR, ELL, COO, and HYB SpMV kernels.

2 Related Work

Bolz *et al.* [2] first implemented SpMV computing on GPUs. There have been some existing modeling approaches focusing on performance prediction for SpMV on GPUs. Dinkins [5] proposed a model for predicting SpMV performance using memory bandwidth requirements and data locality. Guo and Wang [6] presented a cross-architecture performance modeling tool to accurately predict SpMV performance on multiple different target GPUs based on the measured performance on a given reference GPU. Li *et al.* [9] proposed a modeling approach for SpMV performance estimation by analyzing the distribution characteristics of non-zero elements. The modeling approaches focusing on performance optimization includes [3, 7, 8, 12]. Choi *et al.* [3] designed a blocked ELLPACK (BELLPACK) format and proposed a performance model to predict matrix-dependent tuning parameters. Guo *et al.* [7] designed a dynamic-programming algorithm to optimize SpMV performance based on the prediction results reported by modeling tool. Karakasis *et al.* [8] presented a performance model that can accurately select the most suitable blocking sparse matrix storage format and its proper configuration. Xu *et al.* [12] proposed the optimized SpMV based on ELL format and a performance model.