The co-computing technique and practice for SpMV on CPU-GPU heterogenous platform

Kenli Li
Hunan University
National Supercomputing Center in Changsha
Tel: 13036781568
Email: lkl@hnu.edu.cn
Outline

- Sparse matrix storage formats
- SpMV on GPU
- SpMV on CPU-GPU
Motivation

- **SpMV**
  - The core operation of iterative solving methods for sparse linear systems
  - 70% the amount of calculations
Sparse matrix

$$A = \begin{bmatrix}
3 & 0 & 0 & 0 & 0 \\
0 & 1 & 4 & 0 \\
9 & 0 & 2 & 10 \\
0 & 3 & 0 & 7
\end{bmatrix}$$
Compressed storage formats

- **COO (Coordinate Format)**
  
  \[
  A = \begin{bmatrix}
  3 & 0 & 0 & 0 \\
  0 & 1 & 4 & 0 \\
  9 & 0 & 2 & 10 \\
  0 & 3 & 0 & 7 
  \end{bmatrix}
  \]

  row=[0 1 1 2 2 3 3]

  column=[0 1 2 2 3 1 3]

  value = [3 1 4 2 10 3 7]

- **CSR (Compressed Sparse Row Format)**
  
  \[
  Av = [3 \quad 1 \quad 4 \quad 9 \quad 2 \quad 10 \quad 3 \quad 7]
  \]

  \[
  Aj = [0 \quad 1 \quad 2 \quad 0 \quad 2 \quad 3 \quad 1 \quad 3]
  \]

  \[
  Ap = [0 \quad 1 \quad 3 \quad 6 \quad 8]
  \]

- **ELL (ELLPACK Format)**

  \[
  EData = \begin{bmatrix}
  3 & 0 \\
  1 & 4 \\
  2 & 10 \\
  3 & 7 
  \end{bmatrix}
  \]

  \[
  Offsets = \begin{bmatrix}
  0 & -1 \\
  1 & 2 \\
  2 & 3 \\
  1 & 3 
  \end{bmatrix}
  \]

- **DIA (Diagonal Format)**

  \[
  D = \begin{bmatrix}
  * & 3 & 0 \\
  9 & 2 & 10 \\
  3 & 7 & * 
  \end{bmatrix}
  \]

  Offsets = [−2 0 1]
Challenges

- For irregular sparse characteristic of the sparse matrix, it is difficult to achieve better compression effect using single compressed storage format.

- For SpMV, it is difficult to obtain higher speedup on GPU.
Hybrid storage formats

- For sparse quasi-diagonal matrix, a hybrid of the diagonal format (DLA) and the compressed sparse row format (CSR) (HDC) is presented.
Optimization of quasi-diagonal matrix–vector multiplication on GPU

- Performance comparisons for different implementations of SpMV
Probabilistic Modeling of SpMV

- For various sparsity characteristics of sparse matrices, it is a challenging issue to adopt an appropriate algorithm to implement and optimize SpMV.

- We addressed this challenge by presenting a performance modeling and analysis method to estimate and optimize SpMV performance on GPU using a probabilistic model.
Probability Mass Function

- $A$ is a sparse matrix
- The discrete random variable $X$ represents the number of non-zeros of one row in $A$.
- Define the probability mass function (PMF) of discrete random variable $X$ as $P$

$$P(X = i) = p_i = \frac{b_i}{N}, \ i = 0, 1, 2, \ldots, M,$$  where

(i) $p_i \geq 0, \ i = 0, 1, 2, \ldots, M$;

(ii) $\sum_{i=0}^{M} p_i = \sum_{i=0}^{M} \left(\frac{b_i}{N}\right) = \frac{1}{N} \sum_{i=0}^{M} b_i = \frac{N}{N} = 1$. 
Probability Distributions of some Sparse Matrices
Performance Analysis for SpMV using PMF

The appropriate storage format can be selected according to the distribution pattern of a sparse matrix through the performance analysis using PMF.
Performance Analysis and Optimization for SpMV on GPU using Probabilistic Modeling

- Comparison of the estimated speedup and the tested speedup

IEEE Transactions on Parallel and Distributed System, 2015, 26(1)
The performance optimizing for SpMV using HYB by PMF

If a sparse matrix $A$ is stored in HYB, $A$ should be split into COO and ELL formats. A parameter $K$ should be provided when splitting. The choice of parameter $K$ can affect the performance of SpMV using HYB.

The percentage of speedup improvement using HYB through threshold $K$ optimization.
CPU-GPU Heterogeneous Computing Platform

Motivation

- GPU has been extensively used for resolving a broad range of computationally demanding and complex problems.

- The parallel computing capability of CPU is also rising to an increase in the number of cores in CPU.

- There are 6 supercomputers using CPU-GPU heterogeneous computing platform in top 20 of world.
GPU-CPU Heterogeneous programming

Challenges

- How to make full use of computing resources of GPU and Multicore CPU?
- How to reduce the cost of data transmission between CPU and GPU?
- How to play to their respective advantages computing of CPU and GPU?
GPUs and Multicore CPUs Hybrid Parallel Programming Model
Parallel Implementation of SpMV Based on Partitioning

- **Motivation:** How to make full use of computing resources of GPUs and Multicore CPUs for SpMV
- **Strategy:** Partition into some blocks according to the computing power of CPUs and GPUs and use suitable storage formats for GPUs and CPUs
Optimization Strategies for CPU-GPU Heterogeneous Computing

- Load-balancing between the CPU and GPU
  - Partitioning strategies according to the computing power of CPU and GPU
  - Improve the occupancy of SMs
  - Dynamic task scheduling and work stealing based on asynchronous streams

- Hidden data access latency
  - Data access coalescing
  - On-demand data transfer mode based on UVA
  - Pipeline model based on CPU and GPU
Performance Optimization Using Partitioned SpMV on GPUs and Multicore CPUs

- The flop-rates of BCOO, BCSR, PMF, and PKT on CPU-GPU heterogenous platform.
- F1:BCOO; F2:BCSR; F3:PMF; F4: PKT

IEEE Transaction on Computers, 2015, 64(9)
Future Works

- Solving large-scale sparse linear equations using our SpMV methods on large-scale heterogeneous computing systems which have been developed by our country.
Thanks Any Questions