Exploiting Data Locality for Unified CPU/GPU Memory Space using OpenMP

ABSTRACT

To facilitate GPU programming and accelerate applications with large working set, recent GPUs support unified CPU/GPU memory space addressing. In such systems, CPU and GPU can conveniently address each other's memory and data is moved between different memory on demand. However, our investigation shows that the current data migration mechanism is not aware of data locality, resulting in poor performance and redundant data movement. The upcoming OpenMP 5.0 will include new data locality features to address the complex memory hierarchy in today's systems, however, current proposed features do not take unified memory into consideration and cannot address its performance problem. To solve this problem, we propose to extend OpenMP data locality features to improve unified memory performance. The proposed extension exposes different GPU memory management choices for programmers to exploit GPU data locality explicitly. In scenarios with complex data locality, programmers are also allowed to pass hints so that OpenMP compiler and runtime make better GPU data management decisions. Preliminary evaluation shows that our proposal can significantly improve GPU performance and reduce redundant data movement between CPU and GPU for benchmarks with large working set.

KEYWORDS

GPU, unified virtual memory, compiler optimization

ACM Reference format:
. 2017. Exploiting Data Locality for Unified CPU/GPU Memory Space using OpenMP. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 1 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Since version 4.0, OpenMP has included GPU offloading support, which makes it an attractive option for GPU programming.

In this paper, we focus on discreted GPUs.

Conference'17, July 2017, Washington, DC, USA
2017. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn