論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	System Software Studies for Seamlessly Overcoming GPU Resource Limitations
著者(和文)	MARKTHUBPAK
Author(English)	Pak Markthub
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第10943号, 授与年月日:2018年9月20日, 学位の種別:課程博士, 審査員:松岡 聡,遠藤 敏夫,脇田 建,額田 彰,横田 理央
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第10943号, Conferred date:2018/9/20, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	要約
Type(English)	Outline

（博士課程）
Doctoral Program

論 文 要 約

THESIS OUTLINE

専攻： Department of	数理・計算科学　　専攻	申請学位（専攻分野）： Academic Degree Requested	博士 Doctor of　（　理学　）
学生氏名： Student's Name	MARKTHUB Pak	指導教員（主）： Academic Supervisor(main)	特任教授　松岡　聡
		指導教員（副）： Academic Supervisor(sub)	

要約
Thesis Outline

This dissertation titled "System Software Studies for Seamlessly Overcoming GPU Resource Limitations" provides discussion and analysis to two common problems in heterogeneous high-performance-computing (HPC) systems containing GPUs, and presents system software solutions that are seamless, universal, and efficient to address the problems. It composes of eight chapters.

Chapter 1 gives high-level discussion about two common problems in GPU HPC system: the GPU Memory Capacity Limitation problem and the Scattered Idle GPU problem. In addition, this chapter provides motivation and an overview of our proposed solutions: DRAGON, and mrCUDA and MRQ.

Chapter 2 provides prerequisite background for understanding the rest of this dissertation. It contains a brief review of the GPU and CUDA platform, Unified Memory, GPU page-faulting mechanism, GPU driver architecture, and NVM storage, which are essential to understand the GPU Memory Capacity Limitation problem and our solution called DRAGON. This chapter also includes a summary concerning resource-sharing systems and job scheduling to supplement our discussion regarding the Scattered Idle-GPU problem and our proposed mrCUDA and MRQ solution.

Chapter 3 contains the essence of important previous studies that tried to address those two problems. Discussion regarding their shortcomings and motivate why we need better solutions are also included in this chapter.

Chapter 4 begins the journey to the GPU Memory Capacity Limitation problem. We discuss the problem and our motivation in details. Then we introduced our proposed solution DRAGON: a framework that enables all classes of GPGPU applications to transparently operate on very large datasets residing in Non-Volatile Memory (NVM) storage, while also ensuring the integrity of data buffers. DRAGON leverages GPU page-faulting mechanism and extends capabilities of Unified Memory to provide transparent data access to terabytes of NVM. We explain the architecture and operations of DRAGON along with some optimizations to keep its overhead within 10% compared with using POSIX I/O with Unified Memory (baseline) in this chapter. We also provide the results and discussion with eight benchmark applications and two case studies on Caffe, a popular deep learning framework.

Chapter 5 revisits the design of DRAGON and discuss its imperfection in term of performance. We propose the use of direct page-cache access with two-level prefetching to remove the performance penalty. We further discuss new optimization techniques based on the new design. The evaluation and case study sections are similar to chapter 4 but show how the new design achieve up to 2.3x performance improvement compared with the baseline in chapter 4.

Chapter 6 shifts our focus to the second problem, the Scattered Idle-GPU problem. We explain the problem in details and motivate why we need GPU remoting and migration to solve the problem. We then provide brief discussion about rCUDA and dive deep into the analysis of its overhead. We propose mrCUDA, a remote-to-local GPU migration middleware, to mitigate unnecessary penalty of GPU remoting when more local GPUs become available. We discuss its design and the detailed investigation of mrCUDA's overhead using mathematic and real-world applications.

Chapter 7 directly addresses Scattered Idle-GPU Problem. We provide a minimum procedure for integrating rCUDA and mrCUDA to job scheduling algorithms. We then use the first-come-first-serve (FCFS) scheduling policy to study the effectiveness of our method by comparing with MRQ, the FCFS scheduling algorithm with rCUDA and mrCUDA integrated. By ways of simulation, we show and discuss various scenarios using both synthetic and recorded workloads.

Chapter 8 concludes our journey and all knowledge we have learnt. We also provide some important directions for improving our study.