T2R2 東京科学大学 リサーチリポジトリ
Science Tokyo Research Repository

# 論文 / 著書情報
# Article / Book Information

| | |
|---|---|
| 題目(和文) | GPU上の仮想シストリックアレイ |
| Title(English) | A Virtual Systolic Array on GPUs |
| 著者(和文) | 陳 鵬 |
| Author(English) | Peng Chen |
| 出典(和文) | 学位:博士(学術),<br>学位授与機関:東京工業大学,<br>報告番号:甲第11530号,<br>授与年月日:2020年3月26日,<br>学位の種別:課程博士,<br>審査員:松岡 聡,遠藤 敏夫,額田 彰,脇田 建,横田 理央 |
| Citation(English) | Degree:Doctor (Academic),<br>Conferring organization: Tokyo Institute of Technology,<br>Report number:甲第11530号,<br>Conferred date:2020/3/26,<br>Degree Type:Course doctor,<br>Examiner:,,,, |
| 学位種別(和文) | 博士論文 |
| Category(English) | Doctoral Thesis |
| 種別(和文) | 論文要旨 |
| Type(English) | Summary |

Powered by T2R2 (Science Tokyo Research Repository)

（博士課程）
Doctoral Program

# 論 文 要 旨

THESIS SUMMARY

| | | | |
|---|---|---|---|
| 系・コース：<br>Department of, Graduate major in | 数理・計算科学 系<br>数理・計算科学 コース | 申請学位（専攻分野）：<br>Academic Degree Requested | 博士 （ 学術 ）<br>Doctor of |
| 学生氏名：<br>Student's Name | 陳鵬 | 指導教員（主）：<br>Academic Supervisor(main) | 特任教授 松岡 聡 |
| | | 指導教員（副）：<br>Academic Supervisor(sub) | |

要旨（英文 800 語程度）
Thesis Summary（approx.800 English Words ）


Driven by the critical demands of intensive computation in a variety of fields, an old and heavily-researched parallel computer architecture, namely systolic array, is being revived in the post-Moore's law era, e.g. Google's Tensor Processing Unit (TPU), NVIDIA's Tensor Core. Such a kind of domain-specific processors can provide extremely high TOPS (TeraOps/Second) and TOPS/Watt for compute-intensive algorithms, i.e. convolution, dense matrix multiplication, AI-purposed computation, Discrete Cosine Transform (DCT). In the last decade, we witnessed the emergence of Graphics Processing Units (GPU). As one of the most popular accelerators recently, GPUs are adopted to speed up the computation in a wide range of fields, i.e. scientific, engineering. Taking the latest TOP500 rankings in High-Performance Computing (HPC) for instance, more than half of the FLOPS comes from the NVIDIA Tesla GPUs.

A systolic array is a network of processing elements (PEs) that rhythmically compute, accumulate and transfer data through the system. In typical systolic arrays, all of the PEs are often nested as a two-dimensional mesh. The systolic array architecture is simple yet very effective to achieve both computation and energy efficiencies with very limited memory bandwidth. Inspired by the mechanism of the hard-wired systolic arrays, we innovate a versatile execution model on the top of CUDA architecture for optimizing applications on GPUs. More specifically, we mimic the behavior of systolic arrays by CUDA warp, register files, and shuffle instruction. It is noteworthy that the three key techniques contribute to our model. First, register cache; second, partial sums accumulation and transfer; third, in-register computation. Our model improves the performance of regular memory-bound kernels by taking advantage of thread-private registers as a cache to perform the efficient in-register computation and employing the shuffle intrinsic to exchange partial sums between CUDA threads within a single CUDA Warp. Our model can be viewed as Software Systolic Array Execution Model (SSAM). In other words, we build a virtual systolic array on the top of CUDA architecture.

Regarding most of the scientific applications (or kernels), the increasing flops-to-bytes ratio of GPUs makes their computational performances bounded by the memory bandwidth. Memory-bound kernels that have a regular pattern of computation are particularly challenging since they appear to be simple, yet they require very complex data reuse schemes to effectively utilize the CUDA memory hierarchy, e.g. global memory, shared memory, register memory, etc. Typically, advanced GPU implementations for memory-bound kernels on structured grids rely on the optimized use of fast on-chip scratchpad memory: the programmer uses this user-managed scratchpad memory for reducing the global memory access. Indeed, there exists a plethora of work proposing variations and combinations of the three locality schemes that rely on scratchpad memory: spatial blocking, temporal blocking, and a wavefront pipeline. Those complex locality schemes enabled strides in performance improvements. However, they essentially moved the bottleneck from the global memory to the faster, yet smaller, scratchpad. The objective of this study is to yet again move the bottleneck from the scratchpad to a faster resource: register files. Hence, we prioritize to use register files to cache data rather than shared memory, resulting in better data locality and higher computational efficiency.

A wide class of memory-bound kernels (or applications) can benefit from this execution model. In this study, we focus on mapping typical memory-bound kernels in SSAM, i.e. 2D convolution, 2D/3D stencils, Summed Area Tables (SAT). There is a strong motivation to optimize these kernels

on GPUs. The 2D convolution and SAT computation become increasingly important in the Deep Learning workload. The stencil is a fundamental computation pattern in most of the scientific applications. To further improve the dependency graph in SSAM, we propose a novel algorithm to transpose the register cache locally (termed as BRLT). Unlike 2D convolution and stencils, we propose a collection of SSAM-based algorithms for SAT computation by the BRLT algorithm. To insight the performance advances of SSAM-based algorithms, we build performance models to analyze their mechanisms while using micro-benchmarking to measure some constant variables of GPUs.

The contributions in this study are as follows. First, we design and formulate a software systolic array on the top of CUDA architecture for speeding up the computation of memory-bound kernels with regular access on GPUs. Second, we analyze the data reuse and redundancy schemes to quantify the efficiency and limitations of SSAM. Third, we evaluate the proposed model for a wide variety of 2D/3D stencils, 2D general convolution, and SAT kernels on the latest NVIDIA Tesla GPUs and demonstrate that SSAM-based algorithms outperform the top reported state-of-the-art libraries, e.g. NPP, ArrayFire, OpenCV. Fourth, we propose a novel algorithm to transpose the register matrix (used as register cache) locally and concurrently, resulting in improving the program dependency graph in SSAM for better locality and higher parallelism. Finally, we pave a novel way for code automation on GPUs due to the versatility and simplicity of SSAM for optimizing a large variant of algorithms.

備考 ： 論文要旨は、和文 2000 字と英文 300 語を 1 部ずつ提出するか、もしくは英文 800 語を 1 部提出してください。
Note : Thesis Summary should be submitted in either a copy of 2000 Japanese Characters and 300 Words (English) or 1copy of 800 Words (English).

注意：論文要旨は、東工大リサーチリポジトリ(T2R2)にてインターネット公表されますので、公表可能な範囲の内容で作成してください。
Attention: Thesis Summary will be published on Tokyo Tech Research Repository Website (T2R2).