The Name of the Title Is Hope

Lars Thørväld
The Thørväld Group
Hekla, Iceland
larst@affiliation.org

Valerie Béranger
Inria Paris-Rocquencourt
Rocquencourt, France

Aparna Patel
Rajiv Gandhi University
Doimukh, Arunachal Pradesh, India

ABSTRACT

As the compute capabilities of major graphics APIs have matured, there has been growing utilization of compute shaders for graphical tasks, and as such, there has been corresponding interest in porting over primitives like prefix scan from compute languages to shading languages. However, the current state-of-the-art prefix scan, *Chained Scan with Decoupled Lookback*, relies on properties provided by the NVIDIA ecosystem that, once removed, force developers to make trade-offs between maintainability, portability, and performance. We describe *DecoupledFallback*, a fully portable, single-pass prefix scan method capable of reaching speed of light scan performance across different hardwares with minimal developer intervention. Our implementation is built in WGSL and is compatible with any implementation of the WebGPU standard (I think? It definitely does not work on WARP . . . which is concerning).

CCS CONCEPTS

• **Do Not Use This Code → Generate the Correct Terms for Your Paper**; *Generate the Correct Terms for Your Paper*; Generate the Correct Terms for Your Paper; Generate the Correct Terms for Your Paper.

KEYWORDS

Do, Not, Us, This, Code, Put, the, Correct, Terms, for, Your, Paper

ACM Reference Format:
Lars Thørväld, Valerie Béranger, and Aparna Patel. 2018. The Name of the Title Is Hope. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Prefix scan is one of the most fundamental primitives in parallel computing, with uses including compaction, sorting [1, 19], SPMV [] etc.. Sometimes referred to as *scan* or *prefix reduction*, prefix scan is typically defined on a monoid, though it can also be defined on a semigroup in certain cases[7]. In a prefix scan, the result at element n is the reduction of the preceding subset of elements in the sequence. If the reduction subset includes the n-th element, it is called *inclusive*; if it excludes the n-th element, it is called *exclusive*. The binary operator used in the scan must be associative, but it need not be commutative—for instance, in the case of the stack monoid. The most common type of prefix scan is prefix sum, where the binary operator is addition. For example[1]:

(PREFIX SUM EXAMPLE WITH BINARY REDUCTION OPERATOR)

Contemporary GPU's are characterized by their high memory bandwidth, high arithmetic throughput, high memory latency, and hierarchical memory model. With proper latency-hiding, prefix scan is computationally light enough that it is memory-bandwidth bound, and thus the focus of contemporary scan strategies has been global communication avoidance. The current state-of-the-art prefix scan is *Chained Scan with Decoupled Lookback*[10] (referred to hereon as *DecoupledLookback*). Although a chained scan[2] may seem inimical to parallelism, the key innovation of GPU chained scanning lies in its hybridization of parallel and serial scan strategies at different levels of the GPU memory hierarchy, achieving parallelism at the intra-workgroup level while minimizing global data movement through serial scan operations at the inter-workgroup level. *DecoupledLookback* leverages this hybrid strategy while also being capable of fully saturating global memory bandwidth, and as a result, with approximately ~$2n$ global data movement—one read and one write per processed element—it matches the performance of a copy operation, effectively achieving "speed of light" efficiency.

Although *DecoupledLookback* achieves near-ideal performance on NVIDIA hardware and implemented in CUDA, it relies on a set of architectural and language-specific guarantees that no longer hold once outside of the NVIDIA ecosystem: forward progress guarantees (FPG), fixed subgroup sizes, explicit divergence handling, and memory fences. Without forward progress guarantees, the algorithm risks deadlock, forcing either a regression to the slower *Reduce-then-Scan* approach or maintenance of multiple scan variants. Without fixed subgroup sizes, developers must again regress to slower scan implementations without subgroup acceleration or maintain multiple scan variants. Ambiguous divergence behavior can produce subtle correctness or performance issues, and without explicit masks, developers must rely on compiler and vendor-specific heuristics, increasing the risk of unpredictable behavior and further complicating portability efforts. Absent memory fences, as is the case in the D3D12 API, control over memory ordering is significantly diminished and can result in unbounded redundant work during the scan.

While previous work has focused on improving the performance of prefix scan, the contribution of this work is portability. This work presents *Chained Scan with Decoupled Lookback and Decoupled*

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

[1] In this paper we will use the terminology of WGSL, but we note that subgroup is interchangeable with warp(CUDA), wave(HLSL), and simd_group(Metal), while workgroup is interchangeable with block(CUDA), group(HLSL), and ???(Metal)
[2] A chained scan is a scan performed serially.

Fallback, a fully portable prefix scan capable of reaching speed of light performance without FPG, and implemented in the WGSL shading language. On an intra-workgroup level, we contribute a subgroup-size-agnostic scan pattern, which enables a single scan pattern to be used for all possible subgroup sizes. On an inter-workgroup-level, we contribute *DecoupledFallback*, an extension of the *DecoupledLookback* technique that is no longer dependent on FPG to execute correctly. Our implementation is guided by the following goals:

- **Portability: The implementation must execute correctly on all hardwares and backend graphics API's supported by the WebGPU standard.** Our scan should execute correctly on any hardware and API, regardless of the underlying subgroup size, scheduling model, or divergence behavior.
- **Performance: The implementation must achieve speed of light performance whenever possible.** Our scan should achieve speed of light performance whenever possible, without relying on vendor-specific adaptations.
- **Maintainability: The implementation must use a single variant for all hardwares and must be minimally compiled.** Creating statically specialized per-vendor shader variants increases both maintenance and compilation times. Our goal is to make our implementation as lightweight to maintain and compile as possible.
- **Ease of use: Using the implementation should require as little developer intervention as possible.** Developers should not have to be conscious of, or make decisions based on, their target hardware when using our scan.

2 BACKGROUND

The study of prefix scan patterns can be traced to the design of adder circuits and beyond[9, 20]. Indeed, at their most granular level, contemporary GPU scan patterns still utilize classical parallel prefix adders like Hillis-Steele/Kogge-Stone[6, 8], Brent-Kung[2], or Sklansky[16].

2.0.1 *Evolution of Intra-Workgroup Scan Strategies.*

2.1 Evolution of Inter-Workgroup Scan Strategies

In a prefix scan, the reduction at each element is dependent on the reduction of preceding elements. Thus a serial inter-workgroup dependency is created when the size of the scan exceeds the capacity of a single workgroup.

2.1.1 *4n: Scan-Then-Propagate.* [5, 14, 15]

2.1.2 *3n: Reduce-Then-Scan.* [3, 4, 11]

2.1.3 *2n: Single-Pass Scan.* [10, 21]

2.2 Portability

[12, 13, 17, 18]:

- **Forward Progress Guarantee:** Beginning with the Volta architecture, NVIDIA formalized forward-progress-guarantees (FPG) at both the workgroup and subgroup level. Prior to Volta, FPG was likely already present at a workgroup level.
- **Fixed Subgroup Size:** On NVIDIA hardware, the subgroup width is fixed at 32, enabling static optimizations at that specific width. Furthermore, the square of the subgroup width is CUDA's maximum workgroup size, enabling straightforward implementation of workgroup reductions.
- **Unambiguous Divergence Behavior:** In CUDA, developers can explicitly provision subgroup functions with a mask of participating threads, unambiguously defining the function's behavior irrespective of potential divergence.
- **Memory Fence:** CUDA offers developers memory fence functions to enforce sequentially consistent atomic memory accesses, enabling more efficient inter-workgroup synchronization.

3 SUBGROUP SIZE AGNOSTIC PATTERN

4 CHAINED SCAN WITH DECOUPLED LOOKBACK AND DECOUPLED FALLBACK

ACKNOWLEDGMENTS

REFERENCES

[1] Andy Adinets and Duane Merrill. 2022. Onesweep: A Faster Least Significant Digit Radix Sort for GPUs. arXiv:2206.01784 [cs.DC] https://arxiv.org/abs/2206.01784

[2] Brent and Kung. 1982. A Regular Layout for Parallel Adders. *IEEE Trans. Comput.* C-31, 3 (1982), 260–264. https://doi.org/10.1109/TC.1982.1675982

[3] Siddhartha Chatterjee, Guy E. Blelloch, and Marco Zagha. 1990. Scan primitives for vector computers *(Supercomputing '90)*. IEEE Computer Society Press, Washington, DC, USA, 666–675.

[4] Yuri Dotsenko, Naga K. Govindaraju, Peter-Pike Sloan, Charles Boyd, and John Manferdelli. 2008. Fast scan algorithms on graphics processors. In *Proceedings of the 22nd Annual International Conference on Supercomputing* (Island of Kos, Greece) *(ICS '08)*. Association for Computing Machinery, New York, NY, USA, 205–213. https://doi.org/10.1145/1375527.1375559

[5] Mark Harris, Shubhabrata Sengupta, and John Owens. 2007. *Parallel prefix sum (scan) with CUDA.* Addison Wesley, Chapter 39.

[6] W. Daniel Hillis and Guy L. Steele. 1986. Data parallel algorithms. *Commun. ACM* 29, 12 (Dec. 1986), 1170–1183. https://doi.org/10.1145/7902.7903

[7] Ralf Hinze. 2004. An Algebra of Scans. In *Mathematics of Program Construction*, Dexter Kozen (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 186–210.

[8] Peter M. Kogge and Harold S. Stone. 1973. A Parallel Algorithm for the Efficient Solution of a General Class of Recurrence Equations. *IEEE Trans. Comput.* C-22, 8 (1973), 786–793. https://doi.org/10.1109/TC.1973.5009159

[9] Richard E. Ladner and Michael J. Fischer. 1980. Parallel Prefix Computation. *J. ACM* 27, 4 (Oct. 1980), 831–838. https://doi.org/10.1145/322217.322232

[10] Duane Merrill and Michael Garland. 2016. *Single-pass Parallel Prefix Scan with Decoupled Look-back.* Technical Report NVR-2016-002. NVIDIA Corporation. https://research.nvidia.com/publication/2016-03_single-pass-parallel-prefix-scan-decoupled-look-back

[11] Duane Merrill and Andrew Grimshaw. 2009. *Parallel Scan for Stream Architectures.* Technical Report. University of Virginia. https://doi.org/10.18130/V3XN3C Accessed: 2024-12-09.

[12] NVIDIA Corporation. 2017. *NVIDIA Volta Architecture Whitepaper.* Technical Report. NVIDIA Corporation. https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

[13] NVIDIA Corporation. 2024. *CUDA C Programming Guide.* https://docs.nvidia.com/cuda/cuda-c-programming-guide/ Accessed: 2024-12-06.

[14] Shubhabrata Sengupta, Mark Harris, and Michael Garland. 2008. *Efficient Parallel Scan Algorithms for GPUs.* Technical Report NVR-2008-003. NVIDIA Corporation. https://research.nvidia.com/publication/2008-12_efficient-parallel-scan-algorithms-gpus

[15] Shubhabrata Sengupta, Mark Harris, Yao Zhang, and John D. Owens. 2007. Scan primitives for GPU computing. In *Proceedings of the 22nd ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware* (San Diego, California) *(GH '07)*. Eurographics Association, Goslar, DEU, 97–106.

[16] J. Sklansky. 1960. Conditional-Sum Addition Logic. *IRE Transactions on Electronic Computers* EC-9, 2 (1960), 226–231. https://doi.org/10.1109/TEC.1960.5219822

[17] Tyler Sorensen, Hugues Evrard, and Alastair F. Donaldson. 2018. GPU Schedulers: How Fair Is Fair Enough?. In *29th International Conference on Concurrency Theory (CONCUR 2018) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 118)*, Sven Schewe and Lijun Zhang (Eds.). Schloss Dagstuhl

– Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 23:1–23:17. https://doi.org/10.4230/LIPIcs.CONCUR.2018.23

[18] Tyler Sorensen, Lucas F. Salvador, Harmit Raval, Hugues Evrard, John Wickerson, Margaret Martonosi, and Alastair F. Donaldson. 2021. Specifying and testing GPU workgroup progress models. *Proc. ACM Program. Lang.* 5, OOPSLA, Article 131 (Oct. 2021), 30 pages. https://doi.org/10.1145/3485508

[19] Elias Stehle and Hans-Arno Jacobsen. 2017. A Memory Bandwidth-Efficient Hybrid Radix Sort on GPUs. In *Proceedings of the 2017 ACM International Conference on Management of Data* (Chicago, Illinois, USA) *(SIGMOD '17)*. Association for Computing Machinery, New York, NY, USA, 417–432. https://doi.org/10.1145/3035918.3064043

[20] A. Weinberger and J. L. Smith. 1956. A One-Microsecond Adder Using One-Megacycle Circuitry. *IRE Transactions on Electronic Computers* EC-5, 2 (1956), 65–73. https://doi.org/10.1109/TEC.1956.5219801

[21] Shengen Yan, Guoping Long, and Yunquan Zhang. 2013. StreamScan: fast scan algorithms for GPUs without global barrier synchronization. In *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (Shenzhen, China) *(PPoPP '13)*. Association for Computing Machinery, New York, NY, USA, 229–238. https://doi.org/10.1145/2442516.2442539

A RESEARCH METHODS

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009