Preliminary work on radix-based multi-select on GPUs

I. Komarov, R. D'Souza

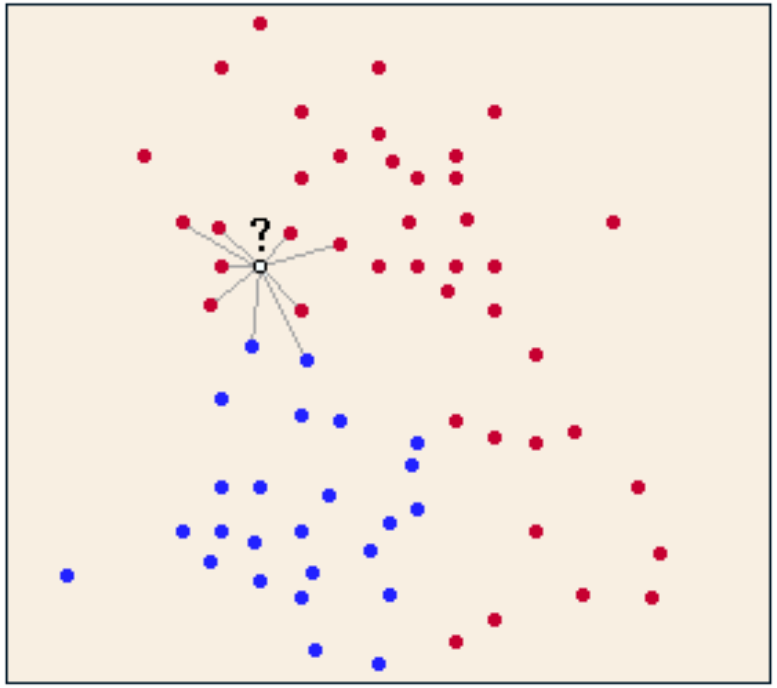
University of Wisconsin-Milwaukee

dsouza@uwm.edu

March 26, 2014

Overview

k-NN



Given a set of data points S, and a set of query points Q, for each point $q \in Q$ find a subset $K \subset S$, where size$(K) = k$, such that

$$d(s_k, q) < d(s_j, q) \forall s_k \in K, s_j \in S - K$$

where $d(a, b)$ is the distance metric with the following properties

$$d(a, b) = d(b, a) \geq 0$$
$$d(a, b) = 0 \Rightarrow a = b$$

Applications

- Bioinformatics [RMW+07, WEZ+04]
- Pattern recognition [DHS01]
- Cluster analysis [MHvL09, TSL00]
- Data mining [ZH00]

Methods for k-NN search

- Space partitioning (exact)
  - k-D Trees, Ball Trees [JOR11, AMN+94]
- Local Sensitive Hashing (approximate) [DIIM04]
- Brute force

Brute-force methods

- Computation of distance matrix
  - Formulated as a matrix multiplication operation (use CUBLAS)
- Selection of k-NN
  - Full sort and selection
  - Insertion sort-based selection
  - Heap sort-based selection
  - Quick select
  - Truncated bitonic sort
  - Radix select

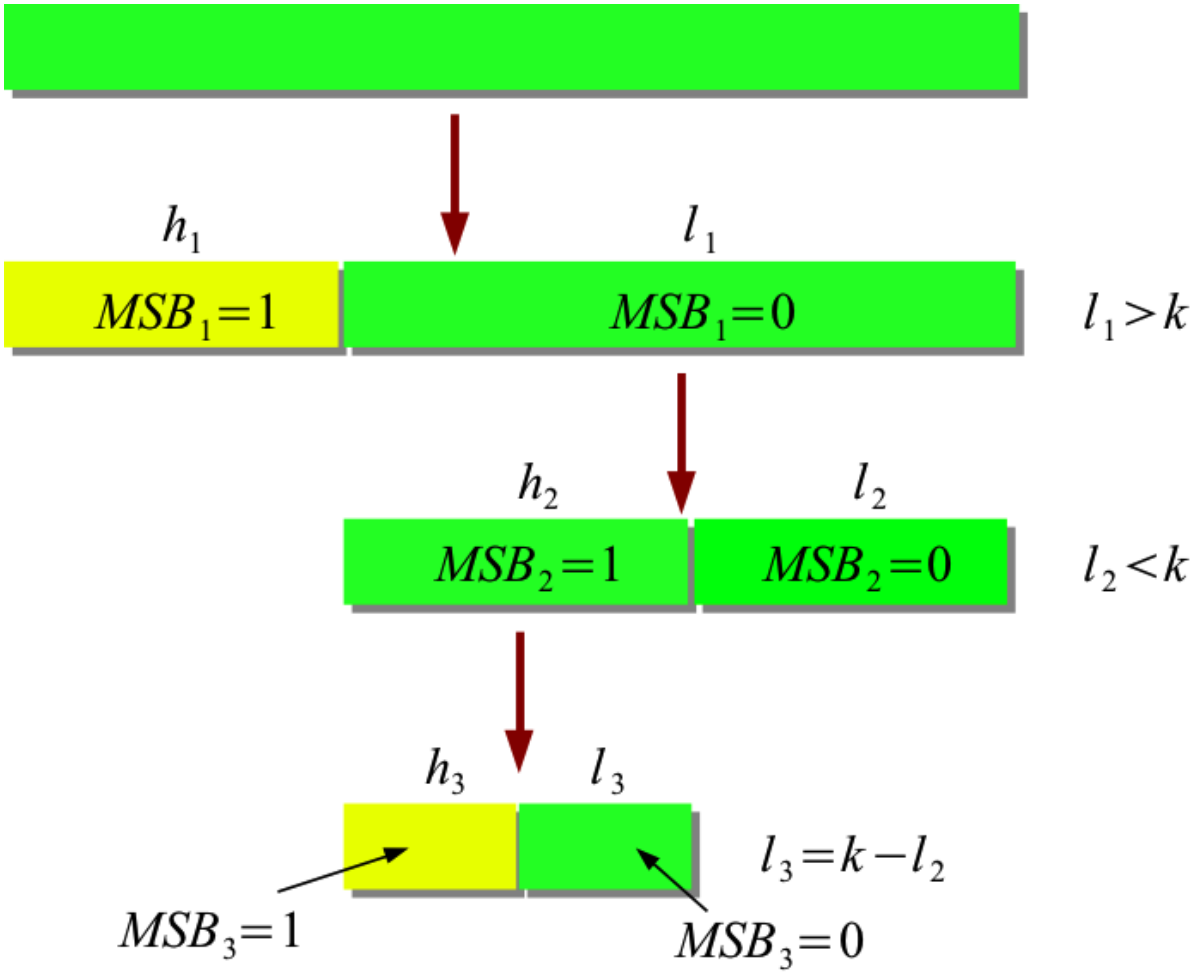
GPU-based brute-force methods

Differences pertain only to method of selection

- Heap sort-based selection [GDNB10, BGT+11, KH10]
- Insertion sort-based selection [ARBM12]
- Truncated bitonic sort [SPS12]

Radix Select

- Partition based on MSB
- Pruning



Partitioning

- Naïve approach (used by Cederman for quick sort [CT10])

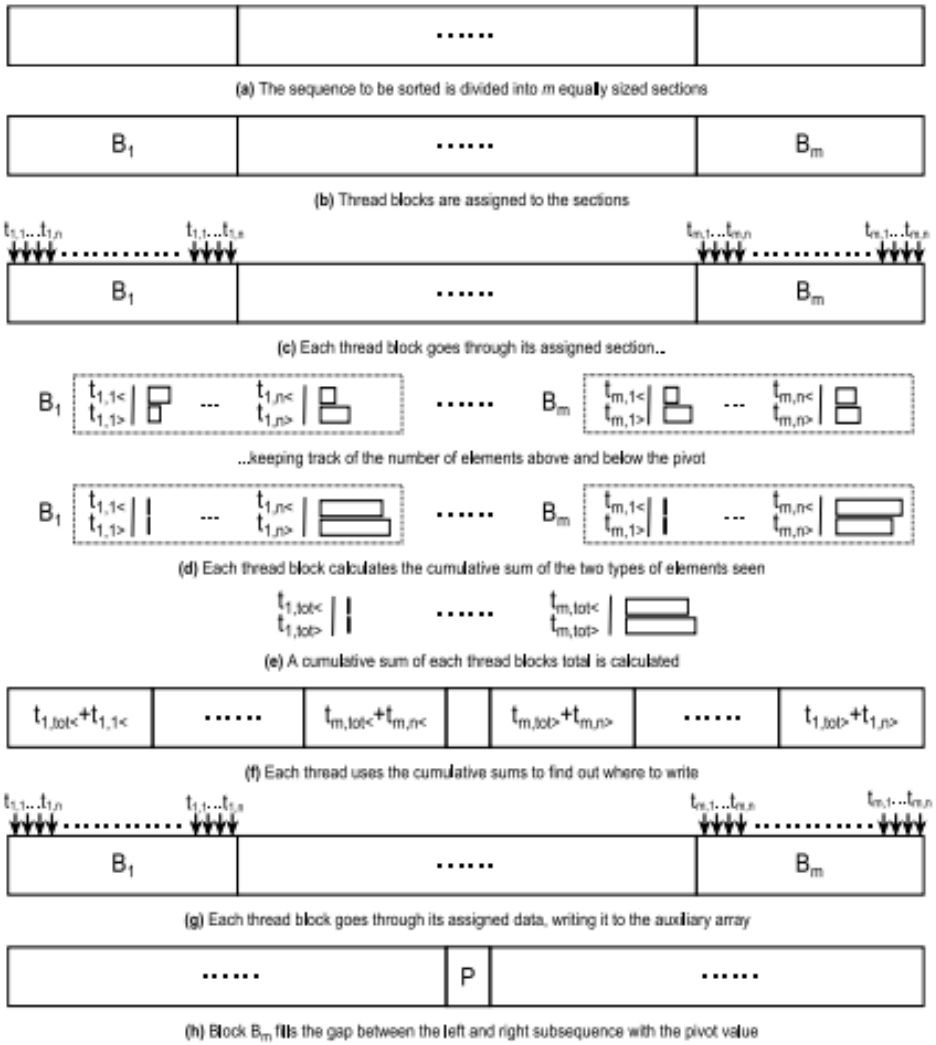


Fig. 3: Partitioning a sequence

Issues

- Two passes $\Rightarrow$ two global memory reads (even though both are coalesced)
- Read is coalesced, write is un-coalesced

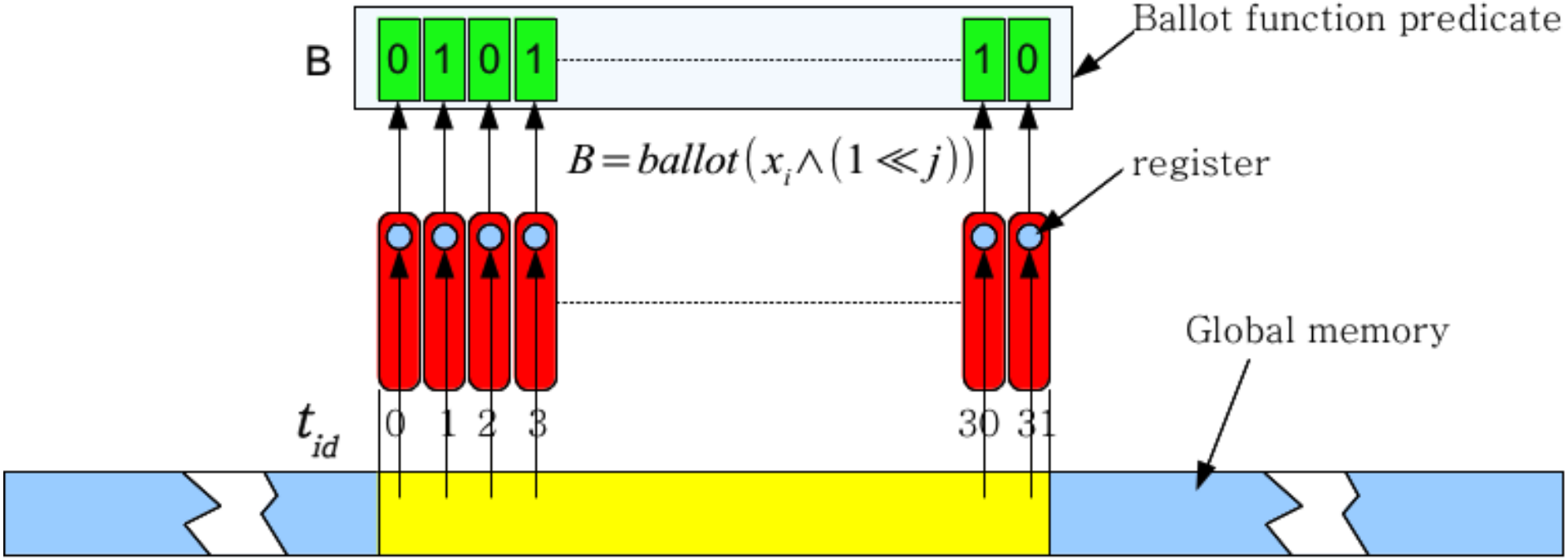
Radix Multi-Select

- Process multiple queries at the same time
  - One row of the distance matrix per thread block/thread warp
- BIG IDEA ?
  - Incremental partitioning
  - Using a combination of shared memory, `__ballot()` , bit shifting, and `__popc()`

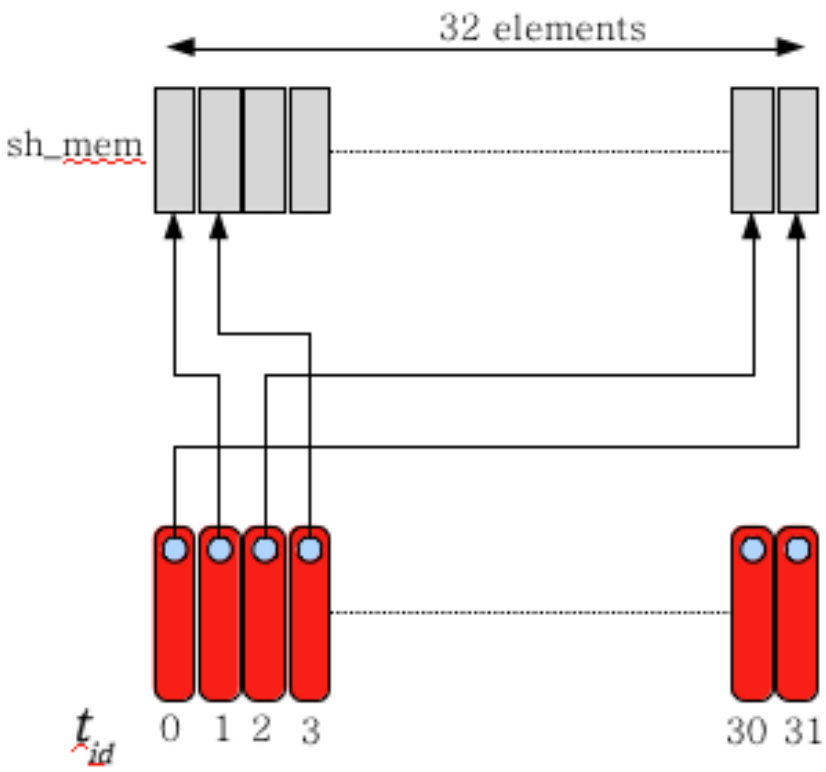
Intrinsic functions

- `unsigned int __ballot(`*perdicate*`)`
  - when predicate evaluates to TRUE the function returns a value with the Nth bit set
- `int __popc ( unsigned int` x `)`
  - returns the number of bits in x set to 1

Partition process: Read incrementally

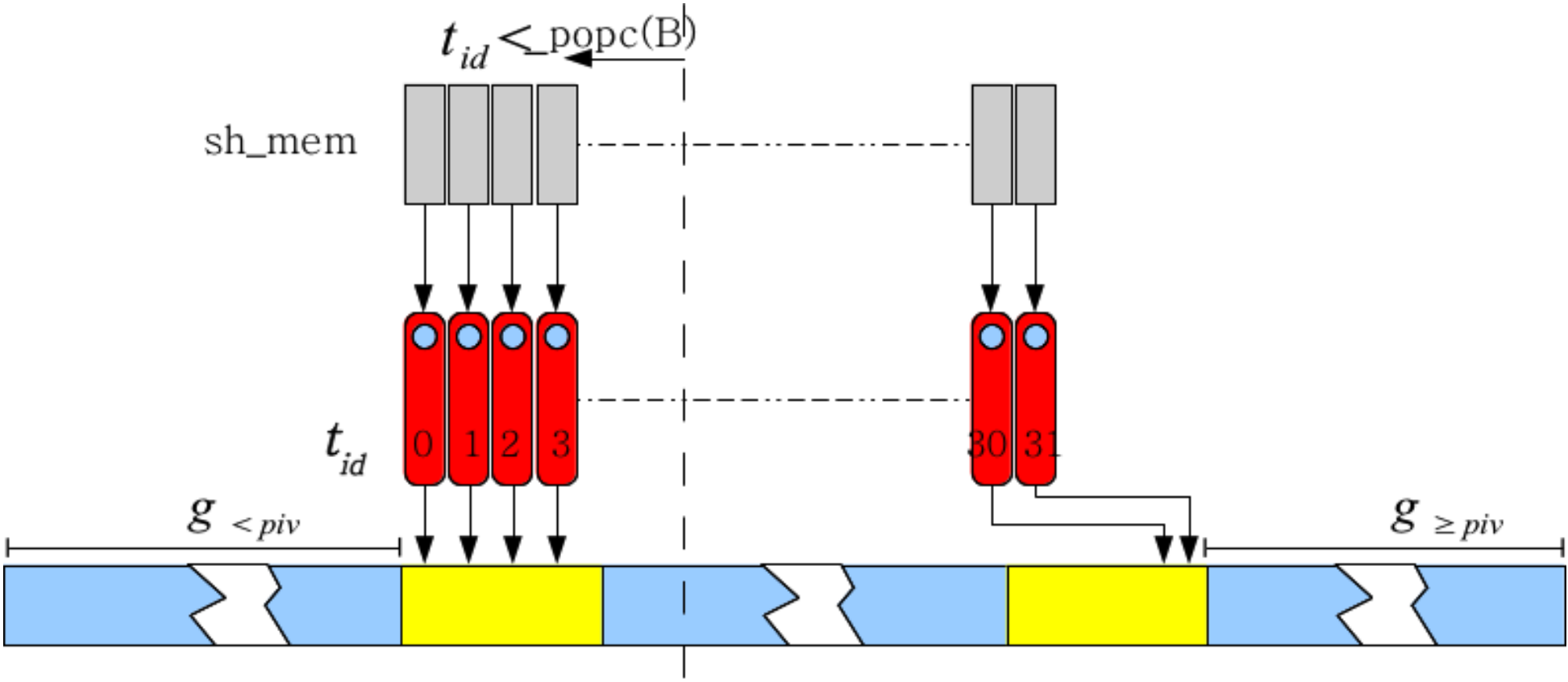


Partition process: Partition in shared memory



$$
\begin{aligned}
&\textit{if}\ (\neg(x_i \wedge (1 \ll j)))\ /{*}\text{elements with MSB=0}{*}/\\
&\qquad b = \neg B \gg (31-i)\\
&\qquad \text{sh\_mem}[\_\text{popc}(b)] = x_i\\
&\textit{else}\\
&\qquad b = B \gg (31-i)\\
&\qquad \text{sh\_mem}[31-\_\text{popc}(b)] = x_i\\
&\textit{end}
\end{aligned}
$$

Partition process: Write incrementally to auxiliary array



Thread Block per row

Each thread warp in the TB works as shown previously. The start point during the write process for the left and right hand side for each warp has be to offset to account for previous warps. This is accomplished using a summing operation within the TB.

Benchmark vs Truncated Bitonic Sort[SPS12]

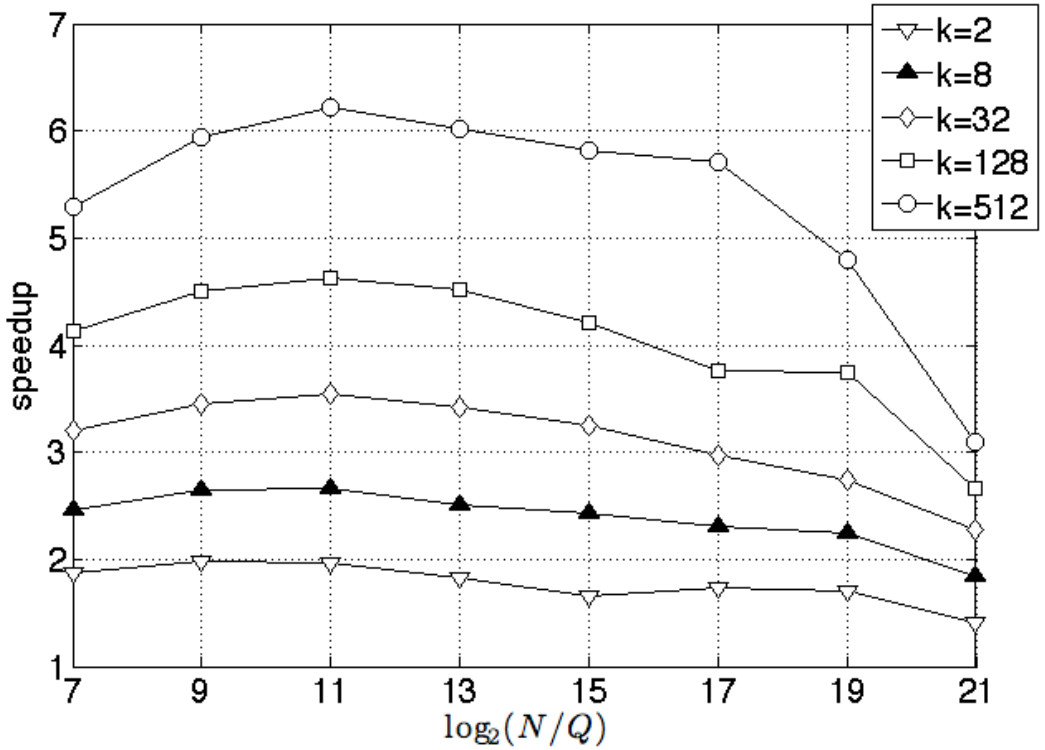


Figure: These tests were performed for elements that contained both values and indices. $-3 \leq \log_2(n/Q) \leq 20$. $n \times Q = 2^{27}$. $2 \leq k \leq 512$. Performance peaks at $\log_2(n/Q) = 7$ stays flat till $\log_2(n/Q) = 11$ and starts falling at $\log_2(n/Q) = 13$. The decrease in performance corresponds to the number of queries Q falling below 128 where our method underutilizes GPU resources.

Future work

- Processing multiple MSB in one partition
  - Will require binning - juggling multiple intrinsic calls
  - Will reduce global memory reads ?
- See what we can do with Maxwell architecture for decision making process

Acknowledgements

This work was partially funded by the National Science Foundation (NSF) through grant CCF-1013278. Views expressed are those of the authors alone and not necessarily those of the NSF.

- Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Wu, *An optimal algorithm for approximate nearest neighbor searching*, Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), SODA '94, Society for Industrial and Applied Mathematics, 1994, pp. 573–582.
- Ahmed Shamsul Arefin, Carlos Riveros, Regina Berretta, and Pablo Moscato, *Gpu-fs-knn: A software tool for fast and scalable knn computation using gpus*, PLoS ONE **7** (2012), no. 8, e44000.
- Ricardo J. Barrientos, José Ignacio Gómez, Christian Tenllado, Manuel Prieto-Matias, and Mauricio Marin, *knn query processing in metric spaces using gpus*, Euro-Par (1)'11, 2011, pp. 380–392.
- Daniel Cederman and Philippas Tsigas, *Gpu-quicksort: A practical quicksort algorithm for graphics processors*, J. Exp. Algorithmics **14** (2010), 4:1.4–4:1.24.

- Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, 2 ed., Wiley-Interscience, November 2001.
- Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, *Locality-sensitive hashing scheme based on p-stable distributions*, Proceedings of the twentieth annual symposium on Computational geometry (New York, NY, USA), SCG '04, ACM, 2004, pp. 253–262.
- Vincent Garcia, Eric Debreuve, Frank Nielsen, and Michel Barlaud, *K-nearest neighbor search: Fast gpu-based implementations and application to high-dimensional feature matching*, ICIP, 2010, pp. 3757–3760.
- Peter W. Jones, Andrei Osipov, and Vladimir Rokhlin, *Randomized approximate nearest neighbors algorithm*, Proceedings of the National Academy of Sciences **108** (2011), no. 38, 15679–15686.

- Kimikazu Kato and Tikara Hosino, *Solving k-nearest neighbor problem on multiple graphics processors*, Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (Washington, DC, USA), CCGRID '10, IEEE Computer Society, 2010, pp. 769–773.
- Markus Maier, Matthias Hein, and Ulrike von Luxburg, *Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters*, Theor. Comput. Sci. **410** (2009), no. 19, 1749–1764.
- Adam Roberts, Leonard McMillan, Wei Wang, Joel Parker, Ivan Rusyn, and David Threadgill, *Inferring missing genotypes in large snp panels using fast nearest-neighbor searches over sliding windows*, Bioinformatics **23** (2007), no. 13, i401–i407.
- N. Sismanis, N. Pitsianis, and Xiaobai Sun, *Parallel search of k-nearest neighbors with synchronous operations*, High Performance Extreme Computing (HPEC), 2012 IEEE Conference on, 2012, pp. 1–6.

- J. B. Tenenbaum, V. Silva, and J. C. Langford, *A Global Geometric Framework for Nonlinear Dimensionality Reduction*, Science **290** (2000), no. 5500, 2319–2323.
- J. Weston, A. Elisseeff, D. Zhou, C. S. Leslie, and W. S. Noble, *Protein ranking: from local to global structure in the protein similarity network.*, Proc Natl Acad Sci U S A **101** (2004), no. 17, 6559–6563.
- Mohammed Javeed Zaki and Ching-Tien Ho (eds.), *Large-scale parallel data mining, workshop on large-scale parallel kdd systems, sigkdd, august 15, 1999, san diego, ca, usa, revised papers*, Lecture Notes in Computer Science, vol. 1759, Springer, 2000.