GPGPU-based Data Structures and Algorithms for Geospatial Computation – A Summary of Results and Future Roadmap

Prof. Sushil K. Prasad
PhD Students: Xi He, Satish Puri, and Mike Mcdermott
Georgia State University

Prof. Shashi Shekhar
PhD Students: Xun Zhou and Mike Evans
University of Minnesota

Why GPGPU and GIS?

The computer architecture now is massively parallel and hybrid, with a pair of multi-core CPU and many-core GPGPU (with dozens of cores on a CPU and hundreds of cores on a GPGPU) now commonplace in laptops and desktops and the computer nodes of high-performance machines and clouds. Parallel and distributed processing is, therefore, imperative for data-and-compute-intensive geospatial computations, such as for polygonal overlay [12-17], evacuation routing [20-21], and interesting interval/region discovery [9, 22-24]. An efficient utilization of the CPU-GPGPU pair is critical, else the GIS programs will remain inefficient, incurring loss of one to two orders of magnitude in speedup. In addition, from GIS user perspective, a key niche for GPGPU (in contrast with cloud computing or cluster) may lie in price-performance trade-off - most laptops and workstations come with GPUs providing opportunities for an order of magnitude faster response time for many GIS problems without additional hardware cost. The existing parallel algorithms and data structures port reasonably well to multi-core CPUs, but poorly to GPGPUs because of latter's atypical fine-grained, single-instruction multiple-thread (SIMT) architecture, extreme memory hierarchy and coalesced access requirements, and delicate CPU-GPU coordination.

Therefore, we foresee the need for significant undertaking by the geospatial and parallel processing communities to redesign the traditional parallel data structures and algorithms and discover new parallel techniques and tools for GPGPU platform.

Summary of our Current Results on GPGPUs

We have undertaken parallelization of two key tree-based data structures, namely R-tree and heap, and have employed parallel R-tree in polygon overlay system and have parallelized interest interval discovery problem. These data structure parallelization are hard because of the underlying tree topology and the fine-grained computation leading to frequent access to such data structures severely stifling parallel efficiency. Therefore, the current best parallelization of R-tree on GPU was limited to about 20-fold speedup [10,11,18,19]. A potentially transformative concept we have developed is *Geo-Packing*. On GPGPUs, the spatially co-related objects can be exploited both for efficient parallel construction of R-tree data structure via their bottom-up packing, and for efficient parallel search of R-tree by their top-down packing. Such Geo-Packing principle can harness GPU-specific memory coalescing and synchronous thread grouping (warp). We summarize our current results as follows.

Parallel R-Tree: Our new parallel algorithms for construction and search has yielded the first demonstrated 200-fold speedup in R-tree construction on a GPU (patent pending, [1]). This would be useful for large-scale range querying and can serve as template for parallelizing other tree-based structures such as MVR tree, TPR tree, and other generalizations of spatio-temporal datasets.

Parallel Polygon Overlay System: Our GIS system employing the parallel R-tree can process about 200K polygons within a few seconds with 70-fold speedup on a [12-node Linux cluster that previously took tens of minutes [2,4-6]. Thus, it has the potential for bringing a practical overlay tool to the Geo Scientists.

Parallel Priority Queue: Our parallel heap data structure supports large batches of extracting highest priority items and inserting newly produced items with 30-fold speedups (patent pending, [3]). This has potential application to shortest path and evacuation route planning. For latter, the current algorithms are sequential and slow. For instance, state of art CCRP routing algorithm took more than a day of computation time to compute evacuation routes and schedules for San Francisco [7,8].

Parallel Interest Intervals: Most recently, our parallel program based on [9] calculates one-dimensional interesting intervals over an image representing the normalized difference vegetation indices of Sahel area of Africa within 30 ms on an NVIDIA 480GTX. This allowed us to process 610 images representing biweekly data from July 1981 through Dec 2006 within 20 seconds. We were also able to pipe the output to a display in almost real-time, which would interest Climate Scientists.

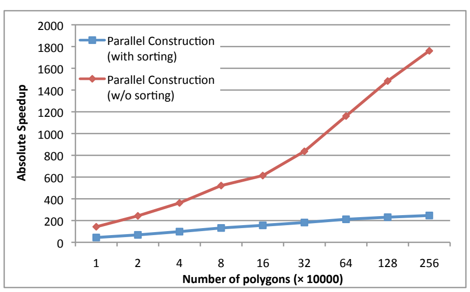

Figure 1: R-tree construction

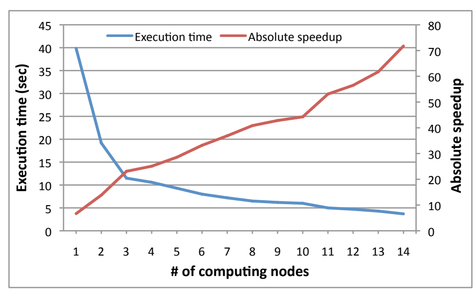

Figure 2: GIS polygonal overlay system

Future Roadmap

A GIS overlay and evacuation routing system: For a metropolitan area of 5 Million people over 500 square mile, such as for Houston, we expect a complete evacuation routing plan within one minute, which currently takes over an hour, representing two orders of magnitude improvement. For overlay operations such as intersection of sufficiently large GIS file, we aim scaling up to 100 CPU-GPGPU node cluster while maintaining 70-80% efficiency.

Interesting intervals/region discovery: We expect real-time visualizations for two and three-dimensional problems with video-like scrolling, zooming, etc., using a large GPU cluster.

References

1. Xi He and S. K. Prasad. 2013. GPU-based Parallel R-tree Construction and Querying and Its Application to GIS Polygon Overlay Processing, *Provisional Patent Application.*
2. Satish Puri, Dinesh Agarwal, Xi He, and Sushil K. Prasad. 2013. MapReduce algorithms for GIS Polygonal Overlay Processing. In *IEEE International Parallel and Distributed Processing Symposium workshops (HPGC),* Boston, May. (Download Paper)
3. Xi He, Dinesh Agarwal, and S. K. Prasad. 2012. Design and Implementation of a Parallel Priority Queue on Many-core Architectures. In *Procs. Intl High Performance Computing (HiPC)*, IEEE, Pune, Dec. (Download Paper)
4. Dinesh Agarwal, and S. K. Prasad. 2012. Lessons learnt from the development of GIS overlay processing application on Azure Cloud platform. In IEEE CLOUD 2012 5th International Conference on Cloud Computing Honolulu, Hawaii, USA, June. (Download Paper)
5. Dinesh Agarwal, Satish Puri, Xi He, and S. K. Prasad. 2012. An end-to-end system for GIS polygonal overlay computation on Linux Cluster - An experience and performance report. *In Proceedings of the 26th IEEE International Parallel & Distributed Processing Symposium, Workshops and Phd Forum.,* Shanghai, China. (Download Paper)
6. Dinesh Agarwal, Satish Puri, Xi He, and S. K. Prasad. 2012. *Cloud Computing for Fundamental Spatial Operations on Polygonal GIS Data. In Cloud Futures 2012 - Hot Topics in research and education,* Berkeley, California, May 2012. (Download Paper)
7. Q. Lu, B. George, and S. Shekhar, Capacity constrained routing algorithms for evacuation planning: A summary of results," Advances in Spatial and Temporal Databases, 2005.
8. K. Yang, V. Gunturi, and S. Shekhar, A dartboard network cut based approach to evacuation route planning: A summary of results," Geographic Information Science, 2012.
9. Xun Zhou, Shashi Shekhar, Pradeep Mohan, Stefan Liess, and Peter K. Snyder. 2011. Discovering interesting sub-paths in spatiotemporal datasets: a summary of results. In Proceedings of the *19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems(GIS '11),* Divyakant Agrawal, Isabel Cruz, Christian S. Jensen, Eyal Ofek, and Egemen Tanin (Eds.). ACM, New York, NY, USA, 44-53. DOI=10.1145/2093973.2093981 http://doi.acm.org/10.1145/2093973.2093981
10. M. Kunjir and A. Manthramurthy. Using Graphics Processing in Spatial Indexing Algorithms. Research report, Indian Institute of Science, Database Systems Lab**, 2009.**
11. L. Luo, M. D. F. Wong, and L. Leong. Parallel implementation of r-trees on the GPU. In ASP-DAC, pages 353–358, 2012.
12. K. Wang, Y. Huai, R. Lee, F. Wang, X. Zhang, and J. H. Saltz. Accelerating pathology image data cross-comparison on CPU-GPU hybrid systems. Proceedings of the VLDB Endowment, 5(11):1543–1554, 2012.
13. T. M. Chan. A Simple Trapezoid Sweep Algorithm for Reporting Red/Blue Segment Intersections. In In Proc. 6th Canad. Conf. Comput. Geom, pages 263–268, 1994.
14. B. Chazelle and H. Edelsbrunner. An Optimal Algorithm for Intersecting Line Segments in the Plane. In FOCS, pages 590–600, 1988.
15. R. G. Healey, M. J. Minetar, and S. Dowers, editors. Parallel Processing Algorithms for

GIS. Taylor & Francis, Inc., Bristol, PA, USA, 1997.

16. J. D. Hobby. Practical segment intersection with finite precision output. Computational Geometry, 13(4):199 – 214, 1999.

17. F. Wang. A parallel intersection algorithm for vector polygon overlay. Computer Graphics and Applications, IEEE, 13(2):74 –81, mar 1993.

18. J. K. Chen, Y.-F. Huang, and Y.-H. Chin. A Study of Concurrent Operations on R-Trees. Inf. Sci., 98(1-4):263–300, 1997.

19. B. Schnitzer and S. T. Leutenegger. Master-Client R-Trees: A New Parallel R-Tree Architecture. In SSDBM, pages 68–77, 1999.

20. H. Hamacher and S. Tjandra, “Mathematical modelling of evacuation problems–a state of the art,” Pedestrian and Evacuation Dynamics, vol. 2002, pp. 227–266, 2002.

21. Cova, Thomas J., and Justin P. Johnson. "A network flow model for lane-based evacuation routing." Transportation Research Part A: Policy and Practice 37.7 (2003): 579-604.

22. J. M. Kang, S. Shekhar, C. Wennen, P. Novak, Discovering Flow Anomalies: A SWEET Approach, In the Eighth IEEE International Conference on Data Mining (ICDM '08), pp. 851-856, Pisa, Italy, December 15-19, 2008

23. Kulldorff, Martin. "A spatial scan statistic." *Communications in Statistics-Theory and methods* 26.6 (1997): 1481-1496.

24. Kulldorff, Martin, et al. "A space–time permutation scan statistic for disease outbreak detection." *PLoS medicine* 2.3 (2005): e59.