Fundamental & Computational Sciences

Advanced Computing, Mathematics and Data Division Research Highlights

January 2015

The Speed to Solution

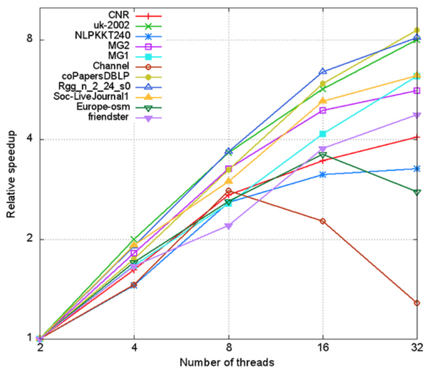
Addressing fast community detection and other related problems

Notably, in science, the problems you start with may not be the only ones you solve.

Most research begins with a need to address a problem. Highly technical or surprisingly simple, the problem dictates the path to solution. As with most things, solving those problems also can come with deviations and additions. For scientists in the Advanced Computing, Mathematics, and Data Division at Pacific Northwest National Laboratory, their work often crosscuts many domain science sectors within the Laboratory and among external collaborators. In this case, seeking a way to use algorithmic graph theory to enhance data analytics of biological sequences led to a distinct intersection with work being done for high-performance computing (HPC) applications contending with obstacles related to power constraints and massive data movement.

The Starting Point

While working with biologists on research involving biological sequencing, Ananth Kalyanaraman, an associate professor at Washington State University's (WSU) School of Electrical Engineering and Computer Science, faced a challenge: because of the sheer scope of data generated from varied technologies, it was essential to find a way to make said data 1) less redundant and 2) useful. He partnered with Mahantesh Halappanavar, the current Analysis and Algorithms team lead within the ACMD Division's Data Sciences group, seeking a way to improve the value of the biological data using mathematical tools, namely graph theory and a clustering operation known as "community detection," which is used in graph applications to reveal natural divisions that exist in networks without size or element constraints. One obvious example of community detection at work is Facebook, where millions of users connect through "friend" links, and rich social networks are organized via these connections.

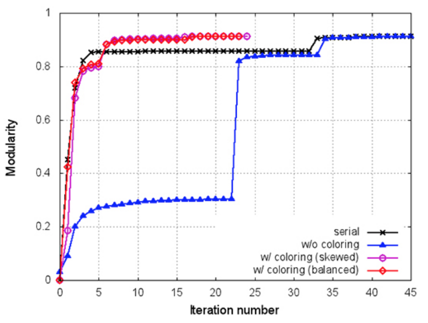


Speedup of Grappolo on a multi-core platform for different inputs. Speedups are relative to the same code using two threads. Enlarge Image.

"We modeled biological sequences in terms of their functional or structural similarity, such as being similar in DNA or as a protein," Kalyanaraman explained. "This resulted in large graphs with millions of nodes representing proteins or DNA segments and hundreds of millions to billions of edges representing their pairwise relationships. However, by organizing the graphs and grouping by similarity, we could rank and identify the biological communities or networks, which were fewer in number than the nodes. From there, we could determine what these sequences do based on their community affiliations. Ultimately, we were left with a manageable database containing biological sequences that relate to specific biological functions. The utility in this database is its ability to generate testable hypothesis."

Kalyanaraman and Halappanavar were able to use this methodology because the biological sequencing question opened the door to addressing one big problem with community detection in graph-theoretic applications: limited support on large-scale parallel computers.

The Louvain Method



Evolution of modularity gain as the algorithm progresses (iterations along X axis) for CNR data set. Parallel implementation without coloring converges slowly. With coloring, the parallel algorithm matches the serial algorithm's performance (Blondel et al. 2008). Enlarge Image.

In an effort to amplify community detection in graphs, Kalyanaraman and Halappanavar used the Louvain method to identify communities in the large biological networks. This method first locally optimizes clusters then combines those belonging to the same community into a network. The Louvain method can do this in a fast, memory-efficient manner. However, its traditional formulation is inherently sequential, which poses a distinct problem for the scalability needed to support complex and expanding data analytics. Kalyanaraman and Halappanavar, with graduate student Hao Lu from WSU, devised a variant of the Louvain algorithm that uses novel heuristics, or methods that speed up processes to practical solution, for parallelization on multithreaded architectures (improving weaker serial hardware performance). Their parallel algorithm, named *Grappolo*, the Italian word for "bunch" (or, cluster) of grapes, employs heuristics that take advantage of some key structural properties of the input graph and uses a technique known as "graph coloring" in deriving an efficient parallel schedule.

Their experiments, featuring 11 real-world networks with diverse characteristics and spanning applications in biology, social networking, and scientific computing, produced excellent scaling using the parallelization Louvain algorithm while achieving equivalent modularity (a metric to assess the quality of clusters detected) comparable to the serial implementation. It also proved to be stable, producing consistent output across varying numbers of threads. Moreover, they were able to produce high-quality clustering with measureable speedups (16x using 32 threads) and scalability (up to 32 threads) on a standard multi-core Intel Xeon processing platform.

"Using graph coloring to parallelize the Louvain method opened up new possibilities for us," Halappanavar said. "Otherwise, the convergence of the parallel algorithm is slow, which reduces the parallel efficiency. Coloring not only provides a means to exploit concurrency, it also offers a way to maximize modularity."

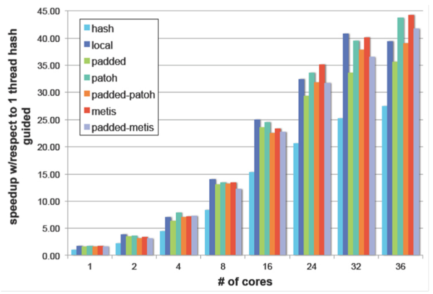
An open-source version of Grappolo is available online and can be downloaded at: http://hpc.pnl.gov/people/hala/grappolo.html.

Energy-efficient Grappolo

An interesting side effect of the research that started with Kalyanaraman's biology data dilemma at WSU is that it inspired conversations with Daniel Chavarría-Miranda, a senior scientist in ACMD Division's High Performance Computing group, who has been experimenting with optimizing irregular applications to improve their performance and power on many-core computing architectures. The work with Grappolo and the underlying data analytics efforts spawned yet another problem to explore: tackling the challenges posed by power constraints and data movement, which currently inhibit HPC applications.

"Power-aware energy consumption for HPC is growing, especially in commercial sectors with massive data," Chavarría-Miranda explained. "In terms of economic cost, power dictates how big a company can make its data centers and what services it can provide. We are seeking an optimization process that can reduce energy-consumption significantly to enable faster analysis of even bigger data sets. Ultimately, you want the software to reduce the energy consumption while achieving the same output."

The Grappolo implementation added a significant dimension to Chavarría-Miranda's work using low-power processors, as he noted: "Compared to serial detection, which is excellent on its own, we had to come up with non-trivial ways to achieve parallelism."



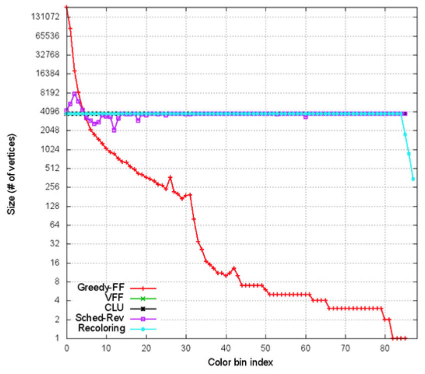
Strong scaling results on Tilera for the Europe data set using guided scheduling for different techniques. Speedup is relative to serial performance in Blondel et al. (2008). Enlarge Image.

Chavarría-Miranda, Halappanavar, and Kalyanaraman worked on multiple design strategies to develop a scalable tool for community detection on the Tilera platform—the first effort of its kind to address graph algorithms using the many-core architecture. They observed excellent scaling on a 36-core Tilera platform. At PNNL, there are four Tilera testbeds, which operate at half the speed of typical processors. They are distinct from processors found, for example, in a desktop computer in that they are built at every level, including the component parts, to be power efficient. Their "low-speed" design also stems from being dedicated systems without the need to maintain the varied compatibility required by a desktop. As such, these architectures are relevant in addressing the mismatch between voluminous (and growing) data sets and constrains in power and energy available to individual platforms and data centers.

Balancing Colors

One of the primary improvements in Grappolo is its use of distance-1 graph coloring, a fundamental graph problem where each vertex is assigned a color so no two neighboring vertices have the same color while minimizing the total number of colors used. The distance-1 graph coloring heuristic assures fast and computationally efficient convergence on the many-core platform by avoiding making concurrent decisions on neighboring vertices. However, the widely varied sizes of these color sets (number of vertices with a given color) posed a distinct challenge to Tilera, reducing the degree of parallelism available as the algorithm proceeds through each step. To remedy this problem, the researchers employed a bundling heuristic, where attention to colors was applied only among a certain percentage of vertices based on the size of color sets. Empirically, the researchers processed about 20 percent of the vertices based on their colors, but the remaining 80 percent were bundled and processed concurrently. The process resulted in speedups up to 47x on 36 cores over an earlier serial implementation on Tilera in 2008 by Blondel et al.

While the gains proved impressive, the researchers still felt there was more problem left to solve. They partnered with Assefaw Gebremedhin, an assistant professor also from WSU, and developed several serial and parallel heuristics for balanced coloring. As the name implies, "balanced coloring" aims to balance the number of vertices in a color class while minimizing the number of colors used, and the heuristics applied in their experiments proved able to achieve this goal efficiently. In addition to community detection, balanced coloring has numerous applications in scientific computing, including machine learning, finite element analysis, and fluid dynamics—all areas currently being explored by the PNNL and WSU teams.

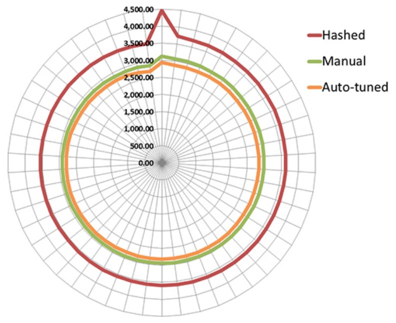


Distribution of color-class sizes for coloring computed from different heuristics. The red line represents the original algorithm with significant variance. All other lines are based from the new heuristics: the flatter the line, the better. Enlarge Image.

What's Next?

Work involving algorithmic graph theory; data analytics; and power, performance, and reliability continues on several fronts. One aspect is examining how irregular applications targeted specifically to low-power architectures, like Tilera, may help overcome the power limitations imposed by emerging processors. Auto-tuning techniques also are being developed that perform much better than default schemes adopted by the system and manual efforts from a skilled developer. Porting Grappolo to other many-core platforms, such as those from Nvidia and Intel, also is being explored. On the algorithmic front, newer metrics for clustering are being considered, as well as novel applications for balanced coloring.

Markedly, the case studies presented by these combined works involving algorithm development, parallelization, and testing for the community detection application show how the evolution of specific graph theory methods, such as balanced coloring, are a boon to developers seeking to improve application performance. Most directly, continued mathematical evaluation and experimentation of graph-theoretic applications are critical to dealing with increasingly larger data sets and resolution limits posed by existing hardware.

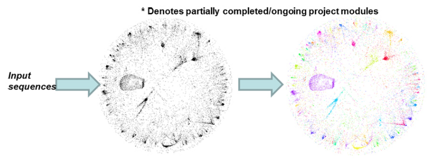


Energy consumption of different implementations for community detection on Tilera. Lower consumption is better. Enlarge Image.

"Right now, we are generating data without knowing what we get—there is just so much data to process," Kalyanaraman said. "Using these methods reduces the millions of data points to something smaller for more focused hypothesis testing. The clusters gleaned from our community detection methods will provide an indication of what is happening. For example, in one effort, we started with roughly 2.5 million sequences derived from a gut microbial community. After community detection, we got down to 200,000 sequences, ranking them further into close to 90 information-rich, related groups. It shows we can hone in on a point that works together rather than try to test every possible solution. These applications supply hypotheses that can really be tested, which provides a good start to domain scientists."

Acknowledgments:

At PNNL, the graph algorithm work and research was supported by the Center for Adaptive Supercomputing Software, while power and performance optimization for high-performance computing architectures was supported by the U.S. Department of Defense under the Autotuning for Power, Energy & Resilience (ATPER) project. At WSU, the research was made possible by the U.S. Department of Energy Office of Advanced Scientific Computing Research Early Career Research award DE-SC-0006516.



This graph contains open reading frames (as vertices) from an ocean metagenomics set. The graph features 26,000 vertices and 1 million edges. Clustering was performed using OpenMP implementation, which took only 7 seconds to cluster using 8 cores and produced an output with modularity of 0.824 (visual demonstration by Gephi). Enlarge Image.

Related Work:

Blondel VD, J-L Guillaume, R Lambiotte, and E Lefebvre. 2008. "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment* 2008(10):P10008. DOI: 10.1088/1742-5468/2008/10/P10008.

Lu H, M Halappanavar, and A Kalyanaraman. "Parallel Heuristics for Scalable Community Detection." *Parallel Computing*, submitted, October 2014. http://arxiv.org/abs/1410.1237v2.

Chavarría-Miranda D, M Halappanavar, and A Kalyanaraman. 2014. "Scaling Graph Community Detection on the Tilera Many-core Architecture." Presented at: *IEEE International Conference on High Performance Computing (HiPC)*. December 17-20, 2014, Goa, India. IEEE Computer Society, Washington, D.C.

Lu H, M Halappanavar, D Chavarría-Miranda, A Gebremedhin, and A Kalyanaraman. 2015. "Balanced Coloring for Parallel Computing Applications." In: *29th IEEE International Parallel & Distributed Processing Symposium*. May 25-29, 2015, Hyderabad, India. IEEE Computer Society, Washington, D.C. (Accepted).

Chavarría-Miranda D, A Tumeo, J Manzano, and M Halappanavar. "Optimizing Irregular Applications for Performance and Energy on the Tilera Many-core Architecture." (In review).