U.S. DEPARTMENT OF ENERGY
This work is supported by the Office of Advanced Scientific Computing Research (ASCR)

Scalable and Power Efficient Data Analytics for Hybrid Exascale Systems

Alok Choudhary[1], Wei-keng Liao[1], William Hendrix[1], Prabhat Kumar[1], Nagiza Samatova[2], John Wu[3]
[1]Northwestern University, [2]North Carolina State University, [2]Oak Ridge National Laboratory, [3]Lawrence Berkeley National Laboratory

Project Goals

Data Mining Kernels on Hybrid Architectures

- GPU-based analysis kernels
- Coordinated CPU-GPU programming
- SSDs for out-of-core computation

Energy Efficiency

- Optimizing data access in memory hierarchy
- Developing approximate analytics kernels
- Performance and energy trade-offs

Index-based Query and Analysis

- Data analysis on indexed data based on FastBit
- Parallel query on distributed indexed data
- Perturbation analytics for noisy and uncertain data

Data Analytics Kernels

The frequency of kernel operations in illustrative data mining algorithms

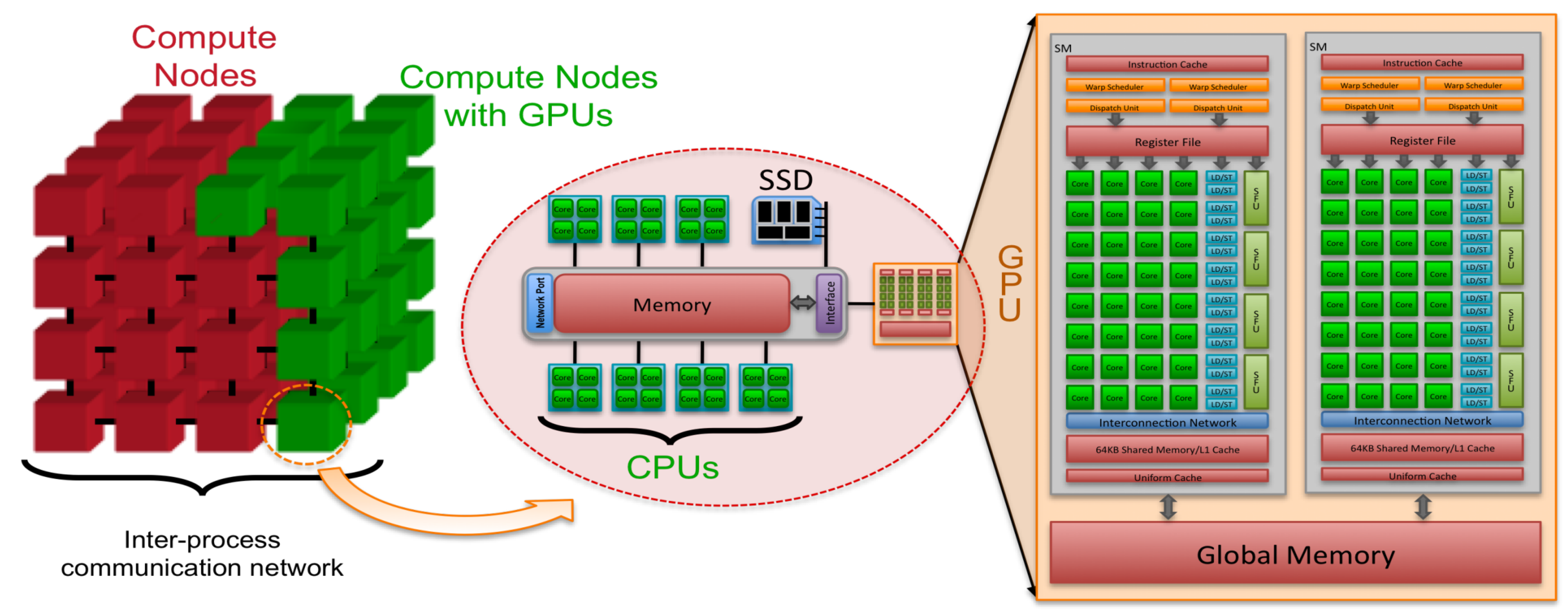
Application	Top 3 Kernels			Sum %
	Kernel 1 (%)	Kernel 2 (%)	Kernel 3 (%)	
K-means	Distance (68)	Center (21)	minDist (10)	99
Fuzzy K-means	Center (58)	Distance (39)	fuzzySum (1)	98
BIRCH	Distance (54)	Variance (22)	redist.(10)	86
HOP	Density (39)	Search (30)	Gather (23)	92
Naïve Bayesian	probCal (49)	Variance (38)	dataRead (10)	97
ScalParC	Classify (37)	giniCalc (36)	Compare (24)	97
Apriori	Subset (58)	dataRead (14)	Increment (8)	80
Eclat	Intersect (39)	addClass (23)	invertC (10)	72
SVMlight	quotMatrix(57)	quadGrad (38)	quotUpdate(2)	97

- Perfomance of representative data mining algorithms is dominated by a small number of kernels
- Top 3 kernels usually exceed 90% of execution time
- If these kernels were effectively executed, the overall applications could be significantly accelerated
- Research products
  - A C/Fortran/Cuda library of highly optimized analytical kernels
  - A framework for programmers to combine these kernels
  - Integration to popular analytics/visualization software, such as Matlab, R, VisIt.

An example of data analytics, statistics, and mining functions that will be explored as a part of this project

Functionality	Examples
Global reduction	sum, max, min, mean
Time Series Similarity	TAPER, mutual information
Distribution	standard deviation, histograms, etc
Data Preprocessing (e.g., dimensionality reduction)	PCA, ABB, LVF
Clustering	K-means, MAFIA, DBSCAN, Bisecting K-means, SNN
Anomaly/Outlier Detection	LOF, Outlier Detection
Change Detection	Change detection in time series
Predictive modeling, classification	ScalParC, Decision trees, Naïve Bayesian, RIPPER, SVMlight
Association rule mining	Apriori, FP-growth, MAFIA
Feature extraction	Edge detection, Blob detection

Harnessing Hybrid Architectures



- Design algorithms for data analysis kernels accelerated on hybrid HPC with a mix of multi-core CPUs, GPUs, and SSDs
- Data access management of memory hierarchy (hard disks / SSD / host memory / GPU memory)
  - Develop a memory tiling technique
  - Coordinated CPU and GPU analytical computing
  - Out-of-core analytics using SSDs
  - Utilize data locality and parallel I/O techniques

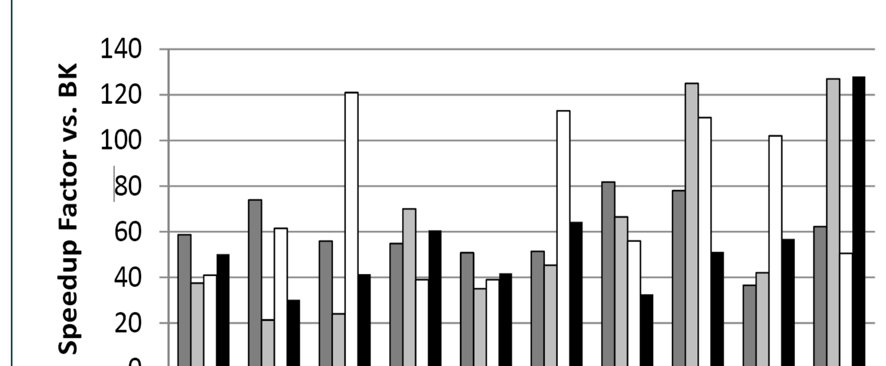
Lessons Learned from Exploring Backtracking Paradigm on the GPU

- Backtracking is a depth-first exploration of a problem space, ubiquitous in memory-intensive data analytics problems.
  - Using exemplar problem of Maximal Clique Enumeration, performance of *hard* backtracking problems on GPU limited to ~2.25 times a single CPU core (see table for def. of hardness).
  - Optimizations due to efficient output buffering, load-balanced, fine-grain parallelization of search, and memory latency hiding through saturation and efficient memory usage essential to performance.

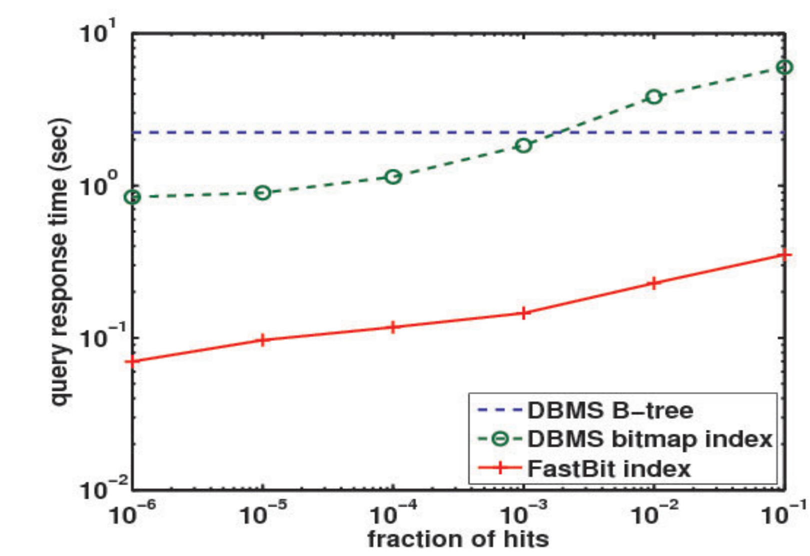
	Backtracking (worst-case)	GPU optimal
Problem Instance	Irregular access pattern (e.g. sparse matrix multiplication)	Regular access with locality (e.g. dense matrix multiplication)
Work Unit	Variable in size and computation (e.g. CSP on large sets)	Constant size, SIMD (e.g. stream processing)
Output	Exponential size, hard to estimate (e.g. subset enumeration)	Polynomial size, apriori (e.g. dense matrix multiplication)
Search space	Tree-based, unbalanced (e.g. 8-queens)	Fixed, apriori (if applicable) (e.g. k-d trees)

Utilizing Indexed Data

- Develop index-based data analysis kernels and algorithms for performance and power optimizations
- Query Processing
  - Indexing array-based data
  - FastBit compressed bitmap indexing
  - FastBit indexes can answer queries more than 10X faster than commonly used technique
  - FastBit indexing has been extended to HDF5 and NetCDF formats



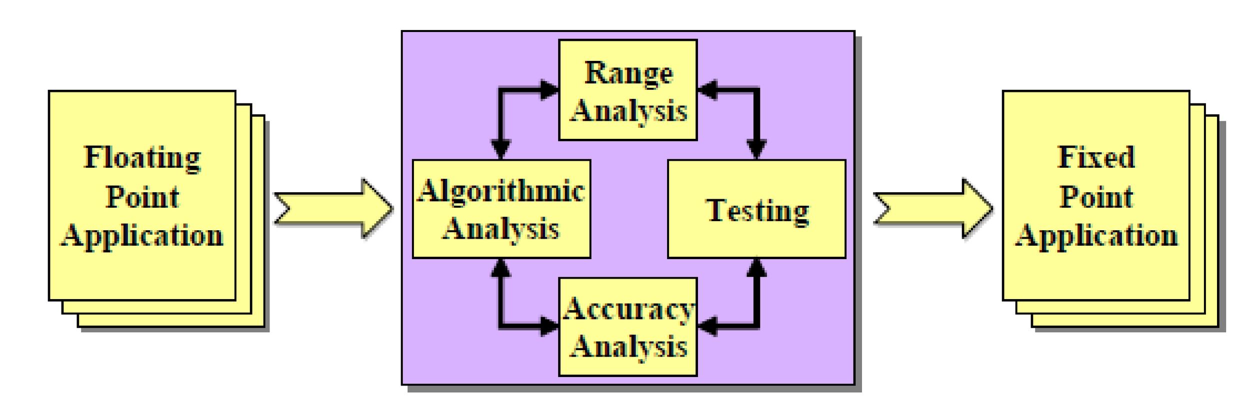
The speedup of community enumeration with the perturbed algorithm vs. the traditional algorithm for perturbed graphs for different edge-weight thresholds in protein interaction networks (E. coli (white), H. pylori (black), Synecosystis (grey), and S. typhimurium (dark grey)).



FastBit indexes answering queries more than 10X faster than commonly used techniques

Approximate analytics algorithms

- Convert floating-point operations into integer operations
- Provide better performance and are more energy efficient
- Use multi-level strategies via divide-and-conquer
- Approaches
  - Parallelization and acceleration on GPUs
  - Determining performance and accuracy with respect to original algorithms
  - User power measurement devices to evaluate the actual energy consumption



Project contact: Alok Choudhary <choudhar@eecs.northwestern.edu>