EDGAR

- **Energy-efficient Data and Graph Algorithms Research**
- Funded by Applied Math, ASCR
- Program manager: Sandy Landsberg
- Early Career Research Program (start: 2013)
- PI: Aydın Buluç (Berkeley Lab)
- Postdoctoral Fellows:
  - Ariful Azad (100%, Feb 2014-now)
  - Harsha Simhadri (40%, Sep 2013- Sep 2014)
- Students (short term):
  - Chaitanya Aluru (Nov 2014-now)
  - Eric Lee (Summer 2014)
**Problem:** maximum cardinality matching in bipartite graph.

**Application:** block triangular forms of matrices, least square problems, circuit simulation, weighted matching.

**Algorithm:** Search for disjoint paths alternating between matched and unmatched edges.

**Innovation:** Re-use search trees created in one phase in the next phase by grafting branches of trees. Significantly reduces work in the tree-traversal and exposes more parallelism.

---

**Performance:** On 40-core Intel Westmere-EX
On average **7x** faster than current best algorithm. Can be up to **42x** faster.

![Graph showing relative performance of different algorithms on various datasets]

**Scaling:** One node of Edison (24-core Intel Ivy Bridge).
On average **17x** speedups relative to serial algorithm.

**Future direction:** expose parallelism suitable for distributed matching (no practical algorithm is available); apply to static pivoting for solving systems of sparse linear equations.
Parallel Algorithms for De Novo Genome Assembly

1. K-mer analysis
2. Graph const. & traversal (contig generation)
3. Scaffolding (alignment)

Improvement for first two steps:
Human: 60 hours → 2 minutes
Wheat: 140 hours → 15 minutes

Parallel Genome Alignment for De novo Assembly

- In de novo assembly, billions of reads must be aligned to contigs
- First aligner to parallelize the seed index construction ("fully" parallel)

Fast Parallel Block Eigensolvers

• Block iterative methods:
  - have data locality and cache re-use
  - expose more parallelism
• Sparse matrix vector block multiplication
• Motivated by nuclear structure calculations
• Applications: data analysis & spectral clustering

**Filtered Semantic Graph Processing**

- **Filters** enable efficient processing of semantic graphs (with edge/vertex attributes)
- **Semirings** enable customization of graph algorithms using matrix primitives
- Filters & semirings can be a performance bottleneck in high-level languages
- SEJITS enable just-in-time compilation of filters/semirings into **Combinatorial BLAS**, matching performance low-level code

<table>
<thead>
<tr>
<th>Filter Permeability</th>
<th>Mean BFS Time (seconds, log scale)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>0.1</td>
</tr>
<tr>
<td>10%</td>
<td>1</td>
</tr>
<tr>
<td>100%</td>
<td>10</td>
</tr>
</tbody>
</table>

Abstract-- It is our view that the state of the art in constructing a large collection of graph algorithms in terms of linear algebraic operations is mature enough to support the emergence of a standard set of primitive building blocks. This paper is a position paper defining the problem and announcing our intention to launch an open effort to define this standard.

- The Graph BLAS Forum: http://istc-bigdata.org/GraphBlas/
Direction-optimizing BFS on distributed memory

Up to 8X faster on Kronecker (Graph500) inputs (largest run on 16 billion vertices and 256 billion edges)

For a fixed-sized real input, direction-optimizing algorithm needs 1/16th of processors (and energy)

Implemented on top of Combinatorial BLAS (i.e. uses 2D decomposition)

Beamer, B., Asanović, Patterson, “Distributed Memory Breadth-First Search Revisited: Enabling Bottom-Up Search”, IPDPSW’13
Communication-avoiding All-Pairs Shortest-Paths

Strong Scaling on Hopper (Cray XE6 up to 1024 nodes = 24576 cores)

Algorithm based on Kleene’s recursive formulation on the (min,+) semiring

A = A*; % recursive call
B = AB; C = CA;
D = D + CB;
D = D*; % recursive call
B = BD; C = DC;
A = A + BC;

Matrix multiplication:
\[
\forall (i,j) \in n \times n, 
C(i,j) = \sum_k A(i,k)B(k,j),
\]

Previous Sparse Classical Lower Bound:
\[
\Omega \left( \frac{\#FLOPs}{(\sqrt{M})^3 \cdot \frac{M}{P}} \right) = \Omega \left( \frac{d^2 n}{P \sqrt{M}} \right)
\]
[Ballard, et al. SIMAX’11]

New Lower bound for Erdős-Rényi(n,d) :
\[
\Omega \left( \min \left\{ \frac{dn}{\sqrt{P}}, \frac{d^2 n}{P} \right\} \right)
\]

- Expected (Under some technical assumptions)
- **Two new algorithms (3D iterative & recursive)** that attain the new lower bound
- No previous algorithm attain these.

---

**Assumption:** Assignment of data and work to processors is sparsity-pattern-independent

---