Large Scale Graph Analysis

Erik Saule

HPC Lab
Biomedical Informatics
The Ohio State University

March 11, 2013
UMass Boston
Outline

1 Introduction

2 the advisor
   - Citation Analysis for Document Recommendation
   - A High Performance Computing Problem
   - Result Diversification

3 Centrality
   - Compression and Shattering
   - Storage format for GPU acceleration
   - Incremental Algorithms

4 Data Management
   - Middleware for Data Analysis
   - Out-of-Core Computing

5 Conclusion
Data in the Modern Days

Facebook

1B active users a month. Each day:
- 2.5B content items shared
- 2.7B Likes
- 300M photos
- 500TB data
## Data in the Modern Days

### Facebook

1B active users a month. Each day:
- 2.5B content items shared
- 2.7B Likes
- 300M photos
- 500TB data

### Twitter

- 500M users
- 340M tweets/day (2,200/sec)
- 24.1M super bowl tweets
### Data in the Modern Days

**Facebook**
- 1B active users a month. Each day:
  - 2.5B content items shared
  - 2.7B Likes
  - 300M photos
  - 500TB data

**Twitter**
- 500M users
- 340M tweets/day (2,200/sec)
- 24.1M super bowl tweets

**Academic networks**
- 1.5M papers/year (4,000/day)
- 100,000 papers/year in CS
Data in the Modern Days

**Facebook**
- 1B active users a month. Each day:
  - 2.5B content items shared
  - 2.7B Likes
  - 300M photos
  - 500TB data

**Twitter**
- 500M users
- 340M tweets/day (2,200/sec)
- 24.1M super bowl tweets

**Academic networks**
- 1.5M papers/year (4,000/day)
- 100,000 papers/year in CS

**Transportation**
- 10M trips in Paris public transportation/day
- 2.5M registered vehicles in LA
- 1.2M used for commuting/day

---

Erik Saule
Ohio State University, Biomedical Informatics
HPC Lab [http://bmi.osu.edu/hpc]
## Data in the Modern Days

### Facebook
- 1B active users a month. Each day:
  - 2.5B content items shared
  - 2.7B Likes
  - 300M photos
  - 500TB data

### Academic networks
- 1.5M papers/year (4,000/day)
- 100,000 papers/year in CS

### Transportation
- 10M trips in Paris public transportation/day
- 2.5M registered vehicles in LA
- 1.2M used for commuting/day

### Twitter
- 500M users
- 340M tweets/day (2,200/sec)
- 24.1M super bowl tweets

### Compositing
Problems can also come from multiple sources, e.g., identify coauthors in Facebook.
“CERN report 1959” about a 1H experiment on the synchrocyclotron

The use of the computer in this sort of measurement is important, not only because of the large amounts of data which must be handled, but because with a modern high speed computer one can search quickly for various systematic errors.
Are these problems new?

“CERN report 1959” about a 1H experiment on the synchrocyclotron

The use of the computer in this sort of measurement is important, not only because of the large amounts of data which must be handled, but because with a modern high speed computer one can search quickly for various systematic errors.

But also...

- Intrusion detection in computer security
- Search engines
- Stock market predictions
- Weather forecast
Are these problems new?

“CERN report 1959” about a 1H experiment on the synchrocyclotron

The use of the computer in this sort of measurement is important, not only because of the large amounts of data which must be handled, but because with a modern high speed computer one can search quickly for various systematic errors.

But also...

- Intrusion detection in computer security
- Search engines
- Stock market predictions
- Weather forecast

Not so new!
So why is it important now?

Ubiquitous

- Scientist (LHC, Metagenomics)
- Big companies (Data companies, Operational marketing)
- Small companies (Website logs, who buys what? where?)
- People (Personal analytics)
So why is it important now?

Ubiquitous

- Scientist (LHC, Metagenomics)
- Big companies (Data companies, Operational marketing)
- Small companies (Website logs, who buys what? where?)
- People (Personal analytics)

In brief, everybody has Big Data problems now!

None of these data can be manually analyzed. Automatic analysis is mandatory.
Variety
unstructured data

Velocity
flowing in the system

Volume
in high volume
## The Three Attributes of Big Data

<table>
<thead>
<tr>
<th>Variety</th>
<th>Velocity</th>
<th>Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>unstructured data</td>
<td>flowing in the system</td>
<td>in high volume</td>
</tr>
<tr>
<td>Graphs</td>
<td>Streaming data</td>
<td>Millions, Billions, Trillions</td>
</tr>
<tr>
<td>Hypergraphs</td>
<td>Temporal data</td>
<td>of vertices and edges</td>
</tr>
<tr>
<td>Conceptual data</td>
<td>Flow of queries</td>
<td></td>
</tr>
</tbody>
</table>

### Problems
- Storing and transporting such data
- Extracting the important data and building a graph (or else)
- Analyzing the graph:
  - static analysis
  - recurrent analysis
  - temporal analysis

---

Erik Saule  
Ohio State University, Biomedical Informatics  
HPC Lab [http://bmi.osu.edu/hpc](http://bmi.osu.edu/hpc)
The Three Attributes of Big Data

**Variety**
- unstructured data

**Velocity**
- flowing in the system

**Volume**
- in high volume
  - Millions,
  - Billions,
  - Trillions
  - of vertices and edges

---

**Graphs**
- Streaming data
- Temporal data
- Flow of queries

**Problems**
- Storing and transporting such data
- Extracting the important data and building a graph (or else)
- Analyzing the graph:
  - static analysis
  - recurrent analysis
  - temporal analysis
My Goal

Study Big Data problems and design solutions for them.
My Goal

Study Big Data problems and design solutions for them.

Applications (Source)

Facebook, the advisor, twitter, CiteULike, traffic camera, transportation systems
My Goal

Study Big Data problems and design solutions for them.

Applications (Source)
Facebook, the advisor, twitter, CiteULike, traffic camera, transportation systems

Algorithms (Analysis)
Page Rank, Random Walk, Traversals, Centrality, Community Detection, Outlier Detection, Visualization
My Goal

Study Big Data problems and design solutions for them.

<table>
<thead>
<tr>
<th>Applications (Source)</th>
<th>Middleware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook, the advisor, twitter, CiteULike, traffic camera, transportation systems</td>
<td>MPI, Hadoop, Pegasus, Graph Lab, DOoC+LAF, DataCutter, SQL, SPARQL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Algorithms (Analysis)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page Rank, Random Walk, Traversals, Centrality, Community Detection, Outlier Detection, Visualization</td>
</tr>
</tbody>
</table>
My Goal

Study Big Data problems and design solutions for them.

Applications (Source)
- Facebook, the advisor, twitter, CiteULike, traffic camera, transportation systems

Middleware
- MPI, Hadoop, Pegasus, Graph Lab, DDoC+LAF, DataCutter, SQL, SPARQL

Algorithms (Analysis)
- Page Rank, Random Walk, Traversals, Centrality, Community Detection, Outlier Detection, Visualization

Hardware
- Clusters, Cray XMT, Intel Xeon Phi, FPGAS, SSD drives, NVRAM, Infiniband, Cloud Computing, GPU.
My Goal

Study Big Data problems and design solutions for them.

Applications (Source)
Facebook, the advisor, twitter, CiteULike, traffic camera, transportation systems

Algorithms (Analysis)
Page Rank, Random Walk, Traversals, Centrality, Community Detection, Outlier Detection, Visualization

Middleware
MPI, Hadoop, Pegasus, Graph Lab, DOoC+LAF, DataCutter, SQL, SPARQL

Hardware
Clusters, Cray XMT, Intel Xeon Phi, FPGAS, SSD drives, NVRAM, Infiniband, Cloud Computing, GPU.

What to use? When to use them? What is missing?
Outline

1. Introduction

2. the advisor
   - Citation Analysis for Document Recommendation
   - A High Performance Computing Problem
   - Result Diversification

3. Centrality
   - Compression and Shattering
   - Storage format for GPU acceleration
   - Incremental Algorithms

4. Data Management
   - Middleware for Data Analysis
   - Out-of-Core Computing

5. Conclusion
A Use Case

http://theadvisor.osu.edu/

1. Put your references in a BibTeX or RIS file

2. Select the file or simply drag-and-drop, adjust the parameter, then submit!

Let's get started!

1. Select a BibTeX (*.bib), RIS (*.ris) or EndNote (*.xml) file:

   Choose File  No file chosen

   optional  Have a bbl file as well?

2. I want papers to be more

   traditional  recent

   I authorize the use of my activity for research purposes.

Get recommendations →

3. Get citation, venue, and reviewer recommendations, give feedback...

- Papers
  - Marina Drosou, Evaggelia Pitoura: Search result diversification. [bib] [good SIGMOD Record, 2010.
  - Cong Yu, Laks V. S. Lakshmanan, Siyuan Recommendation Diversification Using International Conference on Data Eng
A Use Case

http://theadvisor.osu.edu/

Using the Citation Graph

**Hypothesis:** If two papers are related or treat the same subject, then they will be close to each other in the citation graph (and reciprocal)
Let $G = (V, E)$ be the citation graph.

**Personalized PageRank [Haveliwala02]**

$$\pi_i(u) = dp^*(u) + (1 - d) \sum_{v \in N(u)} \frac{\pi_{i-1}(v)}{\delta(v)}$$

with $\sum p^*(u) = 1$. 

source: wikipedia
Direction Awareness [DBRank12]

Time exploration

What if we are interested in searching papers per years. Recent papers? Traditional papers?

Let $Q$ be a set of known relevant papers.

Direction Aware Random Walk with Restart

$$\pi_i(u) = dp^*(u) + (1 - d)(\kappa \sum_{v \in N^+(u)} \frac{\pi_{i-1}(v)}{\delta^-(v)} + (1 - \kappa) \sum_{v \in N^-(u)} \frac{\pi_{i-1}(v)}{\delta^+(v)})$$

- $d \in (0 : 1)$ is the damping factor.
- $\kappa \in (0 : 1)$.
- $p^*(u) = \frac{1}{|Q|}$, if $u \in Q$, $p^*(u) = 0$, otherwise

Erik Saule  
Ohio State University, Biomedical Informatics  
HPC Lab http://bmi.osu.edu/hpc  
Large Scale Graph Analysis  
theadvisor::Citation Analysis  11 / 43
### Analysis: Time and Accuracy

The diagram illustrates the relationship between the parameter $\kappa$ and the publication year, showing how different values of $\kappa$ affect the accuracy of citation analysis over time.

### Table: Analysis Results

<table>
<thead>
<tr>
<th>Method</th>
<th>hide random</th>
<th>hide recent</th>
<th>hide earlier</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mean</td>
<td>interval</td>
<td>mean</td>
</tr>
<tr>
<td>DARWR</td>
<td>48.00</td>
<td>46.80 49.20</td>
<td>42.22</td>
</tr>
<tr>
<td>P.R.</td>
<td>56.56</td>
<td>55.31 57.80</td>
<td>38.75</td>
</tr>
<tr>
<td>Katz$_\beta$</td>
<td>46.33</td>
<td>45.16 47.50</td>
<td>34.56</td>
</tr>
<tr>
<td>Cocit</td>
<td>44.60</td>
<td>43.39 45.80</td>
<td>14.22</td>
</tr>
<tr>
<td>Cocoup</td>
<td>17.28</td>
<td>16.36 18.20</td>
<td>17.56</td>
</tr>
<tr>
<td>CCIDF</td>
<td>18.05</td>
<td>17.11 19.00</td>
<td>18.97</td>
</tr>
</tbody>
</table>
A Sparse Matrix-Vector Multiplication (SpMV)

Rewriting DaRWWR

\[
\pi_i(u) = dp^*(u) + (1 - d) \left( \kappa \sum_{v \in N^+(u)} \frac{\pi_{i-1}(v)}{\delta^-(v)} + (1 - \kappa) \sum_{v \in N^-(u)} \frac{\pi_{i-1}(v)}{\delta^+(v)} \right)
\]

\[
\pi_i(u) = dp^*(u) + \sum_{v \in N^+(u)} \frac{(1 - d)\kappa}{\delta^-(v)} \pi_{i-1}(v) + \sum_{v \in N^-(u)} \frac{(1 - d)(1 - \kappa)}{\delta^+(v)} \pi_{i-1}(v)
\]

\[
\pi_i = dp^* + A^- \pi_{i-1} + A^+ \pi_{i-1} \quad \text{(CRS Full)}
\]

\[
\pi_i = dp^* + A \pi_{i-1} \quad \text{(CRS Half)}
\]

\[
\pi_i = dp^* + B^- \left( \frac{(1 - d)\kappa}{\delta^-} \pi_{i-1} \right) + B^+ \left( \frac{(1 - d)(1 - \kappa)}{\delta^+} \pi_{i-1} \right)
\]
Partitioning and Ordering
Diversification: Principle

\[\text{Diagram showing diverse symbols} \]
Diversification: Principle

Relevant
Diversification: Principle

Relevant

Relevant Diverse
k-RLM is good.
Results
Here is a distribution of known algorithms.
Here is a distribution of known algorithms. Would such an algorithm be of interest?
Here is a distribution of known algorithms. Would such an algorithm be of interest? That algorithm is query-oblivious!

**Expanded Relevance**

Sum the relevance of all documents at distance \( \ell \) of a recommendation.
Outline

1 Introduction

2 The Advisor
   - Citation Analysis for Document Recommendation
   - A High Performance Computing Problem
   - Result Diversification

3 Centrality
   - Compression and Shattering
   - Storage format for GPU acceleration
   - Incremental Algorithms

4 Data Management
   - Middleware for Data Analysis
   - Out-of-Core Computing

5 Conclusion
Centralities - Concept

Answer questions such as

- Who controls the flow in a network?
- Who is more important?
- Who has more influence?
- Whose contribution is significant for connections?

Applications

- Covert network (e.g., terrorist identification)
- Contingency analysis (e.g., weakness/robustness of networks)
- Viral marketing (e.g., who will spread the word best)
- Traffic analysis
- Store locations
Centralities - Definition

Let $G = (V, E)$ be a graph with the vertex set $V$ and edge set $E$.

- **closeness centrality**: $cc[v] = \frac{1}{\text{far}[v]}$, where the farness is defined as $\text{far}[v] = \sum_{u \in \text{comp}(v)} d(u, v)$. $d(u, v)$ is the shortest path length between $u$ and $v$.

- **betweenness centrality**: $bc(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where $\sigma_{st}$ is the number shortest paths between $s$ and $t$, and $\sigma_{st}(v)$ is the number of them passing through $v$. 

Brandes algorithm computes the shortest path graph rooted in each vertex of the graph. $O(|E|)$ per source. $O(|V||E|)$ in total. Believed to be asymptotically optimal [Kintali08].
Let $G = (V, E)$ be a graph with the vertex set $V$ and edge set $E$.

- **closeness centrality**: $cc[v] = \frac{1}{far[v]}$, where the farness is defined as $far[v] = \sum_{u \in \text{comp}(v)} d(u, v)$. $d(u, v)$ is the shortest path length between $u$ and $v$.

- **betweenness centrality**: $bc(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$, where $\sigma_{st}$ is the number shortest paths between $s$ and $t$, and $\sigma_{st}(v)$ is the number of them passing through $v$.

Both metrics care about the structure of the shortest path graph. Brandes algorithm computes the shortest path graph rooted in each vertex of the graph. $O(|E|)$ per source. $O(|V||E|)$ in total. Believed to be asymptotically optimal [Kintali08].
The image shows a graph comparing the relative time taken by different algorithms on various datasets. The x-axis represents the datasets: Epinions, Gowalla, bcsstk32, NotreDame, RoadPA, Amazon0601, Google, and WikiTalk. The y-axis represents the relative time, with phase 1, phase 2, and preprocessing stages indicated. The graph highlights the computational efficiency of BADIOS, an algorithm designed for large-scale graph analysis, particularly in the context of biomedical informatics research at Ohio State University's HPC Lab. The dataset names and their corresponding runtimes are indicated in the image.
Matrix Representations for GPUs

CRS [Shi11]
1 thread per vertex: bad load balance
ptrs 1 2 10 12 15 18 21 24 27 29 35
adj 2 1 3 4 5 6 7 8 9 ...
adj 2 1 3 4 5 6 7 8 9...
Memory: n + m + 1

Virtual-vertex
Balances load and limits atomics
vmap 1 2 2 3 4 5 6 7 8 9 10 10
vptrs 1 2 6 10 12 15 18 21 24 27 29 33 35
adj 2 1 3 4 5 6 7 8 9 ...
adj 2 1 3 4 5 6 7 8 9 ...
Memory: 2n' + m + 1

COO [Jia11]
1 thread per edge: too many atomics
is 1 2 2 2 2 2 2 ... 9 9 10 10 10 10 10
adj 2 1 3 4 5 6 7 8 9 ...
adj 2 1 3 4 5 6 7 8 9...
Memory: 2m

Stride
Enables coalesced memory accesses
offset 0 0 1 0 0 0 0 0 0 0 1
vmap 1 2 2 3 4 5 6 7 8 9 10 10
nvr 1 2 1 1 1 1 1 1 1 2
ptrs 1 2 10 12 15 18 21 24 27 29 35
adj 2 1 3 4 5 6 7 8 9 ...
adj 2 1 3 4 5 6 7 8 9 ...
Memory: 2n' + n + m + 1
NVIDIA C2050 performance [GPGPU2013]

![Bar chart showing speedup with respect to CPU for different graph analysis tasks on NVIDIA C2050 GPU.](image)
If \( d(u, s) = d(v, s) \)

The shortest path graph does not differ. So the farness of \( s \) is correct.
Edge Insertion for closeness centrality: three cases

If $d(u, s) = d(v, s)$

The shortest path graph does not differ. So the farness of $s$ is correct.

If $d(u, s) + 1 = d(v, s)$

The shortest path graph differs by exactly one edge. The levels stay the same. So the farness of $s$ is still correct.
Edge Insertion for closeness centrality: three cases

If $d(u, s) = d(v, s)$

The shortest path graph does not differ. So the farness of $s$ is correct.

If $d(u, s) + 1 = d(v, s)$

The shortest path graph differs by exactly one edge. The levels stay the same. So the farness of $s$ is still correct.

If $d(u, s) + 1 < d(v, s)$

The shortest path graph differs by at least one edge. The level of $v$ changes (and potentially more). So the farness of $s$ is incorrect.
**Edge Insertion for closeness centrality: three cases**

<table>
<thead>
<tr>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d(u, s) = d(v, s)$</td>
<td>The shortest path graph does not differ. So the farness of $s$ is correct.</td>
</tr>
<tr>
<td>$d(u, s) + 1 = d(v, s)$</td>
<td>The shortest path graph differs by exactly one edge. The levels stay the same. So the farness of $s$ is still correct.</td>
</tr>
<tr>
<td>$d(u, s) + 1 &lt; d(v, s)$</td>
<td>The shortest path graph differs by at least one edge. The level of $v$ changes (and potentially more). So the farness of $s$ is incorrect.</td>
</tr>
</tbody>
</table>

**Algorithm**

- Upon insertion of $(u, v)$
- Compute BFS from $u$ and $v$ (before edge insertion)
- For all $s \neq u, v$, if $|d(u, s) - d(v, s)| > 1$, flag $s$
- Add $(u, v)$ to the graph
- Compute $cc[s]$ for all flagged $s$
## Results: Speedup

<table>
<thead>
<tr>
<th>Graph</th>
<th>CC-B</th>
<th>CC-BL</th>
</tr>
</thead>
<tbody>
<tr>
<td>soc-sign-epinions</td>
<td>3.0</td>
<td>37.8</td>
</tr>
<tr>
<td>loc-gowalla_edges</td>
<td>1.8</td>
<td>17.1</td>
</tr>
<tr>
<td>bcsstk32</td>
<td>1.0</td>
<td>5,493.0</td>
</tr>
<tr>
<td>web-NotreDame</td>
<td>4.9</td>
<td>23.9</td>
</tr>
<tr>
<td>roadNet-PA</td>
<td>1.6</td>
<td>3.0</td>
</tr>
<tr>
<td>amazon0601</td>
<td>1.2</td>
<td>27.6</td>
</tr>
<tr>
<td>web-Google</td>
<td>3.0</td>
<td>26.6</td>
</tr>
<tr>
<td>wiki-Talk</td>
<td>6.8</td>
<td>69.8</td>
</tr>
<tr>
<td>Geometric mean</td>
<td>2.39</td>
<td>43.58</td>
</tr>
</tbody>
</table>

Erik Saule
Ohio State University, Biomedical Informatics HPC Lab http://bmi.osu.edu/hpc

Large Scale Graph Analysis
Centrality::Incremental 28 / 43
Outline

1 Introduction

2 the advisor
   - Citation Analysis for Document Recommendation
   - A High Performance Computing Problem
   - Result Diversification

3 Centrality
   - Compression and Shattering
   - Storage format for GPU acceleration
   - Incremental Algorithms

4 Data Management
   - Middleware for Data Analysis
   - Out-of-Core Computing

5 Conclusion
Data Flow Middleware

Applications
- Image processing
- Video surveillance
- MRI Analysis
- Satellite data processing

Complex Hardware
- Accelerators (GPU, Xeon Phi)
- Clusters
- Grid
- Heterogeneous systems

Versatile [CSUR13]
Supports:
- Pipeline parallelism
- Task parallelism
- Replicated parallelism
- Data parallelism

Optimizes:
- Throughput
- Latency
- Energy
- Reliability
Filter Stream

Programming model that specify the *layout* of an application: a set filters that transform the data streamed through them.

Placement

A given layout can be executed in different ways by the programming framework. Potentially filters can be *replicated*. 
How much to cut the work?
- Small chunks, pay overhead
- Big chunks, pay imbalance
- Adapt to the network
- Adapt to the computing units

Performance Model
Predicts computation time for all size and processing unit.

Workload Partitioner[HiPC10]
Predicts the end of the computation to balance the load.

Distributed Work Stealing
Balances the load across multiple node.

Storage Layer
Schedule data transfers to optimize network performance.
SAR imaging - Weak Scaling - CPU/GPU [Parco12]
A Peta Scale nuclear physics problem

Extract the lowest eigenpairs of a large Hamiltonian matrix, whose size grows with the number of particles and truncation parameter in the atom. For Boron 10, with $N_{\text{max}}=8$ with 2 body interactions (Toy case): 160 millions of rows, 124 billions of non zero elements.
A Peta Scale nuclear physics problem

Extract the lowest eigenpairs of a large Hamiltonian matrix, whose size grows with the number of particles and truncation parameter in the atom. For Boron 10, with $N_{\text{max}}=8$ with 2 body interactions (Toy case): 160 millions of rows, 124 billions of non zero elements.

Two options: use a really large machine or use Out-of-Core (SSD).
LOBPCG
End-User Code
…
SymSpMM(H, psi)
dot(\phi^T, \phi)
…

LOBPCG.cpp

Primitive Conversion

Global Task Graph

Global Scheduler

DOoC

Compute Node - 1

Local Scheduler

SpMM

Storage Service

Data
Chunks

Out Data

In Data

Exec

Compute Node - 2

Local Scheduler

SpMM

Storage Service

Data
Chunks

Out Data

In Data

Exec

Compute Node - 3

Local Scheduler

SpMM

Storage Service

Data
Chunks

Out Data

In Data

Exec

Global Task Graph

Global Scheduler

Erik Saule
Ohio State University, Biomedical Informatics
HPC Lab http://bmi.osu.edu/hpc
Large Scale Graph Analysis
Data Management::Out-of-Core 35 / 43
Linear Algebra Frontend

Provides data types and operations for Linear Algebra.

Lanczos

Lanczos(v_in, M, a_in, b_in, v_out, a_out, b_out) {
Vector w(out.meta());
Vector wprime(out.meta());
Vector wsecond(out.meta());
symSpMV(w, M, v_in);
axpyV(wprime, w, v_in, 1, -b_in);
dot(a_out, wprime, v_in);
axpyV(wsecond, wprime, v_in, 1, -a_out);
dot(b_out, wsecond, wsecond);
vector_scale(v_out, wsecond, 1/b_out);
}

Supported Operations

<table>
<thead>
<tr>
<th>Primitives</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primitives that creates Matrix</td>
<td></td>
</tr>
<tr>
<td>MM, (Sym)SpMM</td>
<td>( C = AB )</td>
</tr>
<tr>
<td>addM</td>
<td>( C = A + B )</td>
</tr>
<tr>
<td>axpyM</td>
<td>( C = aA + b )</td>
</tr>
<tr>
<td>randomM</td>
<td>( C = \text{random}() )</td>
</tr>
<tr>
<td>Primitives that creates Vector</td>
<td></td>
</tr>
<tr>
<td>MV, (Sym)SpMV</td>
<td>( y = Ax )</td>
</tr>
<tr>
<td>addV</td>
<td>( y = x + w )</td>
</tr>
<tr>
<td>axpyV</td>
<td>( y = ax + b )</td>
</tr>
<tr>
<td>Primitives that creates scalar</td>
<td></td>
</tr>
<tr>
<td>dot</td>
<td>( a = \langle x, y \rangle )</td>
</tr>
</tbody>
</table>
5 Lanczos iterations at NERSC [Cluster12]

Erik Saule
Ohio State University, Biomedical Informatics
HPC Lab http://bmi.osu.edu/hpc

Large Scale Graph Analysis
Data Management::Out-of-Core
Outline

1. Introduction
2. The advisor
   - Citation Analysis for Document Recommendation
   - A High Performance Computing Problem
   - Result Diversification
3. Centrality
   - Compression and Shattering
   - Storage format for GPU acceleration
   - Incremental Algorithms
4. Data Management
   - Middleware for Data Analysis
   - Out-of-Core Computing
5. Conclusion
Other things I do

Scheduling, Mapping, Partitioning

Areas:
- Application scheduling
- Cluster scheduling
- Pipelined scheduling
- Spatial workload partitioning

Multi objective:
- Makespan
- Throughput
- Fairness
- Latency
- Reliability

Techniques:
- Optimal algorithms
- Approximation algorithms
- Heuristics
### Scheduling, Mapping, Partitioning

**Areas:**
- Application scheduling
- Cluster scheduling
- Pipelined scheduling
- Spatial workload partitioning

**Multi objective:**
- Makespan
- Throughput
- Fairness
- Latency
- Reliability

**Techniques:**
- Optimal algorithms
- Approximation algorithms
- Heuristics

### Parallel Graph Algorithms

Scalable distributed memory local search for graph coloring. Communication reductions and compression. Hybrid MPI/OpenMP.

---

Erik Saule
Ohio State University, Biomedical Informatics
HPC Lab [http://bmi.osu.edu/hpc](http://bmi.osu.edu/hpc)

Large Scale Graph Analysis
Conclusion: 39 / 43
Other things I do

Scheduling, Mapping, Partitioning

Areas:
- Application scheduling
- Cluster scheduling
- Pipelined scheduling
- Spatial workload partitioning

Multi objective:
- Makespan
- Throughput
- Fairness
- Latency
- Reliability

Techniques:
- Optimal algorithms
- Approximation algorithms
- Heuristics

Parallel Graph Algorithms

Scalable distributed memory local search for graph coloring. Communication reductions and compression. Hybrid MPI/OpenMP.

Cutting Edge Architecture

Investigated graph algorithms and sparse linear algebra operations on pre-release Intel Xeon Phi.

Erik Saule
Ohio State University, Biomedical Informatics
HPC Lab http://bmi.osu.edu/hpc

Large Scale Graph Analysis
Conclusion: 39 / 43
Conclusions - My Philosophy

Applications

- Analyze data sources
- What are we trying to do?
- What is important?
Conclusions - My Philosophy

Applications
- Analyze data sources
- What are we trying to do?
- What is important?

Algorithms
- Design
- Re-engineer
- Approximate
- Incremental
### Conclusions - My Philosophy

#### Applications
- Analyze data sources
- What are we trying to do?
- What is important?

#### Algorithms
- Design
- Re-engineer
- Approximate
- Incremental

#### Middleware
- Makes the software:
  - Easier to write
  - Reusable
  - Efficient
Applications

- Analyze data sources
- What are we trying to do?
- What is important?

Middleware

Makes the software:
- Easier to write
- Reusable
- Efficient

Algorithms

- Design
- Re-engineer
- Approximate
- Incremental

Hardware

- What is suitable?
- How to use it?
- How to improve it?
Applications
- Analyze data sources
- What are we trying to do?
- What is important?

Middleware
- Makes the software:
  - Easier to write
  - Reusable
  - Efficient

Algorithms
- Design
- Re-engineer
- Approximate
- Incremental

Hardware
- What is suitable?
- How to use it?
- How to improve it?

Which is important? All of it!
What’s Next?

Applications
- Multi-graph
- Author Venue Paper
- Personal analytics
- Cross social network application

Middleware
- High Level Query
- Cluster with Accelerator Graph Middleware
  - The MATLAB of graphs

Algorithms
- Streaming
- Community detection
  - Temporal analysis

Hardware
- Cluster with Computational Accelerator (GPU, Xeon Phi)
- Cluster with Storage Accelerator (SSD)
- Both! (Beacon project)
Active Coauthors

The Ohio State University:
- Ümit V. Çatalyürek
- Kamer Kaya
- Onur Küçüktunç
- Ahmet Erdem Saryüce

Grenoble University, France:
- Denis Trystram
- Grégory Mounié
- Pierre-François Dutot
- Jean-François Mehaut
- Guillaume Huard

INRIA, France:
- Yves Robert
- Anne Benoit
- Emmanuel Jeannot
- Alain Girault

Lawrence Berkely National Lab:
- Esmond G. Ng
- Chao Yang
- Hasan Metin Aktulga

University of Tennessee:
- Jack Dongarra

University of Luxembourg:
- Johnatan Pecero-Sanchez

Vanderbilt University:
- Zhiao Shi

Iowa State University:
- James Vary
- Pieter Maris
More information

contact: esaule@bmi.osu.edu
visit: http://bmi.osu.edu/~esaule
http://bmi.osu.edu/hpc/

Research at HPC lab is supported by

[Logos of sponsoring organizations]