Large Scale Graph Analysis

Erik Saule

HPC Lab
Biomedical Informatics
The Ohio State University

February 19, 2013
UNC Charlotte
Facebook

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

<table>
<thead>
<tr>
<th>Social Media</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Facebook</strong></td>
<td>1B active users a month. Each day:</td>
</tr>
<tr>
<td></td>
<td>- 2.5B content items shared</td>
</tr>
<tr>
<td></td>
<td>- 2.7B Likes</td>
</tr>
<tr>
<td></td>
<td>- 300M photos</td>
</tr>
<tr>
<td></td>
<td>- 500TB data</td>
</tr>
<tr>
<td><strong>Twitter</strong></td>
<td>500M users</td>
</tr>
<tr>
<td></td>
<td>- 340M tweets/day (2,200/sec)</td>
</tr>
<tr>
<td></td>
<td>- 24.1M super bowl tweets</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Academic networks</strong></th>
<th>1.5M papers/year (4,000/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100,000 papers/year in CS</td>
</tr>
</tbody>
</table>

---

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

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
### Data in the Modern Days

<table>
<thead>
<tr>
<th>Facebook</th>
<th>Academic networks</th>
<th>Transportation</th>
<th>Compositing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1B active users a month. Each day:</td>
<td></td>
<td></td>
<td>Problems can also come from multiple sources, e.g., identify coauthors in Facebook.</td>
</tr>
<tr>
<td>- 2.5B content items shared</td>
<td>1.5M papers/year (4,000/day)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 2.7B Likes</td>
<td>100,000 papers/year in CS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 300M photos</td>
<td></td>
<td>10M trips in Paris public transportation/day</td>
<td></td>
</tr>
<tr>
<td>- 500TB data</td>
<td></td>
<td>2.5M registered vehicles in LA</td>
<td></td>
</tr>
<tr>
<td>Twitter</td>
<td></td>
<td>1.2M used for commuting/day</td>
<td></td>
</tr>
<tr>
<td>- 500M users</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 340M tweets/day (2,200/sec)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- 24.1M super bowl tweets</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

- **Variety**: unstructured data
- **Velocity**: flowing in the system
- **Volume**: in high volume

---

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)
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>
## 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 of vertices and edges</td>
</tr>
<tr>
<td>Hypergraphs</td>
<td>Temporal data</td>
<td></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
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

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

Large Scale Graph Analysis
Introduction: 7 / 39
# 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
- DOoC+LAF
- DataCutter
- SQL
- SPARQL

## Algorithms (Analysis)

- Page Rank
- Random Walk
- Traversals
- Centrality
- Community Detection
- Outlier Detection
- Visualization
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, DDoC+LAF, DataCutter, SQL, SPARQL

### 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. 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

5. Conclusion
A Use Case

http://theadvisor.osu.edu/

1. Put your references in a BibTeX or RIS file
   - @inproceedings{Kucuktunc12,
     author = {Kucuktunc, Onur and Yilmaz, Baris},
     title = {A Large-Scale Sentence Similarity Measure},
     booktitle = {Proc. 5th Int'l Conf. on Document Analysis and Recognition},
     year = {2012},
   }
   - @inproceedings{weber10,
     author = {Weber, Ingo},
     title = {The demographics of the best Web 2.0 tools},
     booktitle = {Proc. 33rd Int'l Conf. on Database Systems},
     year = {2010},
   }
   - @article{Aral11,
     author = {Aral, Sinan and Varian, Hal},
     title = {The network of scientific collaboration},
     journal = {Science},
     volume = {332},
     number = {6027},
     pages = {777-781},
     year = {2011},
   }

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

2. I want papers to be more
   - traditional
   - recent

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

- Papers
  - Marina Drosou, Evaggelia Pitoura: Search result diversification. [bib] [good]
    SIGMOD Record, 2010.
  - Marina Drosou, Evaggelia Pitoura: Diversity over Continuous Data. [bib] [good]
  - Cong Yu, Laks V. S. Lakshmanan, Sihem Benseddik: Recommendation Diversification Using
    Collaborative Filtering. [bib] [good]
    Proceedings of the 32nd International Conference on Data Engineering, 2016.
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
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
Analysis: Time and Accuracy

The diagram shows a color-coded map indicating the relationship between time (average publication year) and accuracy (κ). The axes are labeled with years ranging from 1980 to 2010.

The table below lists the means and intervals for different measures of accuracy:

<table>
<thead>
<tr>
<th>Measure</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>Katzβ</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>

Erik Saule  Ohio State University, Biomedical Informatics HPC Lab http://bmi.osu.edu/hpc
A Sparse Matrix-Vector Multiplication (SpMV)

**Rewriting DARWR**

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

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

\[ \pi_i = dp^* + A^- \pi_{i-1} + A^+ \pi_{i-1} \]  
\[ \pi_i = dp^* + A \pi_{i-1} \]  
\[ \pi_i = dp^* + B^- \left( \frac{(1 - d)(1 - \kappa)}{\delta^-} \pi_{i-1} \right) + B^+ \left( \frac{(1 - d)(1 - \kappa)}{\delta^+} \pi_{i-1} \right) \]  

\( \pi \) = (CRS Full)  
\( \pi \) = (CRS Half)
Partitioning and Ordering

Large Scale Graph Analysis

the advisor::A HPC computing problem
Runtimes - AMD Opteron 2378 [ASONAM12]

![Graph showing execution time vs. #partitions for different storage formats and partitioning strategies.]

- CRS-Full
- CRS-Full (RCM)
- CRS-Full (AMD)
- CRS-Full (SB)
- CRS-Half
- CRS-Half (RCM)
- CRS-Half (AMD)
- CRS-Half (SB)
- COO-Half
- COO-Half (RCM)
- COO-Half (AMD)
- COO-Half (SB)
- Hybrid
- Hybrid (RCM)
- Hybrid (AMD)
- Hybrid (SB)
Diversification: Principle

![Graphs](image)
Diversification: Principle

Relevant
Diversification: Principle

Relevant

Relevant Diverse
**k-RLM is good.**
Results

Graph mining

Generic SpMV

Multicore

Compression

Partitioning

Eigensolvers

references

recommendations
top-100

Large Scale Graph Analysis
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.
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

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
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 $v$. 
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 $v$.

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].
Compression and Shattering

Large Scale Graph Analysis
Centrality::Shattering
Matrix Representations for GPUs

**CRS [Shi11]**

- **ptrs**: 1 2 10 12 15 18 21 24 27 29 35
- **adjs**: 2 1 3 4 5 6 7 8 9 ...

1 thread per vertex: bad load balance

**Virtual-vertex**

- **vmap**: 1 2 6 10 12 15 18 21 24 27 33 35
- **vptrs**: 1 2 10 12 15 18 21 24 27 33 35
- **adjs**: 2 1 3 4 5 6 7 8 9 ...

Balances load and limits atomics

**COO [Jia11]**

- **is**: 1 2 2 2 2 2 2 2 2 2 ...
- **adjs**: 2 1 3 4 5 6 7 8 9 ...

1 thread per edge: too many atomics

**Stride**

- **offset**: 0 0 1 0 0 0 0 0 0 0 1
- **vmap**: 1 2 3 4 5 6 7 8 9 10 10
- **nvir**: 1 2 1 1 1 1 1 1 1 1 2
- **ptrs**: 1 2 10 12 15 18 21 24 27 35
- **adjs**: 2 1 3 4 5 6 7 8 9 ...

Enables coalesced memory accesses

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

Large Scale Graph Analysis
Centrality::GPU 25 / 39
NVIDIA C2050 performance [GPGPU2013]

<table>
<thead>
<tr>
<th>Dataset</th>
<th>GPU vertex</th>
<th>GPU edge</th>
<th>GPU virtual</th>
<th>GPU stride</th>
</tr>
</thead>
<tbody>
<tr>
<td>amazon0601</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>com-orkut</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>loc-gowalla</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>soc-LiveJournal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>soc-sign-opinions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>web-Google</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>web-NotreDame</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wiki-Talk</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Large Scale Graph Analysis
Centrality::GPU
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.
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

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.

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><strong>Geometric mean</strong></td>
<td>2.39</td>
<td>43.58</td>
</tr>
</tbody>
</table>
Outline

1 Introduction

2 theadvisor
   - 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

5 Conclusion
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).

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

Large Scale Graph Analysis
Data Management:: 30 / 39
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 Nmax=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).

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

Large Scale Graph Analysis
Data Management:: 30 / 39
DOoC + LAF

LOBPCG
End-User Code

LOBPCG.cpp

... SymSpMM(H, psi) dot(phiT, phi) ...

DoC

LAF

Global Task Graph

Global Scheduler

Local Scheduler

Compute Node - 1

Storage Service

Data Chunks

SpMM

dot

In

Out

Data

Data

Data

Exec

Req Data

Compute Node - 2

Storage Service

Data Chunks

SpMM

dot

In

Out

Data

Data

Data

Exec

Req Data

Compute Node - 3

Storage Service

Data Chunks

SpMM

dot

In

Out

Data

Data

Data

Exec

Req Data

Global Task Graph

Global Scheduler

Local Scheduler

End-User Code

SymSpMM(H, psi) dot(phiT, phi)

...
Provides data types and operations for Linear Algebra.

**Lanczos**

```cpp
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 that creates Matrix</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<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 = random()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primitives that creates Vector</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Primitives that creates scalar</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>dot</td>
<td>a = &lt; x, y &gt;</td>
</tr>
</tbody>
</table>
5 Lanczos iterations at NERSC [Cluster12]

![Graph showing time in seconds for different node counts.](image)

- **Total time**
- **Max IO time**
- **Max comp time**
- **Graph handling**

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

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
Other things I do

<table>
<thead>
<tr>
<th>Scheduling, Mapping, Partitioning</th>
<th>Parallel Graph Algorithms</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Areas:</strong></td>
<td>Scalable distributed memory local search for graph coloring. Communication reductions and compression. Hybrid MPI/OpenMP.</td>
</tr>
<tr>
<td>- Application scheduling</td>
<td></td>
</tr>
<tr>
<td>- Cluster scheduling</td>
<td></td>
</tr>
<tr>
<td>- Pipelined scheduling</td>
<td></td>
</tr>
<tr>
<td>- Spatial workload partitioning</td>
<td></td>
</tr>
<tr>
<td><strong>Multi objective:</strong></td>
<td></td>
</tr>
<tr>
<td>- Makespan</td>
<td></td>
</tr>
<tr>
<td>- Throughput</td>
<td></td>
</tr>
<tr>
<td>- Fairness</td>
<td></td>
</tr>
<tr>
<td>- Latency</td>
<td></td>
</tr>
<tr>
<td>- Reliability</td>
<td></td>
</tr>
<tr>
<td><strong>Techniques:</strong></td>
<td></td>
</tr>
<tr>
<td>- Optimal algorithms</td>
<td></td>
</tr>
<tr>
<td>- Approximation algorithms</td>
<td></td>
</tr>
<tr>
<td>- Heuristics</td>
<td></td>
</tr>
</tbody>
</table>
## 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

[http://bmi.osu.edu/hpc](http://bmi.osu.edu/hpc)

Large Scale Graph Analysis

Conclusion: 35 / 39
Other things I do

Scheduling, Mapping, Partitioning

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

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

Dataflow middleware

Auto tuning component for spatial divisible workload on heterogeneous systems.
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
## 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
## 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?
# Conclusions - My Philosophy

## 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