Big Data Everywhere

How do we understand and use Big Data?
Data Analytics

1.11 Billion Users

6 Billion Photos

100 Hrs of Video every minute

400 Million Tweets/day

Machine Learning and Data Mining
It’s about the graphs ...
NUMA & Graph-analytics

Application

Data Analytics

YouTube  Gmail  facebook

Hardware

Processor

Multi-Core

Multi-Socket

Memory

Unified

NUCA

Now

e.g. 80 Cores with 1TB RAM

8 Sockets X (10 Cores with 128GB local RAM)

NLP  Graph-analytics
How about NUMA systems + Graph-analytics?
Contribution

Polymer: NUMA-aware Graph-structured Analytics

- A comprehensive analysis that uncovers issues for running graph analytics system on NUMA platform
- A new system that exploits both NUMA-aware data layout and memory access strategies
- Three optimizations for global synchronization efficiency, load balance and data structure flexibility
- A detailed evaluation that demonstrates the performance and scalability benefits
Outline

Background & Issues
Design of Polymer
Evaluation
Outline

Background & Issues

Design of Polymer

Evaluation
Example: PageRank

A centrality analysis algorithm to measure the relative rank for each element of a linked set

Characteristics

- Linked set \( \rightarrow \) data dependence
- Rank of who links it \( \rightarrow \) predictable accesses
- Convergence \( \rightarrow \) iterative computation

\[
R_i = \alpha + (1 - \alpha) \sum_{(j, i) \in E} \omega_{ij} R_j
\]
The **scatter-gather** model

- **“scatter”**: propagate the current value of a vertex to its neighbors along edges
- **“gather”**: accumulate values from neighbors to compute the next value of a vertex

**In-memory data structure**

- Graph **Topology**
- Application-specific **Data**
- Runtime **State**
Vertex-centric (e.g. Ligra)

STAT/curr

TOPO/vertex

TOPO/out-edge

DATA/curr

DATA/next

STAT/next
Vertex-centric (e.g. Ligra)
Edge-centric (e.g. X-Stream)

- **TOPO/edge**: Shows the topology of the edges.
- **STAT/curr**: Current status.
- **DATA/curr**: Current data.
- **DATA/Uout**: Output data.
- **DATA/Uin**: Input data.
- **DATA/next**: Next data.
- **STAT/next**: Next status.

The diagram illustrates the shuffle phase and partitioning process.
Edge-centric (e.g. X-Stream)
NUMA Characteristics

A commodity **NUMA** machine

- Multiple **processor** nodes (i.e., socket)
- Processor = multiple **cores** + a local DRAM
- A globally **shared** memory abstract (cache-coherence)
- Hallmark: **Non-uniform** memory access

### Latency (Cycle)

<table>
<thead>
<tr>
<th>Inst.</th>
<th>0-hop</th>
<th>1-hop</th>
<th>2-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>80-core Intel Xeon machine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load</td>
<td>117</td>
<td>271</td>
<td>372</td>
</tr>
<tr>
<td>Store</td>
<td>108</td>
<td>304</td>
<td>409</td>
</tr>
</tbody>
</table>

### Bandwidth (MB/s)

<table>
<thead>
<tr>
<th>Access</th>
<th>0-hop</th>
<th>1-hop</th>
<th>2-hop</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>80-core Intel Xeon machine</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>3207</td>
<td>2455</td>
<td><strong>2101</strong></td>
<td>2333</td>
</tr>
<tr>
<td>RAND</td>
<td><strong>720</strong></td>
<td>348</td>
<td><strong>307</strong></td>
<td>344</td>
</tr>
</tbody>
</table>
**NUMA Characteristics**

A commodity **NUMA** machine

- Multiple processor nodes (i.e., socket)
- Processor = multiple cores + a local DRAM
- A globally shared memory abstract (cache-coherence)
- **Hallmark:** Non-uniform memory access & Random remote access is **awesome!**

Sequential remote access is faster than random local access

**Latency (Cycle)**

<table>
<thead>
<tr>
<th>Inst.</th>
<th>0-hop</th>
<th>1-hop</th>
<th>2-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>117</td>
<td>271</td>
<td>372</td>
</tr>
<tr>
<td>Store</td>
<td>108</td>
<td>304</td>
<td>409</td>
</tr>
</tbody>
</table>

**Bandwidth (MB/s)**

<table>
<thead>
<tr>
<th>Access</th>
<th>0-hop</th>
<th>1-hop</th>
<th>2-hop</th>
<th>IL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQ</td>
<td>3207</td>
<td>2455</td>
<td><strong>2101</strong></td>
<td>2333</td>
</tr>
<tr>
<td>RAND</td>
<td><strong>720</strong></td>
<td>348</td>
<td>307</td>
<td>344</td>
</tr>
</tbody>
</table>
NUMA Characteristics

The world we lived in: “first-touch” policy

“binding virtual pages to physical frames locating on a memory node where a thread first touches the pages”

<table>
<thead>
<tr>
<th></th>
<th>Centralized</th>
<th>Interleaved</th>
<th>Associated</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>![CPU Centralized]</td>
<td>![CPU Interleaved]</td>
<td>![CPU Associated]</td>
</tr>
<tr>
<td>MEM</td>
<td>![MEM Centralized]</td>
<td>![MEM Interleaved]</td>
<td>![MEM Associated]</td>
</tr>
</tbody>
</table>

The world we want to lived in
NUMA Characteristics

The world we lived in: “first-touch” policy

“binding virtual pages to physical frames locating on a memory node where a thread first touches the pages”

Both centralized and interleaved data layout will hamper locality and parallelism & Associated layout is the ideal one.

The world we want to lived in
Lack of *locality* (access neighboring vertices)

- It is inevitable to access remote memory
- Random access is always there

How to mix ??

<table>
<thead>
<tr>
<th></th>
<th>SEQ</th>
<th>RND</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Global</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

update
Access Strategy on NUMA

**Vertex-centric Model**

- Completely overlooked (e.g. Ligra)

![Diagram showing vertex-centric model and data access]

Data/curr:
- N0: $D_1, D_2, D_3, D_4, D_5, D_6$
- N1: $D_1, D_2, D_3, D_4, D_5, D_6$

Data/next:
- N0: X X
- N1: X X

**Update**
Access Strategy on NUMA

Edge-centric Model

- Inefficient way (e.g. X-Stream)

shuffle phase
Scalability & Performance on NUMA

Scalability: #Cores vs. #sockets

- **Ligra**
- **X-Stream**
- **Galois**

**Intel 80-cores (8Sx10C)**

- **X-Stream**
  - 8C: 6.92X
  - 8S: 4.58X
- **Galois**
  - 10C: 6.19X
  - 8S: 2.90X

Performance (sec)

- **X-Stream**
  - 1S: 132s
  - 8S: 29s
- **Galois**
  - 1S: 33s
  - 8S: 12s

Scalability: #sockets

- **Intel 80-cores (8Sx10C)**
- **AMD 64-core (8Sx8C)**

**worse!**

- 8 Socket
  - LG: 2.9X
  - XS: 1.4X
Scalability & Performance on NUMA

Scalability: #Cores vs. #sockets

Intel 80-cores (8Sx10C)

X-Stream
8C: 6.92X
8S: 4.58X

Galois
10C: 6.19X

Minimize remote & random accesses

Eliminate the combination of them

Performance (sec)

Runtime (sec)

Normalized Speedup

#sockets

Intel 80-cores (8Sx10C)

X-Stream
1S: 132s
8S: 29s

Galois
1S: 33s
8S: 12s

AMD 64-core (8Sx8C)

Normalized Speedup

#sockets

Worse!

8 Socket
LG: 2.9X
XS: 1.4X

Minimize remote & random accesses
Goal#1: Reduce remote accesses

Co-locating data and computation within the same NUMA node
Graph-aware Data Layout

Co-locating data and computation within the same NUMA node

1. Graph-aware partitioning

TOPO  1 2 3 4 5 6
DATA  \(D_1, D_2, D_3, D_4, D_5, D_6\)
STAT  1 0 1 1 1 0
**Graph-aware Data Layout**

**Co-locating** data and computation within the same NUMA node

1. **Graph-aware partitioning**

   - **N0**
     - 1 2 3
     - 2 3 3 5 2 5 6
     - D1 D2 D3
     - 1 0 1

   - **N1**
     - 4 5 6
     - 1 3 5 1 2 3 6 2
     - D4 D5 D6
     - 1 1 0

Intuitive
Co-locating data and computation within the same NUMA node

1. Graph-aware partitioning

Graph-aware Data Layout
Co-locating data and computation within the same NUMA node

1. Graph-aware partitioning

N0

1 2 3 4 5 6
2 3 3 2 1 3 1 2 3 2
D1 D2 D3
1 0 1

agent

N1

1 2 3 4 5 6
5 5 6 5 6
D4 D5 D6
1 1 0

sophisticated
Co-locating data and computation within the same NUMA node

1. Graph-aware partitioning

2. NUMA-aware allocation

**TOPO**
- 1.seq
- 2.local
- 3.long

**DATA**
- 1.seq/rnd
- 2.global
- 3.long

**STAT**
- 1.seq/rnd
- 2.global
- 3.short
Goal#2: Eliminate “random + remote”

Random remote access

→ access neighboring vertices on other nodes

Distribute the computations on a single vertex over multiple nodes
Goal #2: Eliminate “random + remote”

Random remote access
→ access neighboring vertices on other nodes

distribute the computations on a single vertex over multiple nodes

Each node handles all edges of partial vertices

Each node handles partial edges of all vertices
NUMA-aware Access Strategy

distribute the computations on single vertex over multiple NUMA-nodes

STAT/curr

TOPO/vertex

TOPO/out-edge

DATA/curr

DATA/next

STAT/next

N0

N1
NUMA-aware Access Strategy

distribute the computations on single vertex over multiple NUMA-nodes

**N0**

<table>
<thead>
<tr>
<th>DATA/curr</th>
<th>DATA/next</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_1) (D_2) (D_3)</td>
<td>X X X</td>
</tr>
</tbody>
</table>

**Update**

**N1**

<table>
<thead>
<tr>
<th>DATA/curr</th>
<th>DATA/next</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_4) (D_5) (D_6)</td>
<td>X X X</td>
</tr>
</tbody>
</table>

**N0**

<table>
<thead>
<tr>
<th>DATA/curr</th>
<th>DATA/next</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_1) (D_2) (D_3) (D_4) (D_5) (D_6)</td>
<td>X X X</td>
</tr>
</tbody>
</table>

**Push**

**N1**

<table>
<thead>
<tr>
<th>DATA/curr</th>
<th>DATA/next</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_1) (D_2) (D_3) (D_4) (D_5) (D_6)</td>
<td>X X X</td>
</tr>
</tbody>
</table>

**Push**

**Colours**

- **SEQ**: Red
- **RND**: Green
- **W**: Blue
- **L**: Yellow
1. Rolling update

2. Hierarchical and efficient barrier

3. Adaptive data structure

Optimizations
Outline

Background & Issues

Design of Polymer

Evaluation
Implementation

Polymer

- ~5,300 SLOCs of C++ code
- Based on scatter-gather iterative model
  - Support both push and pull mode
  - Use a synchronous scheduler
- Several typical graph applications
  - Sparse MM: PageRank, SpMV and BP
  - Traversal: BFS, CC and SSSP

Open source:
http://ipads.se.sjtu.edu.cn/projects/polymer.html
Experiment Settings

Baseline: Ligra v2014.3, X-Stream v0.9, and Galois v2.2

Platform

- 80-core Intel machine (w/o Hyper-Threading)
  8 sockets (E7-8850: 10 cores and 128 GB local RAM)
- 64-core AMD machine (also 8 sockets)

Algorithms (6)

- 3 Spars MM algorithms
- 3 Traversal algorithms

Graphs (4)

- 2 real-world and 2 synthetic graphs

| Graph   | |V|     | |E|     |
|---------|-----|-------|------|
| Twitter | 41.7 M | 1.47B |
| rMat27  | 134.2M | 2.14B |
| Powerlaw| 10.0M  | 105M  |
| roadUS  | 23.9M  | 58M   |
### Overall Performance

| Algo. | |V| | Polymer | Ligra | X-Stream | Galois |
|-------|-----------------|-----------------|----------|----------|----------|----------|
| PR    | Twitter         | 5.3             | 15.0     | 28.9     | 11.6     |
|       | rMat27          | 9.6             | 28.0     | 18.2     | 19.6     |
|       | Power           | 1.6             | 30.5     | 6.1      | 6.6      |
|       | roadUS          | 1.2             | 2.3      | 2.8      | 1.4      |
|       | Twitter         | 7.6             | 29.0     | 59.6     | 11.7     |
|       | rMat27          | 19.2            | 54.3     | 52.5     | 41.9     |
|       | Power           | 1.8             | 31.0     | 5.5      | 6.2      |
|       | roadUS          | 1.3             | 2.8      | 3.0      | 3.6      |
|       | Twitter         | 38.0            | 63.1     | 2017     | 57.1     |
|       | rMat27          | 58.3            | 92.8     | 737      | 75.0     |
|       | Power           | 8.0             | 30.7     | 38.3     | 8.6      |
|       | roadUS          | 5.2             | 2.6      | 20.0     | 7.1      |
# Overall Performance

| Algo. | $|V|$ | Polymer | Ligra | X-Stream | Galois |
|-------|-----|--------|-------|----------|--------|
| PR    | Twitter | 5.3 | 2.85X | 5.48X | 2.19X |
|       | rMat27  | 9.6 | 2.91X | 1.89X | 2.04X |
|       | Power   | 1.6 | 18.9X | 3.76X | 4.11X |
|       | roadUS  | 1.2 | 1.92X | 2.31X | 1.14X |
| SpMV  | Twitter | 7.6 | 3.84X | 7.89X | 1.55X |
|       | rMat27  | 19.2 | 2.83X | 2.74X | 2.19X |
|       | Power   | 1.8 | 17.3X | 3.08X | 3.46X |
|       | roadUS  | 1.3 | 2.18X | 2.30X | 2.74X |
| BP    | Twitter | 38.0 | 1.66X | 53.1X | 1.50X |
|       | rMat27  | 58.3 | 1.59X | 12.6X | 1.29X |
|       | Power   | 8.0  | 3.80X | 4.74X | 1.06X |
|       | roadUS  | 5.2  | 2.01X | 3.84X | 1.36X |
## Overall Performance

| Algo. | |V| | Polymer | Ligra | X-Stream | Galois |
|-------|---|---|--------|--------|---------|--------|
| PR    | Twitter | 5.3 | 2.85X | 5.48X | 2.19X |
|       | rMat27  | 9.6 | 2.91X | 1.89X | 2.04X |
|       | Power   | 1.6 | 18.9X | 3.76X | 4.11X |
|       | roadUS  | 1.2 | 1.92X | 2.31X | 1.14X |
| SpMV  | Twitter | 7.6 | 3.84X | 1.55X | 1.92X |
|       | rMat27  | 2.83X | 3.46X | 2.19X | 1.29X |
|       | Power   | 1.3 | 2.18X | 2.30X | 2.74X |
|       | roadUS  | 1.3 | 2.18X | 2.30X | 2.74X |
| BP    | Twitter | 38.0 | 1.66X | 53.1X | 1.50X |
|       | rMat27  | 58.3 | 1.59X | 12.6X | 1.29X |
|       | Power   | 8.0  | 3.80X | 4.74X | 1.06X |
|       | roadUS  | 5.2  | 2.01X | 3.84X | 1.36X |

Compared with best cases

1.06X ~ 3.08X
## Overall Performance

| Algo. | |V| | Polymer | Ligra | X-Stream | Galois |
|-------|-------------|----------------|---------|---------|----------|---------|
| **BFS** | Twitter | 0.98 | 1.02 | 34.39 | 2.45 |
| | rMat27 | 1.56 | 1.86 | 30.18 | 2.54 |
| | Power | 0.36 | 0.39 | 2.58 | 0.36 |
| | roadUS | 1.16 | 6.93 | 557.7 | 5.01 |
| **CC** | Twitter | 4.60 | 5.51 | 54.8 | 31.9 |
| | rMat27 | 8.72 | 7.74 | 40.0 | 33.9 |
| | Power | 1.23 | 2.56 | 5.13 | 3.51 |
| | roadUS | 57.5 | 63.2 | 985 | 1.18* |
| **SSSP** | Twitter | 2.26 | 3.17 | 165 | 26.3 |
| | rMat27 | 5.78 | 5.26 | 17.9 | 28.5 |
| | Power | 0.85 | 1.12 | 126 | 26.6 |
| | roadUS | 341 | 338 | 1225 | 0.33* |
## Overall Performance

| Algo. | |V| | Polymer | Ligra | X-Stream | Galois |
|-------|-----------------|-----------------|---------|---------|---------|---------|
| BFS   | Twitter         | 0.98            | 1.07X   | 35.9X   | 2.56X   |
|       | rMat27          | 1.56            | 1.19X   | 19.4X   | 1.63X   |
|       | Power           | 0.36            | 1.08X   | 7.16X   | 1.02X   |
|       | roadUS          | 1.16            | 5.97X   | 481.1X  | 4.32X   |
| CC    | Twitter         | 4.60            | 1.20X   | 11.9X   | 6.94X   |
|       | rMat27          | 8.72            | 1.12X   | 4.58X   | 3.88X   |
|       | Power           | 1.23            | 2.08X   | 4.17X   | 2.85X   |
|       | roadUS          | 57.5            | 1.10X   | 17.1X   | 1.18X   |
| SSSP  | Twitter         | 2.26            | 1.40X   | 73.1X   | 11.5X   |
|       | rMat27          | 5.78            | 1.09X   | 21.9X   | 4.93X   |
|       | Power           | 0.85            | 1.31X   | 14.5X   | 31.1X   |
|       | roadUS          | 341             | 1.01X   | 3.59X   | 0.33X   |
Scalability on NUMA

Performance & Scalability: #sockets

Intel 80-cores (8Sx10C)

PageRank

Normalized Speedup

#sockets

12.7X

Runtime (sec)

Ligra

X-Stream

Galois

Polymer

5.48X/X-Stream

2.85X/Ligra

2.19X/Galois

Performance & Scalability: #sockets

Intel 80-cores (8Sx10C)

BFS

Normalized Speedup

#sockets

3.78X

Runtime (sec)

Ligra

Galois

Polymer

0.80X/Galois

1.25X/Ligra

2.96X/X-stream
Conclusion

Polymer: NUMA-aware Graph-structured Analytics

- A comprehensive analysis that uncovers several NUMA characteristics and issues with existing NUMA-oblivious graph analytics systems
- A new system that exploits both NUMA- and graph-aware data layout and memory access strategies
  → minimize remote access
  → eliminate remote random access
- Three optimizations for global synchronization efficiency, load balance and data structure flexibility
http://ipads.se.sjtu.edu.cn/projects/polymer.html
Institute of Parallel and Distributed Systems