GEMS: Graph database Engine for Multithreaded Systems

ALESSANDRO MORARI‡, VITO GIOVANI CASTELLANA‡, ORESTE VILLA+,
ANTONINO TUME0‡, JESSE WEAVER‡, DAVID HAGLIN‡,
SUTANAY CHOU DHURY‡, JOHN FEO‡

‡Pacific Northwest National Laboratory, Richland, WA.
+NVIDIA, Santa Clara, CA.
Outline

▶ Introduction
  ■ Motivation and Challenges
▶ GEMS (Graph database Engine for Multithreaded Systems) overview
▶ GMT (Global Memory and Threading) overview
  ■ Two-level message aggregation
  ■ Lightweight software multithreading
▶ Experimental results
  ■ Synthetic benchmarks
  ■ Berlin SPARQL Benchmarks (BSBM)
▶ Conclusions
Many fields require organization, management, and analysis of massive amounts of data
- E.g.: social network analysis, financial risk management, threat detection in complex network systems, and medical and biomedical databases

Graph databases
- Promising solution to store large and heterogeneous datasets of these application fields
- Organize data in form of triples
  - Subject-predicate-object
  - Following the Resource Description Framework (RDF)
  - Set of triples represent a labeled, directed multigraph
- Queried through languages such as SPARQL
  - Fundamental operation is graph matching
Using graphs

Graph benefits
- Graphs are memory efficient for storing heterogeneous or not rigidly structured data
- Graph methods (based on edge traversal) are inherently parallel

Graph challenges
- Fine-grained data accesses
  - size of a pointer, or an of an element of a linked list
- Unpredictable data accesses
- Very difficult to partition
- Parallel graph methods may have high synchronization intensity
Commodity clusters and graph algorithms

- Benefits of commodity clusters
  - Low costs
  - High core count
  - Increasing memory per node
  - Increasing network bandwidth

- Challenges of commodity clusters
  - Processors optimized for locality
    - Deep cache hierarchies
  - Networks optimized for batched data transfer
Return the names of all persons owning at least two cars, of which at least one is a SUV
Addressing Commodity Cluster Limitations

- Custom runtime layer
  - GMT – Global Memory and Threading
  - NOT a general runtime – Deeply customized for the database requirements

- Partitioned Global Address Space (PGAS) data model

- Lightweight software multithreading to hide latency of remote operations

- Asynchronous user level task parallelism
  - Loop level parallelism (parFor)
  - (limited) support for active messages

- Two-level message aggregation
GMT architecture

- Three classes of pthreads (pinned to cores)
  - **Worker**
    - Executes application code through lightweight tasks
  - **Helper**
    - PGAS and communication management
  - **Communication Server**
    - MPI communication
Message aggregation

- Two-level aggregation
  - Queues are per destination node

- Command blocks
  - “local” to a core

- Aggregation queues
  - Common to a node

- Aggregation buffers
  - Buffers where data are effectively copied before an MPI send operation
Multithreading

- A node receives commands to spawn tasks
  - Commands are related to iterations of loops

- A helper parses the commands and pushes the task on the iteration block queue (itb)

- A worker pops some iteration from the itb, and generates the task contexts in its local task queue

- When a task generates a (blocking) remote operation, the worker switches to another task
Experimental Setup

- Olympus supercomputer (PIC - Pacific Northwest National Laboratory’s Institutional Computing)

- 604 nodes
  - Infiniband QDR (4 GB/s theoretical peak bandwidth)
  - 2 Opteron 6272 per node
    - 2.1 GHz
    - 2 dies, 4 “modules” per die (8 integer “cores”, 4 floating point cores)
    - Each module: 2 x 16 KB data cache, 64 KB instruction cache, 2 MB L2
    - 8 MB L3 per die
  - 64 GB DDR3-1600 per node

- Parameters
  - 15 workers, 15 helpers, 1 communication server
  - 1024 tasks per worker
  - 64 KB aggregation buffers

- Benchmarks
  - Synthetic - GMT
  - Berlin SPARQL Benchmark (BSBM) - GEMS
Synthetic Benchmarks – Bandwidth while Increasing the Number of Tasks per Node

2 nodes

128 nodes
BSBM - GEMS

Time (in seconds) to build the database and execute BSBM queries 1-6 with 100M, 1B and 10B triples. Query execution time is an average of the time to obtain the results of a query when 100 queries of the same type run concurrently.

<table>
<thead>
<tr>
<th>Nodes</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>build</td>
<td>199.00</td>
<td>106.99</td>
<td>59.85</td>
<td>33.42</td>
<td>628.87</td>
<td>350.74</td>
<td>200.54</td>
<td>136.69</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q1</td>
<td>1.83</td>
<td>1.12</td>
<td>0.67</td>
<td>0.40</td>
<td>5.65</td>
<td>3.09</td>
<td>1.93</td>
<td>2.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q2</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.05</td>
<td>0.30</td>
<td>0.34</td>
<td>0.23</td>
<td>0.35</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3</td>
<td>4.07</td>
<td>2.73</td>
<td>1.17</td>
<td>0.65</td>
<td>12.79</td>
<td>6.88</td>
<td>4.50</td>
<td>2.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q4</td>
<td>0.13</td>
<td>0.13</td>
<td>0.14</td>
<td>0.15</td>
<td>0.31</td>
<td>0.25</td>
<td>0.22</td>
<td>0.27</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q5</td>
<td>0.07</td>
<td>0.07</td>
<td>0.07</td>
<td>0.11</td>
<td>0.11</td>
<td>0.12</td>
<td>0.14</td>
<td>0.18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q6</td>
<td>0.01</td>
<td>0.02</td>
<td>0.02</td>
<td>0.03</td>
<td>0.02</td>
<td>0.03</td>
<td>0.04</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) 100M triples, 2 to 16 nodes
(b) 1B triples, 8 to 64 nodes
(c) 10B triples, 64 and 128 nodes

July 8, 2014
Conclusions

- Presented GEMS – Graph database Engine for Multithreaded Systems
  - Full software stack for graph databases on commodity clusters
  - Utilizes almost only graph-based methods across all the layers
  - Includes: SPARQL-to-C++ compiler, a library of algorithms and data structures, and a custom runtime (GMT)

- Discussed the runtime (GMT – Global Memory and Threading)
  - Global address space
  - Lightweight software multithreading
  - Message aggregation
  - Customized to the needs of the database

- Demonstrated how this integrated approach provides scaling in size and performance as more nodes are added to the cluster
Questions?

- alessandro.morari@pnnl.gov
- vitoGiovanni.castellana@pnnl.gov
- ovilla@nvidia.com
- antonino.tumeo@pnnl.gov
- jesse.weaver@pnnl.gov
- david.haglin@pnnl.gov
- sutanay.choudhury@pnnl.gov
- john.feo@pnnl.gov