Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

GEMS: Graph database Engine for Multithreaded Systems

ALESSANDRO MORARI‡, VITO GIOVANI CASTELLANA‡, ORESTE VILLA+,
ANTONINO TUMEO‡, JESSE WEAVER‡, DAVID HAGLIN‡,
SUTANAY CHOUDHURY‡, 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

Introduction

- 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

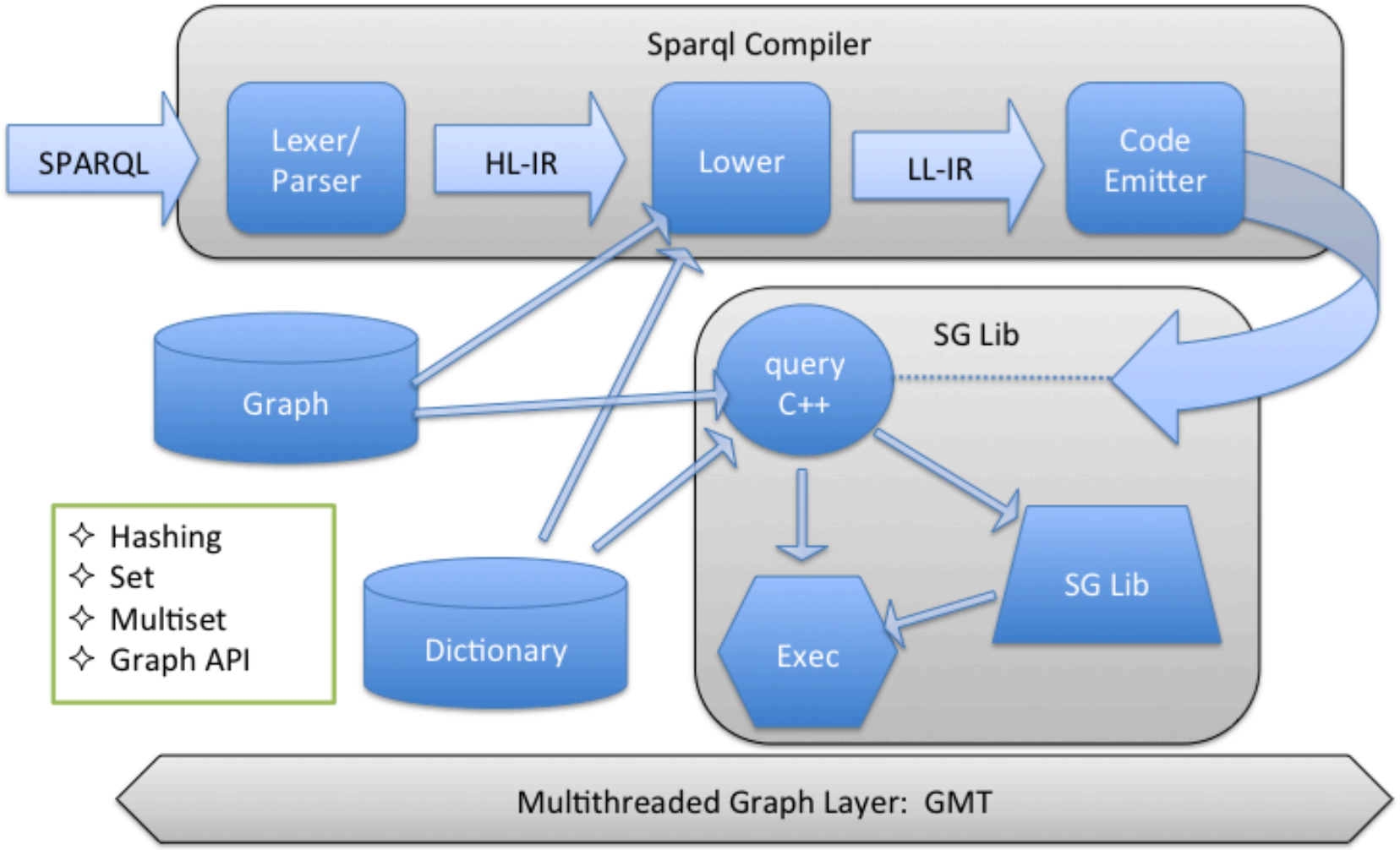
Using graphs

- 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

GEMS

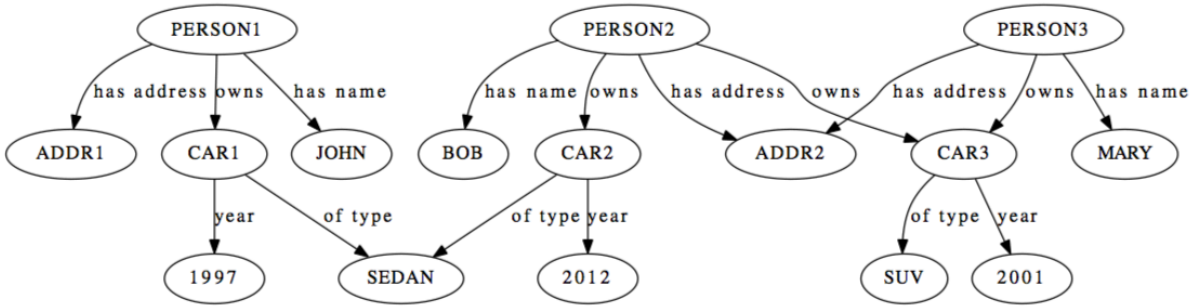

Sparql Compiler
SPARQL
Lexer/
Parser
HL-IR
Lower
LL-IR
Code
Emitter
Graph
SG Lib
query
C++
Hashing
Set
Multiset
Graph API
Dictionary
Exec
SG Lib
Multithreaded Graph Layer: GMT

GEMS example

- Return the names of all persons owning at least two cars, of which at least one is a SUV

```
PERSON1 has_name    JOHN  .
PERSON1 has_address ADDR1 .
PERSON1 owns        CAR1  .
CAR1    of_type     SEDAN .
CAR1    year        1997  .
PERSON2 has_name    BOB   .
PERSON2 has_address ADDR2 .
PERSON2 owns        CAR2  .
CAR2    of_type     SEDAN .
CAR2    year        2012  .
PERSON2 owns        CAR3  .
CAR3    of_type     SUV   .
CAR3    year        2001  .
PERSON3 has_name    MARY  .
PERSON3 has_address ADDR2 .
PERSON3 owns        CAR3  .
```

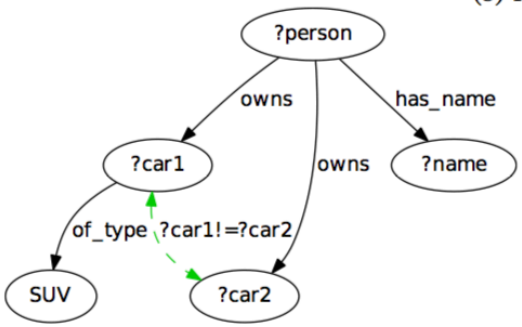
(a) Dataset in simplified N-Triples format



(b) RDF Graph

```
SELECT DISTINCT ?name
WHERE {
   ?person owns     ?car1 .
   ?person owns     ?car2 .
   ?person has_name ?name .
   ?car1   of_type  SUV   .
   FILTER(?car1 != ?car2)
}
```

(c) Simplified SPARQL query



(d) Pattern graph

```
1  has_name = get_label("has_name")
2  of_type  = get_label("of_type")
3  owns     = get_label("owns")
4  suv      = get_label("SUV")
5  forall e1 in edges(*, of_type, suv)
6    ?car1 = source_node(e1)
7    forall e2 in edges(*, owns, ?car1)
8      ?person = source_node(e2)
9      forall e3 in edges(?person, owns, *)
10       ?car2 = target_node(e3)
11       if (?car1 != ?car2)
12         forall e4 in edges(?person,has_name,*)
13           ?name = target_node(e4)
14           tuples.add(<?name>)
15 distinct(tuples)
```

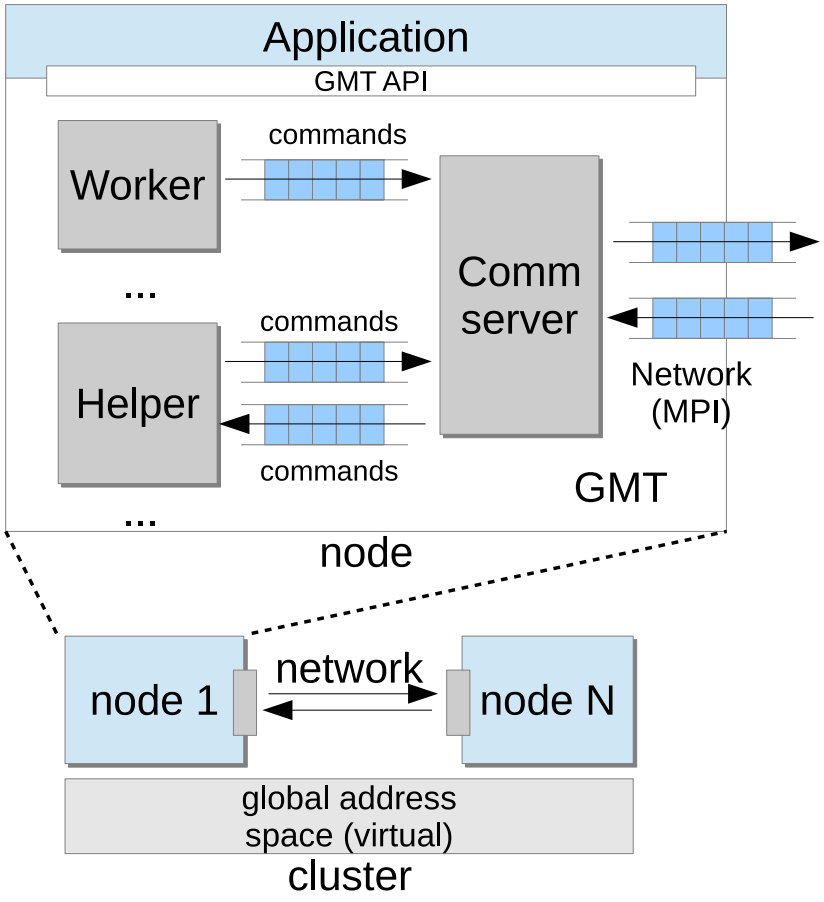
(e) Pseudocode

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

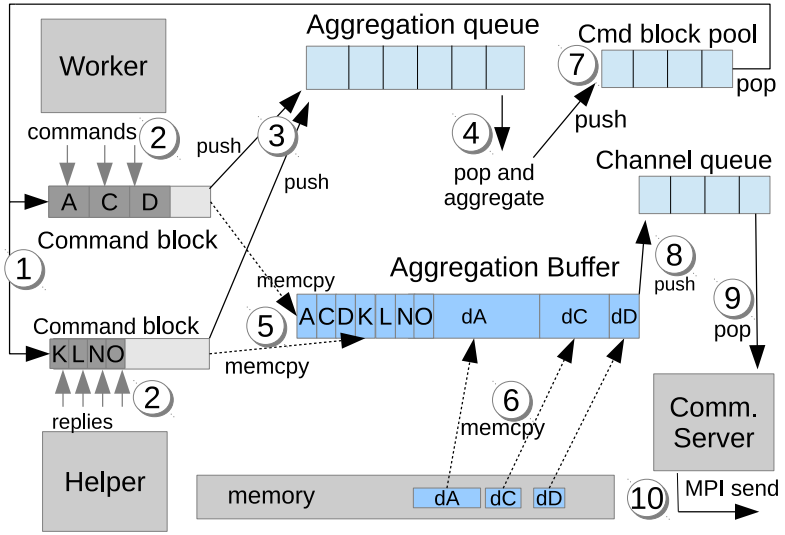
GMT architecture

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



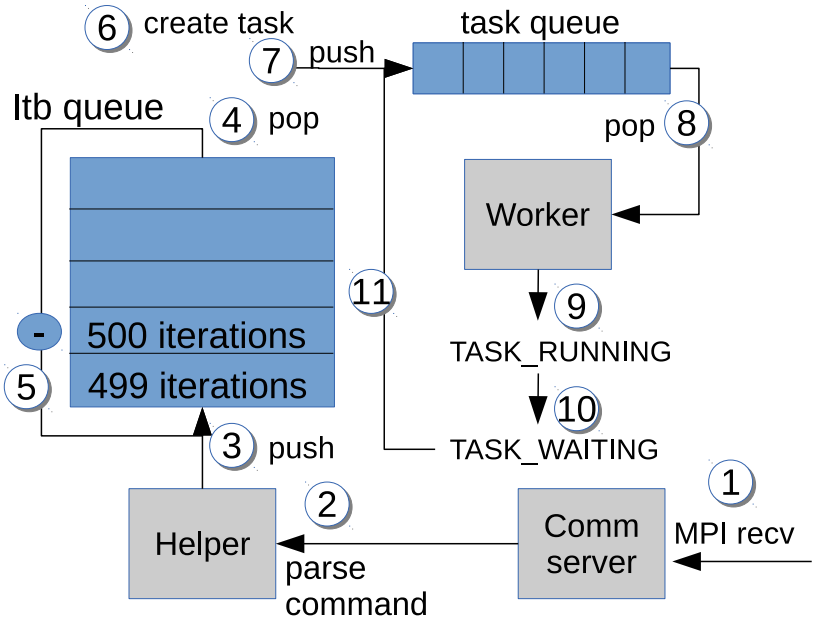
Message aggregation

- 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 generate 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

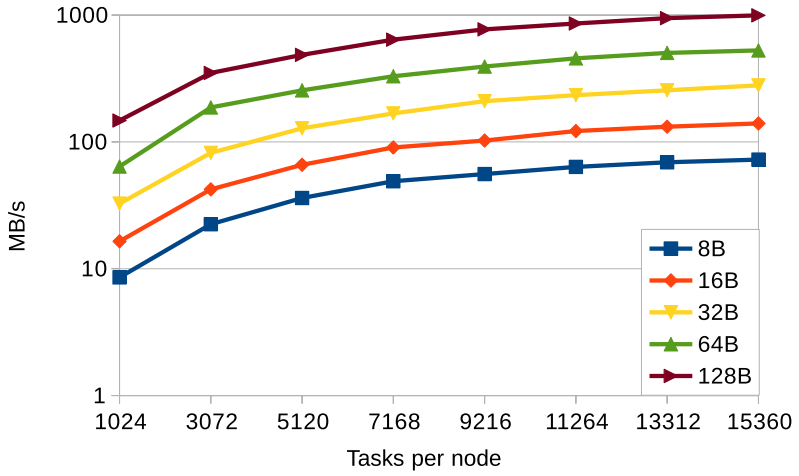
- Parameters
  - 15 workers, 15 helpers, 1 communication server
  - 1024 tasks per worker
  - 64 KB aggregation buffers

- Benchmarks
  - Synthetic - GMT
  - Berlin SPARQL Benchmark (BSBM) - GEMS

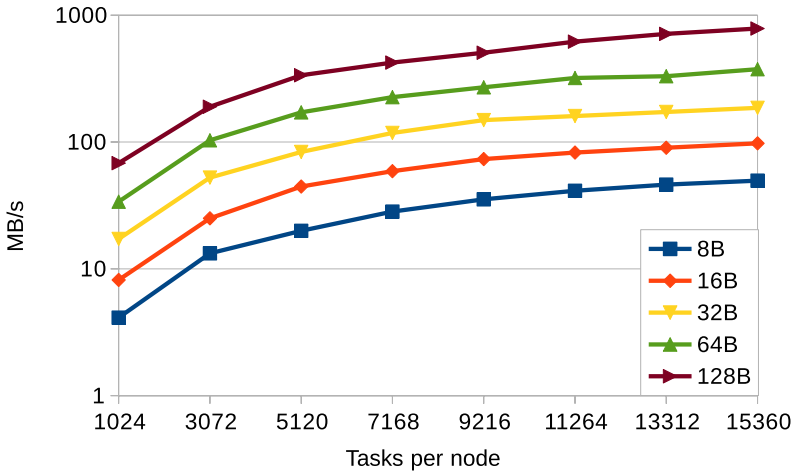
Synthetic Benchmarks – Bandwidth while Increasing the Number of Tasks per Node

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

2 nodes



128 nodes



BSBM - GEMS

Nodes	2	4	8	16
build	199.00	106.99	59.85	33.42
Q1	1.83	1.12	0.67	0.40
Q2	0.07	0.07	0.07	0.05
Q3	4.07	2.73	1.17	0.65
Q4	0.13	0.13	0.14	0.15
Q5	0.07	0.07	0.07	0.11
Q6	0.01	0.02	0.02	0.03

(a) 100M triples, 2 to 16 nodes

Nodes	8	16	32	64
build	628.87	350.74	200.54	136.69
Q1	5.65	3.09	1.93	2.32
Q2	0.30	0.34	0.23	0.35
Q3	12.79	6.88	4.50	2.76
Q4	0.31	0.25	0.22	0.27
Q5	0.11	0.12	0.14	0.18
Q6	0.02	0.03	0.04	0.05

(b) 1B triples, 8 to 64 nodes

Nodes	64	128
build	1066.27	806.55
Q1	27.14	39.78
Q2	1.48	1.91
Q3	24.27	18.32
Q4	2.33	2.91
Q5	2.13	2.82
Q6	0.40	0.54

(c) 10B triples, 64 and 128 nodes

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

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

Thank you for your attention!

Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

- 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