SYSTAP / bigdata®

Open Source
High Performance
Highly Available
SYSTAP, LLC

Small Business, Founded 2006
100% Employee Owned

Customers

• OEMs and VARs
• Government
• Telecommunications
• Health Care
• Network Storage
• Finance
• Design and Manufacturing
• Collaboration & KM Portals

Products and Services

• Bigdata® RDF database
  – Dual license (GPLv2, commercial)
  – Training
  – Support
  – Custom Services
• Advanced Research Projects
The data – it’s about the data
Bigdata 1.3

• Fast, scalable, open source, standards compliant database
  - Single machine to 50B+ triples or quads
    - Plus a dedicated statement level metadata mode (RDR – see web site)
  - Scales horizontally on a cluster
  - SPARQL 1.1 Query, Property Paths, Update, Federated Query, etc.
  - Native RDFS+ inference.
    - Vectored query engine.
  - High Availability
  - RDF Graph Mining
Related “Graph” Technologies

Redpoint repositions existing technology.

MPGraph compares favorably with high end hardware solutions from YARC, Oracle, and SAP, but is open source and uses commodity hardware.
New Products and Services

Redpoint Graph Database
• Larger market, same tech.
  – 2x faster than neo4j.
  – Graph Traversal APIs
  – Efficient Link Attributes
• Open Source (Subscriptions)
• Highly Available

MPGraph
• GPU graph mining library
  – 5-500x faster than graphlab
  – 50,000x faster than graphdbs.
• Open Source
• DARPA funding
• Disruptive technology
  – Early adopters
  – Huge ROIs
Autodesk PLM360

Manufacturing Product Data is Heterogeneous
...and difficult to find, re-use, and share
Road Map

• Column-wise
  – Faster load and query.
  – Increased data density and scaling.
  – Integration point with GPU (shared data).

• Multi-node GPU
  – 2D decomposition (DARPA STTR)

• Performance optimization for scale-out
  – Reducing latency and increasing throughput
  – Integration point for SPARQL acceleration and 2D GPU cluster.

• SPARQL on GPU
  – Query at 3 billion edges/second
  – Same underlying library, but horizontal scaling is NOT 2D.
Hadoop / bigdata® pipeline

Map/Reduce Layer

+/- stmts

Durable Queue

Inference Cloud

BD Journals on PFS / HDFS

+/- stmts

Durable Queue

Scalable inference workload

Can be used for custom quads-mode inference strategies

Linear scaling on query throughput

HA Query Cloud
High Availability

- **Shared nothing architecture**
  - Same data on each node
  - Coordinate *only* at commit

- **Scaling**
  - 50 billion+ triples or quads
  - Query throughput scales linearly

- **Self healing**
  - Automatic failover
  - Automatic resync after disconnect
  - Online single node disaster recovery

- **Online Backup**
  - Online snapshots (full backups)
  - HA Logs (incremental backups)

- **Point in time recovery (offline)**
Self-Healing

Service can fail for a variety of reasons:

- JVM down
- Machine down
- Network partition
- Zookeeper timeout
- Discovery failure
- Wrong commit point
- Severe clock skew

- Goal is to guarantee eventual consistency without allowing intermediate illegal states.
- Persistent state of the service must remain self-consistent

Quorum

\[ k = 3 \]

size = 2

leader

e
d

follower

e
d

join

e
d

synchronize

e
d

(delta)
BSBM 100M (3-Node HA Cluster)

- 3-Node, Shared-Nothing Replication Cluster
  - 3x 2011 Mac Mini (4 cores, 16G RAM and SSD).
- Query Scales Linearly
- CPU bound
  - 70-90k QMpfH on newer servers.
Many Core is the Future

- Top 500 Super Computer Sites:
  - 3 out of top 5 are GPU clusters (11/2011)
  - #1 and #8 (11/2012)
- CPU Clock Rates are stagnant.
- Simple compute units + parallelism => Increased performance.

Historical Single-/Double-Precision Peak Compute Rates

<table>
<thead>
<tr>
<th>Date</th>
<th>GFLOPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>10^1</td>
</tr>
<tr>
<td>2004</td>
<td>10^2</td>
</tr>
<tr>
<td>2006</td>
<td>10^3</td>
</tr>
<tr>
<td>2008</td>
<td>10^3</td>
</tr>
<tr>
<td>2010</td>
<td>10^3</td>
</tr>
<tr>
<td>2012</td>
<td>10^3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Precision</th>
<th>SP</th>
<th>DP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Vendor</th>
<th>AMD (GPU)</th>
<th>NVIDIA (GPU)</th>
<th>Intel (CPU)</th>
<th>Intel Xeon Phi</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOP 10 Systems - 11/2011

1. K computer, SPARC64 VIIfx 2.0GHz, Tofu interconnect
2. NUDT YH MPP, Xeon X5670 6C 2.93 GHz, NVIDIA 2050
3. Cray XT5-HE Opteron 6-core 2.6 GHz
4. Dawning TC3600 Blade, Intel X5650, NVIDIA Tesla C2050 GPU
5. HP ProLiant SL390s G7 Xeon 6C X5670, NVIDIA GPU, Linux/Windows
GPUs – A Game Changer for Graph Analytics?

- Graphs are everywhere in data, also a powerful data model for federation
- GPUs may be the technology that finally delivers real-time analytics on large graphs
  - 10x speedup over CPU
  - 10x DRAM bandwidth
- This is a hard problem
  - Data dependent parallelism
  - Non-locality
  - PCIe bus is bottleneck
- Significant speed up over CPU on BFS
  - 3 billion edges per second on one GPU (see chart).
- Roadmap
  - GPU accelerated vertex-centric graph mining platform.
  - GPU accelerated graph query

Breadth-First Search on Graphs
10x Speedup on GPUs

Million Traversed Edges per Second

Average Traversal Depth
GAS – a Graph-Parallel Abstraction

• Graph-Parallel Vertex-Centric API ala GraphLab
• “Think like a vertex”

• Gather: collect information about my neighborhood

• Apply: update my value

• Scatter: signal adjacent vertices

• Can write all sorts of graph algorithms this way
Graph Mining on GPU Clusters

- 2D partitioning (aka vertex cuts)
  - Compute grid defined over virtual nodes.
  - Patches assigned to virtual nodes based on source and target identifier of the edge.

- Minimizes the communication volume.
  - All messages in row or col of grid.
    - One hop in-edge neighborhood is a column.
    - One hop out-edge neighborhood is a row.
    - For the diagonal only, the source and target vertices are in the same patch.

- Batch parallel Gather in row, Scatter in column.
  - Computation decomposed and distributed using binary (albian) operator (sum, min, max, etc.)
  - Local aggregates combined at caller.
Similar models, different problems

• Graph query and graph analytics (traversal/mining)
  – Related data models
  – *Very* different computational requirements
• Many technologies are a bad match or limited solution
  – Key-value stores (bigtable, Accumulo, Cassandra, HBase)
  – Map-reduce
• Anti-pattern
  – Dump all data into “big bucket”
    • Storage and computation patterns must be correctly matched for high performance.
• High performance requires
  – Query: Locality (1D partitioning, multiple indices, query optimization, and execution constrained joins to read as little data as possible).
  – Analytics: Parallelism, memory bandwidth, 2D partitioning.
Questions?

Bryan Thompson
SYSTAP, LLC
bryan@systap.com
202/462-9888
Unifying Architecture (example)

- **Unified Data Model**: Resource Centric (Linked Data)
  - Heterogeneous Data Sources as Input
  - Streams
  - Unstructured
  - Semi-structured
  - Structured

- **Unified Compute and Storage Model**
  - Data Bus
    - Business Logic
    - Web Clients
    - Peer Systems
  - Data Cache
    - Key Value Stores
  - Database (SSD)
    - Aggregated –or– federated
    - High-level Query (SPARQL)
  - Graph Mining (GPUs)
    - Graph traversal / mining
    - “Think like a vertex”

- **Data Updates and Queries**
  - Discover
  - Federate
  - Aggregate
  - Query
  - Update

**Key Points**
- Unstructured data sources
- Resource centric (linked data)
- Federated data processing
- High-level query language (SPARQL)
- Graph mining with GPUs