BigDAWG: A Polystore System to Support Complex Analytics on Heterogeneous Data

Tim Mattson, Intel labs
timothy.g.mattson@intel.com
Intel-PI for the Big Data “Intel Science and Technology Center”

People I stole content from for this talk: Stavros Papadopoulos (TileDB), Nesime Tatbul (S-store), Narayanan Sundaram and Nadathur Rajagopalan Satish (GraphMAT), Vijay Gadepally, Jennie Duggan, and Aaron Elmore (BigDAWG).
Legal Disclaimer & Optimization Notice

- INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS”. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.
- Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.
- Copyright © , Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Xeon Phi, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Notice revision #20110804
Disclaimer

• The views expressed in this talk are those of the speaker and not his employer.

• If I say something “smart” or worthwhile:
  – Credit goes to the many smart people I work with.

• If I say something stupid...
  – It’s my own fault

I work in Intel’s research labs. I don’t build products. Instead, I get to poke into dark corners and think silly thoughts... just to make sure we don’t miss any great ideas.

Hence, my views are by design far “off the roadmap”.

These two diagrams are equivalent representations of a graph.

$A^T$ = the adjacency matrix ... Elements nonzero when vertices are adjacent
Multiple-source breadth-first search

- Sparse array representation => space efficient
- Sparse matrix-matrix multiplication => work efficient
- Three possible levels of parallelism: searches, vertices, edges

Multiplication of sparse matrices captures Breadth first search and when extended with “arbitrary semi-rings”, serves as the foundation of all algorithms based on the BFS pattern.
I am old … I remember when the BLAS showed up and changed my life!!!

• BLAS: The Basic Linear Algebra subroutines

<table>
<thead>
<tr>
<th>BLAS 1</th>
<th>Lawson, Hanson, Kincaid and Krogh, 1979</th>
<th>LINPACK</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAS 2</td>
<td>Dongarra, Du Croz, Hammarling and Hanson, 1988</td>
<td>LINPACK on vector machines</td>
</tr>
<tr>
<td>BLAS 3</td>
<td>Dongarra, Du Croz, Hammarling and Hanson, 1990</td>
<td>LAPACK on cache based machines</td>
</tr>
</tbody>
</table>

• The BLAS supported a separation of concerns:
  – HW/SW optimization experts tuned the BLAS for specific platforms.
  – Linear algebra experts built software on top of the BLAS .. high performance “for free”.
  – It is difficult to overestimate the impact of the BLAS … they revolutionized the practice of computational linear algebra.

A group of us are working to do for Graph Algorithms what the BLAS did for Linear Algebra…. We call this the “GraphBLAS project”.

3rd party names are the property of their owners
Is the performance from these linear algebra approach any good?

GraphMat: PageRank speedup

GraphMat: High performance graph analytics made productive
Graph algorithms are great for entertaining mathematicians, but to be useful you need to feed them lots of data

• Graph problems are Big Data problems.
• So how do you “feed the beast”?
Are HADOOP and SPARK good enough?

Third party names are the property of their owners.
The performance isn’t good enough: GraphMAT and Spark

- Exceed the performance of Spark’s graph processing framework (GraphX) by up to 4x on BFS and PageRank algorithms
- Less than 8 second overhead for transfer and conversion of 85 million edge LiveJournal dataset

GraphMat: High performance graph analytics made productive

3rd party names are the property of their owners
Big Data in the real world

• Consider patient data in an Intensive Care Unit (e.g. MIMIC II data set*)

  • EKG traces
  • Blood oxygen
  • Blood pressure
  • EEG traces

  • Demographic
  • Caregiver notes
  • Medical charts
  • Lab test results
  • Xray, MRI, etc.

The challenge … apply predictive analytics across all data … so we can show up to restart a heart before it stops beating!!!

Big Data in the real world
Messy, heterogeneous, complex, streaming ...

- Consider patient data in an Intensive Care Unit (e.g. MIMIC II data set*)

# MIMC doesn’t include images. We are talking to several groups to add an image database to our project
Analysis of published MIMICII papers

- Data in databases is used; data in files is not
  - Data in files is nearly equivalent to deleting the data

A disruptive idea: Match data to the data-store technology but present as a single Data Base Management system to the end-users … A disruptive idea we call Polystore.

Source: Vijay Gadepally of MIT Lincoln labs

*Based on PhysioNet MIMIC2 ICU data
**BigDAWG: A Prototype Polystore System**

- **BigDAWG**
  - Polystore: match data to the storage engine ... “one size does not fit all”.

- **BigDAWG Islands**
  - A data model + query operations
  - One or more storage engines

- **One API to bind them all!**
  - BigDAWG Common interface (*interface*)
  - Query translators (*shim*)
  - Data translators (*cast*)

**User specifies the Island**

RELATIONAL (select avg(temp) from device)

ARRAY (multiply(A,B))
BigDAWG Example: MIMC II dataset
Predict severe hypotension from patterns in ECG data*

- Procedure
  - Discrete Wavelet Transform of ECG
  - Generate wavelet coefficient histogram
  - TF-IDF waveform coefficients (weight rare changes higher)
  - Correlate against all other ECGs

- Show timings for individual pieces in two different types of databases
  - Option 1: Pick one DB for the whole workflow
  - Option 2: Let the best DB for each piece do their part

Execute on a system with two Islands: A relational Island (Myria) and an array island (SciDB)

*Thanks to Brandon Haynes, UW
Results: MIMC II dataset
Predict severe hypotension from patterns in ECG data*

Time taken to perform analytic using different technologies

- SciDB
- Myria
- Hybrid

Technology Used

- Discrete Wavelet Transform
- Term Frequency-Inverse Document Frequency
- K-Nearest Neighbors

Time Taken (Seconds)

*ECG: Electrocardiogram

*Thanks to Brandon Haynes, UW
BigDAWG Overhead

Overhead Incurred When Using BigDAWG For Common Database Queries

- **Overhead Incurred (ms)**
- **Query without BigDAWG (ms)**

*Thanks to Peinan Chen, MIT*
BigDAWG enhancements: Add a streaming Island

**S-Store:**
- A system for Streaming transactional semantics (ACID, order, exactly once)
- ETL: **E**xtract + **T**ransform Dispatch to multiple BigDAWG engines

![Diagram of BigDAWG Common Interface with S-Store, Relational Island, Array Island, and Key-Value Island connected by Cast and Shim](https://via.placeholder.com/150)
BigDAWG enhancements: Add Specialized storage managers

TileDB: A new storage manager for Sparse data
TileDB: a new array data storage manager optimized for Sparse Arrays

**Logical representation**
- **attribute values** $(a_1, a_2, ..., a_m)$
- **empty cell**
- **tile**
- **dimensions**

**Physical representation**
- **coordinates**
- **Files**
- **segment**
- **tile**
- **cell**

**Tile:** Atomic unit of processing

**Segment:** Atomic unit of I/O

Manage array storage as tiles of different shape/size in the index space, but with ~equal number of non-empty cells.
TileDB is ideal for storing Genomics Data

- Represent variation of a sample from a reference Genome (Genome Variant Call format or gVCF)
- Store as a sparse 2D array in TileDB ... store a non-empty cell for every END endpoint of the gVCF ranges

Binary files (one per attribute)

- The cells are sorted in column-major order, and compressed
gVCF is the GATK “Genome variant call format”. BCF is Binary variant of gVCF. There is one gVCF file per sample. The combined gVCF file contains data for many samples (A complex op since you have to deal with overlapping Genome ranges).

Intel® Xeon® E5 2697 v2 CPU, 12 cores, dual socket, 128 GB RAM, CentOS6.6, Western Digital 4 TB WD4000F9YZ-0 as a ZFS RAID0 pool.
GenomicsDB/TileDB combine gVCF operation + median (5K random positions)

TileDB’s efficient sparse array representation combined with the GenomicsDB optimized access algorithm really stands out for random reads

Intel® Xeon® E5 2697 v2 CPU, 12 cores, dual socket, 128 GB RAM, CentOS6.6, Western Digital 4 TB WD4000F9YZ-0 as a ZFS RAID0 pool.
TileDB vs. PHDF5 with Dense Data running on multiple cores

PHDF5, 10K random updates in seconds

Note: TileDB single node 10K random updates in 0.2 seconds

Why is TileDB so good?

TileDB stores updates as fragments rather than writes to random file locations

Intel® Xeon® E5 2697 v2 CPU, 12 cores, dual socket, 128 GB RAM, CentOS6.6, Western Digital 4 TB WD4000F9YZ-0 as a ZFS RAID0 pool.

3rd party names are the property of their owners
Future work

• Integrate GraphBLAS into TileDB
  – Integrate fundamental GraphBLAS building blocks into TileDB ... move compute into the storage manager
  – Core data model of GraphMAT is sparse arrays ... should move directly into TileDB
  – Expose GraphMAT as a layer running on top of TileDB ... support “think like a vertex” and “Linear Algebra” approaches.
Conclusion

• Big Data is much more than Hadoop.
  – One Size does not fit all

• The future belongs to polystore systems
  – A single high level data management system that is composed of many individual storage management systems.
    • Storage management matches the data for a better performance.
    • Analytics embedded into the storage managers to keep computing near the data.