GRAPHULO

Server-side Sparse Matrix Multiply in the Accumulo Database

Dylan Hutchison[1,2]*   Vijay Gadepally[1]*   Jeremy Kepner[1]*   Adam Fuchs[3]

[1]MIT Lincoln Laboratory   [2]University of Washington   [3]Sqrrl Inc.

2015 September

Massachusetts Institute of Technology

*This material is based upon work supported by the National Science Foundation under Grant No. DMS-1312831. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

This work is NOT

Creating the best system

for a particular task (matrix multiply)

This work is NOT

Creating the best system

for a particular task (matrix multiply)

This work IS

Adding graph analytic capabilities

(matrix multiply) to an all-around good

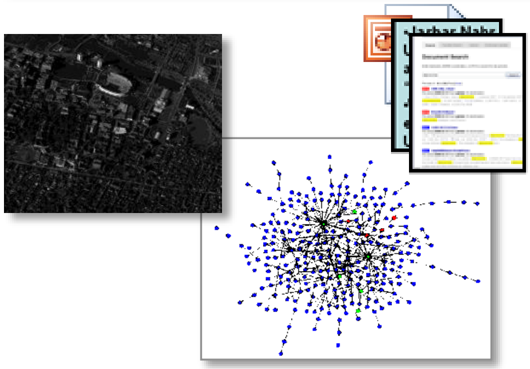
system used in practice today (Accumulo)

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

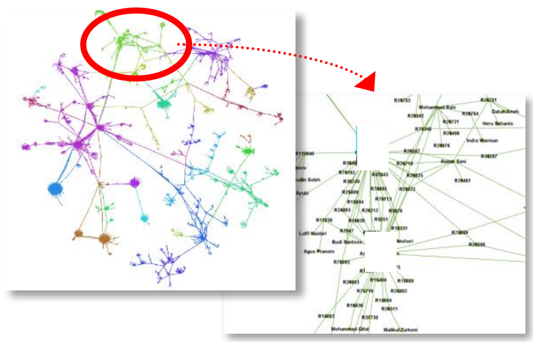
Real Graph Analytics used in Accumulo

ISR



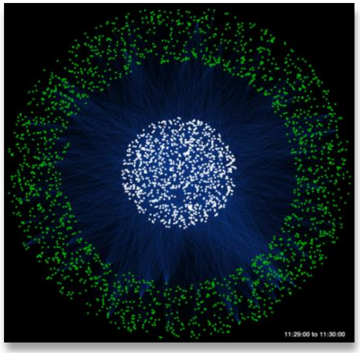
- Graphs represent entities and relationships detected through multi-INT sources
- 1,000s – 1,000,000s tracks and locations
- GOAL: Identify anomalous patterns of life

Social



- Graphs represent relationships between individuals or documents
- 10,000s – 10,000,000s individual and interactions
- GOAL: Identify hidden social networks

Cyber

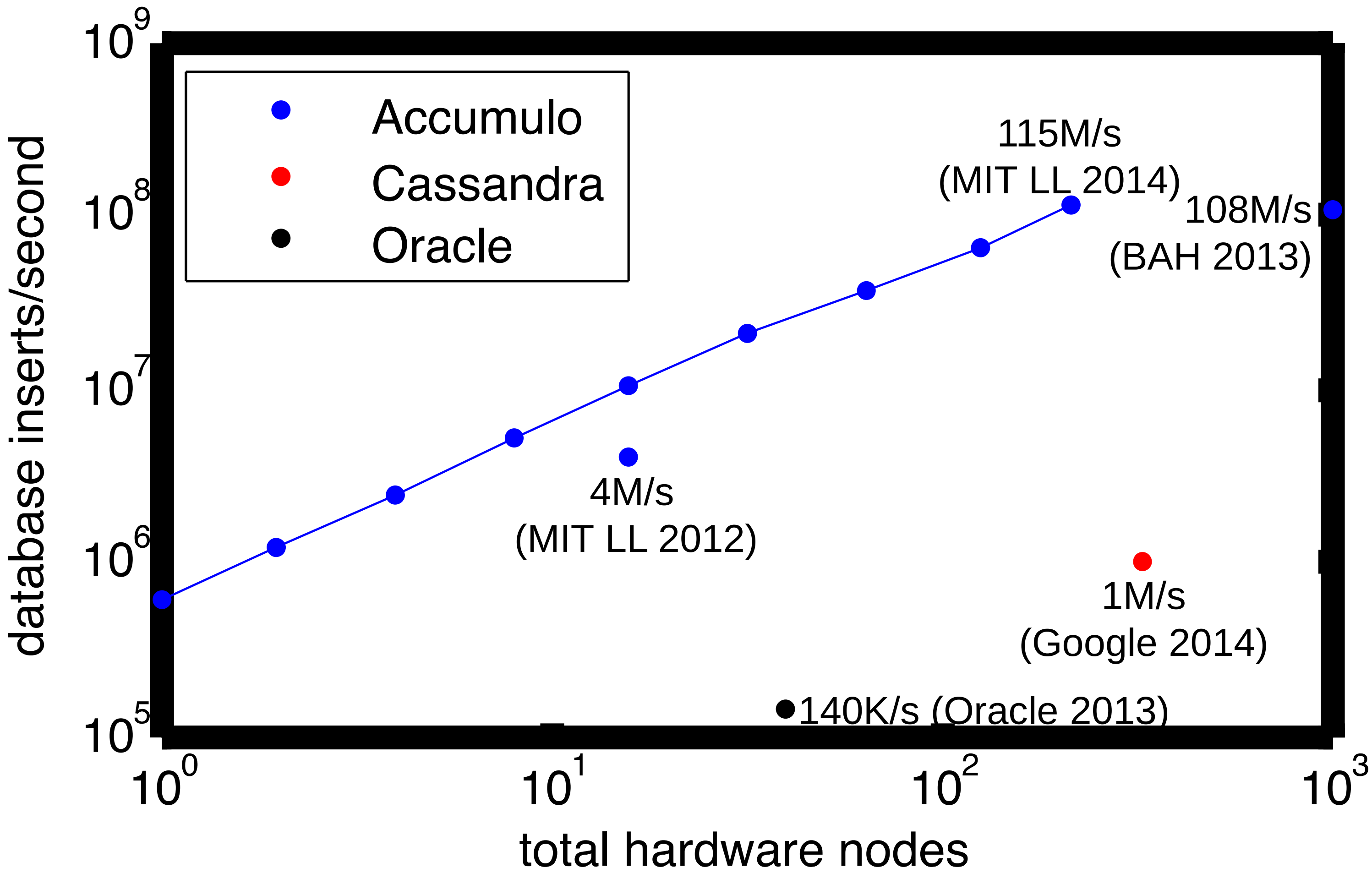


- Graphs represent communication patterns of computers on a network
- 1,000,000s – 1,000,000,000s network events
- GOAL: Detect cyber attacks or malicious software

Many groups store graph data in Accumulo

➔ Need tools for graph analysis in Accumulo

Why Accumulo?



Accumulo ingest performance is 100x greater than competing technologies

Graphulo Overview

- **Primary Goal**
  - **Open source Apache Accumulo Java library that enables many graph algorithms in Accumulo**
- **Core primitives: GraphBLAS**
- **3 Graph Schemas**
  - **Adjacency, Incidence, Single-Table**
- **4 Demonstration Graph Algorithms**
  - **Degree-filtered Breadth First Search, Jaccard coefficients, k-Truss subgraph, Non-negative Matrix Factorization**
- **Focus on Interactive Computing**
  - **"Queued" / Localized analytics within a neighborhood, as opposed to whole table analytics**
  - **Low latency more important than high throughput**
  - **Progress monitoring for user sanity**
    - ***Is the library working or stuck?***

GraphBLAS initial function list

Function	Parameters	Returns	Math Notation
SpGEMM	- sparse matrices **A** and **B** - unary functors (op)	sparse matrix	**C** = op(**A**) * op(**B**)
SpM{Sp}V **(Sp: sparse)**	- sparse matrix **A** - sparse/dense vector **x**	sparse/dense vector	**y** = **A** * **x**
SpEWiseX	- sparse matrices or vectors - binary functor and predicate	in place or sparse matrix/vector	**C** = **A** .* **B**
Reduce	- sparse matrix **A** and functors	dense vector	**y** = sum(**A**, op)
SpRef	- sparse matrix **A** - index vectors **p** and **q**	sparse matrix	**B** = **A**(**p**,**q**)
SpAsgn	- sparse matrices **A** and **B** - index vectors **p** and **q**	none	**A**(**p**,**q**) = **B**
Scale	- sparse matrix **A** - dense matrix or vector **X**	none	check manual
Apply	- any matrix or vector **X** - unary functor (op)	none	op(**X**)

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Matrix Multiply on Big Data

Traditional Matrix Multiply: $AB = C$

$$\begin{bmatrix} 6 & 5 & 0 & 2 \\ 0 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 5 & 0 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 23 \\ 0 & 12 \end{bmatrix}$$

Matrix Multiply on Big Data

Traditional Matrix Multiply: $AB = C$

$$\begin{bmatrix} 6 & 5 & 0 & 2 \\ 0 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 5 & 0 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 23 \\ 0 & 12 \end{bmatrix}$$

- **Row & Column Labels**

Database Table Multiply

$$\begin{array}{l} \\ \text{word|coffee} \\ \text{word|desert} \end{array} \begin{array}{c} \begin{array}{cccc} \text{tod|0500} & \text{tod|0800} & \text{tod|0900} & \text{tod|1400} \end{array} \\ \begin{bmatrix} 6 & 5 & 0 & 2 \\ 0 & 4 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{l} \\ \text{tod|0500} \\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array} \begin{array}{c} \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 5 & 0 \\ 3 & 4 \end{bmatrix} \end{array} = \begin{array}{l} \\ \text{word|coffee} \\ \text{word|desert} \end{array} \begin{array}{c} \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{bmatrix} 6 & 23 \\ 0 & 12 \end{bmatrix} \end{array}$$

Matrix Multiply on Big Data

Traditional Matrix Multiply: $AB = C$

$$\begin{bmatrix} 6 & 5 & 0 & 2 \\ 0 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 5 & 0 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 23 \\ 0 & 12 \end{bmatrix}$$

- **Row & Column Labels**
- **Sparse**

Database Table Multiply

$$\begin{array}{l} \\ \text{word|coffee} \\ \text{word|desert} \end{array} \begin{array}{c} \begin{array}{cccc} \text{tod|0500} & \text{tod|0800} & & \text{tod|1400} \end{array} \\ \begin{bmatrix} 6 & 5 & & 2 \\ & 4 & & \end{bmatrix} \end{array} \begin{array}{l} \\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array} \begin{array}{c} \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{bmatrix} & 3 \\ 5 & \\ 3 & 4 \end{bmatrix} \end{array} = \begin{array}{l} \\ \text{word|coffee} \\ \text{word|desert} \end{array} \begin{array}{c} \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{bmatrix} 6 & 23 \\ & 12 \end{bmatrix} \end{array}$$

Matrix Multiply on Big Data

Traditional Matrix Multiply: $AB = C$

$$\begin{bmatrix} 6 & 5 & 0 & 2 \\ 0 & 4 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 3 \\ 5 & 0 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 23 \\ 0 & 12 \end{bmatrix}$$

- ➢ **Row & Column Labels**
- ➢ **Sparse**

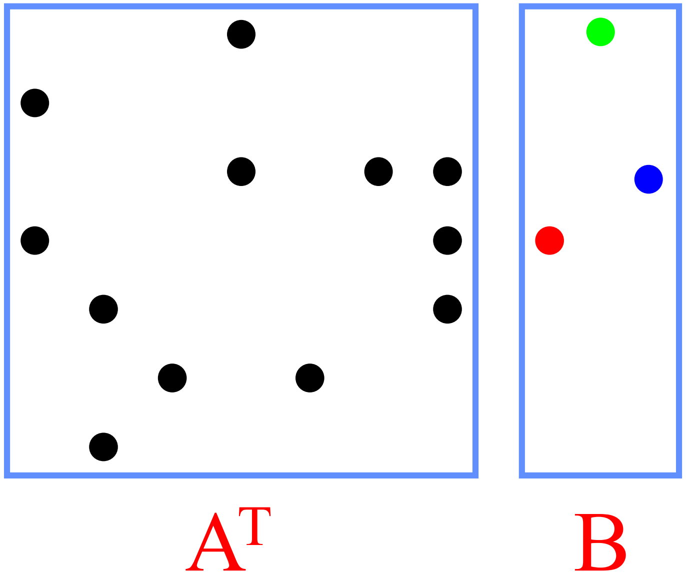
➔ **Associative Array Mathematics**[1]

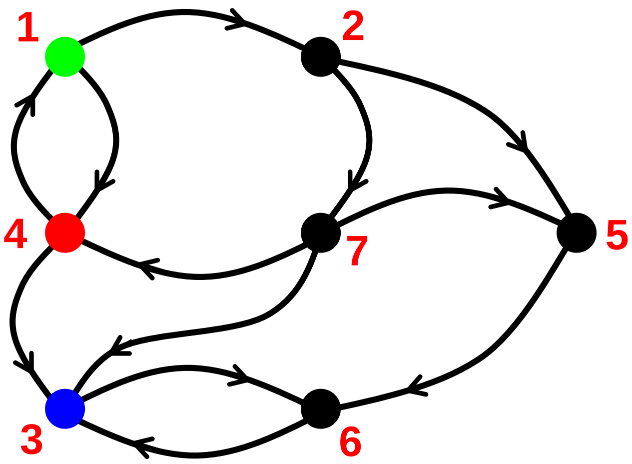
Database Table Multiply

$$\begin{array}{r c} & \begin{array}{ccc} \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \end{array} \\ \begin{array}{r} \text{word|coffee} \\ \text{word|desert} \end{array} & \begin{bmatrix} 6 & 5 & 2 \\ & 4 & \end{bmatrix} \end{array} \begin{array}{r c} & \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{array}{r} \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array} & \begin{bmatrix} & 3 \\ 5 & \\ 3 & 4 \end{bmatrix} \end{array} = \begin{array}{r c} & \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{array}{r} \text{word|coffee} \\ \text{word|desert} \end{array} & \begin{bmatrix} 6 & 23 \\ & 12 \end{bmatrix} \end{array}$$

[1] J. Kepner and V. Gadepally. "*Adjacency matrices, incidence matrices, database schemas, and associative arrays*" in International Parallel & Distributed Processing Symposium Workshops (IPDPSW). IEEE, 2014

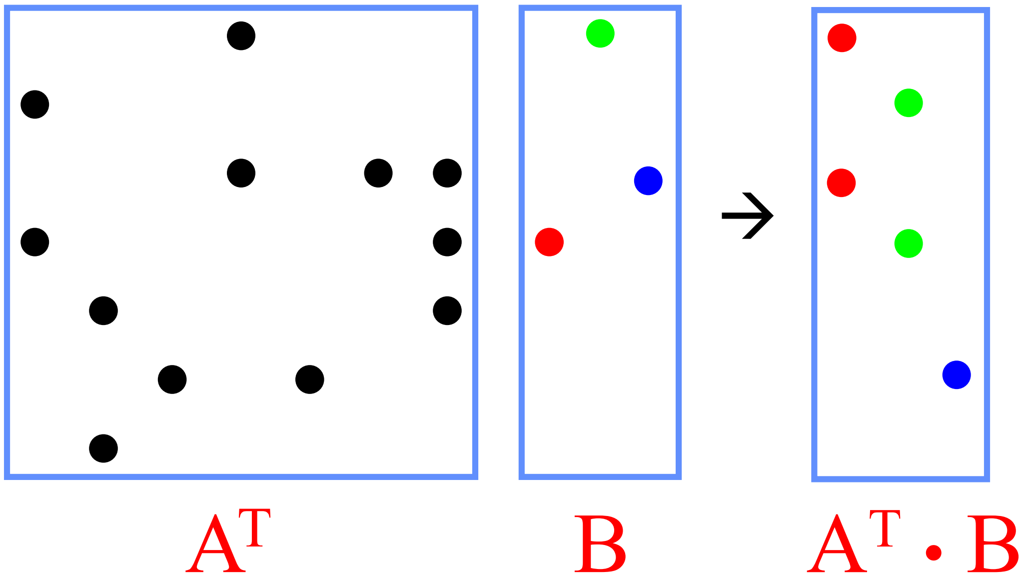
Application: Multi-Source Breadth-First Search

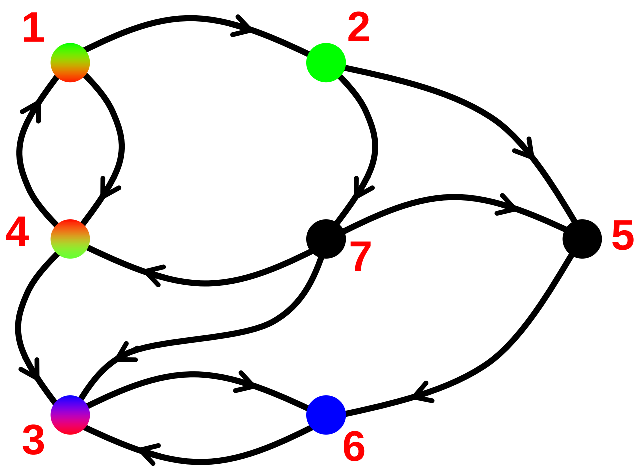




- Sparse array representation => space efficient
- Sparse matrix-matrix multiplication => work efficient
- Three possible levels of parallelism: searches, vertices, edges
- Basis for a wide range of graph algorithms

Application: Multi-Source Breadth-First Search





- Sparse array representation => space efficient
- Sparse matrix-matrix multiplication => work efficient
- Three possible levels of parallelism: searches, vertices, edges
- Basis for a wide range of graph algorithms

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Background on Accumulo

Key					Value
Row ID	Column			Timestamp	
	Family	Qualifier	Visibility		

Best for:

- Large, de-normalized tables (NoSQL)
- Hadoop HDFS / Java ecosystem
- Huge data volume – TBs to PBs
- Cell-level visibility
- Robust horizontal scaling

- Row store by default
  - Scan over rows for O(log n) lookup & sorted order
  - Log-structured Merge Tree design
- Iterator processing framework

Background on Accumulo

Key					Value
Row ID	Column			Timestamp	
	Family	Qualifier	Visibility		

Best for:

- **Large, de-normalized tables (NoSQL)**
- **Hadoop HDFS / Java ecosystem**
- **Huge data volume – TBs to PBs**
- **Cell-level visibility**
- **Robust horizontal scaling**

Use Transpose Tables
see D4M Schema[1]

- **Row store by default**
  - **Scan over rows for O(log n) lookup & sorted order**
  - **Log-structured Merge Tree design**
- **Iterator processing framework**

[1]*D4M 2.0 Schema: A General Purpose High Performance Schema for the Accumulo Database*
Kepner et al, IEEE HPEC 2013

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Table Multiply Before Graphulo

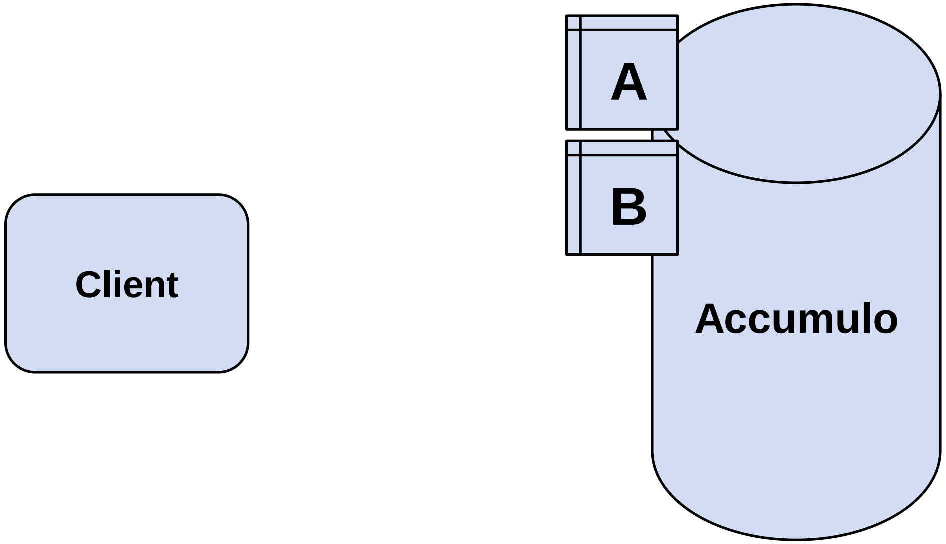


Table Multiply Before Graphulo

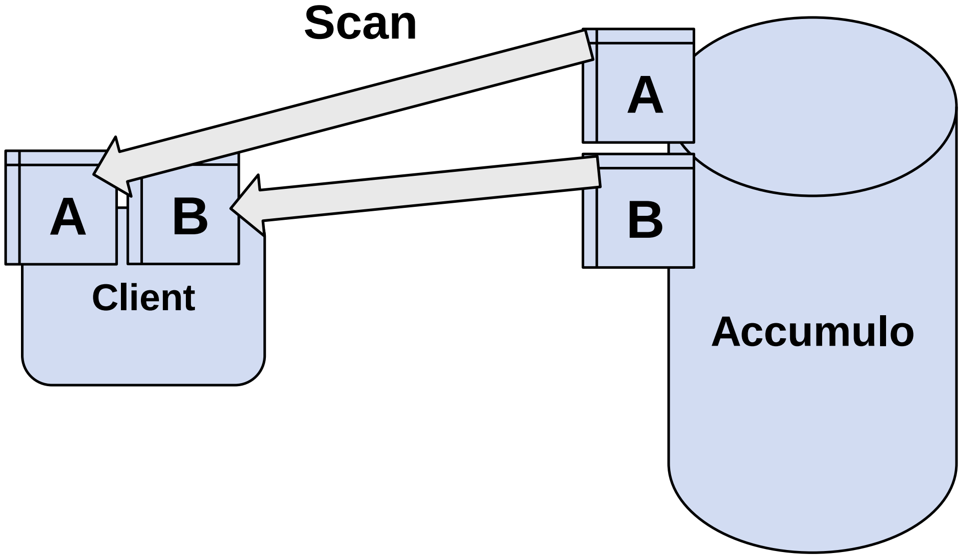
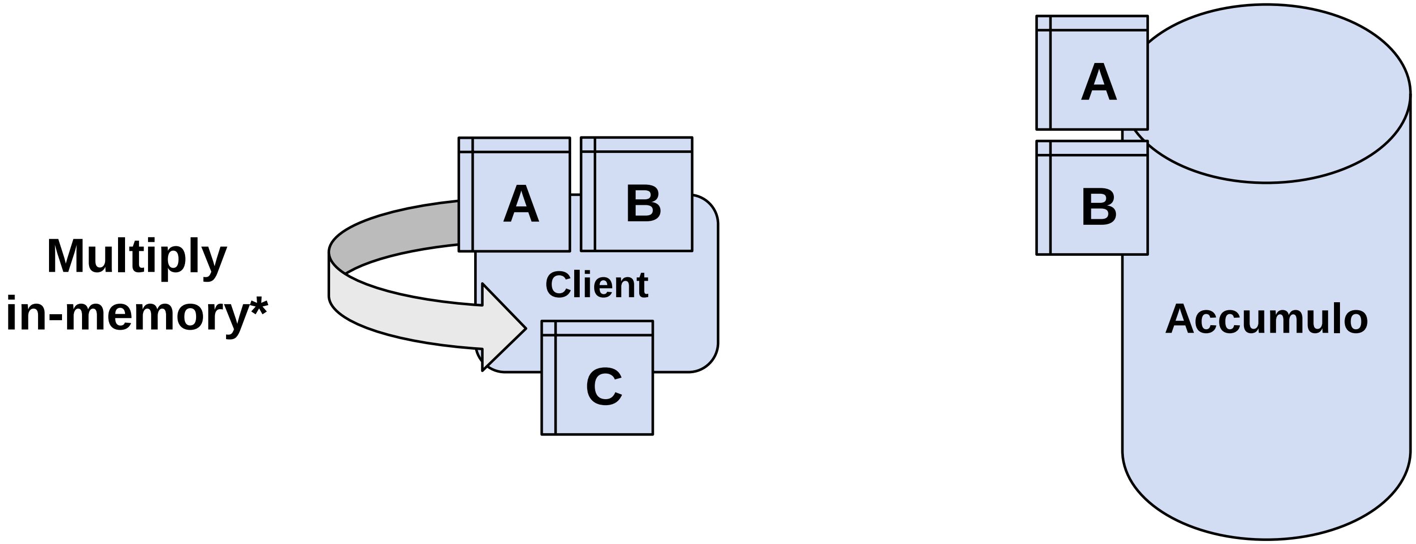


Table Multiply Before Graphulo



*Blocked algorithms exist for large tables at reduced efficiency

Table Multiply Before Graphulo

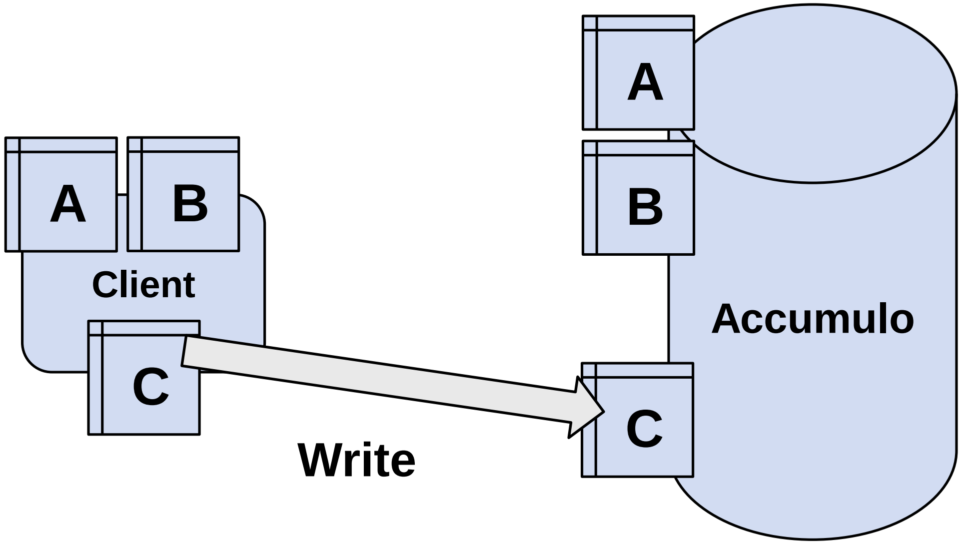
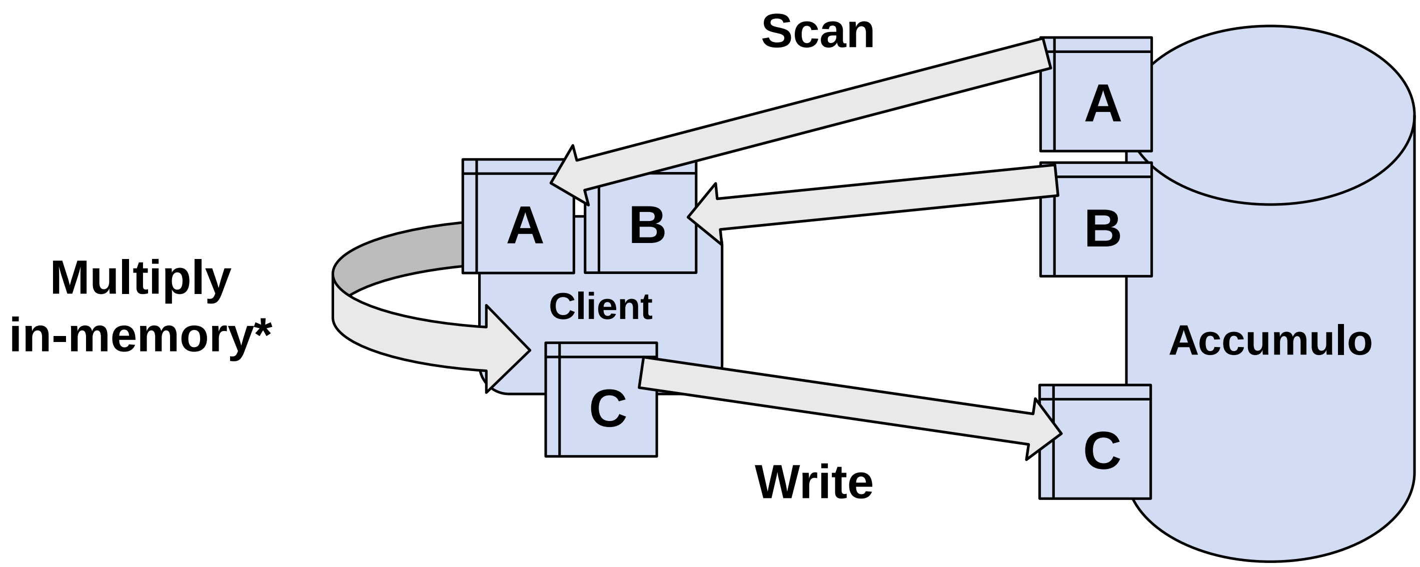


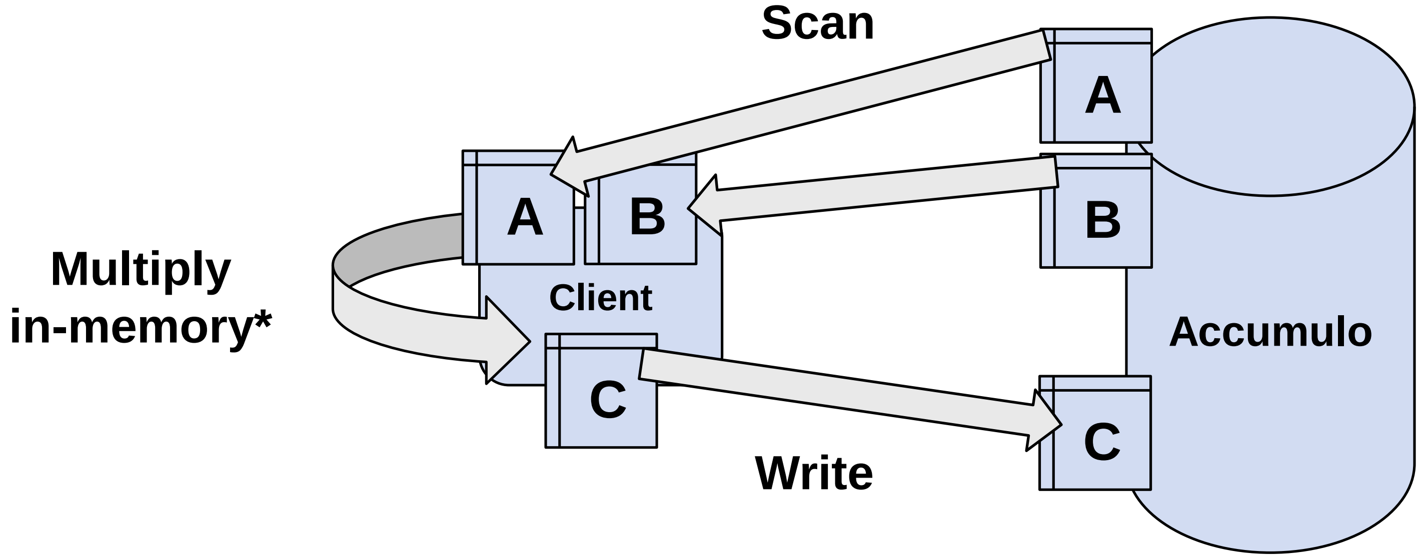
Table Multiply Before Graphulo



Old: DB = Indexed Storage

*Blocked algorithms exist for large tables at reduced efficiency

Table Multiply Before Graphulo



Old: DB = Indexed Storage

New: DB = Indexed Storage + Computation Engine

***Blocked algorithms exist for large tables at reduced efficiency**

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Inner Product

$$
\begin{array}{c|ccc}
 & \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \\
\hline
\text{word|coffee} & \boxed{6} & \boxed{5} & \boxed{2} \\
\text{word|desert} & & 4 &
\end{array}
\;
\begin{array}{c|cc}
 & \text{word|dew} & \text{word|hot} \\
\hline
\text{tod|0800} & & 3 \\
\text{tod|0900} & \boxed{5} & \\
\text{tod|1400} & \boxed{3} & 4
\end{array}
=
\begin{array}{c|c}
 & \text{word|dew} \\
\hline
\text{word|coffee} & \boxed{6}
\end{array}
$$

1

for $i = 1 : N$ = 2
 for $j = 1 : L$ = 2
  for $k = 1 : M$ = 4
   $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C}(i,j) = \bigoplus_{k=1}^{M} \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$$

Inner Product

$$
\begin{array}{r}
 \\ \text{word|coffee} \\ \text{word|desert}
\end{array}
\begin{array}{c}
\begin{array}{ccc} \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \end{array} \\
\begin{bmatrix} \mathbf{6} & \mathbf{5} & \mathbf{2} \\ & \mathbf{4} & \end{bmatrix}
\end{array}
\begin{array}{r}
 \\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix}
\end{array}
=
\begin{array}{r}
 \\ \text{word|coffee}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{① word|dew} & \text{word|hot ②} \end{array} \\
\begin{bmatrix} \mathbf{6} & \boxed{\mathbf{23}} \end{bmatrix}
\end{array}
$$

1st Scan

for $i = 1 : N$ = 2

for $j = 1 : L$ = 2

for $k = 1 : M$ = 4

$\mathbf{C}(i,j) \oplus = \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C}(i,j) = \bigoplus_{k=1}^{M} \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$$

Inner Product

$$\begin{array}{r|ccc} & \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \\ \hline \text{word|coffee} & \mathbf{6} & \mathbf{5} & \mathbf{2} \\ \text{word|desert} & & \mathbf{4} & \end{array} \quad \begin{array}{r|cc} & \text{word|dew} & \text{word|hot} \\ \hline \text{tod|0800} & & \mathbf{3} \\ \text{tod|0900} & \mathbf{5} & \\ \text{tod|1400} & \mathbf{3} & \mathbf{4} \end{array} = \begin{array}{r|cc} & 1 \ \text{word|dew} & \text{word|hot} \ 2 \\ \hline \text{word|coffee} & \mathbf{6} & \mathbf{23} \\ & \boxed{} & \end{array}$$

for $i = 1 : N$ = 2
  for $j = 1 : L$ = 2
    for $k = 1 : M$ = 4
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C}(i,j) = \bigoplus_{k=1}^{M} \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$$

Inner Product

$$
\begin{array}{r|ccc|}
 & \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \\
\text{word|coffee} & 6 & 5 & 2 \\
\text{word|desert} & & \boxed{4} &
\end{array}
\;
\begin{array}{r|cc|}
 & \text{word|dew} & \text{word|hot} \\
\text{tod|0800} & & 3 \\
\text{tod|0900} & 5 & \\
\text{tod|1400} & 3 & 4
\end{array}
=
\begin{array}{r|cc|}
 & \text{word|dew} & \text{word|hot} \\
\text{word|coffee} & 6 & 23 \\
\text{word|desert} & & \boxed{12}
\end{array}
$$

2nd Scan

for $i = 1 : N$ = 2

  for $j = 1 : L$ = 2

    for $k = 1 : M$ = 4

      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C}(i,j) = \bigoplus_{k=1}^{M} \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$$

Inner Product

+ Write locality (sorted)
+ Pre-sum partial products
  (3 entries written)

– N scans over table B

$$
\begin{array}{r}
 \\
\text{word|coffee} \\
\text{word|desert}
\end{array}
\begin{array}{c}
\begin{array}{ccc} \text{tod|0500} & \text{tod|0800} & \text{tod|1400} \end{array} \\
\begin{bmatrix} 6 & 5 & 2 \\ & \boxed{4} & \end{bmatrix}
\end{array}
\begin{array}{r}
 \\
\text{tod|0800} \\
\text{tod|0900} \\
\text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} & 3 \\ 5 & \\ 3 & 4 \end{bmatrix}
\end{array}
=
\begin{array}{r}
 \\
\text{word|coffee} \\
\text{word|desert}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} 6 & 23 \\ & \boxed{12} \end{bmatrix}
\end{array}
$$

2nd Scan

$\textbf{for } i = 1 : N$ = 2
  $\textbf{for } j = 1 : L$ = 2
    $\textbf{for } k = 1 : M$ = 4
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C}(i,j) = \bigoplus_{k=1}^{M} \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$$

Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Outer Product

Now explicitly showing A^T

$$
\begin{array}{r}
 \\
\text{tod|0500} \\
\text{tod|0800} \\
 \\
\text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\
\begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ & \\ \mathbf{2} & \end{bmatrix}
\end{array}
\quad
\begin{array}{r}
 \\
 \\
\text{tod|0800} \\
\text{tod|0900} \\
\text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} & \\ & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix}
\end{array}
=
\begin{bmatrix} \quad & \quad \end{bmatrix}
$$

$$
\begin{aligned}
&\textbf{for } k = 1:M \text{ = 4} \\
&\quad \textbf{for } i = 1:N \text{ = 2} \\
&\quad\quad \textbf{for } j = 1:L \text{ = 2} \\
&\quad\quad\quad \mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)
\end{aligned}
$$

$$
\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)
$$

Outer Product

1. Align Rows

$$\begin{array}{rc} & \begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\ \begin{array}{r} \text{tod|0500} \\ \text{tod|0800} \\ \text{tod|1400} \end{array} & \begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ \mathbf{2} & \end{bmatrix} \end{array} \begin{array}{rc} & \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{array}{r} \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array} & \begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix} \end{array} = \begin{bmatrix} & \end{bmatrix}$$

for $k = 1 : M$ = 4
 for $i = 1 : N$ = 2
  for $j = 1 : L$ = 2
   $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

1. Align Rows

$$
\begin{array}{r c}
 & \begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\
\begin{array}{r} \text{tod|0500} \\ \text{tod|0800} \\ \text{tod|1400} \end{array} &
\begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ \mathbf{2} & \end{bmatrix}
\end{array}
\begin{array}{r c}
 & \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{array}{r} \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array} &
\begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix}
\end{array}
= \begin{bmatrix} \quad & \quad \end{bmatrix}
$$

for $k = 1 : M$ = 4
  for $i = 1 : N$ = 2
    for $j = 1 : L$ = 2
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

2. Cartesian Product

$$
\begin{array}{c|cc}
 & \text{word|coffee} & \text{word|desert} \\
\hline
\text{tod|0500} & \mathbf{6} & \\
\text{tod|0800} & \boxed{\mathbf{5}} & \mathbf{4} \\
\text{tod|1400} & \mathbf{2} &
\end{array}
\quad
\begin{array}{c|cc}
 & \text{word|dew} & \text{word|hot} \\
\hline
\text{tod|0800} & & \boxed{\mathbf{3}} \\
\text{tod|0900} & \mathbf{5} & \\
\text{tod|1400} & \mathbf{3} & \mathbf{4}
\end{array}
=
\begin{array}{c|c}
 & \text{word|hot} \\
\hline
\text{word|coffee} & \boxed{\mathbf{15}}
\end{array}
$$

1

for $k = 1 : M$ = 4
  for $i = 1 : N$ = 2
    for $j = 1 : L$ = 2
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

2. Cartesian Product

$$
\begin{array}{r}
 \\ \text{tod|0500} \\ \text{tod|0800} \\ \\ \text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\
\begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \boxed{\mathbf{4}} \\ & \\ \mathbf{2} & \end{bmatrix}
\end{array}
\quad
\begin{array}{r}
 \\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} & \boxed{\mathbf{3}} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix}
\end{array}
=
\begin{array}{r}
 \\ \text{word|coffee} \\ \text{word|desert}
\end{array}
\begin{array}{c}
\text{word|hot} \\
\begin{bmatrix} \mathbf{15} \\ \boxed{\mathbf{12}} \end{bmatrix}
\end{array}
$$

1

2

for $k = 1 : M$ = 4
    for $i = 1 : N$ = 2
        for $j = 1 : L$ = 2
            $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

1. Align Rows

$$
\begin{array}{r}
\\ \text{tod|0500} \\ \text{tod|0800} \\ \\ \text{tod|1400}
\end{array}
\begin{array}{cc}
\text{word|coffee} & \text{word|desert} \\
\left[\begin{array}{cc} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ & \\ \mathbf{2} & \end{array}\right]
\end{array}
\quad
\begin{array}{r}
\\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400}
\end{array}
\begin{array}{cc}
\text{word|dew} & \text{word|hot} \\
\left[\begin{array}{cc} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{array}\right]
\end{array}
=
\begin{array}{r}
\\ \text{word|coffee} \\ \text{word|desert}
\end{array}
\begin{array}{c}
\text{word|hot} \\
\left[\begin{array}{c} \mathbf{15} \\ \mathbf{12} \end{array}\right]
\end{array}
\begin{array}{l}
\\ 1 \\ 2
\end{array}
$$

for $k = 1 : M$ = 4
  for $i = 1 : N$ = 2
    for $j = 1 : L$ = 2
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

1. Align Rows

$$
\begin{array}{rc}
 & \begin{array}{cc} \text{word}\mid\text{coffee} & \text{word}\mid\text{desert} \end{array} \\
\begin{array}{r} \text{tod}\mid\text{0500} \\ \text{tod}\mid\text{0800} \\ \text{tod}\mid\text{1400} \end{array} &
\begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ \mathbf{2} & \end{bmatrix}
\end{array}
\quad \Rightarrow \quad
\begin{array}{rc}
 & \begin{array}{cc} \text{word}\mid\text{dew} & \text{word}\mid\text{hot} \end{array} \\
\begin{array}{r} \text{tod}\mid\text{0800} \\ \text{tod}\mid\text{0900} \\ \text{tod}\mid\text{1400} \end{array} &
\begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \mathbf{4} \end{bmatrix}
\end{array}
=
\begin{array}{rc}
 & \begin{array}{c} \text{word}\mid\text{hot} \end{array} \\
\begin{array}{r} \text{word}\mid\text{coffee} \\ \text{word}\mid\text{desert} \end{array} &
\begin{bmatrix} \mathbf{15} \\ \mathbf{12} \end{bmatrix}
\end{array}
\begin{array}{c} 1 \\ 2 \end{array}
$$

for $k = 1 : M$ = 4
for $i = 1 : N$ = 2
for $j = 1 : L$ = 2
$\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

2. Cartesian Product

$$
\begin{array}{r} \\ \text{tod|0500} \\ \text{tod|0800} \\ \\ \text{tod|1400} \end{array}
\begin{array}{c} \begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\ \begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ & \\ \boxed{\mathbf{2}} & \end{bmatrix} \end{array}
\quad
\begin{array}{r} \\ \text{tod|0800} \\ \text{tod|0900} \\ \text{tod|1400} \end{array}
\begin{array}{c} \begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\ \begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \boxed{\mathbf{3}} & \mathbf{4} \end{bmatrix} \end{array}
=
\begin{array}{r} \\ \text{word|coffee} \\ \text{word|desert} \end{array}
\begin{array}{c} \begin{array}{cc} 3\ \text{word|dew} & \text{word|hot}\ 1 \end{array} \\ \begin{bmatrix} \boxed{\mathbf{6}} & \mathbf{15} \\ & \mathbf{12} \end{bmatrix} \\ 2 \end{array}
$$

$$
\begin{aligned}
&\textbf{for } k = 1:M = 4 \\
&\quad \textbf{for } i = 1:N = 2 \\
&\quad\quad \textbf{for } j = 1:L = 2 \\
&\quad\quad\quad \mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)
\end{aligned}
$$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

2. Cartesian Product

***Lazy** $\oplus$:
Accumulo stores both 15 and 8 until next scan or compaction

$$
\begin{array}{r}
 \\
\text{tod|0500} \\
\text{tod|0800} \\
\\
\text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|coffee} & \text{word|desert} \end{array} \\
\begin{bmatrix} \mathbf{6} & \\ \mathbf{5} & \mathbf{4} \\ & \\ \boxed{\mathbf{2}} & \end{bmatrix}
\end{array}
\quad
\begin{array}{r}
 \\
\text{tod|0800} \\
\text{tod|0900} \\
\text{tod|1400}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} & \mathbf{3} \\ \mathbf{5} & \\ \mathbf{3} & \boxed{\mathbf{4}} \end{bmatrix}
\end{array}
=
\begin{array}{r}
 \\
\text{word|coffee} \\
\text{word|desert}
\end{array}
\begin{array}{c}
\begin{array}{cc} \text{word|dew} & \text{word|hot} \end{array} \\
\begin{bmatrix} \mathbf{6} & \boxed{\mathbf{23}} \\ & \mathbf{12} \end{bmatrix}
\end{array}
*
$$

for $k = 1:M$ = 4
  for $i = 1:N$ = 2
    for $j = 1:L$ = 2
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Outer Product

- **No write locality; unsorted writes**
- **Hard to pre-sum partial products (4 entries written)**

+ Single scan over table B

***Lazy ⊕:**
Accumulo stores both 15 and 8 until next scan or compaction

$$
\begin{array}{r}
\\
\text{tod|0500} \\
\text{tod|0800} \\
\\
\text{tod|1400}
\end{array}
\begin{array}{cc}
\text{word|coffee} & \text{word|desert} \\
\left[\begin{array}{c} \mathbf{6} \\ \mathbf{5} \\ \\ \boxed{\mathbf{2}} \end{array}\right.
& \left.\begin{array}{c} \\ \mathbf{4} \\ \\ \\ \end{array}\right]
\end{array}
\quad
\begin{array}{r}
\\
\text{tod|0800} \\
\text{tod|0900} \\
\text{tod|1400}
\end{array}
\begin{array}{cc}
\text{word|dew} & \text{word|hot} \\
\left[\begin{array}{c} \\ \mathbf{5} \\ \mathbf{3} \end{array}\right.
& \left.\begin{array}{c} \mathbf{3} \\ \\ \boxed{\mathbf{4}} \end{array}\right]
\end{array}
=
\begin{array}{r}
\\
\text{word|coffee} \\
\text{word|desert}
\end{array}
\begin{array}{cc}
\textcircled{3}\ \text{word|dew} & \text{word|hot}\ \textcircled{1} \\
\left[\begin{array}{c} \mathbf{6} \\ \\ \end{array}\right.
& \left.\begin{array}{c} \boxed{\mathbf{23}}\ \textcircled{4}\,* \\ \mathbf{12} \\ \textcircled{2} \end{array}\right]
\end{array}
$$

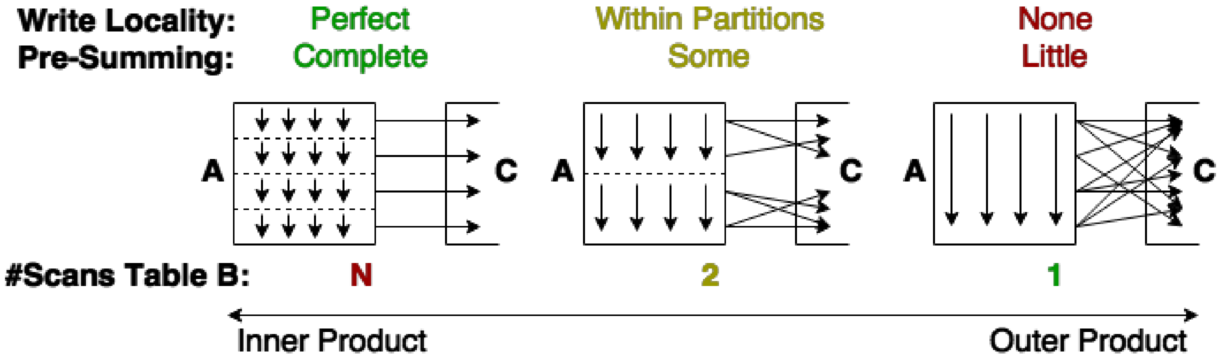
for $k = 1 : M$ = 4
  for $i = 1 : N$ = 2
    for $j = 1 : L$ = 2
      $\mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)$

$$\mathbf{C} = \bigoplus_{k=1}^{M} \mathbf{A}(:,k)\mathbf{B}(k,:)$$

Inner vs. Outer Product

- **Outer product best for Accumulo**
  - **Single pass over table B = single disk read**
  - **BatchWriter ingest handles unsorted writes**
  - **Combiners handle ⊕**
  - **Less extra partial products written for sparse data**

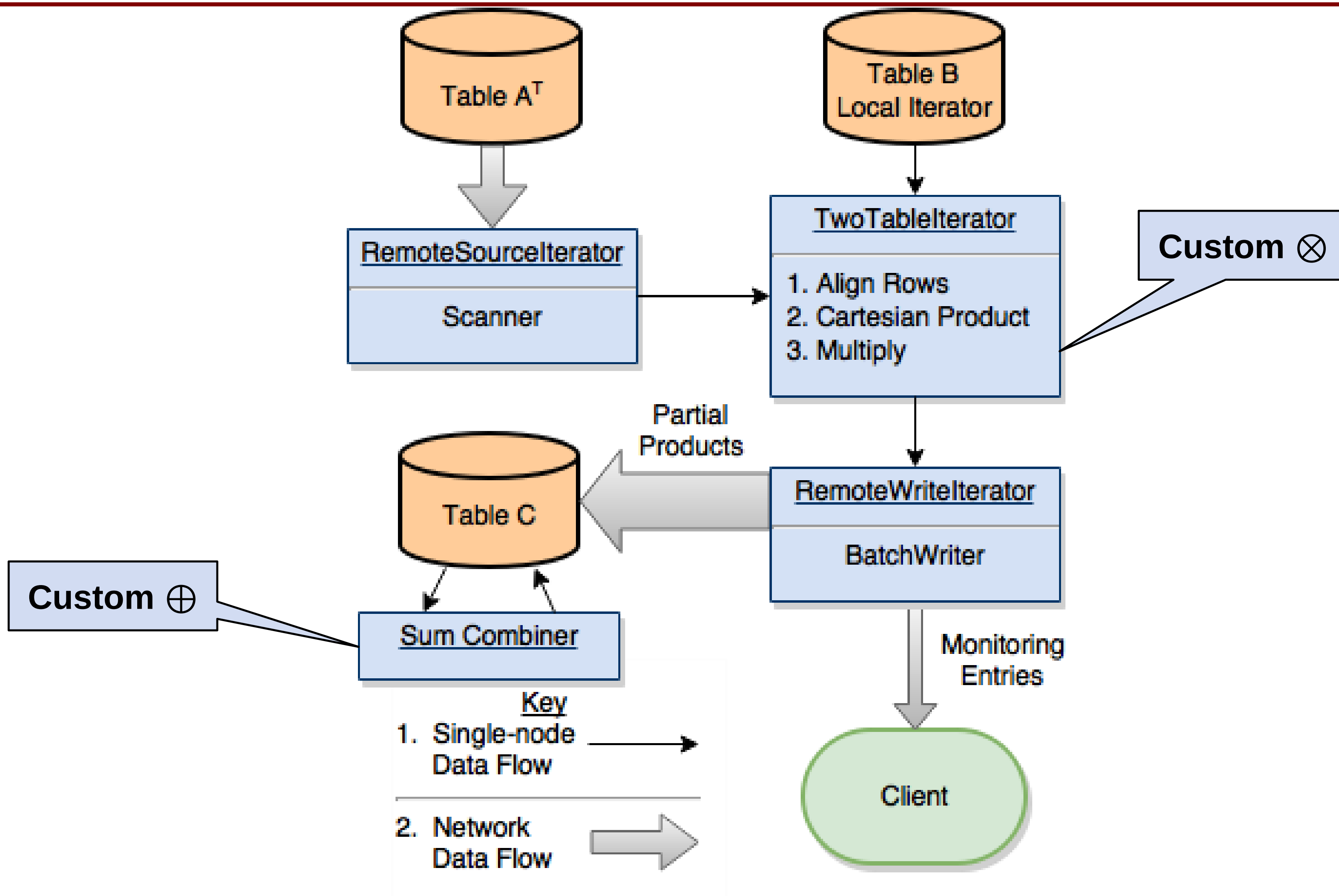
- **Inner product still has merit**
  - **Better for dense data**
  - **Hybrid 2D-like algorithm possible**



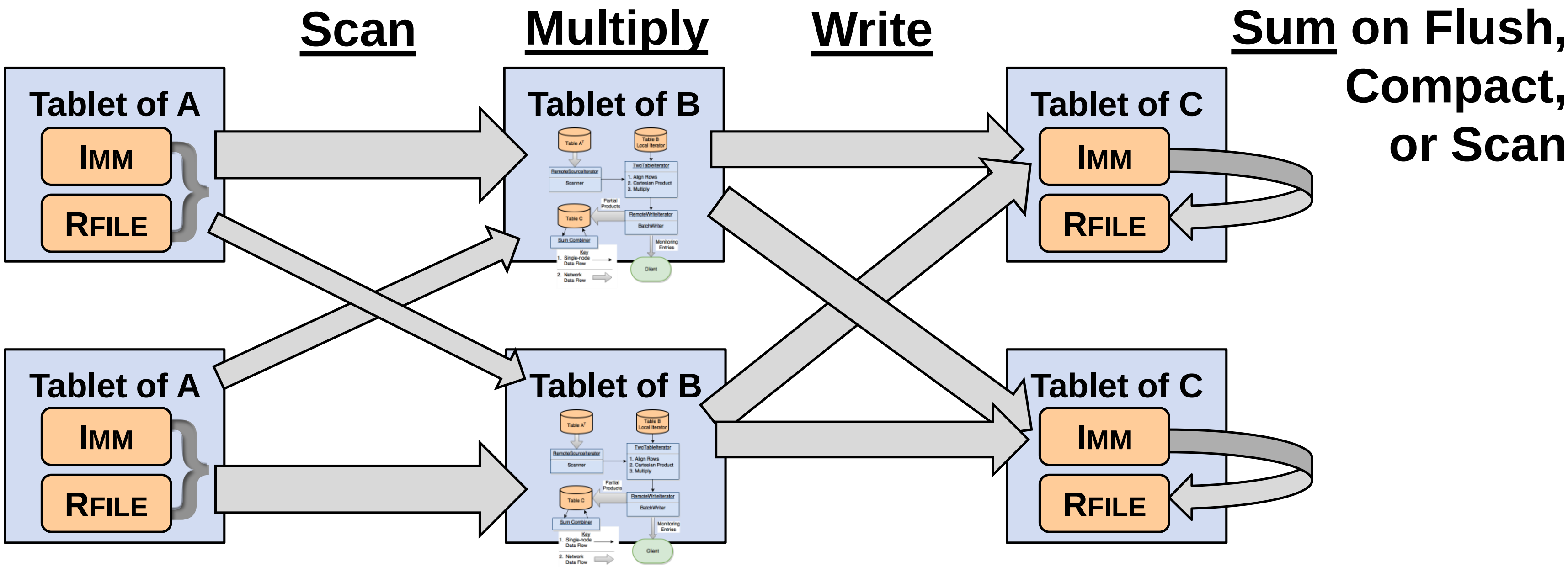
Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Outer Product in Graphulo Iterators



Accumulo Distributes Graphulo Iterators



Key

IMM: In-Memory Map
RFILE: Hadoop File

- Tablets can be hosted on any tablet server
  - Accumulo load balances tablet allocation
- Matrix multiply iterators run on B's tablets in parallel
  - Scan from A's tablets in parallel
  - BatchWrite to C's tablets in parallel

Outline

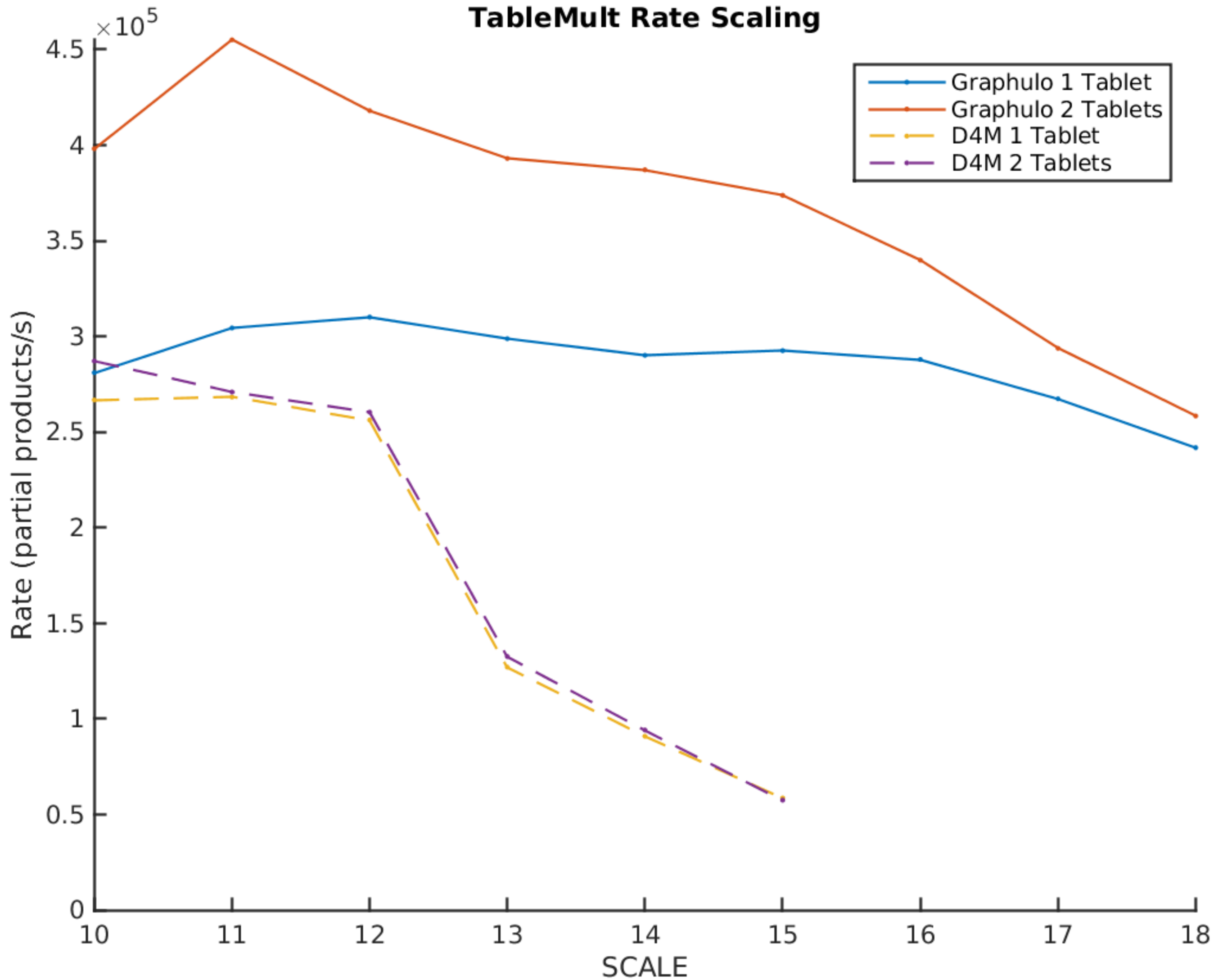
- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Performance Experiment

- **Compare to pre-Graphulo alternative:**
  - **D4M Matlab client as Middleman**
- **Scaled / Weak scaling study:**
  - **How multiply rate varies with increasing problem size at fixed resources**
  - **Ideal: constant multiply rate**
- **Fixed / Strong scaling study:**
  - **How multiply rate varies with increasing resources at fixed problem size**
  - **Ideal: multiply rate scales linearly with increasing resources**

- **Environment:**
  - **Laptop, 16GB RAM, 2 Dual-core i7 processors, Accumulo 1.6.1**
- **Vary problem size between SCALE 10 and 18**
  - **Unpermuted Power law graph generator**
  - **# of nodes in each input table is 2^{SCALE}. Used 16 edges/node**
- **Vary resources with # Accumulo Tablets (Varies # Threads)**

Performance Experiment

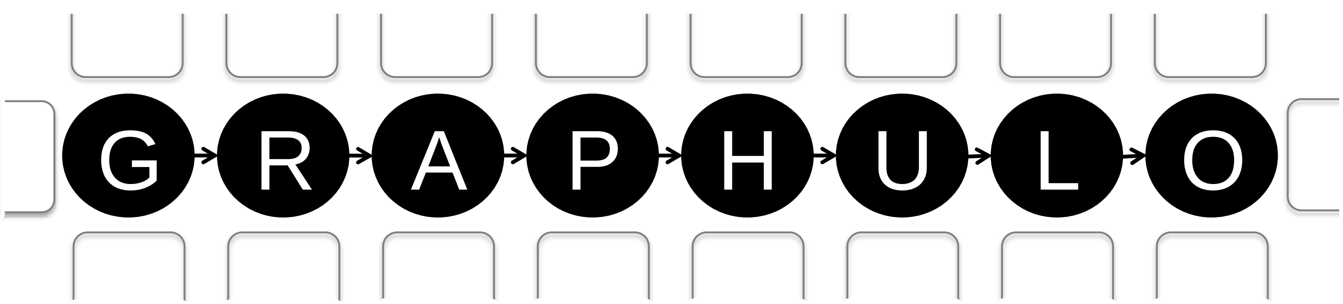


Outline

- Intro to Graphulo
- Intro to Matrix Multiply
- Intro to Accumulo
- Matrix Multiply pre-Graphulo
- Inner Product
- Outer Product
- Accumulo Implementation
- Performance
- Conclusions

Conclusion

- **Promising performance**
  - **Write rates near 400k / sec, near highest single-node recorded rates**
  - **Experiments on a larger cluster will confirm weak & strong scaling**
- **Outer product better suited to Accumulo**
  - **Hybrid inner-outer product algorithms worth studying**
- **Current Graphulo research is**
  - **implementing remaining GraphBLAS**
  - **developing graph algorithms**



Backup

TABLE I: Output Table **C** Sizes and Experiment Timings

SCALE	Entries in Table C		Graphulo 1 Tablet		D4M 1 Tablet		Graphulo 2 Tablets		D4M 2 Tablets	
	PartialProducts	AfterSum	Time (s)	Rate (pp/s)	Time (s)	Rate (pp/s)	Time (s)	Rate (pp/s)	Time (s)	Rate (pp/s)
10	8.05×10^5	2.69×10^5	2.87	2.81×10^5	3.02	2.67×10^5	2.02	3.98×10^5	2.80	2.87×10^5
11	2.36×10^6	8.15×10^5	7.76	3.04×10^5	8.80	2.68×10^5	5.19	4.55×10^5	8.72	2.71×10^5
12	6.82×10^6	2.43×10^6	2.20×10^1	3.10×10^5	2.66×10^1	2.56×10^5	1.63×10^1	4.18×10^5	2.62×10^1	2.60×10^5
13	1.91×10^7	7.04×10^6	6.40×10^1	2.99×10^5	1.50×10^2	1.27×10^5	4.86×10^1	3.93×10^5	1.44×10^2	1.33×10^5
14	5.27×10^7	2.00×10^7	1.82×10^2	2.90×10^5	5.79×10^2	9.09×10^4	1.36×10^2	3.87×10^5	5.59×10^2	9.42×10^4
15	1.47×10^8	5.83×10^7	5.03×10^2	2.93×10^5	2.51×10^3	5.86×10^4	3.94×10^2	3.74×10^5	2.56×10^3	5.75×10^4
16	4.00×10^8	1.63×10^8	1.39×10^3	2.88×10^5			1.18×10^3	3.40×10^5		
17	1.09×10^9	4.59×10^8	4.06×10^3	2.67×10^5			3.70×10^3	2.94×10^5		
18	2.94×10^9	1.28×10^9	1.21×10^4	2.42×10^5			1.14×10^4	2.58×10^5		

Inner-Outer Hybrid Algorithm

$$
\begin{aligned}
&\textbf{for } p = 1 : P \\
&\quad \textbf{for } k = 1 : M \\
&\quad\quad \textbf{for } i = \left(\left\lfloor \frac{(p-1)N}{P} \right\rfloor + 1 \right) : \left\lfloor \frac{pN}{P} \right\rfloor \\
&\quad\quad\quad \textbf{for } j = 1 : L \\
&\quad\quad\quad\quad \mathbf{C}(i,j) \oplus= \mathbf{A}(i,k) \otimes \mathbf{B}(k,j)
\end{aligned}
$$

P = N – Inner Product

P = 1 – Outer Product

D4M Schema for Sparse Arrays in Key/Value Databases (Accumulo)

Accumulo Table: `Ttranspose`

	01-01-2001	02-01-2001	03-01-2001
Col1|a	1		
Col1|b		1	
Col2|b		1	
Col2|c			1
Col3|a	1		
Col3|c			1

Input Data

Time	Col1	Col2	Col3
2001-01-01	a		a
2001-01-02	b	b	
2001-01-03		c	c

	Col1|a	Col1|b	Col2|b	Col2|c	Col3|a	Col3|c
01-01-2001	1				1	
02-01-2001		1	1			
03-01-2001				1		1

Accumulo Table: `T`

- **Tabular data expanded to create many type/value columns**
- **Transpose pairs allows quick look up of either row or column**

[1]*D4M 2.0 Schema: A General Purpose High Performance Schema for the Accumulo Database*
Kepner et al, IEEE HPEC 2013