In-Database Analytics for NoSQL Key-Value Stores

Dylan Hutchison, University of Washington

http://graphulo.mit.edu

Background: Key-Value Analytics?

Key-Value stores used for
> Scale-out to 1000s of machines
> Transparent layout, performance
> Fast key-value reads and writes

Problem: no support for complex analytics inside key-value stores

Example: **Browser Sessions** (e.g. browsing Amazon)

DATA TYPE	KEY	VALUE
Session	User/Session ID	Session Data (pages visited, shopping cart, …)

> Fast read-writes perfect for low-latency web server
> What if the website managers want to run analytics?

Simple filters, aggs — Query: find average session length ✓

Requires re-shuffle — Query: for each page, find average viewing time ✗ (unless you build an index)

Requires iteration — Query: build a histogram of users browsing certain pages, grouped by session age, over various time periods ✗

ML, matrix, graph alg. — Query: cluster users based on pages browsed. Recommend users to browse popular pages browsed in their cluster ✗

✓ = Supported inside key-value databases

✗ = Requires external system

Yet in-database analytics have many benefits:
1. Data Locality
2. Reuse Infrastructure
3. Indexed access, distributed execution

Key					Value
Row	Column			Timestamp	
	Family	Qualifier	Visibility		

Key

🟥 External Process
🟩 Accumulo Iterator

IMM In-Memory Map
RFILE Hadoop File
⟶ Intra-process Data Flow
⇨ Network Data Flow

Experiment Details

- 12 x m3.large Amazon nodes, each 7.5 GB mem, 2 vCPU, 30 GB SSD
- 8 workers, 3 coordinators, 1 monitor
- Graph500 power law matrix generator 2^{10} to 2^{19} rows, 16 nonzeros/row Skew!

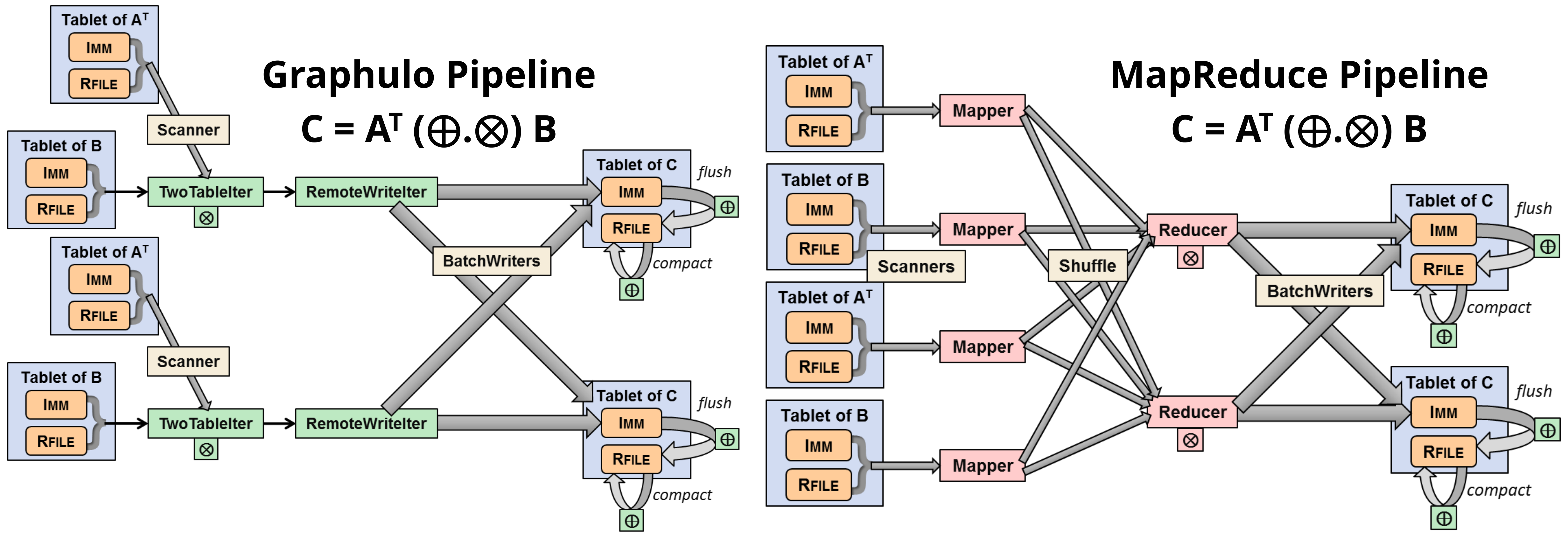
Past Work

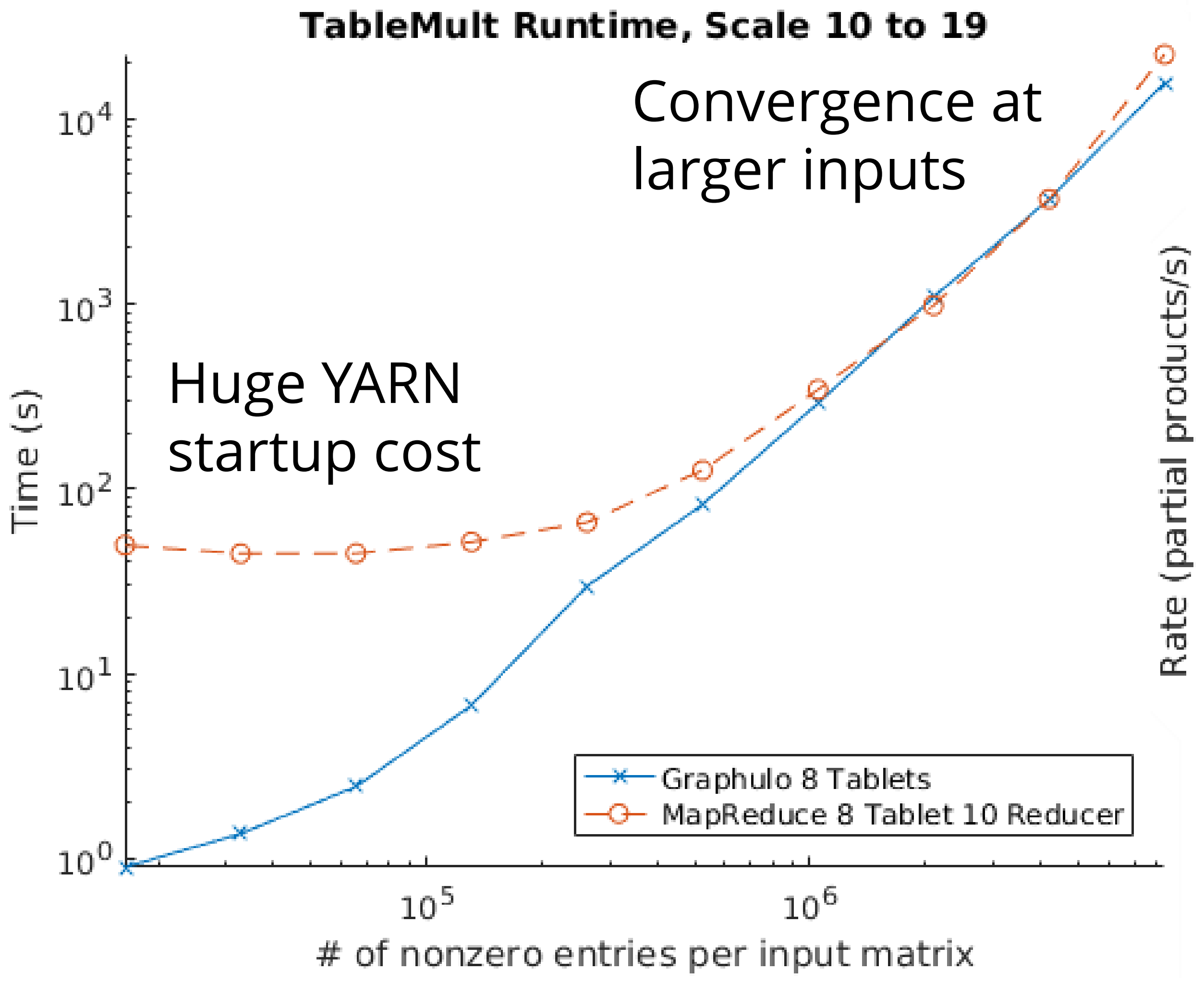
> Showed Graphulo faster than single-node in-memory LA packages on MxM (HPEC '15)
> Confirmed results for more complex I/O-bound, single-pass graph analytics (IPDPS '15, HPEC '16)
> Verified Graphulo scales with Accumulo as cluster size increases (HPEC '16)

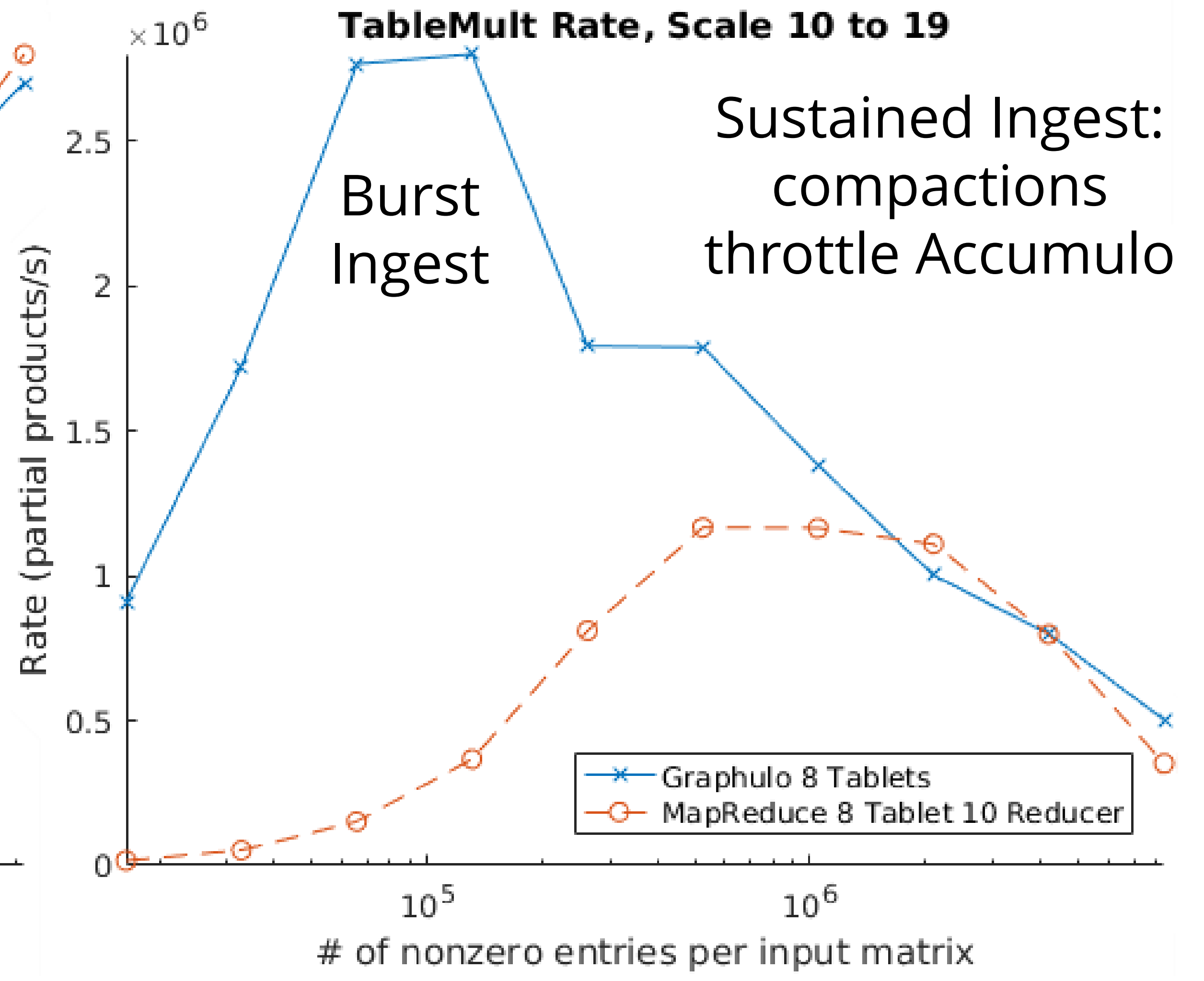
Experiment: Graphulo vs. MapReduce on Matrix Multiply

Goal: Compare Graphulo's in-database approach to an external distributed system

Test assumptions: "Use Accumulo for low-latency queries on subgraphs";
"Use MapReduce for high-throughput analytics" 🡐 Are these true?







Results

> Graphulo dominates at smaller problem sizes
> Graphulo & MapReduce equivalent at larger problem sizes

Guideline

> Graphulo best for **I/O-bound single-pass analytics**
> External systems best for CPU-bound or multi-pass analytics

Analytics inside Key-Value Stores

GRAPHULO — **Linear Algebra** in the Apache Accumulo NoSQL key-value store

RACO — **Relational Algebra** in the Apache Accumulo NoSQL key-value store (not this poster)

GraphBLAS Kernel
BuildMatrix (⊕)
ExtractTuples
MxM (⊕,⊗)
EwiseMult (⊗)
EwiseAdd (⊕)
Extract
Apply (f)
Assign
Reduce (⊕)
Transpose