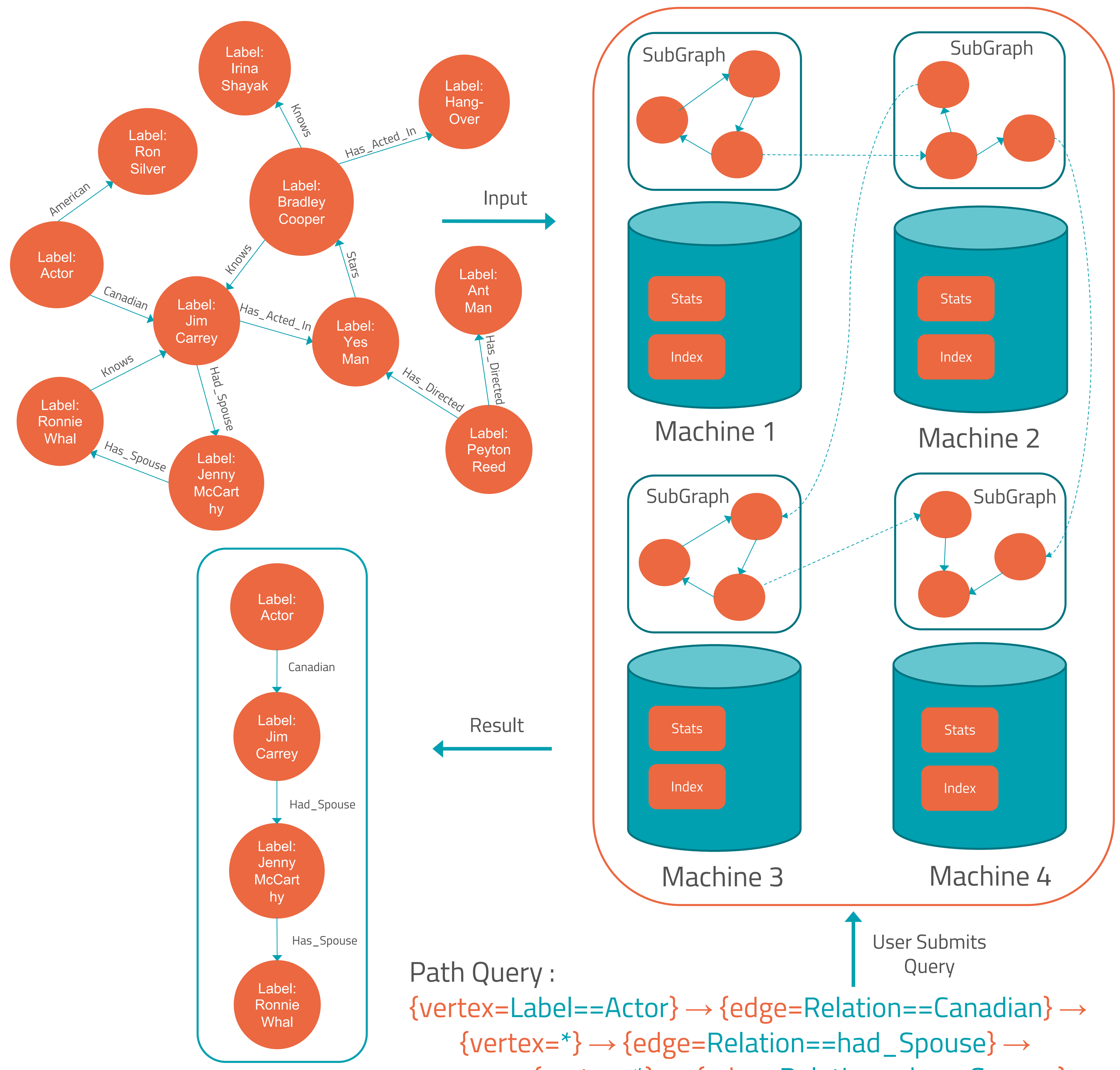
DREAM:Lab

Distributed Querying over Compressed Property Graphs

Swapnil Gandhi, Sayandip Sarkar, Abhilash Sharma and Yogesh Simmhan

Introduction

- Increasing trend toward representing semi-structured data as **Property Graphs** using compact in-memory store.

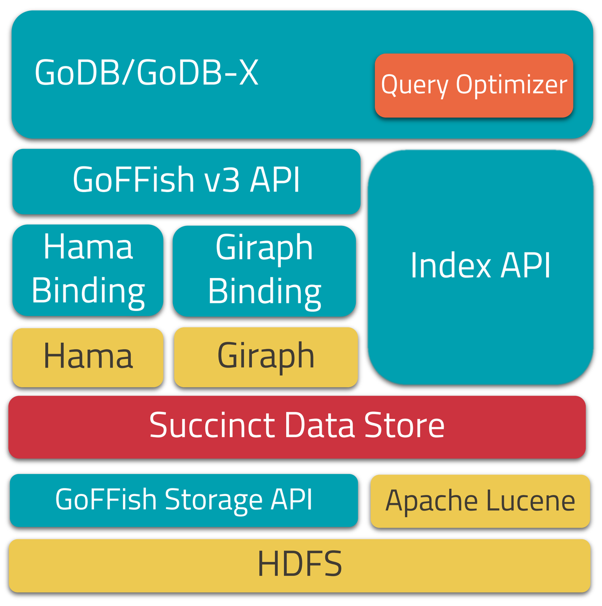


Path Query :
{vertex=Label==Actor} → {edge=Relation==Canadian} → {vertex=*} → {edge=Relation==had_Spouse} → {vertex=*} → {edge=Relation==has_Spouse} → {vertex=Label==Ronnie_Whal}

- Challenges:
  - Low latency interactive querying on compressed data
  - Scaling to Large Graphs (Billions of Vertices and Edges)
  - High Throughput
  - Graphs Keep Growing in Size

GoDB

- Builds upon GoFFish subgraph-centric batch processing framework
- Bulk Sync Parallel Execution Model
- Common graph query types: VE, Path, BFS, Reachability
- Heuristics based Cost Model to select best distributed query execution plan



GoDB-X

- Harnesses query execution model of GoDB
- Uses Succinct as an underlying Data Store
- Reduces memory footprint of Java objects but incurs higher traversal costs
- Offers an interesting trade-off between distributed memory utilized and query latency time

Data-Model

VertexID	Attribute List
4061105	countryID: US, numberOfClaims: 40
132768	countryID: FR, numberOfClaims: 7
696696	countryID: SE, numberOfClaims: 92

→

AttributeID	(Order, Delimiter)
countryID	(0, ¶)
numberOfClaims	(1,§)

\+

▪4061105¶US§40‡
▪132768¶FR§7‡
▪696696¶SE§92‡

Figure 1: VertexFile layout storing vertexID and property name-value pairs

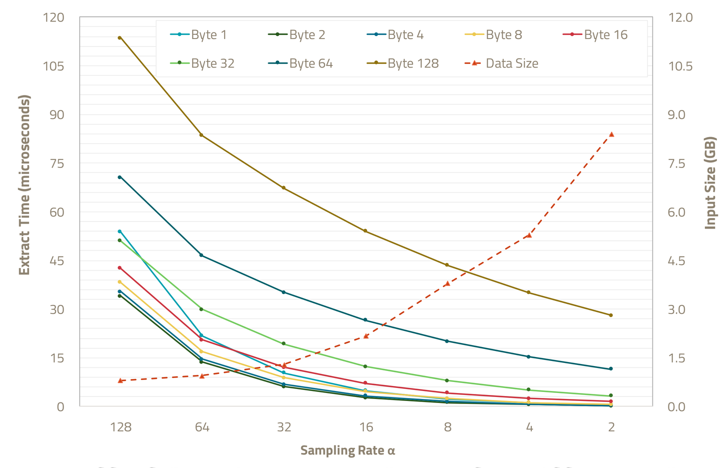
$\$S_1$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
$\$S_2$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
$\$S_3$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡

Figure 2: EdgeFile layout storing vertex adjacency list

Dataset

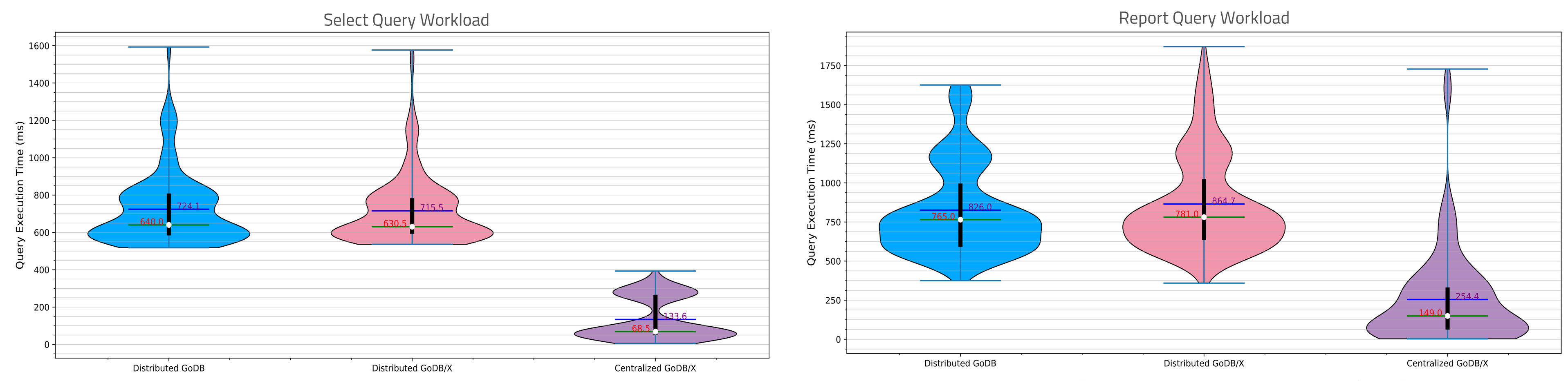
Dataset	Vertices	Edges	Properties	On-disk Size	Number of Select Queries	Number of Report Queries
CITP	3.7 M	16.5 M	2	1 GB	300	300
GITR	170 M	1.2 B	4	38 GB	120	480

Experimental Evaluation



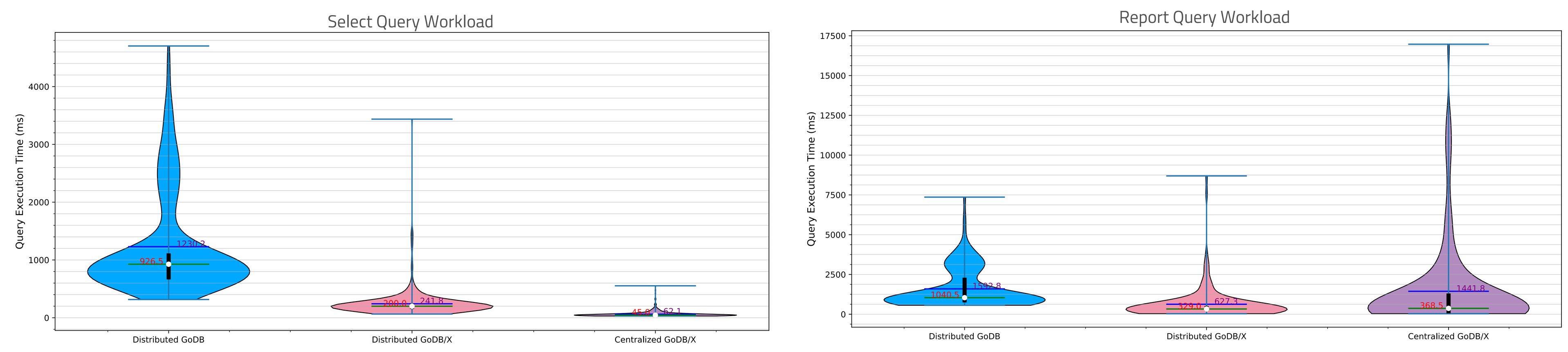
Space-Time trade-off of Succinct Data Store for different Sampling Rate α.
We adopt α=32 in our experiments

CITP Dataset



Though by using compressed data-model we were able to see >75% Reduction in Query Execution Time for Centralized GoDB-X, for Distributed (4 VMs) GoDB-X we see execution time to be marginally slower than GoDB.
Static Overheads of querying compressed data + network communication cost out-strips any advantages offered by parallel execution.

GITR Dataset



>50% Reduction in Query Execution Time for Distributed GoDB-X.
As more operations per Machine are required for GITR, we see advantages of parallel execution out-weighing communication costs and static overheads.

On-Going Work

- Improvements in Data Model

▪4061105¶US§40‡
▪132768¶FR§7‡
▪696696¶SE§92‡

→

4061105
132768
696696

\+

US
FR
SE

\+

40
7
92

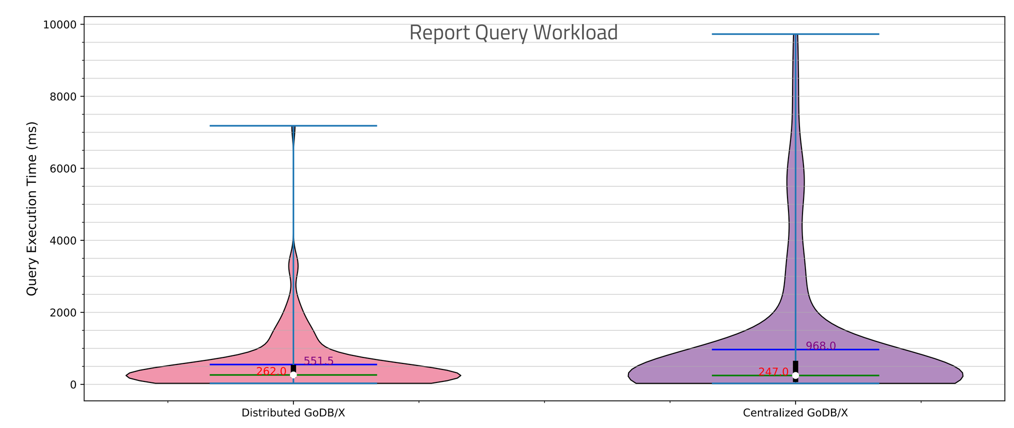
Figure 3: VertexFile further divided into separate columnar files for efficient Extract

$\$S_1$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
$\$S_2$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
$\$S_3$▪localEdgeCount⋆$ID_1:ID_2:ID_3 \ldots ID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡

→

▪localEdgeCount⋆$dID_1:dID_2:dID_3 \ldots dID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
▪localEdgeCount⋆$dID_1:dID_2:dID_3 \ldots dID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡
▪localEdgeCount⋆$dID_1:dID_2:dID_3 \ldots dID_M$ϴ$rD_1:rD_2:\ldots.rD_N$‡

Figure 4: EdgeFile Layout with Implicit VertexIDs



20%-32% Reduction in Query Execution Time compared to previous Data-Model and >65% Reduction compared to GoDB. Improvements in Data Model cause reduction in static overheads of Succinct Data Store Operations, though communication costs remain the same.

- Cost Model Inclusion
- Support query over Time-Series Graph with compound predicates
- Concurrent Queries Optimization

Summary

- Fewer Machines ➲ Smaller Memory Footprint ➲ Reduced Communication Costs ➲ Lower Query Latency

Distributed Research on Emerging Applications & Machines Lab
Department of Computational and Data Sciences
dream-lab.in | Indian Institute of Science, Bangalore

DREAM:Lab