ORACLE

PGX.D aDFS: An Almost Depth-First-Search Distributed Graph-Querying System

Vasileios Trigonakis, Jean-Pierre Lozi, Tomas Faltin, Nicholas P. Roth, Iraklis Psaroudakis, Arnaud Delamare, Vlad Haprian, Calin Iorgulescu, Petr Koupy, Jinsu Lee, Sungpack Hong, Hassan Chafi

Oracle Labs

Appeared in USENIX ATC ‘21

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions.

The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

GraalVM Native Image technology (including SubstrateVM) is early adopter technology. It is available only under an early adopter license and remains subject to potentially significant further changes, compatibility testing and certification.

Complexities in Graph-Query Execution

- Limited locality (especially in a distributed system)
- Intermediate (and final) result explosion

Need a distributed solution that is flexible and can handle the scale

Twitter graph

```
SELECT COUNT(*) MATCH (a)->()
+------------------------+
|    COUNT(*)            |
+------------------------+
| 1,468,365,182          |
+------------------------+
```

1 hop

```
SELECT COUNT(*) MATCH (a)->()->()
+------------------------+
|    COUNT(*)            |
+------------------------+
| 9,324,563,362,739  |
+------------------------+
```

2 hops

spoiler!
PGX.D aDFS
8 machines
~20 minutes
~8B matches/s

```
-- Info of authors who like each other and have < 10 years of age difference
SELECT a1.name, a2.name, a1.country = a2.country,
   ABS(a1.salary - a2.salary) AS salary_diff
MATCH (a1:author) -[:likes]-> (a2:author) -[:likes]-> (a1)
WHERE ABS(a1.age - a2.age) < 10
ORDER BY salary_diff DESC
```

Any user expression in projections and filters

Requires **homomorphic matching** and returns **all result permutations**

Agenda

1. Introduction / Motivation
2. aDFS Design
3. Evaluation
4. Conclusions

aDFS Design Principles

1. Asynchronous operation

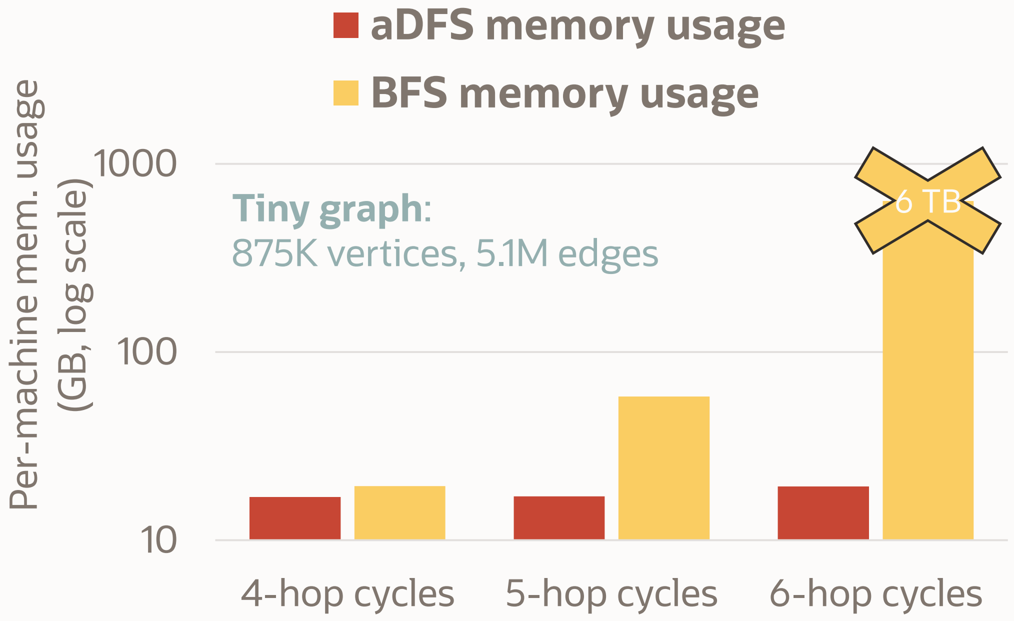
- Workers operate independently
  - on traversals where there is work
- Workers buffer and forget remote traversals

→Workers do not block due to remote communication

2. (Almost) Depth-first traversal

- Eager completion of matches
- Fine-grained flow control

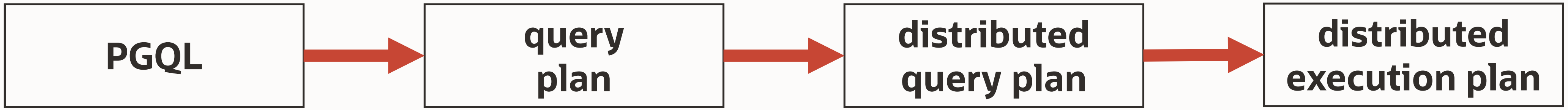
→ Control memory consumption



Tiny graph:
875K vertices, 5.1M edges

In-memory distributed execution with controllable memory usage

From a PGQL Query to an aDFS Execution Plan



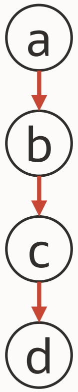
PGQL

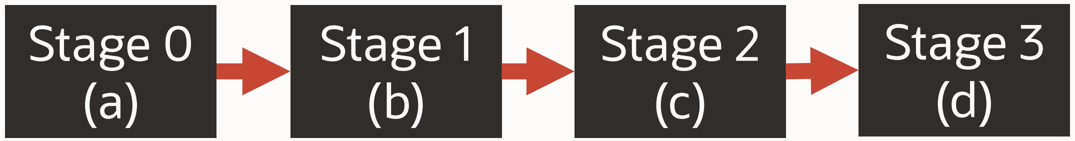
query plan

distributed query plan

distributed execution plan

```
SELECT ...
MATCH
  (a)->(b),
  (b)->(c),
  (c)->(d)
WHERE
  ...
```





Stage 0 (a)

Stage 1 (b)

Stage 2 (c)

Stage 3 (d)

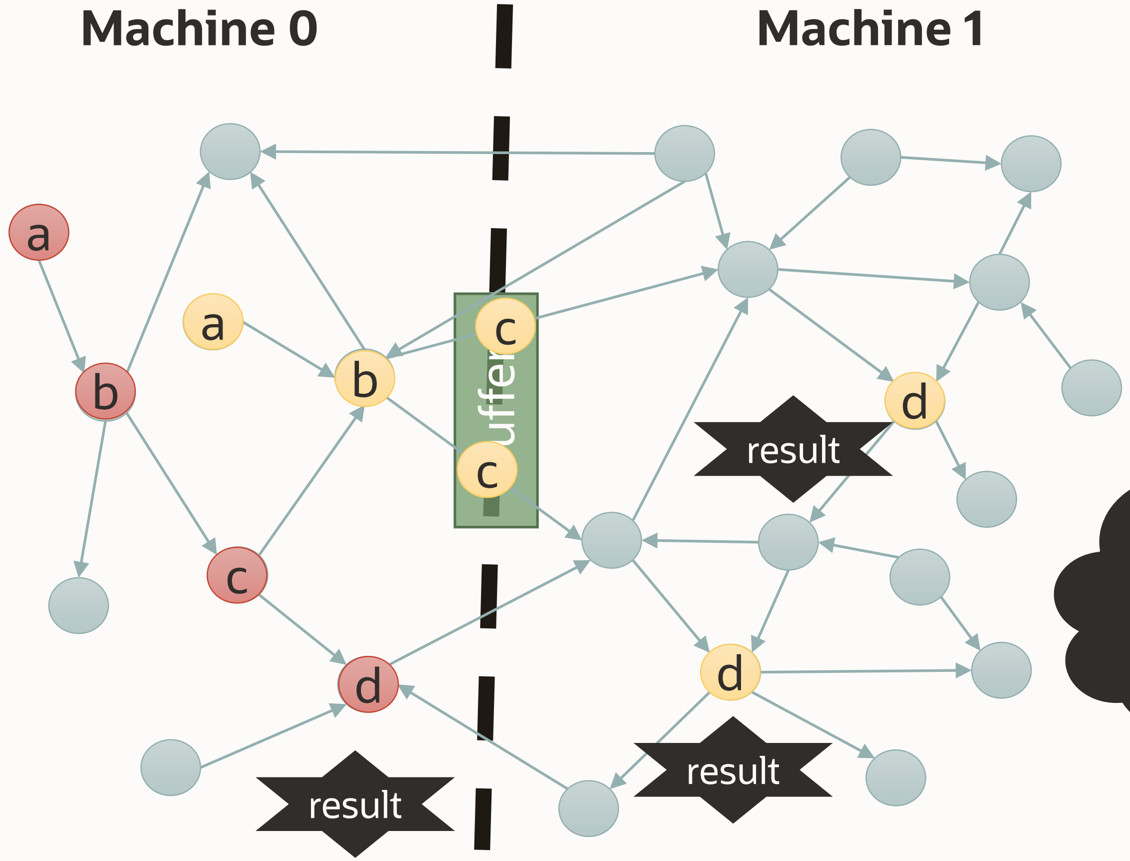
A list of stages that “know” how to

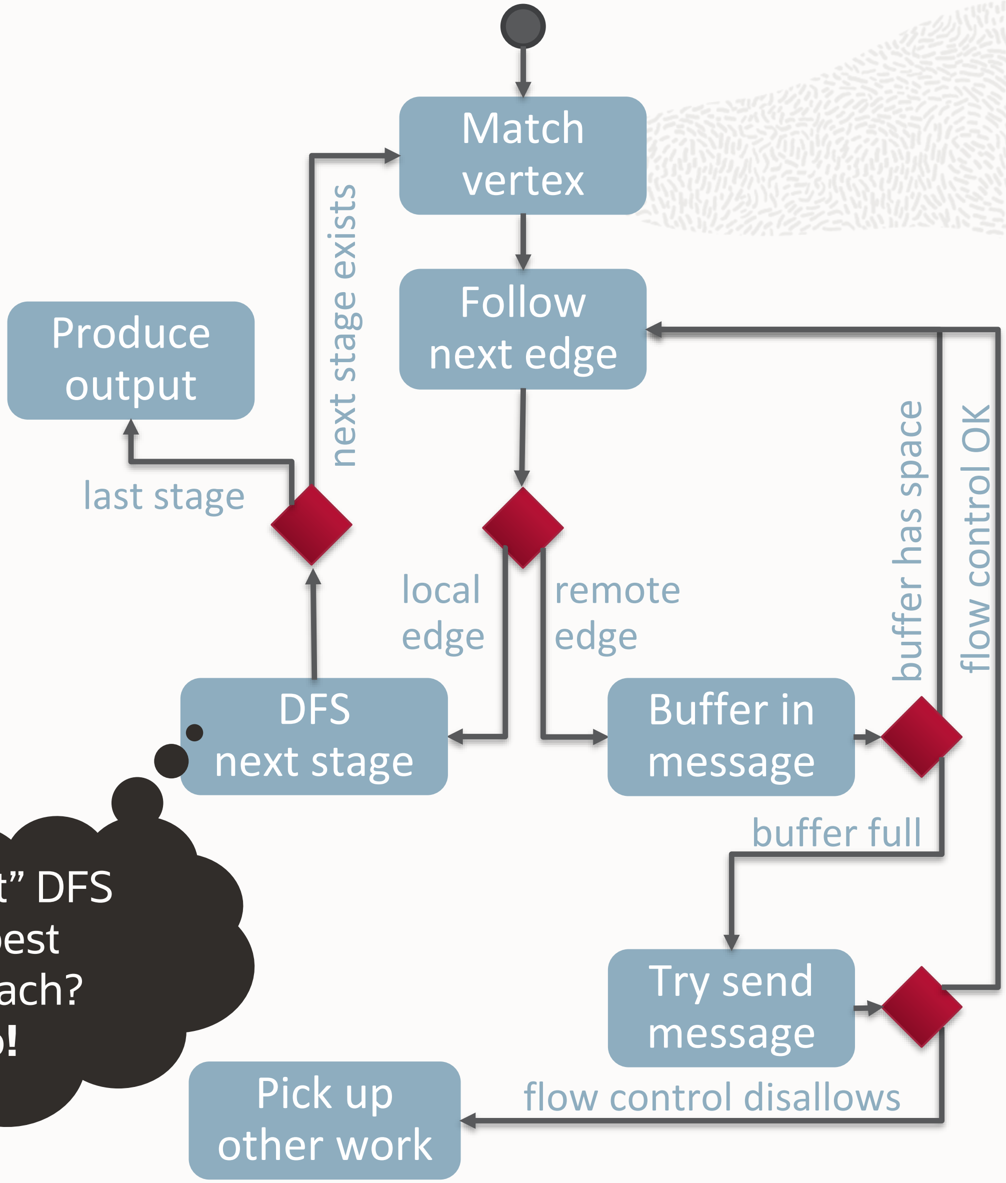
1. match a vertex
2. move to next stage

Asynchronous DFS/BFS Traversals

stage 0 stage 1 stage 2 stage 3

MATCH (a) → (b) → (c) → (d)





Match vertex

Follow next edge

Produce output

last stage

next stage exists

local edge

remote edge

DFS next stage

Buffer in message

buffer has space

flow control OK

buffer full

Try send message

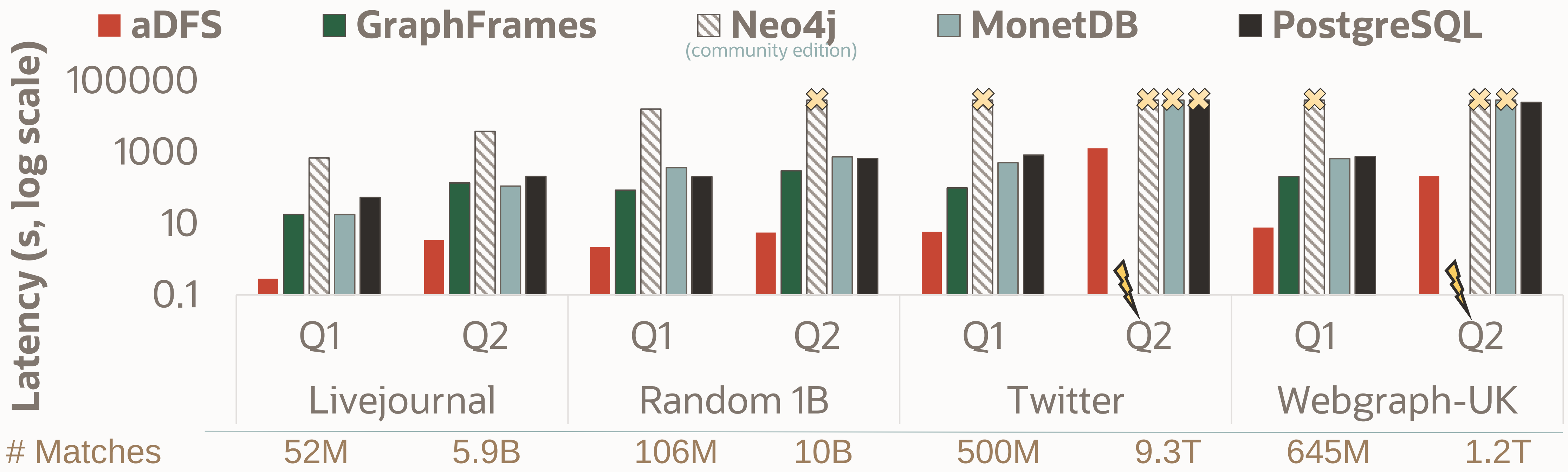
flow control disallows

Pick up other work

Is “strict” DFS the best approach? **No!**

Experimental Evaluation

Schemaless Graphs and Queries



- **Q1: cycle (a)->(b)->(a)** **Q2: 2-hops (a)->(b)->(c)**
- aDFS and GraphFrames with 8 machines / others single machine
- aDFS configured with 1GB memory per machine / others have access to whole machine memory (768 GB)

✖ Did not complete in 8 hours ⚡ Hang due to out of memory

Only aDFS can handle the scale

Conclusions

- **aDFS is a fast and scalable distributed graph querying engine**
  - Provides flexible PGQL querying
  - Combines BFS / DFS
  - Limits max memory usage

- **Also in the paper:** Experiments with the LDBC graph and queries

- **Since the experiments for ATC paper, PGX.D**
  - supports graphs with schema → lower memory and better performance
  - has significantly faster PGQL query execution
  - supports bigger subset of PGQL

Thank you!