Exploiting ancestors for deadlock free transactions in Graph Databases

Ayush Pandey
Sorbonne Université/LIP6
Paris, France
ayush.pandey@lip6.fr

Swan Dubois
Sorbonne Université/LIP6
Paris, France
swan.dubois@lip6.fr

Marc Shapiro
Sorbonne Université/LIP6
Paris, France
marc.shapiro@acm.org

Julien Sopena
Sorbonne Université/LIP6
Paris, France
julien.sopena@lip6.fr

ABSTRACT

TODO

PVLDB Reference Format:
Ayush Pandey, Swan Dubois, Marc Shapiro, and Julien Sopena. Exploiting ancestors for deadlock free transactions in Graph Databases. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at URL_TO_YOUR_ARTIFACTS.

1 INTRODUCTION

Graph databases are becoming ubiquitous for a diverse set of applications. A lot of attention is due to the fact that complex, fine relationships can be represented with ease in graph databases. The ability to represent such relationships opens up the scope for powerful data analysis techniques. A lot of work has been done to make graph databases efficient to query and use for analytics. Languages like Cypher [1] provide extensive SQL like semantics for querying the data. However, similar attention has not been paid to make graph databases efficient for OLTP applications.

2 PROBLEM SCOPE

Define DAGs and the scope of work in terms of the assumptions made for the labelling and how this labelling affects deaclock freedom and tries to minimise the cost.

(1) Graph databases maintain an index.
(2) Node access can be optimised.
(3) Taking a lock in the node can require traversals if intention locks are involved.
(4) Intention locks are prone to deadlocks.
(5) If we dont use intention locks and confine to locking with index, we might still run into deadlocks.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

3 ANCESTORS AND SPECIAL ANCESTOR RELATIONSHIP

In a graph, define what the term ancestor means, how these ancestors are related to a node and the types of ancestors possible. Then proceed to introduce the labelling scheme using ancestors.

(1) Predecessors and ancestors.
(2) Ancestors common on all the paths to a node and the LSA tree for the graph.
(3) Labelling the nodes of a graph based on the nodes encountered on the root to n path in the LSA tree.

3.1 Correctness of the Labelling scheme

With the definitions introduced in the previous section, prove using the paper on new ancestor problems that the strategy works. Use the proof already written

4 IMPLEMENTATION AND TARGET DATABASE USED

Introduce Janusgraph and the database properties along with the language used for iterating and managing transactions in the database. Why janusgraph and how the implementation took place.

(1) Explain how labels are maintained as updates happen.
(2) Locking and transactions in janusgraph
(3) Locking strategy implemented with ancestor calculation
(4) Lock pool
(5) How does this locking compare to standard traversal based locking.

5 EXPERIMENTAL EVALUATION

REFERENCES

[1] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. 2018. Cypher: An Evolving Query Language for Property Graphs. In *Proceedings of the 2018 International Conference on Management of Data* (Houston, TX, USA) *(SIGMOD '18)*. Association for Computing Machinery, New York, NY, USA, 1433–1445. https://doi.org/10.1145/3183713.3190657