TGraph: A Temporal Graph Data Management System

Haixing Huang, Jinghe Song, Xuelian Lin, Shuai Ma, Jinpeng Huai
SKLSDE Lab, Beihang University & Beijing Advanced Innovation Center for BDBC
{huanghx, songjh, linxl}@act.buaa.edu.cn {mashuai, huaijp}@buaa.edu.cn

CIKM 25 ANNIVERSARY 2016
Indianapolis
October 24-28, 2016

Introduction

Temporal graphs are a class of graphs whose nodes and edges, together with the associated properties, continuously change over time. There are some temporal graph databases such as DeltaGraph and G*. But these systems barely support aggregate time range queries. Moreover, these systems cannot guarantee ACID transactions, an important feature for data management systems as long as concurrent processing are involved. To solve these issues, we design and develop **TGraph**, a temporal graph data management system which is designed to manage those temporal graphs whose structures barely change, but the associated properties change frequently, such as road networks (**Fig. 1**).

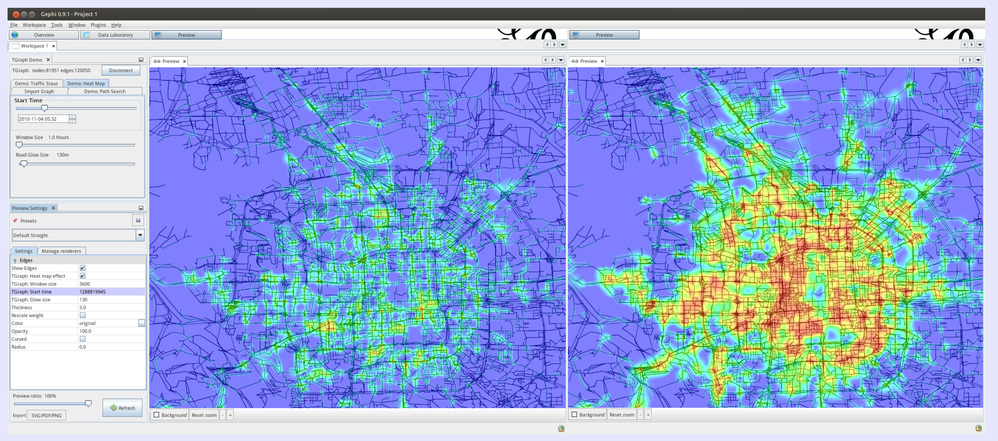

Figure 1. Aggregate traffic situations of Beijing road network (heatmap view). 2010-11-04 05:32~06:32 (left), 07:30~08:30 (right). Support by TGraph's time range query feature.

Architecture

TGraph is an extension of Neo4j [1], a graph database, with two major improvements:

1. **Dynamic Property Storage** that supports fast temporal graph queries.
2. **Enhanced transaction manager** that ensures ACID transaction feature of the system.

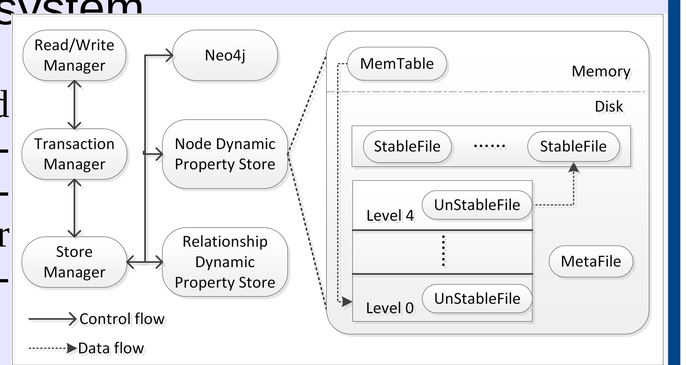

Figure 2. TGraph's Architecture. The Read/Write Manager detects and passes read/write requests to the Transaction Manager with the ACID transaction feature. The Store Manager manages the storage of nodes, relationships, static and dynamic properties.

Dynamic Property Storage (DPS)

In DPS (right side of **Fig. 2**) all data is first written into an in-memory data structure **MemTable**. A MemTable is dumped to the disk when its size reaches a threshold. DPS has three types of local files:

UnStableFiles are used to store the most recent data. When a MemTable is dumped into the local file system a level 0 UnStableFile is created. Then they can be merged into a level 1 UnStableFile, and so on. Level 4 UnStableFiles can be merged into a **StableFile**. Each StableFile/UnStableFile covers a valid time interval. The intervals of StableFiles/ UnStableFiles have no overlaps with each other. Dynamic properties stored in a file have their time fall into the time interval of the file. The **MetaFile** is used to maintain the name and valid time interval of each data file.

Transanction Manager

TGraph extends the resource lock of Neo4j to two types of resource locks. One is for nodes, relationships and static properties, and the other is for dynamic properties. The former is the common read-write lock that uses a node or a relationship as the unit of a lock. The latter uses a dynamic property as the unit of a lock. And unlike the common read-write lock, not all read operations will be blocked by the write lock on the same resource. For example, if a transaction updates a dynamic property $\boldsymbol{p}$ at time $\boldsymbol{t}$ and holds a write lock of $\boldsymbol{p}$, and another transaction needs to read the value of $\boldsymbol{p}$ at the time $\boldsymbol{t'}$, then the read transaction is not blocked if $\boldsymbol{t'} < \boldsymbol{t}$.

Demonstration

We demonstrate TGraph by:

1. Performance evaluation against Neo4j.
2. Use case: A traffic data management system.
3. Use case: API usage (see our demo paper for details).

Dataset : Beijing road network (contains about 110,000 roads) and corresponding traffic data from 2010-05-01 to 2010-11-30.
Environment : CPU (2x Intel Xeon E5-2630 2.4GHz), 64GB memory, 64 bit Window 7 Ultimate system.

Performance Evaluation

(1) We compare the performance of common temporal operations (single thread) between TGraph and Neo4j to show the advantages of TGraph's DPS.

Operation	TGraph	Neo4j
write	52,551 operations/second (**20x faster**)	2,621 operations/second
time point read	12,500 operations/second (**28.5x faster**)	439 operations/second
time range read	10,858 operations/second (**27x faster**)	403 operations/second

(2) The result of concurrent capacities of TGraph vs. Neo4j shows the concurrent capacity of TGraph's transaction manager is 1.13 times better than the one of Neo4j on average (**Fig. 3**).

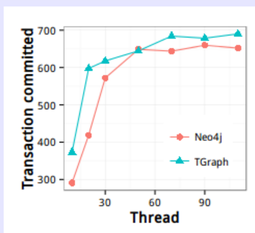

Figure 3. Concurrency

We compare against Neo4j because its popularity and support of ACID transactions. In Neo4j, we encode a dynamic property into an integer array, and store it as a static property of a relationship.

Use Case: Traffic data management system

We build a traffic data management system on top of Gephi [2], a user can:

- specify a time range in history and view the aggregate traffic status of it (**Fig. 1**).
- specify start node, target node, departure time and query for an earliest arrival path [3] (**Fig. 4**).
- query the traffic condition at a history time point.

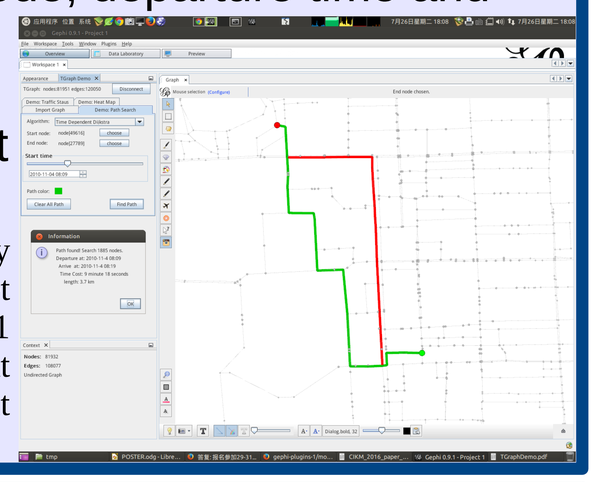

Figure 4. Earliest arrival paths found by the system from start node (red) to target node (green), departed at 2010-11-04 7:01 (red path) and 8:09 (green path) a.m. At 8:09 the system returns a different path that avoids jammed roads.

References

[1]. Neo4j. http://neo4j.com/ .
[2]. Gephi. http://gephi.org/ .
[3]. S. E. Dreyfus. An appraisal of some shortest-path algorithms. Operations Research, 17(3):395–412, 1969

Haixing Huang, Jinghe Song, Xuelian Lin, Shuai Ma*, Jinpeng Huai: TGraph: A Temporal Graph Data Management System, CIKM 2016 (demo paper)
This work is supported in part by 973 program (2014CB340300), NSFC (61322207&61421003), Special Funds of Beijing Municipal Science & Technology Commission, and MSRA Collaborative Research Program.