



Adding Updates to the LDBC SNB Workload

Mahashweta Das, Alkis Simitzis, Kevin Wilkinson

Topics

– Motivation: graph database engine for The Machine

- challenge: graph engine for mixed workloads of navigational and analytic requests and updates

– LDBC SNB workload at a glance

- almost exactly what we need

– Adding updates to SNB

- how we perverted the data and workload generator

A Graph Database Engine for The Machine

- The Machine: Hewlett Packard Labs project to build next-generation computer
 - 100s of cores, 100s TB of non-volatile main-memory (NVM), photonic interconnect
 - need applications to showcase benefits of TM

- Hypothesis: graphs applications are an excellent match for TM
 - in-memory, scale-up graph engines have best performance ... if the graph fits
 - NVM promises near-DRAM speed and near-disk capacity and cost/bit
 - graph access patterns are random, unpredictable – good for large shared memory

But, which type of graph application? Many are already well served by existing engines.

A Broad Characterization of Graph Engines

- Navigational request: access few vertices or edges, requires limited computational resources
- Analytic request: access large fraction of graph, requires most/all computational resources
- Update request
 - insert-only: add new vertex, edge, property value
 - modify/delete: change to an existing vertex, edge or property value

Assertions without proof

1. graph engines are optimized for either navigation or analytics, not both
2. analytic engines have poor support for modify/delete

Goal: High Performance for Mixed Workloads with Updates

Challenge: build a graph engine to

1. provide high performance *concurrently* for navigation and analytic requests ...
2. and where *updates do not degrade* performance

Solution: MAGS ... stayed tuned for Mahashweta's talk tomorrow at GRADES

Who cares? Are there real-world applications with such requirements

Driving Application: Operational Analytics

Defn: capture, analyze and react to events in real-time to improve business operations

– Example: IT security analytics

- capture DNS, proxy, netflow, syslog events to looking for attacks, intrusions, unusual behavior
- IT assets (PCs, servers, printers, routers) come and go or are modified
- security threat patterns come and go and black/white lists are modified

– Example: oil-gas production (and related IoT scenarios)

- capture temperature, pressure, flow at drills to anticipate and avoid slowdowns or failures
- drilling equipment status constantly changes, equipment added, moved or retired

– Example: national security tracking suspected terrorists

- analytics run over snapshot of graph data rather than real-time graph



Problem: Need an Operational Analytics Workload

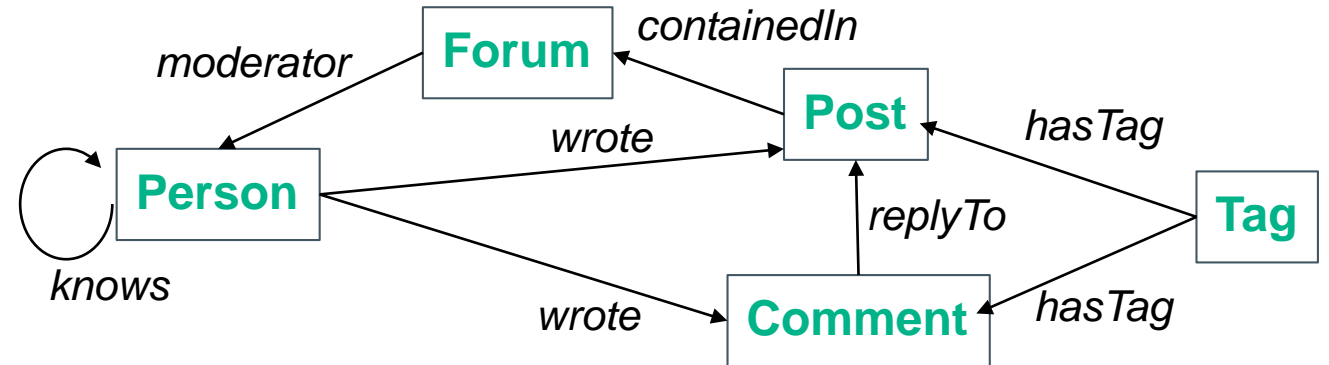
... or an approximation

Solution: LDBC SNB meets most of our requirements

- it models an enterprise
- with realistic, scalable datasets
- it includes navigation and analytic requests
- But, it *does not have modify/delete* requests

DBC SNB Workload at a Glance

- models a *Facebook*-like application
- simplified schema shown
- objects/relations have many attributes



- data generator produces realistic datasets at various scale factors (e.g., non-uniform distributions)
- workload comprises
 - simple reads: e.g., all direct (1-hop) friends of a person, all comments to a post
 - complex reads: e.g., recent posts by all friends of a person, shortest path between two persons
 - updates: e.g., add new person, new post, new friend of a person

Adapting LDBC SNB for Mixed Workloads plus Updates

- Navigational requests: all eight simple reads
- Analytic requests: seven complex reads plus PageRank (from Graphalytics)
- Updates
 - SNB updates not used
 - defined two new update requests, **OldPost**, **NewPost**
 - posts are a large fraction of the data set
- LDBC SNB workload generator not used
 - has complicated relationship among different actions (different frequencies and dependencies)
 - for rapid prototyping, easier/faster to do our own

OldPost and NewPost Transactions

- OldPost deletes a post plus all its likes and tags
- NewPost adds a post plus a random number of likes and tags
- equal number of NewPost, OldPost requests
(database size remains constant)

```
Procedure OldPost ( delld integer ) {  
  
    SqlStmt DelLikes = "delete from PostLikes  
                        where postld = ?delld;"  
  
    SqlStmt DelTags = "delete from PostTags  
                      where postld = ?delld;"  
  
    SqlStmt DelPost = "delete from Post  
                      where postld = ?delld;"  
  
    prepare (DelLikes,DelTags,DelPost);  
  
    execute (DelLikes,DelTags,DelPost);    }  
}
```

```
Procedure NewPost ( insld integer ) {  
  
    SqlStmt InsPost = "insert into Post  
                      (postld, forum, creatorld, ...) values ( ...);"  
  
    SqlStmt InsLikes = "insert into PostLikes  
                       (personld, postld, crdate ) values ( ...);"  
  
    SqlStmt InsTags = "insert into PostTags  
                      (postld, tagld ) values ( ...);"  
  
    prepare (DelLikes,DelTags,DelPost);  
  
    numLikes = urandom (0,20);  
    numTags = urandom (1,6);  
  
    execute (InsPost);  
    for (i=0 .. numLikes) execute(InsLikes);  
    for(i=0 .. numTags) execute(InsTags);    }  
}
```

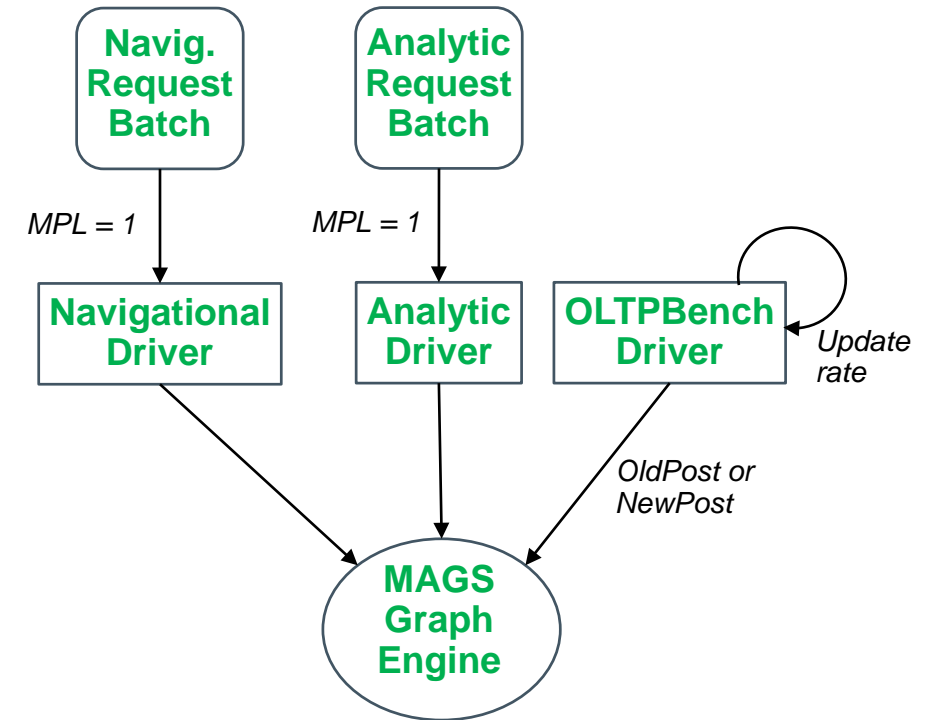
Our Mixed Workload for MAGS

– Input: three concurrent request streams

- navigational: batch of 1021 queries run serially
- analytic: batch 23 queries run serially
- update: OldPost or NewPost request
 - each equally likely
 - update rate fixed per run (range from 0 to 500 rqsts/sec)
- total run time about 5 minutes
- simple drivers for navigational, analytic batches
- OLTPBenchmark used for update driver

– Measures

- total completion time and average latency of each request type
- compare baseline (no updates) vs. increasing update rate
- freshness: delay between commit of an update and visibility to queries



Summary

- data change and delete are important for graph analytics
 - because real world applications have updates
 - updates can greatly impact performance
- LDBC SNB does not address this need
 - it includes insert but no updates
- we demonstrate a true update stream
 - limited but served our needs for a PoC
 - more work needed for a general update stream (update Person, Comment, etc.)
- we hope to generate interest in this topic

