

Efficiently Storing and Retrieving Evolving Knowledge

Ruben Taelman - @rubensworks
imec - Ghent University

Miel Vander Sande, Ruben Verborgh, Erik Mannens

Querying evolving data is important

What was the temperature of sensor X on November 10th at 11:20?

How did the iron-level of patient X evolve between this and last year?

At what times was the rainfall in Belgium above level X?

Goal: Support versioning in Triple Pattern Fragments (TPF) framework

Trade-off between server and client effort for publishing LD

Limit **server** interfaces to **triple pattern queries**

Clients evaluate SPARQL **queries locally**

(Verborgh 2016)

TPF servers usually store data using HDT

Highly compressed triple store

Efficient triple pattern querying with offsets

Immutable snapshots

(Fernández 2010)

Storage requirements for **versioning** in TPF

Highly compressed triple store

Efficient triple pattern querying with offsets **against versions**

~~Immutable snapshots~~ → **Store multiple dataset versions**

New changeset-based triplestore

Exploit efficiency of snapshot storage

Create consecutive **patches** for each version

Compress similarities between versions

Allow efficient querying by triple pattern and version



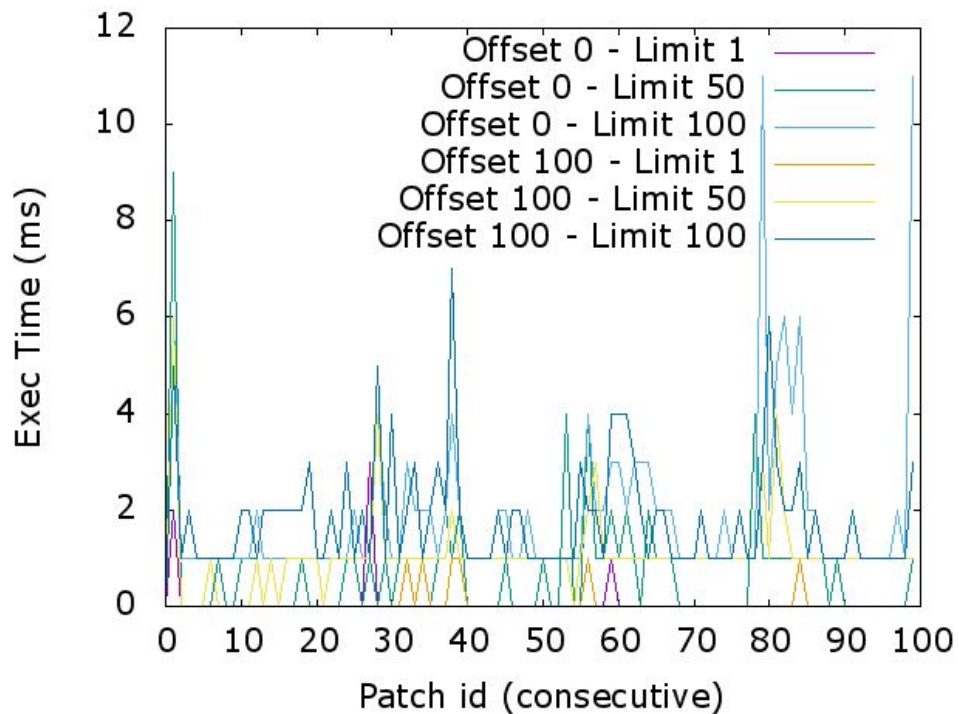
Intermediary results are promising

Medium sized synthetic datasets

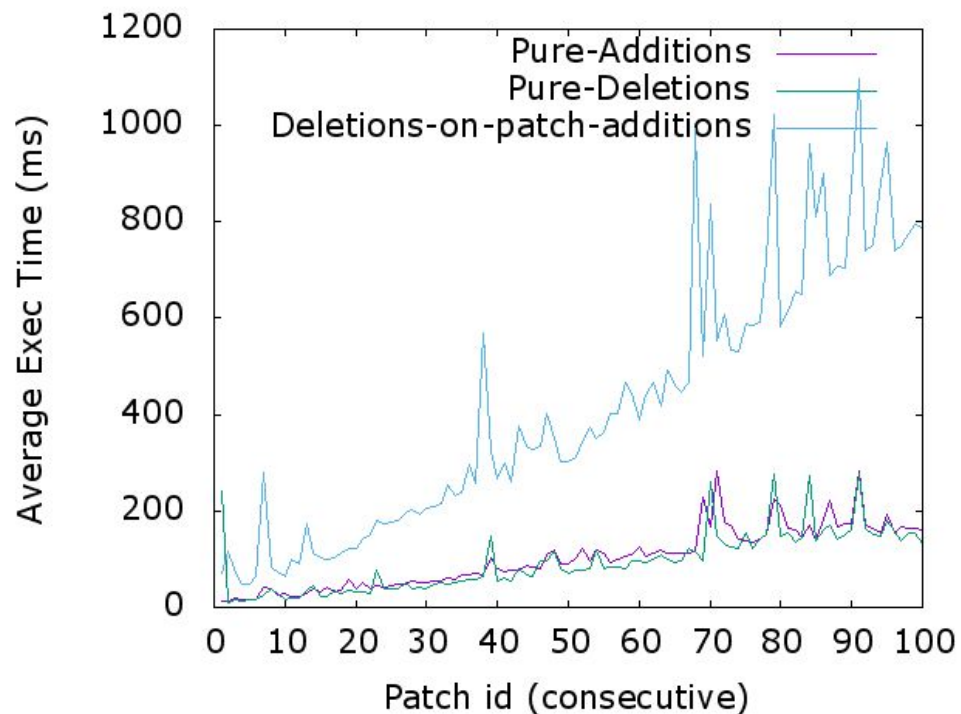
Initial snapshot of 10 000 triples

100 versions each time changing ~100 triples

Querying by triple pattern against any version



Insertion algorithm needs more work



Storage requirements

