トップエスイー修了制作

SPARQLエンドポイントの効率的なキャッシュ手法の提案

株式会社レベルファイブ　佐藤大輔　daisuke.satoh@level-five.jp

開発における問題点

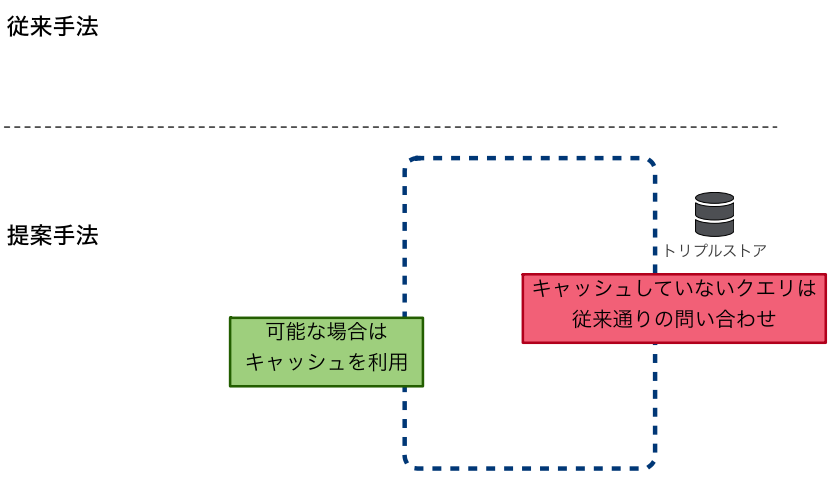
SPARQLクエリを用いてRDFで記述されたデータを検索するには対象のRDFスキーマの理解が必要だが、初学者にとっては大きなコストとなる。SPARQL Builder(SB)は、RDFスキーマの知識がないユーザのクエリ作成を支援するツールであるが、支援に必要な情報を収集するためのクエリが高負荷であるという問題がある。

手法・ツールの適用による解決

SBが発行するクエリに対するキャッシュ手法を提案する。データ投入時に投入されるトリプルがクエリのWHERE句条件を満たすかをチェックし、クエリと等価な回答を返すためのキャッシュを作成しておくことで、SBの問い合わせに素早く応答することができ、データ更新時にも少ないコストでキャッシュを更新できる。

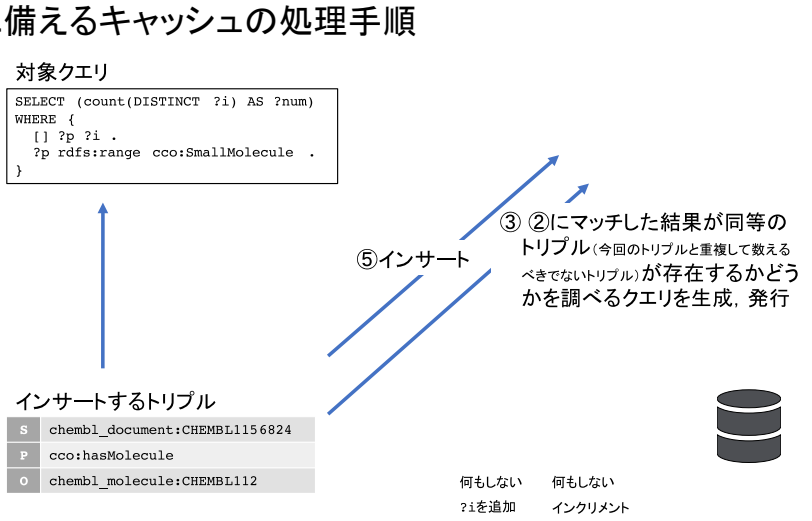
投入されるトリプルに着目したキャッシュ手法の提案

■データ投入とSBのクエリを代理するシステム



SELECT句・WHERE句の内容から7パターンに分類
⇒クエリと等価な回答を返すための情報をデータ投入時にあらかじめ計算しキャッシュする

■キャッシュ管理処理の一例
ある条件にマッチするユニークな要素の数え上げを行うクエリに備えるキャッシュの処理手順



ユニークな要素を数え上げるため、既に数え上げた値はキャッシュしておき、未知の値の時のみカウンタをインクリメントする

結果

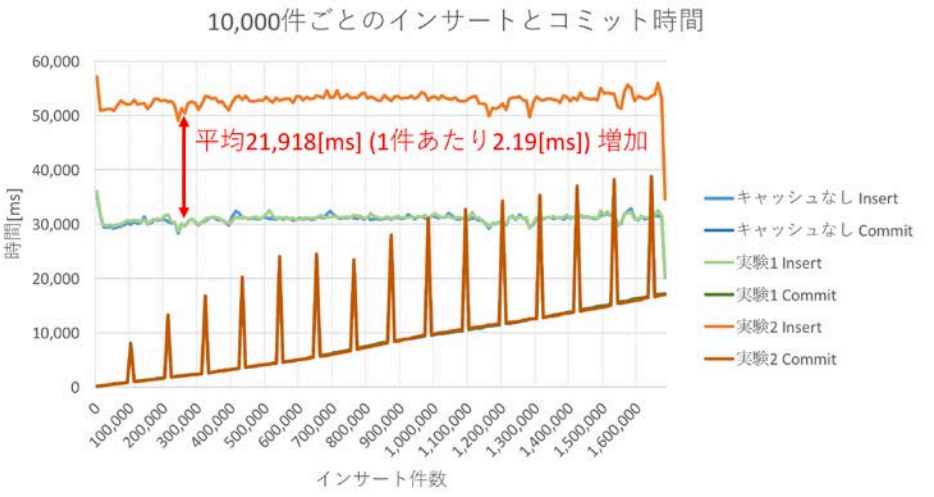
実験1：WHERE句に単一のトリプルが指定されているクエリ
実験2：WHERE句に複数のトリプルが指定されているクエリ

データ投入時間(ミリ秒)

	キャッシュなし(A)	キャッシュ作成(B)	B / A
実験1	6,850,020 (114分10秒)	6,876,584 (114分36秒)	1.004
実験2	6,850,020 (114分10秒)	10,551,602 (175分52秒)	1.540

クエリ実行時間(ミリ秒)

	キャッシュなし(A)	キャッシュあり(B)	B / A
実験1	266,371 (4分26秒)	24	0.000090
実験2	1,016,656 (16分56秒)	25	0.000024



⇒実験1ではキャッシュ作成による著しい処理時間の増加は見られなかった
⇒実験2ではデータの件数が増加しても投入時間が安定しており、高速化による効果が得られていることが分かる