Noria

Noria: dynamic, partially-stateful data-flow for high-performance web applications; 2018
Jon Gjengset;
MIT CASIL

TDT02
Jakob Simonsen
Hva er Noria?

- Partial State
- In Dataflow-Based
- Materialized views

- Et dataflow system som bruker materialized views som cache ved å tillate ikke tilgjengelig states
Select Query
Insert
Web application
Cache

1. Insert vote
2. Invalidate cache
3. Query on cache miss
4. Fill in the cache..?
5. Evict from the cache..?
Dataflow

- Dataflow definerer vi som et push basert data system til compute
- Dataflow er en konstruert graf av operatorer som kan konstrueres fra SQL
- Hver edge i grafen er en data dependency
CREATE MATERIALIZED VIEW
  StoryWithVC
AS SELECT
  stories.*,
  COUNT(votes.user) AS votes
FROM stories
JOIN votes
  ON (votes.story_id = stories.id)
GROUP BY stories.id;
Materialized views

CREATE MATERIALIZED VIEW StoryWithVC
AS SELECT
    stories.*,
    COUNT(votes.user) AS votes
FROM stories
JOIN votes
    ON (votes.story_id = stories.id)
GROUP BY stories.id;

SELECT * FROM StoryWithVC WHERE id = ?
CREATE MATERIALIZED VIEW
   StoryWithVC
AS SELECT
   stories.*,
   COUNT(votes.user) AS votes
FROM stories
JOIN votes
   ON (votes.story_id = stories.id)
GROUP BY stories.id;
Partial state

- Det meste av data I viewet blir ikke brukt
- Slett ubrukte enteries og bare legge til nye ved forespørrsel

- Ved bruk av paramtre kan "partial sate view" vite hvilken parametre applikasjonen spør om og derfor hente infomasjon til viewet.
CREATE MATERIALIZED VIEW
   StoryWithVC
AS SELECT
   stories.*,
   COUNT(votes.user) AS votes
FROM stories
JOIN votes
   ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = ?;
CREATE MATERIALIZED VIEW StoryWithVC
AS SELECT stories.*, COUNT(votes.user) AS votes
FROM stories
JOIN votes
ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = 7;
CREATE MATERIALIZED VIEW
  StoryWithVC
AS SELECT
  stories.*,
  COUNT(votes.user) AS votes
FROM stories
JOIN votes
  ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = 7;
Eviction policy

CREATE MATERIALIZED VIEW
  StoryWithVC
AS SELECT
  stories.*,
  COUNT(votes.user) AS votes
FROM stories
JOIN votes
  ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = ?;
Oppdatering av artikkel med missing state

CREATE MATERIALIZED VIEW StoryWithVC
AS SELECT stories.*, COUNT(votes.user) AS votes
FROM stories
JOIN votes
ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = ?;
Oppdatering av artikkel med missing state

CREATE MATERIALIZED VIEW
  StoryWithVC
AS SELECT
  stories.*,
  COUNT(votes.user) AS votes
FROM stories
JOIN votes
  ON (votes.story_id = stories.id)
GROUP BY stories.id
WHERE stories.id = ?;
Testing på en Amazon EC2 instans med 16 cores
Testing på en Amazon EC2 instans med 16 cores

![Bar diagram comparing performance of MySQL, Noria without partial, and Noria]
Resultater

- Hvorfor skal man bruke Noria?
- Kan man bruke Noria?
- Hva er status I dag?
- Hvordan blir sammenligningen med Redis som cache?
Kilder

- https://www.youtube.com/watch?v=GctxvSPIfr8&t=2079s&ab_channel=JonGjengset
- https://github.com/mit-pdos/noria