Top-k/w publish/subscribe:
A publish/subscribe model for continuous top-k processing over distributed data streams

Ivana Podnar Žarko
University of Zagreb, FER
Contact: ivana.podnar_zarko@fer.hr

Joint work with:
Krešimir Pripužić (Uni Zg), Karl Aberer (EPFL)
Outline

- Motivation
- Data Stream Processing vs. Publish/Subscribe
- Top-k/w Publish/Subscribe Model
- Centralized Top-k/w Processing
- Distributed Top-k/w Processing
- Conclusions and Future Work
Information Explosion

- What is the rate of new information growth each year?
- How much data do we as individuals consume each day?

- “We create as much information in two days now as we did from the dawn of civilization up until 2003.”
  Eric Schmidt, Google CEO, Aug. 2010

- “In 2008, Americans consumed information for about 1.3 trillion hours, an average of almost 12 hours per day. Consumption totaled 3.6 zettabytes (1ZB = 10^{21}) and 10,845 trillion words, corresponding to 100,500 words and 34 gigabytes for an average person on an average day.”
  How Much Information? (HMI) research program, University of California at San Diego, Dec. 2009
“Personal Information Filtering Engine”

- RSS feeds, blogs, tweets, etc.
  - Simply too many information sources to follow!
- But, if I would receive only top-20 notifications w.r.t my profile and personal interest during a day, and also at the time when they are published…
“Auction Site Super-Network”

- Spans over many online auction sites
- A user can define his/her ideal product of interest and receive, e.g., top-10 offers within the course of a day that are most similar to his/her ideal product
Examples of Sensor Web Applications

- **Real-time environmental monitoring**
  - Environmental scientists would like to identify and monitor up to 10 sites with the largest pollution readings over the course of a single day.
  - Identify 10 sensors closest to a particular location measuring the largest pollution levels over time (e.g. top-10 readings are provided on hourly basis).

- **Power grid monitoring**
  - Operators would like to monitor over time 100 sites with the largest or the lowest power production using solar panel current and voltage readings so that they can identify power grid hot-spots.
Outline

- Motivation
- Data Stream Processing vs. Publish/Subscribe
  - Top-k/w Publish/Subscribe Model
  - Centralized Top-k/w Processing
  - Distributed Top-k/w Processing
- Conclusions and Future Work
Paradigm Shift in Data Processing

- **Traditional store-then-query data processing model**
  - One-time queries
  - High querying frequency compared to the frequency of data updates
  - Drawbacks
    - In many scenarios it is impossible to store all produced data
    - Most of the stored data items will never be accessed again
    - Slow for real-time processing

- **Novel query-then-store data processing model**
  - Continuous queries
  - Frequency of query updates is low compared to the frequency of data publications
  - Real-time query processing: data is matched against the queries, some data objects are stored in memory
Data Stream Processing vs. Publish/Subscribe

Data Stream Processing System (DSPS)

Publish/Subscribe System (P/SS)
**DSPS & P/SS: A comparison**

- **Differences in terminology**

<table>
<thead>
<tr>
<th>Data stream processing systems</th>
<th>Publish/subscribe systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous query</td>
<td>Subscription</td>
</tr>
<tr>
<td>Data object</td>
<td>Publication</td>
</tr>
<tr>
<td>Data stream (objects)</td>
<td>Incoming Publications</td>
</tr>
<tr>
<td>Query Result Stream</td>
<td>Delivered Publications</td>
</tr>
</tbody>
</table>

- **Conceptual differences**

<table>
<thead>
<tr>
<th>Data stream processing systems</th>
<th>Publish/subscribe systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tightly coupled components</td>
<td>Loosely coupled components</td>
</tr>
<tr>
<td>Platform dependent</td>
<td>Platform independent</td>
</tr>
<tr>
<td>Less scalable to the number of users</td>
<td>Scalable (distributed architecture)</td>
</tr>
<tr>
<td>Complex (stateful) queries</td>
<td>Simple (stateless) queries</td>
</tr>
</tbody>
</table>
**Boolean Matching in P/SS**

- **Publication** is a point in an attribute space
- **Subscription** is a subspace from the same attribute space

---

**Static stateless Boolean function**

- publications
- subscriptions

---

Universitiy of Zagreb
Drawbacks of the Boolean Matching Model

- How many matching publications will be delivered to a subscriber during a period of time?

General subscription

Over-specified subscription

Top-k/w publish/subscribe
Top-k/w Matching

- publication
- subscription

Score

- yes [0,1]
- no

Top-k/w Matching

Scores of other recent publications

Static threshold?
- Too abstract to define
- The same drawbacks as the Boolean Model

Top-k/w matching
- Assumption: Recent publications are more important than old ones
- Limit the number of matching and delivered publications to k best within a sliding window of size w
- Compare publication score to scores of other recent publications
State-of-the-art publish/subscribe systems do not support publication ranking.

The idea to rank publications in P/SS has been developed in parallel.

Related work (2)

- Pub/sub for information filtering (static threshold)
  - Christos Tryfonopoulos, Manolis Koubarakis, Yannis Drougas: **Information filtering and query indexing for an information retrieval model.** ACM Trans. Inf. Syst. (TOIS) 27(2) (2009)

- Continuous top-k query processing
  - Kyriakos Mouratidis, Spiridon Bakiras and Dimitris Papadias: “**Continuous monitoring of top-k queries over sliding windows**”, SIGMOD 2006
  - Christian Böhm, Beng Chin Ooi, Claudia Plant and Ying Yan: “**Efficiently Processing Continuous k-NN Queries on Data Streams**”, ICDE 2007

- Top-k query processing for vector spaces
  - Parisa Haghani, Sebastian Michel, Karl Aberer: **The gist of everything new: personalized top-k processing over web 2.0 streams.** CIKM 2010: 489-498
Outline

- Motivation
- Data Stream Processing vs. Publish/Subscribe
- Top-k/w Publish/Subscribe Model
  - Centralized Top-k/w Processing
  - Distributed Top-k/w Processing
- Conclusions and Future Work
Top-k/w Publish/Subscribe Model in Brief

- Subscriber controls the number of publications it receives per subscription (top-k) within a sliding window.
- Ranks publications according to the degree of relevance (score) to a subscription.

Subscription is defined by:
1) totally-ordered and time-independent scoring function
2) parameter $k \in \mathbb{N}$
3) parameter $w \in \mathbb{R}^+$ *(time-based)* or $n \in \mathbb{N}$ *(count-based sliding window)*.

Each publication is competing with other publications from the sliding window for a position among top-k publications!
Problem Statement

- When can a publication become a top-k object in the subscription window?
  - Immediately upon publication
  - Later on when it becomes a top-k object in the subscription window

- Maintain a set of candidate (potential top-k) publications in memory!
Candidate Publications

- Which publications can become top-k in the window?
- Non-dominated publications
  - Publications dominated by less than \( k \) other publications within the window
- Dominance property in a two-dimensional score-time space
  - Publication \( a \) is dominated by publication \( b \) iff
    - \( b \) is younger than \( a \)
    - \( b \) has a better rank than \( a \)

Candidate publications are maintained in a \textbf{k-skyband} (Strict Candidate Pruning Algorithm, SA) [SIGMOD 2006, ICDE 2007]

Our contributions:
- relaxed k-skyband
- probabilistic k-skyband
Relaxed Candidate Pruning Algorithm (RA)

Relaxed $k$-skyband may contain dominated publications (periodical pruning)

Advantage

Improved average time complexity compared to SA

$$O\left(\ln\left(k \cdot \ln\left(\frac{n}{k}\right)\right)\right) \text{ vs. } O\left(k \cdot \ln\left(\frac{n}{k}\right)\right)$$

Drawback

Slightly increased (controllable) memory consumption
Observation: a lot of publications from a k-skyband will never become top-k publications
- New ones with low ranks – they are non-dominated when entering the system as they are the youngest!

Probabilistic k-skyband
- Calculate the probability that a publication becomes top-k in future (of course, when it enters the system)
- Probabilistic criterion

\[
p_b(l, n, k) = \frac{n^2}{4n-2} \cdot \sum_{l'=1}^{k} \frac{(n-1)(n-1)}{(2n-2)(2l+l')}. \quad l \text{ – initial rank when publication enters the system}
\]

Valid for random-order data streams
- Any permutation of published publications is equally likely to appear in a stream
Advantages
- Average time complexity $\approx O(1)$
- Controllable and low probability of error

Drawbacks
- Probabilistic algorithm, generates both false positives and false negatives
- Applicable for processing random-order data streams
Recent Object Buffering

- Improves the performance of deterministic algorithms (SA and RA)
- Algorithm outline, originally proposed in [ICDE 2007]
  - Keep a buffer of most recent publications in memory (size \( b << n \)), one for all subscriptions
  - Delay insertion of publications from the buffer into a strict/relaxed k-skyband (of course, those that are not top-k when entering the system)
  - Maintain a **subscription filter** (k-skyband built of non-dominated publications from the buffer)
  - Try inserting an object from the buffer into the strict/relaxed k-skyband twice (1\(^{st}\) attempt when entering, 2\(^{nd}\) attempt when exiting the buffer)
- Our contribution: probabilistic filter (PF) vs. strict filter (SF)
- Algorithms: **SASF**, SAPF, RASF, RAPF
Further Improvement: Subscription Indexing

- Regular grid
- It is possible to identify a subspace of interest per each subscription which contains publications of interest
- The threshold is varying in time!
  - For SASF: threshold = score of the k-th object in a subscription filter
  - For PA: threshold = score of the publication with the worst rank in the set of publications for which the probabilistic criterion holds

An example: k-NN subscriptions
Experimental Evaluation

- Publications and subscriptions are points in a multidimensional attribute space
- Euclidean distance as a scoring function (k-NN subscriptions)
- Datasets
  - one real dataset: LUCE deployment data (environmental data collected from a large-scale wireless sensor networks within the project SensorScope http://sensorscope.epfl.ch/)
  - two synthetic datasets
    - uniform and clustered Gaussian data

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publications</td>
<td>1.000.000</td>
</tr>
<tr>
<td>Subscriptions</td>
<td>400</td>
</tr>
<tr>
<td>Count-based sliding window</td>
<td>40.000</td>
</tr>
<tr>
<td>Recent buffer</td>
<td>2.000</td>
</tr>
<tr>
<td>Data dimensionality</td>
<td>4</td>
</tr>
<tr>
<td>Grid resolution</td>
<td>10</td>
</tr>
<tr>
<td>RA: pruning coefficient</td>
<td>0,2</td>
</tr>
<tr>
<td>PA: probability of error</td>
<td>$10^{-3}$</td>
</tr>
</tbody>
</table>
Processing cost for the uniform dataset without query indexing
Simulation Runtime (2)

Processing cost for the uniform dataset without query indexing

Processing cost for different datasets with query indexing
Number of delivered publications

- Top-k/w matching model adapts well to publication distribution
- The number of matching publications in the Boolean matching model depends heavily on publication distribution

(a) Top-k/w subscription: uniform dataset
(b) Top-k/w subscription: clustered dataset
(c) Top-k/w subscription: real dataset
(d) Boolean subscription: all datasets
To conclude the model evaluation…

- Top-k/w matching model adapts well to various data publication rates and special characteristics of the data set (in terms of the number of delivered publications)
- PA gives the best performance in terms of efficiency and memory consumption
- RAPF is the best performing deterministic algorithm
Outline

- Motivation
- Data Stream Processing vs. Publish/Subscribe
- Top-k/w Publish/Subscribe Model
- Centralized Top-k/w Processing
- Distributed Top-k/w Processing
- Conclusions and Future Work
Motivation

- Why do we need a distributed solution?
- Three problems arise in an implementation of top-k/w model
  - Limited memory
  - Limited processing power
  - Limited bandwidth
- When publications are produced in a distributed environment, we cannot transport them to a centralized processor (especially if they are streaming in at high rates!)
Distributed Top-k/w in Brief

- Network of processing nodes, each node implements one of the previous algorithms (PA, SASF, RAPF)
- Use the threshold for indexing!

An example: rendezvous routing
Case study: Distributed k-NN processing

- Distance scoring functions in a multidimensional attribute space $\mathbb{R}^d$
- Subscription: point in $\mathbb{R}^d$, $k$, $n$, subscribing node
- Built on top of the CAN overlay network (rendezvous-based routing)
- Partitions the attribute space to cells of equal size, while peers are responsible for zones (one or more neighboring cells)
Subscription Activation

Activation message propagation

Merger updating
(merger covers subspaces of interest of the merged queries)
Publication Process

Propagation of messages: notify rendezvous peer and all interested peers

Forwarding a publication to the owner peer
Publications and subscriptions are points in a 4d space

- Euclidean distance as a scoring function (k-NN subscriptions)
- Data sets: uniform and clustered Gaussian data and one real data set
- We simulated PA and RAPF-based subscriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publications</td>
<td>1,000,000</td>
</tr>
<tr>
<td>Subscriptions</td>
<td>400</td>
</tr>
<tr>
<td>Parameter ( k )</td>
<td>9</td>
</tr>
<tr>
<td>Count-based sliding window</td>
<td>40,000</td>
</tr>
<tr>
<td>Recent buffer</td>
<td>2,000</td>
</tr>
<tr>
<td>Data dimensionality</td>
<td>4</td>
</tr>
<tr>
<td>Grid resolution</td>
<td>12</td>
</tr>
<tr>
<td>RA: pruning coefficient</td>
<td>0.2</td>
</tr>
<tr>
<td>PA: probability of error</td>
<td>1,000</td>
</tr>
<tr>
<td>Number of peers</td>
<td>256</td>
</tr>
</tbody>
</table>
Control Traffic Overhead

Average number of all exchanged messages per publication (and 400 subscriptions)
Total number of exchanged messages for different number of subscriptions
Scalability (2)

Total number of exchanged messages for different number of peers
Outline

- Motivation
- Data Stream Processing vs. Publish/Subscribe
- Top-k/w Publish/Subscribe Model
- Centralized Top-k/w Processing
- Distributed Top-k/w Processing
- Conclusions and Future Work
Conclusions and Future Work

- Top-k/w is a promising technique for efficient filtering of distributed data streams and delivery of publications in real time to distributed subscribers

- Open issues
  - Supporting other scoring functions in a distributed scenario
  - Other approaches to publication ranking, e.g. scoring function is time dependant
  - Indexing of top-k/w subscriptions in vector-space

Krešimir Pripužić, Ivana Podnar Žarko and Karl Aberer: Time-Efficient Sliding Window Top-k Query Processing, under submission

Krešimir Pripužić, Ivana Podnar Žarko and Karl Aberer: Top-k/w publish/subscribe: A publish/subscribe model for continuous top-k processing over data streams, under submission