Minimizing the Subscription Aggregation Cost in the Content-based Pub/Sub System

Ning Wang and Jie Wu
Dept. of Computer and Info. Sciences
Temple University
Road Map

- Introduction
- Subscription aggregation problem
- Proposed subscription aggregation algorithm
- Subscription tree construction
- Experiments
- Conclusion and future work
Introduction

- **Content-based pub/sub system**
  - Messages are only delivered to a subscriber if the attributes or content of those messages match a subscription range.
    - Some subscription examples:
      - stock trade (issue = “IBM” & price < 120 & volume > 1000);
      - car brand (made = “ford” & price > 10,000 & price < 20,000);
      - news delivery (all the sports channels).

- **Real projects:**
  - IBM Gryphon, Microsoft’s OpenPS project, WS-Messenger, SIENA, and Hermes.
Subscription Aggregation Problem

- Subscription tree

- Without subscription aggregation

- Problem:
  - Network congestion
  - High broker load

- Subscription aggregation (Benefit)

  - Aggregate several subscriptions into one subscription

  - Reduce the routing table; reduce the bandwidth consumption; accelerate the routing decision
Subscription Aggregation Problem

In a dynamic environment (e.g., s1 leaves and reports immediately)

- **Without subscription aggregation**
  - Update routing table
  - (immediate)

- **Cost of Subscription aggregation**
  - Re-ask subscription
  - Subscription re-transmission
  - (re-configuration needs time)
Challenges

● There is a trade-off between benefit and cost in the subscription aggregation.

Questions:

○ **Where** to do the subscription aggregation?
  ● Which broker should we try to do subscription aggregation?

○ **How** to do the subscription aggregation.
  ● Which subscriptions should be aggregated?
Subscription Aggregation Problem

Model:

- Network benefit
  - Proportional to the bandwidth saving amount.

- Without subscription aggregation

- Subscription aggregation
  - Saving 2 bandwidth

\[ b_1, s_1, s_2 \]
\[ p_1 \]
\[ b_2 \]

\[ [0,5] \] \[ [4,10] \]
\[ [0,5] \] \[ [4,10] \]

\[ [0,10] \]
\[ [0,10] \]
Subscription Aggregation Problem

- **Model:**
  - Cost (the production of the two following metrics)
    - Unique subscription range (false subscription range after one subscriber leaves)
    - Re-configuration delay
      - the largest hop counts to the aggregation broker.
Subscription Aggregation Problem

Model:
- An illustration of cost ($s_1$ leaves the network)
  - Range $[0, 3]$ (size of 3) becomes false subscription range

Cost: 3

Cost: $3 \times 2$
Related works

- Congestion Avoidance with Incremental Filter Aggregation in Content-Based Routing Networks, ICDCS, 2015.

- High-level idea (a threshold-based method):
  - For each broker
    - Calculate the subscription similarity of all the subscriptions through this broker
    - Once the subscription similarity of a broker exceeds a threshold, aggregate all its subscriptions.

- Subscription similarity
  - $1 - \text{unique subscription range/the whole subscription range}$

![Subscription similarity diagram](image)
Related works

- An illustration (s1 leaves the network)

S1, S2 and S3 are aggregated at b2
Benefit: 2  Cost: 3

S1, and S2 are aggregated at b1, then aggregated with s3 at b2
Benefit: 3  Cost: 3*2

S2 and S3 are aggregated at b2
Benefit: 1  Cost: 0
Problem formulation

- **Cost minimization problem**
  - Save a target amount of network resources, while the amount of false-positive publications is minimized.

\[
\begin{align*}
\text{min} & \quad \sum_{i \in X} C_{ij} \times \lambda_{ij} \\
\text{s.t.} & \quad \sum_{i \in X} G_{ij} \times \lambda_{ij} \geq \theta
\end{align*}
\]

- Where \( \lambda_{ij} \) means the aggregation indicator between subscribers \( i \) and \( j \), the corresponding cost and benefit are denoted as \( C_{ij}, G_{ij} \) correspondingly.

- NP-hard
Subscription aggregation algorithm

Most-Efficient-First Algorithm (MEFA)

- Initialize $X = \emptyset$;
- Find maximum $G_{ij}/C_{ij}$;
- // $\delta$ is a control value
- While (benefit in $X < \theta$ & $G_{ij}/C_{ij} \geq \delta$) do
  - Add $\lambda_{ij}$ into set $S$;
  - Propagate the pairwise aggregation result to the publisher;
  - Find maximum $G_{ij}/C_{ij}$;
- Return $X$. 


Subscription aggregation algorithm

An example of MEFA algorithm

Round1

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Benefit</th>
<th>ΔCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1, s2</td>
<td>2</td>
<td>9*1</td>
</tr>
<tr>
<td>s1, s3</td>
<td>1</td>
<td>9*2</td>
</tr>
<tr>
<td>s2, s3</td>
<td>1</td>
<td>2*2</td>
</tr>
</tbody>
</table>

Subscription update at b2

Round2

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Benefit</th>
<th>ΔCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1, s2</td>
<td>2</td>
<td>2*2</td>
</tr>
<tr>
<td>s1, s3</td>
<td>1</td>
<td>2*2</td>
</tr>
</tbody>
</table>
Subscription aggregation algorithm

Calculation of the incremental cost

Round2

<table>
<thead>
<tr>
<th>Aggregation</th>
<th>Benefit</th>
<th>ΔCost</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1, s2</td>
<td>2</td>
<td>2*2</td>
</tr>
<tr>
<td>s1, s3</td>
<td>1</td>
<td>2*2</td>
</tr>
</tbody>
</table>

s2 and s3 aggregate

Incremental cost  2*2

Then s1 and s2 aggregate, after s2 and s3 has aggregated.

Incremental cost  (3-1)*2
Subscription aggregation algorithm

Observations in the MEFA algorithm:

- Incremental benefit is based on individual pairwise units (which can be overlapped) -- linear.
- Incremental cost is calculated from clusters (which are non overlapped) -- sub-modular.

Theorem: The MEFA achieves the $1 + c \ln \theta$ asymptotic approximation ratio, where $c$ is a constant value.
Subscription tree construction

- The construction idea
  - The communication delay \(\quad\) physical distance.
  - The unique subscription range \(\quad\) social distance.

- The traditional method
  - Only consider the communication delay to construct the subscription tree. May achieve relative poor performance in the subscription aggregation.

- Question
  - How to jointly consider the subscription tree construction in these two dimensions?
Subscription tree construction

- An example

For s3, it has two options:
  In subscription tree 1, s3 has smaller delay, larger unique subscription range of b1
  In subscription tree 2, s3 has larger delay, smaller unique subscription range of b1
**Subscription tree construction**

- **Greedy algorithm**
  - Balance the social distance and the physical distances between subscribers.
  - Call BFS algorithm to generate a subscription tree;
  - Initialize $X = \emptyset$;
  - For $i = 1:n$ do
    - If the subscriber $i$ which can be reassigned to another broker $j$;
      - // Denote the $a_{ij}$ as the new assignment for subscriber $i$ to broker $j$.
        - Add $a_{ij}$ into $X$;
      - // Denote the $\Delta m_{ij}$ as the unique subscription range decreasing for $b_j$, due to $a_{ij}$.
        - // Denote the $\Delta d_{ij}$ as the hop count increasing for $s_i$ due to $a_{ij}$.
    - Find maximum $\frac{\Delta m_{ij}}{\Delta d_{ij}}$ in $X$;
    - While ($X \neq \emptyset$ & $\frac{\Delta m_{ij}}{\Delta d_{ij}} > \gamma$) do
      - Change the subscription tree using $a_{ij}$ and delete $a_{ij}$ from $X$;
      - Find maximum $\frac{\Delta m_{ij}}{\Delta d_{ij}}$ in $X$;
  - Subscription tree construction
Experiments

- Trace setting:
  - Real trace
    - Facebook topology trace from Stanford Large Network Dataset Collection (pick first 120 nodes).
      - The node with the largest degree as the publisher.
      - Use BFS algorithm to generate a subscription tree (one pub).
        - 63 leaf nodes are selected as subscribers.
  - Facebook subscription trace from Middleware System Research Group (120 nodes)
    - The min value and max value of a node is regarded as its subscription range's starting point and end point.
    - Average range size 1,687.
    - The subscription range from 267 to 32,947.
Experiments

- Trace setting:
  - Synthetic trace
    - A subscription tree referred from the *.
      - number of nodes from 46 to 96.
      - number of subscribers from 20 to 40.
    - Some topology examples:
      - 46 nodes with 20 subscribers;
      - 96 nodes with 40 subscribers.

Experiments

- **Trace setting:**
  - **Synthetic trace**
    - Node's subscription distribution range
      - Each node has one subscription range.
      - Average subscription size: 20 to 50
      - in a subscription range size of [0,400].
      - Uniform distribution
      - Exponential distribution
      - with parameter 1
  - Subscription tree construction.
    - Physical layer topology
      - Based on the referred topology.
      - Each subscriber randomly has one more connection with the end broker.

Experiments

- Algorithm comparison:
  - **Subscription aggregation**
    - All aggregation (AA) algorithm
      - Aggregate all the subscription range or not.
    - Similarity-pair aggregation (SPA) algorithm
      - Aggregate the subscription ranges based on subscription similarity.
    - Most-efficient-first aggregation (MEFA) algorithm
      - Proposed algorithm.
  - **Subscription tree construction**
    - Distance-only tree construction algorithm (DO)
    - Proposed similarity considered algorithm (DS)
Experiments

- The performance results of subscription aggregation algorithms ($\theta = 100$)

- The proposed MEFA algorithm greatly reduces the subscription aggregation cost, especially when the subscription range distribution is the exponential distribution.

The proposed MEFA algorithm achieve good performance in real trace.

- Synthetic traces
- Facebook traces
Experiments

- The influence of the subscription tree construction ($\theta = 100$)

A good subscription tree can further reduce the subscription aggregation cost.
Conclusions

Subscription aggregation problem
Trade-off between the benefit and the cost

- Partial subscription aggregation
  Greedy solution with approximation bound

- Subscription tree construction
  Further adapt the subscription aggregation
Future Works

- **Churn situation (Subscribers come and leave)**
  - The new coming subscriber can recover the false-positive range

- **multiple subscribers leave in a period**
  - Report the subscription change together, if the time interval is short to save re-configuration times

- Subscription aggregation strategy re-calculation and subscription tree re-build after a period of time
Thanks!

- ning.wang@temple.edu
- jiewu@temple.edu