**CS 347: Parallel and Distributed Data Management**

**Notes XX: Publish/Subscribe**

Hector Garcia-Molina

---

**Publish/Subscribe Communication**

- **Publication**
  - Description: \(D\)
  - Body: \(B\)

- **Subscription**
  - Query: \(Q\)
  - Id: \(I\)

---

**P/S Semantics**

- **Subscribe** (query \(Q\); id \(I\)):
  - add \([Q, I]\) to SDB
- **Publish** (description \(D\); body \(B\)):
  - for \([Q, I]\) in SDB do
  - if \(Q[D]\) then notify(\(I, B\))
P/Q Semantics (Alternative)

- **Publish**(description D; body B): add \([D, B]\) to PDB
- **Query**(query Q; id I):
  for \([D, B]\) in PDB do
  if \(Q[D]\) then notify(I, B)

P/S Features

- *Space decoupling*: interacting parties do not need to know each other
- *Time decoupling*: interacting parties do not need to actively participate at the same time
- *Synchronization decoupling*: publishers and subscribers do not block for each other

Other Communication Models

- Message passing
- RPC (remote procedure call)
- Shared Spaces (bulletin board)
- Message Queues

Description/Query Models

- (flat) Topics
  - e.g.: Topics = \{sports, business, politics, ...\}

Description/Query Models

- (flat) Topics
  - \(T\) is set of possible topics
  - description D is a subset of \(T\)
  - query Q is a subset of \(T\)
  - D matches Q if \(D \cap Q \neq \emptyset\) (empty set)
Description/Query Models

- Topic Hierarchy

- (flat) Topics
  - $T$ is tree of topics (or DAG?)
  - description $D$ is a set of paths in $T$
  - query $Q$ is a set of paths in $T$
  - description path $d$ matches query path $q$
    if $q$ is a prefix of $d$
  - $D$ matches $Q$ if there exists a path in $Q$
    that matches a path in $D$

Description/Query Models

- Key-value pairs

P1:D=[[price, 50], [size, L]]
P2:D=[[price, 80]]
P3:D=[[size, M], [size, L]]
S1.Q=[[price, 50]]
S2.Q=[[price, 50], [size, M]]
S3.Q=[price > 40] AND [size = L]

Matching Descriptions to Queries

Publication
Description: D
Body: B
SDB
Subscription
Query: Q
ID: I
Matching Descriptions to Queries

Publication
  Description: D
  Body: B

Subscription
  Query: Q
  Id: I

Generic Distributed Matching

\[ \text{Pi} \rightarrow \text{P/Sj match} \]
\[ \text{Sj} \rightarrow \text{Pi} \]

- \text{pub to one of \{a,b\}; \{c,d\}; \{e,f\}}
- \text{sub to one of \{a,c,e\}; \{b,d,f\}}

Can use any abort/commit quorums

\[ \text{pub to one of \{(a,b); \{c,d\}\}} \]
\[ \text{sub to one of \{(a,c); \{a,d\}; \{b,c\}; \{b,d\}\}} \]

Issues:
- Replicated data (subs)
- Load balance
- Total work

Simple Cost Model

\[ \text{At node with x subs, y pubs:} \]
\[ \text{work}(x,y) = \text{data}(x) \]
\[ \text{work}(x,y) = y \quad (\text{case (ii)}) \]
\[ \text{data}(x) = x \]

- For 6 node grid: \( s \) subs, \( p \) pubs
- Each node: \( s/2 \) subs, \( p/3 \) pubs
- Scenario I: \( \text{total data} = 6s \quad \text{work}(s/2, p/3) = 6(p/3) = 2p \quad (\text{case (ii)}) \]
- Scenario II: \( \text{total data} = 3s \quad \text{same as before} \]
  \[ \text{total work} = 6s \quad \text{work}(s/2, p/3) = 6(s/2)(p/3) = sp \quad (\text{case (i)}) \]

Compare to single node scenario:
  \[ \text{total data} = s \]
  \[ \text{total work} = p \ (\text{case (iii)}) \quad \text{or} \quad \text{sp} \ (\text{case (i)}) \]
\[ T = \{ t_1, t_2, t_3 \} \]

**Matching with Topic Hierarchy**

- **Publication dissemination tree:**
  - \( t \)
  - \( t/1 \)
  - \( t/2 \)
  - \( t/3 \)
  - \( t/1/1 \)
  - \( t/1/2 \)
  - \( t/3/1 \)
  - \( t/3/2 \)
  - \( t/3/3 \)
  - \( t/1,2 \)
  - \( t/3 \)
  - \( t/3/1 \)
  - \( t/3/2,3 \)

- **Subs for:**
  - \( t/3 \) or \( (t/3/1 \text{ or } t/3/2) \)
  - \( t/3/2 \) or \( (t/3/2 \text{ or } t/3/3) \)

- **NetNews worked like this**

  - **Publication dissemination tree:**
    - \( t \)
    - \( t/1 \)
    - \( t/2 \)
    - \( t/3 \)
    - \( t/1/1 \)
    - \( t/1/2 \)
    - \( t/3/1 \)
    - \( t/3/2 \)
    - \( t/3/3 \)

  - **Note replication**

- **Discussion: Twitter**

  - \( \{ a, b, c, d \} \)
  - \( S(e) = \{ a, d \} \)
  - Pubs by \( e \) have description “e”
  - Body of pub is 140 char max
  - Users periodically check for notifications
Discussion: Twitter

follows inv lists:
S(a): b
S(b): a,c,d,e
S(c): d
S(d): -
S(e): a, d

is-followed inv lists:
S^(-1)(a): b,e
S^(-1)(b): a
d
S^(-1)(c): b
S^(-1)(d): b,c,e
e
S^(-1)(e): b

Strawman Architecture - Centralized

Question: Advantages/disadvantages of "notify by user" storage? Of "pubs by user"?

Strawman Architecture - Distributed

How do we split back end???

Dynamic Dissemination Tree
Dynamic Dissemination Tree

Publication dissemination tree:

new node
interest: t/1
wireless range

dissemination nodes can also publish:
new pub 1: t/1/2
new pub 2: t/3/2

Matching at One Node

• Set \{ [Q_j, I_j] \} of stored subscriptions
• Match one publication p from stream

Index of queries Q_j different from index of descriptions (content)

Example

Inverted Lists:

Subscriptions:

\| a | b | c | d | e | f |
\|---|---|---|---|---|---|
| s1 | a,b | | | | |
| s2 | a,d | | | | |
| s3 | a,d,e | | | | |
| s4 | b,f | | | | |
| s5 | c,d,e,f | | | | |

dissemination nodes can also publish:
new pub 1: t/1/2
new pub 2: t/3/2

Matching at One Node

• Set \{ [Q_j, I_j] \} of stored subscriptions
• Match one publication p from stream

Index of queries Q_j different from index of descriptions (content)
**Example**

Subscriptions:
- s1 (a, b)
- s2 (a, d)
- s3 (a, d, e)
- s4 (b, f)
- s5 (c, d, e, f)

Sample publication:
- a
- c
- a
- f
- b
- c

Inverted Lists:
- s1: (a, b)
- s2: (a, d)
- s3: (a, d, e)
- s4: (b, f)
- s5: (c, d, e, f)

- Intersection of lists (=null) not useful
- Union of lists: {s1, s2, s3, s4, s5} gives candidate subscriptions
- Need to check each candidate (e.g., s1 matches, s2 does not)

---

**Counting Method**

Inverted Lists:
- a
- c
- a
- f
- b
- c

Total count:
- s1: 2
- s2: 3
- s3: 3
- s4: 2
- s5: 4

Distinct word set:
- a
- b
- c
- f

Subscriptions:
- s1
- s2
- s3
- s4
- s5

- As we union lists, count number of times each sub appears
- If count ≥ total, then sub matches

---

**Key Method**

Inverted Lists:
- a
- c
- a
- f
- b
- c

Total count:
- s1: 2
- s2: 3
- s3: 4
- s4: 2
- s5: 4

Distinct word set:
- a
- b
- c
- f

Subscriptions:
- s1
- s2
- s3
- s4
- s5

- Sub in only one inverted list
- Each IL entry contains other terms in sub
- Occurrence table is for fast hash lookup

---

**Summary: Publish/Subscribe**

- P/S semantics
- Various query/description models
- Distributed matching
- Matching at one node