Finding Related Package Updates with a P2P Notification System

Presented by Rubi Boim

Seminar on Managing Information on the Web,
Tel-Aviv University 09.03.08

Motivation

Getting up-to-date with new and related software packages…

- Packages are constantly updated (or added):
  → Bugs fix
  → New innovating features

- How do we discover these updates??

Motivation

- The number of packages is enormous → one could not read all updates..
- We would like to receive only related updates (packages) to our environment
- Using a Scalable, Efficient and Distributed System → P2P

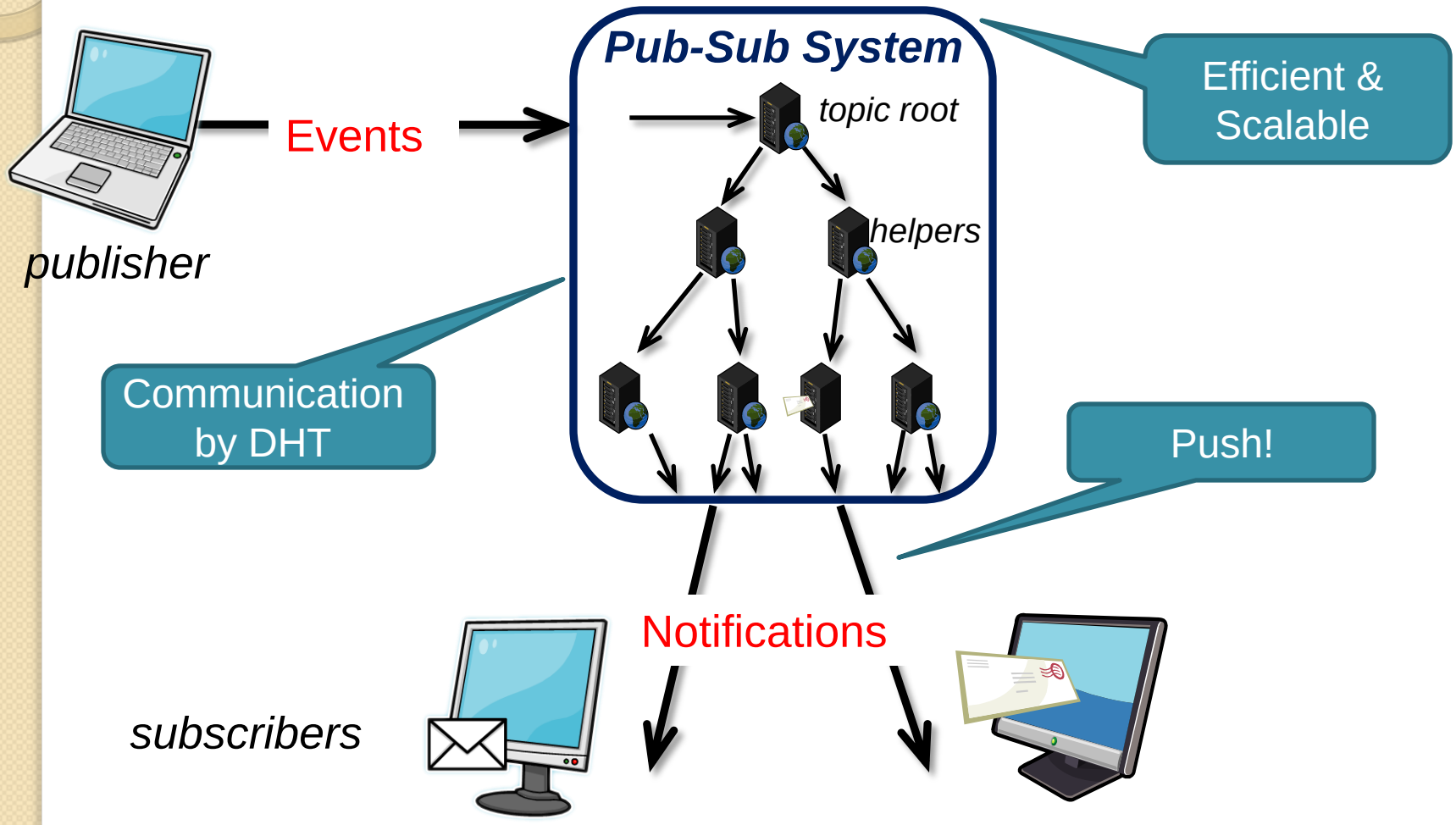
P2P

Publish-Subscribe Systems

- A set of *Publishers*
- A set of *Subscribers*
- Publisher sends an event → Subscribers are notified
- *Topic-Based* - subscription by topic (package) name

P2P

Publish-Subscribe Systems (P2P)



Related Notifications

- Currently, we are getting notifications for <u>only</u> the packages we already have!
- We would like to get notifications for related ones as well
- But what are related notifications??

Related Notifications

Two main properties:

- Similar description (each update is added with a short text description)

- Package dependency
  (each package has a <u>dynamic</u> set of packages it depends on)

- <u>How do we find them??</u>

Solution 1?

- Whenever an updated is published, compare it to all other updates previously published

- Impractical..
  → Too many updates
  → P2P nature
  → Unknown topics (packages)

Solution 2?

- Manual classify all the topics
- Compare the message with the ones published by the corresponding topics

- Impractical..
  → Manual classification → errors (sports can be either basketball, football)
  → Could not cover all fields
  → Dynamics (features, dependencies constantly changing)

Our Solution

- Dynamically cluster topics (packages) together according to their current profiles
- Compare the message with the ones published by the corresponding topics
- What is a topic profile?

Our Solution – Topic Profile

Represents the current state of the topic by:

- <u>Dynamic Dictionary</u>
  constructed by Feature Extraction, Sliding Window, Stemming, Stopwords...

- <u>Dependencies Set</u>
  Represents the current packages dependencies

Our Solution

Use the topics profiles to decide how much two topics are related:

- Two topics with similar dictionaries focus on the same subjects
- Two topics with the similar dependencies are also related

Our Solution - Algorithm

Main Characteristics

- Distributed algorithm
- Local updates operations
- Uses formula *F* to determine cost effective

Our Solution - Algorithm

To complete the picture

- Can’t check all topics
- Dynamic nature (needs to recheck)
- No API for traversing topics

→ *By the users subscriptions*
(users that subscribe to a given topic are likely to subscribe also to related topics)

Finding Related Updates

- Message (Update) profile is similarly extracted
- Compare messages profiles within the cluster

Questions