

A Comparative Study of Pub/Sub Methods in Structured P2P Networks

DBISP2P 2006

Matthias Bender, **Sebastian Michel**, Sebastian Parkitny,
Gerhard Weikum

Max-Planck-Institut für Informatik, Saarbrücken, Germany

Seoul, 2006-09-11

Outline

- 1 Introduction
- 2 Pub/Sub over DHTs
- 3 Experiments
- 4 Conclusion and Outlook

Motivation

Peer-to-Peer

- Became famous through file-sharing applications like Gnutella, KaZaA, Napster, BitTorrent
- Today: Applications like: Skype, pub/sub, Web search

Publish/Subscribe as an Application

- Naturally follows P2P paradigm

Pub/Sub Usage Scenarios

Publisher-to-User

- Services a la Google Alert, News, etc.
- Small number of publishers

User-to-User

- BLOGS, WIKIS, ...
- Users are publishers
- Large number of publishers

Pub/Sub over DHTs

DHTs

- mapping from key to peer
- efficient lookup, e.g. $\log n$ network hops

...term based IR

- Basic trick: use DHT to map hash(term) to directory peer
- Store term specific information at directory peers
- Term \rightarrow List of per-term, per-peer information (e.g. subscriptions, profiles, ...)

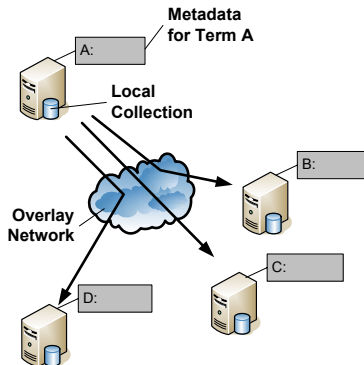


Figure: Publishing information

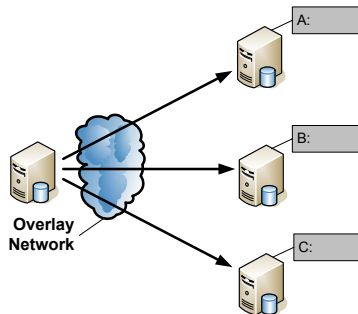


Figure: Retrieval of information

Why this paper?

- A lot of different architectures.
- Different assumptions. Load. Churn....
- Goal: understand tradeoffs and design “sweet spots”

Contribution

- Identify scenarios (Design Patterns)
- Evaluation

Two Different Design Patterns

Store-Sub

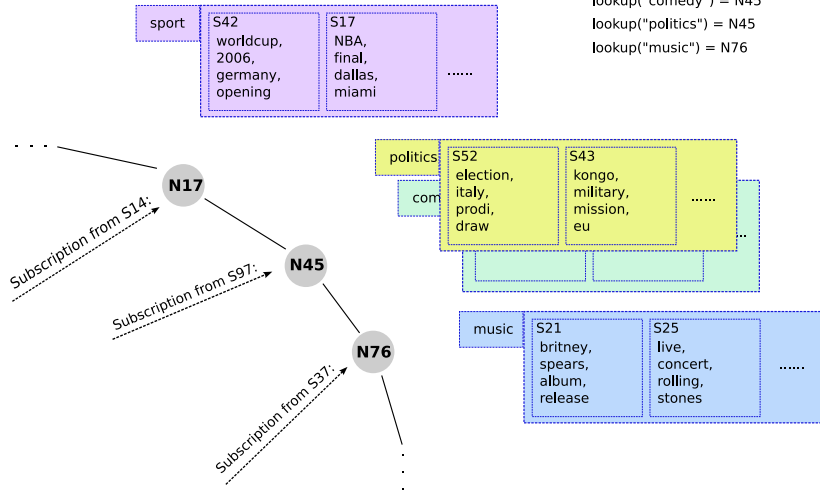
- Subscribers store subscriptions in a DHT
- Publisher retrieve all applicable subscriptions from DHT when publishing a document

Store-Pub

- Publisher store compact, term-specific profiles in a DHT
- Subscribers identify promising publishers based on these profiles
- Subscriber directly subscribe at these publishers

Store-Sub: Store Subscriptions in DHT

lookup("sport") = N17
lookup("comedy") = N45
lookup("politics") = N45
lookup("music") = N76



Complexity of Store-Sub

Message Types

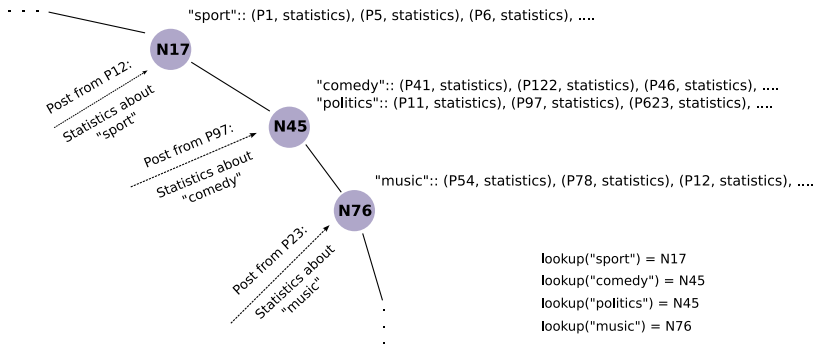
- Subscribers publish/re-publish their subscriptions
- When publishing new content: publisher retrieve subscriptions

	Complexity
Send subs	$O(S_{new} * sub_{avg} * f_s * \log(N))$
Retrieve subs	$O(P * rate * f_p * \log(N))$

Table: Complexity Store-Sub

$N = \#$ peers, $P =$ publishers, $S =$ subscribers, $f_p = \#$ directory nodes to be contacted, rate= publishing rate, $sub_{avg} =$ avg. $\#$ subscriptions per subscriber

Store-Pub: Store Publisher Profiles in DHT



Complexity of Store-Pub

Message Types

- Publisher need to announce their profiles
- Subscriber need to retrieve profiles from directory

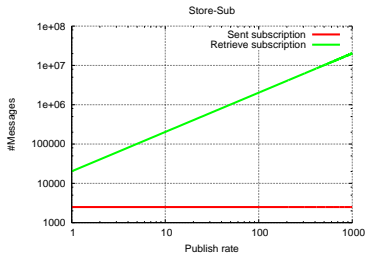
	Complexity
Send profiles	$O\left(\left(P_{new} + \frac{ P }{interval}\right) * F * \log(N)\right)$
Retrieve profiles	$O\left(\left(S_{new} + \frac{ S }{interval}\right) * sub_{avg} * f_s * \log(N)\right)$

Table: Complexity Store-Pub

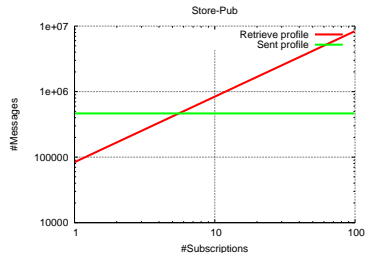
N = # peers, P = publishers, S = subscribers, f_p = # directory nodes to be contacted, interval = refresh interval, sub_{avg} = avg. # number of subscriptions per subscriber, F = number of features.

Discussion

- + Subscribers have full control
- + Subscribers can fine-tune the amount of content they receive
- + Subscribers do not expose information to the public
- + Subscribers can subscribe to particular publishers
 - Publishers need to announce profiles
 - Subscribers base their decision on publishers' profiles describing the past
 - No guarantee to receive all relevant information (approximate)
- ? Chance to be more efficient but with only small decrease of recall



Store-Sub message types



Store-Pub message types

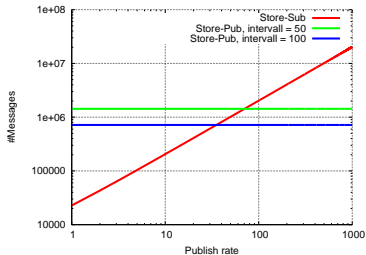
Experimental Evaluation

Two different approaches:

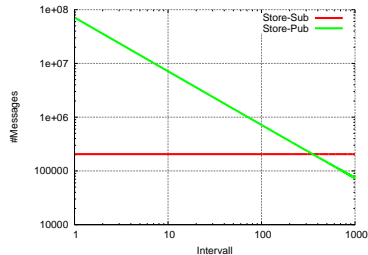
- Analytical results
- Simulation

Analytical Results: Setup

- 100 publishers
- 100,000 subscribers; on avg 3 subscriptions each; on avg 5 terms per subscription



Sensitivity to Publishing Rate

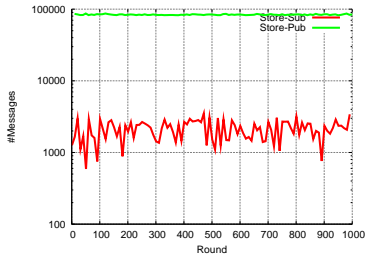


Sensitivity to refresh time interval

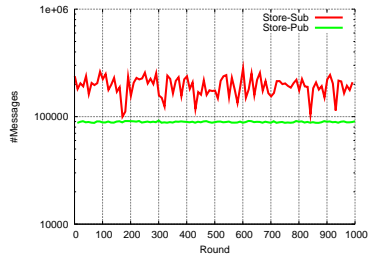
Simulation: Setup

- 10 publishers
- Documents sampled from Zipf-like term distribution
- Initially 5,000 subscribers; 3 subscriptions each; 5 terms per sub on avg
- Churn: each round up to 10 subscribers enter/leave the system

Simulation: Results



Publishing rate 1 per round



Publishing rate 100 per round

Conclusion and Outlook

Conclusion

- Pub/Sub over DHTs
- Identified and characterized two fundamentally different architectures
- Experiments in dynamic networks
- Insight: No “one size fits all” approach.

Outlook

- Evaluate existing systems
- Consider implementation tricks
- Measure accuracy of Store-Pub