# Tagster - Tagging-Based Distributed Content Sharing

Olaf Görlitz, Sergej Sizov, and Steffen Staab

ISWeb, University of Koblenz-Landau, Germany
{goerlitz,sizov,staab}@uni-koblenz.de,
http://isweb.uni-koblenz.de

**Abstract.** Collaborative tagging systems like Flickr and del.icio.us provide centralized content annotation and sharing which is simple to use and attracts many people. A combination of tagging with peer-to-peer systems overcomes typical limitations of centralized systems, however, decentralization also hampers the efficient computation of global statistics facilitating user navigation. We present Tagster, a peer-to-peer based tagging system that provides a solution to this challenge. We describe a typical scenario that demonstrates the advantages of distributed content sharing with Tagster.

## 1 Introduction

Collaborative Tagging systems have become quite popular in recent years. Their success comes from their ease of use for collaboratively annotating and sharing information objects, e.g. photos in Flickr, videos in YouTube, or bookmarks in del.icio.us. However, centralized systems have a number of serious drawbacks, including limited resource allocation (users are charged for additional contents beyond the strict limitation of free space), vulnerability to denial-of-service attacks with possible temporal unavailability of the service, and the need to sign up with multiple services (which all need to be trusted) when different types of resources shall be shared.

Modern peer-to-peer (P2P) systems instead are self-organizing, decentralized infrastructures which can handle huge amounts of available resources. To this end, they are highly attractive for content sharing applications with collaborative annotation (tagging) of contents. In this demo paper we describe *Tagster*, a distributed content sharing application with embedded tagging functionality. It comes as a small Java client program that, similar to a normal file browser, allows the owner to navigate through his locally stored resources (e.g. media files) and to assign arbitrary text labels (tags) to them. Additionally, the tagging information becomes instantly available for all other users that are connected to the Tagster Peer-to-Peer (P2P) overlay network. A distributed index structure ensures that tagging metadata is always available to all other users even if not all peers are permanently online. Furthermore, based on this index structure, Tagster incorporates a tag-based user characterization that takes into account the global tag statistics for better navigation and ranking of resources.

S. Bechhofer et al.(Eds.): ESWC 2008, LNCS 5021, pp. 807–811, 2008.

## 2 Using Tagster

As a motivation example for Tagster, we consider the common scenario of conference participation for computer science researchers.

Our sample user Tom is attending the ESWC 2008 conference. He has submitted two contributions, downloaded three other papers he's interested in, and he has already taken 200 photos and 10 video clips at several conference locations. All that data is stored on his laptop and Tom is looking for an easy way to organize it and share the resources with colleagues and friends.

Using Tagster, Tom annotates his resources with 'eswc' and '2008' by selecting the respective files and folders and typing in these tags. He also adds annotations 'paper' and 'publication' to the papers and some contextual annotations like names (e.g. 'Bill', 'George') and the location (e.g. 'Tenerife') to his photos and videos. Additional media-specific information like photo resolution, format, or creation timestamp is automatically extracted by Tagster from the corresponding media files and added as further tags to their annotation.

After organizing all files, Tom likes to get an overview over the annotations of his resources. The annotation summary is represented by a so-called 'tag cloud' which visualizes aggregated statistics of the tag usage. Tom realizes that his preferable tags for the ESWC trip are 'eswc', '2008' and 'photo'. The click on a tag in the tag cloud opens an overview of associated resources. Multiple tag selections narrow down the set of displayed items.

In the next step, Tom wants to share his resources with other ESWC participants and also see the resources of other colleagues. With Tagster it does not require any additional effort as all tagging metadata is automatically published in the corresponding peer-to-peer overlay network. Thus, searching for 'eswc' and 'photo' will not only return Tom's own locally stored media files but also a list of all resources tagged with those two tags by all other users connected to the overlay network.

Assuming that Tom's search returns a number of photos taken by Bill. Tom can select/download particular resources of Bill and also display their annotations. Furthermore, he can list all resources offered by this user. For better navigation, Tagster displays Bill's profile including an overview of his shared resources, and the corresponding user-characteristic tag cloud.

Finally, Tom aims to find in the network other users with similar interests. A common way for similarity search is the analysis of user-characteristic tag clouds. To return the ranked list of most relevant recommendations, Tagster internally maps the user-specific tag clouds onto vectors in a multi-dimensional feature space. The values of particular features are constructed with respect to local frequencies of tag occurrences in the user's own data, and global tag statistics, analogously to tf*idf feature weighting known from text IR. An important point is the accurate approximation of global tag statistics on particular peers in a decentralized overlay network, which aims to avoid unnecessary high communication overhead.

## 3 Design Choices and Related Work

One drawback of centralized tagging platforms is the limitation to a single media type. Some exceptions exist, like BibSonomy[1] (bookmarks + bibtex), sevenload[2] (pictures + video), or technorati[3] (blogs + video). But still they are far from being a comprehensive platform for organizing all types of personal data. MyTag[4] partially overcomes this limitation by providing a single interface to retrieve combined results from Flickr, YouTube, and del.icio.us. However, manipulating the data is not possible. Therefore, Tagster is designed as a distributed application for organizing any type of locally stored data and globally sharing the tagging metadata.

Peer-to-peer systems have drawn a lot of attention in the last decade. Two major types of peer-to-peer systems exist: unstructured and structured ones.

**Unstructured**peer-to-peer networks are, e.g., Gnutella[5] and its successors. For example, Bibster[3] is a peer-to-peer application based on an unstructured network and that allows for sharing bibliographic data. The bibliographic description and search is based on an ontology. Query propagation is done by semantic routing based on content similarity. However, the centrally-defined ontology impedes user annotation and distributed tagging is hampered as it is difficult to determine global tag use - leading to problems with search and browsing scalability.

**Structured** peer-to-peer networks, like distributed hashtables (DHT) as Chord[8] or P-Grid[1], have the advantage that every piece of distributed information can be located with low overhead, usually within $\log n$ hops where $n$ is the number of peers in the network. There are also sophisticated replication mechanisms available to cope with offline or frequently leaving peers.

The recent popularity of tagging systems has also increased research efforts in order to understand tagging behavior and semantics, or to improve access to data found in such tagging systems. For instance, folkrank[5] is a PageRank like mechanism for recommending resources.

For Tagster, we have focused on approaches based on the vector space model that are suitable for DHT-based P2P systems. An important problem in peer-to-peer systems is the cardinality estimation for item sets, which is necessary for constructing feature weights in our approach. It is possible to directly exploit the underlying network structure [4], as with DHTs, or use a gossip-based approach [6]. However, tracking a huge number of cardinalities cannot be efficiently implemented in such a way. Therefore, we have developed and integrated PINTS, an original dynamic algorithm for computing and updating such feature vectors. The explanation of PINTS is beyond the scope of this demo description and we refer the reader to [2].

[1] http://bibsonomy.org
[2] http://sevenload.com
[3] http://technorati.com
[4] http://mytag.uni-koblenz.de
[5] http://gnutella.org

## 4 System Architecture

All tagging data, i.e. the tag assignments relation between user, tag, and resource, is represented as RDF[6] triples conforming to a tagging ontology. With the possibility to integrate more semantic tagging information and having different tagging ontologies for different users, this provides the necessary flexibility and scalability for future development. Sesame[7] is used as a persistent RDF data store. Queries against the repository are done with the SPARQL[8] query language which allows for flexible and complex queries on the tagging data.

Tagster implements a distributed index structure that holds user-tag, user-resource, and resource-tag relations. For example, accessing a resource ID in the index will return all tags assigned to that resource and all users who tagged it. The index has been realized with Bamboo[9], a distributed hashtable (DHT) implementation similar to Pastry[7] which allows for efficient access to the stored data. DHTs generally ensure that within a given id space, evenly distributed among all peers, all key/value pairs get assigned to exactly one responsible peer and can be stored and retrieved with at most $log(n)$ routing hops. When selecting a tag in Tagster, both the local repository and the distributed index are queried to retrieve the respective resources associated with the tag as well as all users using the tag.

## 5 Tagging Statistics

The knowledge of global statistics about tagging data is useful for different purposes. The frequency of tags, for example, as seen in the whole system or of different users, together with the tags global popularity, can be used to find similar resources or users with similar interests. Moreover, information about frequently co-occurring tags is helpful for discovering semantic relations in the tagging data. The latter can then be used to identify concepts and construct concept hierarchies or simply for clustering tagging data.

The big challenge in a distributed tagging system, however, is to efficiently gather all collection statistics. The main problem is that the user's statistics are kept at the user peer while a collection's cardinality information is stored at the index peer and both are updated independently of each other. In the naive case, an update would require to contact all peers in the network to see if the current statistics is still correct. Obviously, this is not feasible in terms of scalability and message complexity. Instead, we need to estimate these collection cardinalities. Since tagging systems evolve and new tags and resources are continuously added it also has to be flexible enough to accommodate to these changes.

We have developed and implemented such a dynamic algorithm[2] for Tagster. It predicts the future collection cardinality development and automatically

[6] http://www.w3.org/RDF/
[7] http://openrdf.org/
[8] http://www.w3.org/TR/rdf-sparql-query/
[9] http://bamboo-dht.org

updates the respective feature vector entries if the actual deviation violates a predefined error margin. This ensures consistent user statistics in the whole network while keeping the peer-to-peer networks message complexity as low as possible.

## 6 Outlook

The next steps in the development of Tagster will include (i) the collection of experience data from Tagster use, (ii) sophisticated means for recommending resources from the distributed peers, and (iii) additional wrappers for facilitating data collection from further semantic and non-semantic sources. With such support Tagster may offer a viable open alternative to closed, centralized systems.

Our long-term objective is the efficient and effective infrastructure for decentralized, self-organizing Web 2.0 applications which allows for scalable sharing, annotation, searching, and browsing of relevant resources.

**Acknowledgements.** This work has been supported by the EU FP6 research project Tagora - Semiotic Dynamics in Online Social Communities (IST-2006-34721, `http://tagora-project.eu`).

## References

1. Aberer, K., Cudré-Mauroux, P., Datta, A., Despotovic, Z., Hauswirth, M., Punceva, M., Schmidt, R.: P-Grid: A Self-organizing Structured P2P System. SIGMOD Record 32(3), 29–33 (2003)
2. Görlitz, O., Sizov, S., Staab, S.: PINTS: Peer-to-peer infrastructure for tagging systems. In: Proceedings of the Seventh International Workshop on Peer-to-Peer Systems, IPTPS 2008, Tampa Bay, USA (February 2008)
3. Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., Tempich, C.: Bibster - a semantics-based bibliographic peer-to-peer system. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 122–136. Springer, Heidelberg (2004)
4. Horowitz, K., Malkhi, D.: Estimating network size from local information. Inf. Process. Lett. 88(5), 237–243 (2003)
5. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
6. Jelasity, M., Montresor, A.: Epidemic-style proactive aggregationin large overlay networks. In: Proceedings of The 24th International Conference on Distributed ComputingSystems (ICDCS 2004), Tokyo, Japan, pp. 102–109. IEEE Computer Society, Los Alamitos (2004)
7. Rowstron, A.I.T., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
8. Stoica, I., Morris, R., Karger, D.R., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup protocol for internet applications. In: Proc. of the ACM SIGCOMM, San Diego, August 2001, pp. 149–160 (2001)