Interest-Aware Information Dissemination in Small-World Communities

Adriana Iamnitchi
Department of Computer Science
Duke University, Durham, NC 27708
Email: anda@cs.duke.edu

Ian Foster
Department of Computer Science
University of Chicago, Chicago, IL 60637
Email: foster@cs.uchicago.edu

Abstract

Information dissemination is a fundamental and frequently occuring problem in large, dynamic, distributed systems. We propose a novel approach to this problem, interest-aware information dissemination, that takes advantage of small-world usage patterns in data-sharing communities. These small-world characteristics suggest that users naturally form groups of common interest. We propose algorithms for identifying these groups dynamically, without a need for explicit classification of topics or declaration of user interests. These algorithms use information about the data consumed by users to identify, via online computation, groups with similar interests.

As a proof of concept, we apply this methodology to the problem of locating files in large user communities. Using real-world traces from a scientific community and from a peer-to-peer system, we show that proactive information dissemination within groups of common interest can reduce the search load by up to 70%. In addition, this approach naturally supports the efficient discovery of collections of files, a requirement specific to scientific data analysis tasks. We hypothesize that our algorithms can find numerous other uses in distributed systems, such as reputation management.

1 Introduction

Many services in federated distributed systems—large resource-sharing systems spanning different administrative domains, such as the Grid—need access to information that is inherently distributed. Examples include monitoring and information services [8, 29], and resource [20] and file location [5].

Centralized solutions that collect all information to a single access point can work well for small and stable communities, but encounter scalability, reliability and cost limits at larger scales. Decentralized or hybrid solutions are necessary for supporting larger, heterogeneous, and dynamic communities.

Decentralized information solutions span a continuum. At one extreme, there are systems based purely on *query propagation*, such as most file-location solutions in peer-to-peer networks [6, 10, 33]. At the other, there are systems based entirely on *information propagation*, serving hard requirements for low response latency [32, 29]. In between these extremes, solutions based on proactive replication of information have been proposed to support searches in structured geometries [18], such as distributed-hash tables [31, 36]. However, these latter solutions are oblivious to user interests.

Given limited available bandwidth and storage space, inaccurate knowledge of global state, and challenges such as scale and dynamic behavior, it is often inefficient or even infeasible to disseminate information everywhere. Ideally, we should send data only to where it is wanted. Unfortunately, we do not know what people *will* want in the future. However, in a previous study [22] on three file-sharing communities, we discovered small-world patterns in data usage. These patterns suggest that users naturally form interest-based groups. This result suggests the approach that we pursue in this paper, which is to disseminate information selectively to groups of users with common interests. To this end, we propose a decentralized algorithm for identifying groups of users with similar interests. This algorithm does not require any classification of topics or explicit declaration of user interests. Instead, it uses information about the data consumed by users to identify, via an online computation, groups with similar interests.

In order to enable evaluation of this concept and algorithm within the context of a real application and user community, we work with file location as a case study. File location is an interesting case study because it is a building blocks for distributed data-sharing communities. In addition, despite the many solutions from areas such as P2P [6, 35, 11, 41] and Grids [5], specific requirements remain unsatisfied. For example, within scientific computing, a frequently occurring requirement is to obtain access simultaneously to multiple input files, potentially distributed on remote storage locations, for purposes of data analysis [1, 21, 30]. We show that disseminating information in interest-based groups identified with the algorithms proposed in this paper naturally supports such searches for collections of files.

Section 2 gives an overview of our solution and places it in the context of related work. Section 3 presents our decentralized online algorithms for identifying groups of users with shared interest. Section 4 grounds the discussion of interest-aware information dissemination in the context of locating files in a file-sharing community. Section 5 describes our experimental setup and results, and Section 6 concludes.

2 Overview and Related Work

In [22] we analyzed the usage patterns in three file-sharing communities by focusing on the overlap in users interest in data. We defined a new data structure: *the data-sharing graph*, that connects users with similar interests (inferred from the files they request). We consider two users have similar interests if the number of common requests is above a threshold μ over a time period τ. Thus, in a data-sharing graph nodes are users and edges connect users with more than μ common file requests over a period τ.

We built families of graphs of parameters μ and τ using traces from three file-sharing communities: the D0 high-energy physics [9], the Kazaa peer-to-peer file-sharing system [26], and the web user community as seen from the Boeing traces [3]. We showed that for various such graph families (that is, for different values for μ and τ), all data-sharing graphs have small-world properties: a large clustering coefficient, that demonstrates that users form tightly knitted communities of similar interests; and small average path length, that show that traversing these communities can be done in a small number of hops.

One model to visualize small-world graphs, as proposed by Watts and Strogatz [40], is represented in Figure 1: a loosely connected set of highly connected nodes. This is the basis of our intuition for disseminating information to groups of common interest: once we identify these interest-based groups, we can proactively direct information to where it is likely to be consumed. In addition, we want to identify these groups dynamically, adaptively, and in a decentralized manner.

The contributions of this paper are:

1. A decentralized technique for building overlays that mirror shared interest as revealed by the data-sharing graph (Section 3.1).

2. A decentralized, adaptive technique for identifying groups of users with common interests (Section 3.2). This fairly simple strategy for on-line clustering uses requests for data to infer user interests. Thus, it does not require a classification of topics or explicit declaration of interests and does not rely on centralized components to collect information about user interests.

3. An evaluation of these techniques for interest-aware information dissemination in the context of file location. We evaluate the benefits and overheads of the techniques above using real-world traces [22] from a scientific and a P2P data sharing community.

Interest-aware data replication can be applied in various contexts. For example, in a wide-area scientific collaboration typical of Grid environments, data replication is a means to reduce latency, bandwidth consumption, and hot spots. A proactive replica placement strategy could benefit from detecting the group of users to which specific data might be of interest and replicate data closer to them. Similarly, when computation is less expensive than moving data across the network, information about groups of interest can be used to decide where data should be precomputed. The same intuition can be used for content distribution networks to prefetch data of interest closer to users and for recommendation systems to infer potential future interests based on the distance between nodes or on membership to an interest group.

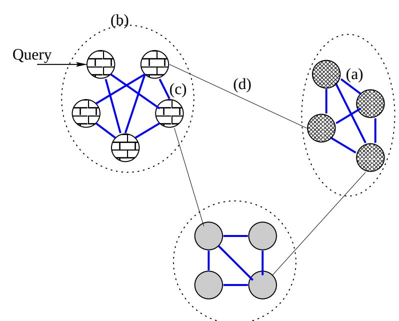

Figure 1. Small worlds seen as loosely connected collections of well connected clusters.

Multiple areas relate to the contributions presented in this paper.

Clustering has been studied intensively (see [23] for a comprehensive survey). Clustering is the problem of discovering natural groups in data sets by partitioning the given data points into a number of groups, such that points within a group are more similar to each other than to points from different groups. Similarity between data points can often be represented as a (weighted) graph. Solutions generally rely on a predefined number of resulting clusters, a specified cluster size limit, and global knowledge of the data set.

Identification of groups of interest has been studied in contexts such as inferring communities on the web graph [12, 13, 25] and identifying communities of scientists connected in a co-authorship graph [15]. The solutions proposed require information that cannot be provided in the data-sharing graph context, such as the location of nodes with particular properties (i.e., a sink) or the full graph topology.

Recommendation systems [27] are intended for off-line processing and rely on global knowledge of the graph topology.

Interest-based guided search for files relies on guided propagation of file requests rather than information dissemination. Content-based routing has been explored in [6]. History-based query routing [35] has been explored for reducing the Gnutella flooding costs by using peers who proved helpful in the past. In [7] searches are guided in unstructured overlays using the market-basket concept applied to files *stored* on nodes.

Dissemination of information to target groups has been proposed in Kelips [18] in support for searches in structured geometries. Information is disseminated within groups of users clustered based on their node IDs (and thus, oblivious to usage behavior).

Group communication mechanisms [24] are complementary to our work: our techniques identify interest-based target destinations to be provided to these mechanisms.

3 Identifying Interest Locality in Small-World Communities

The large clustering coefficient of the data-sharing graphs suggests the existence of groups of users with common interests. Our approach is to dynamically identify these groups and disseminate information that better fits the interests of each group. Specifically, we propose an overlay construction algorithm capable of allowing users with similar interests to learn of each other and connect in a geometry that mirrors the data-sharing graph (Section 3.1). Once the overlay is in place, users are able to identify who else is in their own group starting with the direct neighbors in the data-sharing graph(Section 3.2). Disseminating information in these groups can then be implemented using gossip protocols for group communications (Section 3.3).

3.1 Overlay Construction

The overlay construction component builds the data-sharing graph in a decentralized, adaptive manner. Its purpose is to allow users with similar interests to learn of each other without relying on a central meeting point.

We propose a solution that uses data storage nodes as meeting points for users interested in the same data. Figure

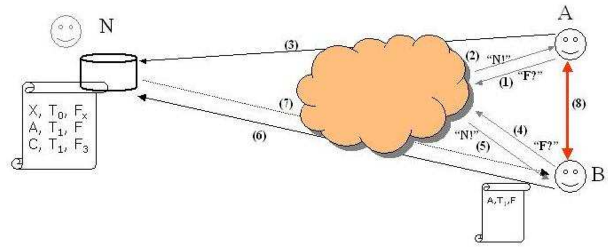

Figure 2. Overlay construction: (1) Node A requests file F; (2) A learns that F is stored on node N; (3) A contacts N to fetch file. N logs A's request and time; (4) Node B requests file F; (5) B learns that F is stored on node N; (6) B contacts N to fetch F; (7) N sends the relevant log with latest requests for file F; (8) B initiates dialog with A

2 illustrates this scenario in a file-location context: node A learns for the file location mechanism where file F is stored. When A *fetches* file F from N, N will record A's interest in its file and the access time. If user B then fetches the same file within a time shorter than an aging time, N informs B about A's interest in the same file F. Thus, B can contact A if it is interested in connecting to new peers and if A's interests satisfy B's interest similarity criterion. For example, B may choose to contact A only if more than μ common files have been requested within an interval τ. The value μ can be adapted by each node independently to satisfy local constraints, such as the number of connections or better interest overlap. This approach allows for true adaptability to node heterogeneity relating, for example, to communication capabilities or high load/reduced processing power.

It may be difficult to determine a good threshold μ that differentiates between popular interests and specific user interests, especially when item popularity follows a Zipf distribution, as it is often the case [4]. One solution is to weight the sharing of files by their popularity: one access to a highly popular file is a weaker proof of common interest than an access to a less popular file. However, in this study we consider μ fix for ease of experimentation.

This overlay construction mechanism uses storage nodes as meeting points for users interested in the same files. Hence, it is fully decentralized (since storage is distributed) and adaptive to changes in user interests: periodically, nodes reevaluate their relationships with their neighbors to adapt to changes in interests or select a smaller/larger number of neighbors.

We acknowledge this is a sketchy presentation of the mechanism and a rigorous design and performance evaluation is in out plans for future work. The experimental evaluations use an idealized, off-line version of this overlay, that helps better focus on the costs and benefits of information dissemination rather than on the costs of maintaining the overlay in a dynamic context.

3.2 Cluster Identification

Once the overlay that mirrors the data-sharing graph is in place, users need to identify the group with similar interests. Given the dynamics of a potentially large user community with intermittent participation and changing interests, we can assume only local knowledge about the graph topology. The challenge is, thus, to design a decentralized clustering algorithm that can be run by each node and needs only local information about the overlay topology.

However, a node only needs to know which of its neighbors in the overlay are in its own group of interests. This observation reduces the clustering problem to an edge-labeling strategy: each node labels its own edges, based on local information, as *long* or *short*. Nodes exchange information of common interest along the short edges. Thus, a cluster is defined by a collection of nodes connected by short edges.

Multiple definitions can be proposed for long/short edges. We propose one definition that we call *triad labeling* that mirrors the small-world topology: the tightly connected nodes—which may have large clustering coefficients—are connected by short edges. We define a short edge as an edge that is part of a triad (triangle in graph) or is a dead end (Figure 3). Otherwise, an edge is labeled long.

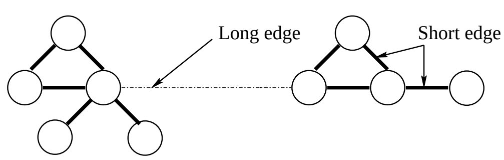

Figure 3. Triad labeling: edges in triangles and dead-ends are considered *short*, while the others are considered *long*.

3.3 Disseminating Information within Clusters

The clustering component alone allows nodes to learn which of their direct neighbors in the overlay have shared interest, but it does not allow them to learn of nodes that are not direct neighbors but to which are connected via paths of short edges. While knowing only direct neighbors connected by short edges is sufficient to propagate information in the entire cluster, there are some disadvantages. First, the time to propagate information is highly dependent on the cluster geometry. Second, failure of a node can cause the cluster to disconnect and hinder information dissemination.

Gossip-based mechanisms have been used extensively [14, 16, 17] for creating and maintaining group membership in dynamic communities. Gossip protocols [24, 39] have also been employed as scalable and reliable information dissemination mechanisms for group communication. Each node in the group knows a partial, possibly inaccurate set of group members. When a node has information to share, it sends it to some constant number of randomly chosen nodes in its set. A node that receives new information processes it (for example, combines it with or updates its own information) and gossips it further to some random nodes from its own set.

For the membership mechanism, we employ SCAMP (Scalable Membership Protocol) [14]: nodes start with a small number of contact addresses (the neighbors in the data-sharing graph) and build up a partial view of the cluster membership of size $O(log(N))$, where N is the number of nodes in cluster. A larger membership view is beneficial but not mandatory, since $log(N)$ acts as a performance threshold: the probability that a notification reaches everyone exhibits a sharp threshold if $log(N)$ or more members are known. Alternatively, smaller than $log(N)$ membership size hinders the performance of disseminating information to all nodes in cluster.

In addition to providing performance and reliability guarantees, knowing cluster membership makes the solution viable for solutions that can exploit the naturally emerging groups of interests, such as proactive data replication. Also, it can be used to build a fully decentralized file-location mechanisms that works by disseminating information within clusters and propagating requests among clusters [19].

4 Case Study: File Location

To ground our solution in a realistic context, we evaluate it in the context of file location. We stress this is a proof of concept rather than a final solution: we focus solely on the performance metrics specific to information dissemination and how they improve the file-location performance, but we do not evaluate the resulting file location mechanism.

In a file location context, disseminating information in clusters of shared interests can be used to ease the load on a central index that serves file location requests (in a Napster-like design) or as a component in a fully decentralized algorithm.

In the particular context of file location our generic solution presented in Section 3 needs to be instantiated in two respects: the information to disseminate and the performance metrics.

4.1 What to Disseminate

The intuition of the data-sharing graph suggest that location of files discovered in the previous interval τ is the relevant information to share between users in the same group of interest. The format of the information sent is thus:

```
<downloader IP, storage IP, filenames>
```

The first field contributes to the membership information while the remaining two fields build up the file location database. The first field will always refer to nodes in the current cluster: this can be easily proven from the observation that gossips propagate along short edges, hence they collect information only from nodes that are connected via short edges.

In contrast, `storage IP` can be also from outside the current cluster, depending on what files are advertised: nodes may advertise own files as well as remote files of which they learned recently. If local files are advertised, then the first two fields are identical.

The list of files `filenames` can be compressed using Bloom filters [2] or left uncompressed to support partial matches.

4.2 Performance Metrics

The obvious performance metric in proactive information replication is hit rate: what percentage of requests can be served from proactively replicated information. In the context of file location, the higher the hit rate, the lower the average response latency (since a hit is served from local storage), thus saving communication costs in terms of bandwidth and latency. The costs in proactive information replication are storage and communication costs. By grounding information dissemination in the context of file location and using real traces we are able to evaluate these costs in realistic applications.

5 Experimental Evaluation

We conduct our experiments using two sets of traces, from D0 and Kazaa. The D0 experiment [9] is a scientific collaboration comprising hundreds of physicists from more than 70 institutions in 18 countries. Its purpose is to provide a worldwide system of shareable computing and storage resources that can together solve the common problem of extracting physics results from about a Petabyte of measured and simulated data. In this system, data files are read-only and typical jobs analyze and produce new, processed data files. The tracing of system utilization is possible via a software layer (SAM [28]) that provides centralized file-based data management. We used traces from the first six months of 2002, amounting to about 23,000 jobs submitted by more than 300 users and involving more than 2.5 million requests for about 200,000 distinct files. A data analysis job typically runs on (and thus requests) multiple files (117 on average).

Kazaa is a popular peer-to-peer file-sharing system with an estimated number of more than 2 million concurrent users as of February 2005 [34]. We had access to five days of Kazaa traffic, during which 14,404 users downloaded 976,184 files, of which 116,509 were distinct. The user population is formed of Kazaa users who are clients of an ISP.

For each set of traces we consider multiple instances of data-sharing graphs (i.e., for various values of the time interval τ and the similarity criterion μ). For each definition, we build the data-sharing graph for the interval τ and "freeze" it at the end of the interval. We then apply the triad labeling technique to identify clusters of interest and disseminate information within clusters. In all cases, the information disseminated is about location of files. We analyzed two instances: in *experience sharing*, the location of files discovered in the previous interval is disseminated. In *summaries sharing*, the location of the files *stored* on the local peer is disseminated. In both cases we measure the hit rate: what percentage of queries sent during an interval τ can be answered from the information just disseminated. Due to space constraints, we present only the most relevant case, the experience sharing scenario. Summary sharing setup and results are presented in [19].

We do not rely on caching information disseminated more than one interval τ ago because of the volatility of the system: the overlay and consequently the clusters change frequently to adapt to user interests; files and nodes unpredictably join and leave the system.

Note that in a real implementation, the overlay varies and adjusts to user behavior smoothly over time. This may lead to better hit rates, since the overlay and clusters adapt faster to changing in user interests.

5.1 Clustering

When gossiping information within clusters, the cluster size determines the number of messages exchanged and the time it takes for all group members to receive a new piece of information. We thus evaluate the size of the clusters resulting from the triad labeling technique.

Triad labeling leads to skewed cluster sizes in all traces. Typically, a large cluster that comprises 60-80% of all nodes is created, along with a fair number of much smaller clusters. Figure 4 presents the average cluster size for each of the data-sharing graph definitions analyzed. The average cluster size is in the orders of tens of nodes, but given the different population sizes, the number of clusters resulted is highly different, from under 10 in D0 to 150 in the Kazaa.

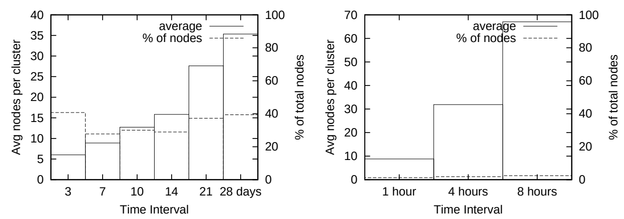

Figure 4. Average cluster size resulting from triad labeling in D0 (left) and Kazaa (right). The left-hand axis shows the average number of nodes per cluster. The right-hand axis shows the percentage of the average cluster size from the total number of nodes.

To understand whether the unbalanced clusters resulted are due to the clustering approach or to the graph topology, we compare it against a global-knowledge clustering algorithm [37, 38]. Similar results were obtained: a large cluster often comprising more than 60% of all nodes and many small clusters. Moreover, the clusters obtained with the two methods overlap significantly.

Obviously, the best case scenario is to disseminate all information everywhere. A large cluster that includes most of the user community gets close to this scenario and biases the results. We will show in the following sections that, even when we ignore the largest cluster and its optimistic benefits on information dissemination we obtain good hit rates. Nevertheless, designing decentralized clustering techniques that limit the size of resulting clusters remains a relevant problem for future work.

5.2 Sharing Experience

In this scenario, peers gossip the experience accumulated during the previous interval τ. This experience consists of the set of file locations learned (as responses to queries) during this interval.

Caching influences when the same user asks same requests in consecutive intervals: their later requests can therefore be answered from the local cache and should not be attributed to information dissemination. However, the percentage of requests of this sort is low: it varies up to 15%, with averages between 5% and 10% for different intervals in D0. Surprisingly, the Kazaa users show the same behavior, with similar average cache rates (around 7%): while it is conceivable that scientists repeat requests for the same data to run their computations, it is less intuitive that a Kazaa user would repeat requests for same music at intervals of one hour. This behavior may be explained by unsatisfactory answers: files were not found, were not downloaded properly, or were corrupted.

The results in this section are based on the assumption that all requests are answered within the interval in which they are asked. This assumption is correct in a hybrid scenario in which information dissemination within clusters is used to reduce the load on a centralized index such as the one used in Napster.

Figure 5 shows the average hit rate for different data-sharing graph definitions. For D0 the hit rate decreases with the increase in duration. A possible explanation for this different behavior is a stronger time locality in scientific communities: scientists repeatedly request the same files as they repeat their experiments on the same data. Once they finish with that region of the problem space, they move to a new set of files. This assumption may be true at the group level but it is not true at the individual level: the percentage of a user's repeated requests at consecutive intervals in general increases slightly with the duration of the interval (6.35%, 7.5%, 7.6%, 8.7%, 9.1%, and 7.75% for $\mu = 100$ and τ =3, 7, 10, 14, 21, and 28 days).

The benefits of disseminating information in D0 are significant: more than 50% of queries do not impose any costs on the network and their answer latency is that of a local lookup. The price for these advantages is paid in dissemination costs and storage of the information disseminated. Table 1 presents these costs: storage space required per node is a function of the number of files disseminated within a cluster; communication costs per node depends on the number of files disseminated within the cluster and the size of the cluster.

Table 1. Experience dissemination costs in D0.

Data-sharing graph (τ, $\mu = 100$)	Storage/node # Filenames	Storage/node MB	MB/node sent during τ
3 days	8787	2.6	4.66
7 days	18182	5.2	11.36
10 days	24912	7.1	18.05
14 days	32239	9.2	25.40
21 days	44911	12.8	42.47
28 days	56924	16.3	58.10

The longest filename in the D0 traces has 183 characters. Assuming a pair filename–location takes 300 bytes on average, the storage cost to maintain all file information disseminated in cluster is obtained by multiplying the number of files with the storage cost per file. Over a three-day interval this cost is 2.6MB per node. For the 28-day interval, this cost grows to 16MB per node.

Given the gossip-based information dissemination mechanisms used, the communication costs are estimated by multiplying storage costs with the natural logarithm from the number of nodes in cluster. (This is because each message needs to be gossiped to approximately $ln(N)$ peers in cluster to ensure the message will reach all N members [24].) Consequently, communication costs are not high, either: 4.6MB of data transmitted per node in a 3-day interval, growing to 58MB for the 28-day interval case. In D0, there are fewer than 200 nodes in the network during any of the intervals considered: this results in at most 12GB of data exchanged over a month. To put this in context, compare with the Gnutella traffic as measured in late 2000 [33]: 6GB per node or 1.2TB of data exchanged by 200 Gnutella nodes over a month.

In Kazaa (Figure 5, right) the hit rate is constantly high (around 70%) for all durations, in spite of the decrease in the user's cached repeated requests (from almost 8% for one-hour to 5.5% for 8-hour intervals). On the other hand, the influence of the largest cluster increases with duration: from 13% for 1-hour to 44% for 8-hour intervals.

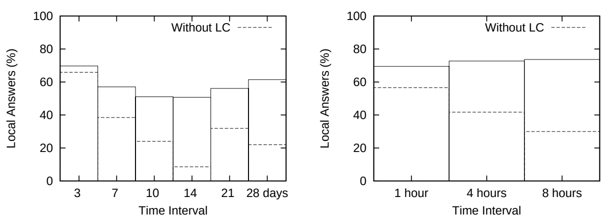

Figure 5. Hit rate with experience dissemination in D0 (left)and Kazaa traces (right). Dashed lines show hit rate when the effect of the largest cluster obtained with the triad clustering method is ignored.

Table 2. Experience dissemination costs in Kazaa.

Data-sharing	Storage/node		MB/node
graph (τ, $\mu = 1$)	# Filenames	MB	sent during τ
1 hour	1345	0.3	0.6
4 hours	5303	1.5	5.2
8 hours	10178	2.9	15.0

Dissemination costs for Kazaa are smaller (Table 2): 3MB of storage are necessary for storing the data disseminated over 8-hour intervals and 15MB of data are to be sent by each node over an 8-hour interval. For about 3000 nodes, about 45GB of data are transferred over 8-hour intervals, which is 4 times less data than transferred by the same number of Gnutella users at the end of year 2000.

The benefits of disseminating experience within clusters formed with triad labeling are significant for both communities, ranging from 40 to 70% of the queries being answered from local storage. Also, eliminating the optimistic effects of the imperfect clustering algorithm, the hit rate is between 30% and 65%. Shorter time intervals for the data-sharing graph are preferable to keep costs down. Interestingly, the average hit rate over all except the largest cluster is higher for shorter intervals.

5.3 Disseminating Information in Random Clusters

To isolate the effects of interest-aware information dissemination from those of general information dissemination, we performed the same experiments by disseminating data in random groups with the same number of users as resulted from triad labeling. This setup maintains the same storage and communication costs. We measured again the hit rate due to information dissemination. The results differ drastically from the results presented in Figure 5: with random clustering, the average hit rate is under 5% for both D0 and Kazaa traces when the largest cluster is ignored. Even for the largest cluster alone the hit rate is significantly (10–30%) lower than in the triad-labeling experiments.

These results clearly differentiate the benefits of information dissemination from the benefits of *interest-aware* information dissemination.

5.4 Support for collections

In D0 requests are for collections of files (on average, 117 files per request), a feature characteristic to scientific communities. Figure 6 presents the support for collections with experience-sharing in D0: it presents empirical cumulative distribution functions for various intervals.

The results show that 40 to 50% of collections have all their files disseminated within the same cluster. About 40% of requests for collections of files found none of the files locally. On average, 44 to 60% of files requests on a collection are known to the cluster due to experience sharing. In file location solutions based only on request propagation, a request for a set of N files translates into N requests for one file. In our case, a request for a set of N files propagates on average less than $\frac{N}{2}$ requests into the network.

Interestingly, the support for larger collections is better: collections with more than half of their files found locally are larger, on average, than collections with less than half of their files found locally (Table 3).

6 Conclusions and Future Work

We have introduced and demonstrated the utility of a new approach to disseminating information based on the dynamic computation of shared interest communities. This work was inspired by our previous study of file-sharing

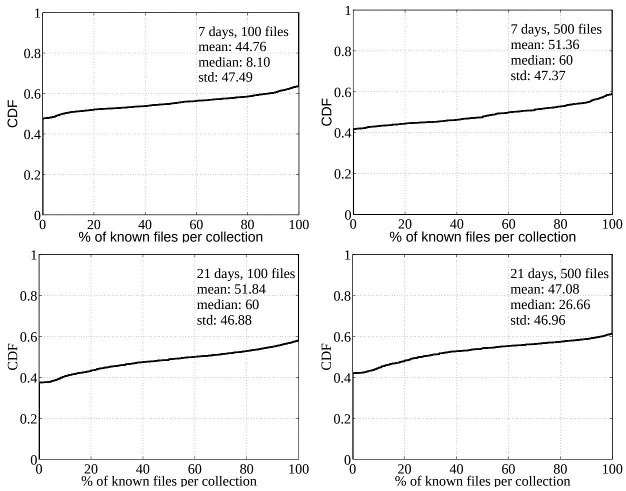

Figure 6. Cumulative distribution function of files per collection found locally due to experience dissemination. A collection is the set of input files for the same computational task (or *project*, in the D0 terminology).

communities, in which we discovered small-world data sharing patterns. In the work presented here, we show how these patterns can be exploited to optimize file location by proposing a decentralized overlay construction technique that adaptively mirrors the data-sharing graph and a dynamic clustering technique based on local topological information. These two techniques allow users to detect groups of interest dynamically without subscribing to predefined topics. Instead, clues about commonal interests are picked up from storage nodes that act as distributed meeting points for users to learn of each other.

We have evaluated the feasibility of these ideas in the context of information dissemination for supporting file location. Using trace-driven experiments, we show that exploiting interest locality can save up to 70% of search costs, with only modest storage and off-line communication costs.

This initial study encourages us to continue to improve and evaluate our techniques. We are planning to evaluate experimentally our overlay construction techniques and their stability in the case of a community with rapidly changing interests. We also plan to investigate how to set the data-sharing graph parameters μ and τ adaptively, and to understand the impact of changing these values on the overall solution.

On a more ambitious note, we plan to investigate other mechanisms that may exploit small-world patterns in data-sharing communities. In particular, we are interested in investigating proactive file-replication techniques that place copies of possibly large data closer to interested users. This

Table 3. Average size (in number of requested files) of collections for which more than 50% files (respectively less) are found locally. Results from multiple data-sharing graph definitions (τ, μ) are showed.

Interval	$\mu = 100$ files		$\mu = 500$ files	
τ	$> 50\%$	$< 50\%$	$> 50\%$	$< 50\%$
7 days	162.52	123.24	203.59	101.36
14 days	154.02	121.60	185.16	144.16
21 days	139.79	94.71	153.32	108.70
28 days	83.20	305.90	50.78	164.94

problem will open a new set of questions, since network locality, in addition to interest locality, will significantly influence the feasibility and success of such a solution. Other problems of interest concern reputation and trust, two potentially important metrics that may have some relationship to data sharing.

References

[1] M. Allen and R. Wolski. The Livny and Plank-Beck problems: Studies in data movement on the computational grid. In *Supercomputing*, 2003.

[2] B. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[3] Boeing proxy logs, ftp://researchsmp2.cc.vt.edu/pub/boeing/boeing.990301-05.notes.

[4] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web caching and Zipf-like distributions: Evidence and implications. In *InfoCom*, New York, NY, 1999. IEEE Press.

[5] A. Chervenak, E. Deelman, I. Foster, L. Guy, W. Hoschek, A. Iamnitchi, C. Kesselman, P. Kunszt, M. Ripeanu, B. Schwartzkopf, H. Stockinger, K. Stockinger, and B. Tierney. Giggle: A framework for constructing scalable replica location services. In *Supercomputing*, 2002.

[6] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *International Workshop on Designing Privacy Enhancing Technologies*, volume 44-66, Berkeley, CA, 2000. Springer-Verlag.

[7] E. Cohen, A. Fiat, and H. Kaplan. Associative search in peer to peer networks: Harnessing latent semantics. In *Infocom*, San Fancisco, CA, 2003.

[8] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid information services for distributed resource sharing. In *10th IEEE International Symposium on High Performance Distributed Computing (HPDC-10)*, pages 181–184. IEEE Press, 2001.

[9] The DZero Experiment., http://www-d0.fnal.gov.

[10] F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan. Building peer-to-peer

systems with Chord, a distributed lookup service. In *Proceedings of 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*. Elmau/Oberbayern, Germany, 2001.

[11] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *Proceedings of the 18th ACM Symposium on Operating Systems Principles (SOSP '01)*, Chateau Lake Louise, Banff, Canada, Oct. 2001.

[12] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities,. In *6th ACM SIGKDD*, pages 150–160, 2000.

[13] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[14] A. Ganesh, A. Kermarrec, and L. Massoulie. Peer-to-peer membership management for gossip-based protocols. *IEEE Transactions on Computers*, 52, February 2003.

[15] M. Girvan and M. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99:8271–8276, 2002.

[16] R. A. Golding. *Weak-consistency group communication and membership*. PhD thesis, University of California at Santa Cruz, 1992.

[17] R. A. Golding and K. Taylor. Group membership in the epidemic style. Technical Report UCSC-CRL-92-13, University of California, Santa Cruz, Jack Baskin School of Engineering, March 1992 1992.

[18] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse. Kelips: Building an efficient and stable P2P DHT through increased memory and background overhead. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, 2003.

[19] A. Iamnitchi. *Resource Discovery in Large Resource-Sharing Environments*. PhD thesis, University of Chicago, 2003.

[20] A. Iamnitchi and I. Foster. On fully decentralized resource discovery in grid environments. In *International Workshop on Grid Computing*, Denver, Colorado, 2001. IEEE.

[21] A. Iamnitchi and M. Ripeanu. Myth and reality: Usage behavior in a large data-intensive physics project. Technical Report TR2003-4, GriPhyN, 2003.

[22] A. Iamnitchi, M. Ripeanu, and I. Foster. Small-world filesharing communities. In *Infocom*, Hong Kong, China, 2004.

[23] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[24] A.-M. Kermarrec, L. Massoulie, and A. Ganesh. Probabilistic reliable dissemination in large-scale systems. *IEEE Transactions on Parallel and Distributed Systems*, 14(3), March 2003.

[25] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[26] N. Leibowitz, M. Ripeanu, and A. Wierzbicki. Deconstructing the Kazaa network. In *Workshop on Internet Applications*, San Francisco, CA, 2003.

[27] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January/February 2003.

[28] L. Loebel-Carpenter, L. Lueking, C. Moore, R. Pordes, J. Trumbo, S. Veseli, I. Terekhov, M. Vranicar, S. White, and V. White. SAM and the particle physics data Grid. In *Computing in High-Energy and Nuclear Physics*, Beijing, China, 2001.

[29] M. L. Massie, B. N. Chun, and D. E. Culler. The Ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing*, 30(5-6):817–840, 2004.

[30] E. J. Otoo, D. Rotem, and A. Romosan. Optimal file-bundle caching algorithms for data-grids. In *Supercomputing*, 2004.

[31] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *SIGCOMM*. ACM, 2001.

[32] M. Ripeanu and I. Foster. A decentralized, adaptive, replica location service. In *Proceedings of the 11th IEEE International Symposium on High Performance Distributed Compuing (HPDC-11)*. IEEE, July 2002.

[33] M. Ripeanu, A. Iamnitchi, and I. Foster. Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *Internet Computing*, 6(1):50–57, 2002.

[34] http://www.slyck.com.

[35] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *INFOCOM*, San Francisco, 2003.

[36] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *SIGCOMM*, San Diego, USA, 2001.

[37] S. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, May 2000.

[38] S. van Dongen. MCL – a cluster algorithm for graphs, http://micans.org/mcl/.

[39] W. Vogels, R. van Renesse, and K. Birman. Using epidemic techniques for building ultra-scalable reliable communication systems. In *Workshop on New Visions for Large-Scale Networks: Research and Applications*, Vienna, VA, 2001.

[40] D. Watts and S. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393, 1998.

[41] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *Infocom*. 2002.