InfoBeacons: A loosely-coupled P2P data system

Brian F. Cooper
College of Computing, Georgia Institute of Technology
cooperb@cc.gatech.edu

Abstract

Peer-to-peer data management systems have gained increasing attention lately. Most of these systems are *tightly-coupled* to sources, in the sense that there is tight integration with the schema or data of the source. Here, we propose that another approach is also useful for many applications: *loose-coupling*, where data and sources are treated "as is." Loose coupling provide several advantages, such as increased flexibility and scalability at low cost, to applications that can tolerate weaker query semantics. We describe *InfoBeacons*, a peer data system we are building based on the loose coupling philosophy, and present some preliminary experimental results which show that our system can find good information sources despite loose coupling. We also examine several challenges that must be overcome in order to get high quality results to queries.

1 Introduction

There is an explosion of data available on the Internet. Although in theory the connectivity exists to retrieve any information that is available, data management techniques are needed to deal with the scale, diversity and decentralization of the data. A peer-to-peer architecture seems especially appropriate for managing Internet data stored at scattered, diverse hosts. The goal is to process data scattered throughout the Internet without having to transfer it all to a massive central database.

However, several researchers [7, 4] have pointed out that P2P systems usually lack one trait offered by traditional data management systems: a strong, consistent data model. Recent work [4, 12, 8] aims to apply stronger data semantics to the peers and data in the network. Such systems provide strong semantics by *tightly-coupling* the sources to the P2P data system, either by performing schema-level data integration or requiring peers to export their data (or summaries) to specialized peer data management software.

We are building a peer-to-peer system, called *InfoBeacons*, that takes a different approach to managing Internet data. InfoBeacons organizes sources into a loosely-coupled confederation, utilizing existing software and services at data sources to process information. In contrast to tightly-coupled techniques, InfoBeacons asks only that sources continue to run their existing information services, and then tries to compose these diverse services into a common information system. Special peers, called *beacons*, do the work of finding and organizing information sources, routing queries to appropriate sources, and retrieving, processing and returning results. The goal of a beacon is to "guide" users to sources, where the data will be processed by native services.

Although the tight coupling in other systems is key to achieving precise query results and strong semantics, there are a class of applications for which these goals matter less than the flexibility introduced by loose coupling. The benefits of this flexibility include:

- *Low barrier to entry* - peers are more likely to participate in a system if there is little effort to join and little work required to stay in the system
- *Tolerance of dynamism* - peers may frequently join and leave the system, and may frequently change the data that they are sharing
- *Scalability despite heterogeneity* - there may be hundreds of thousands or millions of peers, and the nature and organization of data may vary widely

Examples of applications in this class include web information search, text mining and multimedia resource location. Indeed, the popularity of systems such as Gnutella, Kazaa and Freenet demonstrates that for certain tasks, a strong, consistent data model is not necessary. Note that we are not arguing against the tight-coupling approach

per se; rather, we are arguing that both tight and loose coupling are interesting approaches to building a P2P data system, and that both should be investigated.

Despite the benefits of loose coupling, technical challenges exist that must be overcome in order to build an effective system. For example, a beacon must determine which sources to gather information from for a given query. In the tightly-coupled approach, sources may be asked to export their data (or summaries) to the rest of the system to aid in source selection. However, many information sources are unwilling to export their data, due both to the cost of exporting data and the value which the source places on the data. A loosely-coupled beacon does not require a source to export its data; instead it submits queries and retrieves results from the source itself. But the beacon still must choose the right sources, and techniques must be developed to perform source selection without a complete picture of the source's data. Other research challenges are identified in this paper.

In this position paper, we outline the design philosophy behind the InfoBeacons system, and provide preliminary evidence that the system is effective despite its loose coupling. Specifically:

- We present the design of the InfoBeacons system, a loosely-coupled peer-to-peer information system, and list the advantages that the system can provide over more tightly-coupled solutions (Section 2)
- We examine experimental results gathered while using our beacon prototype to guide queries to World Wide Web information sources. Our results show that a beacon can effectively choose which sources are useful for a given query. (Section 3)
- We outline open research challenges in the design and implementation of a loosely-coupled system like InfoBeacons. (Section 4)

In Section 5 we survey related work, and in Section 6 we discuss our conclusions. While this paper focuses on the philosophy of loose coupling, elsewhere [3] we examine the details of our implementation, such as the source selection process.

2 InfoBeacons architecture

The architecture of the InfoBeacons system is shown in Figure 1. The key component of the architecture is a *beacon*. Beacons are peers that organize diverse sources into

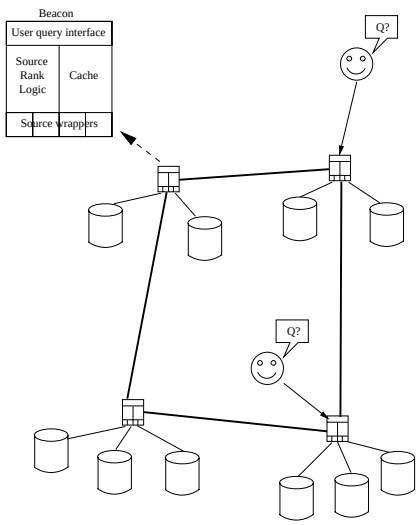

Figure 1: InfoBeacons architecture.

a coherent information system, and are organized into a P2P overlay in a manner similar to a super-peer network. Each beacon is responsible for some of the data sources in the system, and beacons cooperate to query all of the sources. Unlike super-peers, the beacon cannot expect autonomous sources to export data summaries. Beacons may run on the same host as the data sources, or may run on separate hosts.

Users submit queries to a beacon, which decides which of its local sources to forward the query to. When these sources return information, the beacon packages this information and returns it to the user. The beacon also may forward the query to other beacons if necessary. Most of the heavyweight query processing is done by the information sources. The beacon is a lightweight component, designed only to guide the queries to the right sources. When the user has received results from the beacons system, she can either use those results as is, submit a revised query to the system, or contact information sources directly to do more in-depth querying.

In our initial prototype we have focused on choosing the right sources despite the limited information available under loose coupling. Beacons keep caches of previous results from the source. Source ranking logic uses the cache to choose sources for future queries. For more details on the source selection problem, see [3].

The beacon accepts keyword queries, which are an effective "lowest common denominator" that works across a variety of sources. Source wrappers handle communi-

cation between the beacon and the source. Because the beacon is loosely-coupled to sources, these lightweight wrappers deal only with connecting to the source, submitting keyword queries, and receiving results.

We have implemented a beacon prototype that provides the basic functionality described above. More advanced capabilities to deal with the issues raised in Section 4 are currently under development.

2.1 Benefits of loose-coupling

There are several benefits that a loosely-coupled P2P data system can offer to applications. First, such a system requires only that sources do what they are already doing: offer an interface for finding and retrieving results. Many information sources are unwilling to run foreign software in order to participate in a peer-to-peer system, both because of the resource requirements and because they do not trust what is to them essentially "foreign" software. Even if the source is willing to run peer software, they are unlikely to be willing to modify their core information service or export their data to the peer software.

Because a loosely-coupled system like InfoBeacons reduces the barrier to entry of new sources, many more sources can be convinced to participate in the system, and a larger amount of useful information becomes available to users. In fact, a loosely-coupled system focuses on getting as many information sources into the system as possible, and doing the best that it can with the information at those sources. In contrast, a tightly-coupled system attempts to bring all of the information in the system together into a semantically strong framework, but to do so imposes requirements on sources that limit the number participating in the system.

Second, the work to integrate a new source into the system is minimized. In many peer-to-peer systems, peers join and leave the system frequently, and it is important the the cost of joining and leaving is minimized. When a new source joins, a beacon must know only the source's address and the mechanism for submitting queries and retrieving results. It is not necessary to know the schema of the source's data, the content of the source or even the topic of the content. Thus, by being loosely-coupled, the InfoBeacons system focuses on the ability to search sources as soon as they join the network, and for as long as they are in the network. In contrast, in a tightly-coupled system, sources are assumed to be longer lived and more willing to undergo relatively expensive integration at join time. Although we would expect information sources in many cases to join and leave less frequently than say peers in the Gnutella network, the source membership may still be highly dynamic. Sources run on personal web servers, laptops or even PDAs may join and leave frequently as users turn them on and off or travel between wireless hotspots.

A third benefit of a loosely-coupled design is that the system can react to frequently changing information. Examples of sources that frequently update their content include web logs, news feeds and real-time sensor proxies. By pushing most of the actual processing to the sources, InfoBeacons ensures that users can get the most current and up-to-date information. Also, results in Section 3 and [3] demonstrate that the InfoBeacons cache is highly adaptable to changing information at sources.

Though many tightly-coupled systems also use the current information at sources, such systems often choose sources by forming precise characterizations of the source contents. In a highly dynamic and diverse environment, much work is needed to maintain these characterizations. For example, sources may be asked to frequently re-distribute their data or lists of changes.

The common theme in all of these advantages is the tradeoff between scalability and flexibility on one hand, and result semantics and precision on the other. Thus, both the loosely-coupled and tightly-coupled approaches to peer-to-peer data management are important, as different applications have needs that trade off precision and flexibility in different ways.

3 Performance results

We have conducted preliminary experiments to evaluate the effectiveness of InfoBeacons. Our primary goal was to demonstrate that our beacons prototype could efficiently locate appropriate sources for a query despite having limited information and cooperation from the source. To do this, we gathered 26,938 web pages from 100 websites. Each website's data was modeled as different information source, and each source used TF/IDF ranking, a common information retrieval metric. A source returned documents for a query if the normalized TF/IDF score was at least 0.1 (from the range 0...1). To model dynamism, only 50 percent of the pages were added to sources at the beginning of the experiment; the remainder were added concurrently with queries. Simul-

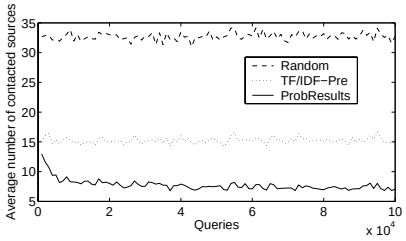

Figure 2: Beacon performance results.

taneously, 50 percent of the documents were removed at some time during the query process. We generated 100,000 queries, each by randomly selecting words from a randomly selected document.

For this experiment, one beacon was used to choose sources for each query from among the 100 sources. Our source ranking function, called *ProbResults*, calculates the probability that a source will return a document containing the query words from the documents in the cache. We compared against *Random* source selection, and also *TF/IDF-Pre*, where sources export their data to be precached by the beacon, and TF/IDF scores are used to rank each source's collection as if it were a single document.

The results are shown in Figure 2. The horizontal axis shows the number of queries, and the vertical axis shows the number of sources contacted on average per query in order to get 10 results. The data series represent a running average, with a snapshot taken every 1000 queries. As the figure shows, the *ProbResults* ranking is very effective, overall contacting 7.6 sources per query on average. Although initially *ProbResults* contacts more sources, once the cache warms up the beacon improves its efficiency. In contrast, Random must contact 32.7 sources per query on average. Even the tightly-coupled TF/IDF-Pre technique performs poorly, contacting 15.2 sources per query on average. This result demonstrates how a static characterization of a site's contents can be inappropriate in a highly dynamic environment.

A more in-depth experimental study is outside the scope of this position paper; see [3].

4 Open problems

There are number of open research problems in developing a loosely-coupled system such as InfoBeacons that we are investigating in our ongoing work.

One key challenge is finding quality data. An advantage offered by the tightly-coupled approach is that high-quality, precise query results can be found. Although the loosely-coupled approach focuses on approximate results, users may want be able to tune the system to achieve a certain quality of results. We need to investigate techniques and system parameters that can be tuned to give users the quality they desire.

Another challenge is in supporting queries that are more complex than simple retrieval via keywords. In the current InfoBeacons prototype, users receive raw data and must do their own data transformations. Ideally, some sort of declarative data processing could be done by the beacon network. This data processing should be efficient, and must deal with a potential lack of schema information or structure within the data. Such processing would likely utilize more approximations than a tightly-coupled system.

A third challenge is to search across multiple types of information. A key motivation for our architecture is that any source can be easily integrated. Using keyword searches is a good start, as techniques have been developed to do keyword searches over XML data [15] and relational databases [13] as well as text. The challenge in our context is how the beacon can pull together information from diverse sources and then present the results to the user in a coherent manner.

A fourth challenge is to automatically discover and integrate new sources. Although complex schema integration is not required, simple wrappers to allow the beacon to communicate with the source are needed. We are hoping to leverage existing work in automatic wrapper generation (e.g, [17]), and potentially expand them to deal with the large number and variety of sources that the InfoBeacons system could search.

Another important challenge is to make the system as robust as possible, both to failures and to untrusted nodes. Redundancy can be used to ensure that the failure of a beacon does not interrupt service. Other investigators have looked at adding resistance to malicious nodes to peer-to-peer systems (e.g., [16]), and we are investigating how to use and extend such techniques for our system.

5 Related work

There has been much work on source discovery in the peer-to-peer community. Much of this work can be clas-

sified as "tightly-coupled", requiring source integration or data export. For example, exported data summaries are used in YouSearch [1] and Galanis et al [9], while Piazza [12] uses mappings between the schemas of sources to process queries. A more loosely-coupled structure is proposed by Kalogeraki et al [14], although their techniques and goals differ from ours.

Source discovery is only one aspect of the services offered by peer-to-peer data management systems such as Piazza and PIER [5]. Of course, we are working to extend InfoBeacons to do more than just source discovery as well. A general model for tightly-coupled systems is presented in [4]. As noted in Section 1, these systems focus on strong semantics and precise results. Various aspects of approximate query processing have been studied in the database field (for example, [2]) but like peer databases, work is just starting in the peer-to-peer community on this topic.

There is also a great deal of work on searching across multiple data sources in other fields, especially information retrieval [10] and databases [6]. Most of these approaches fit in our classification of "tight-coupling", requiring mapping or data export. A more loosely-coupled approach is typified by QProber [11], which probes the source with queries to build up a classification, though it may be difficult to keep this classification current as sources change.

6 Conclusion

We have presented InfoBeacons, a loosely-coupled peer-to-peer data system. A loosely-coupled system offers low barrier to entry for new sources, tolerance to a great deal of dynamism, and the ability to scale to very large numbers of sources. These features are more important for some applications than the strong semantics and precise results provided by data integration, data export or both. In our system, *beacons* pull together multiple sources into a single system. Each beacon is responsible for making sure queries get processed as effectively as possible despite the loose coupling to sources. We have presented some preliminary performance results that show our approach to be effective, allowing queries to find the right sources without too much overhead. We have also discussed some open problems that need to be addressed to effectively build a loosely-coupled peer-to-peer data system.

References

[1] M. Bawa, R. J. Bayardo Jr., S. Rajagopalan, and E. Shekita. Make it fresh, make it quick — searching a network of personal webservers. In *Proc. WWW*, 2003.

[2] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. *VLDB Journal*, 10(2–3):199–223, 2001.

[3] B.F. Cooper. Guiding users to information sources with InfoBeacons. Technical Report. www.cc.gatech.edu/-˜cooperb/pubs/infobeacons.pdf, 2003.

[4] P. Bernstein et al. Data management for peer-to-peer computing: A vision. In *Proc. WebDB*, 2002.

[5] R. Huebsch et al. Querying the Internet with PIER. In *Proc. VLDB*, 2003.

[6] S. Chawathe et al. The tsimmis project: Integration of heterogeneous information sources. In *In Proc. of IPSJ Conference*, October 1994.

[7] S. Gribble et al. What can databases do for peer-to-peer. In *Proc. WebDB Workshop*, 2001.

[8] W. Nejdl et al. Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks. In *Proc. WWW*, 2003.

[9] L. Galanis, Y. Wang, S.R. Jeffrey, and D.J. DeWitt. Locating data sources in large distributed systems. In *Proc. VLDB*, 2003.

[10] L. Gravano, H. Garcia-Molina, and A. Tomasic. GlOSS: Text-source discovery over the internet. *ACM TODS*, 24(2):229–264, June 1999.

[11] L. Gravano, P.G. Ipeirotis, and M. Sahami. QProber: A system for automatic classification of hidden-web databases. *ACM TOIS*, 21(1):1–41, January 2003.

[12] A.Y. Halevy, Z.G. Ives, P. Mork, and I. Tatarinov. Piazza: Data management infrastructure for semantic web applications. In *Proc. WWW*, 2003.

[13] V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-style keyword search over relational databases. In *Proc. VLDB*, 2003.

[14] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proc. CIKM*, 2002.

[15] C. Botev L. Guo, F. Shao and J. Shanmugasundaram. XRANK: Ranked keyword search over XML documents. In *Proc. SIGMOD*, 2003.

[16] E. Sit and R. Morris. Security considerations for peer-to-peer distributed hash tables. In *Proc. IPTPS*, 2002.

[17] J. Wang and F. Lochovsky. Data extraction and label assignment for web databases. In *Proc. WWW*, 2003.