Effect of Joining Decisions on Peer Clusters

Stéphane Airiau
Mathematical & Computer Sciences Department
600 South College avenue
Tulsa, OK 74104, USA
stephane@utulsa.edu

Sandip Sen
Mathematical & Computer Sciences Department
600 South College avenue
Tulsa, OK 74104, USA
sandip@utulsa.edu

Prithviraj Dasgupta
Computer Science Department University of Nebraska Omaha, NE 68182, USA
pdasgupta@mail.unomaha.edu

ABSTRACT

Super-peer networks have been proposed to address the issue of search latency and scalability in traditional peer-to-peer (P2P) networks. In a super-peer network, instead of having a fully distributed systems of peer nodes with similar or comparable capabilities, some nodes that possess considerable computing power and resources are designated as super-peers. Each super-peer acts as a server for multiple client peers under it. This hierarchical structure of a super-peer network improves the performance of a super-peer network over traditional P2P networks by handling most search queries between the few super-peer nodes, thereby reducing overall network traffic and improving the search latency. In this paper, we address the problem of mutual selection by super-peers and client peers. In particular, we evaluate alternative decision functions used by super-peers to allow new client peers to join the cluster of clients under it. We experiment with peers with known resources and demands. By formally representing and reasoning with capability and query distributions, we develop peer-selection functions that either promote concentration or diversification of capabilities within a cluster. We evaluate the effectiveness of these different selection functions for different environments where peer capabilities are aligned or are independent of their queries. We offer insight and analysis on the effects on inter and intra-peer bandwidth consumption which will allow a super-peer to adopt appropriate peer-selection functions given their expectations about the problem domain.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

General Terms

Algorithms, Design

Keywords

Super-peer, Peer-to-peer, Formation

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

1. INTRODUCTION

Current research in peer-to-peer network mainly focuses on the capacity of the network to provide efficient search capabilities, to treat queries in a timely and accurate fashion, and to maximize the bandwidth to transfer the data between peers. Super-peer networks [3, 5, 13] have been proposed to improve the search latency and scalability of traditional P2P systems. Instead of having a fully distributed systems of peer nodes with commensurate computing capabilities and resources, a super-peer network comprises a hierarchical arrangement of nodes with different computational capabilities. A super-peer network is a two-tiered structure: one level consists of super-peer nodes that have considerable computing capability and resources with each super-peer managing operations such as searching and query forwarding for multiple client peers under it. The benefits of this architecture, compared to a fully decentralized approach, is an improvement in search latency and scalability as most queries are routed between few super-peers or between client peers under one super-peer. In particular, every peer no longer needs to handle the network traffic associated with every search query. A super-peer, however, can become a bottleneck for its clients: it handles incoming and outgoing queries on their behalf, which requires maintenance of an accurate description of all clients.

Under the assumption that all peers are contributing to the community, [13] gives advice for building efficient super-peer networks. An open problem is the choice of a super-peer when a new peer joins the network. A super-peer decides to accept or decline a join request from a new peer. In [5], the design of the network corresponds to ontologies. The idea is to cluster peers falling in the same category so as to make search more efficient in a semantic peer-to-peer network. As a result super-peers will have similar clients. For example, it is possible to create a community of super-peers hosting computer science related information. Each super-peer could correspond to an area of computer science, say language, operating system, artificial intelligence (AI), etc. and the peers of the super-peer corresponding to AI could host information about agents, planning, Bayesian inference, etc. Depending on the need or the knowledge of the new peer, a similarity measure or the use of an ontology can dictate which super-peer to become client of. [5] prescribes a structured network organization such a hypercube that determines the topology of client peers under a super-peer. However, the problem of network organization becomes challenging and difficult to design a-priori if we consider an unstructured P2P network. Hence, we are interested in studying decision functions that super-peers and peers can use to mutually select each other and the effects on such decision functions on peer cluster composition.

Consider the case where a super-peer can answer a query from one of its clients by using its other clients. There may not be any

need to send a query to other super-peers. Hence, in addition not to have to deal and coordinate search with other super-peers, answering a query by another peer in the cluster decreases the traffic between super-peers. This phenomenon can be highlighted by considering super-peer networks where each cluster is quasi-self-sufficient: each query being almost always answered by peers from the same cluster. The research question is how to form clusters of clients to significantly decrease the traffic between clusters. Should a super-peer seek heterogeneity of interest and capabilities of its peers, or should it attempt to build a community of peers having similar interests and capabilities.

In this paper, we want to dynamically build the network of super-peers from a fully distributed network. We want to ensure that peers are contributing to the community. To this end, we use a reciprocity based mechanism to promote collaboration between the super-peers. We use a cost function to capture the preference of intra-cluster data transfer over inter-cluster data transfer. We consider that peers are defined by their interests, which are used to generate queries, and their knowledge or competencies, that are used to answer queries. In the real world, we believe that knowledge and competencies are often aligned: interest in a subject leads to an increase knowledge in that topic. In such situations, peers with similar interests should be able to sustain self-sufficient communities, needing the help of other clusters very rarely. On the other hand, if interest and knowledge are not aligned, it might be preferable to create heterogeneous groups where peers have complementary knowledge and interests. We want to study how the join function effects the cluster compositions and the total cost of answering queries. In particular, we will consider environments where agents that have similar interest and knowledge, and environment where interest and knowledge are not correlated.

2. RELATED WORK

P2P systems have become an area of active research and development since the popularity of online resource sharing services such as Freenet [1], Gnutella [2], Napster [4] and SETI@home [9]. Resource management in P2P networks has been an important and challenging issue for researchers. The most common techniques for P2P resource management include structured P2P networks that employ distributed hash tables (DHT). In DHT-s, resources are strategically placed on nodes to improve resource availability and enable rapid lookup [10, 7, 14]. In DHT-s each node and resource is associated with a key computed from a hash function. A mathematical function [6, 11] is then used to strategically place resources in different nodes to preserve the network topology and balance the network load throughout the system. However, DHT-s require additional overhead in the form of updates to local hash tables within a node when nodes and resources join or leave the network, and, forwarding the updates to neighbor nodes.

Yet another mechanism for P2P resource management is super-peer networks [12, 13] that uses a tiered network structure within an unstructured P2P network. In a super-peer network, some nodes acting as super-peers or managers that supervise and coordinate the operation of several peers or clients under them. Super-peers maintain meta-information about peers supervised by them. A super-peer can interact with the peers its supervises and with other super-peers to route search queries and implement load balancing algorithms. In super-peer networks, a peer wishing to search for a resource, contacts the super-peer supervising it with the search request. The super-peer first searches for the resource within other peers supervised by itself. If the resource is not found within its supervised peers, the super-peer directs the search query to other super-peers. Each super-peer that receives a search request from another super-peer searches for the resources within its supervised peers and responds to the super-peer that initiated the query at the super-peer level if the resource is found. The super-peer that initiated the super-peer level query then forwards the resource-found response to the peer under it that originated the query. In [5], the super-peer framework has been implemented within a structured P2P network. DHT-based algorithms are used to determine the network topology and resource placement within the network. Database schema based techniques are also used to organize the content on different peers. In [3], a protocol for dynamically updating the topology of a super-peer network is described. Super-peers exchange meta-information about peers with each other to reconfigure the network and achieve load balancing. In contrast to these research, our paper describes mechanisms that can be used by peers and super-peers for mutual selection within an unstructured P2P network.

3. SUPER-PEER NETWORK MODEL

In this section, we describe our model of the peer-to-peer system and the query protocol. We assume a fully connected network of super-peers, which allows a super-peer to be contacted directly (a query can be targeted to a precise set of peers). Each super-peer manages a cluster of client peers.

Each peer is defined by its interests and its competence in providing information to other peers. The interests are used to generate queries, and the competencies are used to answer queries. We will investigate two kind of environment: one where competencies and interest are the same, the other where they are not correlated. Each resource is indexed by a set of interests or keywords and peers can query for a resource based on its associated interest set. Interests are represented as words (character strings). We have used a hash function to map the different interests to a natural number in the range $\mathcal{S} = [0..dim]$. Every peer uses the same hash function to ensure uniformity in the interpretation of interests across the network. This allows us to model the competence and interest on a 1-dimensional line. The interests of a peer are represented by a probability distribution function $\mathcal{I}$ over $\mathcal{S}$. Peaks on $\mathcal{I}$ represents the main interests of a peer. The competence of the peer is represented by a function $\mathcal{C}$ which, given a point q in $\mathcal{S}$, output the probability $\mathcal{S}(q)$ of answering a query for that input. We have chosen this approach to have a static description of the peers' interests and capabilities, to facilitate analysis and experimentation, which would not be possible if peer capability was represented by a dynamic database of resources.

A super-peer has the responsibility to handle queries from its clients and other super-peers. It maintains a balance of help over the past interactions with the other super-peers, and with its own peers. Each time a peer answers a query from another peer, it incurs a cost proportional to the amount of data transferred. We assume that transfer of data within a cluster is cheaper than transferring data between cluster. In order to promote contribution of other super-peers, the balance of cost is used in a probability-based mechanism to make the decision [8] of answering or not answering queries from other super-peers. A super-peer wants to minimize the cost incurred by its own clients by promoting intra-cluster communication. If a super-peer or one of its peers does not collaborate sufficiently, the super-peer can decide not to block their future queries until the balance improves.

We are interested in joining decision mechanism that are function of the interest and competence of the peer and cluster, which requires a peer to reveal an estimate of its interest and competence. Because of the trust mechanism used by the super-peer, the peer does not have incentives to lie about its competence or interest.

However, we assume that peers can generate their competence and interest vector, which may not be a simple task. In addition, we assume that the super-peer is also a peer, hence it can ask queries and answer queries.

3.1 Cost and Reciprocity Framework

The cost metric is a function of the volume of data transferred and whether the communication is intra or inter clusters. We use two rates: c_l for intra-cluster communication and c_w for inter-cluster communication. The difference in cost models a search cost incurred by the super-peer when it needs to deal with the other super-peers.

Each super-peer records the past interactions of help with the other super-peers. For a super-peer i, the super-peers maintains:

- credit $c(i)$, i.e., cost of the help received from i.
- debit $d(i)$, i.e., cost of the help provided to i.
- balance $b(i) = d(i) - c(i)$

To determine whether to answer a query q from super-peer i, the peer gets the cost c_q of answering q using its knowledge about the capabilities of its client peers. The probability of answering q is

$$P(i,q) = \frac{1}{1 + e^{\frac{c_q - b(i) - c_0}{\tau}}},$$

where c_0 is the initial inclination to help and τ controls the shape of the probability function. The super-peer will sample this probability to decide whether to help or not the other peer. If c_0 is large, the super peer is more inclined to help other super-peers. In Figure 1(a) and 1(b), we present the probability to answer a query with respect to its cost. In Figure 1(a), we study the effect of c_0 on the probability function. The higher the cost, the less likely it is to provide an answer. In addition, the higher the value of c_0, the more likely it is to answer a query. In Figure 1(b) we vary the value of τ for a fixed value of c_0: large value of τ make the probability function to answer a query quasi-linear in the cost, when small values of τ gives a logistic shape to the function: high probability of answering a query for low cost, low probability for high cost.

3.2 Query and Protocol

The following describes the different aspects of the query generation and response process:

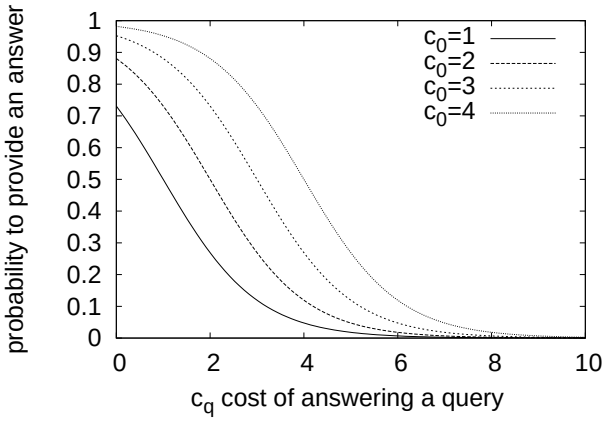
Query generation: A query corresponds to one point q in $\mathcal{S}$, the interest space. A peer generates a query by sampling its probability distribution $\mathcal{I}$. The message containing the query also contains the address of the requesting peer and its super-peer.

Query-answering capability: The client has an answer to q with a probability $\mathcal{C}(q)$, and do not have an answer with probability $1 - \mathcal{C}(q)$.

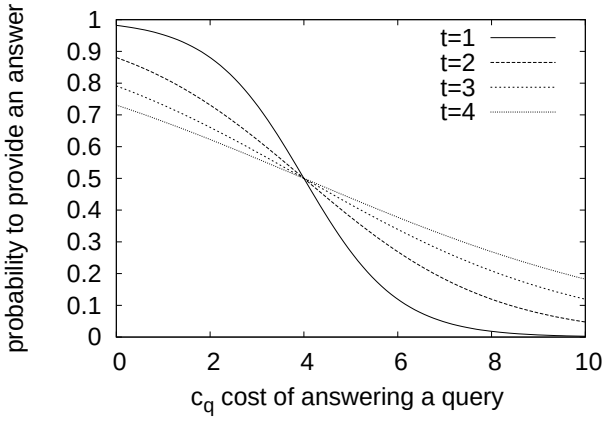
Replying to a query q: A super-peer answers a query by sending a message, r_a containing its address, the address of one of its clients, a that will answer q and the associated cost. The clients are then responsible for opening a connection between themselves and transfer the data.

Protocol: The protocol for the interaction of the source of a query and the responder is as follows:

1. an initiator peer P_q generates a query q and sends it to its super-peer $SP(P_q)$.



(a) Variation of c_0.



(b) Variation of τ.

Figure 1: Influence of c_0 and τ on the probability to help.

2. $SP(P_q)$ checks on it database whether one of its client can answer q. We consider that a super-peer has accurate knowledge of the capabilities of its client peers. In our model, this is implemented by $SP(P_q)$ sampling the probability $\mathcal{C}_i(q)$ to determine whether the peer i can answer the query q or not. If one of the clients under $SP(P_q)$ can answer the query, the query is forwarded to it. If more than one client can answer q, the super-peer chooses the one with the lowest balance (upload minus download volume). This allows the super-peer to distribute the load of query answering between its client peers and helps maintain the satisfaction level of peers.

3. If the clients of $SP(P_q)$ can not answer the query, $SP(P_q)$ broadcasts the query to other super-peers. When another super-peer SP_a receives a query from $SP(P_q)$ it performs the following steps:

   (a) SP_a determine whether one of its peer can answer the query. Again, this is implemented by sampling the probability $\mathcal{C}_i(q)$ for each of its peer i. If some peers can answer the query, SP_a picks the peer P_a with the low-

est balance as described above. If no client can answer the query, then SP_a cannot answer the query.

(b) If some client can respond to the query, SP_a uses the reciprocity framework (Section 3.1) to decide whether or not to provide the answer.

(c) If SP_a decides not to answer, an empty message is sent.

(d) If SP_a decides to help, it forwards the query to P_a, and sends a reply r_a to $SP(P_q)$. The reply message contains the address of the peer that will answer the query and the associated cost. At this time, SP_q may or may not use this answer, hence SP_a does not update its balance of yet at this time, it will wait a notification that its peer has provided an answer (see step 4)

$SP(P_q)$ gets all the reply messages r_a from all the super-peers. It chooses to use the answer provided by the super-peer SP_a with the smallest balance, i.e. it picks the super-peers that owes the most. $SP(P_q)$ updates its balance with SP_a and forwards the reply message to P_q.

4. when P_q receives the reply, it directly contacts the answering peers P_a and transfers the data. At the end of the transfer, P_a notifies its super-peer about the transaction so that SP_a can update its balance with $SP(P_q)$.

3.3 Mechanisms of cluster formation

We assume the existence of a network of super-peers. The peer nodes enter the system one at a time and are assigned to one of the super-peers by the process described below. Once assigned, a peer-node does not change its super-peer.

3.3.1 Random peer assignment

As a baseline scheme for comparison we consider a random assignment of new peers to any of the super-peers.

3.3.2 Negotiated peer assignment

We consider three negotiated mechanisms. In each of these methods new peers negotiate with existing super-peers and final assignment is made by mutual selection.

When a peer seeks to join a cluster, it sends its capability and interest information to all super-peers. The super-peers estimate the "usefulness" of having this new peer in their cluster and reply with the estimate. The peer greedily chooses to join the super-peer that provided the best estimate. To measure this "usefulness", we consider three metrics:

Competence alignment: By using this metric, a super-peer seeks peers that have similar competence, which improves the probability to answer queries on particular topics. The super-peer builds the aggregate competence vector of its cluster of peer nodes. The aggregate competence of a cluster for a given domain is the probability that at least one peer answers the query. Consider that a super-peer has n peers, and let $c_{i,j}$ denote the probability of the i^{th} peer to answer a query for the j^{th} domain in the competence space $\mathcal{S}$. The probability for at least one peer to answer the query is

$$c_j = 1 - \Pi_{i=1}^{n}(1 - c_{i,j}), j \in [1..dim].$$

The metric returned is the Euclidean distance between the aggregate competence of the cluster and the competence of the peer. The smaller the distance, the better the offer made to the peer.

Competence diversity: With this metric, a super-peer seeks to form a heterogeneous group able to answer the most diverse range of queries. For example, a peer who does not bring any new expertise to a cluster is of little interest. The super-peer computes the probability of answering a query about each domain in $\mathcal{S}$ in the absence or presence of the requesting peer. The difference is the measure of the impact of the peer on the group. The "usefulness" metric sent to the requesting peer is:

$$\Delta = \frac{1}{dim}\sum_{j=1}^{dim}\left(c_j^+ - c_j\right),$$

where dim is the dimension of $\mathcal{S}$, c_j (resp c_j^+) is the aggregate probability of the cluster (resp the cluster and the peer) to answer a query about the j^{th} domain in $\mathcal{S}$ (we assume a uniform distribution of queries over $\mathcal{S}$). A super-peer seeks peers with higher Δ value. The peer chooses the super-peer with the largest Δ.

Competence and interest complementarity: By using this metric, a super-peer seeks peers that can answer queries from the current cluster and vice versa. When this metric is used, peers and super-peers use a two-step interaction protocol: the super-peer will first evaluate the ability of the peer to answer queries from the cluster's member. If the likelihood is acceptable, the peer will send the likelihood for cluster's member to answer a query from the new peer. We now provide more details. First the peer sends its competence vector so that the super-peer can estimate the potential of this peer to answer a query issued by one current cluster's member. The super-peer invites a peer only if it can bring sufficient new expertise to the cluster. If the probability of answering queries from its cluster exceeds a fixed threshold ϵ_u, the super-peer responds favorably to the peer. The peer replies to interested super-peers and sends them its interest vector. The super-peers return the probability of the cluster to answer a query issued by the new peer. This one chooses the super-peer with the highest likelihood of answering its query. Let q denote the interest probability distribution and let c denote a competence vector. The probability that a peer or cluster with a competence c can answer a query from a peer or cluster of interest q is $\sum_{i=1}^{d} q_i c_i$. In a nutshell, the super-peer accepts beneficial peers, and the peer joins the most promising super-peer, i.e. one that is most likely to able to answer its queries.

4. SIMULATION

To evaluate the relative effectiveness of different cluster formation mechanisms, we experimented with different environmental configurations. We generated the interest and the competence of the peers assuming the existence of a preponderant topic of competence and a preponderant topic of interest, which need not be identical. When they are identical (competence and interest are aligned), we have peers knowledgeable about a particular topic and wanting to know more about that topic. When the preponderant topics are different (competence and interest are not aligned), we have peers that have knowledge in a domain, want to learn about a different domain. In Figure 2, we present a typical competence vector of a peer. The algorithm for generating a competence vector for a peer is presented in Algorithm 1. The interest vector is generated in a similar way, it is normalized to meet the requirements of a probability distribution. We have experimented with two types of peers: when interest and competence are aligned and when they are not correlated. In all experiments we ensure that the preponderant competence and interest are independently drawn for a uniform distribution.

4.1 Settings

The results presented in this section are averaged over 20 different initial assignments of the peers to the clusters (the order of introduction of the peers is different). For each initial assignment, we ran the simulation 5 times, varying the order of the peers asking

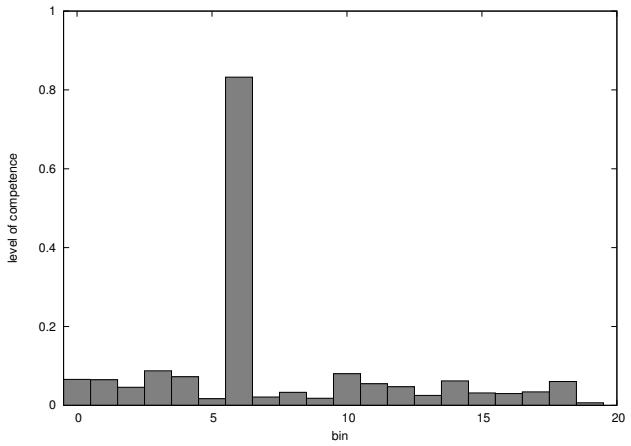


Figure 2: Example of the competence vector of one peer, $d = 20$.

Algorithm 1 Form competence c for peer p.

p_{high}, p_{low} and v are given
for $i = 1$ to d **do**
  $\epsilon \leftarrow$ normal(0,1)
  if main interest of p is i **then**
    $c_i \leftarrow p_{high} + \epsilon v$
  else
    $c_i \leftarrow p_{low} + \epsilon v$

queries and the peers that changes interest, which yields different final configuration based on the change of clusters. Each simulation consists of generating 10,000 queries. For each query, a peer is chosen randomly, and then its interest distribution is sampled to generate the query type. There is a fixed number of super-peers, one for each domain, and the number of peers varies between 20 and 300.

4.2 Influence of the cost difference between intra and inter communication

First, we study the influence of the cost difference between intra cluster communication and inter cluster communication on the environment where interest and competence are aligned. For these experiments, the reciprocity framework is initialized with $\tau = 1$ and $c_0 = c_{inter}$, i.e. initially, the super-peer have a 50% chance to to answer one query issued from another super-peer. In Figure 3, we present results of experiments with a system of 500 peers, 50 super-peers and the number of domain is set to $d = 50$. The cost for communication within the group is fixed to 1.0 per answer.

Since for one peer, the preponderant competence domain and interest domain are the same, the optimal assignment occurs when peers with the same preponderant domain are grouped in the same cluster. They are likely to answer most queries of other members of the cluster, thus minimizing total cost of answering queries. Requesting the help of peers from other clusters is needed when a query is outside the preponderant domain, which is much less frequent. Under these conditions, the join function promoting complementarity between peers and the join method based on the alignment of the competence produce best performance. We noticed that as the size of a cluster increases the cost increases slightly. Even if all the cluster's peers have the same main expertise, the probability to answer other types of query also increases, which may "mislead" the assignment of some peers. These peers cause the performance to drop (i.e. the cost to increase) since the query is less likely to be answered by the cluster's member. Surprisingly, promoting the diversity of competence in the cluster performs better than random assignment. The use of a diversity based mechanism should be costly since it is unlikely to have two peers with the same interest/competence coexisting in the same cluster. The relative better performance compared to random is due to the existence of few larger clusters that are more capable of being self-sufficient.

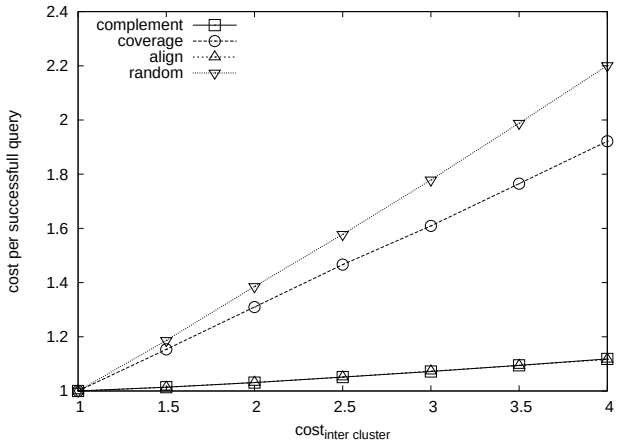


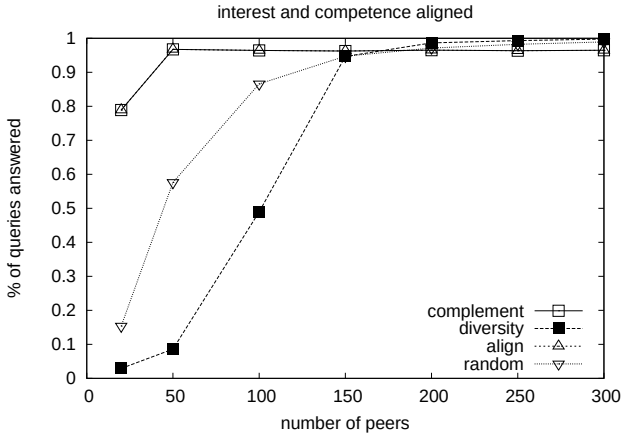
Figure 3: Influence of the ratio cost between inter and intra cluster communication on average cost of answering a query under different cluster formation mechanism.

4.3 Influence of the number of peers

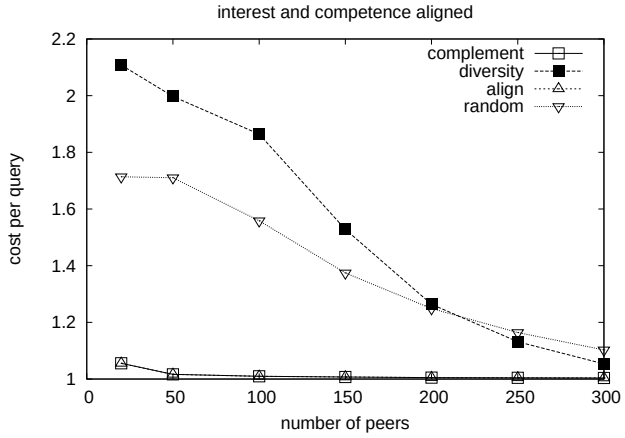
For the rest of the study, we fixed the ratio of the cost of inter and intra-cluster communication to 2.5, the cost to communicate one answer within a cluster is set to 1.0. In the next set of experiments, we fixed the dimension of the interest space and the number of super-peers to 10. We study the influence of the number of peers in the system in environment where either competence and interest are aligned or not correlated. Our first metric is the average cost of a query. If it is 1.0, all the queries are answered within the cluster. The second metric is the percentage of queries answered (see Section 3.2). A query may not be answered for multiple reasons. First, when a peer p answers a query about domain $i \in \mathcal{S}$, the distribution $\mathcal{C}(i)$ is sampled. Even if i is the domain of expertise of p, it may not get the answer and no one else may be able to answer. In addition, a peer might be capable of answering the query, but the super-peer may block the query because the requesting peer has a low balance.

The results when the competence and interest of a peer are aligned are presented in Figure 4. The complement-based and the alignment-based join functions produce clusters of peers with similar expertise and are performing the best in this environment (the corresponding curves are overlapping). As peers generate queries in their area of interest, other peers in the same group can answer these queries with a high probability and hence they do not require help from other super-peers, keeping the cost low. For a small number of peers in the system, the join function based on diversity is performing worse than random assignment. This is because all the peers in the cluster have different competence/interest, and they have a small probability to answer queries from the other peers of the cluster. When the number of peers is high, each group contains sufficient number of experts of each type in each cluster to answer

queries issued by any peer and hence the performance of clusters generated by all mechanisms become equivalent. This is shown both by an increase in the success rate of answering queries and a corresponding decrease in the cost per query.



(a) Success Rate

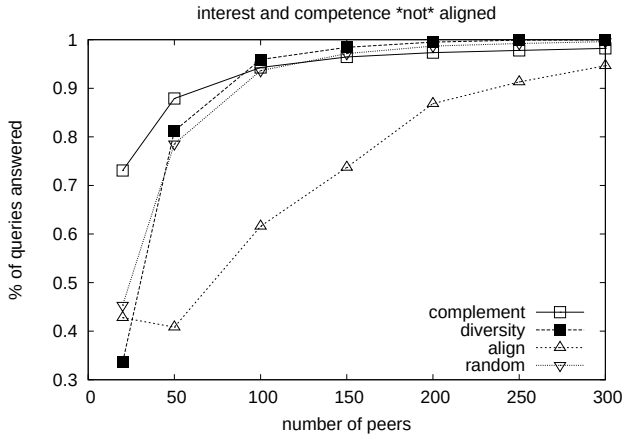


(b) Cost

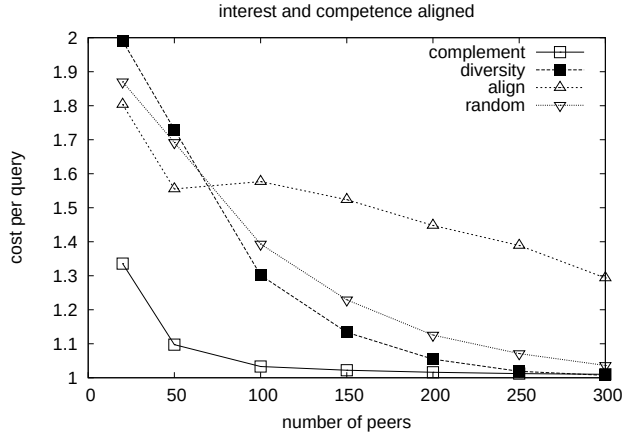
Figure 4: Influence of the number of peers: case where interest and competence are aligned.

When the preponderant competence and interest of a peer are not correlated, the join function based on complementary is still performing much better with a small number of peers in the system (see Figure 5). The first significant difference with the previous scenario is that the alignment-based join method is performing much poorer: as peers with the same main expertise do not necessarily have the same main interest, a cluster made of peers with the same main expertise will require the help from many other clusters. The second main difference is that the diversity-based join function takes less number of peers in the system to perform better. The promotion of diversity in the cluster formation increases the probability of answering any kind of query by another member of the same cluster, which is the key element here.

From the above scenari, the main conclusion to be drawn is that the join method based on complementarity is dominating the other cluster formation methods in a wide variety of environmental conditions. The inherent reciprocal benefit consideration underlying this cluster formation scheme enables such clusters to be largely "self-sufficient" in most environments, thus increasing query answering rate and reducing the cost for answering queries.



(a) Success Rate



(b) Cost

Figure 5: Influence in the number of peers, case where interest and competence are not aligned.

5. CONCLUSIONS

We have investigated the effects of different join-decision-functions on the performance of super-peer networks. Super-peers are responsible to find other peers which can provide an answer to a query, either by using peers from its pool of clients, or by requesting help to other super-peers. Super-peers use a reciprocity mechanism to ensure that there are no free-riders in the system. Each super-peer also ensures that all its client peers are contributing by enforcing load balancing within the cluster.

We use a probability function to model the competence and a probability distribution to model the interest of a peer. Under these conditions, we found out that forming peers based on complementarity between a new peer and a cluster is beneficial although this

join method, unlike the other, is a two-steps process. Looking for diversity to form a cluster can also be beneficial when there is no correlation between the competence and the interest of a peer. However, when interest and query is aligned, alignment of competence in a cluster is preferable to competence diversity. As the number of peers in each cluster increases, however, the performance difference between the different cluster formation mechanisms monotonically decreases.

We plan to study the effect of the relative ratio of the number of super-peers to the number of capability and interest types. In this paper, agents do not change clusters; we plan to investigate dynamic cluster dissolution and reorganization schemes. In particular, we will need to adjust the reciprocity framework. Another limitation in the current work is the static interest and capability vectors for peers. We believe that in a number of domains both capabilities and interests of peers can vary over time. That dynamism presents a significant challenge to developing adaptive schemes that will continue to maintain the performance of peer clusters. We will investigate predictive cluster selection schemes to address this critical issue.

Acknowledgment: US National Science Foundation award IIS-0209208 partially supported this work.

6. REFERENCES

[1] Freenet. URL http://www.freeproject.org.

[2] Gnutella. URL http://www.gnutella.com.

[3] A. Montresor. A robust protocol for building superpeer overlay topologies. In *Proceedings of the 4th International Conference on Peer-to-Peer Computing*, August 2004.

[4] Napster. URL http://www.napster.com.

[5] W. Nejdl, M. Wolpers, W. Siberski, C. Schmidt, M. Schlosser, I. Brunkhorst, and A. Löser. Super-peer-based routing and clustering strategies for rdf-based peer-to-peer networks. In *Proceedings of WWW2003*, 2003.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*, pages 161–172, 2001.

[7] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings ofthe IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[8] S. Sen. Believing others: Pros and cons. *Artificial Intelligence*, 142(2):179–203, December 2002.

[9] Seti. URL http:// setiathome.ssl.berkeley.edu.

[10] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM Conference*, 2001.

[11] C. Tang, Z. Xu, and M. Mahalingam. Peersearch: Efficient information retrieval in peer-peer networks. Technical Report HPL-2002-198, Hewlett-Packard Labs, 2002.

[12] B. Yang and H. Garcia-Molina. Improving search inpeer-to-peer networks. In *Proceedings of the 22ndInternational Conference on Distributed Computing Systems (ICDCS'02)*, pages 5–14, July 2002.

[13] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the Nineteenth International Conference on Data Engineering*, pages 49–62, 2003.

[14] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault resilient wide-area location and routing. Technical Report UCB CSD-01-1141, University of California, Berkeley, 2001.