A Scalable Peer-to-Peer System for Music Content and Information Retrieval

George Tzanetakis
Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15218
gtzan@cs.cmu.edu

Jun Gao
Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15218
jungao@cs.cmu.edu

Peter Steenkiste
Computer Science and Electrical Engineering
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15218
prs@cs.cmu.edu

Abstract

Currently a large percentage of Internet traf£c consists of music £les, typically stored in MP3 compressed audio format, shared and exchanged over Peer-to-Peer (P2P) networks. Searching for music is performed by specifying keywords and naive string matching techniques. In the past years the emerging research area of Music Information Retrieval (MIR) has produced a variety of new ways of looking at the problem of music search. Such MIR techniques can signi£cantly enhance the ways user search for music over P2P networks. In order for that to happen there are two main challenges that need to be addressed: 1) scalability to large collections and number of peers, 2) richer set of search semantics that can support MIR especially when retrieval is content-based. In this paper, we describe a scalable P2P system that uses Rendezvous Points (RPs) for music metadata registration and query resolution, that supports attribute-value search semantics as well as content-based retrieval. The performance of the system has been evaluated in large scale usage scenarios using "real" automatically calculated musical content descriptors.

1 Introduction

One could argue that both the ideas of Music Information Retrieval (MIR) and Peer-to-Peer networks (P2P) essentially started with Napster (http://www.napster.com). Although crude both in terms of search capabilities and in terms of P2P performance, Napster for the £rst time provided an example of sharing large amounts of musical data over large adhoc networks. Despite this early connection there has not been much progress in combining these two areas. Although better P2P paradigms have been proposed, searching for music is currently still performed using traditional keyword-based text search. While a variety of novel ways of searching and retrieving music, especially in audio format, have been proposed, they haven't found their way into P2P networks and remain largely academic exercises.

There are many advantages to P2P networks such as distributed computing and storage power, low bandwidth, fault-tolerance and reliability. Although because of copyright restrictions major recording labels have been reluctant to follow this paradigm the emergence of audio £ngerprinting technology Haitsma and Kalker (2002) is likely going to change this attitude. One of the greatest potential bene£ts of P2P networks is the ability to harness the collaborative efforts of users to provide semantic, subjective and community-based tags to describe musical content.

Centralized P2P such as Napster, are not robust and may be vulnerable to Denial-of-Service attacks, since the central server forms the system's single point of failure. Such a system does not scale well as registration and query load increases. Distributed P2P systems, such as Gnutella (http://www.gnutella.org) and KaZaA (http://www.kazaa.com), are more robust, but since peers do not explicitly register their shared £les, a query may have to be broadcast throughout the network to get resolved. The potentially large number of messages involved limits the system's scalability and performance. Distributed Hash Table (DHT) based systems, such as the ones described in Stoica et al. (2001); Rowstron and Druschel (2001), achieve good scalability by deploying a structured overlay P2P network that supports ef£cient content location. However, the basic set of applications built on top of DHT, only supports exact £le name look up and does not allow the rich search semantics desired for MIR.

In this paper, we describe a robust, scalable P2P system that provides ¤exible search semantics based on attribute-value (AV) pairs and supports automatic extraction of musical features and content-based similarity retrieval. The system is shown to perform well under realistic loads consisting of features automatically extracted from a large database of audio recordings. The main contributions of this work are: a general content discovery mechanism that supports exact and similarity search based on AV-pairs and its evaluation using a speci£c set of audio features computed on actual audio recordings. We believe that the proposed system provides the necessary ¤exibility and performance for effective use of Music Information Retrieval for searching in Peer-to-Peer networks.

2 Related Work

The main focus of this work is the retrieval of music in audio format over P2P networks. There is a lot of recent exciting work in MIR that is relevant to the design of our system and we review some representative publications. Although this paper mainly describes similarity retrieval, the underlying framework of features and distance calculations forms the basis of a variety of audio analysis algorithms such as: musical genre classi£cation (Tzanetakis and Cook (2002); Aucouturier and Pachet (2003)), beat detection and analysis (Foote and Mathew (2002)), similarity retrieval (Logan and Salomon (2001); Aucouturier and Pachet (2002) Yang (2002)), audio £ngerprinting (Haitsma and Kalker (2002)) and clustering and visualization (Rauber et al. (2002)). In addition to features computed from automatic analysis of audio content, features computed based on text analysis of critics reviews as well as P2P usage patterns have been shown to be effective for classi£cation in Whitman and Smaragdis (2002). These articles are representative of each category. A general overview of the current status in MIR and an extensive bibliography can be found in Futrelle and Downie (2002). Another good overview of the current state of the art and challenges in MIR is Pachet (2003).

DHT-based systems such as Stoica et al. (2001); Rowstron and Druschel (2001) solve some of the scalability problems of the more well known broadcast-based systems such as Gnutella (`http://www.gnutella.org`) and KaZaA (`http://www.kazaa.com`). The Content Discovery System (CDS) proposed in this paper is built on top of such a DHT-based system. The idea of using MIR over a P2P was proposed in Wang et al. (2002). However the proposed system architectures suffer from scalability problems and only the retrieval of symbolic data is examined. The potential of integrating MIR and the evolving semantic web was explored in Baumann and Kluter (2002). More recently content-based retrieval over a P2P network using the JXTA programming framework was presented in Baumann (2003). The proposed system mainly is concerned with integrating feature representations into the P2P system rather than using the structure of the network to support exact and similarity search. An initial description of the system presented in this paper can be found in Gao et al. (2003). The main differences of this paper from previous work is the scalable and ef£cient support for both exact and similarity searching based on Rendezvous Points and the performance evaluation of using audio features computed from actual audio recordings rather than simulated synthetic data.

3 System Overview

Following recent work in P2P networks, our system decouples the process of locating content from the process of downloading it. Each node in the network not only stores music £les for sharing but also information about the location of other music £les in the network. Therefore when the user submits a search to the P2P system, the system returns a set of peers from which the music £les that satisfy the search criteria can then be downloaded. Each £le in the system is described by a Music File Description (MFD) which essentially is a set of attribute-value (AV) pairs. For example a possible MFD might be the following: $\{$*artist = U2, album = Rattle and Hum, song = Desire, ..., specCentroid= 0.65, mfcc2 = 0.85, ...*$\}$. Notice that the description contains both attributes that can be manually speci£ed by the user as well as automatically extracted features for describing musical content (underlined). The Music Feature Extraction Engine (MFEE) is the component that calculates these features from the audio £les. These automatically extracted features in addition to being used for content-based similarity search can also be used for various other types of audio analysis such as musical genre classi£cation.

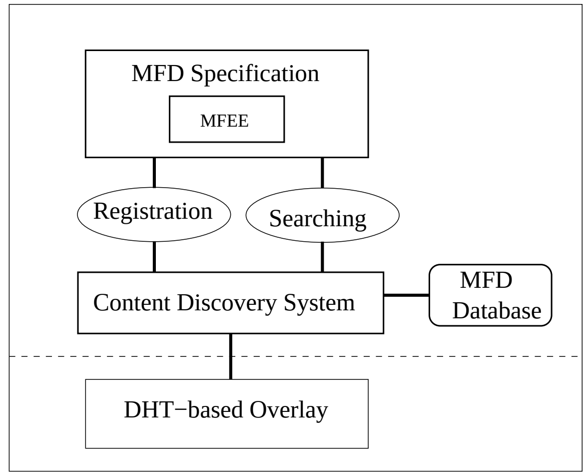

Figure 1: Software architecture on a peer node.

There are two operations that are supported by the system and both take MFDs as arguments. In registration, a new music £le is made available for sharing and its associated MFD is registered to the P2P system. During searching, the user query is converted into an appropriate MFD which is then used to locate the nodes that contain £les that match the search criteria. Once the nodes are located they are contacted directly to start the actual downloading for the £le. The main concern in the design of our system is the ef£cient content discovery rather than the actual downloading mechanism.

Figure 1 shows the software architecture on each peer node. The MFD of either registration or search query is passed to the Content Discovery System (CDS), which runs on top of a Distributed Hash Table (DHT) based P2P system, such as Chord Stoica et al. (2001). In a DHT, each peer is responsible for a region, represented with a node ID, in a contiguous m-bit virtual address space. A data item such as a £le name, is associated with some value in this address space, e.g., by applying a uniform hash function to the data item, and stored on the peer whose region covers the value. Correspondingly, by applying the same computation to the data item, a peer can locate it from the same peer who stores it. We present the algorithm used by the CDS to distribute MFDs to peers in Section 5. The underlying mechanism of DHT ensures routing and message forwarding ef£ciency in such a system: in Chord, a peer only needs to keep information about $O(\log N_c)$ neighboring peers, and the number of overlay hops between two peers is $O(\log N_c)$, where N_c is the total number of peers in the system. Each peer maintains a local MFD database to store the MFDs it is assigned by the CDS. Upon the arrival of a query, each peer examines its local MFD database and returns the set of MFDs that match the query to the query initiator. Subsequently, the query initiator can download the actual music £le from the peer that owns the music.

4 Music Feature Extraction

The MFEE component takes as input an audio £le in either PCM (pulse code modulated) or compressed format, such as MP3, the MPEG audio compression standard, and outputs a feature vector, also known as the content-based vector, of AV-pairs that characterizes the particular musical content of the £le. In our system, we use the feature set proposed in Tzanetakis and Cook (2002) for the purpose of musical genre classi£cation. This feature vector captures aspects of instrumentation and sound texture (what instruments are playing and their density distribution over time), rhythm (fast-slow, strong-weak), and pitch content (harmony) and has been shown to be an effective representation for the purposes of classi£cation and retrieval of music. More speci£cally, features based on the Short Time Fourier Transform as well as Mel-Frequency Cepstral Coef£cients are used to represent sound texture, and features based on Beat and Pitch Histograms are used to represent rhythm and pitch content. More speci£cally, the means and variances of the Spectral Centroid, Rolloff, Flux and ZeroCrossings and the £rst 5 Mel Frequery Cepstral Coef£cients (MFCC) over a 1 second texture window using 20 millisecond window are calculated for representing Spectral Texture. For the Beat Histogram calculation a Discrete Wavelet Transform £lterbank is applied and autocorrelation-based envelope periodicity detection is performed. For the Pitch Histogram calculation the multiple pitch detection algorithm described in Tolonen and Karjalainen (2000) is used. The different types of information represented by the feature vector combined with the query ¤exibility of the system supports a rich variety of possible query speci£cations. For example, a user can search only on the basis of rhythmic content while ignoring other aspects.

We use standard linear quantization and normalization to transform the dynamic ranges of the continuous features into discrete values necessary for searching based on AV-pairs. Linear quantization was chosen so that the statistics of the distribution of the features do not change. In our system, each feature is quantized to 100x1 discrete values. Experiments comparing automatic classi£cation of the original continuous features and the quantized features showed no signi£cant differences in the results. Using the features and dataset (10 genres) described in Tzanetakis and Cook (2002) and a Gaussian classi£er we obtain 57.5% accuracy using the unquantized features and 58% accuracy using the quantized version. The results of the MFEE component together with manually annotated metadata such as artist and album name are combined to form a Music File Description (MFD), which is a collection of AV-pairs. As an example, $MFD_1 : \{a_1 = v_1, ..., a_n = v_n\}$ consists of n AV-pairs, where $a_i, i = 1..n$ can be either a manually annotated attribute or a content-based feature attribute. For specifying queries, MFDs are similarly formed to represent the search criteria. In particular, the MFEE is used to generate a query MFD when the user provides a sample piece of music. Any subset of the MFD can be used for query speci£cation and using named AV-pairs in MFDs allows more possibilities than traditional keyword-based search. Some examples of possible queries of increasing complexity are the following:

- Search for $\{$*artist = U2*$\}$
- Search for $\{$*artist = U2, year = 1985, tempo = 80 beats-per-minute (BPM) - 100 BPM* $\}$
- Search for $\{$ *10 most similar to x.mp3*$\}$ (content-based similarity search)
- Search for $\{$ *10 most similar to x.mp3, artist = U2*$\}$

These are just a symbolic representations of queries. In an actual implementation a variety of user interfaces for query speci£cation would be provided by the system (for examples see Tzanetakis et al. (2002)). For queries with content-based parts such as the last two ones, the audio £le, *x.mp3*, is £rst converted using the MFEE to numerical features that describe musical content which are subsequently used for similarity search. This mechanism allows any audio £le to be used in the system even if it doesn't have any metadata information associated with it (for example a £le recorded off a radio broadcast).

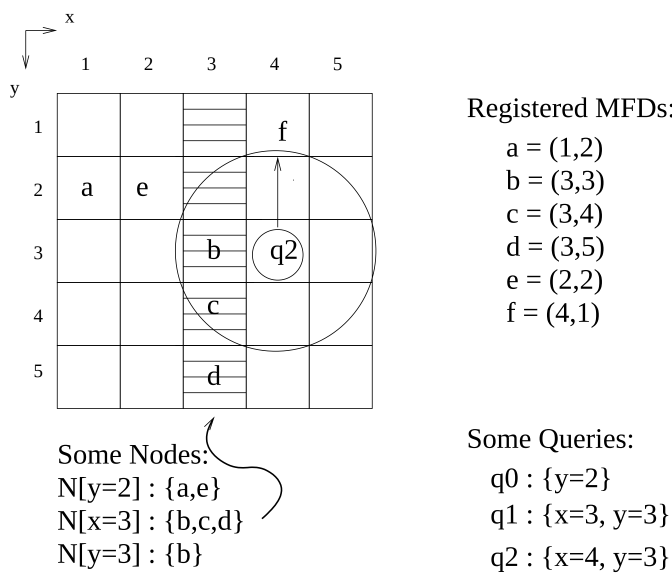

Figure 2: Illustration example of Content Discovery System

5 Scalable Content Discovery

Unlike centralized systems where £les are registered at a single place or broadcast-based systems where a query may potentially be sent to all peers in the system, CDS uses a scalable approach based on Rendezvous Points (RPs) for registration and query resolution. Essentially the P2P network is structured to ef£ciently represent the space of AV-pairs for search and retrieval.

5.1 MFD registration

To register an MFD, the CDS applies a uniform hash function $\mathcal{H}$ such as SHA-1 to each AV-pair in the MFD to obtain n node IDs: $\mathcal{H}(a_i = v_i) \rightarrow N_i, i = 1..n$, where N_i is the ID of a peer in the system. The MFD is then sent to each of these peers, and this set of peers is known as the Rendezvous Points(RP) set for this MFD. Upon receiving an MFD, the peer inserts it into its database. Hence each peer is responsible for the AV-pairs that are mapped onto it. For example, node N_1 will receive all MFDs that contain $\{a_1 = v_1\}$. Figure 2 shows a made-up small scale example of how the system works. The MFD is 2-dimensional with the horizontal coordinate corresponding to the £rst attribute and the vertical coordinate corresponding to the second attribute. This is only done for illustration purposes. The actual system can handle multidimensional data as well. One can visually observe the distribution of the MFD *(a,b,c,d,e,f)* and how they are assigned to nodes such as *N[x=3]*. Basically in two dimensions each node is responsible for a row or a column of the attribute-value grid. Obviously, this results in signi£cant overlap between the entries of the local databases on each node something which enhances the robustness of the system. Even in this contrived example it can be seen that the load on each node depends on the distribution of speci£c AV pairs such as *x=3*. This issue is addressed by the load balancing scheme described in Sect. 5.3.

Since the number of AV-pairs in an MFD is typically small (e.g., < 50), the size of the RP set for an MFD is small and registrations can be done ef£ciently. Different MFDs will have different corresponding RP sets, which naturally separates the system's registration load. Registering each AV-pair of an MFD individually allows the MFD to be searched using any subset of its AV-pairs which important to allow the rich set of possible query semantic we desire for MIR.

5.2 MFD searching

We classify searches conducted by a user into two categories: exact searches and similarity searches. In an exact search, the user is looking for MFDs that match all the AV-pairs speci£ed in the query simultaneously, and any extra AV-pairs that may be in the MFDs but not in the query are ignored. Suppose the query is $Q : \{a_1 = v_1, ..., a_m = v_m\}$. Since the MFDs that match Q are registered at RP peers N_1 through N_m, where $N_i = \mathcal{H}(a_i = v_i)$, the CDS can send a single query message to any of these m peers to have the query fully resolved. For ef£cient resolution, the CDS chooses the peer that has the smallest MFD database. Once a query is received, the peer conducts a pairwise comparison between the query and all the entries in its database to £nd the matching MFDs. An example may be $\{$*artist = U2, year = 1985, tempo = 100 bpm (beats per minute)* $\}$, which means the matched MFD must have the above three AV-pairs in their description. Most likely the node that contains the locations of all U2 songs will have the smallest local MFD database so it will be contacted and locally searched for MFDs with the correct year and tempo.

To further illustrate this process, in Figure 2, the query $q0 : \{y = 2\}$ will go to the corresponding node *N[y=2]* and directly return the correct answer $\{$*a,e*$\}$. The query $q1 : \{x = 3, y = 3\}$ will check the database size of nodes *N[x=3]* and *N[y=3]*, select the smallest *N[y=3]* , and then return the correct answer $\{$*b*$\}$. Note that the answer is the location of the music £le which can then be subsequently downloaded.

In a similarity search, the user is trying to £nd music £les that have a similar feature vector to what is speci£ed in the query. Suppose the user has a clip *unknown.mp3* with an extracted feature vector $\{f_1 = v_1, ..., f_m = v_m\}$, and wants to £nd the 10 songs that are most similar to the clip. Using the same technique as above, the CDS may select a pair, e.g., $\{f_1 = v_1\}$ and send the query to the peer $N(= \mathcal{H}(f_1 = v_1))$. This peer, instead of conducting a pairwise equality test, computes the "distance" between the query vector and each MFD in its database. In our current system, Manhattan distance de£ned as $d(f, f') = |v_1 - v'_1| + ... + |v_m - v'_m|$ is used, where v_i's and v'_i's are the values of vector f and f' respectively. More sophisticated way of computing distance, such as "cosine distance" may also be used. The distances are then ranked and the 10 MFDs that have the smallest distance are returned to the user.

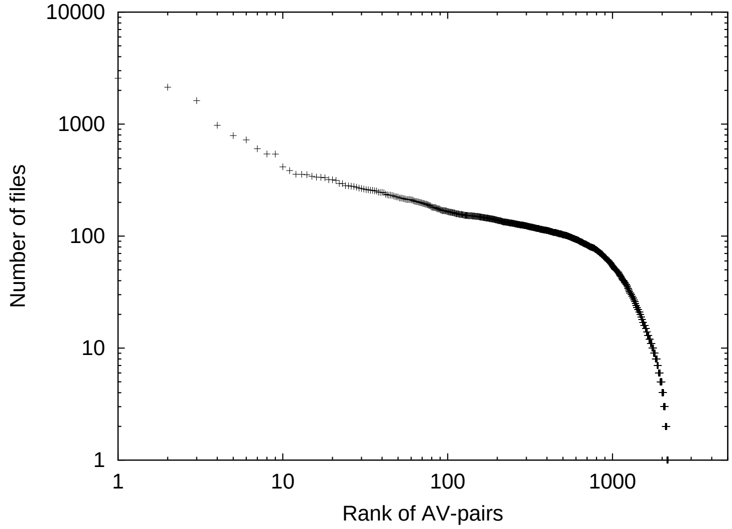

Figure 3: Popularity distribution of feature attributes.

However, sending the query to peer N alone will fail to discover the MFDs that slightly differ from the query in f_1, but are similar or identical regarding other features, because those MFDs are not registered with N. This is undesirable especially when N does not have enough matches. In this case, our system uses a limited expanding ring search to gather more results: in addition to N, the query is sent either by N or the query initiator to peers that correspond to values that are near v_1, e.g., $f_1 = v_1 \pm 1, f_1 = v_1 \pm 2, ...$ Accordingly, these peers will carry out the distance computation and return any results. Figure 2 schematically shows this expanding ring search for query $q2 : \{x = 3, y = 4\}$ as two cocentric circles. Of course it is also possible to combine exact AV search and content-based similarity search. This last point is important and directly in¤uenced the design of our system. Although it is possible to use more elaborate data structures for multidimensional nearest neighbor search (the similarity search) such as KD-trees (Bentley (1975)) and to distribute them over the P2P network these structures do not directly support searching for arbitrary subsets of AV-pairs as our system does. Supporting such searches is important for MIR because we would like to combine both metadata and content-based retrieval in the same query. Finally, range queries such as $\{$*tempo=80-120 bpm*$\}$ are resolved by issuing multiple queries corresponding to the values within the range. We are experimenting with a more ef£cient multiresolution approach to range searching where not only values but also ranges of values are assigned to nodes. That way the number of nodes that need to be visited in order to resolve a range query is reduced in $O(logL)$ where L is the length of the range.

5.3 Load balancing

By using Rendezvous Points, network-wide message ¤ooding is avoided at both registration and query times. However, in practice, some AV-pairs may be much more common or popular in MFDs than others. It has been observed that the popularity of keywords in Gnutella follows a Zipf-like distribution (Sripanidkulchai (2001)). Such a distribution will cause a few peers being overloaded by registrations or queries, while the majority of peers in the system stay underutilized. Figure 3 shows a distribution of AV-pairs computed from automatically extracted musical content features described in more detail in the section 6. To improve system's throughput under skewed load, the CDS deploys a distributed dynamic load balancing mechanism described in Gao and Steenkiste (2003), where multiple peers are used as RP points to share the heavy load incurred by popular AV-pairs. When an AV-pair appears in many MFDs, instead of sending all the MFDs to one peer, the system partitions them among multiple peers. Similarly if there are a large number of queries for the same AV-pair, the system allows the original peer who is responsible for this pair to replicate its database at other peers. As a concrete example some particular AV pair such as *tempo = 100 BPM* might be very common. Partitioning the set of songs that have that particular AV pair among multiple RP points balances the registration load. Another type of load which is possibly independent of the registration load is the popularity of queries for example *artist = Madona*, *year =2003* for a new release that will initially only be registered with a few RPs. Replicating their information balances the query load. The partitions and replicas corresponding to one AV-pair are organized into a two dimensional logical matrix, the Load Balancing Matrix (LBM), and the matrix automatically expands or shrinks based on this pair's current registration and query load. LBMs help to eliminate hot spots in the system under skewed load, and the system can maintain high throughput in processing registrations and queries (Gao and Steenkiste (2003)).

6 System Evaluation

The MFEE is built using Marsyas (Tzanetakis and Cook (2000)), a free software framework for audio analysis. We evaluate our system using an event-driven simulator (Gao and Steenkiste (2003)). For our experiments, we set up a P2P network that has 10,000 peers, and each peer is con£gured with DSL-level link bandwidth ($\sim 500Kbps$). As MFDs, 30 music content-based features are used as attributes. They were automatically extracted from 5,000 MP3 £les representing a variety of genres and styles. Figure 3 shows the log-log plot of the AV-pair distribution in these £les. There are 2,178 distinct AV-pairs, and the distribution is highly skewed: the most common AV-pair (ranked 1) appears in 53% of the MFDs and 41 AV-pairs only appear in 1 MFD.

For registration workload, we generated 100,000 MFDs by replicating each of the 5,000 £les 20 times, and assigning them to random peers. Each peer registers the £les it is assigned with the system. Due to the skewed feature distribution, registrations of common AV-pairs result in multiple partitions. For query load, 100,000 queries were generated following a query popularity Zipf distribution which is independent from the AV-pair distribution shown above in Figure 3. Each query corresponds to the features of one particular music £le. This is done in order to emulate the behavior of a user who submits a music clip and looks for similar music. The most popular MFD occurs in over 10% of the queries, and the majority of the MFDs only occur in a few queries. A query's initiator is randomly picked from all peers, and for simplicity, only exact matches are returned. A peer rejects a query and generates a failure when the peer's link utilization has reached 100% due to simultaneous queries.

Figure 4 compares the query success rate as a function of query arrival rate (Poisson arrival) to the system under two scenarios. In the £rst scenario, when reaching link capacity, a peer simply rejects new queries that arrive at it without replicating its content at other peers. Since for each query the CDS has 30 candidate AV-pairs, query load is spread evenly among peers even without replication. Therefore the system achieves a high success rate under high load, e.g., the success rate is 94% for a query rate of $10,000q/s(queries/s)$. However, as load increases further, peers corresponding to popular queries will be saturated, and the success rate drops quickly. In the second scenario, by using the dynamic replication mechanism, peers who observe high load will replicate their databases at other peers to dissipate concentrated query load. As a result, we observe that with replication, the system can sustain a much higher query rate while keeping the success rate above 95%.

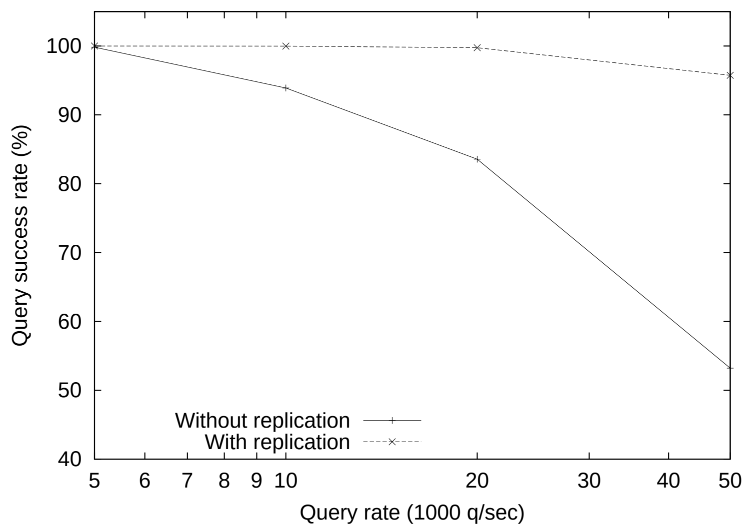

Figure 4: Query success rate comparison.

7 Conclusions

In this paper, we described a scalable, and load-balanced P2P system that supports a rich set of music search methods. In particular, our automatic music feature extraction technique enables sophisticated music content based searches such as content-based similarity retrieval. The RP-based registration and query speci£cation scheme ensures system scalability by avoiding network wide message ¤ooding encountered in current P2P systems. We evaluated the system using a realistic registration load obtained from a large set of music £les. Our dynamic load balancing mechanism allows the system to maintain high throughput under skewed Zipf query load. It is our hope that the design of our system will inspire additional research in the interesting area of bringing state of the art Music Information Retreival algorithms to the increasingly popular Peer-to-Peer networks. Another important aspect of the proposed system is that new attributes can be incorporated into the system with minimum effort.

We are currently re£ning the design of our system to handle range queries more ef£ciently and plan to further evaluate our system with traces acquired from real users. Further experiments are necessary for a more detailed evaluation of system performance. In addition, user studies need to be conducted to explore how the users interact with the system and what are their typical queries. We are working on implementing a prototype of our system across a medium area (university campus) LAN to obtain more information about usage patterns and the performance of the system.

In order to handle different types of range queries we are planning to explore multi-resolution quantization grids. Another important direction is the and inclusion and creation of various query speci£cation user interfaces for specifying the MFDs. Finally, duplicate copies of the same audio content can be detected by adapting an audio £ngerprinting scheme such as Haitsma and Kalker (2002) to work with our P2P system.

Acknowledgements

This work was supported by NSF grant 0085945 of the National Science Foundation. Additional support was provided by Intel.

References

Aucouturier, J.-J. and Pachet, F. (2002). Music Similarity Measures : What's the use ? In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 157–163, Paris, France.

Aucouturier, J. J. and Pachet, F. (2003). Musical Genre: a Survey. *Journal of New Music Research*, 32(1).

Baumann, S. (2003). Music Similarity Analysis in a P2P environment. In *Proc. of the 4th European Workshop on Image Analysis for Multimedia Interactive Services*, pages 314–319, London, UK.

Baumann, S. and Kluter, A. (2002). Super-Convenience for Non-musicians: Querying MP3 and the Semantic Web. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 297–298, Paris, France.

Bentley, J. (1975). Multidimensional Binary Search Trees used for Associative Searching. *Communications of the ACM*, 18(9).

Foote, J. and Mathew, C. (2002). Audio Retrieval by Rhythmic Similarity. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 265–266, Paris, France.

Futrelle, J. and Downie, S. J. (2002). Interdisciplinary Communities and Research Issues in Music Infomration Retrieval. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 215–221, Paris, France.

Gao, J. and Steenkiste, P. (2003). Design and Evaluation of a Distributed Scalable Content Discovery System. *IEEE Journal on Selected Areas in Communications (Special Issue: Recent Advances in Service Overlay Networks)*. (to appear).

Gao, J., Tzanetakis, G., and Steenkiste, P. (2003). Content-based retrieval of music in scalable peer-to-peer networks. In *Proc. Int. Conference on Multimedia and Expo (ICME)*, Baltimore, US.

Haitsma, J. and Kalker, T. (2002). A Highly Robust Audio Fingerprinting System. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 107–115, Paris, France.

Logan, B. and Salomon, A. (2001). A Music Similarity Function based on Signal Analysis. In *Int. Conf. on Multimedia and Expo (ICME)*. IEEE.

Pachet, F. (2003). Content Management for Electronic Music Distribution: The Real Issues. *Communications of the ACM*, 46(4).

Rauber, A., Pampalk, E., and Merkl, D. (2002). Using Psycho-Acoustic Models and Self-Organizing Maps to Create a Hierarchical Structure of Music by Sound Similarity. In *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, pages 71–80, Paris, France.

Rowstron, A. and Druschel, P. (2001). Pastry: Scalable Distributed Object Location and Routing for Large-scale Peer-to-Peer Systems. In *Proc. of Middleware*, pages Heidelberg, Germany.

Sripanidkulchai, K. (2001). The Popularity of Gnutella Queries and Its Implications on Scalability. `http://www.cs.cmu.edu/~kunwadee/research/p2p/gnutella.html`,.

Stoica, I., Morris, R., Karger, D., Kaashoek, F., and Balakrishnan, H. (2001). Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *Proc. of SIGCOMM*, pages 149–160, San Diego, CA.

Tolonen, T. and Karjalainen, M. (2000). A Computationally Ef£cient Multipitch Analysis Model. *IEEE Trans. on Speech and Audio Processing*, 8(6):708–716.

Tzanetakis, G. and Cook, P. (2000). Marsyas: A framework for audio analysis. *Organised Sound*, 4(3).

Tzanetakis, G. and Cook, P. (2002). Musical Genre Classi£cation of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5).

Tzanetakis, G., Ermolinskyi, A., and Cook, P. (2002). Beyond the Query-by-Example Paradigm: New Query Interfaces for Music Information Retrieval. In *Proc. Int. Computer Music Conference (ICMC)*. Gothenburg, Sweden.

Wang, C., Li, J., and Shi, S. (2002). A Kind of Content-Based Music Information Retrieval Method in a Peer-to-Peer Environment. In *Proc. Int. Conf. on Music Information Retrieval (ISMIR)*, pages 178–186, Paris, France.

Whitman, B. and Smaragdis, P. (2002). Combining Musical and Cultural Features for Intelligent Style Detection. In *Proc. Int. Conf. Music Information Retrieval (ISMIR)*, pages 47–52, Paris, France.

Yang, C. (2002). The MACSIS Acoustic Indexing Framework for Music Retrieval: An Experimental Study. In *Proc. Int. Conf. on Music Information Retrieval*, pages 52–62, Paris, France.