A MELODY-RETRIEVAL SYSTEM ON PARALLEL-IZED COMPUTERS

Tomonari Sonoda, Toshiya Ikenaga, Kana Shimizu and Yoichi Muraoka
School of Science and Engineering, Waseda University
3-4-1 Ohkubo Shinjuku-ku, Tokyo 169-8555, Japan.
{sonoda,ikenaga,kana,muraoka}@muraoka.info.waseda.ac.jp

Abstract This paper describes a method for a WWW-based melody-retrieval system, takes a melody sung by a user as a search clue and sent over the Internet and uses it to retrieve the song's title from a music database of standard MIDI files(SMF). It was difficult to build a melody-retrieval service with a large database and with a lot of user accesses since it was quite difficult to build a system which could achieve both quick search and high matching accuracy. We propose a method of a scalable melody-retrieval system which achieves 70% matching accuracy against more than 20,000 pieces of music and its search time is within a few seconds.

Introduction

In building melody-retrieval services on the Internet, it is quite important to keep quick and accurate retrieval against a large database. In previous paper [7], we developed an effective indexing method for melody sequences.

It was, however, difficult to keep matching accuracy when we limited the size of index data for reducing the search time. There was about 5 % matching accuracy lost as compared with the case where an index is not used.

In addition, our previous system with a large database which contained 10,000 melodies could accept only one user access every second since the performance was limited by the ability of a computer.

To solve these problems, we propose a method of parallel-ized melody-retrieval servers. The structure of server network is scalable. It can consist of only a server on a computer and it can also be extended to any number of server computers.

Testing of our method with parallel-ized 5 PCs against a database containing 21,500 melody sequences showed that its retrieval time was 5.5 seconds on average. Its matching result was 70% for 1,200 inputs from 3 different people.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35660-0_65

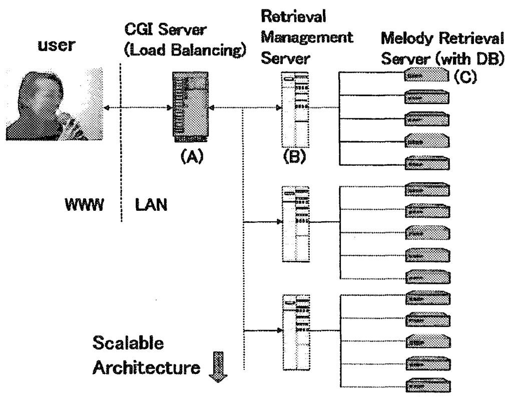
user
CGI Server (Load Balancing)
Retrieval Management Server
Melody Retrieval Server (with DB)
(A)
(B)
(C)
WWW
LAN
Scalable Architecture

Figure 1. A melody-retrieval system on parallel-ized computers.

1. Melody-Retrieval System

This section describes the overview of our system and the necessity of scalable server system.

1.1. Overview

Fig.1 shows our current system which consists of user-side clients and several servers. It is quite different from our previous system[7] which consisted of the client and only one melody-retrieval server.

As shown in Fig.1, the server system has at least one melody-retrieval server with a music database(C) and it can be parallel-ized by several servers(A,B) which relays users' queries and matching results.

Each melody-retrieval server treats a matching process that compares the user's input melody with each database melody. After the matching process, our server returns a list of music titles that have similar melody-part to the user's input melody.

The main function of a user-side client, on the other hand, is to record user's sung melody. A user can sing an arbitrary part of a desired melody by using any key or any tempo. The acoustic-signal of recorded voice is converted to small melody data and it is transmitted to a server system as a search clue. The client finally receives the music list from the server system and shows it to the user as a matching result.

1.2. Necessity of Server Scalability

The most important point in building a melody-retrieval system is to keep both quick search and high matching accuracy against a large database. It had been pointed out that especially the search time is quite difficult to be shortened when we indtend to increase the number of database melodies[1, 2, 4, 5].

To solve this problem we have proposed an indexing method based on dynamic-programming and developed a melody-retrieval system with a large database which could quickly finish retrieval process[7].

Though the method quite effective for reducing the search time, there was a little lost of matching accuracy.

When we intended to keep search time within a second and limited the size of index data up to the fix size, our system lost 5% accuracy against 10,000 pieces of music and the lost of accuracy increased along with the size of database.

It was therefore necessary to develop a method which kept both quick search and high matching accuracy.

To solve this proble, we propose a scalable server system by using parallel-ized computers and achieve both quick search and high matching accuracy against any size of database.

2. A Method of Parallel-ized Servers

This section describes a method of parallel-ized servers and each function of the server.

2.1. A Method of Scalable Server Structure

To parallel-ize several melody-retrieval servers, we propose a method which utilizes following two types of data-relay servers; (A)a CGI-server which relay a user's query data to one of connectable servers and (B)a retrieval managemant server which relay a uer's query data to all of connectable servers.

By using this method, the server system can consist of various combination of servers and can have scalability as follows.

1 Fig.2-(1) shows the minimum structure which consists of only a melody-retrieval server(C). The structure is for a system with a small database and with few simultaneous user accesses.

2 Fig.2-(2) shows a structure which consists of a CGI-server(A) and several melody-retrieval servers(C). The structure is for a system with a small database and with a lot of simultaneous user accesses.

3 Fig.2-(3) shows a structure which consists of a retrieval management server(B) and several melody-retrieval servers(C). The structure is for a system with a large database and with few simultaneous user accesses.

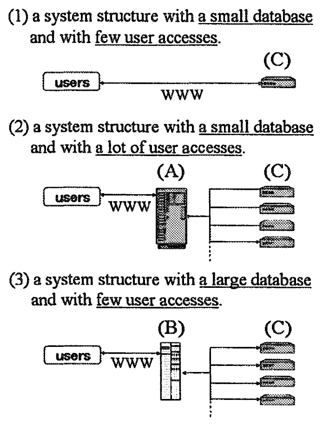
(1) a system structure with a small database and with few user accesses.
(C)
users
WWW
(2) a system structure with a small database and with a lot of user accesses.
(A)
(C)
users
WWW
(3) a system structure with a large database and with few user accesses.
(B)
(C)
users
WWW

Figure 2. Scalable Server System.

4 Fig.1 shows the maximum structure which consists of all of servers (A),(B) and (C). The structure is for a system with a large database and with a lot of simultaneous user accesses.

2.2. Function of Each Server

We describe the function of each server in the following paragraphs.

(A) CGI-server. The current CGI-server is implemented by CGI(Common Gateway Interface) program for a WWW server. When a CGI-server accepts a user's query, it sends the query to a retrieval management servers(B) or to a melody-retrieval server(C) and relay the server's message to the client.

(B) Retrieval Management Server. Each retrieval management server has the list of melody-retrieval servers to manage. After receiving a user's query, it distributes the query to listed melody-retrieval servers like broadcasting. Then the retrieval management server receives ranked music lists as the matching results from each melody-retrieval server and it sorts all of them in order by the similarity to the user's input melody. The sorted music list is returned to the user-client.

(C) Melody-Retrieval Server. Each melody-retrieval server has a database consisting of SMF. After receiving the user's query, the server executes a matching process. It compares the melody sequences of user's query with sequences for each database melody by using DP matching and titles of the corresponding melodies are then ranked based on the distance calculated by the matching process. Finally, the server transmits the list of the ranked song titles to the client as a matching result. Our current system utilizes a matching method which was proposed in the previous paper[3].

3. User-side Client

The user-side client plays a role of recording the user's voice and transmit it to the server-side. In this process, we use our proposed method in the previous paper[3].

In using our system, at first, a user inputs a melody into a microphone by singing. The client allows the user to input from an any part of a piece of music and to use any key or tempo. The client program assumes that each input note begins with a voiceless consonant and ends with a vowel (e.g. ta-ta-ta / cha-cha-cha) and that input sound is only user's voice. This input method has an advantage that the client system clearly detects each note while it is not a heavy task for a user.

Our client then extracts sequences of pitch and span of notes from the sung melody and converts them into relative-value sequences of pitch and span of notes. Those relative-value sequences are transmitted to a melody-retrieval server system as a search clue.

After search process in the server system, our client system receives a music-list as a search result from the server system and shows it to the user.

We developed a client user-interface shown in Fig. 3. It has three buttons, record, stop and search, and a text box to input a melody and text query. The interface also has an area for graphical feedback of user's sung melody.

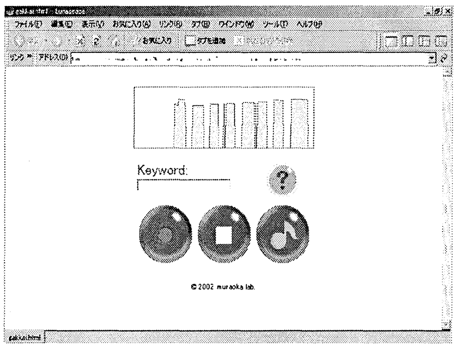

Figure 3. Client Interface.

4. Experiments and Results

We evaluate the proposed method and verify its effectiveness in this section.

To test our melody-retrieval server, we ran it on several PCs and each of PC has 850-MHz $IntelPentium^{TM}III$ processor and 256 MB of RAM.

We used a database containing 21,500 melodies, consisting of 20,000 pieces of randomly generated music and 1,500 pieces of music from popular songs.

We used 1,200 input queries from 3 people (2 men and 1 woman). Each recoding time of voice was limited within 15 seconds. The average number of notes of input melodies was 28.0.

We evaluated our system in 3 ways; (1) throughput of the system, (2)database scalability, and (3) the entire performance.

4.1. Evaluation for Throughput of the System

To evaluate the relationship between throughput of the system and the number of melody-retrieval servers, we measured the number of user queries which can be processed a second by increasing the number of PCs from 1 to 5. We used 6 PCs for 1 CGI-server and 5 melody-retrieval servers and built the network same with being shown in Fig.2-(2). In this experiment, each server had the same database of 21,500 melodies and the result of this test is shown in Fig.4.

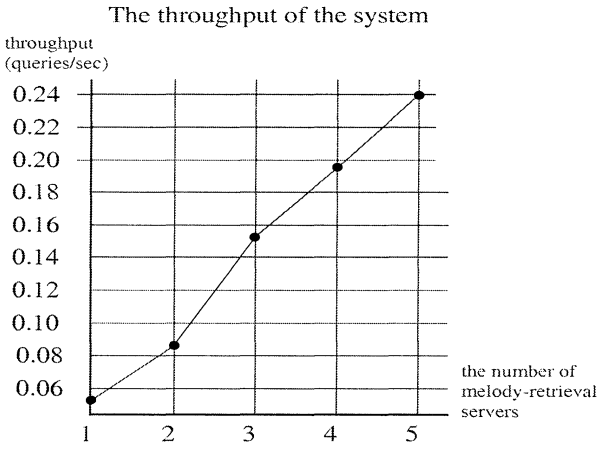

Figure 4. The relationship between the throughput and the number of melody-retrieval servers.

We confirmed that the system throughput improves according to the number of melody-retrieval servers.

4.2. Evaluation for the Database Scalability

To evaluate the database scalability, we measured the relationship between the database size and the search time. We used 1 retrieval management server and 5 melody-retrieval servers for this experiment. Each melody-retrieval server had a different database which contained 4,300 melodies. We increased the number of melody-retrieval servers from 1 to 5 and increased the database size according to the number of servers from 4,300 to 21,500. We built the server network same with being shown in Fig.2-(3) and tested the system by using a query which note length was 28. Table 1 shows the search time for each number of melody-retrieval servers. Each search time is the average of 5-time-search.

This result shows that our system could keep the search time within the fix time by using a retrieval management server and parallel-ized melody-retrieval servers.

Table 1. Database Scalability

parallel-ized servers	database size	the search time(sec)
1	4,300	4.62
2	8,600	5.22
3	12,900	5.44
4	17,200	5.45
5	21,500	5.48

4.3. Evaluation for the System Performance

We finally evaluated overall performance of our system against 21,500 pieces of database by measuring matching accuracy and search time. We used 6 PCs for 1 retrieval management server and 5 melody-retrieval servers, which was the same with previous evaluation. The matching accuracy for 1,200 queries was calculated by counting the case that user's intended music is ranked in the top of a music list of the search result.

Table 2 shows the result and we confirmed that our system could keep both high matching accuracy and quick search against a large music database.

5. Conclusion

We have proposed an effective method for a WWW-based melody-retrieval system on parallel-ized computers. The method enables a melody-retrieval system to keep high matching accuracy and quick search against a large database and we achieved to build a system which matching accuracy was 70% and its

Table 2. System Performance

the database melodies	21,500
matching accuracy	70.2%
average of the search time	5.48 sec
parallel-ized melody-retrieval servers	5
the number of queries for testing	1,200
the average length of queries	28.0

search time was 5.5 seconds against a database which contained more than 20,000 pieces of music when we used 5 parallel-ized PCs.

As the future work, we plan to extend our system so that not only SMF but actual music can be searched[6].

We are also planing to apply our system for using over phones so that user's can use the system at any time.

References

[1] T.Kageyama, K.Mochizuki, and Y.Takashima *Melody Retrieval with Humming*, ICMC Proc., pp.349-351, 1993.

[2] Asif Ghias and Jonathan Logan *Query By Humming – Musical Information Retrieval in an Audio Database*, ACM Multimedia 95, Electronic Proc., 1995.

[3] Tomonari Sonoda, Masataka Goto, Yoichi Muraoka: *A WWW-based Melody Retrieval System*, pp.349-pp.352, ICMC'98 Proc., 1998.

[4] Lloyd A. Smith, Rodger J. McNab, and Ian H. Witten: *Sequence-Based Melodic Comarison: A Dynamic-Programming Approach*, pp.101-pp.116, COMPUTING IN MUSICIOLOGY 11, 1997-98.

[5] David Bainbridge: *MELDEX: A Web-based Melodic Locator Service*, pp.223-pp.229, COMPUTING IN MUSICIOLOGY 11, 1997-98.

[6] Masataka Goto, and Satoru Hayamizu : *A Real-time Music Scene Description System: Detecting Melody and Bass Lines in Audio Signals*, Working Notes of the IJCAI-99 Workshop on Computational Auditory Scene Analysis, pp.31-40, August 1999.

[7] Tomonari Sonoda and Yoichi Muraoka: *A WWW-based Melody Retrieval System -An Indexing Method for a Large Database-*, ICMC2000 Proc., 2000.