MIRACLE: A Music Information Retrieval System with Clustered Computing Engines[1]

J.-S. Roger Jang, Jiang-Chun Chen, Ming-Yang Kao

Multimedia Information Retrieval Laboratory
Computer Science Department, National Tsing Hua University, Taiwan
{jang, jtchen, gao}@cs.nthu.edu.tw

Keywords
Content-based Music Retrieval, Audio Indexing and Retrieval, Dynamic Programming, Dynamic Time Warping, Query by Singing, Cluster Computing, Parallel & Distributed Computing

1. Introduction

This paper presents a music information retrieval system based on parallel and distributed computing. The system, called MIRACLE (Music Information Retrieval Acoustically with Clustered and paralleL Engine), can take a user's acoustic input (about 8 seconds) and perform similarity comparison on a group of clustered PCs. The system is a paradigm of client-server distributed computing since the pitch tracking is performed at the client computer while the time-consuming process of similarity comparison is performed at the server. Moreover, the similarity comparison is handled by a master server which partitions the comparison task into subtasks and dispatches these subtasks to a collection of slave servers. Currently there are more then 10,000 songs in MARACLE and the average retrieval time for "match beginning" is less than 1 seconds. The top-10 recognition rate for "match beginning" is 72%, and for "match anywhere", 56%. Extensive tests and performance evaluation demonstrates that MIRACLE is a feasible system that suits common people's needs of music information retrieval.

2. Related Work of MIR Systems

As the needs for music information retrieval rises, there are many MIR systems reported in the literature, including QBH (Query by Humming) by Ghias et al. [1], MELDEX (Melody Indexing) by Bainbrideg et al. [6], SoundCompass by Kosugi et al. [5], Super MBox by Jang et al. [2], Themefinder by Kornstadt et al.[3], MELODISCOV by Rolland et al.[7], etc. However, most of the above systems do not allow acoustic input from users directly; therefore the usability of those systems is significantly limited. Even within those MIR systems based on acoustic input, only MELDEX and Super MBox (a precursor of MIRACLE) have web deployment, which allows general public access. Particularly, as far as we know, MIRACLE is the first MIR system that is based on cluster computing.

The authors have also published their work on a content-based MIR system called Super Mbox [2]. The focus of the publications is on the use of dynamic programming techniques for elastic match in the comparison engine. A significant advantage of using DTW is that users are not required to singing "ta" to facilitate note segmentation, as required by MELDEX and SoundCompss. Being a precursor of MIRACLE, Super MBox only allows the use of DTW on a single processor. MIRACLE, on the other hand, adopts a two-step hierarchical filtering method (HFM) that filters out 90% candidates using an efficient linear-scaling comparison, and then employs DTW to compare the survived 10% candidates.

3. Distributed and Parallel Computing

MIRACLE is composed of a client and a server component. The client component takes users' acoustic input and transforms it into a pitch vector. The resulted pitch vector is then send to the server for similarity comparison. At the server side, the request is first handled by a master server which partitions the whole song list into partial lists, and then dispatches these partial lists to 18 client PC servers (ranging from Pentium 166 MHz to 1 GHz). Once a slave server receives its candidate list from the master server, it starts similarity comparison and return top-20 most likely candidate songs to the master server. The master server then combines and reorders the top-20 lists from all slave servers to generate the overall top-20 ranking list.

The initial length of the comparison song list for slave server p, denoted by l_p, is proportional to the clock rate of the slave server, namely, $l_p = l * \frac{C_p}{\sum_{k=1}^{18} C_k}$, where C_k is the clock rate of slave server k and l is the length of the total song list. To adaptively change the formula for l_p at request i, denoted by $l_p(i)$, can be expressed as $l_p(i+1) = l_p(i) * \exp\left\{\frac{t_{ave}(i) - t_p(i)}{k}\right\}$, where $t_p(i)$ is the response time of slave server p at request i; $t_{ave}(i) = \frac{1}{18}\sum_{j=1}^{18} t_j(i)$ is the average response time over all slave servers at request i; and k is a constant used to control the sensitivity of the adaptation. The goal of the adaptive load balancing strategy is to eventually have a same response time for each slave server. In fact, the above expression for $l_p(i+1)$ is not the final value for our experiment since $l_p(i+1)$ must fulfill the

---

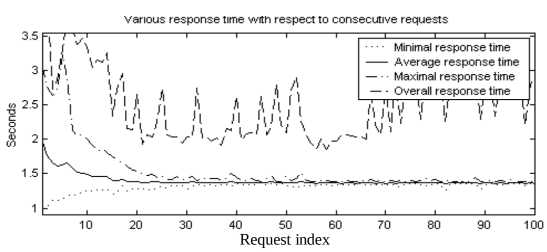
[1] This is an on-going project supported by Cweb Inc. (http://www.4music.com.tw) and the Excellency Project sponsored by the National Science Council at Taiwan, where the authors express many thanks.

constraint that the total length should be equal to l . Hence the final value of $l_p(i+1)$ is $l_p(i+1) = l * \frac{l_p(i+1)}{\sum_{j=1}^{18} l_j(i+1)}$.

4. Performance and Discussions

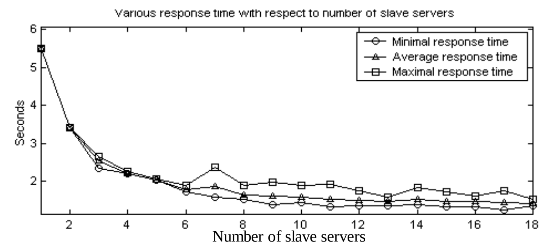
To test the recognition rate of MIRACLE, we have a large collection of 1550 vocal clips from 20 persons. 1054 of the vocal clips are from the beginning of songs, while the other 496 are from the middle. Some of the recordings are obtained via regular handsets or mobile phones to test the robustness of the pitch tracking algorithm. For the case of "match beginning", we sent the 1054 files to MIRACLE that employs two-step HFM as the comparison procedure. The average response time is about 2.29 seconds. The top-3 recognition rate is 65.75%; top-10 is 70.68%. If we choose DTW instead of two-step HFM, the top-3 recognition rate is 65.56% and top-10 72.58%. It is obvious that DTW and two-step HFM have comparable recognition rates. However, DTW's average search time is about 5 seconds, which is much longer than that of two-step HFM. For "match anywhere", we sent the 496 vocal clips to the master server and the average response time is about 5 seconds. The top-3 recognition rate is 43.29%; top-10 is 49.42%.

To test the adaptive strategy for load balancing, we measured various response time for 100 consecutive requests of "match anywhere" sent to the master server. The plot of various response time with respective to request indices can be shown in the following figure:



Obviously our adaptive strategy can effectively balance the loads such that the response time of each slave server approaches the same. Since the slave servers are not dedicated to MIRACLE only, we can see some sudden increases in the slowest response time.

The following plot shows the curves of various response time (after 100 requests, and taking the average of the last 10 requests) with respect to number of slave servers:



From the above plot, we can observe that the average response time levels off when the number of slave servers is 10 or more. We can conclude that for a MIR system with 10,000 songs, 10 clustered PCs are qualified for the requests of "match anywhere".

To test drive MIRACLE, please follow the link of "Online demo of Super Mbox" at the author's homepage at:

http://www.cs.nthu.edu.tw/~jang

5. References

[1] Ghias, A. J. and Logan, D. Chamberlain, B. C. Smith, "Query by humming-musical information retrieval in an audio database", ACM Multimedia '95 San Francisco, 1995. (http://www2.cs.cornell.edu/zeno/Papers/humming/humming.html)

[2] Jang, J.-S. Roger and Gao, Ming-Yang "A Query-by-Singing System based on Dynamic Programming", International Workshop on Intelligent Systms Resolutions (the 8th Bellman Continuum), PP. 85-89, Hsinchu, Taiwan, Dec 2000.

[3] Kornstadt, Andreas. "Themefinder: A Web-based Melodic Search Tool," *Melodic Comparison: Concepts, Procedures, and Applications*, *Computing in Musicology* 11 (1998), 231-236.

[4] Kosugi, N. Y., Kon'ya, Nishihara, S., Yamamura, M. and Kushima, K. "Music Retrieval by Humming – Using Similarity Retrieval over High Dimensional Feature Vector Space," pp 404-407, IEEE 1999.

[5] Kosugi, N., Nishihara, Y., Sakata, T., Yamamuro, M., and Kushima, K., "A practical query-by-humming system for a large music database," In Proc. ACM Multimedia 2000, November 2000.

[6] McNab, R. J., Smith, L. A. and Witten, Jan H. "Towards the Digital Music Library: Tune Retrieval from Acoustic Input" ACM, 1996.

[7] Rolland, P.-Y. "Discovering patterns in musical sequences," Journal of New Music Research, vol. 28, no. 4, pp 334-350, 1999.