Overview of OMEN

Daniel McEnnis, Cory McKay, and Ichiro Fujinaga
Music Technology, Schulich School of Music, McGill University
555 Sherbrooke Street West
Montreal, QC H3A 1E3
{daniel.mcennis, cory.mckay}@mail.mcgill.ca, ich@music.mcgill.ca

Abstract

This paper introduces OMEN (On-demand Metadata Extraction Network), which addresses a fundamental problem in MIR: the lack of universal access to a large dataset containing significant amounts of copyrighted music. This is accomplished by utilizing the large collections of digitized music available at many libraries. Using OMEN, libraries will be able to perform on-demand feature extraction on site, returning feature values to researchers instead of providing direct access to the recordings themselves. This avoids copyright difficulties, since the underlying music never leaves the library that owns it. The analysis is performed using grid-style computation on library machines that are otherwise under-used (e.g., devoted to patron web and catalogue use).

Keywords: Music database, datasets, feature extraction, distributed computing

1. Introduction

It is becoming increasingly important for music information retrieval (MIR) researchers to have access to a large, varied dataset of music with which to test their algorithms. Researchers also need access to the same datasets in order to meaningfully compare results. From the 2005 MIREX abstract [1]:

> … there remain several serious challenges that must be overcome in order to conduct future MIREX contests that consistently provide meaningful and fair scientific evaluations. These challenges include:
>
> 1. The continued near impossibility of establishing a common set of evaluation databases or the sharing of databases among researchers due primarily to intellectual property restrictions and the financial implications of those restrictions; …

The size of such a universal dataset is an obstacle. In order to provide a solid foundation for comparing algorithms, a dataset consisting of many gigabytes is needed to perform analysis. Even disregarding copyright issues, a dataset that is sufficiently large to be useful would be prohibitively expensive to transfer between researchers.

Furthermore, researchers often want to make sure that the results obtained during their research are applicable to music in use by the general public. This music is almost exclusively copyrighted music. Transferring this kind of music between researchers can involve insurmountable legal obstacles, even for a relatively small dataset.

These problems make directly copying the database for each researcher infeasible. One solution is to centralize all aspects of the analysis process from feature extraction to output. Alternatively, one can take advantage of the nature of MIR research by splitting tasks into two stages: feature extraction and analysis of these features. Issues relating to the feature extraction stage are addressed by OMEN, and analysis of the features is left to individual researchers.

Although centralizing all aspects of the feature extraction and analysis is the simplest approach, this requires that a central site acquire all the music in the dataset. This is prohibitively expensive. Furthermore, this site must also provide all of the computing resources for all of the analyses—also an expensive proposition.

If one provides access to the underlying features instead of performing all processing locally, then the most important information relating to the dataset can be distributed. This does not violate copyright law provided that the features extracted cannot be used to reconstruct the original recording.

One way of achieving this is to pre-compute features from the dataset and then publish the results. However, there are a number of different parameters involved in feature extraction. Accounting for all possible variations of these parameters would result in a combinatorial explosion of variations to be computed. In addition to being infeasible computationally, the storage requirements for this extraction would be prohibitively large. Furthermore, adding a new feature immediately precipitates additional computation.

In order to circumvent these problems, OMEN calculates features as needed, on-demand. Only those calculations requested by researchers are computed, reducing the computation to a reasonable level. Since calculations can be repeated, if necessary, the amount of

disk space available is not an issue. Furthermore, new features can be added without requiring the additional computation involved with pre-computing features.

OMEN was designed with the following priorities in mind:

- The dataset may be distributed among multiple physically remote computers.
- Libraries should be the primary providers of music material.
- A library's costs (e.g., new hardware) to join OMEN should be minimal.
- Libraries should be able to easily add and modify materials made available through OMEN.
- Libraries should be able to control the use of their own resources.
- Adding new features to OMEN should be easy.
- Extracted feature values should be stored in standard data formats.
- Feature extraction should occur only when requested.
- Researchers should not need to know where datasets are located.
- It should be easy to formulate a feature extraction request.
- There must be a central access point for MIR researchers to make feature extraction requests and otherwise interact with OMEN.
- Researchers should be able to monitor the progress of their requests.
- Space permitting, OMEN should cache extracted features.

There are multiple ways to describe a distributed system such as OMEN. This paper utilizes three overlapping approaches. Firstly, the structure of OMEN is described by the activities that occur at each computer. Secondly, the interfaces provided by OMEN are described by how different users of OMEN will interact with the system. Finally, certain fundamental tasks performed by OMEN are described from start to finish.

2. Related Work

There are a number of projects that are either used by OMEN or implement a subset of the functionality OMEN provides. jAudio is the feature extraction system utilized by OMEN. MIREX is a competition that could greatly benefit from a universal music dataset made available through OMEN. There are also a number of alternative datasets intended for MIR research. Finally, there exist several other systems that provide distributed feature extraction.

2.1 jAudio

jAudio [2,3] provides an interface and engine for extracting features from audio files. It supplies features, metafeatures, and aggregators to accomplish this task. jAudio also has the important advantage that it can add new features during execution when used as a library. jAudio serves as OMEN's feature extraction library.

2.2 MIREX

The first MIREX competition [1], held in 2005 in conjunction with ISMIR 2005, provided a forum where MIR researchers could evaluate their algorithms for various tasks using common datasets. As a result, the organizers of MIREX faced the problem associated with building datasets for each of the different MIR tasks.

MIREX chose to use a centralized approach to address these problems, relying on donations of music from participants to construct their datasets. Some of these donations were of questionable legality since they involved copyrighted music. This resulted in two problems: the datasets were small, and researchers could not legally test their algorithms against a subset of the final dataset before submission.

2.3 Music Databases

There have been previous attempts to generate datasets intended for MIR research. These efforts have centered on creating copyright-free datasets that could be distributed to researchers.

2.3.1 RWC

The RWC database [4] is a small dataset consisting of original recordings of jazz, classical, and pop music commissioned by the Real World Computing Partnership in Japan. This dataset is small enough that transmission is not an issue, and it is free from copyright restrictions. Unfortunately, there is not enough music present in the dataset to effectively evaluate most MIR algorithms and it does not include the popular music that many researchers are interested in.

2.3.2 Music Audio Benchmark Dataset

This dataset [5] is derived from music freely available from the GarageBand website [6]. Unfortunately, the dataset is still relatively small and also does not cover popular (copyrighted) music.

2.3.3 Magnatune

The Magnatune [7] dataset is a collection of music that can be licensed for research purposes. This dataset is relatively small and consists entirely of music under the Creative Commons license, which effectively excludes most popular music.

2.3.4 Classical Music Archive

The Classical Archive [8] is an online repository of over 38,000 classical music files. This excludes both popular music and any classical music not in the public domain. Furthermore, access to high quality recordings is restricted to those who have paid for a subscription, and limits are

placed on the number of recordings that can be downloaded each day

2.3.5 Variations2

Variations 2 [9] provides access to the digital music repositories of Indiana University's music library. Access to the music within this archive is restricted to Indiana University students and employees.

2.4 Distributed Feature Extraction

A number of other music analysis packages support distributed feature extraction.

2.4.1 M2K

M2K [10] provides an environment where new features can be loaded at startup (by adding class files to its plugins folder), but not during execution. The way in which distributed computation is implemented in M2K/D2K is also somewhat inflexible, and does not incorporate distributed computation according to resource allocation policies. M2K is also built upon D2K, which, unfortunately, is a commercial product.

2.4.2 Marsyas 0.2

Version 0.2 of Marsyas [11] introduces both distributed computing and runtime configuration of features. However, adding new features to Marsyas requires recompilation.

3. Structure

OMEN constructs its dataset by utilizing existing libraries' digitized music collections in order to provide a large and diverse collection of music for analysis. Several requirements influence the overall structure.

As discussed in Section 1, OMEN requires both a central access point for researchers and distributed control nodes for librarians. This mandates that there exist a central server with decentralized computers controlling library resources.

Libraries have computers that are typically idle during off-peak hours and unused when libraries are closed. In order to minimize the computational costs associated with on-demand feature extraction, OMEN utilizes grid computing to take advantage of these under-utilized library computers for the computationally intensive task of feature extraction.

OMEN is split into three distinct kinds of participating computers—a Master Node that coordinates all tasks, Library Nodes that coordinate all tasks for a given digital library, and Worker Nodes that perform feature extraction. Each type of Node communicates with the other types via a collection of web services.

3.1 Master Node

The Master Node is the central point coordinating all aspects of OMEN. It maintains metadata on all the music files currently present in the dataset (across all libraries) and presents this dataset as a single entity to researchers via an interface. The Master Node also hosts the Administrator Interface. In order to accommodate these needs, the Master Node utilizes Tomcat [12] to provide a Servlet/JSP container, MySQL for database support, and Axis [13] to provide web services support. The Master Node provides the following web services to Library Nodes:

- PublishResults: allows Library Nodes to return the results of a feature extraction request to the Master Node.
- NotifyAnalysisFailure: method for Library Nodes to notify the Master Node if feature extraction failed for a given request.
- FileChange.loadFileRecord: method for adding new files to the dataset.
- FileChange.changeFileRecord: method for allowing libraries to change metadata for a file already in the dataset.
- FileChange.deleteFileRecord: method for removing a file from the dataset.
- FileChange.listFileRecord: method for listing all files that a library has contributed to the OMEN dataset.

3.2 Library Nodes

Each digital library that participates in OMEN has one Library Node. Each Library Node maintains on its hard drive the portion of the dataset that this library makes available. When the Master Node delivers a feature extraction request, the Library Node divides the request into pieces, each consisting of one file, and oversees distribution of these pieces to its Worker Nodes. Furthermore, each Library Node includes the Librarian Interface so that librarians can manage each library's participation in OMEN. In order to accommodate these needs, each Library Node utilizes Tomcat to provide a Servlet/JSP container, MySQL for database support, and Axis to provide web services support. The Library Node also provides a collection of web services for both the Master Node and its Worker Nodes:

- ExecuteBatch: this service provides a means for the Master Node to request feature extraction on a number of files located at the library. The list of features to extract and their parameters are passed as arguments.
- AddFeature: the Master Node utilizes this service to disseminate to the Library Nodes new features that have been approved for use by the administrator.
- PublishResults: this service allows Worker Nodes to return to the Library Node the results of their analysis. The results, in either ARFF [14] or ACE XML [15] format, are passed as parameters.

- Music: Worker Nodes utilize this service to retrieve files to be analyzed from the Library Node. Files may be in any of a number of formats, including mp3, wav, aiff, or au. While only the distribution of files stored on the local hard drive is implemented, it is possible to implement a stand in system that instead pulls music files from an existing system provided it implements the same web services as the basic implementation.
- NotifyAnalysisFailure: Worker Nodes utilize this service to notify the Library Node if analysis could not be completed.

3.3 Worker Nodes

Library computers that are provided to patrons for Internet access and searching of bibliographic records are used as Worker Nodes. OMEN utilizes grid computing in order to exploit this under-utilized resource for feature extraction. In order to prevent library patrons from experiencing degraded service, Worker Nodes provide a number of settings that assist in minimizing the impact on patrons' tasks. In order to accommodate these needs, each Worker Node utilizes Axis, within Tomcat, to provide web services support. Feature extraction is performed by jAudio. jAudio was chosen for this task because it is written in Java, is easy to embed in other applications, implements a wide variety of features, and allows OMEN to add new features during execution. Worker Nodes provide the following web services to communicate to their Library Node:

- AddFeature: method for Library Nodes to add new features to jAudio.
- ExecuteBatch: method used to initiate feature extraction on a file.
- ApplySettings: method for configuring the policies for minimizing impact on the Worker Nodes' primary task of serving patrons.

4. Interfaces

In order to be useful to the MIR community, OMEN must provide user-friendly interfaces to each of its different types of users. The Librarian Interface provides a mechanism for librarians to each control their respective library's contributions. The Researcher Interface allows MIR researchers to view the dataset's metadata and extract features from it. Finally, the Administrator Interface permits the overall administrator to maintain and police OMEN.

4.1 Librarian Interface

In order for OMEN to function, librarians must have an easy mechanism for both making their digitized music collection available to the MIR community and controlling OMEN's use of computing resources within the library. The library interface, hosted at each Library Node, accomplishes these tasks.

OMEN provides an interface for allowing librarians to modify the recordings whose features are made available to the public. There are several methods available.

One mechanism to do this is to use iTunes to rip CDs into an archive. The GraceNote CDDB [16] service provides metadata. Ideally, this metadata should be cleaned using software such as jMetaManager [17]. A librarian then uploads the location of the files and the associated metadata into OMEN by uploading the XML document created by iTunes that describes the ripped music.

Alternatively, librarians can use their own favorite program for ripping CDs and enter the metadata in a web-based form. This web form can also be used to modify or delete files from the database.

While the legality of ripping CDs has not been tested in U.S. court, an attorney of the Recording Industry Association of America (the most likely plaintiff) has testified in the U.S. Supreme Court that they believe that ripping CDs is not a copyright infringement [18].

Library Nodes must also control how their Worker Nodes perform the feature extraction. As it is desirable to prevent OMEN from interfering with these computers' primary purpose, librarians are presented with options for accomplishing this goal.

One approach is to set the priority of the computation sufficiently low that it does not interfere with other activities. A second approach is to restrict the time periods during which Worker Nodes can extract features so that they only operate during the hours a library is closed. Since computations generally take at most minutes, Worker Nodes effectively stop computation when the Library Portal stops distributing tasks. Any changes made in these settings are communicated to all Worker Nodes in the library via the ApplySettings web service.

4.2 Researcher Interface

The Researcher Interface is hosted by the Master Node and provides researchers with access to the dataset that is spread across all participating libraries as if it were a single entity. Researchers can choose what subset of the entire dataset to utilize, which features to extract, and the parameters to use for these features. Additionally, researchers can submit new features for use in OMEN.

OMEN provides a search mechanism that allows researchers to search the metadata of all the files in the dataset. Returned results can be refined either by additional searches or by manually pruning the result set. Once the desired dataset has been constructed, the researcher can save this result for future use.

Researchers have two different mechanisms for establishing the parameters to be used in feature extraction. One method is to save a settings file in jAudio and then upload this file in the Researcher Interface. The other is to manually specify the settings via a web interface.

Whichever method is used, researchers can store these settings for later use.

Once a researcher has chosen the query set and feature parameters, an extraction request can then be executed. The Master Node parses the query set, divides the set according to the libraries holding the requested data, and uses the ExecuteBatch service provided by the Library Nodes to begin extraction.

OMEN also serves as a repository for feature extraction algorithms. Researchers are encouraged to submit new features to OMEN so that they can be used by other researchers. In the current version of OMEN, features must be implemented in the Java programming language by sub-classing either the FeatureExtractor or Aggregator abstract classes of jAudio.

4.3 Administrator Interface

The Administrator Interface (located at the Master Node) provides an administrator with the tools needed to maintain OMEN. The administrator adds and removes library accounts using this interface. In addition, the administrator can delete researchers or reset their passwords. Furthermore, this interface allows the administrator to view submitted new features and either approve or reject them.

5. Implementation

Here we describe OMEN by following the execution path of several example tasks.

5.1 Adding Files to OMEN

A librarian adds music files to the system via one of two methods: by uploading an iTunes XML file or by adding the files manually and then using the Librarian Interface to add the metadata. In either case, the Library Node uploads the metadata about the files to the Master Node.

Upon receiving the metadata, the Master Node adds this metadata to its database. During this process, the Master Node generates a list of unique IDs for the files added, uniquely identifying them in OMEN.

This list of IDs is returned to the Library Node that added the files. These IDs are then linked to the location of the file and stored in the database locally.

5.2 Feature Extraction

Feature extraction is initiated in the Researcher Interface at the Master Node. The researcher provides the list of files to be analyzed and the feature settings to be used.

First, the request is analyzed and altered as necessary to avoid violating copyright law. The request is then split into a separate request for each library that has files in the dataset, identifying each music file by the unique ID it was given when it was added to OMEN. The request is added to the list of outstanding requests for this researcher. The outstanding request is then added to the Researcher Interface. Then the request is sent to each Library Node via the ExecuteBatch web service.

The Library Node parses the requests into numerous small requests of one file each. These requests are queued internally. As Worker Nodes become available, the requests are dispatched using each Worker Node's ExecuteBatch web service.

Each Worker Node that receives a request first checks to see if it has the file requested in its internal cache. If not, the Worker Node calls the Music web service at the Library Node to download it. Feature extraction is performed using jAudio and the results are returned to the Library Node via its PublishResults web service.

The Library Node collects all the results from the Worker Nodes. Once the results are all collected, it relays the extracted features back to the Master Node via its PublishResults web service.

As Library Nodes return results, the Researcher Interface is updated to reflect ongoing progress. Once all results are returned, they are made available through the Researcher Interface as either Weka ARFF [14] or ACE XML files [15].

5.3 Adding a New Feature

A researcher can add a new feature by submitting Java source code through the Researcher Interface. The code is then made available to the administrator through the Administrator Interface. The administrator then checks the code for security. Once the code is cleared, the administrator compiles the code and submits the compiled code for distribution. The Master Node distributes this code to each Library Node via the AddFeature web service and updates its internal list of features.

Each Library Node takes the incoming compiled code and then redistributes this code to its Worker Nodes via their AddFeature web service.

Each Worker Node updates its list of features with the new feature and adds the compiled code to the directory that jAudio automatically searches for new features.

6. Example Dataset

The functionality of OMEN has been successfully tested with two different music libraries, namely the Codaich database [17] (16 803 songs totaling 80 GB) and a private CD collection (577 songs totaling 3 GB).

7. Conclusions and Future Work

OMEN provides a mechanism for legally providing all MIR researchers with access to a large dataset of music. This will permit researchers to compare results of MIR algorithms using the same music at any time, not just during specific competitions. It will also save researchers significant amounts of time by removing the need to construct their own datasets or resort to non-representative music in the public domain.

However, there is still work to be done. In particular, the system requires additional security to prevent unauthorized access of OMEN resources. This includes functionality such as digital signatures for authenticating communications between nodes.

Another area for improvement is the scheduling algorithm for the Library Nodes. The required bandwidth could be greatly reduced if more care were taken to distribute feature extraction requests to Worker Nodes that already have needed files in their caches.

8. Acknowledgments

We would like to thank the Social Sciences and Humanities Research Council of Canada for their financial support and the Canada Foundation for Innovation for supplying the equipment used in this project.

References

[1] J. S. Downie, K. West, A. Ehmann, and E. Vincent, "The 2005 Music information retrieval evaluation exchange (MIREX 2005): Preliminary overview," in *Proceedings of the International Conference on Music Information Retrieval*, 2005, pp. 320–323.

[2] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle, "jAudio: A feature extraction library," in *Proceedings of the International Conference on Music Information Retrieval,* 2005, pp. 600–603.

[3] D. McEnnis and I. Fujinaga. "jAudio: Additions and Improvements," in *Proceedings of the International Conference on Music Information Retrieval,* 2006 (in press).

[4] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *Proceedings of the International Conference on Music Information Retrieval*, 2002, pp. 287–288.

[5] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. "A benchmark dataset for audio classification and clustering," in *Proceedings of the International Conference on Music Information Retreival,* 2005, pp. 528–531.

[6] "GarageBand.com," [Web site] 2006, [2006 March 27], Available: http://www.garageband.com

[7] "Magnatune: MP3 music and music licensing," [Web Site] 2006, [2006 March 27], Available: http://magnatune.com

[8] "Classical Music Archives," [Web Site] 2006, [2006 July 6], Available: http://www.classicalarchives.com

[9] J. W. Dunn and M. Notess. "Variations 2: The Indiana University Digital Music Library Project" in *Proceedings of Digital Library Federation Fall Forum*, 2002.

[10] J. S. Downie, J. Futrelle, and D. Cheng. "The international music information retrieval systems evaluation laboratory," in *Proceedings of the International Conference on Music Information Retreival*, 2004, pp. 9–14.

[11] S. Bray and G. Tzanetakis. "Distributed audio feature extraction for music," in *Proceedings of the International Conference on Music Information Retrieval,* 2005, pp. 434–437.

[12] "Apache Tomcat," [Web site] 2006, [2006 April 14], Available: http://tomcat.apache.org

[13] "Web Services – Axis," [Web site] 2006, [2006 April 14], Available: http://ws.apache.org/axis

[14] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed., New York: Morgan Kaufman, 2005.

[15] C. McKay, D. McEnnis, R. Fiebrink, and I. Fujinaga. "ACE: A general purpose classification ensemble optimization framework," in *Proceedings of the International Conference on Music Information Retrieval,* 2005, pp. 161–164.

[16] "Gracenote," [Web site] 2006, [2006 March 29], Available: http://www.gracenote.com

[17] C. McKay, D. McEnnis, and I. Fujinaga. "A large publicly accessible annotated audio music research database," in *Proceedings of the International Conference on Music Information Retrieval*, 2006 (in press).

[18] "Supreme Court of the United States: Argument Transcripts" [Web site] 2006, [2006 July 1], Available: http://www.supremecourtus.gov/oral_arguments/argument_transcripts/04-480.pdf