

# An Architecture Proposal for Musical Applications and Services on Computational Clouds

Ernesto Trajano de Lima  
Federal University of Ceará  
Virtual University Institute  
(UFC VIRTUAL)  
Fortaleza, Ceará, Brazil  
ernesto@virtual.ufc.br

Emanuel Ferreira  
Coutinho  
Federal University of Ceará  
Master and Doctorate in  
Computer Science (MDCC)  
Fortaleza, Ceará, Brazil  
emanuel@virtual.ufc.br

Gabriel Antoine Louis  
Paillard  
Federal University of Ceará  
Virtual University Institute  
(UFC VIRTUAL)  
Fortaleza, Ceará, Brazil  
gabriel@virtual.ufc.br

Leonardo de Oliveira  
Moreira  
Federal University of Ceará  
Virtual University Institute  
(UFC VIRTUAL)  
Fortaleza, Ceará, Brazil  
leomoreira@virtual.ufc.br

## ABSTRACT

Nowadays cloud computing continues its consolidation, both for technologies virtualization as for platforms development and application availability. These environments are naturally distributed and constituted by many types of software and hardware features, serving a broad range of customers with different requirements. A type of service that has had its use increasingly demanded is the processing of multimedia applications, such as audio and video. This type of service is characterized by different QoS, network and devices requirements, which fits well to cloud computing environments. Many applications or music service providers are available, both for web applications and mobile devices, such as Spotify, Deezer and Last.fm. Non-exclusive applications for music, but also videos, like YouTube and Netflix, are also widely employed by users of multimedia services. This paper describes and discusses an architecture for the implementation and development of computer music services in a cloud computing environment.

## Categories and Subject Descriptors

C.0 [General]: [Modeling of computer architecture, System architectures]; H.5.5 [Sound and Music Computing]: [Methodologies and techniques, Modeling, Systems]

## General Terms

Management

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EATIS '16 April 27 - 29 2016, Cartagena de Indias, Bolivar, Colombia  
IEEE 978-1-5090-2435-3/16/\$31.00 ©2016 IEEE.

## Keywords

Cloud Computing, Computer Music, Architecture, PaaS, Multimedia

## 1. INTRODUÇÃO

Computação em nuvem é um paradigma tecnológico de utilização de recursos computacionais sob demanda, segundo o qual infraestrutura de *hardware*, *software* e plataformas para o desenvolvimento de novas aplicações são oferecidos como serviços remotos e em escala global [9].

As nuvens computacionais permitem que as aplicações especifiquem um modelo de Qualidade de Serviço (QoS) baseado em um Acordo de Nível de Serviço (SLA) [5]. Um SLA é um contrato entre aplicação e provedor de nuvem, onde o provedor deve se ajustar para garantir o modelo de qualidade especificado para a aplicação.

Um tipo de serviço que está cada vez mais sendo utilizado é o processamento de aplicações multimídia, como áudio e vídeo. Esse tipo de serviço se caracteriza por diferentes necessidades de QoS, rede e dispositivos, justificando o uso das característica de um ambiente de nuvem computacional. Atualmente, o processamento da dados multimídia em nuvens computacionais impõe grandes desafios [10], tais como:

- Multimídia e heterogeneidade de serviços: a nuvem deve apoiar diferentes tipos de serviços multimídia para diferentes tipos de usuários simultaneamente;
- Heterogeneidade de QoS: a nuvem deve fornecer QoS e suporte para vários tipos de serviços multimídia de forma a atender diferentes requisitos de QoS;
- Heterogeneidade de rede: a nuvem deve adaptar conteúdos multimídia para entrega otimizada em diferentes tipos de dispositivos, com diferentes larguras de banda de rede e latências.
- Heterogeneidade de dispositivos: a nuvem deve ter capacidades de adaptação multimídia para atender diferentes tipos de dispositivos, incluindo CPU, GPU,

vídeo, memória, armazenamento e consumo de energia.

Nesse contexto, aplicações musicais se encaixam bem, pois também são aplicações multimídia, possuindo as mesmas características de heterogeneidade descritas em [10]: serviços, QoS, rede e dispositivos.

Este trabalho tem como objetivo propor uma arquitetura para a execução ou desenvolvimento de serviços de computação musical, em um ambiente de computação em nuvem. Dessa maneira, com a instanciação dessa arquitetura será possível a efetivação de diversas tarefas, tais como: execução de *benchmarks*, coleta de dados, integração com outras plataformas, consolidação e armazenamento de dados.

O trabalho está organizado da seguinte forma: a seção 2 discute a proposta arquitetural deste trabalho, destacando seus componentes, funcionalidades e os fluxos de execução; já a seção 4 apresenta e discute as perspectivas de pesquisa envolvendo o âmbito do trabalho; por fim, a seção 5 contém as conclusões e os direcionamentos futuros deste trabalho.

## 2. PROPOSTA DE ARQUITETURA

Nesta seção uma arquitetura para execução de serviços de aplicações musicais nos ambientes de computação em nuvem é descrita. Seus componentes estão dispostos na Figura 1, assim como seus relacionamentos. Cada componente possui suas próprias características a serem descritas nas subseções a seguir.

### 2.1 API-PAAS

Este é o componente principal da arquitetura proposta e que é responsável pelo provimento de uma interface para a construção de aplicações ou serviços, disponibilizados em uma nuvem computacional, de serviços musicais. Sua intenção é possibilitar uma Plataforma como Serviço (PaaS) para serviços musicais. Esta plataforma permite a utilização de diversos modelos de implantação de nuvens computacionais (privada, pública e híbrida), e suas possíveis combinações para a alocação de recursos.

### 2.2 Composição

Componente responsável pela composição de músicas utilizando trechos ou músicas completas. Assim, é possível criar *mashups*<sup>1</sup>. A este componente podem ser adicionados, ainda, recursos capazes de auxiliar o compositor na criação de novas músicas como, por exemplo, bibliotecas de auxílio à composição ou ambientes para a composição em tempo real.

### 2.3 Comparação

Componente responsável pela comparação entre músicas ou trechos de músicas, considerando diversos elementos, tais como: gênero musical, intérprete, instrumentos e ritmo. Este componente deve permitir que diversas medidas de similaridade musical possam ser utilizadas durante a comparação.

### 2.4 Recuperação

Componente responsável pela extração e inferência de propriedades e características da música. De posse do dado musical extraído, este componente deve representá-lo, indexá-lo

<sup>1</sup>*Mashup* é uma composição musical que resultada da mistura de músicas preexistentes.

e armazená-lo adequadamente para posterior recuperação. Como no caso do componente de Comparação, aqui também devem existir a possibilidade de alteração da medida de similaridade musical utilizada, uma vez que resultados diferentes podem ser obtidos.

### 2.5 Visualização

Componente responsável pela visualização dos dados e informações musicais. As formas de visualização podem ocorrer de diversas maneiras, sejam tabelas ou gráficos, e podem ser aplicadas às mais diversas áreas. Diversos autores na literatura discutem profundamente técnicas de visualização de dados, seja do ponto de vista científico, seja do ponto de vista de *design* da informação [3, 1].

### 2.6 Atualização

Componente responsável por atualizar dados de elementos musicais, tais como: gênero, autor, intérprete e instrumentação. *A priori*, a atualização se dará apenas nos metadados das músicas, uma vez que a atualização ou modificação de elementos da estrutura da música podem alterá-la por completo.

### 2.7 Processamento

Componente responsável por processar informações em um trecho musical. Esse processamento pode ser uma busca por informações mais complexa, pode ser a identificação de trechos por similaridade musical, etc.

### 2.8 Alocador

O Alocador tem como aspecto principal possibilitar a instanciação de serviços ou processos nas instâncias da plataforma nas nuvens. Portanto, ele é responsável por manter as credenciais de acesso, criação e manutenção e dos usuários, criação de máquinas virtuais e a implementação de componentes adequados e serviços na nuvem. Além disso, também é responsável pela alocação de operações nas configurações da nuvem.

Uma vez que esse componente está em uma nuvem computacional, todos os benefícios agregados da computação em nuvem poderão estar disponíveis para a plataforma proposta pela arquitetura, tais como disponibilidade, medição, QoS e elasticidade.

### 2.9 Comunicação

O componente Comunicação é responsável pela transferência de informação e dados entre as instâncias da plataforma proposta neste trabalho. A plataforma proposta neste trabalho pode estar contida em diferentes nuvens computacionais. Portanto, o componente Comunicação realiza toda a troca de mensagens entre essas diversas nuvens que contém a plataforma descrita neste trabalho.

### 2.10 Banco de Dados Pessoal

Banco de dados que armazena as informações pessoais do usuário e configurações de utilização da API. Este banco de dados é utilizado para o armazenamento de dados das aplicações dos usuários. É comum que as aplicações armazenem dados de configurações pessoais para modificação posterior ou adaptação às necessidades do usuário.

Em adição, este banco de dados pode também, armazenar dados históricos relacionados à utilização do ambiente, aplicação ou API. Adicionalmente, com esses dados é possível

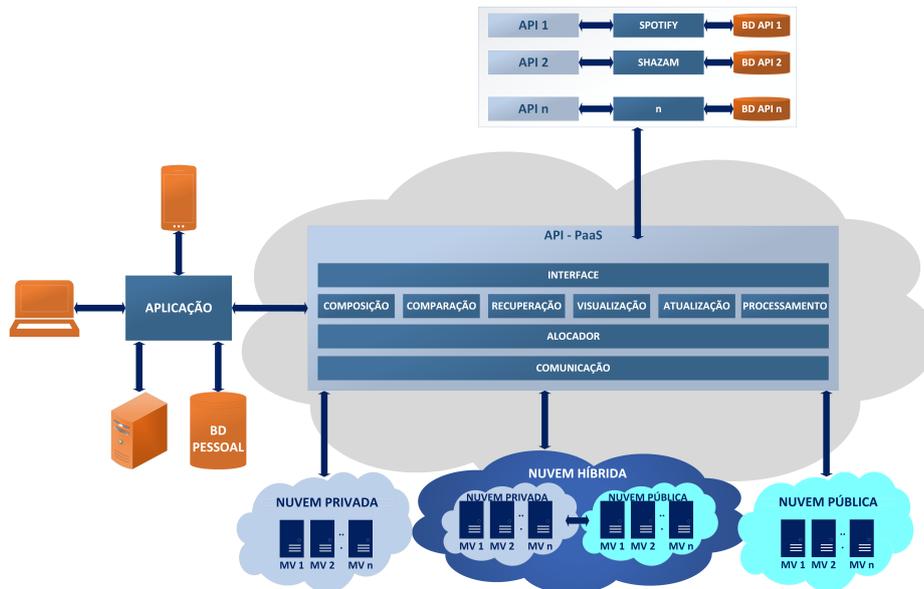


Figure 1: Visão Geral da Arquitetura Proposta

monitorar o consumo dos recursos, executar algum tipo de auditoria no ambiente, e assim possibilitar a recomendação baseada na utilização da aplicação pelo usuário, tanto do ponto de vista de serviços quanto de infraestrutura. Esse banco de dados pode estar disponível localmente ou remotamente em um nuvem computacional.

### 2.11 Aplicação do Usuário

Aplicação a ser desenvolvida com a utilização da API-PAAS. Essa aplicação ou serviço pode usufruir das diversas operações disponibilizadas pela API, e pode ser construída em diferentes plataformas, como dispositivos móveis, computadores pessoais e aplicações *desktop* ou remotas.

Uma vez que a aplicação implemente as funcionalidades do API-PAAS, o desenvolvedor/usuário. Além disso, um provedor de serviços pode também criar suas próprias aplicações e prover diferentes serviços aos usuários. Por meio desse componente, é possível desenvolver serviços de *Software as a Service* (SaaS) para o acesso a serviços musicais.

### 2.12 APIs

O componente arquitetural API está diretamente associado às APIs de outros serviços, como o Spotify, Deezer e Last.fm. Diversos serviços musicais possuem APIs para o desenvolvimento de aplicações independentes, muitas vezes oferecendo apenas ao usuário logado. Em geral, essas APIs oferecem os serviços do aplicativo disponibilizado como serviço, e assim possibilitando que desenvolvedores construam suas próprias aplicações ou integrem serviços de terceiros.

Esses serviços/provedores de serviços musicais também possuem sua base de dados, que podem ser acessadas por meio de API própria. Em relação à arquitetura proposta, diversos componentes possuem a mesma funcionalidade dessas APIs, e a proposta do componente API é permitir uma interface entre as APIs disponíveis no mercado e a arquitetura proposta, ou seja, possibilitar uma interface comum.

### 2.13 Banco de Dados Externos

Serviços como o Spotify, Deezer e Last.fm possuem suas próprias bases de dados, acessíveis por meio de suas aplicações ou de APIs disponibilizadas para a comunidade. Essas bases de dados contêm diversas informações musicais, que podem ser recuperadas e utilizadas para as mais diversas finalidades.

### 2.14 Aplicações e Sistemas de Terceiros

Diversas aplicações/provedores de serviços musicais estão disponíveis. Muitos desses serviços são pagos, e alguns possuem APIs para suportar o desenvolvimento de aplicações. Exemplos desses serviços são o Spotify, Deezer e Last.fm.

## 3. EXEMPLO DE APLICAÇÃO DA ARQUITETURA

Considere a oportunidade de se desenvolver uma aplicação a ser executada em um servidor *web* hospedado em uma rede local para a disponibilização de *playlists* e de composições de músicas do usuário. Essa composição musical consiste de trechos musicais coletados e combinados conforme a necessidade do usuário, e não à criação de uma música inédita. Ela seria, portanto, o equivalente a um *mashup*. Considere também que, além de um computador *desktop*, o usuário também utilizará a aplicação através de dispositivos móveis em Android.

Para isto, por exemplo, um usuário deverá acessar os serviços da plataforma (API-PaaS) disponíveis previamente em uma máquina virtual instalada em uma nuvem pública. A interface da plataforma disponibiliza recuperação da informação de músicas, que podem ser obtidas de diferentes fontes. No caso, a aplicação se integra com serviços disponibilizados pelo Spotify, assim como com serviços existentes na plataforma do Last.fm. Ambas aplicações possuem APIs para serem utilizadas por desenvolvedores. Sendo assim, de posse de dados de músicas previamente selecionadas e recuperadas, o usuário pode iniciar o processo de composição de sua música, baseado em um processo de combinação de trechos musicais. Uma vez finalizado, o aplicativo possui uma fun-

cionalidade de disponibilização da *playlist* do usuário, na forma de um vídeo. Essa disponibilização é realizada pelo componente alocador, que utiliza os recursos de comunicação de rede disponibilizados pelo componente de comunicação. Além disso, um serviço particular, onde qualquer pessoa com acesso a uma aplicação hospedada em uma máquina virtual em uma nuvem privada construída com o OpenStack pode acessar. O banco de dados pessoal, configurado localmente, armazena dados pessoais da aplicação e do usuário. Essa aplicação particular é mais robusta, possuindo serviços de administração e visualização dos dados musicais em um nível mais baixo e detalhado das informações.

## 4. PERSPECTIVAS DE PESQUISA

Nesta seção são apresentadas algumas oportunidades de pesquisa onde a arquitetura aqui proposta pode ser aplicada. Não necessariamente todos os componentes da arquitetura são utilizados, podendo as oportunidades de pesquisa ser uma expansão das funcionalidades propostas pela arquitetura ou integração com outras aplicações.

### 4.1 Infraestruturas para Benchmark de Algoritmos de MIR

A Sociedade Internacional de Recuperação da Informação Musical (ISMIR, na sigla em inglês) abriga durante sua conferência anual o MIREX—*Music Information Retrieval Evaluation eXchange*<sup>2</sup>. Nesta espécie de competição, pesquisadores inscrevem algoritmos que realizam diferentes tarefas da área de recuperação da informação musical, como classificação automática de gênero musical, identificação automática da pulsação, extração de melodia a partir arquivos áudio, entre outros.

Todos os algoritmos são executados processando os mesmos conjuntos de dados. Os modelos construídos são, então, aplicados a dados previamente preparados por especialistas (*ground truth*). Os resultados obtidos são, então, tabulados, ranqueados e disponibilizados para a comunidade científica. Mais que uma competição, o MIREX constitui-se em uma oportunidade dos pesquisadores compararem o desempenho de suas soluções junto às de seus pares, ou seja, o MIREX possibilita a realização de *benchmarks* de diversos algoritmos de recuperação da informação musical.

Uma limitação do MIREX é que ele é um evento anual, ou seja, um pesquisador que tenha desenvolvido um novo método de identificação de tonalidade, por exemplo, não pode, a qualquer tempo, comparar sua técnica com a de outros pesquisadores. É necessário esperar pela abertura de submissões para que uma técnica possa ser comparada com outra. Considerando a adoção da infraestrutura em nuvem aqui proposta pela organização da competição, este pesquisador teria em mãos uma excelente ambiente de testes.

### 4.2 Algoritmos de Recuperação de Informações Musicais

Se por um lado a construção de uma infraestrutura como a que é apresentada neste artigo pode ensejar avanços para os pesquisadores em recuperação da informação musical (seção 4.1), esta mesma infraestrutura também pode ajudar desenvolvedores de serviços de música e, conseqüentemente, usuários destes serviços, em tarefas como recuperação de trechos, identificação de estilos musicais, identificação de artis-

<sup>2</sup>[http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

tas, construção de *playlists*, comparação de trechos musicais, entre outras.

Com acesso aos diversos serviços apresentados na seção 2, desenvolvedores podem utilizar algoritmos que representam o estado da arte em recuperação de informações musicais. Além disso, o serviço pode apresentar melhoras sem a necessidade de intervenção do desenvolvedor, já que, quando um novo algoritmo com melhor performance é desenvolvido, ele pode ser empregado sem a necessidade de modificação da aplicação que utiliza o serviço.

A arquitetura proposta é aplicada a uma nuvem computacional. Então, todos os benefícios agregados da computação em nuvem podem estar disponíveis para a plataforma proposta pela arquitetura, tais como disponibilidade, medição, QoS e elasticidade.

### 4.3 Composição de Músicas

Em certos gêneros musicais, como a música clássica, por exemplo, a composição de uma música é um ato solitário. O compositor, a partir de suas próprias ideias e técnicas composicionais, desenvolve estruturas musicais que se transformam em uma obra musical. Existem, porém, vários outros gêneros musicais onde vários compositores, em conjunto, organizam e desenvolvem as estruturas musicais que formam uma música. Blues, Rock, Jazz, Samba, só para citar alguns exemplos, são estilos onde vários compositores se reúnem para compor músicas.

Compositores que, colaborativamente, precisem compor uma nova música, poderiam utilizar uma ferramenta de composição construída sobre um ambiente em nuvem como o proposto neste artigo. Alguns ganhos que um ambiente como este poderia trazer são a possibilidade de trabalhar na mesma música, mesmo estando geograficamente deslocados ou ainda em momentos diferentes; poderiam ter acesso a um histórico de modificações que a obra sofreu durante seu processo de composição; ou, ainda, incorporar trechos musicais já prontos que se encontrem armazenados na nuvem (desde que direitos autorais não sejam infringidos).

Outros ganhos que compositores poderiam ter ao utilizar um ambiente como o aqui descrito é na possibilidade de utilização de algoritmos de recuperação de informação musical para obter trechos que satisfaçam certas características ou mesmo utilizar algum sistemas de regras que validem aspectos considerados estilisticamente adequados<sup>3</sup>.

### 4.4 Live Coding

*Live coding* [2] é uma prática de programação baseada na implementação em tempo real de códigos que criam sons, músicas e/ou imagens utilizando algum tipo de linguagem de programação interativa, técnicas de composição algorítmica e de improvisação. O objetivo dos praticantes é, através do computador, obter novas formas de expressão musical.

Neste contexto, um serviço de nuvem computacional como o aqui proposto poderia servir como um ambiente colaborativo, onde as composições poderiam ser criadas, modificadas e adaptadas por diferentes programadores ou compositores, incluindo os que estejam geograficamente dispersos. Um ambiente como este poderia servir, portanto, para ampliar ainda mais as formas de expressão musical, tão caras

<sup>3</sup>O software *OhmStudio* (<http://www.ohmstudio.com>) implementa aspectos de composição colaborativa, mas não existe nenhum tipo de sistema de recomendação ou verificação de restrições musicais como os aqui mencionados.

aos praticantes do *live coding*.

## 4.5 Mineração de Dados Musicais

Em [8], cap. 4, os autores descrevem sistemas de recuperação de informações musicais sensíveis ao contexto, ou seja, sistemas que se adaptam dinamicamente ao ambiente do usuário ou ao seus estados afetivos e cognitivos. Para realizar este tipo de análise e identificar padrões e informações que possam ser utilizados na identificação de estados afetivos ou cognitivos, assim como do ambiente e das preferências do usuário é necessário uma quantidade significativa de dados.

Considerando este tipo de pesquisa, uma infraestrutura em nuvem que continuamente colete, consolide e armazene dados de serviços como Twitter, Facebook, Shazam, Rdio, Deezer, Last.fm, etc. seria de providencial utilidade. A partir dos dados coletados, seria possível, com o uso de técnicas de mineração de dados, encontrar regularidades, preferências, padrões de escuta, etc. Ou seja, seria possível uma efetiva pesquisa sobre recuperação de informações musicais sensíveis ao contexto.

Os próprios algoritmos de mineração poderiam se utilizar da infraestrutura em nuvem para executar. Perguntas de pesquisa como “qual a relação entre quantidade de dados e a acurácia de classificador?” ou “grandes quantidades de dados implicam necessariamente em melhores predições?” poderiam ser investigadas já que seria possível através da infraestrutura em nuvem tanto coletar uma grande quantidade de dados quanto executar regularmente os algoritmos de mineração.

## 4.6 Visualização de Dados Musicais

Apesar de ser um fenômeno sonoro, vários aspectos da música são descritos através de algum tipo de meio visual. Tais aspectos são os mais variados como, por exemplo, estruturas formais da música [6], tonalidade [4], gênero musical<sup>4</sup> e até memos coleções de arquivos musicais [7].

No contexto particular da arquitetura aqui proposta e considerando o apresentado na seção 4.5, serviços em nuvem podem prover diferentes formas de visualização, tanto dos dados brutos quanto dos resultados obtidos com o uso técnicas de mineração de dados. Tais visualizações podem ajudar a promover diferentes *insights* sobre variados aspectos da música, sejam eles aspectos pertencentes à própria música (relações estruturais, por exemplo) ou aspectos alheios à música, mas influenciados por ela (padrões de escuta por regiões de uma cidade ou estado, por exemplo).

Esses serviços visuais podem ser desenvolvidos como componentes que utilizam as funções disponibilizadas na própria plataforma baseada na arquitetura proposta, e assim compõem aplicações das mais diversas formas de interação com usuário: apenas consulta, atualização de dados, visualização de dados, comparação entre diferentes aspectos musicais, visualização no tempo ou estática.

## 5. CONCLUSÃO

Este trabalho propôs uma arquitetura genérica para a criação e utilização de serviços musicais em ambientes de nuvens computacionais. Além da descrição dos componentes que integram a arquitetura, foram descritas várias oportuni-

dades de pesquisa listadas nesse trabalho que motivam tanto a utilização da arquitetura proposta quanto sua expansão e melhoria, incluindo aí a possibilidade de disponibilização de outros serviços para os usuários.

Como trabalhos futuros, temos a intenção de efetivar as oportunidades de pesquisa listadas nesse trabalho na forma de aplicações e serviços reais providos pela arquitetura proposta. Além disso, aspectos diretamente relacionados à computação em nuvem, como serviço medido e elasticidade poderão ser bastante utilizados em conjunto com aplicações musicais, especialmente quando estas requererem muitos recursos e possuírem muitos usuários simultâneos, e, assim, termos um desempenho melhor. Por fim, aspectos relacionados a direitos autorais e ao sigilo de informações precisam ser levados em consideração e adequadamente tratados pelos componentes da arquitetura.

## 6. REFERÊNCIAS

- [1] M. Bostock, V. Ogievetsky, and J. Heer. D3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, Dec. 2011.
- [2] N. Collins, A. McLean, J. Rohrerhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8:321–330, 12 2003.
- [3] M. de Oliveira and H. Levkowitz. From visual data exploration to visual data mining: a survey. *Visualization and Computer Graphics, IEEE Transactions on*, 9(3):378–394, July 2003.
- [4] A. Mardirossian and E. Chew. Visualizing music: Tonal progressions and distributions. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 189–94, Vienna, Austria, 2007.
- [5] L. O. Moreira, F. R. C. Sousa, J. G. R. Maia, V. A. E. Farias, G. A. C. Santos, and J. C. Machado. A live migration approach for multi-tenant rdbms in the cloud. In *28th Brazilian Symposium on Databases, SBBD '13*, pages 73–78, 2013.
- [6] M. Müller and N. Jiang. A scape plot representation for visualizing repetitive structures of music recordings. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 97–102, Porto, Portugal, 2012.
- [7] E. Pampalk, A. Rauber, and D. Merkl. Content-based organization and visualization of music archives. In *Proceedings of the Tenth ACM International Conference on Multimedia, MULTIMEDIA '02*, pages 570–579, New York, NY, USA, 2002. ACM.
- [8] M. Schedl, E. Gómez, and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends<sup>®</sup> in Information Retrieval*, 8(2-3):127–261, 2014.
- [9] F. R. Sousa, L. O. Moreira, and J. C. Machado. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. In *III Escola Regional de Computação Ceará, Maranhão e Piauí ERCEMAPI*, pages 150–175. EDUFPI, 2009.
- [10] W. Zhu, C. Luo, J. Wang, and S. Li. Multimedia cloud computing. *Signal Processing Magazine, IEEE*, 28(3):59–69, May 2011.

<sup>4</sup>Vide <https://research.google.com/bigpicture/music/#>.