Alisson de Villa Geronimo, Matheus Medeiros Anacleto

UM SISTEMA DE RECOMENDAÇÃO DE CONTEÚDO SUPORTADO PELA COMPUTAÇÃO DISTRIBUÍDA

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Tecnologias da Informação e Comunicação.
Orientador: Prof. Dr. Alexandre Leopoldo Gonçalves.

Araranguá
2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

, Geronimo, Alisson de Villa; Anacleto, Matheus Medeiros
UM SISTEMA DE RECOMENDAÇÃO DE CONTEÚDO SUPORTADO PELA COMPUTAÇÃO DISTRIBUÍDA / Geronimo, Alisson de Villa; Anacleto, Matheus Medeiros ; orientador, Alexandre Leopoldo Gonçalves - Florianópolis, SC, 2014.
87 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Araranguá. Graduação em Tecnologias da Informação e Comunicação.

Inclui referências

1. Tecnologias da Informação e Comunicação. 2. Tecnologias da Informação e Comunicação. 3. Sistemas de Recomendação. 4. Computação Distribuída. 5. Web 2.0. I. Gonçalves, Alexandre Leopoldo. II. Universidade Federal de Santa Catarina. Graduação em Tecnologias da Informação e Comunicação. III. Título.

Alisson de Villa Geronimo, Matheus Medeiros Anacleto

UM SISTEMA DE RECOMENDAÇÃO DE CONTEÚDO SUPORTADO PELA COMPUTAÇÃO DISTRIBUÍDA

Esta Monografia foi julgada adequada para obtenção do Título de "Bacharel em Tecnologias da Informação e Comunicação", e aprovada em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, 14 de Julho de 2014.

Prof. Vilson Gruber, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Alexandre Leopoldo Gonçalves, Dr.
(Orientador)

Prof. Juarez Bento da Silva, Dr.

Profa. Olga Yevseyeva, Dra.

Este trabalho é dedicado a todos que direta ou indiretamente contribuíram em nossa formação acadêmica.

AGRADECIMENTOS

Gostaríamos de agradecer a todos os professores do curso de Tecnologias de Informação e Comunicação da Universidade Federal de Santa Catarina. Em especial ao nosso orientador o Prof. Dr. Alexandre Leopoldo Gonçalves pela sua colaboração em nossas formações acadêmicas. Tanto neste trabalho com sua dedicação e paciência, assim como em todas as matérias lecionadas durante o curso. Agradecemos aos colegas de curso que de uma forma ou de outra contribuíram para que pudéssemos atingir nosso objetivo. Agradecemos também a todos os demais servidores da UFSC pelos serviços prestados.

Gostaria de agradecer primeiramente a Deus pelo dom da vida que me concedeu e por sempre estar ao meu lado me iluminando e guiando meus passos. Agradeço aos meus pais e a meu irmão pelos ensinamentos que me apresentaram e tem me apresentado por todos esses anos, com dedicação, forte incentivo e por estarem sempre ao meu lado. Agradeço a minha avó paterna Terezinha Maria Elias João Geronimo que se mostrou sempre disposta a me ajudar e apoiar, fornecendo conselhos e ensinamentos ao longo de minha vida. Agradeço ao meus tios Olacir e Maria Aparecida por sempre estarem me incentivando, me apoiando e contribuindo com meu ensino. Por fim, não menos importante, agradeço a todos meus amigos e familiares que sempre me ajudaram, me apoiaram e contribuíram em toda a minha trajetória.

Alisson de Villa Geronimo

Gostaria de agradecer primeiramente a Deus por se fazer presente desde o ventre da minha mãe e por colocar ao meu lado pessoas fundamenteis para alcançar esta vitória. Dedico este trabalho a ele. Agradeço aos meus pais pelo cuidado com a minha vida e por todas as noites em claro cuidando de mim. Agradeço a minha esposa Heloisa de Souza Brillinger Anacleto pela sua infinita paciência e por todo o incentivo que me foi dado para fazer o vestibular e também em todos os dias desta jornada, não me deixando desistir nos momentos mais críticos. Agradeço aos meus pastores Vladimir Bento Motta e Vandir Cardoso Motta pelo encorajamento diário. Por fim, mas não menos importante agradeço a todos os meus amigos pelas palavras de incentivo e pela compreensão quanto a minha ausência em todos os ensaios e momentos de diversão que estivemos distantes.

Matheus Medeiros Anacleto

“Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá!”

Ayrton Senna da Silva

RESUMO

Desde a sua criação, a *Internet* e mais especificamente a Web, vem passando por grandes modificações. Atualmente, usuários possuem um papel fundamental, não somente consumindo informações, mas também provendo novos conteúdos. Este cenário e os avanços da Tecnologia da Informação tem promovido um aumento vertiginoso no volume de informações disponíveis. A partir disto surgem desafios, entre eles, como permitir que o usuário realize escolhas mais adequadas. Neste contexto, encontram-se os Sistemas de Recomendação com o intuito de auxiliar usuários na tomada de decisão, bem como, a Computação Distribuída como infraestrutura de base para lidar com grandes volumes de informação. A partir disto, o presente trabalho propõe um sistema voltado à recomendação de conteúdo textual através das abordagens de filtragem colaborativa e baseada em conteúdo. Visando permitir a avaliação da proposição deste trabalho foi elaborado um modelo de dados e desenvolvido um protótipo. O protótipo possibilita a geração de informações nas duas principais abordagens de recomendação. Possui ainda a capacidade de realizar o processamento de maneira distribuída. As informações processadas e geradas através da aplicação do protótipo permitem a sugestão de itens, em que no presente trabalho se referem a documentos. Pode-se afirmar que os resultados no que tange a sugestão de conteúdo são consistentes e compatíveis com a literatura da área de Sistemas de Recomendação. Ressalta-se ainda que o desenvolvimento de sistemas distribuídos contribui para área em questão visto que o desempenho frente a grande volumes de informação é fundamental para que se possa produzir insumos que auxiliem usuários em suas escolhas.

Palavras-chave: Sistemas de Recomendação; Computação Distribuída; Filtragem Colaborativa; Recomendação Baseada em Conteúdo.

ABSTRACT

Since its creation the Internet and more specifically the Web has changed dramatically. Nowadays, users have a key role not only consuming information but also providing new content. This scenario and the advances in Information Technology have fostered the increase in the volume of information available. From this challenges arise, among them, how to allow users to perform more appropriate choices. In this context, there are the Recommender Systems in order to aid users in decision making and Distributed Computing as the base infrastructure to handle large volumes of information. From this, the present work proposes a system towards recommendation of textual content through collaborative filtering and content-based approaches. To allow the evaluation of the proposition a data model has been designed as well as has been developed a prototype. The prototype enables the generation of information on the two major recommendation approaches. It also has the ability to carry out the processing in a distributed manner. The information generated and processed by the prototype allows the suggestion of items which in the present study refers to documents. It can be stated that the results regarding the suggested content are consistent and compatible with the literature in the area of Recommender Systems. It is noteworthy that the development of distributed systems contributes to the area in question since performance against large volumes of information is crucial in order to produce products that can assist users in their choice.

Keywords: Recommender Systems; Distributed Computing; Collaborative Filtering; Content-based Recommendation.

LISTA DE FIGURAS

Figura 1 - Comparação entre as classificações de Teo e outros usuários.....34
Figura 2 - Filtragem híbrida.....39
Figura 3 - Arquitetura cliente-servidor. 44
Figura 4 - Arquitetura descentralizada.....46
Figura 5 – Modelo lógico do sistema proposto.....54
Figura 6 – Modelo físico do sistema proposto.....56
Figura 7 – Modelo de dados. 58
Figura 8 – Estrutura do XML. 61
Figura 9 – Comando SQL genérico para buscar recomendações de um item....68
Figura 10 – Retorno genérico das recomendações.....68
Figura 11 – Comando SQL aprimorado para buscar recomendações de um item.
.....69
Figura 12 – Retorno para filtragem colaborativa baseado em avaliações. 70
Figura 13 – Retorno para filtragem baseada em conteúdo.....70
Figura 14 – Fluxo de execução da tarefa de geração de filtragem colaborativa.71

LISTA DE TABELAS

Tabela 1 - Base de dados de classificações. 33
Tabela 2 - Banco de dados de avaliações médias ajustadas. 37
Tabela 3 - Dados iniciais para o cálculo de recomendação (SESSÃO x ITEM). 62
Tabela 4 - Média das avaliações por sessão. 63
Tabela 5 - Matriz de avaliações ajustada pela média. 63
Tabela 6 - Matriz de recomendação (ITEM x ITEM). 64
Tabela 7 - Dados iniciais representando itens e suas características. 65
Tabela 8 – Matriz de pesos dos itens e suas características. 66
Tabela 9 – Matriz com os pesos ITEM x ITEM baseada em conteúdo. 66
Tabela 10 – Tempos de execução da tarefa de filtragem colaborativa. 72
Tabela 11 – Tempos de execução da tarefa de recomendação baseada em conteúdo. 74

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface
CERN – Conseil Européen pour la Recherche Nucléaire
FBC – Filtragem Baseada em Conteúdo
FC – Filtragem Colaborativa
JVM – Java Virtual Machine
JDBC – Java Database Connectivity
JSON – JavaScript Object Notation
P2P – Peer-to-peer
RI – Recuperação de Informação
SOA – Service Oriented Architecture
SRC – Sistemas de Recomendação Colaborativa
SQL – Structured Query Language
TF-IDF – Term Frequency / Inverted Document Frequency
TI – Tecnologias de Informação
UFSC – Universidade Federal de Santa Catarina
VSM – Vector Space Model
WWW – World Wide Web
XML – eXtended Markup Language

SUMÁRIO

1. INTRODUÇÃO 18
1.1 PROBLEMÁTICA 21
1.2 OBJETIVOS 22
1.2.1 Objetivo Geral 22
1.2.2 Objetivos Específicos 22
1.3 METODOLOGIA 22
1.4 ORGANIZAÇÃO DO TEXTO 23
2. SISTEMAS DE RECOMENDAÇÃO 25
2.1 ABORDAGENS 28
2.1.1 Filtragem Baseada em Conteúdo 28
2.1.2 Filtragem Colaborativa 32
2.1.3 Filtragem Híbrida 38
2.2 APLICAÇÕES 39
3. COMPUTAÇÃO DISTRIBUÍDA 41
3.1 ARQUITETURAS 43
3.1.1 Centralizada 44
3.1.2 Descentralizada 45
3.1.3 Híbrida 46
3.2 MIDDLEWARE 47
3.3 ORGANIZAÇÃO DE SERVIDORES 47
3.3.1 Cluster 48
3.3.2 Grid 49
3.3.3 Computação nas Nuvens 50
3.4 VANTAGENS E DESVANTAGENS 52
3.5 EXEMPLOS DE PROJETOS E APLICAÇÕES 53
4. SISTEMA PROPOSTO 54
4.1 MODELO LÓGICO 54
4.2 MODELO FÍSICO 55
4.2.1 Modelo de Dados 57
4.2.2 Geração de Recomendações 59

5. DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS60
5.1 CENÁRIO ELABORADO 60
5.2 DETALHAMENTO DOS CÁLCULOS.....61
5.2.1 Cálculo para a Filtragem Colaborativa 61
5.2.2 Baseado em Conteúdo 64
5.3 EXEMPLOS DE RECOMENDAÇÃO.....67
5.3.1 Consulta sobre a tabela de interação.....67
5.4 PROCESSAMENTO DISTRIBUÍDO.....70
5.4.1 Filtragem Colaborativa 71
5.4.2 Recomendação Baseada em Conteúdo 73
6. CONSIDERAÇÕES FINAIS 75
REFERÊNCIAS 77

1. INTRODUÇÃO

Atualmente a *Internet* faz parte da sociedade e é vista como uma ferramenta utilizada por todas as classes sociais para as mais variadas finalidades. Em 2006, um bilhão de pessoas em todo o mundo já possuíam acesso à *Internet* e os dispositivos móveis superavam os computadores de mesa em uma proporção de dois para um (MUSSER; O'REILLY, 2007). Atrelado a *Internet* está o termo *Web*, criado por Tim Berners-Lee. A *World Wide Web*, ou apenas *Web*, como é popularmente conhecida, surgiu no CERN (*Conseil Européene pour la Recherche Nucléaire*). Em 1989, Tim Berners-Lee deparou-se com o problema de perda de informações que era ocasionado pela intensa rotatividade de integrantes do CERN. As informações não eram registradas em documentos e perdiam-se com a saída dos membros que as possuíam. Para melhorar a busca por informações, Tim Berners-Lee criou um sistema distribuído aplicando os conceitos de nodos e hipertexto e posteriormente em 1990, criou um servidor para seu computador com um navegador de interface gráfica marcando o inicio da *World Wide Web*. Inicialmente, a *Web* era composta apenas por páginas com imagens e textos estáticos. Esta primeira geração da *Web* ficou conhecida como *Web* 1.0. (BERNERS-LEE, 1989).

Com o passar do tempo, a *Web* foi evoluindo e deixando de ser apenas um conjunto de páginas de textos estáticos. Em 2005, durante uma reunião em uma conferência, Tim O'Reilly apresentou o termo *Web* 2.0 que marcaria uma revolução na *Internet* (O'REILLY, 2005). Para Krotzfleisch et al. (2008) o termo *Web* 2.0 sugere um salto tecnológico, na medida em que é, de fato, usado para caracterizar uma nova configuração de tecnologias voltada à *Internet*. Em contraste com a *Web* 1.0, que se preocupava basicamente em como definir e criar destinos para os usuários da *Internet*, a *Web* 2.0 é sobre pessoas e conteúdo.

Para Jannach et al. (2011), na *Web* 2.0 os usuários da *Internet* assumem gradativamente o papel de provedores de conteúdo. Esta característica pode ser observada através das ações dos usuários da *Web* de hoje, que de maneira ativa e voluntaria publicam conteúdos em portais populares. Krotzfleisch et al. (2008) afirmam que a *Web* 2.0 capacitou o usuário a produzir conteúdo e a compartilhá-lo com qualquer habitante do mundo, tornando conhecidos mundialmente portais de informação que possuem esta finalidade como MySpace®, Wikipedia®, Youtube®, Facebook®. Em 1980 Alvin Toffler, identificou

esta característica e criou um termo para classificar este tipo de usuário. Nascia então o conceito denominado *prosumers*, usuários que agregam características de produtores e consumidores de informação (TOFFLER, 1980). Conforme Voigt e Ernst (2010), a *Web* 2.0 também é uma realidade no cotidiano de muitas empresas dentre os mais variados tipos de negócios.

De acordo com Musser e O'Reilly (2007), no primeiro trimestre de 2006, 280 mil novos usuários se inscreveram no MySpace® por dia, e o *site* obteve o segundo maior tráfico de dados na *Internet*. No segundo trimestre de 2006, foram criados 50 milhões de novos blogs adicionados a uma taxa de dois por segundo. Em 2005, o eBay® realizou cerca de 8 bilhões de transações.

Segundo Hilbert e López (2011), em 1986 a capacidade tecnológica de comunicação mundial era de 432 *exabytes*. Esta capacidade sofreu um significativo aumento, chegando à marca de 1.15 *zettabytes* em 2007. A capacidade de armazenamento também sofreu um crescimento. Em 1986 a capacidade de armazenamento era de 2,6 *exabytes*, chegando a 309 *exabytes* em 2007.

De acordo com IBM (2014), todos os dias 2,5 quintilhões de *bytes* de dados são criados e deste total, 90 por cento dos dados atualmente no mundo foram produzidos nos últimos dois anos. Segundo Wu et al. (2014), desde o advento da tecnologia da informação a capacidade humana de gerar dados expandiu vertiginosamente. Como outro exemplo, os autores citam que em 4 de outubro de 2012, o primeiro debate presidencial entre o Presidente Barack Obama e o Governador Mitt Romney produziu mais de 10 milhões de *tweets* em 2 horas.

Diante deste cenário, encontrar as informações de interesse tem se tornado uma tarefa árdua para os usuários da *Internet*. Para Ferreira e Oliveira (2012), a sobrecarga de informação é um fenômeno que não veio para auxiliar o usuário no processo de escolha e tomada de decisão. Inicialmente auxiliados por motores de busca, as informações encontradas na maioria das vezes não vinham de encontro ao seu interesse, surgindo a necessidade da filtragem das informações.

Com o imenso volume de informações, muitas opções eram oferecidas aos usuários. As dificuldades de realizar uma escolha eram comuns até mesmo para o usuário mais experiente. Fazia-se necessário o auxílio de um sistema computacional que executasse o papel de um facilitador nas escolhas e tomadas de decisão. Tal situação tornou-se uma importante fonte de pesquisa e originou o surgimento da área de

Sistemas de Recomendação (RESNICK et al., 1994, SHARDANAND; MAES, 1995).

Em meados da década de 90 surgiram os primeiros sistemas de filtragem colaborativa, que posteriormente se tornaram a base para os Sistemas de Recomendação (GOLDBERG et al., 1992, RESNICK; VARIAN, 1997). Os primeiros a propor um sistema de recomendação foram Goldberg et al. (1992) com o *Tapestry*. Este sistema foi desenvolvido com a finalidade de auxiliar seus usuários a encontrarem documentos que lhe interessassem, diante de uma enxurrada de conteúdo causada pelo aumento do uso dos correios eletrônicos. Um princípio básico do trabalho do *Tapestry* é que uma filtragem mais eficaz pode ser feita através do envolvimento humano no processo. Pouco tempo depois Resnick e Varian (1997) apresentaram um segundo trabalho denominado *GroupLens*. Um sistema que classificava notícias de acordo com a avaliação feita por outros usuários.

Os sistemas de recomendação realizam uma coleta de dados de um grupo de usuários. Com base nestas informações, estes sistemas têm como objetivo principal gerar recomendações significativas de itens de maneira geral (livros, músicas, filmes, entre outros), que estejam de acordo com o interesse de um grupo de usuários (BOBADILLA et al. 2013, MELVILLE; SINDHWANI, 2010).

Em meio a este cenário repleto de dados em larga escala a questão do processamento deve ser observada, pois o volume de informação tende a crescer exponencialmente. Como solução deste problema surge a questão dos sistemas distribuídos que, segundo Kshemkalyani e Singhal (2008), tratam-se de componentes independentes que se comunicam por meio de um canal que em conjunto possuem a tarefa de resolver um problema que não poderia ser solucionado individualmente. Muitos dos problemas resultantes do processamento de grandes volumes de dados requerem computadores fortemente acoplados contendo baixa latência e altas larguras de banda de comunicação (FOSTER; KESSELMAN; TUECKE, 2001).

Ainda conforme os autores Kshemkalyani e Singhal (2008), sistemas distribuídos consistem em um conjunto de processadores que não compartilham uma memória global e que estão conectados por meio de uma rede, sendo que neste canal são realizadas as trocas de mensagens entre os mesmos. Afirmar que uma determinada aplicação está sendo executada em um sistema distribuído é expressar que um conjunto de processos está alocado a diferentes processadores interligados na rede.

Para Coulouris, Dollimore e Kindberg (2005), a construção de sistemas distribuídos promove desafios como a heterogeneidade de seus componentes, isto é, variedades e diferenças no que diz respeito a *hardware* de computador, sistemas operacionais, linguagens de programação, entre outros. Além disto, deve ser um sistema aberto, o que permite a adição ou substituição de seus componentes sem maiores problemas. Deve também, prover segurança, tratamento de falhas, escalabilidade, ou seja, o desempenho do sistema não sofrerá grandes impactos com a alteração do número de usuários. Por fim, deve possibilitar o gerenciamento de concorrência de componentes, no que diz respeito ao acesso de usuários ao mesmo tempo a determinado recurso que é compartilhado e possuir transparência ao usuário final, onde não é possível perceber a maneira pelo qual o sistema opera.

1.1 PROBLEMÁTICA

Com a rápida evolução da rede mundial dos computadores, diariamente cresce o número de pessoas se conectando por meio dos mais variados dispositivos.

A *Web* deixou de ser apenas um conjunto de páginas estáticas e sem atrativo, se incorporando ao cotidiano de todos através das mais variadas formas. Ela atende as necessidades tanto do usuário mais simples, que deseja apenas acessar sua conta de *email*, até o usuário mais exigente, que a utiliza para interagir de forma mais dinâmica com dezenas de pessoas em um único dia.

Esta evolução fez com que seus usuários, que durante muito tempo, apenas consumiam informações, passassem a desempenhar também o papel de produtores de informação. Com tantos produtores ativos espalhados pelo globo a quantidade de dados gerados têm tido um crescimento vertiginoso. Diante desta demanda, encontrar informações qualificadas se tornou uma tarefa árdua.

Com a intenção de auxiliar os usuários nestas escolhas, foram apresentados os primeiros sistemas de filtragem colaborativa, que posteriormente serviram como motivação para o surgimento dos sistemas de recomendação. Os sistemas de recomendação coletam dados resultantes da interação de usuários em determinado sistema ou tarefa e, a partir desses dados, geram recomendações de itens que se encaixem a necessidade de um usuário ou de um grupo de usuários.

Mas para que os sistemas de recomendação obtenham sucesso, é necessário o auxilio de sistemas distribuídos para manipular grandes

volumes de dados. Os sistemas distribuídos com sua flexibilidade e escalabilidade podem oferecer um alto poder de processamento, permitindo que grandes volumes de dados sejam processados em alta velocidade.

Diante destas informações revela-se como pergunta de pesquisa: “Como propor um sistema voltado à recomendação de conteúdo textual capaz de manipular grandes volumes de dados?”.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Propor um sistema voltado à recomendação de conteúdo textual suportado pela computação distribuída.

1.2.2 Objetivos Específicos

Visando atingir o objetivo principal, alguns objetivos específicos são requeridos, entre eles:

- Realizar um levantamento bibliográfico sobre as áreas de pesquisa envolvidas no trabalho, sendo elas, Sistemas de Recomendação e Computação Distribuída;
- Propor um modelo de dados que promova suporte a recomendação de conteúdo;
- Desenvolver um protótipo voltado a recomendações de conteúdo textual que possibilite a proposição deste trabalho;
- Realizar uma discussão dos resultados obtidos através da utilização do protótipo.

1.3 METODOLOGIA

O presente trabalho possui caráter exploratório que comportará o desenvolvimento de um protótipo de Sistema de Recomendação com suporte da Computação Distribuída.

Deste modo foram adotadas as seguintes etapas para a elaboração deste trabalho:

- Levantamento da literatura abrangendo as áreas de Sistemas de Recomendação e Computação Distribuída;
- Definição de sistema com foco na recomendação de conteúdo detalhando o modelo lógico e físico. No modelo físico são apresentadas todas as tecnologias envolvidas no processo;
- Desenvolvimento de um protótipo de Sistemas de Recomendação que será dividido em dois módulos, sendo:
    - O primeiro referindo-se a implementação dos algoritmos com base nas abordagens de Filtragem Colaborativa e Baseada em Conteúdo;
    - O segundo módulo objetiva criar a infraestrutura para o processamento distribuído de modo que as informações geradas possam ser futuramente recuperadas através de consultas na base de dados;
- Avaliação do protótipo de Sistema de Recomendação desenvolvido considerando as duas principais abordagens da área (Filtragem Colaborativa e Baseada em Conteúdo);
- Discussão dos resultados alcançados com o protótipo de sistema de recomendação distribuído proposto no presente trabalho.

1.4 ORGANIZAÇÃO DO TEXTO

O presente documento está estruturado em seis capítulos. No primeiro capítulo, sendo o atual, é apresentada uma contextualização sobre o assunto que será abordado neste trabalho. Também é definida uma problemática bem como são declarados os objetivos, geral e específicos.

O segundo capítulo aprofunda as considerações iniciais, que foram brevemente discutidas no primeiro capítulo sobre Sistemas de Recomendação, procurando enfatizar suas definições, seus modelos existentes, suas aplicações e trabalhos relacionados.

No terceiro capítulo é abordado com mais detalhamento a área de Computação Distribuída, elucidando suas definições, arquiteturas, características, classificações, aplicações e trabalhos relacionados.

O quarto capítulo visa apresentar o sistema proposto neste trabalho, dividido em modelo lógico e físico.

No quinto capítulo é detalhado o desenvolvimento do protótipo apresentando um cenário de utilização que possibilite a avaliação dos resultados, tanto ao nível dos algoritmos implementados quanto ao nível da distribuição do processamento.

No sexto capítulo são apresentadas as considerações finais sobre o presente trabalho, indicando suas limitações e procurando explorar as possibilidades de continuação em trabalhos futuros.

2. SISTEMAS DE RECOMENDAÇÃO

Em 1989, o CERN (*Conseil Européen pour la Recherche Nucléaire*) passava por sérios problemas de perda de informações causados pela alta rotatividade de seus colaboradores. Motivado a solucionar este problema, Tim Berners-Lee desenvolveu uma ferramenta baseada nos conceitos de nodos e uso do hipertexto. Nascia então a versão 1.0 da *Wold Wide Web*, ou simplesmente *Web* como é conhecida (BERNERS-LEE, 1989). Desde sua criação, a *Web* está em constante evolução e agregando diversas funcionalidades.

Em 2005, Tim O'Reilly cunhou a expressão *Web* 2.0 durante uma reunião em uma conferência (O'REILLY, 2005). Segundo Krotzfleisch et al. (2008), a *Web* 2.0 promoveu um salto tecnológico utilizando uma nova configuração de tecnologia voltada à *Internet*. Ela está muito relacionada a dinamicidade das pessoas e conteúdos, o que a difere da a *Web* 1.0 que era mais estática. Conforme Voigt e Ernst (2010), a *Web* 2.0 também já está inserida nas atividades diárias de muitas empresas, independentemente das suas áreas de atuação. Conforme Monteiro e Fidencio (2013), devido a este crescimento a *Web* vem se ampliando surgindo outros conceitos, tais como: *Web* invisível, *Web* visível, *Web* Semântica, *Web* Pragmática, *Web* Social ou 2.0.

A *Web* vem capacitando seus usuários a serem produtores de informação que por sua vez têm disponibilizado este conteúdo com outros usuários geograficamente distribuídos. Tal fato popularizou portais de informação como MySpace®, Wikipedia®, Youtube®, Facebook® em todos os continentes do planeta (KROTZFLEISCH et al., 2008). De acordo com Manno e Shahrabi (2010), a *Web* 2.0 está remodelando a forma como ocorrem as coisas em uma sociedade.

Ainda conforme Valsamidis et al. (2013), a *Web* 2.0 esta presente nos campos e trazendo informações importantes para a agricultura, onde *blogs* vem sendo utilizados como ferramentas para compartilhar informações sobre agricultura de precisão.

Com a *Web* 2.0, os usuários da *Internet* absorveram também a função de provedores de conteúdo. Voluntariamente, todos os dias, usuários da *Internet* publicam e compartilham conteúdo nos mais diversos portais de informação. Alvin Toffler em 1980 cunhou uma expressão para identificar estes usuários híbridos da *Internet*. Desde então os batizou como *prosumer*. Uma expressão originada do inglês, resultante da junção das palavras *producer* (produtor) e *consumer* (consumidor) (TOFFLER, 1980; JANNACH et al., 2011).

Com o crescimento dos produtores de conteúdo, houve um aumento exponencial no volume de dados gerados. Conforme estudos apontados por Lyman e Varian (2003), em 2003, a produção anual de novas informações atingiu a marca de 2 *exabytes*. Dividindo esta produção entre todos os habitantes da terra, encontra-se um coeficiente aproximado a 250 *megabytes* por pessoa.

De acordo com Hilbert e López (2011), em meados da década de 80 a capacidade tecnológica de comunicação mundial era de 432 *exabytes* e a capacidade de armazenamento era apontada em 2,6 *exabytes*. Com o surgimento da *Web* 2.0, notou-se um considerável crescimento, alcançando em 2007 a marca de 1.15 *zettabytes* e 309 *exabytes*, respectivamente.

Informações fornecidas pela IBM (2014) apresentam a expressiva marca de 2,5 quintilhões de *bytes* de dados gerados diariamente. Destaca-se também o fato de que 90 por cento dos dados gerados atualmente no mundo foram produzidos entre os anos de 2012 e 2014. Wu et al. (2014), afirmam que a capacidade humana de produzir dados aumentou exponencialmente devido ao advento da tecnologia da informação.

Com este volume de dados produzidos, os usuários da *Internet* necessitam aplicar seus esforços para encontrar as informações desejadas. Com a compreensão de que a sobrecarga de informação é um fenômeno que não veio para auxiliar o usuário no processo de escolha e tomada de decisão, tornou-se necessário o auxílio de motores de busca. Contudo, na maioria das vezes o resultado das buscas não gerava satisfação ao usuário quanto à efetividade das informações, surgindo a necessidade da filtragem das informações (FERREIRA; OLIVEIRA, 2012). Carrer-Neto et al. (2012), asseguram que esta dificuldade ocorre porque as informações disponíveis na *Internet* foram concebidas para serem lidas apenas por seres humanos, e por isso os sistemas computacionais não conseguem processá-las, nem interpretar os dados presentes nelas.

O crescimento explosivo e a variedade de informações disponíveis na *Web*, juntamente com a rápida introdução de novos serviços de *e-business* (compra de produtos, comparação de produtos, leilão, etc), têm deixado os usuários frequentemente sobrecarregados, levando-os a tomar decisões pouco adequadas. A disponibilidade de opções, em vez de produzir um benefício, tende a diminuir o bem-estar dos usuários. Neste sentido, torna-se necessário o uso de sistemas computacionais capazes de auxiliar nas escolhas e na tomada de decisão. Essa situação configurou-se como uma importante fonte de pesquisa e

promovendo o surgimento dos Sistemas de Recomendação (RESNICK et al., 1994, SHARDANAND; MAES, 1995, RICCI; ROKACH; SHAPIRA, 2011).

O crescimento exponencial da *Web* e o surgimento dos *e-commerces* foram os principais motivos que levaram ao desenvolvimento dos Sistemas de Recomendação (CHEN; CHENG; CHUANG, 2008, CHO; KIM; KIM, 2002, MIN; HAN, 2005). Os primeiros Sistemas de Recomendação surgiram ao decorrer da década de 1990. Inicialmente, os Sistemas de Recomendação eram conhecidos como sistemas de filtragem colaborativa (GOLDBERG et al., 1992, RESNICK; VARIAN, 1997).

Goldberg et al. (1992), desenvolveram um sistema focado no auxílio de seus usuários nas buscas por conteúdos específicos. Esta tarefa, aparentemente simples, se tornava árdua diante do grande volume de informações gerado pelo aumento do uso dos *emails*. Surgia então, o primeiro Sistema de Recomendação chamado de *Tapestry*. Um sistema cujo principio básico era aumentar a eficácia de filtragem através da participação humana neste processo. Um segundo trabalho similar ao *Tapestry* foi desenvolvido por Resnick e Varian (1997). Chamado de *GroupLens*, este sistema classificava e compartilhava notícias de acordo com o resultado da avaliação realizada por muitos outros usuários.

Os Sistemas de Recomendação são ferramentas que realizam uma coleta de dados de um grupo de usuários. Com base nestas informações, estes sistemas têm como objetivo principal gerar recomendações significativas de itens de maneira geral (livros, músicas, filmes, entre outros), que estejam de acordo com o interesse de um grupo de usuários (BOBADILLA et al., 2013, MELVILLE; SINDHWANI, 2010).

Concordando com as informações anteriores, Mahmood e Ricci (2009), afirmam que Sistemas de Recomendação são aplicações inteligentes que ajudam os usuários em suas tarefas de busca de informação, sugerindo os itens que melhor atendam às suas necessidades e preferências. Durante a interação, os usuários fornecem suas preferências e *feedbacks*. Tais informações são coletas e usadas para construir o perfil de usuário e a partir do perfil traçado, itens correspondentes a este perfil são utilizados em futuras recomendações.

Os usuários de Sistemas de Recomendação são beneficiados pelo ganho de tempo e pela facilidade do uso, pois estes sistemas proporcionam um baixo nível de esforço e não é necessário possuir grande experiência. Aliado a estes benefícios está o eficiente retorno para cada busca destinada a encontrar itens de seus interesses (FERREIRA; OLIVEIRA, 2012).

Geralmente os Sistemas de Recomendação podem ser divididos em duas categorias, sistema de recomendação pessoal e sistema de recomendação de grupo. O primeiro é eficaz na filtragem de informações úteis que se adéquam às necessidades de cada usuário. Já o segundo deve fornecer sugestões eficazes para permitir a decisão em grupo e satisfazer as necessidades dos usuários em atividades de grupo (CHEN; CHENG; CHUANG, 2008). Adomavicius e Tuzhilin (2005) classificam os Sistemas de Recomendação em três tipos, considerados tradicionais, são eles: Sistemas de Recomendação com Filtragem Colaborativa, Sistemas de Recomendação com Filtragem Baseada em Conteúdo e uma terceira abordagem que é gerada pela união dos primeiros, denominada como Sistemas de Recomendação Híbridos.

Ainda há autores que identificam outras abordagens como, por exemplo, Jannach et al. (2011) e Carrer-Neto et al. (2012) citam em seus trabalhos a filtragem baseada em conhecimento. Nesta abordagem o perfil do usuário é modelado de modo que, através de algoritmos de inferência, seja possível identificar a correlação entre as suas preferências e produtos existentes, serviços ou conteúdos. Montaner et al. (2003), ainda mencionam em sua pesquisa uma outra abordagem. Esta por sua vez é denominada como Filtragem Demográfica, onde as descrições dos usuários são utilizadas para estabelecer uma relação entre um determinado item e o perfil dos usuários que gostarão deste. Os perfis são classificados de acordo com estereótipos preestabelecidos. As informações que compõem o perfil são cedidas pelo próprio usuário através do preenchimento de um formulário de cadastro.

Neste trabalho serão discutidas as seguintes abordagens de Sistemas de Recomendação consideradas tradicionais: Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Filtragem Híbrida (PARK et al., 2012, ADOMAVICIUS; TUZHILIN, 2005).

2.1 ABORDAGENS

2.1.1 Filtragem Baseada em Conteúdo

A abordagem baseada em conteúdo tem suas raízes na área de Recuperação de Informação (RI) e nos filtros de pesquisas de informação. Esta abordagem exerce recomendações de itens de acordo com a sua similaridade, onde um item é recomendado de acordo com a classificação obtida previamente por outro usuário com as mesmas

preferências. Por exemplo, em um sistema de recomendação de filmes com o objetivo de recomendar filmes para o usuário *(a)*, o sistema de recomendação com filtragem baseada em conteúdo tenta compreender as semelhanças entre os filmes que o usuário *(a)* avaliou no passado (atores específicos, diretores, gêneros, assunto, etc.). Em seguida, serão recomendados os filmes que têm um alto grau de semelhança de acordo com as preferências do usuário *(a)* (ADOMAVICIUS; TUZHILIN, 2005; CHEN; CHENG; CHUANG, 2008; LÜ et al., 2012).

Para Melville e Sindhwani (2010), muitas pesquisas nesta área tem se concentrado em recomendação de itens com conteúdo textual associado como: páginas *Web*, livros e filmes. Várias abordagens têm tratado estes casos como Recuperação de Informação (RI), pois as interações do usuário com o sistema geram um perfil do item (vetor de informações). Estes vetores de informações contêm os atributos mais relevantes de um item e futuramente serão utilizados na comparação dos atributos de outros itens com o objetivo de recomendar itens que possuam vetores com informações similares.

2.1.1.1 Modelo Vetorial

Em aplicações de Recuperação de Informação (RI), é muito comum o uso do Modelo Espaço Vetorial (*Vector Space Model* - VSM) (MANNING; SCHÜTZE, 1999). O amplo uso do VSM em RI é creditado a sua simplicidade e aplicabilidade, onde VSM trata a proximidade semântica como proximidade espacial. Ao aplicar o VSM na RI, transforma-se um documento em um vetor de espaço *n*-dimensional, onde *n* indica o número dos diversos termos (RUSSEL; NORVIG, 1995). Os vetores em conjunto formam a matriz documento-termo, em que esta matriz pode ser armazenada com uma estrutura de índice invertido (GONÇALVES, 2006).

Cada elemento do vetor recebe também um identificador e um peso representando a sua importância em relação ao conteúdo de um documento. De acordo com Trstenjak, Mikac e Donko (2014), este peso pode ser determinado através de um método estatístico numérico chamado TF-IDF (*Term Frequency / Inverted Document Frequency*). Este método é frequentemente utilizado em processamento de linguagem natural (NLP) ou na recuperação de informação e mineração de texto (MASUDA; MATSUZAKIB; TSUJIIC, 2011, FRIEDMAN; RINDFLESCH; CORN, 2013). O Método TF-IDF determina a frequência relativa de termos em um documento específico através de uma proporção inversa do termo ao longo de todo o conteúdo do

documento. Na determinação do valor, o método usa dois elementos: A frequência que o termo é encontrado em um documento (TF), e a frequência inversa de documentos que contêm o termo (IDF). O peso de cada termo é incrementado proporcionalmente ao número de exibições deste termo no documento (TRSTENJAK; MIKAC; DONKO, 2014). Trazendo a aplicação do TF-IDF para o contexto de Sistemas de Recomendação, o item que na Recuperação de Informação é o documento passa a ser um item (elemento de análise), então os termos alocados no vetor passarão a serem os atributos deste item (REINEHR, 2013).

Uma vez estipulado o vetor base pode-se recuperar outros vetores através de suas similaridades. Medidas de similaridades como, o produto interno (SALTON; BUCKLEY, 1988) e o cosseno (JONES; FURNAS, 1987) são utilizadas para determinar a distância entre estes vetores. Ainda autores como Lee et al. (2014), utilizam a distância Euclidiana para medir a distância entre dois vetores. Para Nouali e Blache (2004), o modelo vetorial possibilita que os documentos recuperados sejam facilmente classificados e avaliados conforme sua importância, tornando-o um modelo flexível.

2.1.1.2 Similaridade entre Vetores

De acordo com Jones e Furnas (1987), para se encontrar a similaridade entre vetores, deve-se aplicar a medida que indica a proximidade entre eles em um determinado universo Ω, onde Ω é identificado como um grupo de documentos. Sendo assim, um universo contendo um grupo de objetos que podem ser representados através de um vetor, pode ser representado por Ω.

Diversas equações que permitem calcular a distância entre vetores são apontadas por EGGHE e MICHEL (2002), como: o índice Jaccard, a medida overlap (máxima e mínima), a medida do cosseno e a medida do pseudo-cosseno.

A similaridade entre vetores é encontrada ao calcular o ângulo do cosseno que é formado pelos vetores que representam os documentos contendo os termos e frequências (ADOMAVICIUS; TUZHILIN, 2005). A equação que mede o ângulo entre dois vetores pode apresentar o resultado com uma variação entre 1.0 (cos(0°) = 1.0) onde os vetores apontam na mesma direção, 0.0 (cos(90°) = 0.0) quando os vetores formam um ângulo reto e -1.0 (cos(180°) = -1.0) quando os vetores apontam em direções opostas (GONÇALVES, 2006). A equação pode ser representada da seguinte maneira:

$$\cos\theta = \frac{\sum_{i=1}^{n}(t_i x q_i)}{\sqrt{\sum_{k=1}^{n}(t_k)^2}\, x \sqrt{\sum_{j=1}^{n}(q_j)^2}}$$

Sendo que t_i e t_k representam o peso contido nas posições as *i*th e *k*th do vetor *t*, assim como q_i e q_j representam os pesos encontrados nas posições *i*th e *j*th do vetor *q*. Ao aplicar esta fórmula será obtido um resultado variante ente 0 e 1 que determinará o grau de similaridade dos vetores.

2.1.1.3 Limitações

Diversos autores alertam para as limitações da filtragem baseada em conteúdo, sendo as três principais:

- Análise limitada pelo conteúdo: É difícil analisar o conteúdo de dados pouco estruturados. A filtragem baseada em conteúdo tem uma aplicação muito complexa quando se trata de conteúdos multimídia com imagens e vídeos (CAZELLA et al., 2010, BOBADILLA et al., 2013, RICCI et al., 2011). Outra característica mencionada por Jannach et al. (2011) é que a filtragem baseada em conteúdo não consegue distinguir se um texto foi bem ou mal escrito.
- Superespecialização: A filtragem baseada em conteúdo não pode recomendar itens que são diferentes de qualquer coisa que o usuário tenha visto antes. Além disso, em certos casos, os itens não devem ser recomendados se eles são muito similares a algo que o usuário já tenha visto, como uma notícia diferente descrevendo o mesmo evento (ADOMAVICIUS E TUZHILIN, 2005, CARRER-NETO et al., 2012)
- Problema com novo usuário: O usuário tem que avaliar um número suficiente de itens para que um sistema com filtragem baseada em conteúdo entenda suas preferências e apresente a este novo usuário recomendações confiáveis. Portanto, um novo usuário, tendo pouquíssimas avaliações, não seria apto a obter recomendações precisas (ADOMAVICIUS E TUZHILIN, 2005).

2.1.2 Filtragem Colaborativa

Os primeiros trabalhos no campo da Filtragem Colaborativa (FC) foram publicados no início de 1990. Goldberg et al. (1992), apresentaram o sistema *Tapestry* que utilizava filtragem colaborativa para filtrar *e-mails* simultaneamente, a partir de várias listas de discussão, baseado na opinião de outros usuários sobre suas leituras. Resnick et al. (1994), descreveram o sistema *GroupLens* que foi uma das aplicações pioneiras do campo onde os usuários avaliavam os artigos em uma escala de 1-5. Após a leitura, os usuários ofereciam suas sugestões (TAKÁCS; PILÁSZ; NÉMETH; TIKK, 2009).

A maioria das abordagens colaborativas se concentra em encontrar usuários com interesses semelhantes, a fim de compartilhar recomendações entre eles (CARRER-NETO et al., 2012). Sendo assim, itens que foram bem qualificados por um grupo de usuários, serão sugeridos a outro usuário que possui gostos e preferências semelhantes a este grupo (CHEN; CHENG; CHUANG, 2008).

Atualmente esta técnica é amplamente utilizada, sendo o método mais aplicado para uso de recomendação pessoal (CHEN; CHENG; CHUANG, 2008). Um exemplo de sua aplicação é encontrado no portal de vendas Amazon®, conhecido mundialmente (LINDEN; SMITH; YORK, 2003).

Alguns autores como Knijnenburg et al. (2012) classificam a filtragem colaborativa em dois métodos: Elicitação explícita e implícita. Na elicitação explícita, os usuários avaliam os itens através de uma escala, por exemplo, atribuindo de uma a cinco estrelas (GENA et al., 2011; POMMERANZ et al. 2012). Na elicitação implícita, as preferências são derivadas de uma análise do comportamento de navegação e seleção de usuários. Em concordância, Yang et al. (2014), tratam as elicitações como *feedbacks* de usuários aos quais dividem em *feedback* explícito onde o usuário atribui uma classificação a um item e *feedback* implícito obtido quando o usuário clica em um *link*, ouve uma música, ou compra um item. Segundo Koren et al. (2009), pesquisas mostram que uma combinação de resultados entre os *feedbacks* explícitos e implícitos proporcionam uma precisão de recomendação superior.

A seguir, serão exploradas duas abordagens de recomendações aplicadas na filtragem colaborativa: Recomendações da vizinhança baseada em usuários e baseada em itens. Tais abordagens assim como seus exemplos, foram baseadas Jannach et al. (2011).

2.1.2.1 Recomendações de vizinhos próximos com base em usuários

A primeira abordagem a ser explorada, será a recomendação do vizinho mais próximo baseado no usuário. Para aplicar esta abordagem é necessária uma base de dados de avaliações. A partir do usuário ativo, são formados pares que identificam os vizinhos mais próximos através das preferências semelhantes. É aplicada uma previsão calculada, para cada item avaliado pelos pares que o usuário ativo ainda não tenha conhecido.

A Tabela 1 exibe a comparação entre as avaliações dos usuários para os itens identificando-se o usuário ativo como Teo. O peso das avaliações varia entre 1 e 5, sendo 1 para itens que o usuário não gostou e 5 para itens que ele gostou muito. Visto que Teo ainda desconhece o item 5, um sistema de recomendação deverá definir se Teo irá gostar deste item baseando-se na avaliação dos usuários semelhantes.

Tabela 1 - Base de dados de classificações.

	Item 1	Item 2	Item 3	Item 4	Item 5
Teo	5	3	4	4	?
Usuário 1	3	1	2	3	3
Usuário 2	4	3	4	3	5
Usuário 3	3	3	1	5	4
Usuário 4	1	5	5	2	1
Usuário 5	1	2	3	4	5

Fonte: Jannach et al. (2011).

A seguir serão abordados os cálculo necessários para a recomendação. Para identificar o conjunto de usuários será utilizado $U = \{u_1, ..., u_n\}$, para o conjunto de produtos $P = \{p_1, ..., p_m\}$ e R sendo uma matriz n x m de classificações $r_{i,j}$. As posições da matriz $r_{i,j}$ ficam vazias quando não existem avaliações do usuário i para o item j.

Para determinar a semelhança ente usuários, utiliza-se o coeficiente de Pearson onde a similaridade *sim(a, b)* entre os usuários *a* e *b*, para a matriz de classificação R, é definido em fórmula abaixo. O coeficiente de Pearson extrai a média das classificações para tornar os usuários compatíveis. Isto ocorre, pois alguns usuários possuem a tendência de sempre aplicar avaliações altas, enquanto outros são pré-dispostos a aplicar baixas classificações. A classificação média do usuário é representada pelo símbolo $\overline{r_a}$.

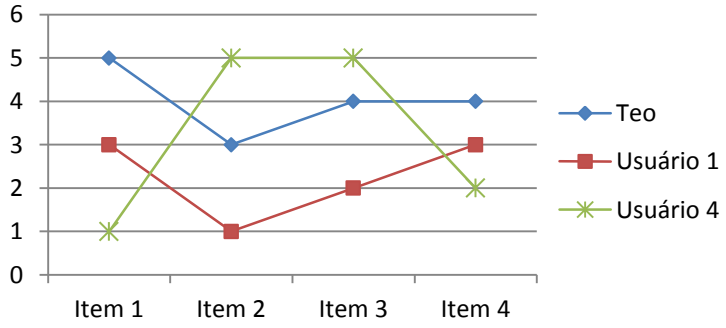
$$sim(a,b) = \frac{\sum_{p\in P}(r_{a,p} - \overline{r_a})(r_{b,p} - \overline{r_b})}{\sqrt{\sum_{p\in P}(r_{a,p} - \overline{r_a})^2}\sqrt{\sum_{p\in P}(r_{b,p} - \overline{r_b})^2}}$$

A semelhança entre Teo e o Usuário 1 é, portanto, como se segue $(\overline{r_{Teo}}) = (\overline{r_a}) = 4, (\overline{r_{Usuário1}}) = (\overline{r_b}) = 2.4$:

$$\frac{(5-\overline{r_a}) * (3-\overline{r_b}) + (3-\overline{r_a}) * (1-\overline{r_b}) + \cdots + (3-\overline{r_a}) * (4-\overline{r_b})}{\sqrt{(5-\overline{r_a})^2 + (3-\overline{r_a})^2 + \cdots}\sqrt{(3-\overline{r_b})^2 + (1-\overline{r_b})^2 + \cdots}} = 0.85$$

O coeficiente de correlação de Pearson tem valores entre +1 (forte correlação positiva) e -1 (forte correlação negativa). Baseado nestes cálculos, percebe-se que houve semelhança entre as classificações de Teo entre o Usuário 1 e Usuário 2. Aplicando o coeficiente de Pearson, percebe-se que existe uma semelhança entre as classificações de Teo e do Usuário 1 pois existe uma correlação linear clara entre as classificações como pode ser visualizado na Figura 1.

Figura 1 - Comparação entre as classificações de Teo e outros usuários.



Fonte: Jannach et al. (2011).

Para fazer uma previsão para o Item 5, deve-se decidir qual das classificações vizinhas deverão ser consideradas. Uma possível fórmula para calcular a previsão de classificação de um usuário qualquer (a) para um determinado item (p) é considerando a média de classificação $\overline{r_a}$ em que os vizinhos mais próximos (N) atribuíram a este item (p), como na fórmula a seguir:

$$prev(a,p) = \overline{r_a} + \frac{\sum_{b \in N} sim(a,b) * \left(r_{b,p} - \overline{r_b}\right)}{\sum_{b \in N} sim(a,b)}$$

Neste exemplo a previsão da classificação que Teo fará para o item 5, baseada nas classificações de seus vizinhos próximos Usuário1 e Usuário 2, será a seguinte:

$$4 + 1/(0.85 + 0.7) * (0.85 * (3 - 2.4) + 0.70 * (5 - 3.8)) = 4.87$$

Com base nestes cálculos, podem-se aplicar previsões de classificações para itens desconhecidos por um usuário. Deve-se considerar que na realidade as bases de dados de classificações podem ser consideravelmente maiores contendo milhões de usuários e itens, sendo necessário um expressivo poder computacional para atender a esta demanda.

2.1.2.2 Recomendações de vizinhos próximos com base em itens

A Filtragem Colaborativa já provou sua qualidade em vários domínios de aplicações. Porém, quando se trata de uma aplicação com um grande volume de dados como, por exemplo, um site de comercio eletrônico com milhares de itens e usuários, torna-se praticamente impossível calcular recomendações em tempo real, pois o numero de usuários pode crescer em larga escala. Como o número de itens não acompanha este ritmo de expansão criou-se a recomendação de vizinhos mais próximos com base em itens onde a ideia principal é recomendar itens conforme a similaridade entre eles.

No exemplo proposto na Tabela 1, percebe-se que as avaliações existentes para o item 5 se assemelham às classificações do item 1. As recomendações baseadas em itens simplificam a visão das classificações do usuário ativo para os itens similares. Para gerar uma previsão de

classificação para o item 5, será aplicado a média ponderada com base nas classificações anteriores do usuário ativo.

Para encontrar itens semelhantes, uma medida de similaridade deve ser definida. Em abordagens de recomendação baseadas em itens, a similaridade do cosseno é estabelecida como uma métrica padrão, uma vez que foi demonstrado que produz os resultados mais precisos.

A similaridade entre dois itens *a* e *b*, visto como os vetores de classificação correspondentes $\vec{a}$ e $\vec{b}$ formalmente definido como a seguir:

$$sim(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| * |\vec{b}|}$$

A seguir, encontra-se a aplicação da fórmula aos dados propostos para o item 1 e o item 5 na Tabela 1 para encontrar a similaridade através do cosseno.

$$sim(I5, I1) = \frac{3 * 3 + 5 * 4 + 4 * 3 + 1 * 1}{\sqrt{3^2 + 5^2 + 4^2 + 1^2} * \sqrt{3^2 + 4^2 + 3^2 + 1^2}} = 0.99$$

Agora, deve-se aplicar a medida cosseno ajustada, como na medida Pearson abordada na seção anterior, neste caso sendo U o conjunto de usuários que avaliaram os itens *a* e *b*. A medida do cosseno ajustado é então calculada da seguinte forma:

$$sim(a, b) = \frac{\sum_{u \in U}(r_{u,a} - \overline{r_u})(r_{u,b} - \overline{r_u})}{\sqrt{\sum_{u \in U}(r_{u,a} - \overline{r_u})^2} \sqrt{\sum_{u \in U}(r_{u,b} - \overline{r_u})^2}}$$

Portanto, pode-se transformar a base de dados de classificações originais e substituir os valores de classificação originais com os desvios em relação as avaliações médias, como mostrado na Tabela 2.

Tabela 2 - Banco de dados de avaliações médias ajustadas.

	Item 1	Item 2	Item 3	Item 4	Item 5
Teo	1.00	-1.00	0.00	0.00	**?**
Usuário 1	0.60	-1.40	-0.40	0.60	0.60
Usuário 2	0.20	-0.80	0.20	-0.80	1.20
Usuário 3	-0.20	-0.20	-2.20	2.80	0.80
Usuário 4	-1.80	2.20	2.20	-0.80	-1.80

Fonte: Jannach et al. (2011).

O valor ajustado de similaridade do cosseno para Item 5 e Item 1 para o exemplo é encontrado da seguinte forma:

$$\frac{0.6 * 0.6 + 0.2 * 1.2 + (-0.2) * 0.80 + (-1.8) * (-1.8)}{\sqrt{(0.6)^2 + 0.2^2 + (-0.2)^2 + (-1.8)^2} * \sqrt{0.6^2 + 1.2^2 + 0.8^2 + (-1.8)^2}} = 0.80$$

Após encontrar as semelhanças entre os itens, pode-se prever a classificação de Teo para Item 5, calculando uma soma ponderada das classificações de Teo para os itens que são semelhantes ao Item 5. Formalmente, pode-se prever a classificação para usuário *U* para um produto *P* como se segue:

$$perd(u,p) = \frac{\sum_{i \in ClassificacaoItem(u)} sim(i,p) * r_{u,i}}{\sum_{i \in ClassificacaoItem(u)} sim(i,p)}$$

Como na abordagem com base no usuário, o tamanho da vizinhança é normalmente considerado também limitado a um tamanho específico, ou seja, nem todos os vizinhos são considerados para a previsão.

2.1.2.3 Limitações

É importante ressaltar que assim como a filtragem baseada em conteúdo, a filtragem colaborativa também possui as suas limitações (WU; CHANG; LIU, 2014, CAZELLA et al., 2010, ADOMAVICIUS; TUZHILIN, 2005):

- Problema do primeiro avaliador: Não há como recomendar um novo item cadastrado a outros usuários, até que este obtenha avaliações de outros usuários.

- Problema de avaliações esparsas: A probabilidade das avaliações se tornarem muito esparsas é grande quando a quantidade de usuários é pequena em relação ao volume de informações.
- Similaridade: Se um usuário possuir preferências muito variadas a ponto de sair de um padrão, ficará muito complicado encontrar usuários com perfis semelhantes. Este cenário aumenta a possibilidade de gerar recomendações pobres.

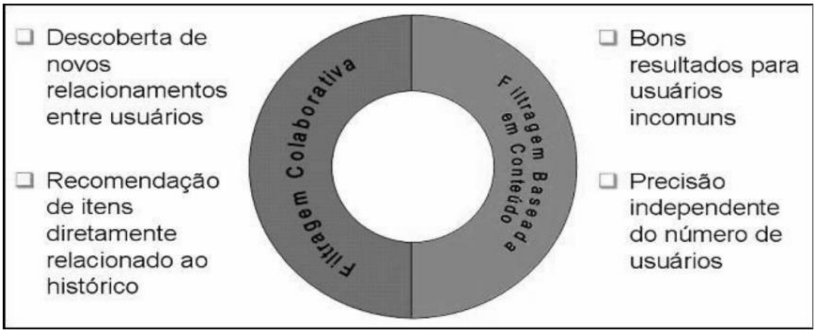
2.1.3 Filtragem Híbrida

Vários sistemas de recomendação aplicam uma abordagem híbrida, combinando métodos colaborativos e baseados em conteúdo, o que ajuda a evitar certas limitações de sistemas baseados em conteúdo e colaboração. Segundo Adomavicius e Tuzhilin (2005), existem diferentes maneiras de combinar métodos colaborativos e baseados em conteúdo em um sistema de recomendação híbrido. Estes podem ser classificados da seguinte forma:

- Implementação de métodos colaborativos e baseados em conteúdo separadamente e combinar as suas previsões;
- Incorporando algumas características baseadas em conteúdo em uma abordagem colaborativa;
- Incorporando algumas características colaborativas em uma abordagem baseada em conteúdo;
- Construção de um modelo de unificação geral que incorpora características das filtragens baseadas em conteúdo e filtragem colaborativa.

Cazella et al. (2011), resumem de maneira simples a abordagem de filtragem híbrida, sendo uma combinação dos pontos fortes das abordagens baseada em conteúdo e filtragem colaborativa, com o objetivo de criar um sistema que atenda melhor as necessidades de seus usuários. Deste modo, os pontos fracos das duas abordagens são neutralizados. A Figura 2 ilustra a ideia dos autores sobre a abordagem híbrida.

Figura 2 - Filtragem híbrida.



Fonte: Adaptado de Cazella et al. (2010).

2.2 APLICAÇÕES

Desde seu surgimento em meados dos anos 90, os Sistemas de Recomendação têm se tornado uma importante área de pesquisa. Suas primeiras aplicações são encontradas nos sistemas *Tapestry* (GOLDBERG et al., 1992) e o *GroupLens* (RESNICK et al., 1994), conforme citado anteriormente.

Entende-se como Sistemas de Recomendação, aqueles *softwares* com o objetivo de sugerir determinados itens para determinados usuários, sendo estes itens dos mais variados tipos, de acordo com as áreas onde o mesmo está inserido (RESNICK; VARIAN,1997; MAHMOOD; RICCI, 2009, RICCI; ROKACH; SHAPIRA, 2011).

Ao passar dos anos esta área foi chamando a atenção de muitas outras áreas e teve sua aplicação diversificada, porém, com maior atuação no setor do *e-commerce*. Um exemplo muito conhecido é encontrado na Amazon® com a sugestão de seus produtos (GHAZANFAR; PRÜGEL-BENNETT, 2014). Também são encontradas aplicações destes sistemas na sugestão de filmes com o MovieLens® (MILLER et al., 2003) e mais recentemente com a empresa Netflix® (HERNANDO et al., 2013) que traz em seu histórico uma marca que comprova o investimento nas pesquisas de geração de recomendações, onde ofereceu um prêmio de 1 Milhão de dólares para quem implementasse um sistema que superasse o utilizado pela empresa na época (BELL; KOREN, 2007). Os Sistemas de Recomendação também estão sendo aplicados na área do turismo (HSU; LIN; HO, 2012, GAVALAS et al., 2014). Também são encontrados em aplicações com foco na avaliação de satisfação de clientes (JIANG; SHANG; LIU,

2010). Com foco no meio ambiente e no uso eficaz dos meios de transportes, encontramos Sistemas de Recomendação aplicados com o objetivo de melhorar o uso dos táxis, recomendando aos motoristas rotas onde é grande a probabilidade de se concretizar um encontro com um cliente em potencial. Fazendo assim com que este automóvel percorra o menor trajeto possível com poucas pessoas (GE et al., 2010).

Recentemente os Sistemas de Recomendação foram inseridos na área da saúde, como por exemplo, estudos que apontam o uso da ferramenta como auxílio ao tratamento da diabetes (CHEN et al., 2012).

3. COMPUTAÇÃO DISTRIBUÍDA

No período inicial da era moderna dos computadores, compreendido entre 1945 a 1985, aproximadamente, os computadores eram grandes e caros e nem mesmo os minicomputadores deixavam de entrar neste grupo, por custarem preços muito elevados. Este cenário começa a sofrer modificações por volta da década de 1980 influenciado por dois avanços tecnológicos. Inicia pelo desenvolvimento de microprocessadores com maior capacidades de processamento, que, inicialmente eram máquinas de 8 *bits* e que foram evoluindo para 16, 32 e 64 *bits*. A evolução também ocorre a partir do surgimento de redes de computadores de alta velocidade (TANENBAUM; STEEN, 2007). Buyya e Ramamohanarao (2007) afirmam que a história da computação distribuída é tão antiga quanto a das redes de computadores.

Desde a década de 1970, com o surgimento da *Internet* e da ARPANET, as novas aplicações desenvolvidas passaram a demandar maior poder de processamento. Este quadro foi se efetivando em função dos avanços na tecnologia de *hardware* e de rede ao longo dos anos (KSHEMKALYANI; SINGHAL, 2008). Para Tanenbaum e Steen (2007), nos últimos 50 anos a evolução que tem ocorrido na tecnologia de computadores tem sido vertiginosa, sendo incomparável com qualquer outra evolução já ocorrida. Isso é notável a partir da perspectiva histórica da computação, onde máquinas que antes tinham valores elevados, chegando a milhões de dólares e que executavam uma única instrução por segundo, agora apresentam valores muito mais acessíveis executando bilhões de instruções por segundo.

Segundo Rao et al. (2013), com este rápido desenvolvimento das tecnologias computacionais e do surpreendente avanço da *Internet*, os recursos da computação foram se tornando mais acessíveis, mais poderosos e estando praticamente quase em todo lugar. Buyya e Ramamohanarao (2007) complementam ainda que os sistemas de computação distribuída não estão mudando apenas a computação, mas também, o modo de como as pessoas vivem, trabalham e interagem com a sociedade.

Esta mudança foi possível devido a crescente demanda de processamento de dados com grandes volumes, onde, inicialmente o processamento era realizado por um único computador, evolui, e passa a utilizar um processo distribuído (DADAN; MINQI; AOYING, 2009).

Deitel, Deitel e Choffnes (2005) complementam ainda que a implementação de sistemas distribuídos pode estar diretamente

relacionada a necessidade de aprimorar a capacidade de um sistema, seja de processamento e de desempenho, por exemplo, melhorar a confiabilidade de uma única máquina, assim como também podem ser projetados para suprir a alta demanda de usuários. Segundo Coulouris, Dollimore e Kindberg (2005), a maior motivação da implementação de sistemas distribuídos está ligada ao fato de haver o compartilhamento de recursos, tais como impressoras, arquivos, páginas *Web* ou registros de banco de dados, entre outros. Os autores ainda citam como exemplo os servidores *Web*, onde gerenciam páginas da *Web* que são ditas como recursos e que são requisitadas por clientes específicos, que neste caso são denominados de navegadores.

A invenção da *Internet* permitiu o acesso remoto de vários serviços, independente da localização dos mesmos e de onde eram acessados. As organizações começaram a gerenciar redes de *Internet* locais, ou *Intranets*, provendo serviços para acesso privado às próprias companhias, e também fornecendo serviços abertos para a *Internet*, tanto para usuários locais quanto para usuários remotos (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Um sistema distribuído se caracteriza por ser um conjunto de computadores independentes, porém, interconectados por uma rede e que trabalham de forma cooperativa, com um objetivo específico e apresentam ao usuário a sensação de estar utilizando um único sistema sem que os mesmos percebam estar utilizando um sistema distribuído (TANENBAUM; STEEN, 2007). Deitel, Deitel e Choffnes (2005) complementam que os sistemas distribuídos podem conter computadores que não necessariamente estão juntos no mesmo ambiente, ou seja, se distribuem geograficamente podendo estar em outros continentes, por exemplo.

Tanenbaum e Steen (2007) declaram que para se obter sucesso no desenvolvimento de sistemas é crucial projetar ou adotar uma arquitetura no que tange a organização lógica do sistema distribuído em componentes de *software*. Sendo assim os autores citam três tipos de arquiteturas: as centralizadas, as descentralizadas e as híbridas e ainda destacam a importância de definir uma camada que forneça transparência no processo distribuído chamada de *middleware* que ao longo do capítulo serão discutidas.

3.1 ARQUITETURAS

Segundo Coulouris, Dollimore e Kindberg (2005), um modelo de arquitetura de um sistema distribuído é caracterizado pelo posicionamento das partes do sistema bem como o relacionamento entre ambas. A especificação dos componentes deste sistema distribuído caracteriza a sua arquitetura. Um dos pontos importantes em que os autores defendem, é a questão da estrutura do sistema ter a capacidade de suprir as demandas atuais bem como até as futuras. Desta forma tornando-se um ponto fundamental na projeção de sistemas distribuídos objetivando uma visão a longo prazo.

Tanenbaum e Steen (2007) iniciam a discussão sobre a organização de sistemas distribuídos destacando duas arquiteturas: as de *software* e de sistema. No que diz respeito às arquiteturas de *software*, ambos afirmam que a partir do momento em que um sistema é visualizado enfatizando seus componentes de *software*, tanto ao nível de organização quanto de interação, passa-se a declarar esta organização como uma arquitetura de *software*, bem como a definição de funções e *interfaces* dos componentes de *software*, relacionamentos entre componentes de *software* (ERICKSON et al, 1993)

Esta organização de componentes de *software*, sob a perspectiva de interação entre ambos e a maneira de como são colocados, levam Tanenbaum e Steen (2007) a caracterizarem-na como estruturas finais denominadas de arquiteturas de sistemas.

Em computação distribuída existem diversas maneiras de organização dos sistemas distribuídos, porém, para Abdul-Fatah e Majumdar (2002), o modelo mais utilizado é o baseado no conceito de cliente-servidor. Coulouris, Dollimore e Kindberg (2005) argumentam da mesma maneira, afirmando o fato de o mesmo ser amplamente empregado e por ser o modelo mais discutido em sistemas distribuídos, além disso, caracterizam-no por ser a arquitetura mais importante historicamente.

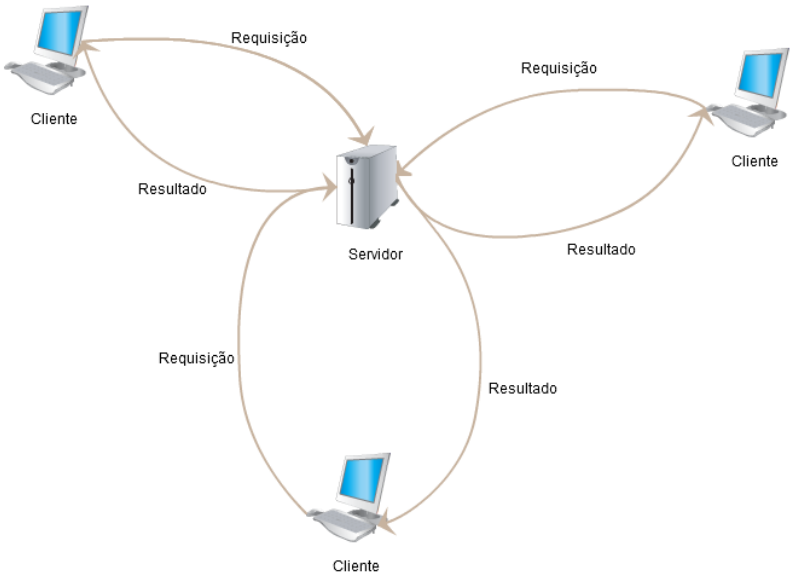
Apesar de o modelo cliente-servidor ser a arquitetura mais utilizada no universo dos sistemas distribuídos, além desta arquitetura centralizada, existem outros dois conceitos intitulados por Tanenbaum e Steen (2007), as arquiteturas descentralizadas e as híbridas.

3.1.1 Centralizada

Conforme Abdul-Fatah e Majumdar (2002), a arquitetura centralizada é composta por duas entidades de *software:* a parte cliente e a parte servidora. A entidade cliente é responsável pela solicitação de informações, enquanto a entidade servidora fica encarregada de fornecer o serviço que fora solicitado anteriormente.

Tal arquitetura pode ser evidenciada na Figura 3 enaltecendo as ligações que existem entre clientes e servidores.

Figura 3 - Arquitetura cliente-servidor.



Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2005).

Os servidores, além de desempenharem a função de provedor de serviços, também podem exercer o papel de cliente, ou seja, quando os mesmos requisitam serviços que são providos por outros servidores (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Para Luh, Chiou e Chang (1996) a implementação da tecnologia cliente-servidor em ambientes distribuídos traz benefícios tais como: um custo relativamente baixo de implementação, uma alta produtividade, um longo ciclo de vida do sistema e melhor capacidade de reutilização

do *software*. Além disso, a arquitetura centralizada reduz a complexidade do controle de *software*, bem como torna as aplicações independentes entre si, o que facilita a manutenção deste tipo de arquitetura.

Muitas das empresas que conceberam o modelo cliente-servidor não conseguiram ser bem sucedidas, fracassando até mesmo em seus segundos e terceiros projetos de cliente-servidor. Isso se deve principalmente pela ausência de um planejamento inicial de um projeto de arquitetura de sistemas. Um projeto inicial define os sistemas e banco de dados necessários e os mapeia para plataformas de computação planejadas e locais dentro da empresa. O sucesso na implementação da tecnologia cliente-servidor depende do planejamento inicial de toda estrutura (FURMSTON-EVANS, 1995).

3.1.2 Descentralizada

Neste tipo de arquitetura não há mais a presença específica das entidades cliente e servidor. Todos os processos que estão envolvidos trabalham de forma semelhante e interagem de forma cooperativa como pares (*peers*), diferentemente do modelo cliente-servidor, onde cada um possui uma especialidade bem definida, apesar da possibilidade de poderem exercer as duas funções (COULOURIS; DOLLIMORE; KINDBERG, 2005).

Os autores Boukhadra, Benatchba e Balla (2013), também enaltecem esta linha de raciocínio, afirmando que no modelo de arquitetura P2P (*peer-to-peer*) todos os processos estão no mesmo nível, ou seja, tanto podem oferecer (função de servidor) quanto podem requisitar serviços e recursos (função de cliente). Além disso, ainda colocam o modelo P2P como uma alternativa ao modelo cliente-servidor, pelo fato de o mesmo oferecer, escalabilidade, robustez, tolerância a falhas, alto desempenho.

Diferentemente do modelo centralizado, cliente-servidor, onde existe a presença de um servidor o qual é encarregado de processar as informações conforme as solicitações de seus clientes, no modelo descentralizado, P2P, as informações estão contidas nos vários computadores da rede, ou *peers* (pares), sendo que as requisições de informações são realizadas diretamente entre ambos. Deste modo, não existe a figura propriamente dita dos servidores, já que os vários computadores da rede podem assumir as funções de cliente e servidor (TOMOYA; SHIGEKI, 2003).

Tal arquitetura pode ser evidenciada na Figura 4 enaltecendo as ligações que existem entre os clientes.

Figura 4 - Arquitetura descentralizada.

Fonte: Adaptado de Coulouris, Dollimore e Kindberg (2005).

3.1.3 Híbrida

Apesar da existência de arquiteturas centralizadas e descentralizadas, há também a combinação das mesmas que acaba sendo denominada como arquitetura híbrida. Como exemplo Tanenbaum e Steen (2007) colocam sistemas de servidores de borda como um importante exemplo de arquiteturas híbridas. Segundo os autores estes sistemas são providos de servidores que se encontram na borda da rede, sendo esta ultima formada pelo limite de redes corporativas e a própria *Internet*.

Tanenbaum e Steen (2007) também classificam como arquiteturas híbridas os sistemas distribuídos colaborativos. Para este tipo de sistema os autores consideram o sistema de compatilhamento de arquivos BitTorrent, caractrerizado por ser um sistema P2P de transferência de arquivos entre clientes.

Em palavras gerais, este sistema tem seu funcionamento baseado na colaboração, onde os clientes da rede trabalham em conjunto fornecendo entre si porções de determinados arquivos até que o arquivo seja completamente formado e para que isso aconteça é fundamental que

exista a colaboração entre os clientes envolvidos na rede para a troca de arquivos. (TANENBAUM; STEEN, 2007)

3.2 MIDDLEWARE

Residindo entre a aplicação e o sistema operacional, o *middleware* é uma camada de *software* que atua como uma espécie de intermediador, sendo responsável por fornecer transparência no que diz respeito a heterogeneidade em sistemas distribuídos, permitindo realizar um mascaramento da heterogeneidade encontrada nas redes de computadores, no *hardware*, em sistemas operacionais e em linguagens de programação. (TANENBAUM; STEEN, 2007, COULOURIS; DOLLIMORE; KINDBERG, 2005).

Júnior (2008) reforça esta ideia, ressaltando que os *middlewares* além de fornecerem uma visão homogênea de redes, protocolos e recursos de sistemas operacionais que envolvam todo o sistema, também propiciam a integração de componentes de *software* reutilizáveis, visando a diminuição de tempo e esforço empregado no desenvolvimento de aplicações e serviços por parte do desenvolvedor.

3.3 ORGANIZAÇÃO DE SERVIDORES

De acordo com Sadashiv e Kumar (2011), a Computação de Alto Desempenho (*High Performance Computing* - HPC) já esteve restrita apenas às instituições que possuíam a capacidade de arcar com os supercomputadores, pois eram significativamente caros. Encontrou-se então a necessidade de aplicar a Computação de Alto Desempenho em pequena escala e com um custo menor, dando origem aos *clusters*. O surgimento de plataformas de *cluster* foi impulsionado por uma série de projetos acadêmicos, como Beowulf, Berkeley NOW e HPVM. Para Li (2008), um sistema de computação em *cluster* é um tipo de sistema de processamento paralelo e distribuído que consiste de uma coleção de computadores autônomos interconectados trabalhando juntos como um recurso de computação integrado.

Ainda segundo Sadashiv e Kumar (2011), a popularidade da *Internet* e a disponibilidade de computadores poderosos e tecnologias de rede de alta velocidade mudou a forma como os computadores são utilizados, possibilitando o surgimento da computação em grade originada na academia em meados de 1990. Este tipo de estrutura possui o objetivo de possibilitar aos usuários utilizar remotamente o poder

computacional ocioso dentro de outros centros de computação de maneira mais fácil. De acordo com Li (2008), em uma escala maior, a computação em grade permite o compartilhamento e agregação de recursos distribuídos geograficamente e oferece suporte a ampla área de computação distribuída.

Cita-se ainda a computação em nuvem sendo um modelo de computação surgido em meados dos anos 2000 (SADASHIV; KUMAR, 2011). A computação em nuvem é a evolução mais recente dos modelos de computação distribuída com foco na virtualização multinível e abstração através da integração de uma variedade de modelos de computação, armazenamento de dados, aplicativos e outros recursos. Ela oferece uma facilidade de uso aos seus usuários e uma poderosa capacidade de computação e armazenamento (MOLLAH; ISLAM; ISLAM, 2012).

Oferece também um conjunto de recursos de computação que os usuários podem acessar através da *Internet*. O princípio básico da computação em nuvem é mudar a computação feita a partir do computador local para a rede. Isso faz com que a empresa use o recurso que inclui rede, servidor, armazenamento, aplicação, serviço e assim por diante, sem grande investimento em suas aquisições, implementações e manutenções (JADEJA; MODI, 2012).

3.3.1 Cluster

Os *clusters* surgiram como resultado da convergência de várias tendências, incluindo a disponibilidade de microprocessadores de alto desempenho de baixo custo e redes de alta velocidade, impulsionados pelo desenvolvimento de ferramentas de *software* padrão para alto desempenho em computação distribuída, e a crescente necessidade de poder de computação para ciência computacional e aplicações comerciais (APON et al., 2004).

Um *cluster* é uma coleção de computadores paralelos ou distribuídos que estão interligados entre si através de redes de alta velocidade, tais como *Ethernet gigabit*, *Myrinet* e *Infiniband*. É usado para enviar e receber mensagens entre processadores. Eles trabalham juntos na execução de intensos processamentos de dados uma vez que a execução em um único computador tornaria a tarefa inviável (YU; ZHOU, 2010; SADASHIV; KUMAR, 2011). Um ambiente de *cluster* tem a capacidade de virtualizar os recursos computacionais avançados, tais como processadores, capacidade de armazenamento, largura de

banda de comunicação e bancos de dados (LI, 2008; SADASHIV; KUMAR, 2011).

A principal motivação de um sistema de computação em *cluster* é fornecer aos usuários e aplicações de acesso generalizado e contínuo, vastos recursos de computação de alto desempenho, criando uma ilusão de uma única imagem do sistema (LI, 2008). Do ponto de vista estrutural um *cluster* é composto de várias máquinas, mas eles funcionam como uma única máquina virtual. Os pedidos dos usuários são recebidos e distribuídos entre todos os computadores independentes para formar um único bloco. Isso resulta em trabalho computacional equilibrada entre máquinas diferentes, melhorando o desempenho dos sistemas (YU; ZHOU, 2010; SADASHIV; KUMAR, 2011).

Tecnologias de *cluster* e de rede oferecem vários tipos de serviços, tais como serviços de computação de alto desempenho, serviços de aplicativos, serviços de dados, serviços de informação e serviços de conhecimento. Estes serviços são prestados pelos servidores, também chamados de nós ou anfitriões em um sistema de computação em *cluster* (LI, 2008). Eles são usados para fins de alta disponibilidade, pois mantêm nós redundantes que são usados para fornecer o serviço quando algum componente do sistema falhar. De modo geral, o desempenho do sistema é melhorado, porque mesmo com a falha de determinado nó, existe outro nó de espera que irá realizar a tarefa eliminando assim pontos únicos de falha (SADASHIV; KUMAR, 2011).

3.3.2 Grid

A computação em *Grid* combina computadores de vários domínios para alcançar um objetivo comum, para resolver uma única tarefa, e pode, em seguida, desaparecer com a mesma rapidez. É análogo ao da rede de energia. Uma das principais estratégias de computação em grade é a utilização de *middleware* para dividir e distribuir pedaços de um programa entre vários computadores. Computação em *Grid* envolve computação de forma distribuída, o que pode também envolver a agregação de sistemas de computação baseados em *cluster* em grande escala. O tamanho de uma grade pode variar de pequeno, como por exemplo, uma rede de estações de trabalho dentro de uma empresa até grandes colaborações em muitas empresas e redes (SADASHIV; KUMAR, 2011). Em larga escala, a computação em grade permite o

compartilhamento e agregação de recursos distribuídos geograficamente e oferece suporte a ampla área de computação distribuída (LI, 2008).

Chetty e Buyya (2002) definem grade como um tipo de sistema paralelo e distribuído que permite o compartilhamento, seleção e agregação de recursos autônomos distribuídos geograficamente de forma dinâmica em tempo de execução, dependendo de sua disponibilidade, capacidade, desempenho, custo, entre outros. Já Foster, Kesselman e Tuecke (2001), definem sistemas em grade como um modelo que coordena recursos que não estão sujeitos ao controle centralizado, utilizando padrões e protocolos de uso geral e interfaces para entregar serviços não triviais e com qualidade.

3.3.3 Computação nas Nuvens

A Computação nas Nuvens (do inglês *Cloud Computing*) é considerada como uma evolução da computação em grade para extrair mais ou aumentar os serviços baseados em infraestrutura (MITTAL; KESSWANI; GOSWAMI, 2013, ZHENGQIAO; DEWEI, 2012, KAHANWAL; SINGH, 2012, SHIRAZ; GANI; KHOKHAR; BUYYA, 2013).

Cloud Computing é um conceito em que os computadores em uma rede são capazes de cooperar uns com os outros para fornecer serviços de rede de grande alcance (YANG; LIU; HUANG; JIANG, 2014). De acordo com Buyya et al.(2009), *Cloud Computing* é um tipo de sistema paralelo e distribuído que consiste em uma coleção de computadores interconectados e virtualizados que são dinamicamente provisionados e apresentados como um ou mais recursos de computação unificada com base no acordo de nível de serviço.

Através da *Cloud Computing*, recursos como poder de processamento e espaço de armazenamento de computação podem ser compartilhados através da *Internet* assim como o acesso ao *hardware*, *software* e recursos de dados, ou seja, um grande espaço virtual. A quantidade de usuários pode ser em larga escala e ainda assim obter os recursos necessários a qualquer momento. (MITTAL; KESSWANI; GOSWAMI, 2013, ZHENGQIAO; DEWEI, 2012). Este modelo permite um acesso ubíquo e conveniente, sob demanda de rede a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados e liberados com um esforço de gerenciamento (SADASHIV; KUMAR, 2011). As tarefas de

processamento são distribuídas em todo o conjunto de recursos formado por um grande número de computadores para permitir uma variedade de aplicações de acesso ao poder de processamento, espaço de armazenamento e uma variedade de serviços de *software*, conforme necessário (HUANG; ZUO; RONG, 2010). Deste modo, o desperdício de recursos redundante em computadores individuais é evitado e a eficiência dos recursos é melhorada (YANG; LIU; HUANG; JIANG, 2014).

Uma das principais vantagens da *Cloud Computing* é que o consumidor utiliza apenas os serviços que ele precisa. Os recursos estão disponíveis para acesso em 24x7 (24 horas por dia, 7 dias por semana) e o consumidor pode acessá-lo a partir de qualquer local via *Internet*. Seus usuários não precisam se preocupar em como os servidores ou recursos são mantidas nos bastidores, eles podem simplesmente comprar recursos de acordo com a necessidade ou até mesmo alugar recursos como, por exemplo, recursos de armazenamento de dados (MITTAL; KESSWANI; GOSWAMI, 2013, YANG; LIU; HUANG; JIANG, 2014).

A *Cloud Computing* oferece confiabilidade, virtualização, qualidade de serviço, agilidade, adaptabilidade e a distribuição de acordo com as necessidades, baixo preço e versatilidade para que os consumidores domésticos desfrutem de armazenamento de alto desempenho e poder de processamento. Ela também elimina a necessidade de investimento inicial caros em Tecnologia da Informação (TI). Isso em geral promove facilidades no gerenciamento por profissionais de TI (ZHENGQIAO; DEWEI, 2012, KAHANWAL; SINGH, 2012, MITTAL; KESSWANI; GOSWAMI, 2013).

De acordo com Zhengqiao e Dewei (2012), a arquitetura de computação da *Cloud Computing* é dividida em 4 níveis: recursos físicos, conjunto de recursos, gestão de *middleware* e de camada de serviço. Conforme Mittal, Kesswani e Goswami (2013), existem quatro tipos de *Cloud Computing*, sendo Nuvem Pública, Nuvem Privada, Nuvem em Comunidade e Nuvem Híbrida.

Os serviços oferecidos pela *Cloud Computing* podem ser agrupados, basicamente, em três tipos de serviço: *Software as a Service* (SaaS), *Infrastructure as a Service* (IaaS) e *Platform as a Service* (PaaS) (SADASHIV; KUMAR, 2011, YANG; LIU; HUANG; JIANG, 2014, SHIRAZ; GANI; KHOKHAR; BUYYA, 2013).

3.4 VANTAGENS E DESVANTAGENS

De acordo com Freeman, Arnold e Hupfer (1999), ao desenvolver uma aplicação voltada para um ambiente computacional distribuído/paralelo, encontra-se uma série de vantagens quando comparado a uma aplicação sequencial. As seguintes vantagens podem ser destacadas:

- Desempenho: Cada unidade de processamento possui um limite de processos em execução. Para que este limite seja superado, torna-se necessário a inclusão de mais unidades de processamento no ambiente. Com um número maior de unidades de processamento pode-se obter um considerável decréscimo no tempo total de processamento da aplicação.
- Custo: Mais unidades de processamento agrupadas geram uma melhoria na relação custo/benefício.
- Escalabilidade: A expansão dos recursos não provoca mudanças na estrutura do sistema. Se um determinado recurso requer o aumento de demanda, o sistema se adapta a esta demanda.
- Integração: Permite a comunicação entre aplicações de diferentes empresas.
- Tolerância a falhas: Se ocorrer a falha de um processo em uma aplicação monolítica, toda a aplicação é afetada. Se a mesma falha ocorrer em uma aplicação distribuída, uma unidade processadora substitui a unidade que apresentou o problema sem que o usuário perceba.
- Acesso a recursos remotos: Uma unidade de processamento pode acessar recursos geograficamente muito distantes.

No entanto, Freeman, Arnold e Hupfer (1999), ressaltam que a utilização de sistemas distribuídos/paralelos apresentam algumas desvantagens, tais como:

- Latência: A latência é proporcional a distancia geográfica entre as unidades de processamento.
- Sincronização: É necessário que seja utilizado métodos de sincronização de processos devido ao fato da aplicação ser dividida em vários processos.
- Falha parcial: A ocorrência de falhas torna-se mais provável conforme o tempo de execução da aplicação aumenta e mais unidades de processamento são inseridas no ambiente.

3.5 EXEMPLOS DE PROJETOS E APLICAÇÕES

A computação em nuvem também tem sido chamado de *utility computing* ou "sob demanda". É um novo modelo de negócio que utiliza as mais recentes tecnologias, como virtualização e multilocação. Ambos os serviços são utilizados para obter vantagens de economia de escala e reduzir o custo dos recursos de TI. Exemplo geral dos serviços em nuvem é o Google Apps® provido pela Google® e Microsoft SharePoint® (SHAIKH; HAIDER, 2011).

A plataforma *Cloud Computing* já foi implantada para os usuários em todo o mundo compartilharem os benefícios da computação em nuvem. Atualmente ela é utilizada por algumas das grandes empresas da *Internet*, como Google®, IBM®, Microsoft®, etc. (ZHENGQIAO; DEWEI, 2012).

Os sistemas de computação em nuvem emergentes, como Amazon® EC2 (*Elastic Compute Cloud*) é a recente aplicação de centros de dados (KANAGASABAI et al., 2013). Eles têm alto potencial para permitir a criação de mercados que virtualizam mais nuvens de diferentes fornecedores, atraindo clientes interessados em negócios voltados a computação em grade (BUYYA; SULISTIO, 2008).

Outro exemplo de aplicação de *Cloud Computing* é encontrado no Windows Azure®, uma plataforma de computação em nuvem extensível e aberta que fornece os serviços para desenvolver, implantar e operar aplicações e serviços em centros de dados em nuvem. Esta plataforma é simples, generalizada, e poderosa para a criação de aplicações *Web* e serviços. No que diz respeito ao armazenamento de dados, vários serviços de armazenamento de arquivos *on-line* estão disponíveis no servidor de nuvem para aumentar o potencial de armazenamento dos dispositivos de cliente, como AmazonS3®, o Google Docs®, o MobileMe® e DropBox® (SHIRAZ; GANI; KHOKHAR; BUYYA, 2013).

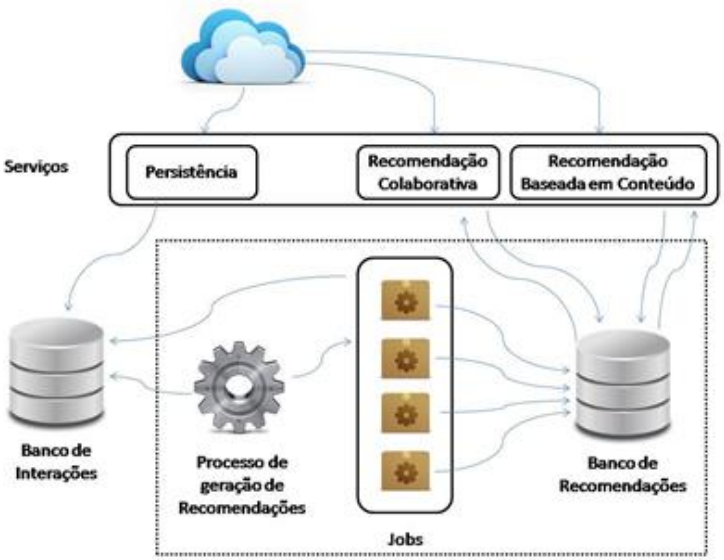
4. SISTEMA PROPOSTO

Neste capítulo será apresentado o sistema proposto. A apresentação será realizada em duas etapas. A primeira etapa refere-se ao modelo lógico em que são detalhadas as interação entre os módulos componentes. A segunda etapa representará o modelo físico, descrevendo os componentes tecnológicos, bem como, a justificativa da utilização dos mesmos.

4.1 MODELO LÓGICO

O modelo lógico representado pela Figura 5 é composto por camadas que possibilitam a interconexão de conteúdo, objetivando prover a recomendação destes conteúdos aos usuários. Na Figura 5 encontra-se a nuvem como ponto inicial. Esta nuvem representa a *Web*, bem como seus usuários e o conteúdo gerado e consumido por estes dentro de um ambiente.

Figura 5 – Modelo lógico do sistema proposto.



Fonte: Autores.

Em um nível abaixo à nuvem, encontra-se a camada de serviços que proverá a interação entre a arquitetura proposta e a *Web*. Esta camada é formada pelos serviços de persistência de dados e fornece acesso às recomendações colaborativas e recomendações baseadas em conteúdo.

As interações dos usuários da *Web* dentro deste ambiente específico são coletadas e persistidas em uma base de dados. Inicialmente, os dados coletados são persistidos no Banco de Interações.

Na Figura 5 o sistema proposto é destacado com uma área pontilhada sendo esta o foco do trabalho. As demais partes constam apenas para apresentar a necessidade das informações básicas (interações geradas pelos usuários), bem como os serviços que irão consumir as recomendações.

Na área delimitada encontra-se o processo de geração de recomendações. Como o nome identifica, este processo gerará as recomendações baseadas em conteúdo e recomendações baseadas em filtragem colaborativa. O processo de geração de recomendação, localiza na base de interações os itens ainda não processados ou que sofreram atualizações.

Uma vez de posse de um item específico, o processo de geração de recomendações acionará um ou mais *Jobs* responsável pela geração das recomendações. O conjunto de possíveis *Jobs* utilizados no processo complementam o sistema proposto representando um espaço destinado ao processamento distribuído voltado a execução da tarefa de geração de recomendação. Cada *Job*, de posse de um ou mais itens, executa os cálculos de similaridade relacionando os itens. Ao final as informações são persistidas em uma base de dados denominada de Banco de Recomendações.

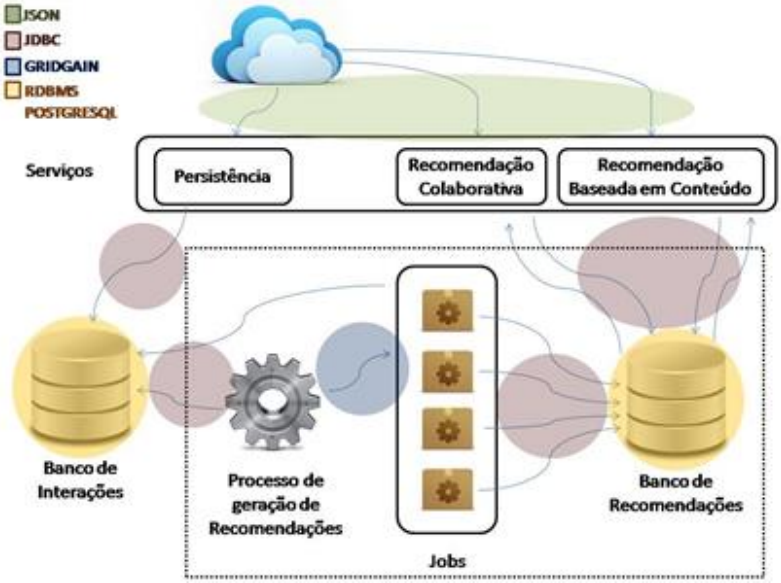
O Banco de Recomendações possuirá os dados já tratados pelo processo de geração de recomendações auxiliado pelos *Jobs*. Ao final de cada verificação de quais interações devem ser processados e dos cálculos necessários para realizar as associações, as informações ficam disponíveis para que a camada de serviços possa obter as recomendações através das abordagens colaborativas ou baseadas em conteúdo.

4.2 MODELO FÍSICO

O modelo físico (Figura 6) apresentara em mais detalhes os componentes tecnológicos e como ocorrem as interações entre eles com

o objetivo de proporcionar uma visão mais detalhada do sistema proposto.

Figura 6 – Modelo físico do sistema proposto.



Fonte: Autores.

Os componentes tecnológicos estão destacados na Figura 6 e identificados através das cores de acordo com a legenda. Conforme ilustrado, a comunicação entre a aplicação que interage com o usuário localizada na *Web* e a camada de serviços é efetivada através de mensagens no padrão *JavaScript Object Notation* (JSON) e é identificada através da área destacada em verde.

As bases de dados destacadas na cor amarela, foram implementadas utilizando o banco de dados relacional PostgreSQL®, por ser um banco de dados gratuito e de ampla aceitação no mercado.

O processo de geração de recomendações foi escrito na linguagem de programação Java®, sendo que esta escolha se deu pelos mesmos motivos da utilização do banco de dados PostgreSQL®.

Para realizar a comunicação, entre as bases de dados com a camada de serviços e com o processo de geração de recomendações, foi aplicada a tecnologia *Java Database Connectivity* (JDBC). Esta tecnologia fornece suporte à manipulação de informações através da

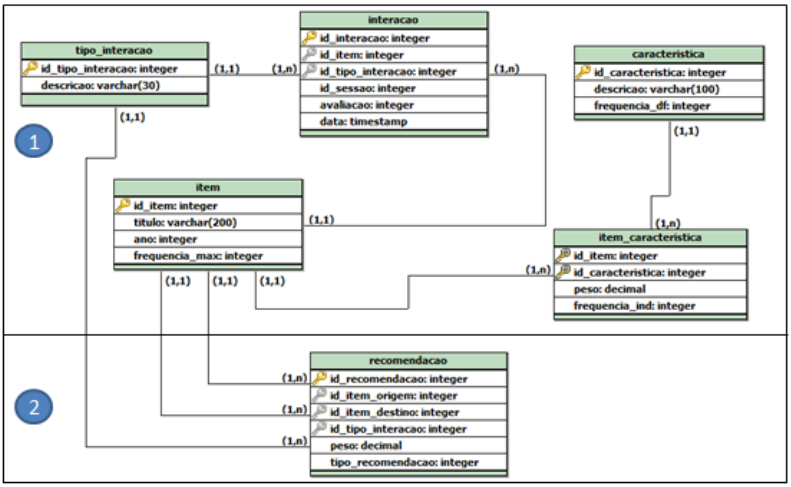
execução de comandos escritos na linguagem SQL, sendo uma especificação implementada na linguagem de programação Java®. As interações são identificadas por meio das regiões destacadas com a cor lilás.

Com o objetivo de aumentar o poder de processamento, o sistema foi desenvolvido com suporte do *middleware* GridGain® para distribuir os *Jobs* entre os nós do *cluster*. A aplicação do GridGain®, na Figura 6, é destacada pela região na cor azul. Sua escolha se deve ao fato da mesma ser gratuita, possuir ampla aceitação e ser de fácil utilização na implementação de sistemas distribuídos complexos.

4.2.1 Modelo de Dados

O modelo de dados representado pela Figura 7 pode ser agrupado em duas camadas. Na primeira camada (identificada pelo número 1) estão localizadas as tabelas necessárias para armazenar as interações geradas pelo usuário através de uma aplicação qualquer (tabela INTERACAO); os tipos de interação possíveis, sendo consulta de documento e avaliação de documento (tabela TIPO_INTERACAO); os itens que fazem parte do domínio em questão, neste caso, representados por documentos (tabela ITEM); as características dos itens, representado pelas palavras-chave dos documentos (tabela CARACTERISTICA); e a associação entre itens e características (tabela ITEM_CARACTERISTICA). Essa primeira camada foi definida no trabalho de Laurindo e André (2014).

Figura 7 – Modelo de dados.



Fonte: Autores.

Na segunda camada (identificada pelo número 2) encontra-se a tabela (RECOMENDACAO) que armazena as recomendações geradas. Para esta tabela será necessário um maior detalhamento, pois a mesma representa a fonte de busca das recomendações. A utilização desta tabela ocorre na última etapa do processo de tratamento das informações que resultarão em recomendações.

A tabela de recomendações é composta pelos seguintes campos: ID_RECOMENDACAO, campo identificador da tabela (chave primária) que rotula determinada recomendação. O campo ID_ITEM_ORIGEM representa a chave estrangeira que identifica o item que gerou a recomendação. O campo ID_ITEM_DESTINO também representa uma chave estrangeira que se refere ao item chamado de destino que está associado ao item de origem. O campo ID_TIPO_INTERACAO, define se a interação foi uma avaliação do item ou uma visualização. Por sua vez, o campo PESO define o grau de similaridade entre os itens que formam uma recomendação. Finalmente o campo TIPO_RECOMENDACAO define se a recomendação é baseada em conteúdo ou filtragem colaborativa. De modo geral, dado um item de interesse, localiza-se este item na origem ou destino e recomendasse todos os itens associados, ou seja, itens que irão aparecer na origem ou destino, mas diferentes do item de interesse.

4.2.2 Geração de Recomendações

As recomendações são geradas a partir das interações realizadas pelos usuários sob a aplicação, estando armazenadas no banco de dados de interações. As interações são resultantes do acesso ao item para uma análise mais detalhada ou da avaliação do item (LAURINDO; ANDRÉ, 2014). A avaliação ocorre considerando uma escala de importância variando de 1 (um) a 5 (cinco). Estas informações promove suporte às recomendações baseadas em filtragem colaborativa.

Além das interações também são armazenados os itens e suas características. Estas informações foram o insumo para a geração das recomendações baseadas em conteúdo. Neste sentido, após o processamento, dado um item serão recuperados os itens mais similares considerando as características que estes compartilham.

Para o desenvolvimento do sistema proposto foi utilizada a linguagem de programação orientada a objetos Java®. Para o armazenamento dos dados manipulados neste modelo foi utilizado o banco de dados relacional PostgreSQL®. A conexão entre a aplicação e o banco de dados relacional, foi feita através de um conjunto de interfaces e classes desenvolvidas em Java® que estabelecem comunicação com a base de dados através do envio de comandos na linguagem SQL. Esta conexão é chamada de *Java Database Connectivity* (JDBC).

Devido ao custo de processamento computacional em situações que potencialmente possam envolver a manipulação de muitos dados torna-se necessário recorrer ao uso de processamento distribuído. Diante desta necessidade utilizou-se um *software middleware JVM - based* (*Java Virtual Machine-based*) chamado GridGain®. Este *software* permite desenvolver a computação intensiva de dados e aplicações distribuídas de alto desempenho. O GridGain® trabalha com *Tasks* ou tarefas literalmente traduzido para a língua portuguesa. Estas *Tasks* organizam as informações que devem ser processadas para então serem alocadas em *Jobs* que funcionam como pacotes de informações. Estes *Jobs* são organizados em uma estrutura de fila e são enviados para os nós processadores do *cluster* ou grade. Cada nó detentor de um *Job* efetua o devido processamento das informações. Após o processamento o *Job* é retornado com a informação resultante para a *Task* que o remeteu ao nó processador.

5. DESENVOLVIMENTO E ANÁLISE DOS RESULTADOS

No presente capítulo será realizado um maior detalhamento do desenvolvimento do protótipo, explorando com maior profundidade os processos de geração das recomendações e seus cálculos. Também será apresentado o cenário de aplicação do protótipo, exemplos de recomendações e o detalhamento do processo distribuído de geração de recomendações.

5.1 CENÁRIO ELABORADO

O cenário de aplicação do protótipo se assemelha ao cenário proposto por Sérgio (2013) no que tange ao suporte para a geração de recomendações baseadas em conteúdo. Como declara o autor a coleta de artigos da base de dados da revista *ScienceDirect*® foi realizada manualmente. Ao todo foram coletados 306 documentos. A escolha por esta base deu-se devido a algumas características significativas, como por exemplo, a abrangência das áreas contempladas e o reconhecimento obtido no cenário mundial por conter um grande volume de publicações.

É importante ressaltar que assim como na proposição de cenário feita por Sérgio (2013), este cenário também não contém todo o acervo de publicações da revista. Para a formação da base foram considerados artigos obtidos a partir de termos de busca, entre eles, *Biotechnology*, *Semantic Web* e *Ontology*. Preservou-se o idioma original dos termos para manter a concordância com a pesquisa aplicada na língua inglesa.

A partir de cada documento coletado, foram extraídos os dados relevantes para gerar a meta informação que foi estruturada no formato XML, conforme ilustrado na Figura 8. Estas informações foram utilizadas na carga das Tabelas Item, Caracteristica e Item_Caracteristica do modelo. Os dados que compõe a estrutura XML são:

- <ID>: Identificador do documento constituído por um número sequencial;
- <TITLE>: Título do documento;
- <YEAR>: Ano da publicação do artigo;
- <AUTHORS>: Nome(s) do(s) autor (es);
- <KEYWORDS>: Preenchido com as palavras - chave do artigo.

Figura 8 – Estrutura do XML.

```
<DOCUMENT
        ID = ""
        TITLE = ""
        YEAR = "" >

        <AUTHORS>
                <ITEM NAME = "" ORGANIZATION = ""/>
        </AUTHORS>

        <KEYWORDS>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
                <ITEM NAME = ""/>
        </KEYWORDS>
</DOCUMENT>
```

Fonte: Sérgio (2013).

Para permitir a geração de recomendações de filtragem colaborativa utilizou-se os dados gerados no trabalho de Laurindo e André (2014). As interações foram produzidas através de um sistema, que seguindo algumas definições, preenche a tabela de interações. Cada entrada na tabela representa uma interação com determinado item, em uma sessão em particular e um tipo específico de interação (consulta do item que representa um documento ou avaliação do item).

Deste modo, uma sessão pode ter múltiplas interações, em que cada sessão representa um possível usuário. Ao todo foram geradas 149.000 interações distribuídas em 10.000 sessões e 306 itens.

5.2 DETALHAMENTO DOS CÁLCULOS

Nesta seção serão detalhados os cálculos realizados pelo sistema proposto de maneira distribuída para a aplicação da filtragem colaborativa e a filtragem baseada em conteúdo.

5.2.1 Cálculo para a Filtragem Colaborativa

A filtragem colaborativa tem como foco gerar recomendações com base na análise de classificações produzidas por usuários com perfil

ou preferências semelhantes ao qual a recomendação se destina. Devido ao crescimento maior de usuários em relação a quantidade de itens optou-se por utilizar a abordagem colaborativa baseada em item.

Neste trabalho, o usuário é substituído por uma sessão, que é caracterizada por um período de tempo envolvendo as ações do usuário em um ambiente *Web*. Durante a sessão o usuário fará avaliações aos documentos, genericamente tratados como itens. Estas avaliações variam de 1 a 5, sendo 1 a menor avaliação e 5 a maior avaliação. Quando um item não receber avaliações em uma sessão, seu valor de avaliação será substituído pelo sinal de interrogação (?).

Para gerar as recomendações, deve-se inicialmente buscar as informações das avaliações dos itens. A organização destas informações resulta em uma matriz esparsa. A Tabela 3 ilustra com dados fictícios das avaliações de 5 itens em 5 sessões.

Tabela 3 - Dados iniciais para o cálculo de recomendação (SESSÃO x ITEM).

	Item 1	Item 2	Item 3	Item 4	Item 5
Sessão 1	5	3	?	4	?
Sessão 2	0	1	2	3	3
Sessão 3	4	3	?	3	?
Sessão 4	3	?	1	?	4
Sessão 5	1	?	?	2	1

Fonte: Autores.

Nesta etapa deve considerar que as avaliações realizadas por usuário podem denotar tendências comportamentais. Enquanto alguns usuários tendem a realizar somente avaliações elevadas, outros possuem a tendência inversa e realizam apenas classificações baixas. Visando minimizar isto deve-se ponderar cada avaliação pela média das avaliações efetuadas em uma sessão.

Com base na Tabela 3, foram calculadas as médias de classificações das sessões (Tabela 4).

Tabela 4 - Média das avaliações por sessão.

	Média
Sessão 1	4,00
Sessão 2	1,80
Sessão 3	3,33
Sessão 4	2,67
Sessão 5	1,33

Fonte: Autores.

A próxima etapa consiste em montar a matriz ajustada, onde será substituída a avaliação do item pelo resultado obtido ao subtrair a avaliação inicial da sessão para determinado item do valor da média da sessão. Por exemplo, a sessão 1 possui como média de avaliações o valor 4, conforme Tabela 4, e a avaliação inicial desta sessão para o item 1 é o valor 5, conforme Tabela 3. Sendo assim, na matriz ajustada, a avaliação da sessão 1 para o item 1 será agora o valor 1.

A seguir a Tabela 5 exibe a matriz ajustada para as avaliações utilizadas no exemplo ilustrado na Tabela 3.

Tabela 5 - Matriz de avaliações ajustada pela média.

	Item 1	Item 2	Item 3	Item 4	Item 5
Sessão 1	1,00	-1,00	-	0,00	-
Sessão 2	-1,80	-0,80	0,20	1,20	1,20
Sessão 3	0,67	-0,33	-	-0,33	-
Sessão 4	0,33	-	-1,67	-	1,33
Sessão 5	-0,33	-	-	0,67	-0,33

Fonte: Autores.

Tendo a matriz ajustada preenchida, pode-se então aplicar o cálculo do cosseno, conforme a seção 2.1.1.2, onde será comparado o primeiro vetor com os outros 4 (quatro) vetores, o segundo vetor com os outros 3 (três) vetores, seguindo assim até a comparação do quarto vetor com o quinto. Como resultado final tem-se uma matriz ITEM x ITEM representada pela Tabela 6 onde cada posição identifica a similaridade entre os itens.

Tabela 6 - Matriz de recomendação (ITEM x ITEM).

	Item 1	Item 2	Item 3	Item 4	Item 5
Item 1	-	0,07	-0,24	-0,83	-0,40
Item 2	0,07	-	-0,07	-0,45	-0,39
Item 3	-0,24	-0,07	-	0,10	-0,64
Item 4	-0,83	-0,45	0,10	-	-0,47
Item 5	-0,40	-0,39	-0,64	-0,47	-

Fonte: Autores.

Após cada cálculo as informações são persistidas na tabela RECOMENDACAO. Por exemplo, se determinado nó estiver calculando a similaridade entre o Item 1 e o Item 2 e o cosseno for maior do que zero será incluída uma linha na tabela em que o Item 1 será registrado no campo de origem e o Item 2 no campo de destino. Adicionalmente, são incluídos o peso (resultado do cálculo do cosseno) que conforme o exemplo acima é 0,72 e as demais informações, com destaque para tipo de recomendação que neste caso é indicado pelo código 1 (filtragem colaborativa).

Após o cálculo do cosseno entre todos os itens de maneira distinta e efetuado o armazenamento da informação na tabela RECOMENDACAO pode-se então obter os itens semelhantes a partir de um item de interesse (no contexto do trabalho representado por um documento). Exemplos de consulta serão detalhados na seção 5.3.1.

5.2.2 Baseado em Conteúdo

Como já mencionado no capítulo 2 do presente trabalho, a Filtragem Baseada em Conteúdo originou-se na Recuperação de Informação (RI) e nos filtros de pesquisas de informação. A filtragem Baseada em conteúdo tem como foco gerar recomendações com base na analise de similaridade entre as características dos itens. Neste caso, itens semelhantes a um item em específico, serão sugeridos ao usuário ao qual a recomendação se destina.

A Tabela 7 exemplifica com dados fictícios a transformação do item em um vetor. Cada item agora transformado possui uma quantidade de termos. Para cada termo é contabilizado o número de ocorrências deste termo no documento.

Tabela 7 - Dados iniciais representando itens e suas características.

	Caract. 1	Caract. 2	Caract. 3	Caract. 4	Caract.5
Item 1	2	0	1	0	4
Item 2	0	1	2	3	0
Item 3	1	3	2	0	0
Item 4	2	4	0	0	2
Item 5	0	0	1	2	3

Fonte: Autores.

Uma vez preenchida a tabela de ocorrência ITEM x CARACTERÍSTICA, o passo seguinte será o cálculo do peso que define a importância desta característica (temo) relacionada ao item. Para tal, será aplicado o método estatístico numérico TF-IDF. Onde *TF* é relativo a frequência que o termo *t* é encontrado no item (documento) *d*, divido pela máxima frequência entre todas as características encontradas no item *d*, sendo representado pela equação:

$$TF(t,d) = \frac{freq(t,d)}{maxFreq(t,d)}$$

A segunda etapa do método é responsável por encontrar o valor de IDF que representa a frequência inversa dos itens. Este valor é encontrado pelo logaritmo do quociente obtido através da divisão do total de itens *D* pelo total de itens em que a característica ocorre (d(t)). Esta etapa por sua vez pode ser representada pela equação:

$$IDF(t) = \log_2 \frac{D}{d(t)}$$

Sendo assim, o método TF-IDF, pode ser representado pela equação:

$$TF - IDF(t,d) = \left(\frac{freq(t,d)}{maxFreq(t,d)}\right) * \left(\log_2 \frac{D}{d(t)}\right)$$

Como exemplo de aplicação da equação acima, utilizou-se os dados encontrados na característica 1 referente ao item 1 ilustrados na

Tabela 7. Tem-se então a equação com os seguintes dados:

$$TF - IDF(t,d) = \left(\frac{2}{4}\right) * \left(\log_2 \frac{5}{3}\right) = 0{,}3685$$

Aplicando a equação a todas as características e itens contidos na Tabela 7 tem-se a Tabela 8 preenchida com todos os valores de cada característica.

Tabela 8 – Matriz de pesos dos itens e suas características.

	Caract. 1	Caract. 2	Caract. 3	Caract. 4	Caract.5
Item 1	0,3685	0	0,0805	0	0,7370
Item 2	0	0,1842	0,1610	0,9914	0
Item 3	0,1842	0,5527	0,1610	0	0
Item 4	0,3685	0,7370	0	0	0,3685
Item 5	0	0	0,0805	0,6610	0,5527

Fonte: Autores.

Depois de calculado o peso, aplica-se o cálculo do cosseno para todas as possibilidades distintas de itens de maneira distribuídas, ou seja, cada nó do *cluster* irá receber um intervalo de itens para calcular a similaridade. Além disso, cada nó receberá a lista de todos os itens que devem ser comparadas. Por exemplo, o primeiro nó recebe toda a lista de itens e o intervalo que deve ser processado. Aplica-se então o cálculo do cosseno conforme a seção 2.1.1.2, onde será comparado o primeiro vetor com os outros nove vetores, o segundo vetor com os outros oito vetores, seguindo assim até a comparação do novo com o décimo vetor. Como resultado final tem-se uma matriz ITEM x ITEM, representado pela Tabela 9, onde cada posição identifica a similaridade entre os itens.

Tabela 9 – Matriz com os pesos ITEM x ITEM baseada em conteúdo.

	Item 1	Item 2	Item 3	Item 4	Item 5
Item 1	-	0,17	0,16	0,55	0,58
Item 2	0,17	-	0,21	0,15	0,76
Item 3	0,16	0,21	-	0,87	0,02
Item 4	0,55	0,15	0,87	-	0,26
Item 5	0,58	0,76	0,02	0,26	-

Fonte: Autores.

Apesar da representação matricial, visando o desempenho, após cada cálculo as informações são persistidas na tabela RECOMENDACAO. Por exemplo, se determinado nó estiver calculando a similaridade entre o Item 1 e o Item 2 e o cosseno for maior do que zero será incluída uma linha na tabela em que o Item 1 será registrado no campo de origem e o Item 2 no campo de destino. Adicionalmente, são incluídos o peso (resultado do cálculo do cosseno) que conforme o exemplo acima é 0,72 e as demais informações, com destaque para o tipo de recomendação que neste caso é indicado pelo código 2 (baseada em conteúdo).

Após o cálculo do cosseno entre todos os itens de maneira distinta e efetuado o armazenamento da informação na tabela RECOMENDACAO pode-se então obter os itens semelhantes a partir de um item de interesse (no contexto do trabalho representado por um documento). Exemplos de consulta serão detalhados na seção 5.3.1.

5.3 EXEMPLOS DE RECOMENDAÇÃO

Nesta seção será abordada com maiores detalhes a busca pelas recomendações geradas pela filtragem baseada em conteúdo e filtragem colaborativa.

5.3.1 Consulta sobre a tabela de interação

A visualização das informações persistidas na tabela de interação pode ser feita através de consultas utilizando a linguagem SQL. De maneira mais genérica, a busca por recomendações geradas para um determinado item podem ser realizadas com a execução do código SQL ilustrado na Figura 9. Neste caso utilizou-se como exemplo a busca por todas as recomendações existentes para o item 10, onde as recomendações foram geradas com base neste item (ID_ITEM_ORIGEM) e também as recomendações que este foi apontado como semelhante (ID_ITEM_DESTINO).

Figura 9 – Comando SQL genérico para buscar recomendações de um item.

```
1  select *
2    from recomendacao
3   where id_item_origem  = 10
4      or id_item_destino = 10
```

Fonte: Autores.

A busca pelas recomendações relacionadas ao item 10 resulta no retorno exibido na Figura 10. Devido ao grande volume de dados serão exibidas as cinco recomendações iniciais.

Figura 10 – Retorno genérico das recomendações.

Data Output | Explain | Messages | History

	id_recomendacao integer	id_item_origem integer	id_item_destino integer	id_tipo_interacao integer	peso numeric	tipo_recomendacao integer
1	9	1	10	0	0.156847221691832	2
2	76	2	10	0	0.282579289867896	2
3	146	3	10	0	0.304757583056233	2
4	215	4	10	0	0.250506444383714	2
5	283	5	10	0	0.371676110710556	2

Fonte: Autores.

Ainda que executada com sucesso, a busca genérica deve ser aprimorada, pois até o momento as recomendações para o item em questão estão desordenadas e não especificam qual o tipo de filtragem que às originou, bem como o tipo de interação quando ocorrido. Estas especificações podem ser obtidas através da execução do comando SQL exibido na Figura 11. Nesta etapa, o resultado será ordenado de maneira decrescente pelo peso da recomendação. Também é incluída a opção de limite de resultados de acordo com a necessidade do usuário. A aplicação específica deste comando SQL, assim como seu resultado, será abordado nos tópicos seguintes do presente trabalho.

Figura 11 – Comando SQL aprimorado para buscar recomendações de um item.

```
select  item,
        titulo,
        ano,
        peso
  from ( select a.id_item_destino item,
                b.titulo,
                b.ano,
                a.peso
           from recomendacao a,
                item b
          where a.id_item_origem = ?
            and a.id_tipo_interacao = ?
            and a.tipo_recomendacao = ?
            and a.id_item_destino = b.id_item
          union
         select a.id_item_origem item,
                b.titulo,
                b.ano,
                a.peso
           from recomendacao a,
                item b
          where a.id_item_destino = ?
            and a.id_tipo_interacao = ?
            and a.tipo_recomendacao = ?
            and a.id_item_origem = b.id_item
        ) c
 order by peso desc
limit?
```

Fonte: Autores.

5.3.1.1 Filtragem colaborativa

Para obter o resultado da consulta sobre as interações geradas por filtragem colaborativa, deve-se aplicar no script de consulta SQL os seguintes ajustes: O campo TIPO_RECOMENDACAO deverá ser igualado ao valor 1, que representa a filtragem colaborativa. O campo ID_TIPO_INTERACAO deverá ser igualado ao valor 1 para quando necessitar encontrar interações do tipo visualização de documentos. Quando desejado visualizar interações do tipo avaliação de documentos, o campo ID_TIPO_INTERACAO deverá ser igualado ao valor 2. A seguir, a Figura 12 ilustra o resultado obtido ao buscar os cinco itens mais recomendados com base no item 10, utilizando a opção de filtragem colaborativa e considerando apenas as interações de avaliação de usuários.

Figura 12 – Retorno para filtragem colaborativa baseado em avaliações.

Data Output | Explain | Messages | History

	item integer	titulo character varying(200)	ano integer	peso numeric
1	140	Pragmatic applications of the Semantic Web using SemTalk	2003	0.127202896764164
2	218	Retreating Recurrent Breast Cancer with the same CMF-cont	1997	0.118428630884246
3	117	Knowledge Management: a Strategic Agenda	1997	0.115064225231012
4	63	DATA MINING IN FINANCE: USING COUNTERFACTUALS TO GENERATE	1998	0.104167077388026
5	150	Extracting focused knowledge from the semantic web	2001	0.0935579305595056

Fonte: Autores.

5.3.1.2 Filtragem baseada em conteúdo.

Para obter o resultado da consulta sobre as interações geradas por filtragem baseada em conteúdo, deve-se aplicar na consulta SQL os seguintes ajustes: o campo TIPO_RECOMENDACAO deverá ser igualado ao valor 2 que representa a filtragem baseada em conteúdo. O campo ID_TIPO_INTERACAO deverá ser igualado ao valor 0, pois na filtragem baseada em conteúdo não existem interação com o usuário. A seguir, a Figura 13 ilustra o resultado obtido ao buscar os cinco itens mais recomendados com base no item 10, utilizando a filtragem baseada em conteúdo.

Figura 13 – Retorno para filtragem baseada em conteúdo.

Data Output | Explain | Messages | History

	item integer	titulo character varying(200)	ano integer	peso numeric
1	5	Biotechnology, a new industrial revolution	1981	0.371676110710556
2	27	Genetic Engineering and Related Biotechnologies	1983	0.347338457791684
3	23	Contribution of molecular biology to bioremediation□	1991	0.340374520482045
4	3	Biotechnology and the Law: Recombinant DNA and the Contro	1979	0.304757583056233
5	2	Biotechnology - pulling the threads together	1979	0.282579289867896

Fonte: Autores.

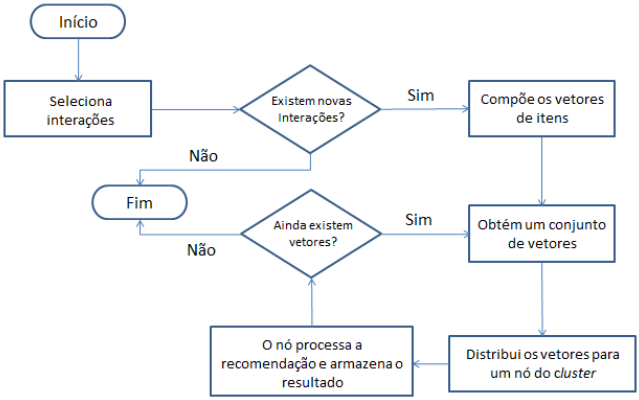
5.4 PROCESSAMENTO DISTRIBUÍDO

Esta seção apresenta os aspectos envolvidos no desenvolvimento do protótipo no que tange a distribuição do processamento para a geração das recomendações. Para tal, o protótipo foi desenvolvido com o auxílio do *middleware* GridGain® versão 4.0.2c.

5.4.1 Filtragem Colaborativa

O processamento distribuído das recomendações ocorre conforme a Figura 14. Primeiramente, a tarefa verifica se existem interações ainda não processadas. Em caso afirmativo ocorre o processo de formação dos vetores.

Figura 14 – Fluxo de execução da tarefa de geração de filtragem colaborativa.



Fonte: Autores.

No caso da filtragem colaborativa cada vetor representa um item e cada dimensão do vetor uma sessão em que este teve determinada interação, seja de consulta ou de avaliação de um documento. Conforme apresentado na seção 5.1.1 as dimensões devem ser normalizadas pela média das interações de cada sessão.

Após isso, seleciona-se um conjunto de vetores considerando um fator *k*, ou seja, um número utilizado para determinar o intervalo inicial e final de vetores que devem ser processados pelos nós (nodos) do *cluster*. Esta estratégia é fundamental, pois a comunicação frequente com os nós do *cluster* gera tráfego excessivo na rede degradando o desempenho.

De posse dos vetores estes são enviados pela tarefa a um nó para que o mesmo execute o serviço (*Job*). O serviço tem como objetivo comparar os vetores recebidos e verificar a similaridade entre os mesmos. Vetores que possuem similaridade superior a zero, considerando a equação do cosseno, são registrados na tabela de recomendações.

Depois de enviar um conjunto de vetores para serem processados, a tarefa fica esperando que determinado nó esteja novamente disponível. Em havendo ainda vetores os mesmos são submetidos até o limite estabelecido por *k*. Caso não existam mais vetores o processamento é finalizado.

Como resultado o processo gera múltiplas linhas na tabela de recomendações considerando os identificadores dos vetores (vetor que representa o item origem e vetor que representa o item destino), o tipo de interação, o peso da comparação (similaridade dos vetores) e o tipo de recomendação, neste caso, tipo 1 (filtragem colaborativa).

5.4.1.1 Execução do processamento

Os testes de execução do protótipo, considerando a característica de cálculo distribuído, foram efetuados em um *cluster* composto por três computadores com Sistema Operacional Windows 7, sendo 2 computadores com processador I5, de 1,40GHz e 4GB de memória RAM e um computador com processador I7 com 6GB, de 2.40GHz e 6GB de memória RAM.

A Tabela 10 apresenta os tempos de execução considerando 1, 2 e 3 nós do *cluster*.

Tabela 10 – Tempos de execução da tarefa de filtragem colaborativa.

Nós	Tempo Médio (Segundos)	Redução no Tempo (%)
1	09:756	-
2	07:211	26,1%
3	06:620	32,1%

Fonte: Autores.

Como pode ser analisado, a distribuição do cálculo da abordagem de filtragem colaborativa promove um ganho de desempenho. Ainda que o decréscimo do tempo não seja linear, a medida que novos nós são incluídos no *cluster*, pode-se verificar uma melhora do tempo médio de execução. A execução com dois nós promove uma redução de 26,1% no tempo, enquanto a execução com três nós possui uma redução de 32,1% em relação a execução com apenas um nó.

Vale ressaltar que os tempos poderiam ser melhorados caso fosse implementada uma estratégia de balanceamento de carga. Nos testes, o

valor de *k* foi determinado pelo total de itens dividido pelo total de nós do *cluster* para determinada execução. Considerando 306 itens disponíveis na base de dados nas três execuções, *k* teve os valores 306, 153 e 102, respectivamente.

Visto que o total de associações geradas a partir de um conjunto de vetores recebidos por determinado nó pode ser maior em relação a outro nó do *cluster* e que o *cluster* não possui homogeneidade em relação aos computadores, os tempos computados representam a média dos 5 (cinco) maiores tempos independente de nó. Por exemplo, considerando o teste com 3 nós no *cluster*, na primeira execução o nó 1 leva mais tempo para executar, na segunda execução o nó 3 leva mais tempo para executar, e assim por diante.

Ao final, foram gerados 10753 associações na tabela de recomendações. Considerando 306 itens (vetores representando documentos) o total de possibilidades distintas seria de mais de 46.000. Isto não ocorre uma vez que nem todos os vetores possuem associações.

5.4.2 Recomendação Baseada em Conteúdo

O processamento distribuído para o cálculo de recomendações baseada em conteúdo segue o mesmo fluxo apresentado na

Figura 14. A principal diferença reside na elaboração dos vetores.

Ao contrário do modelo anterior em que os vetores eram compostos a partir das interações ocorridas nas sessões, esta abordagem leva em consideração as características dos itens (documentos no cenário deste trabalho).

Para tal, considerando cada item (documento) as suas dimensões (características) são normalizadas através da equação do TF-IDF. Após isso os vetores são distribuídos entre os nós que então calculam as similaridades e armazenam o resultado na tabela de recomendações.

5.4.2.1 Execução do processamento

Os testes de execução do protótipo, considerando a característica de cálculo distribuído para a presente tarefa, foram executados em um *cluster* composto por três computadores com Sistema Operacional Windows 7, sendo 2 com processador I5 com 4GB de memória RAM e um com processador I7 com 6GB de memória RAM.

A Tabela 11 apresenta os tempos de execução considerando 1, 2 e 3 nós do *cluster*.

Tabela 11 – Tempos de execução da tarefa de recomendação baseada em conteúdo.

Nós	Tempo Médio (Segundos)	Redução no Tempo (%)
1	04:239	-
2	03:159	25,5%
3	02:776	34,5%

Fonte: Autores.

O comportamento dos cálculos para a tarefa de recomendação baseada em conteúdo é similar a abordagem de filtragem colaborativa. Pode-se observar que os tempos das execuções sofrem um decréscimo a medida que mais nós são adicionados. A execução com dois nós promove uma redução de 25,5% no tempo, enquanto a execução com três nós possui uma redução de 34,5% em relação a execução com apenas um nó. A determinação do valor de k seguiu a mesma estratégia do cálculo de filtragem colaborativa, ou seja, a divisão do total de itens pelo número de nós.

No caso anterior, filtragem colaborativa, a medida que novas sessões são criadas os vetores dos itens, que sofreram interações em determinada sessão, tem sua dimensionalidade alterada. Deste modo mais associações serão formadas gerando mais linhas na tabela de recomendações e aumentando o tempo de cálculo. Este fato tem menos impacto na abordagem baseada em conteúdo, pois as características dos itens tendem a não se modificarem ao longo do tempo.

Ao final, foram geradas 5558 associações na tabela de recomendações. Considerando 306 itens (vetores representando documentos) o total de possibilidades distintas seria de mais de 46.000. Como mencionado anteriormente, isto não ocorre porque nem todos os vetores possuem associações.

6. CONSIDERAÇÕES FINAIS

O objetivo geral desse trabalho foi propor um sistema voltado à recomendação de conteúdo textual suportado pela computação distribuída.

Buscando atingir este objetivo, realizou-se inicialmente o levantamento bibliográfico nas áreas de pesquisa envolvidas no trabalho, citam-se os Sistemas de Recomendação e a Computação Distribuída. Com os conhecimentos adquiridos nesta etapa, foi possível elaborar a base de dados e desenvolver um protótipo voltado à recomendação de conteúdos textuais.

Para a aplicação da filtragem baseada em conteúdo foi utilizado uma base de dados composta por centenas de documentos conforme apresentado na seção 5.2 em que cada documento é composto por um conjunto de características (palavras-chave). De posse destas informações foi possível realizar as recomendações baseadas em conteúdo.

Para as recomendações geradas pela filtragem colaborativa foi necessário verificar as interações (consulta de documentos e avaliação) dos usuários com os documentos de texto. As interações foram geradas de maneira aleatória conforme discutido na seção 5.2. As interações persistidas em uma base de dados serviram posteriormente como insumo para a execução dos cálculos que geraram as recomendações nesta abordagem.

O desenvolvimento do protótipo teve como objetivo aplicar os conceitos de Recomendação baseada em Filtragem Colaborativa e Recomendação Baseada em Conteúdo pertencentes aos Sistemas de Recomendação e auxiliados pela Computação Distribuída. O protótipo foi executado em um ambiente onde existia uma base de dados com centenas de documentos e milhares de interações. O protótipo foi desenvolvido com o auxilio de um *software middleware* chamado GridGain® que propicia suporte ao processamento distribuído.

Através dos resultados obtidos foi possível constatar o êxito na geração das recomendações originadas pela filtragem colaborativa através da avaliação de usuários e a filtragem baseada em conteúdo através da analise entre os termos contidos nos documentos. O sistema demonstrou resultados satisfatórios quanto ao auxilio da computação distribuída relacionado ao aumento no poder de processamento.

No desenvolvimento do trabalho foram encontradas algumas limitações relacionadas a realização do mesmo. Destaca-se o volume do

conteúdo da base de dados utilizada que foi restrita a um número específico de documentos, pois a formação de uma base de dados textual volumosa depende de processos de coleta, muitas vezes bastante especializado. Quanto ao protótipo, não foi possível testá-lo em uma estrutura de *cluster* mais robusta composta por vários computadores (nós).

O desenvolvimento do trabalho levantou possibilidade para trabalhos futuros. Dentre estas opções, destaca-se o volume de dados relacionados à interação dos usuários com o protótipo, podendo ser desenvolvida uma aplicação que gere um grande volume de informações, permitindo assim, a realização de testes mais precisos envolvendo o conceito de computação distribuída.

A evolução do protótipo poderia ter algumas características interessantes. Na versão atual o mesmo precisa ser executado manualmente. Em uma aplicação real deveria se pensar em um módulo executando como um serviço, onde em determinados períodos de tempo o mesmo pudesse avaliar as novas interações e documentos, visando a execução constante do processo de geração de recomendações.

Por fim, o foco do trabalho esteve na recomendação de documentos textuais. Uma possibilidade seria a adaptação do mesmo para trabalhar genericamente com qualquer tipo de item, bem como, possibilitar a geração de perfis de usuários que subsidiem recomendações mais precisas.

REFERÊNCIAS

ABDUL-FATAH, I.; MAJUMDAR, S. Performance of CORBA-Based Client-Server Architectures. **Parallel and Distributed Systems, IEEE Transactions On**, v. 13, n. 2, p.111-127, 2002.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. **IEEE Transactions on Knowledge and Data Engineering**, New York, v. 17, n. 6, p. 734-749, 2005.

APON, A.; MACHE, J.; BUYYA, R.; JIN, H. Cluster Computing in the Classroom and Integration With Computing Curricula 2001. **Education, IEEE Transactions On,** v. 47, n. 2, p.188-195, mai. 2004.

BELL, R.; KOREN, Y. Improved Neighborhood-based Collaborative Filtering, **KDD-Cup and Workshop,** ACM press, 2007.

BERNERS-LEE, T. **Information Management: A Proposal**. 1989. Disponível em: < http://www.w3.org/History/1989/proposal.html >. Acesso em: Janeiro 2014.

BOBADILLA, J.; ORTEGA, F.; HERNANDO, A.; GUTIÉRREZ, A. Recommender systems survey. **Knowledge-based Systems**. v. 46, p. 109-132, 2013.

BOUKHADRA, A.; BENATCHBA, K.; BALLA, A. HPS5DSWS: A Hybrid P2P Strategy of the Distributed Discovery Mechanism for Semantic Web Services. **P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference On**, Compiegne, p.29-36, out. 2013.

BUYYA, R.; RAMAMOHANARAO, K. An Innovative Master's Program in Distributed Computing. **IEEE Distributed Systems Online**, vol. 8, n. 1, art. p.701-1002, 2007.

BUYYA, R.; SULISTIO, A. Service and Utility Oriented Distributed Computing Systems: Challenges and Opportunities for Modeling and Simulation Communities. **Simulation Symposium, 2008. ANSS 2008. 41st Annual,** Ottawa, p. 68-81, 2008.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S.; BROBERG, J.; BRANDIC, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Generation Computer Systems,** Amsterdam, v. 25, n. 6, p. 599-616, jun. 2009.

BUYYA, R.; YEO, C. S.; VENUGOPAL, S.; BROBERG, J.; BRANDIC, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. **Future Generation Computer Systems,** Melbourne, v. 25, n. 6, p.599-616, jun. 2009.

CARRER-NETO, W.; HERNÁNDEZ-ALCARAZ, M. L.; VALENCIA-GARCÍA, R.; GARCÍA-SÁNCHEZ, F. Social knowledge-based recommender system. Application to the movies domain. **Expert Systems With Applications**, p.10990-11000, 2012.

CAZELLA, S. C.; NUNES, M. A. S. N.; REATEGUI, E. A Ciência da Opinião: Estado da Arte em Sistemas de Recomendação. In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski. (Org.). **Jornada de Atualização de Informática-JAI** 2010 – CSBC 2010. Rio de Janeiro: Puc RIO, 2010, v. 1, p. 161-216.

CHEN, Y. L.; CHENG, L. C.; CHUANG, C. N. A group recommendation system with consideration of interactions among group members. **Expert Systems with Applications**, p.2082-2090, 2008.

CHEN, R.; HUANG, Y.; BAU, C.; CHEN, S. A recommendation system based on domain ontology and SWRL for anti-diabetic drugs selection. **Expert Systems with Applications**, p.3995-4006, 2012.

CHETTY, M.; BUYYA, R. Weaving Computational Grids: How Analogous Are They with Electrical Grids? **Computing in Science and Engineering (CiSE)**, v. 4, n. 4, p. 61-71, 2002.

CHO, Y. H.; KIM, J. K.; KIM, S. H. A personalized recommender system based on web usage mining and decision tree induction. **Expert Systems with Applications**, v.23, n.3, p.329-342, 2002.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems: Concepts and Design**. 4ª ed. Boston, MA, USA: Addison Wesley Longman Publishing Co., Inc., 2005.

DADAN, Z.; MINQI, Z.; AOYING, Z. A Data Accessing Method in Distributed Massive Computing **Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference On**, Shenyang, v. 2, p.437-440, ago. 2009.

DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas Operacionais**. 3ª edição São Paulo: Pearson Prentice Hall, 2005.

EGGHE, L.; MICHEL, C. Strong similarity measures for ordered sets of documents in information retrieval. **Information Processing and Management: an International Journal**, v. 38, n. 6, p. 823-848, 2002.

ERICKSON, R. L.; GRIFFETH, N. D.; LAI, M. Y.; WANG, S. Y. Software Architecture Review for Telecommunications Software Improvement. **Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference On**, Geneva, v. 2, p. 616-620, mai. 1993.

FERREIRA, F. C.; OLIVEIRA, A. A. Os Sistemas de Recomendação na Web Como Determinantes Prescritivos na Tomada de Decisão. **Revista de Gestão da Tecnologia e Sistemas de Informação**. Brasil, p. 353-368, ago. 2012.

FOSTER, I.; KESSELMAN, C.; TUECKE, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. **International Journal Of High Performance Computing Applications**, p. 200-222. 2001.

FREEMAN, E.; ARNOLD, K.; HUPFER, S. **JavaSpaces Principles, Patterns, and Practice.** United Kingdom: Addison-wesley Longman Ltd. Essex, 1999.

FRIEDMAN, C.; RINDFLESCH, T. C.; CORN, M. Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine. **Journal of Biomedical Informatics**, p.765-773, 2013.

FURMSTON-EVANS, I.R. Successful System Architecture Development for Enterprise Wide Client Server Implementation. **Client/Server Computing. Seminar Proceedings (IEE Digest No. 1995/184), International Seminar On**, La Hulpe, v. 1, 515 p., out. 1995.

GAVALAS, D.; KONSTANTOPOULOS, C.; MASTAKAS, K.; PANTZIOU, G. Mobile recommender systems in tourism. **Journal Of Network And Computer Applications,** Philadelphia, v. 39, p.319-333, mar. 2014.

GE, Y.; XIONG, H.; TUZHILIN, A.; XIAO, K.; GRUTESER, M.; PAZZANI, M. An energy-efficient mobile recommender system. **Proceedings Of The 16th Acm Sigkdd International Conference On Knowledge Discovery And Data Mining.** Washington, p. 899-908. jun. 2010.

GENA, C.; BROGI, R.; CENA, F.; VERNERO, F. Impact of rating scales on user's rating behavior. In: **19th International Conference on User Modeling, Adaptation, and Personalization,** LNCS, vol. 6787, p. 123-134. Girona, Spain, 2011.

GHAZANFAR, M. A.; PRÜGEL-BENNETT, A. Leveraging clustering approaches to solve the gray-sheep users problem in recommender systems. **Expert Systems With Applications,** Philadelphia, v. 41, p.3261-3275, 2014.

GOLDBERG, D.; NICHOLS, D.; OKI, B.; TERRY, D. Using collaborative filtering to weave an information tapestry. **Communications of the Association of Computing Machinery**, v. 35, n. 12, p. 61-70, 1992.

GONÇALVES, A. L. **Um modelo de descoberta de conhecimento baseado na correlação de elementos textuais e expansão vetorial aplicado à engenharia e gestão do conhecimento.** Florianópolis, SC, 2006. 196 f. Tese (Doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia de Produção.

HERNANDO, A.; BOBADILLA, J.; ORTEG, F.; GUTIÉRREZ, A. Trees for explaining recommendations made through collaborative filtering. **Information Sciences,** Philadelphia, v. 239, p.1-17, ago. 2013.

HILBERT, M.; LÓPEZ, P. The World's Technological Capacity to Store, Communicate, and Compute Information. **Science Magazine**, Califórnia, v. 332, p. 60-65, abr. 2011.

HSU, F. M; LIN, Y. T.; HO, T. K. Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model, Bayesian network and Google Maps. **Expert Systems with Applications**, v. 39, n. 3, p. 3257-3264, 2012.

HUANG, C.; ZUO, M.; RONG, X. Design of Mobile Learning System Base on Cloud Computing. **Modern Educational Technology**, v.20, n.8, p.102-105, ago. 2010.

IBM. **What Is Big Data: Bring Big Data to the Enterprise**, 2014. Disponível em: http://www-01.ibm.com/software/au/data/bigdata/, Acessado: 22/01/2014.

JADEJA, Y.; MODI, K. Cloud Computing - Concepts, Architecture and Challenges. **Computing, Electronics And Electrical Technologies (ICCEET), 2012 International Conference On,** Kherva, India, p.877-880, mar. 2012.

JANNACH, D.; ZANKER, M.; FELFERNIG, A.; FRIEDRICH G. **Recommender Systems:** An Introduction. New York: Cambridge University Press, New York, 352 p., 2011.

JIANG, Y.; SHANG, J.; LIU, Y. Maximizing customer satisfaction through an online recommendation system: A novel associative classification model. **Decision Support Systems,** v. 48, n. 3, p. 470-479, 2010.

JONES, W. P.; FURNAS, G. W. Pictures of Relevance: A geometric Analysis of Similarity Measures. **Journal of the American Society for Information Science**, Maryland, v.36, n. 6, p.420-442,1987.

JÚNIOR, Elias Teodoro da Silva. **Middleware adaptativo para sistemas embarcados e de tempo real**. 2008. 127 f. Tese (Doutorado) –

Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008.

KAHANWAL, B.; SINGH, T. The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle. **International Journal Of Latest Research In Science And Technology,** v. 1, n. 2, p.183-187, ago. 2012.

KANAGASABAI, R.; NGAN, L. D.; FENG, Y.; VEERAMANI, A.; EN, J. K. C.; KEONG, C. C.; TSAI, F. S.; ANDRZEJAK, A. EC2BargainHunter: It's Easy to Hunt for Cost Savings on Amazon EC2! **Services (services), 2013 IEEE Ninth World Congress On,** Santa Clara, p.480-487, jul. 2013.

KNIJNENBURG, B.; WILLEMSEN, M.; GANTNER, Z.; SONCU, H.; NEWELL, C. Explaining the user experience of recommender systems. **User Model User-Adap. Inter.** v.22, p. 441-504, 2012.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. **IEEE Computer Society**. v. 42, p.30-37, 2009.

KROTZFLEISCH, H.; MERGEL, I.; MANOUCHEHRI, S.; SCHAARSCHMIDT, M. 2008. **Corporate Web 2.0 Applications**. Hass, B. H., Walsh, G., Kilian, T. (ed.): Web 2.0 - Neue Perspektiven für Marketing und Medien, Berlin Heidelberg, p.73-87, 2008.

KSHEMKALYANI, A. D.; SINGHAL, M. **Distribuited Computing – Principles, Algorithms, and Systems**. Cambridge University Press, 2008.

LAURINDO, S. M.; ANDRÉ, P. B. **Uma Arquitetura de Serviços voltada à Recuperação de Informação e Recomendação de Conteúdo**. 2014. 77 f. TCC (Graduação) - Curso de Tecnologias de Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2014.

LEE, A. J. T.; YANG, F. C; TSAI, H. C; LAI, Y. Discovering content-based behavioral roles in social networks. **Decision Support Systems,** Taipei, Taiwan, n. 59, p.250-261, 2014.

LI, K. Optimal load distribution in non dedicated heterogeneous cluster and grid computing environments. **Journal Of Systems Architecture,** New York, n. 54, p.111-123, 2008.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: item-to-item collaborative filtering. **IEEE Computer Society: Internet Computing, IEEE**, v. 7, n. 1, p. 76-80. fev. 2003.

LYMAN, P.; VARIAN, H. R. **How much information?** Executive summary. 2003.

LÜ, L.; MEDO, M.; YEUNG, C. H.; ZHAN, Y. C.; ZHOU, T. Recommender systems. **Physics Reports.** Hangzhou, p. 1-49. 2012.

LUH, Y.; CHIOU, S,; CHANG, J. Design of distributed control system software using client-server architecture. **Industrial Technology, 1996. (ICIT '96), Proceedings of The IEEE International Conference On**, Shanghai, p.348-350, dez. 1996.

MAHMOOD, T., RICCI, F. **Improving recommender systems with adaptive conversational strategies**. In: CATTUTO, C.; RUFFO, G.; MENCZER, F. (eds.). Hypertext, ACM; 2009, p. 73-82.

MANNING, C. D.; SCHÜTZE, H. **Foundations of statistical natural language**. The MIT Press, Cambridge, Massachusetts, 1999.

MANNO, A.; SHAHRABI, K. **Annual National Conference**: Web 2.0: How It Is Changing How Society Communicates. 2010. Disponível em: <http://www.asee.org/documents/sections/middle-atlantic/fall-2009/01-Web-20-How-It-Is-Changing-HowSociety-Communicates.pdf>. Acesso em: 27 fev. 2014.

MASUDA, K.; MATSUZAKI, T.; TSUJII, J. Semantic Search based on the Online Integration of NLP Techniques. **Pacific Association for Computational Linguistics (PACLING 2011)**, p.281-290, 2011.

MELVILLE, P.; SINDHWANI, V. **Recommender Systems**. Enclyclopedia of Machine Learning, In: SAMMUT, C., WEBB, G. I. (eds.), Springer, p. 829-838, 2010.

MILLER, B. N.; ALBERT, I.; LAM, S. K.; KONSTAN, J. A.; RIEDL, J. MovieLens unplugged: experiences with an occasionally connected recommender system. **Proceedings Of The 8th International Conference On Intelligent User Interfaces.** New York, p. 263-233. jan. 2003.

MIN, S. H.; HAN, I. Detection of the customer time-variant pattern for improving recommender systems. **Expert Systems with Applications**, v.28, n.2, p.189-199, 2005.

MITTAL, G.; KESSWANI, N.; GOSWAMI, K. A Survey of Current Trends in Distributed, Grid and Cloud Computing. **International Journal Of Advanced Studies In Computer Science And Engineering (IJASCSE),** v. 2, n. 3, p.1-6, ago. 2013.

MOLLAH, M. B.; ISLAM, K. R.; ISLAM, S. S. Next Generation of Computing through Cloud Computing Technology. **Electrical & Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference On,** Bangladesh, p. 1-6, mai. 2012.

MONTANER, M.; LÓPEZ, B.; DE LA ROSA, J. L. A Taxonomy of Recommender Agents on the Internet. **Artificial Intelligence Review.** Netherlands: Kluwer Academic Publishers, p. 285-330, 2003.

MONTEIRO, S. D.; FIDENCIO, M. V. As dobras semióticas do ciberespaço: da web visível à invisível. **Transinformação**, v. 25, n. 1, p.35-46, 2013.

MUSSER, J.; O'REILLY, T. **Web 2.0 Report: Principles and Best Practices.** O'Reilly Radar Series. O'Reilly Media, Incorporated, 2007. 101 p.

NOUALI, O.; BLACHE, P. A semantic vector space and features-based approach for automatic information filtering. **Expert Systems with Applications**, v. 26, n. 2, p. 171-179, 2003.

O'REILLY, T. What Is Web 2.0: **Design Patterns and Business Models for the Next Generation of Software**. 2005. Disponível em: < http://oreilly.com/pub/a/web2/archive/what-is-web-20.html>. Acesso em: Janeiro 2014.

PARK, D. H.; KIM, K. H.; CHOI, Y. I.; KIM, K. J. A literature review and classification of recommender systems research. **Expert Systems with Applications**, v.39, n. 11, p. 10059-10072, 2012.

POMMERANZ, A.; BROEKENS, J.; WIGGERS, P.; BRINKMAN, W.-P.; JONKER, C. M. Designing interfaces for explicit preference elicitation: a user-centered investigation of preference representation and elicitation process. **User Model User-Adap. Inter** v.22, p. 357-397, 2012.

RAO, T. K. R. K.; KHAN, S. A.; BEGUM, Z.; DIVAKAR, C. Mining the E-commerce cloud: A survey on emerging relationship between web mining, E-commerce and cloud computing. **Computational Intelligence and Computing Research (ICCIC), 2013 IEEE International Conference On**, Enathi, p. 1-4, 2013.

REINEHR, G. F. **Um Sistema de Recomendação Voltado à Negociação de Bens e Serviços.** 2013. 68 f. TCC (Graduação) - Curso de Tecnologias de Informação e Comunicação, Universidade Federal de Santa Catarina, Araranguá, 2013.

RESNICK, P.; IAKOVOU, N.; SUSHAK, M.; BERGSTROM, P.; RIEDL, J. GroupLens: An open architecture for collaborative filtering of netnews. **Proceeding of the ACM Conference on Computer Supported Cooperative Work**, p. 175-186, 1994.

RESNICK, Paul; VARIAN, Hal. R. Recommender Systems. **Communications of the ACM**, v. 40, n. 3, p. 56–58, 1997.

RICCI, F.; ROKACH, L.; SHAPIRA, B. **Recommender Systems Handbook**. Berlin: Springer Science + business Media, 2011.

RUSSEL, S.; NORVIG, P. **Artificial intelligence: a modern approach.** Prentice-Hall: New Jersey, 1995. 932p.

SADASHIV, N.; KUMAR, S. M. D. Cluster, Grid and Cloud Computing: A Detailed Comparison. **Computer Science & Education (ICCSE), 2011 6th International Conference On,** Singapura, p. 477-482, ago. 2011.

SALTON, G.; BUCKLEY, C. Therm-weighting approaches in automatic text retrieval. **Information Processing & Management,** v. 24, n. 5, p. 512-523, 1988.

SÉRGIO, M. C. **Uma Arquitetura de Descoberta de Conhecimento Baseada na Correlação e Associação Temporal de Padrões Textuais.** 2013. 125 f. TCC (Graduação) - Curso de Tecnologias da Informação e Comunicação, Universidade Federal de Santa Catarina, Florianópolis, 2013.

SHAIKH, F. B.; HAIDER, S. Security threats in cloud computing. **Internet Technology And Secured Transactions (ICITST), 2011 International Conference For,** Abu Dhabi, p.214-219, dez. 2011.

SHIRAZ, M.; GANI, A.; KHOKHAR, R. H.; BUYYA, R. A Review on Distributed Application Processing Frameworks in Smart Mobile Devices for Mobile Cloud Computing. **Communications Surveys & Tutorials, IEEE,** v. 15, n. 3, p.1294-1313, jul. 2013.

TAKÁCS, G.; PILÁSZ, I.; NÉMETH, B.; TIKK, D. Scalable Collaborative Filtering Approaches for Large Recommender Systems. **Journal Of Machine Learning Research,** Budapest, v. 10, p.623-656, 2009.

TANENBAUM, A. S.; STEEN, M. V. **Sistemas distribuídos: princípios e paradigma**. 2. ed. São Paulo: Pearson Prentice Hall, 2007.

TOFFLER, A. **The third wave**. Bantam Books: New York, 1980.

TOMOYA, K.; SHIGEKI, Y. Application of P2P (Peer-to-Peer) Technology to Marketing. **Cyberworlds, 2003. Proceedings. 2003 International Conference On**, p.372-379, dez. 2003.

TRSTENJAK, B.; MIKAC, S.; DONKO, D. KNN with TF-IDF Based Framework for Text Categorization. **Procedia Engineering: 24th DAAAM International Symposium on Intelligent Manufacturing and Automation,** Vienna, v. 69, p.1356-1364, mar. 2014.

VALSAMIDIS, S.; THEODOSIOU, T.; KAZANIDIS, I.; NIKOLAIDIS, M. A Framework for Opinion Mining in Blogs for Agriculture. **Procedia Technology** 8, p.264-274, 2013.

VOIGT, K. I.; ERNST, M. Use of Web 2.0 applications in product development: an empirical study of the potential for knowledge creation and exchange in research and development. **International Journal of engineering**, Science And Technology, v. 2, n. 9, p.54-68, 2010

WU, M. L.; CHANG, C. H.; LIU, R. Z. Integrating content-based filtering with collaborative filtering using co-clustering with augmented matrices. **Expert Systems With Applications.** Jhongli, v. 41, p. 2754-2761, 2014.

WU, X.; ZHU, X.; WU, G. Q.; DING, W. Data Mining with Big Data. **Knowledge and Data Engineering, IEEE Transactions On**. v.26, n. 1, 2014.

YANG, X.; GUO, Y.; LIU, Y.; STECK, H. A survey of collaborative filtering based social recommender systems. **Computer Communications.** p. 1-10. mar. 2014.

YANG, C.; LIU, J.; HUANG, K.; JIANG, F..A method for managing green power of a virtual machine cluster in cloud. **Future Generation Computer Systems,** Taichung, v. 37, p.26-36, mar. 2014.

YU, K. M.; ZHOU, J. Parallel TID-based frequent pattern mining algorithm on a PC Cluster and grid computing system. **Expert Systems With Applications,** v. 37, n. 3, p.2486-2494, mar. 2010.

ZHENGQIAO, X.; DEWEI, Z. Research on Clustering Algorithm for Massive Data Based on Hadoop Platform. **Computer Science & Service System (CSSS), 2012 International Conference On,** Nanjing, p.43-45, ago. 2012.