Ferramenta para extração de dados semiestruturados para carga de um Big Data

João Carlos Furtado[1]
Gabriel Merten Bulsing[1]
Eduardo Kroth[1]
Elpídio Oscar Benitez Nara[1]
Liane Malhmann Kipper[1]

Resumo: Big Data é um termo utilizado para descrever grandes volumes de dados, e vem ganhando destaque à medida que a sociedade se depara com um aumento sem precedentes na quantidade de informações geradas diariamente. As dificuldades em armazenar, analisar e utilizar esse volume de dados tem sido um considerável gargalo para as empresas. O objetivo do Big Data é permitir que as empresas consigam analisar dados de diversas fontes, apresentando os resultados em um menor tempo de requisição possível, auxiliando assim no processo de tomada de decisões. Esse trabalho tem como objetivo o desenvolvimento de uma ferramenta para a extração de dados semiestruturados na Web, para posterior carga em um Big Data.

Palavras-chave: Armazenamento de Dados. Big Data. Gestão da Informação.

Abstract: *Big Data is a term used to describe large volumes of data, and is gaining prominence as the company faces an unprecedented increase in the amount of information generated every day. The difficulties in storing, analyzing and using this data volume has been a considerable bottleneck for businesses. The goal of Big Data is to allow for companies to analyze data from different sources, presenting the results in a shorter time Requisition possible, thus aiding in the decision-making process. This work aims to develop a tool for extracting semi-structured data on the Web, for later loading into a Big Data.*

Keywords: *Data Storage. Big Data. Information Management.*

1 Introdução

Um dos grandes desafios computacionais da atualidade é armazenar, manipular e analisar de forma inteligente a grande quantidade de dados existente. Sistemas corporativos, serviços e sistemas Web, mídias sociais, entre outros, produzem juntos um volume impressionante de dados, alcançando a dimensão de petabytes diários. Com essa evolução da tecnologia da informação (TI) surgiram vários recursos em processamento e armazenamento de dados que visam organizar as bases de informação [1].

Resumidamente, pode-se conceituar o termo Big Data em cinco 'Vs': volume, velocidade e variedade, com os fatores veracidade e valor aparecendo posteriormente nesse conceito.

Na prática, o Big Data oferece a possibilidade de analisar qualquer tipo de informação digital (estruturadas e não estruturadas) em um curto período de tempo, o que é fundamental para que a tomada de decisões seja baseada em fatos, e não apenas em intuições.

[1] Programa de Pós-Graduação em Sistemas e Processos Industriais, UNISC, Av. Independência - 2293 – Santa Cruz do Sul (RS) - Brasil
{jcarlosf@unisc.br, gabrielbulsing@hotmail.com, kroth@unisc.br, elpidio@unisc,br, liane@unisc.br}

http://dx.doi.org/10.5335/rbca.2015.3842

A principal base tecnológica para Big Data são os bancos de dados NoSQL (not only SQL), os quais foram projetados para manipular grandes volumes de dados com performance superior aos tradicionais Sistemas Gerenciadores de Bancos de Dados. Além de lidar com volumes extremamente grandes de dados dos mais variados tipos, soluções de Big Data também precisam trabalhar com distribuição de processamento e elasticidade, ou seja, precisam suportar aplicações com volumes de dados que crescem substancialmente em pouco tempo. É nesse conceito que surge a necessidade da computação paralela e distribuída, que por meio de um conjunto de computadores é possível agregar maior poder de processamento.

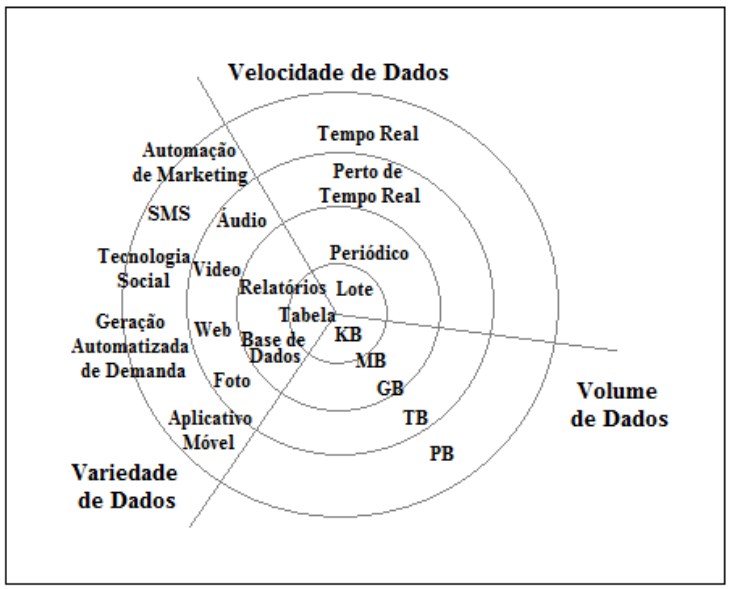
O principal objetivo deste trabalho é desenvolver uma ferramenta que realize a extração de dados de diversas fontes da Web, para armazená-los em um Big Data. Por meio dessa ferramenta será possível analisar informações que antes não estavam disponíveis, auxiliando as empresas no processo de tomada de decisões com maior velocidade. Para esse desenvolvimento foi realizado um estudo exploratório e descritivo que segundo [2] é exploratório por que visa proporcionar maior familiaridade com o assunto realizando levantamento de dados, informando a sua real importância, o estágio teórico em que se encontra, envolvendo levantamento bibliográfico. É descritivo, pois a partir da pesquisa exploratória foi realizado o levantamento das características que fazem parte do problema. Como procedimento de coleta e análise de dados, a metodologia foi estudo de caso, com foco na avaliação do uso da ferramenta.

2 Conceitos de Big Data

Um dos grandes desafios, atualmente, na área da Computação diz respeito à manipulação e ao processamento de grande quantidade de dados no contexto de Big Data [3]. Devido ao crescimento exponencial dos dados nas corporações, Big Data vem ganhando destaque no mercado. As empresas que souberem como minerar essas informações obterão vantagem competitiva em relação a seus concorrentes. Com o Big Data as empresas poderão ter, por exemplo, muito mais flexibilidade e agilidade na mobilização de dados para atender às demandas do negócio. Ou seja, permite a tomada de decisões cada vez mais baseadas em fatos e não apenas em amostragens e intuição.

O autor [4] refere-se ao termo Big Data como: "Banco de Dados de tamanho significativamente maior que os bancos que usualmente conhecemos". Logicamente, é uma definição bastante subjetiva, pois um tamanho considerado grande pode ser tornar pequeno em poucos anos. Outra definição, trazida por [5], é a seguinte: "Big Data trata-se dos dados que excedem a capacidade de processamento dos sistemas de banco de dados convencionais." Para o autor [6], pode-se também conceituar Big Data em três V's, conforme mostra a Figura 1.

Figura 1: Big Data em relação à volume, velocidade e valor dos dados



- Volume: a cada dia são gerados petabytes de dados. E estima-se que este volume dobre a cada 18 meses. O desafio é como gerenciar essa grande quantidade de informações.
- Velocidade: em muitos casos é necessário agir praticamente em tempo real sobre este imenso volume de dados, como em um controle automático de tráfego nas ruas.
- Variedade: os dados são gerados de sistemas estruturados (hoje minoria) e não estruturados (grande maioria) [7]. Gerados por mídias sociais como Facebook e Youtube, documentos eletrônicos, e-mails, mensagens instantâneas, câmeras de vídeo, etiquetas RFID, dentre outros.

Além das três características citadas acima, [4] acrescenta mais duas características presentes no contexto do Big Data:

- Veracidade: é importante ter certeza que os dados fazem sentido e são autênticos.
- Valor: é necessário que as organizações que implementam projetos de Big Data obtenham retorno sobre estes investimentos.

Na prática, a dificuldade do Big Data, além das questões de variedade, velocidade e volume de dados, está no fato de que grande parte desses dados não está estruturada. Ou seja, não são gerados de maneira organizada ou padronizada. Podemos citar alguns exemplos de dados não estruturados, como fotos, vídeos, mídias sociais, sensores, dentre outros [8].

Além dos dados externos, sistemas de informação corporativos captam a todo instante dados internos de vendas, relacionamento com o cliente, faturamento, da concorrência e de movimentos do mercado. Não é raro que esses dados estejam em diferentes sistemas ou aplicações, que não se encontram devidamente integrados, ou seja, estão armazenados em bases paralelas de CRM, BI, Call Center e demais bancos de dados sem comunicação efetiva. Como consequência, novas demandas estão surgindo, como a demanda por análise de grande volume de dados em tempo real (data analytics), o aumento do detalhamento das informações bem como plataformas escaláveis e eficientes de baixo custo [9]. Os sistemas baseados em Big Data são geralmente separados em três fases: aquisição, transformação e análise.

Como visto anteriormente, a aquisição pode ser realizada por meio de diversas fontes de dados, que podem ser armazenados de forma estruturada (banco de dados), ou não estruturada. A transformação costuma ser feita por appliances que utilizam o paradigma MapReduce para extrair a informação, transformando grandes quantidades de dados em informação mais útil para ser analisada na terceira fase. A fase de análise é feita por aplicações de Data Warehouse ou analíticas para aproveitar a informação extraída nas fases anteriores. Nesse momento, a informação está pronta para ser usada por sistemas de tomada de decisões.

3 Tecnologias envolvidas na implantação de um Big Data

3.1 Banco de dados NoSQL

Os bancos de dados NoSQL surgiram no ano de 2009, como uma solução para a questão da escalabilidade no armazenamento e processamento de grandes volumes de dados na Web. NoSQL propõe algumas alternativas ao modelo relacional, porém com uma grande diferença: o modelo NoSQL não tem como objetivo invalidar ou efetivar a total substituição do modelo relacional, e sim o fim do modelo relacional como a única solução correta ou válida. É importante entender que NoSQL não significa “no SQL” (não ao SQL), mas sim “not only SQL” (não só SQL) [10].

Existe uma grande adoção e difusão de tecnologias NoSQL nos mais diversos domínios de aplicação no contexto de Big Data. Esses domínios envolvem, em sua maioria, os quais os SBGD tradicionais ainda são bastante utilizados, como por exemplo, agências governamentais, instituições financeiras, e comércio de produtos de varejo. Isso pode ser explicado pelo fato de que existe uma demanda muito grande para soluções que tenham alta flexibilidade, escalabilidade, performance, e suporte a diferentes modelos de dados.

Em relação aos SGBD tradicionais, a distribuição dos dados de forma elástica é inviabilizada, pois o modelo de garantia de consistência é fortemente baseado no controle transacional ACID. Esse tipo de controle transacional é praticamente inviável quando os dados e o processamento são distribuídos em vários nós. O teorema CAP (Consistency, Availability e Partition tolerance) mostra que somente duas dessas três propriedades podem ser garantidas simultaneamente em um ambiente de processamento distribuído de grande porte [11].

A partir do teorema CAP, os produtos NoSQL utilizam o paradigma BASE (Basically Available, Soft-state, Eventually consistency) para o controle de consistência, o que traz como consequência uma significativa diminuição no custo computacional para a garantia de consistência dos dados em relação a SGBD tradicionais.

Existem atualmente quatro modelos de banco de dados NoSQL: orientado a documentos, orientado a colunas, orientado a grafos, e chave-valor. Por se tratar de uma solução NoSQL com alto desempenho e utilizado por grandes empresas do mercado, optou-se por implementar nesse trabalho o modelo de banco de dados orientado a colunas Apache Cassandra.

3.1.1 Apache Cassandra

O Banco de Dados Apache Cassandra é uma implementação da família de colunas NoSQL que suporta o modelo de dados Big Table e usa aspectos de arquitetura introduzidos por Amazon Dynamo. Inicialmente foi criado pelo Facebook, que abriu seu código-fonte no ano de 2008, e atualmente é mantido por desenvolvedores da fundação Apache. Alguns dos pontos positivos do Cassandra são:

- Alta escalabilidade e disponibilidade, sem um ponto único de falha
- Implementação da família de colunas NoSQL.
- Rendimento de gravação muito alto e bom rendimento de leitura.
- Linguagem de consulta semelhante a SQL (a partir da versão 0.8) e suporte para procura por índices secundários.
- Consistência ajustável e suporte para replicação.
- Esquema flexível.

Esse banco de dados foi desenvolvido para gerenciar grandes quantidades de informações utilizando uma diversificada quantidade de servidores convencionais, mantendo as propriedades supracitadas mesmo em um cenário onde existam nós localizados em diferentes partes do globo e eventuais falhas de comunicação entre os mesmos [12].

Ao contrário do modelo relacional, o modelo de dados do Cassandra não tem tabelas, relacionamentos ou normalizações. É baseado em estruturas mais simples e flexíveis, que podem ser mais facilmente particionadas e replicadas. Essas estruturas consistem em colunas, linhas, famílias de colunas e keyspaces. A seguir serão apresentadas de forma mais detalhada cada um desses modelos.

- Coluna: cada coluna tem um nome (que a identifica), um valor e um timestamp, sendo que tanto o valor quanto o timestamp são fornecidos pela aplicação cliente quando um dado é inserido. Além disso, uma aplicação pode criar uma coluna em tempo de execução, sem precisar declarar ou alterar nada. Além das colunas, existem as super colunas (Super Columns), que são colunas que em vez de terem objetos como valores, possuem outras colunas. Por exemplo, é possível ter uma coluna chamada “Endereço” que tem como valores as colunas “Rua” e “Cidade”, de forma que tanto a coluna “Rua” como a coluna “Cidade” apresentam um valor e um timestamp.
- Linhas: uma coleção de colunas rotuladas com um nome. O Cassandra consiste em muitos nós de armazenamento e armazena cada linha em um único nó. Em cada linha, Cassandra sempre armazena as colunas classificadas por seus nomes. Usando essa ordem de classificação, o Cassandra suporta consultas de fatia, nas quais, dada uma linha, os usuários podem recuperar um subconjunto de suas colunas que estejam em um dado intervalo de nomes de coluna. Por exemplo, uma consulta de fatia com o intervalo tag0 a tag9999 retornará todas as colunas cujos nomes estão entre tag0 e tag9999.
- Família de colunas: conceito similar com as tabelas de um banco de dados relacional. Diferentemente das colunas, as Famílias de Colunas não são dinâmicas e precisam ser declaradas anteriormente em um arquivo de configuração. Analogamente às colunas, também existem as Super Famílias de Colunas

(Super Column Families), que são Famílias de Colunas que possuem como colunas somente Super Colunas. Por exemplo, uma Super Família de Colunas chamada “Agenda Endereços”, pode ter as colunas representando cada entrada na agenda de endereços. Dentro de cada coluna pode-se ter as colunas “rua”, “zip” e “cidade”, ou seja, para cada entrada na agenda, temos a rua do endereço, a cidade e o código postal.

- Keyspace – similar ao nome do banco de dados que onde será alocada as tabelas em um banco relacional. No Cassandra o Keyspace é o recipiente mais externo dos dados, onde é possível criar quantos Keyspace for necessário por cluster.

3.2 Computação Distribuída (Apache Hadoop)

Impulsionada pela grande demanda de computação e por restrições físicas das arquiteturas convencionais, a computação paralela e distribuída acena como alternativa para amenizar alguns dos grandes desafios computacionais [13]. Esse modelo de computação desempenha, atualmente, um papel fundamental no processamento e na extração de informação relevante das aplicações Big Data. Essa computação é normalmente realizada em aglomerados (clusters) e grades computacionais, que com um conjunto de computadores comuns, conseguem agregar alto poder de processamento a um custo associado relativamente baixo.

O Hadoop é um projeto de software livre desenvolvido pela Apache Software Foundation, o qual é uma plataforma de computação distribuída, com alta escalabilidade, grande confiabilidade e tolerância a falhas.

A ideia de promover soluções para os desafios dos sistemas distribuídos em um único arcabouço é o ponto central do projeto Hadoop. Nesse arcabouço, problemas como integridade dos dados, disponibilidade dos nós, escalabilidade da aplicação e recuperação de falhas ocorrem de forma transparente ao usuário. Além disso, seu modelo de programação e sistema de armazenamento dos dados garante um rápido processamento, muito superior às outras tecnologias similares.

Embora uma aplicação Hadoop normalmente seja executada em um conjunto de máquinas, ela pode também ser executada em um único nó. Essa possibilidade permite simplificar as configurações para as fases iniciais de implementação e testes, visto que configurar aplicações distribuídas é uma tarefa complexa.

Posteriormente, outras configurações mais sofisticadas podem ser utilizadas para usufruir de todas as vantagens oferecidas pelo Hadoop. Existem três modos possíveis de execução: modo local, modo pseudo-distribuído e modo completamente distribuído.

3.3 Modelo de programação MapReduce

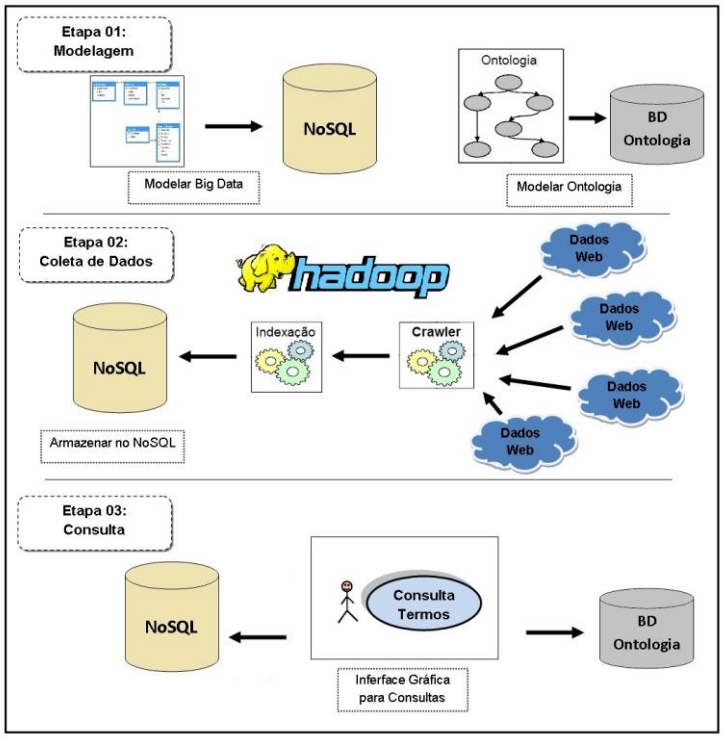
O MapReduce é um conjunto de bibliotecas que permite realizar processamento em paralelo, de grandes quantidades de dados, usando todo o hardware disponível no cluster Hadoop, dividindo esse processamento em duas etapas, uma chamada Map, que é o mapeamento e a validação dos dados e a outra chamada Reduce, que tem como entrada o resultado da fase Map anterior, gerando o resultado final [14].

Para uma melhor compreensão do funcionamento desse modelo de programação, assume que possuímos cinco arquivos de entrada e cada arquivo possua duas colunas, uma que representa a chave e outra o valor. Cada par desses representa uma cidade e a temperatura medida nessa cidade em um dia qualquer. Deseja-se encontrar a temperatura máxima registrada de cada cidade entre todos os dados que temos. Cada arquivo pode apresentar múltiplas leituras de uma mesma cidade, e diferentes arquivos podem conter medidas da mesma cidade. O Framework do MapReduce irá dividir essa tarefa em cinco operações de Map, cada uma atuando em um arquivo diferente. Cada Map realizará a leitura de cada arquivo, retornando a temperatura máxima para cada cidade. Os resultados dos Mappers são então alimentados ao Reducer. O Reducer combina os resultados da operação de Map e retorna como saída um valor único para cada cidade, que representa o valor máximo de temperatura para a localidade [15].

4 Solução desenvolvida – Big Crawler

Este trabalho tem o intuito de desenvolver uma ferramenta para a extração de dados na Web, para carregá-los em um Big Data. A Figura 2 mostra a estrutura adotada para o desenvolvimento dessa ferramenta.

Figura 2: Visão geral do ambiente de trabalho



A primeira etapa (modelagem) consiste em desenvolver um ambiente para identificação e modelagem de documentos semiestruturados e não estruturados provenientes da Web. Para armazenas as informações extraídas da Web, foi modelado um banco de dados NoSQL Apache Cassandra. Nessa etapa, também ocorre a modelagem de uma ontologia, para representar o conhecimento de acordo com necessidades apontadas por uma situação real. Essa etapa deve ser realizada primeiro, antecedendo a etapa de coleta de dados, pois compreende a prévia identificação e expressão de uma área do conhecimento, em que será indexado o conteúdo buscado pela web crawler. A construção da ontologia utilizada no projeto foi realizada manualmente, e armazenada em um banco de dados relacional. As URLs dos sites iniciais, as quais são utilizadas para o Crawler iniciar a busca, também são armazenadas em um banco de dados MySQL e devem ser definidas antes de realizar a execução do motor de busca.

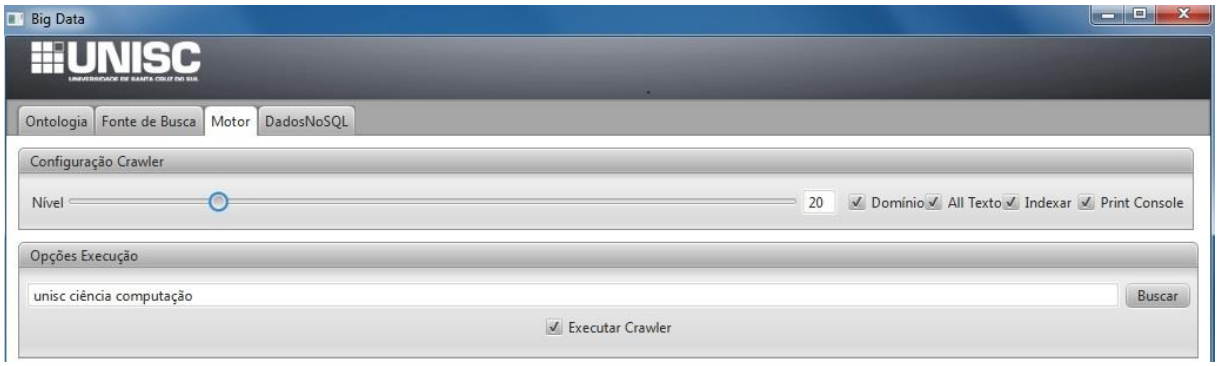
Na segunda etapa, é realizada a coleta de dados através de um motor de busca, realizada a indexação e o armazenamento em um banco de dados. Para a coleta de dados da web, o robô de captura (web crawler), irá recuperar as informações alimentando a base de dados do Apache Cassandra. Nessa etapa, também ocorre a indexação do conteúdo coletado pela web crawler. Todo o processo de indexação será executado e controlado pelo Apache Lucene.

Ainda durante a segunda etapa, entra em funcionamento o Framework Hadoop. Esse framework para computação distribuída utiliza o modelo de programação MapReduce, o qual foi implementado nesse projeto. Resumidamente, o funcionamento do Hadoop na ferramenta Big Crawler funciona da seguinte maneira: existem cinco nós de processamento executando a busca de forma paralela. Cada um desses nós recebe uma das URLs cadastradas na primeira etapa (processo chamado de Map), realiza a busca conforme o número de níveis, e envia para o Reduce armazenar no banco de dados Cassandra. Dessa forma é possível processar cinco buscas por URLs paralelamente, otimizando o tempo para a execução do Web Crawler.

Na terceira etapa (consulta), foi desenvolvida uma interface gráfica amigável para visualização dos arquivos extraídos da Web, e armazenados no banco de dados Cassandra conforme a ontologia especificada na primeira etapa. Também é possível visualizar uma busca em forma de ranking, semelhante a um site de busca.

Após realizar as etapas iniciais de cadastro da ontologia e dos sites sementes, pode-se executar o motor de busca. A Figura 3 apresenta a configuração do motor de busca na aplicação desenvolvida:

Figura 3: Configurações do motor de busca



Antes de executar o Crawler, é importante entender a funcionalidade de cada recurso exibido na configuração do motor de busca:

- Nível: por meio dessa opção é definida a quantidade de níveis em que o Crawler irá buscar a partir dos sites semente. Ressaltando que quanto mais níveis de busca forem configurados, maior será o tempo para retorno das informações extraídas.
- Domínio: com essa opção ativada, o Crawler visita e coleta somente sub-links das páginas que pertençam ao mesmo domínio da página semente, excluindo links externos ao domínio (como por exemplo, as propagandas).
- All Texto: ativando essa opção, é realizada a coleta de todo o texto da página independente de qual tag HTML ele estiver incluso. Ou seja, é coletado todo o texto entre as tags <BODY> e </BODY>. Já com essa opção desativada, a coleta é realizada apenas do parágrafo (entre as tags <P> e </P>).
- Indexar: possibilita que a ferramenta já realize a indexação logo após realizar a extração dos dados. Essa opção deve ser marcada quando se deseja executar uma consulta em conjunto com a execução do motor de busca.
- Print Console: habilitando essa opção, permite visualizar no console do Eclipse (IDE utilizada para o desenvolvimento da aplicação) as URLs que estão sendo percorridas pelo motor de busca. Após finalizar o processo exibe os resultados da busca, como o número de páginas encontradas e a duração da busca.

Para a realização da indexação do conteúdo extraído foi utilizada a ferramenta Apache Lucene, pois se trata de uma biblioteca confiável e já consolidada no mercado.

O Lucene provê basicamente um motor de busca, com uma interface que dá acesso às funcionalidades de indexação das informações provenientes de diversas fontes, consulta e apresentação de resultados. Cabe ao desenvolvedor que a utiliza, definir quais serão as informações submetidas ao motor de busca e também qual serão suas fontes.

4.1 Estudo de caso

Com o objetivo de avaliar o funcionamento da ferramenta desenvolvida nesse trabalho foi realizado um estudo de caso, executando o Big Crawler em um ambiente de trabalho real. Esse estudo é fundamental para avaliar os resultados da ferramenta, como o volume de dados extraídos e a velocidade para extração e indexação das informações. Para isso, será modelada uma ontologia com o auxílio de um especialista do segmento, e também a definição das URLs iniciais onde o Web Crawler iniciará o seu processo de busca.

Antes de iniciar as validações, é necessário preparar a estrutura para a instalação do Big Crawler. A ferramenta desenvolvida foi instalada em um servidor com processador Intel Core i7, com memória RAM de 16Gb.

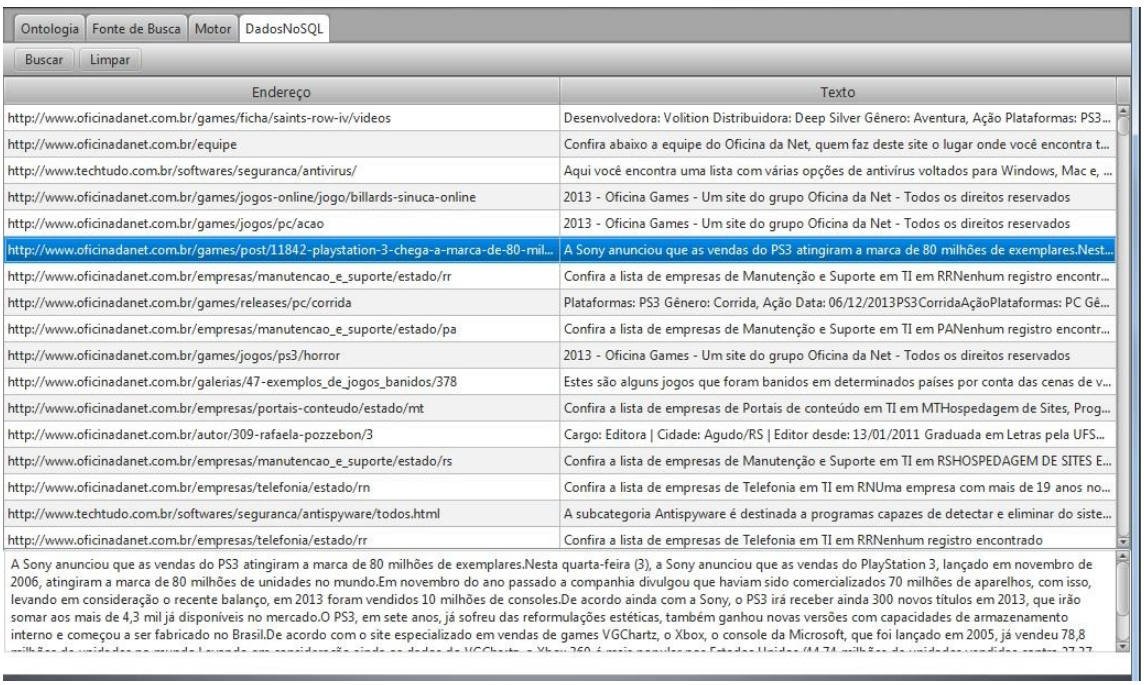
Como o Hadoop será executado no modo “pseudo-distribuído” para o processamento de dados, esse servidor irá simular a execução de cinco clusters. Ou seja, teremos cinco Maps processando os dados em paralelo, exigindo assim um maior processamento da máquina. Porém, o Hadoop permite trabalhar no modo distribuído com milhares de clusters por meio do uso de máquinas menos robustas.

O desenvolvimento da ontologia foi realizado em conjunto com um profissional da empresa, o qual preparou uma listagem de termos e a ligação entre eles para um estudo de tendências do mercado de informática. Após realizar o cadastro da ontologia foram definidos os sites sementes, os quais são o ponto de partida para a extração dos dados pelo Crawler. Nessa etapa, o profissional da empresa cadastrou vinte URLs relacionadas ao mercado de informática. Após a configuração do Big Crawler, foi realizada uma busca nos sites sementes em até dezenove subníveis conforme a ontologia especificada. Considerando o tamanho reduzido da ontologia cadastrada, e a reduzida quantidade de sites sementes, avalia-se que essa quantidade de níveis já retornará resultados satisfatórios para o estudo de caso. Os termos utilizados na busca foram “mercado software”, e os resultados são exibidos logo abaixo:

- Quantidade de páginas encontradas: 12.370.
- Tempo para execução da busca: 00:17:49.

A seguir, são apresentados os resultados da busca de armazenamento na Cassandra.

Figura 4: Visualização dos dados armazenados no Cassandra



É possível também verificar que os resultados foram exibidos em uma velocidade maior em relação ao comparativo das ferramentas, trazendo informações relevantes de maneira mais ágil para uma futura análise das informações.

Portanto, foi constatado que o uso da ontologia aumenta o volume de resultados retornados da base de dados, onde também a qualidade dos resultados retornados apresentou um alto índice de aprovação. Com a aplicação da ontologia na consulta, tende-se a apenas retornar resultados que tenham ligação com o assunto desejado, otimizando o conteúdo da informação.

Outro fato apontado ao final do estudo foi o de que o trabalho seria melhor avaliado se houvesse uma melhor modelagem da ontologia, com mais termos relacionados ao assunto, e também com um maior número de sites sementes. Sendo assim, testes mais específicos poderiam ser realizados, armazenando um volume ainda maior de informações no banco de dados.

5 Conclusão

Com base nas necessidades de armazenar informações de fontes não estruturadas, conforme mencionado, este estudo propôs a criação de uma ferramenta para extração de dados na Web, para posterior carga em um ambiente de Big Data.

Com esse objetivo, foi desenvolvido um robô de busca que realiza a extração das páginas da web por meio da modelagem de uma ontologia realizada pelo usuário. Essas informações são indexadas e armazenadas em um Big Data, para que futuramente possam ser facilmente recuperadas, auxiliando, assim, no processo de tomada de decisões.

Para modelar um ambiente de Big Data foram necessários estudos aprofundados nos bancos de dados NoSQL e nas tecnologias de computação distribuída/paralela. Dentre as ferramentas disponíveis no mercado, optou-se por modelar um banco de dados orientado a colunas Apache Cassandra, e para a realização da computação distribuída foi utilizado o Apache Hadoop, o qual funciona sob o modelo de programação MapReduce.

Com base em uma ontologia, é possível simular o conhecimento de acordo com as atividades da empresa e suas áreas de interesse, otimizando assim a busca pelas informações mais relevantes ao usuário. Sendo assim, esse conhecimento pode ser persistido para que futuramente possa ser aprimorado e também reutilizado. A ontologia foi aplicada na etapa de recuperação da informação, por meio do uso da biblioteca Apache Lucene para realizar a consulta na base de dados indexada. Para realizar a manipulação da ontologia, como a inclusão, a exclusão e a ligação entre os termos, desenvolveu-se uma interface gráfica para facilitar o usuário nessas tarefas.

Mesmo fugindo do escopo do trabalho, foram desenvolvidas duas formas de visualização das informações extraídas da internet, facilitando assim a análise dos resultados. As informações podem ser visualizadas em forma de ranking, de acordo com a relevância da busca, e também conforme o armazenamento no banco de dados Cassandra.

Portanto, conclui-se que os objetivos do trabalho foram alcançados ao apresentar resultados satisfatórios na extração de dados na web, principalmente quanto à velocidade na busca e na relevância das informações armazenadas.

Agradecimentos

Agradecemos o apoio da Universidade de Santa Cruz do Sul – Unisc e ao Programa de Pós-Graduação em Sistemas e Processos Industriais – Brasil.

Referências

[1] BRITO, C. T. et al. Single Sign-On: um estudo de caso em banco de dados Oracle. *Revista Brasileira de Computação Aplicada*, Passo Fundo, v. 4, n. 2, p. 28-41, 2012.

[2] SANTOS, R. A. *Metodologia Científica*: a construção do conhecimento. 4. ed. Rio de Janeiro: DP&A, 2001.

[3] HASHEM, I. A. et al. The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, v. 47, p. 98-115, 2015.

[4] TAURION, C. *Big Data*: a nova fronteira da inovação. 2011. Disponível em: <http://imasters.com.br/artigo/22095/tendencias/big-data-a-nova-fronteira-da-inovacao/>. Acesso em: 15 mar. 2013.

[5] DUMBILL, E. *What is big data?* 2012. Disponível em: <http://strata.oreilly.com/2012/01/what-is-big-data.html>. Acesso em: 13 mar. 2013.

[6] WILLIAMS, N.; FERDINAND, N. P.; CROFT, R. Project Management Maturity in the age of Big Data. *International Journal of Managing Projects in Business*, v. 7, n. 2, 2014.

[7] CHEN, H.; CHIANG, R. H.; STOREY, V. C. Business Intelligence and Analytics: from Big Data to Big Impact. *MIS Quarterly*, v. 36, n. 4, p. 1165-1188, 2012.

[8] MATHEW, P. A. et al. Big-data for building energy performance: Lessons from assembling a very large national database of building energy use. *Applied Energy*, v. 140, p. 85-93, 2015.

[9] KI, Seo Jin; KIM, Hyeon-Ju; KIM, Albert S. Big data analysis of hollow fiber direct contact membrane distillation (HFDCMD) for simulation-based empirical analysis. *Desalination*, v. 355, p. 56-67, 2015.

[10] PORCELLI, A. O que é NoSQL? *Revista Java Magazine*, n. 86, 2011. Disponível em: <http://www.devmedia.com.br/o-que-e-nosql-java-magazine-86/18777>. Acesso em: 28 maio 2013.

[11] VIEIRA, M. R. *Bancos de Dados NoSQL*: Conceitos, Ferramentas, Linguagens e Estudos de Casos no Contexto de Big Data. 2012. Disponível em: <http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf>. Acesso em: 21 mar. 2013.

[12] MARROQUIM, M.; RAMOS, R. *Distribuição de dados em escala global com Cassandra*. 2012. Disponível em: <http://mariomarroquim.github.io/research/artigo-mariomarroquim-cassandra.pdf>. Acesso em: 30 ago. 2013.

[13] WANG, R. et al. Learning ELM-Tree from big data based on uncertainty reduction. *Fuzzy Sets and Systems*, v. 258, p. 79-100, 2015.

[14] FONTE, F. *O que é Hadoop?* 2013. Disponível em: <http://www.bigdatabrazil.blogspot.com.br/2013/06/o-que-e-o-hadoop.html>. Acesso em: 15 out. 2013.

[15] ÉVORA, L. *MapReduce*: Solução para o Big Data? 2013. Disponível em: <http://www.twistsystems.com/blog/2013/01/27/mapreduce-solucao-para-o-big-data/#.Uo5aNdLBNkV>. Acesso em: 21 out. 2013.