



Patterns Identification in the Consumption of Oracle Database Resources Using Weka Tool in Data Mining

Júlio César Brasil Pereira¹, Luciano Carvalho da Rocha Filho², Meirielle Silva e Silva³,
Rilmar Pereira Gomes⁴

^{1,2,3} Centro Universitário Fametro - Av. Constantino Nery, 3000 - Manaus/AM.

⁴ M.Sc. Departamento de Pesquisa, Universidade Federal do Pará (UFPA).

Email: jcb.pereira19@gmail.com, lucianorfilho@gmail.com, meirielle19@gmail.com, rilmargomes@hotmail.com

ABSTRACT

This article focuses on the identification of periods of consumption of database resources helping the management of I.T. in decision-making. The purpose of this work is to mine data dictionary information from a given database, the Oracle Database System Manager, to obtain information to assist in analyzing the features of this database. For this, the KDD information extraction process was used in order to extract knowledge from the database by the identification of patterns. As a tool, we used Weka to develop this study, because it is open-source and has a range of algorithms that contains mining techniques, where it is possible to apply the last two steps of KDD. The applied methodology was a case study, making use of bibliographical techniques to collect information pertinent to the topic, in which all the 5 attributes related to the computational resources utilization of an Oracle DBMS were used. This data set consists of 5 attributes and 120 instances. Analyzing the statements generated by the rules it is possible to know which resource consumption is more or less used during a period of the day, proving to be useful in determining the resource consumption characteristics and thus assist in the decision to define a specific time to execute processes without affecting the overall performance of the database.

Keywords: DBMS, data dictionary, Weka, KDD, data mining.

Identificação de padrões em consumo de recursos de banco de dados Oracle utilizando a ferramenta Weka na mineração de dados

RESUMO

Este artigo apresenta como foco a identificação de períodos de consumo de recursos de banco de dados, auxiliando a gestão de T.I nas tomadas de decisões. É apresentado como objetivo minerar informações do dicionário de dados do Sistema Gerenciador de Banco de Dados Oracle de uma determinada base de dados, para que seja obtido informações que auxiliem na análise de recursos do banco de dados. Para isto, foi utilizado o processo de extração de informações KDD a fim de extrair conhecimento da base de dados a partir da identificação de padrões. Como ferramenta foi utilizado o Weka, para desenvolver este estudo por ser open-source e possuir uma gama de algoritmos que contém técnicas de mineração, onde é possível aplicar as duas últimas etapas do KDD. A metodologia aplicada foi estudo de caso fazendo uso de técnicas bibliográficas para coletar informações pertinentes ao tema em estudo, no qual foram utilizados todos os 5 atributos relacionados a utilização de recursos computacionais de um SGBD Oracle. Este conjunto de dados é composto 5 atributos e 120 instâncias. Ao analisar as afirmações geradas pelas regras é possível saber qual o consumo de recurso é mais e o menos utilizado durante um período do dia, mostrando-se útil em determinar características de consumo de recurso e assim auxiliar na decisão de definir um horário específico para executar processos sem afetar a performance global do banco de dados.

Keywords: SGBD, dicionário de dados, Weka, KDD, mineração de dados.

Received: December 30th, 2018

Accepted: January 15th, 2019

Published: March 31th, 2019

Copyright ©2016 by authors and Institute of Technology Galileo of Amazon (ITEGAM).

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



I. INTRODUÇÃO

Com o aumento exponencial das informações e competitividade entre organizações, é fundamental que haja uma TI bem gerenciada para que esta seja capaz de através de determinadas tecnologias evitar que uma empresa chegue a falência, com a integração de softwares, Data Mining, Data Warehouse e outros que auxiliem em tomadas de decisões [1]. As decisões a serem tomadas não se limitam apenas as extraídas das informações armazenadas no banco de dados, há também a necessidade em saber sobre o funcionamento e capacidade do seu SGBD (Sistema Gerenciador de Banco de Dados).

Através do dicionário de dados, podemos coletar várias informações do SGBD, como estatísticas e métricas, inclusive informações de auditoria [2]. Ao aplicar mineração de dados em cima dos dados obtidos do dicionário de dados, é possível gerar conhecimento baseado nos valores de tempo de consumo de recurso de disco, e se os discos estão sendo capazes de lidar com a carga de processamento atual ou não.

A mineração de dados lida com grandes bancos de dados, sendo capaz de encontrar padrões que venham a ser úteis para a descoberta de conhecimento [3]. Desta forma, é capaz de ser implantada nas mais diversas áreas, encontrando padrões que muitas vezes não são notadas, mas que são primordiais para os negócios.

O conhecimento sobre o consumo de recursos do SGBD e os períodos em que acontecem, assim como a identificação dos fatores que podem influenciar no seu desempenho, são primordiais tanto para uma equipe técnica, como para a gestão de TI nas suas tomadas de decisões, na identificação de recursos que necessitam de investimento em seu parque tecnológico, ou recursos que estão sendo subutilizados, de modo que este seja capaz de gerar as informações de forma simples e claras.

Diante do exposto, este artigo tem como objetivo minerar informações do dicionário de dados de uma determinada base dados Sistema Gerenciador de Banco de Dados Oracle, para que seja obtido informações que auxiliem na análise de recursos do banco de dados. A partir desses dados colhidos, partimos para o KDD (Knowledge Discovery in Databases) e por fim, utilizamos o Weka (Waikato Environment for Knowledge Analysis), para conhecer qual período do dia teve mais processos e consumo de recursos.

Para a realização desse estudo, a metodologia aplicada seguiu o método do estudo de caso e pesquisa bibliográfica em periódicos e livros, para conhecer o atual cenário da pesquisa desse tema.

II. MATERIAIS E MÉTODOS

II.1 METODOLOGIA

Neste trabalho é empregado uma pesquisa bibliográfica e estudo de caso aplicado, que tem a finalidade de extrair dados estatísticos de consumo de recursos de um SGBD Oracle. A partir da coleta de informações de seu dicionário de dados e após prepará-los para mineração o intuito é identificar a frequência de padrão de consumo de recursos utilizados em determinados períodos do dia, onde é utilizada a ferramenta Weka, aplicando técnicas de associação através de um algoritmo chamado APRIORI para a geração de regras, afim de abstrair conhecimento através do processo chamado KDD (Knowledge Discovery in Database).

II.2 KDD-KNOWLEDGE DISCOVERY IN DATABASES

CO KDD é um processo utilizado em mineração de dados, cujo objetivo principal é extrair conhecimento da base dados, sabendo quais resultados deseja-se atingir. Esta descoberta de conhecimento envolve uma sequência de etapas que devem ser obedecidas e repetida quantas vezes for necessário na busca de melhores resultados.

Este processo consiste em 5 grandes etapas necessárias para a descoberta de conhecimento, são elas: seleção, pré-processamento, transformação dos dados, mineração dos dados, avaliação dos modelos encontrados/apresentação do conhecimento [4].

Na Figura 1 são representadas as etapas do processo KDD, e manipulação dos dados ao longo desse processo.

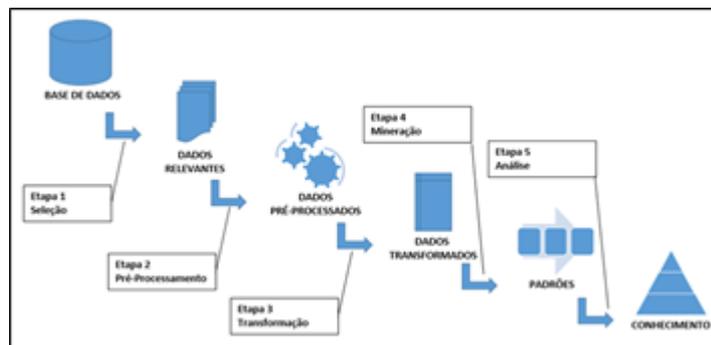


Figura 1: Processo KDD.

Fonte: Autores, (2018).

A primeira etapa do processo do KDD é a seleção conforme a Figura 1, tem por objetivo definir as fontes de dados que contém as informações necessárias para o processo de mineração, reunidos em uma única base.

A base de dados utilizada neste estudo de caso, são as informações de estatísticas do dicionário de dados de um sistema gerenciador de banco de dados Oracle 11gR2, e consiste em informações sobre a carga de servidor, número de transações, consumo de leitura/escrita em disco e tempo de utilização do banco de dados.

Para armazenamento destes dados foi criado uma tabela que foi alimentada a partir da execução de uma procedure, que precisou ser criada para este fim, onde coletou informações de estatísticas de consumo de recurso do banco de dados a partir das views internas chamadas v\$sysstat e v\$osstat, que fazem parte do dicionário de dados [5].

Foi realizado a coleta dessas informações durante 5 dias em um intervalo de 1 hora, onde gerou uma tabela, conforme figura 2, com os seguintes campos: id, data, ioread_time, iowrite_time, load e transaction.

ID	DATA	IOREAD_TIME	IOWRITE_TIME	COMMIT	ROLLBACK
31147	13/08/2018 01:00	1325038	916579	20519	6026
31148	13/08/2018 02:00	3915	186930	20193	3136
31149	13/08/2018 03:00	6235	194009	22524	4216
31150	13/08/2018 04:00	15177	191855	22262	3466
31151	13/08/2018 05:00	9592	178112	20698	5574
31152	13/08/2018 06:00	39549	109085	9043	19919
31153	13/08/2018 07:00	677225	307936	10854	99664
31154	13/08/2018 08:00	296015	259312	37603	159046
31155	13/08/2018 09:00	312894	191930	29256	170014
31156	13/08/2018 10:00	269389	267040	29073	149802
31157	13/08/2018 11:00	852574	431566	40086	111323
31158	13/08/2018 12:00	204282	333149	34142	98376
31159	13/08/2018 13:00	286128	188200	22322	135489
31160	13/08/2018 14:00	388823	189226	28159	132434
31161	13/08/2018 15:00	720761	331067	23662	120152
31162	13/08/2018 16:00	102947	204272	18687	89455
31163	13/08/2018 17:00	54750	240417	29053	61249
31164	13/08/2018 18:00	63207	224413	28614	41390

Figura 2: Tabela temporária de dados brutos.

Fonte: Autores, (2018).

Em seguida foi iniciada a etapa de seleção, onde foi possível identificar problemas e formular hipóteses após a compreensão dos dados.

Seguindo a segunda etapa do KDD conforme exibida anteriormente na Figura 1 chamada de pré-processamento, esta etapa consiste na verificação da consistência das informações, correção de possíveis erros, escolha de estratégias para manipular campos de dados com grande variabilidade e a formatação de dados, de maneira a adequá-los ao algoritmo de mineração APRIORI que será utilizado neste processo.

De acordo com a Figura 1, a terceira etapa é chamada de transformação, onde consiste em formatar os dados de um estado bruto a partir da análise feita na etapa anterior em um estado que os dados se encontrem formatados para aplicar o algoritmo de mineração.

Na etapa de transformação aplica-se operações que melhoram a visualização dos dados [6]. Transformando faixa de valores em grupos, a fim de facilitar a sumarização dos dados. Para isso alguns campos necessitam ter os seus tipos de dados alterado para facilitar o uso do algoritmo APRIORI.

Para realizar os ajustes desses dados, foram criadas funções no banco de dados para tratar essas informações, de definição de faixas de valores em grupos.

Para o campo "Data", que descreve o dia e horário em que a coleta foi realizada no banco de dados, foi criada uma função que trata a variedade de dados, adaptando estes para obter os valores conforme é exibida na Tabela 1.

Tabela 1: Conteúdo transformado do campo Data.

Data - DateTime	Data – AlfaNumérico
DD/MM/YY 00:00 – 05:59	Madrugada
DD/MM/YY 06:00 – 11:59	Manha
DD/MM/YY 12:00 – 17:59	Tarde
DD/MM/YY 18:00 – 23:59	Noite

Fonte: Autores, (2018).

Para o campo "IOWrite_Time", que descreve o tempo de execução em que o SGBD esteve em espera por escrita em disco, foi criada uma função que trata a faixa de valores de tempo, adaptado conforme é exibida na Tabela 2.

Tabela 2: Conteúdo transformado do campo IOWrite_Time.

IOWrite_Time – Numérico (Seg)	IOWrite_Time – AlfaNumérico
Menos de 100	Muito Baixo
$\geq 100 - < 400$	Baixo
$\geq 400 - < 700$	Médio
$\geq 700 - < 1000$	Alto
Acima de 1000	Muito Alto

Fonte: Autores, (2018).

Para o campo "IORead_Time", que descreve o tempo de execução em que o SGBD esteve em espera por leitura em disco, foi criada uma função que trata a faixa de valores de tempo, adaptado conforme é exibido na Tabela 3.

Tabela 3: Conteúdo transformado do campo IORead_Time.

IORead_Time – Numérico (Seg)	IORead_Time – AlfaNumérico
Menos de 100	Muito Baixo
$\geq 100 - < 400$	Baixo
$\geq 400 - < 700$	Médio
$\geq 700 - < 1000$	Alto
Acima de 1000	Muito Alto

Fonte: Autores, (2018).

Para o campo "Load", que descreve um valor de carga de processamento do servidor que o SGBD está instalado. Foi criada uma função que trata a faixa de valores, adaptado conforme Tabela 4.

Tabela 4: Conteúdo transformado do campo Load.

Load – Numérico	Load – AlfaNumérico
Menos de 1	Baixo
$\geq 1 - < 3$	Médio
Acima de 3	Alto

Fonte: Autores, (2018).

Para os campos "commit" e "rollback", que descrevem um valor quantitativo de transações que o SGBD está executando, foi criada uma função que trata a faixa de valores de quantidade, adaptado conforme é exibido na Tabela 5.

Tabela 5: Conteúdo transformado dos campos Commit e Rollback.

Transações – Numérico (Quant.)	Transações – AlfaNumérico
Menos de 50.000	Muito Baixo
50.000 – 100.000	Baixo
100.00 – 150.000	Médio
150.00 – 200.000	Alto
Acima de 200.000	Muito Alto

Fonte: Autores, (2018).

Após a criação das funções para ajustes dos dados, temos os dados todos formatados, onde é observado na Figura 3, e pronto para exportação em arquivo no formato .csv, extensão necessária para importação dos dados na ferramenta Weka.

HORARIO	LOAD	IOREAD	IOWRITE	TRANSACOES
madrugada	alto	baixo	medio	muito baixo
madrugada	medio	muito baixo	muito baixo	muito baixo
madrugada	medio	muito baixo	muito baixo	muito baixo
madrugada	medio	muito baixo	muito baixo	muito baixo
madrugada	medio	muito baixo	muito baixo	muito baixo
madrugada	alto	muito baixo	muito baixo	muito baixo
manha	alto	baixo	baixo	medio
manha	alto	baixo	baixo	alto
manha	alto	baixo	muito baixo	alto
manha	medio	baixo	baixo	alto
manha	alto	baixo	baixo	alto
manha	medio	baixo	baixo	medio
tarde	medio	baixo	muito baixo	alto
tarde	medio	baixo	muito baixo	alto
tarde	alto	baixo	baixo	medio
tarde	alto	muito baixo	baixo	medio
tarde	medio	muito baixo	baixo	baixo
tarde	medio	muito baixo	baixo	baixo
noite	medio	muito baixo	baixo	baixo

Figura 3: Tabela temporária de dados consolidados.

Fonte: Autores, (2018).

II.3 FERRAMENTA WEKA (WAIKATO ENVIRONMENT FOR KNOWLEDGE ANALYSIS)

Weka (Waikato Environment for Knowledge Analysis) é uma ferramenta open-source criada pela Universidade de Waikato, situada na Nova Zelândia, e no meio acadêmico vem sendo muito utilizada na mineração de dados. A ferramenta foi escolhida para desenvolver este estudo por ser open-source e

possuir uma gama de algoritmos que contém técnicas de mineração, onde é possível aplicar as duas últimas etapas do KDD [7], conforme explanada na Figura 1.

As técnicas de mineração consistem em métodos de classificação, regressão, agrupamento e regras de associação, e para cada um destes existem algoritmos específicos. A técnica empregada neste estudo foi a de associação, utilizando o algoritmo APRIORI, onde este algoritmo é um dos mais conhecido e utilizado nas regras de associação.

A associação consiste na identificação de padrões dentro de um conjunto de dados, ou seja, encontrar conjuntos de itens que ocorram simultaneamente e de forma frequente em um banco de dados. A regra de associação tem o formato $A \rightarrow B$, onde A e B são itemsets, sendo A chamado de antecedente, e B, chamado de conseqüente, e o conjunto de A mais o B, são chamados de conjunto de itens ou transações, e a regra pode ser lida como: o atributo A frequentemente implica no atributo B.

Aplicando este conceito no estudo e obtido os seguintes atributos: “Horario A, Load B, IORead C, IOWrite D \rightarrow Transações E”, pode ser lido da seguinte forma: existe ocorrências que no Horário A, com Load B, IORead C e IOWrite D, implica em uma quantidade de transações E.

O APRIORI realiza técnicas exploratórias, fazendo buscas recursivas na base de dados à procura dos conjuntos frequentes, onde irá gerar regras que descrevem os padrões mais relevantes presentes nos dados, na sua execução utiliza conceitos de suporte e confiança [8].

O suporte de um itemset pode ser entendido como a porcentagem de transações em que um itemset aparece no banco de dados. Sua fórmula é $\text{suporte}(X) = \text{transações}(X) / \text{transações}(BD)$, onde $\text{suporte}(X)$ é o valor de suporte do itemset X, $\text{transações}(X)$ é o número de transações em que o itemset X aparece e $\text{transações}(BD)$ é o número de transações total do banco de dados.

A confiança é ideal para identificar quanto uma regra é significativa, onde sua fórmula é: $\text{confiança}(A \rightarrow B) = \text{suporte}(AB) / \text{suporte}(B)$, onde $\text{confiança}(A \rightarrow B)$ é a confiança da regra $A \rightarrow B$, $\text{suporte}(AB)$ é o suporte do itemset AB e $\text{suporte}(B)$ é o suporte do itemset B.

A execução do algoritmo APRIORI consiste de três fases: geração, poda e varredura. A geração consiste em gerar itemsets maiores a partir de itemsets menores frequentes. Já a poda elimina itemsets gerados que não possuem probabilidade de serem frequentes. A varredura percorre todas as transações do banco de dados, verificando se os itemsets que passaram da poda são frequentes.

Na obtenção dos itemsets de tamanho 1, o algoritmo realiza apenas uma varredura no banco de dados. No caso dos itemsets de tamanho 2, o algoritmo executa apenas a geração e varredura. Para os itemsets de tamanho 3 ou maiores acontecem todas as três etapas.

Aplicando na base de dados do estudo de caso, mostrado na Figura 4(a) onde cada letra representa um valor de consumo de recurso na utilização de um banco de dados, onde o primeiro parâmetro se refere ao horário, com valor A ou B, o segundo ao load com valor C ou D, o terceiro ao ioread com valor E ou F, será usado apenas 3 dos 5 atributos para facilitar a exemplificação. Considere também um suporte mínimo de 50% e confiança de 80%.

transação	recurso		
1	A	C	E
2	B	C	E
3	B	D	F
4	B	C	F
5	B	C	E

(a) base de dados

itemset
BCE

regra	confiança
$C \rightarrow A$	100%
$E \rightarrow A$	100%
$C,E \rightarrow B$	66,6%

(c) regras e sua confiança

(b) itemset após poda

Figura 4: Regras e Confiança.

Fonte: Autores, (2018).

Inicialmente, o APRIORI varre o banco de dados verificando os itemsets de tamanho 1. No exemplo, os itemsets A, B, C, D, E e F, que possuem os respectivos suportes 40%, 80%, 80%, 40%, 60% e 40%. Portanto, apenas os itemsets B, C e E são frequentes.

Após esta etapa, o algoritmo gera itemsets de tamanho 2 a partir dos itemsets de tamanho 1 frequentes. Em seguida, é verificado se os itemsets gerados são frequentes. Em nosso exemplo, os itemsets de tamanho 2 gerados com seus respectivos suportes são BC (60%), BE (20%), CE (60%). Portanto apenas BC e CE são frequentes.

Para gerar itemsets de tamanho 3, itemsets de tamanho 2 frequentes são combinados. A seguir, os itemsets de tamanho 3 que contenham itemsets de tamanho 2 não frequentes são podados. Neste exemplo, o itemset BCE não foi podado, pois os itemsets BC e CE são frequentes. O resultado da poda nos itemsets de tamanho 3 pode ser visto na Figura 4(B).

Após a poda o suporte dos itemsets não podados é verificado. A partir disso, o algoritmo prossegue realizando estes passos iterativamente para itemsets de tamanho maior, até que seja obtido um conjunto vazio.

As regras de associação são geradas a partir de itemsets de tamanho 2 ou maiores. Um itemsets pode gerar mais de uma regra, dependendo da escolha do antecedente e conseqüente. Por exemplo, o itemsets AB pode gerar as regras $A \rightarrow B$ ou $B \rightarrow A$. Para todas as regras, o algoritmo verifica se a confiança da regra atende a confiança mínima. Por exemplo, a regra $C, E \rightarrow B$ possui confiança de 66%, que é inferior à confiança mínima de 80%, portanto esta regra é eliminada. A Figura 4(C) exhibe algumas regras com sua confiança.

Neste estudo de caso, serão utilizados todos os cinco atributos relacionados a utilização de recursos computacionais de um SGBD Oracle. Este conjunto de dados é composto cinco atributos e cento e vinte instâncias.

III. RESULTADOS E DISCUSSÃO

Seguindo o processo do KDD descrito anteriormente na Figura 1, em sua quarta etapa, que envolve mineração dos dados, utiliza-se o algoritmo APRIORI através do Weka, onde surge a necessidade de carregar o arquivo .csv gerado na etapa de transformação, após ter sido carregado pelo Weka, salvamos com extensão .arff, formato utilizado pela ferramenta.

Ao executar a mineração de dados utilizando o APRIORI, com seu valor de suporte mínimo 0.1 (10%) e suporte máximo 1.0 (100%), e seu valor de confiança 0.90 (90%). Obtém-se como resultados as principais regras que atendam aos valores de suporte ordenados pelo valor de confiança, conforme exibido na Figura 5.

Instances:	120	
Attributes:	5	
	HORARIO	
	IOREAD	
	IOWRITE	
	LOAD	
	TRANSACOES	
1.	HORARIO=madrugada	TRANSACOES=muito baixo <conf:(1)>
2.	HORARIO=madrugada	LOAD=medio ==> TRANSACOES=muito baixo <conf:(1)>
3.	HORARIO=madrugada	IOWRITE=muito baixo ==> TRANSACOES=muito baixo <conf:(1)>
4.	HORARIO=madrugada	IOWRITE=muito baixo LOAD=medio ==> TRANSACOES=muito baixo <conf:(1)>
5.	HORARIO=madrugada	IOREAD=baixo ==> TRANSACOES=muito baixo <conf:(1)>
6.	HORARIO=madrugada	IOREAD=baixo LOAD=medio ==> TRANSACOES=muito baixo <conf:(1)>
7.	IOREAD=baixo	IOWRITE=muito baixo TRANSACOES=muito baixo ==> LOAD=medio <conf:(0.96)>
8.	HORARIO=madrugada	IOWRITE=muito baixo ==> LOAD=medio <conf:(0.96)>
9.	HORARIO=madrugada	IOWRITE=muito baixo TRANSACOES=muito baixo ==> LOAD=medio <conf:(0.96)>
10.	HORARIO=madrugada	IOREAD=baixo ==> LOAD=medio TRANSACOES=muito baixo <conf:(0.96)>
11.	HORARIO=madrugada	IOREAD=baixo ==> LOAD=medio <conf:(0.96)>
12.	HORARIO=madrugada	IOREAD=baixo TRANSACOES=muito baixo ==> LOAD=medio <conf:(0.96)>
13.	HORARIO=madrugada	IOREAD=baixo ==> LOAD=medio TRANSACOES=muito baixo <conf:(0.96)>
14.	HORARIO=noite	IOREAD=baixo ==> LOAD=medio <conf:(0.96)>
15.	IOREAD=baixo	TRANSACOES=muito baixo ==> LOAD=medio <conf:(0.96)>

Figura 5: Regras obtidas com cinco atributos e cento e vinte instâncias.

Fonte: Autores, (2018).

As regras listadas na Figura 5, foram geradas a partir da análise de todos atributos, oriundos do arquivo carregado pela ferramenta anteriormente, onde após análise de todas as regras geradas, foram separadas a seguir:

2. *HORARIO=madrugada* *LOAD=medio* ==> *TRANSACOES=muito baixo* <conf:(1)>

9. *HORARIO=madrugada* *IOWRITE=muito baixo* *TRANSACOES=muito baixo* ==> *LOAD=medio* <conf:(0.96)>

O item 2, com confiança de 1 (100%), afirma que no horário da "madrugada", com uma carga de processamento "média", o SGBD tem uma quantidade de transações "muito baixo". O item 9, com confiança de 0.96 (96%), afirma que no horário da "madrugada", o acesso de escrita em disco "muito baixo", quantidade de transações "muito baixo" e a carga de processamento é "médio". A partir dos itens 2 e 9, obtém-se um padrão de consumo de recurso durante o horário da madrugada.

Uma nova mineração é realizada, retirando as instâncias "madrugada" e "noite", para se obter regras de outros períodos do dia, permanecendo o valor de suporte mínimo 0.1 (10%) e suporte máximo 1.0 (100%), e seu valor de confiança 0.90 (90%), resultando em cinco atributos e sessenta instâncias, conforme resultado na Figura 5.

Instances:	60	
Attributes:	5	
	HORARIO	
	IOREAD	
	IOWRITE	
	LOAD	
	TRANSACOES	
1.	TRANSACOES=baixo	==> HORARIO=tarde <conf:(1)>
2.	HORARIO=manha	TRANSACOES=medio ==> IOWRITE=baixo <conf:(1)>
3.	LOAD=medio	TRANSACOES=baixo ==> HORARIO=tarde <conf:(1)>
4.	IOWRITE=baixo	TRANSACOES=baixo ==> HORARIO=tarde <conf:(1)>
5.	HORARIO=manha	LOAD=medio TRANSACOES=medio ==> IOWRITE=baixo <conf:(1)>
6.	HORARIO=tarde	IOWRITE=baixo TRANSACOES=alto ==> IOREAD=medio <conf:(1)>
7.	IOWRITE=baixo	LOAD=medio TRANSACOES=baixo ==> HORARIO=tarde <conf:(1)>
8.	HORARIO=tarde	IOWRITE=baixo LOAD=medio TRANSACOES=alto ==> IOREAD=medio <conf:(1)>
9.	IOREAD=baixo	TRANSACOES=baixo ==> HORARIO=tarde <conf:(1)>
10.	IOWRITE=baixo	TRANSACOES=muito alto ==> IOREAD=medio <conf:(1)>
11.	IOREAD=medio	IOWRITE=baixo TRANSACOES=medio ==> LOAD=medio <conf:(1)>
12.	TRANSACOES=baixo	==> LOAD=medio <conf:(0.9)>
13.	HORARIO=tarde	TRANSACOES=baixo ==> LOAD=medio <conf:(0.9)>
14.	TRANSACOES=baixo	==> HORARIO=tarde LOAD=medio <conf:(0.9)>

Figura 6: Regras obtidas com cinco atributos e sessenta instâncias.

Fonte: Autores, (2018).

De todas as regras geradas, mostrada na Figura 6, separamos as mais relevantes para o estudo a seguir:

2. *HORARIO=manha* *TRANSACOES=medio* ==> *IOWRITE=baixo* <conf:(1)>

8. *HORARIO=tarde* *IOWRITE=baixo* *TRANSACOES=alto* ==> *IOREAD=medio* <conf:(1)>

O item 2 com confiança de 1, afirma que durante o horário da "manhã", com uma quantidade de transações "média", o acesso de escrita em disco é "baixo". O item 8 com confiança de 1, afirma que durante a "tarde", com acesso de escrita em disco "baixo", e quantidade de transação "alta", o acesso de leitura em disco é "médio".

Com as regras selecionadas a partir das duas minerações, conclui-se o processo do KDD, que ao ser analisada as afirmações geradas por estas regras, leva a um conhecimento do consumo de recurso computacional de um SGBD Oracle em alguns períodos do dia, nos levando a afirmar as seguintes premissas:

- ✓ Durante a madrugada tem uma quantidade muito baixa de transações, mas carga de processamento é médio e consumo de escrita em disco é muito baixo.
- ✓ O horário da manhã tem uma quantidade média de transações e o consumo de escrita em disco é baixo.
- ✓ Durante o horário da tarde o consumo de escrita em disco é baixo, quantidade alta de transações e um consumo de leitura em disco é médio.

Portanto, baseado nestas definições, podemos afirmar que a escrita em disco durante todos os períodos do dia, que são "manhã", "tarde", "noite" e "madrugada", possui um consumo baixo ou muito baixo, e que instruções de insert's no SGBD em qualquer destes períodos não degradariam sua performance. Já durante o horário da tarde, o SGBD apresenta uma grande quantidade de transações, e apesar de ter um consumo de escrita baixo, possui um consumo de leitura médio, sendo assim um horário que possui muitas transações realizando leituras de dados.

Diante de uma eventual necessidade de carga de dados nesse banco de dados, pode-se afirmar com total certeza, a partir dos resultados obtidos com a mineração de dados, que o horário da madrugada que compreende entre 00:00 às 05:59 será o melhor horário devido ao consumo de escrita em disco ser muito baixo, em comparação aos valores dos outros horários que é baixo, deixando de evitar a concorrência. Assim como para a extração de dados desse banco de dados o pior horário para a realização dessa atividade será no horário da tarde, que compreende entre 12:00 às 17:59, por apresentar um consumo de leitura em disco médio.

IV. CONCLUSÃO

A mineração de dados é muito utilizada principalmente para lidar com bases de dados muito grandes tanto em órgãos públicos quanto em organizações privadas. Com os dados que foram obtidos utilizando o algoritmo de associação APRIORI foi possível identificar padrões de consumo de recursos e suas ocorrências em períodos do dia. Os testes realizados foram limitados apenas para saber de consumos em disco, não sendo estendidos para os outros recursos no momento. Foram utilizados apenas 5 atributos e 120 instancias pois foi notado que com estes é possível obter conhecimentos com uma precisão considerável para este estudo. Sendo o objetivo abstrair conhecimento a partir das descobertas de padrões de consumo de recursos computacionais de um SGBD Oracle. O conjunto de informações relacionados à leitura e escrita em disco, transações e carga de processamento foi extraído do seu dicionário de dados, no qual foi aplicado o processo do KDD, onde este mostrou-se útil para descoberta de conhecimento. É sugerido para futuros estudos estender os testes para saber o consumo real dos outros recursos de banco de dados e até possível aplicação de outros algoritmos. Com os resultados obtidos na mineração de dados, foi possível obter os padrões de consumo de recursos e suas ocorrências em

períodos do dia, desta forma possibilitado a gestão de TI tomar decisões com índices maiores de acertos com relação a durabilidade, períodos de consumo e subutilização dos recursos, sendo de extrema importância para qualquer organização.

V. REFERÊNCIAS

[1] Hora, Gleidison Santos; Menezes, Jislane Silva Santos; Júnior, Gilson Pereira dos Santos; Neto, Almerindo Nascimento Rehem. Avaliação de Ferramentas de Mineração de Dados: Uma Abordagem com Modelo TAM. Interfaces Científicas - Exatas e Tecnológicas, V.2, N.3, p. 109 – 121, Fev. 2018.

[2] Silva, Renato Etevaldo. “Dicionário de Dados do Oracle.”. 2016. Disponível em: < <https://vidadba.wordpress.com/2016/10/03/dicionario-de-dados-no-oracle/> >. Acesso em: 16/08/2018.

[3] Silberschatz, A.; Korth, H. F.; Sudarshan, S. Sistema de Banco de Dados. 6ª. ed. Rio de Janeiro - RJ: Elsevier, 2012.

[4] Farias, Vinicius Tasca; Facó, Julio Francisco Blumetti; Andrade, Alexandre Acácio. Mineração de Dados para Análise de Banco de Dados Empresariais. XVII Safety, Health and Environment World Congress. p. 159-164. Julho, 2017.

[5] Carmo, William. “Dicionário de dados.”. 2011. Disponível em: < <http://aprenderoracle.com/2011/05/13/dicionario-de-dados/> >. Acesso em: 16/08/2018.

[6] Costa, Felipe Garcia. Visualização de Dados e Sua Importância na Era Big Data. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2017.

[7] Rabbi, Bruno; Rabbi, Diane Bossler Klug; Gonçalves, Virginia Siqueira; Junior, Elias Rocha Gonçalves. BRASIL, Juliana Araujo. Mineração de Dados Aplicada a Base de Reclamações sobre Produtos e Serviços do Programa de Proteção e Defesa do Consumidor. Braz. J. of Develop., v. 4, n. 5, Edição Especial, p. 1689-1701, ago. 2018.

[8] Fernandes, Warley Leite. Pitangui, Cristiano Grijó. Andrade, Alessandro Vivas. Assis, Luciana Pereira. Previsão de Desempenho de Estudantes usando o Algoritmo de Classificação Associativa. VI Congresso Brasileiro de Informática na Educação (CBIE 2017).