KDD-NetManager: uma Ferramenta de Descoberta de Conhecimento em Bases de Dados (DCBD) Aplicada à Gerência Proativa em Redes de Comunicação

Flávia Pereira de Carvalho[12], Marcelo Azambuja[12], Jorge Guedes Silveira[2]

[1] Faculdade de Informática de Taquara
Faculdades de Taquara (FACCAT)
Av. Oscar Martins Rangel, 4500 - Taquara, RS - Brasil.

[2] Laboratório GPARC – Faculdade de Engenharia Elétrica
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Avenida Ipiranga, 6681 - Porto Alegre, RS – Brasil.

{fpereira, azambuja}@faccat.br, guedes@ee.pucrs.br

Resumo. *Este trabalho apresenta estudos e resultados relacionados ao uso de Descoberta de Conhecimento em Base de Dados (DCBD) sobre a base de uma ferramenta de Gerenciamento de Redes. Esses resultados foram obtidos através do uso da ferramenta de Mineração de Dados KDD-NetManager, desenvolvida no transcorrer desta pesquisa, aplicada à base do Sistema Gerente Free Network Management System (FreeNMS), sistema este também desenvolvido pelo grupo de pesquisa envolvido neste trabalho. Através dos testes realizados com a ferramenta KDD-NetManager sobre a base de dados da empresa testbed utilizada nos testes, foi possível comprovar o grande benefício que as técnicas de DCBD podem fornecer para a área de gerenciamento de redes, auxiliando e sugerindo tomada de decisões.*

1. Introdução

Técnicas de Descoberta de Conhecimento em Bases de Dados (DCBD) são alvos de estudo já há algum tempo pela comunidade científica. O volume de dados disponível nos SGBDs (Sistemas Gerenciadores de Bancos de Dados) das corporações é cada dia maior, tornando os sistemas de DCBD um meio de encontrar informações úteis "escondidas" nestes enormes volumes de dados [Feldens 1997].

A motivação para o uso de técnicas de DCBD neste trabalho ocorreu devido a esse crescente aumento do volume de dados que também ocorre em Sistemas de Gerenciamento de Redes (NMS – Network Management System). Com o auxílio da ferramenta de DCBD implementada durante as pesquisas, foi possível gerar ações de pró-atividade nas atividades de operação e manutenção da rede de um grande provedor de serviços de rede. Especificamente, os testes envolveram as informações do Sistema de Registro de Problemas (TTS – Trouble Ticket System) da referida empresa utilizada como plataforma de testes (*testbed*). As funcionalidades implementadas através do uso de técnicas de DCBD permitiram, por exemplo, que imediatamente à abertura de um "chamado" no sistema TTS (registro de um problema sobre um equipamento ou serviço), o sistema KDD-NetManager indique (apontando inclusive a "precisão" desta dica) o motivo para a falha registrada, bem como a solução para esta falha.

O referido sistema TTS é parte integrante da Plataforma de Gerenciamento de Redes FreeNMS [FreeNMS 2004], também desenvolvido pelo grupo que compõe este

trabalho. Técnicas e conceitos de RNA (Redes Neurais Artificiais), Gerência de Redes Baseada na Web (Web-based Management) e desenvolvimento baseado em Softwares Livres foram utilizados neste trabalho. Em [Fayyad 1995] pode-se encontrar os principais conceitos e técnicas da área de DCBD.

2. Ferramenta de Gerência de Redes - FreeNMS

O Free Network Management System (FreeNMS) é uma plataforma de Gerência de Redes que possui entre suas funcionalidades, características de Gerência de Falhas, Performance, Contabilização, Configuração e Segurança [FreeNMS 2004]. As RFCs (Request for Comments) 1156 e 1213, bem como [Soares 1995] e [Perkins 1997] relatam com detalhes o modelo de gerenciamento SNMP de redes.

Como adicional, o FreeNMS possui um Sistema de Registro de Problemas (TTS – Trouble Ticket System) que permite o registro e acompanhamento de todos os problemas que ocorrem na rede gerenciada ([Melchiors 1999] discute a necessidade e conceitos de sistemas TTS). Como exemplo de algumas características do TTS-FreeNMS pode-se citar o registro completo de todos os problemas em elementos de rede, encaminhamento de problemas para o corpo técnico responsável de forma automática e a manutenção do histórico de todas as ações tomadas, referentes a um problema, desde a sua abertura até o seu fechamento, além da integração absoluta com o sistema FreeNMS [Lunardelli 2002].

3. Especificação e Projeto do Sistema KDD-NetManager

Nesta seção são descritos todos os procedimentos técnicos envolvidos no estudo de caso prático realizado para DCBD no banco de dados do Sistema FreeNMS, utilizando o sistema de DCBD KDD-NetManager.

De acordo com [Feldens 1997] e [Fayyad 1995], os sistemas de DCBD costumam ser "especialistas". Isto significa que estes sistemas são desenvolvidos tendo em vista uma série de características "esperadas" na base de dados que será utilizada nas descobertas de conhecimento. Alguns destes sistemas chegam a ser implementados para uma base de dados específica "única", com nomes de campos e tipos de dados invariáveis (somente os "dados" realmente variando).

Desta forma, o protótipo aqui implementado não foge a esta regra (trabalhos estado-da-arte em DCBD visam melhorar estes algoritmos e sistemas disponíveis, tornando-os menos especialistas ou dependentes do tipo e fonte de dados disponível [Uludag 2003]).

3.1. Ambiente Operacional e de Desenvolvimento

Como Sistema Operacional e servidor Web (HTTP) foram utilizados os Softwares Livres GNU/Linux e Apache, respectivamente. PostgreSQL foi o SGBD acessado e utilizado como base de dados (este SGBD era o mesmo utilizado no ambiente *testbed* – uma empresa parceira do projeto FreeNMS). Para o desenvolvimento foram utilizadas as linguagens de desenvolvimento C e PHP. Naturalmente, SQL (Structured Query Language) foi utilizada intensivamente para o desenvolvimento dos módulos, haja visto a necessidade de acessos constantes ao Banco de Dados (BD). O ambiente e interface para os usuários é totalmente baseado na Web.

3.2. Tipos de Descobertas Implementadas

Genericamente, os tipos de descobertas implementadas foram: Dependência (Regra Curta), Descrição de Conceitos (Aprendizado Supervisionado), Identificação de Classes (Clusterização) e Detecção de Desvio. Houve casos onde junções destas diversas técnicas permitiram criar algumas das funcionalidades pretendidas para o sistema. Maiores detalhes sobre "tipos de DCBD" podem ser vistos em [Fayyad 1995], [Feldens 1997] e [Carvalho 2003].

3.3. Resultados do Protótipo KDD-NetManager

A seguir são listados alguns dos "tipos de descobertas" que foram implementados no protótipo do Módulo DCBD.

3.3.1. Tipo de Descoberta: Dependência (Regra Curta)

Este tipo de descoberta é baseado na dependência entre dois atributos existentes nas tabelas do BD. Quando um valor pode influenciar outro, há uma dependência (ou relação) entre estes valores, e muitas regras podem ser descobertas a partir desta constatação.

Basicamente, o algoritmo procura campos nos diversos registros cujos conteúdos sejam idênticos (entre os registros). Encontrando semelhanças, "contadores" vão sendo incrementados. Aquelas semelhanças, entre todos os registros pesquisados, cujos contadores alcancem um número mínimo de "suporte", serão considerados "regras".

Um exemplo de regra que poderia ser encontrada por esta descoberta na base de dados deste protótipo:

> **SE** < modelo_equipamento = Cisco 2500 > **ENTÃO** < falha = placa queimada > (com 45% de "suporte"[1])

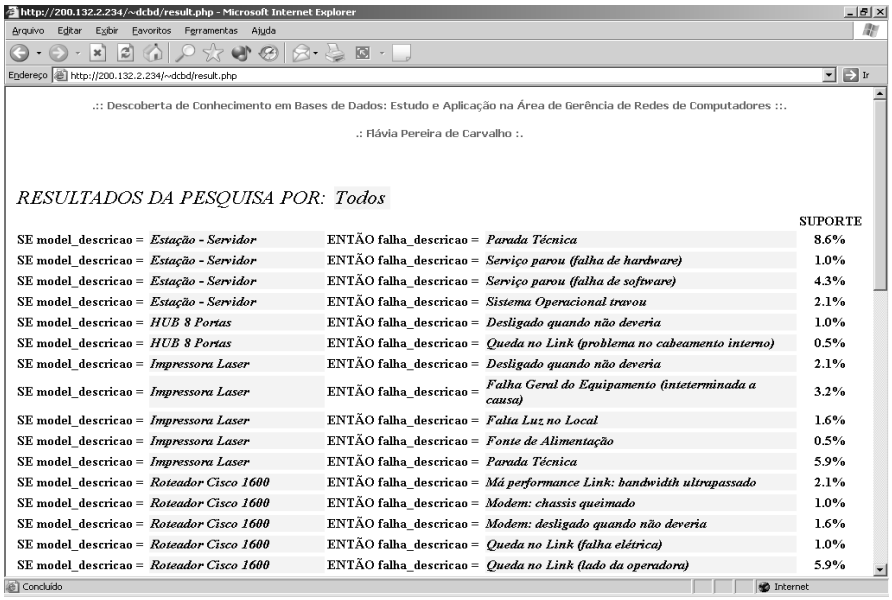

Figura 1. Tela do Protótipo, Utilizando "Regra Curta".

[1] "Suporte" (estatístico): dentre os casos disponíveis para a análise realizada, esta indução é baseada dentro de um percentual de, por exemplo, um mínimo de 45% dos casos.

3.3.2. Tipo de Descoberta: Identificação de Classes (Clusterização)

Enquanto no tipo de descoberta "Descrição de Conceitos (aprendizado supervisionado)" (seção 3.3.3, a seguir) há uma orientação clara e definida sobre como a base de conhecimento deve ser utilizada (os atributos e as classes são previamente definidas e informadas para o processo de DCBD), no tipo de descoberta "Identificação de Classes" (como o nome já deixa claro) as classes não são induzidas pelo software, mas sim necessitam ser detectadas e determinadas pelo sistema de DCBD. Um exemplo de resultado obtido:

- equipamentos tipo "switch", com falhas do tipo "desligado incorretamente", voltaram a repetir o mesmo problema em um prazo menor do que 10 dias.

O algoritmo deverá iniciar lendo o primeiro registro existente na base de dados e compará-lo com todos os demais registros da base (e assim sucessivamente, para cada registro). Os algoritmos que possuírem semelhanças entre si poderão vir a ser considerados clusters (se alcançarem um valor mínimo de "suporte"). No algoritmo implementado, é realizado um ranking das "relações de registros" (dois ou mais registros semelhantes) que mais características idênticas foram localizadas durante a pesquisa envolvendo todos os registros. As semelhanças entre os grupos de registros é que permitirão a identificação de clusters.

3.3.3. Tipo de Descoberta: Descrição de Conceitos (Aprendizado Supervisionado)

Este tipo de descoberta consegue determinar a classe que cada tipo de dado pertence, baseada em características previamente conhecidas (ou descobertas no mesmo momento, iterativamente) e comuns entre os membros de cada classe. Este tipo de descoberta pode colaborar na "aprendizagem baseada em exemplos".

No protótipo implementado neste trabalho, a Descrição de Conceitos usa como base de conhecimento os clusters identificados anteriormente pela descoberta Clusterização. Dada uma determinada Falha, em um determinado Equipamento (isto sendo "analisado" no momento da abertura de um novo "chamado" no sistema), esta descoberta aponta qual a possível solução, conforme demonstram as Figuras 2 e 3.

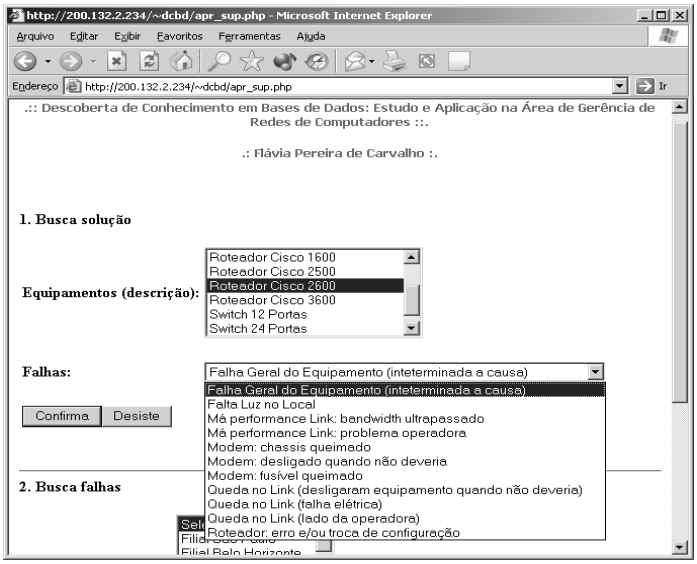

Figura 2. Rotina de DCBD do protótipo "Aprendizado Supervisionado".

3.3.3.1. Exemplo de Descoberta Realizada com Descrição de Conceitos

Neste exemplo, dado o Modelo de Equipamento, uma lista "montada" dinamicamente pelo sistema aponta todos os tipos de falhas que já ocorreram com aquele modelo de equipamento, como pode ser observado na Figura 2.

O usuário então pode selecionar o tipo de falha que o equipamento está apresentando e, com isto, o sistema consegue sugerir a solução para a falha. Um exemplo de solução indicada no momento da "abertura de chamado" simulada com esta DCBD está demonstrado na Figura 3:

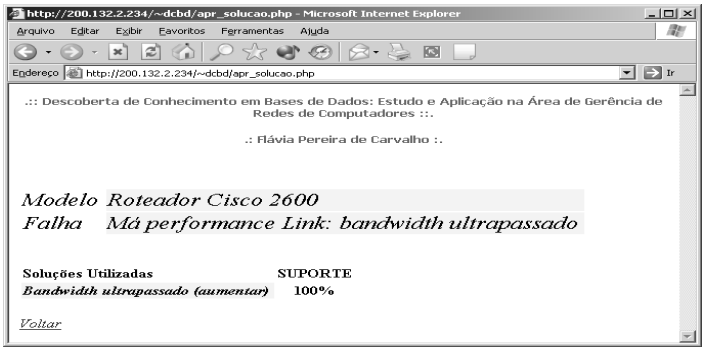

Figura 3. "Aprendizado Supervisionado" - a partir do Modelo do Equipamento e Falha, o sistema sugere a solução provável.

3.3.4. Tipo de Descoberta: Previsão

Este tipo de DCBD implementada visa "prever" possíveis problemas na rede e indicar aos administradores as prováveis próximas falhas. Estas rotinas são muito interessantes e relativamente fáceis de serem desenvolvidas. Apesar da aparente complexidade envolvida, na prática isto não acontece.

O algoritmo implementado no protótipo utiliza os seguintes artifícios:

- Registros que possuam os mesmos modelos de equipamentos, com as mesmas falhas, são analisadas e treinam a base de conhecimento que será necessária (criação de regras e de conhecimento para a rotina de verificação propriamente dita). O que interessa para a rotina de treinamento, além dos campos serem iguais, é "aprender" a média de tempo entre cada ocorrência de um determinado problema para um determinado equipamento.
- Tendo esta base de conhecimento sendo constantemente treinada, a rotina de DCBD necessita percorrer a base real de dados e comparar se há equipamentos entrando na "faixa de risco" que a base de conhecimento aponta (é feita uma comparação entre as duas bases, a cada execução do processo de Previsão).

O ideal em um sistema deste tipo é que, por exemplo, a cada madrugada o processo seja executado, e as previsões encontradas encaminhadas por e-mail para o gerente da rede e para uma tela de "alarmes" específica do sistema.

4. Conclusão

Através dos testes realizados com a aplicação da ferramenta de DCBD KDD-NetManager sobre a base de dados da plataforma de Gerência de Redes FreeNMS,

constatou-se a adequação deste enfoque no desenvolvimento de um sistema especialista de apoio à Gerência de Redes.

Os resultados obtidos não deixam dúvidas sobre a utilidade desta tecnologia quando utilizada em conjunto com um Banco de Dados de um Sistema Gerenciador de Redes típico. Nas pesquisas realizadas com os softwares líderes desta área, tais como HP OpenView e IBM Tivoli/NetView, não foram encontradas características que gerem resultados como os que foram aqui demonstrados. Desta forma, para um sistema em desenvolvimento, como é o caso do FreeNMS, que busca seu lugar no mercado, a utilização em conjunto e otimizada de um software de DCBD como o protótipo iniciado neste trabalho, poderia ser um diferencial de qualidade muito importante.

Como continuação desta pesquisa, o grupo pretende principalmente ampliar os estudos sobre algoritmos e técnicas utilizando RNA, que têm se mostrado muito indicados para trabalhos envolvendo Descoberta de Conhecimento [Mendes 2003].

Referências Bibliográficas

Carvalho, Flávia. (2003) Técnicas de Descoberta de Conhecimento em Bases de Dados Aplicadas à Gerência Proativa em Redes de Comunicação. Dissertação de Mestrado – PUCRS.

Fayyad, Usama; Simoudis, Evangelos. (1995) Tutorial: Knowledge Discovery and Data Mining. Fourteenth International Joint Conference on Artificial Intelligence. Montreal, Quebec, Canadá, IJCAI. Disponível em: <http://www-aig.jpl.nasa.gov/public/kdd95/tutorials/IJCAI95-tutorial.html#Content>. Acesso em: Jan. 2004.

Feldens, Miguel. (1997) Engenharia da Descoberta de Conhecimento em Bases de Dados: Estudo e Aplicação na Área de Saúde. Dissertação de Mestrado – UFRGS.

FreeNMS - Free Network Management System. Documentação e Descrição do Projeto. Disponível em: <http://www.freenms.org>. Acesso em: Mar. 2004.

Lunardelli, Fernando; Azambuja, Marcelo. (2002) Sistemas TTS: Uma abordagem voltada para Sistemas de Gerenciamento de Redes. Projeto Parks/PUCRS Management System. Publicações do Projeto.

Melchiors, Cristina. Tarouco, Liane. (1999) DUMBO: Uma abordagem para Gerenciamento de Falhas Utilizando Raciocínio Baseado em Casos. In: Simpósio Brasileiro de Redes de Computadores, 17., Salvador, Anais... Salvador: SBC, 1999.

Mendes, Daniele; Oliveira, Marcio. (2003) Tutorial de Redes Neurais: Aplicações em Bioinformática. Disponível em: <http://www.lncc.br/~labinfo/tutorialRN/>. Acesso em: Out. 2003.

Perkins, David. McGINNIS, Evan. Understanding SNMP MIBs. Chapter 5. Prentice Hall PTR. Upper Saddle River, New Jersey, 1997.

Soares, L.; Lemos, G.; Colcher, S. Redes de Computadores: Das LANs, MANs e WANs às Redes ATM. Editora Campus, 1995.

Uludag, Mahmut. (2003) Multi-Relational Rule Discovery. Research Progress Report, Eastern Mediterranean University, North Cyprus. Disponível em: <http://cmpe.emu.edu.tr/rila/>. Acesso em: Nov. 2003.