

EXA-01

GERENCIAMENTO DE ACESSO DE URL'S VIA SERVIDOR PROXY/INPA

Cohen, D. A.¹ ; Campos dos Santos, J. L.²

¹ Bolsista CNPq/INPA ² Orientador GTI/COAE

A rede geral do INPA (INPANet) interliga mais de 600 computadores e tem por finalidade disponibilizar serviços como webmail, compartilhamento de arquivos e impressoras, além de oferecer um link para a Internet através de um servidor proxy. A rede adota tecnologia ATM (Asynchronous Transfer Mode) composto de 10 sub-redes que atendem 3 campi: Aleixo I, II e V8, sendo a velocidade de backbone de 155Mbps escalável para 622Mbps. Tendo em vista as limitações da rede integrada quanto à quantidade de dados que pode transmitir, verifica-se a necessidade de usá-la apenas onde realmente seja necessário. O propósito de quando a INPANet foi criada era de auxiliar as pesquisas realizadas no INPA, agilizando o intercâmbio de informações em escala global. Mas, apesar de ter um objetivo direcionado, seu uso não fica restrito a ele sendo muitas vezes utilizado de modo indevido, como o acesso a *sites* considerados ilegais ou inadequados, desperdiçando recursos. Garantir o uso correto desses recursos ajudará na melhoria do desempenho desse ambiente integrado de rede.

O objetivo deste trabalho é de implementar ferramentas automáticas que possam incorporar novas funcionalidades no gerenciamento de rede, criando estatísticas sobre como foi utilizada, e restringir o uso dos recursos da INPANet, através de bloqueios de URL's censuradas, a objetivos que ajudem a pesquisa do INPA, evitando o congestionamento da rede por motivos supérfluos, principalmente nos horários em que o uso da rede torna-se mais sobrecarregada. Porém, ressaltamos que deseja-se, apenas, melhorar o desempenho da rede e o acesso à Internet e não comprometer a privacidade dos usuários. As informações serão utilizadas de forma puramente administrativa atendendo a política institucional. Algumas ferramentas foram consideradas, além de uma linguagem de programação que fosse nativa ao Linux™, com objetivo de minimizar os recursos gastos pelo sistema operacional, e uma outra que não dependesse da plataforma a ser utilizada.

A ferramenta utilizada foi a squidtaild™². Este sistema compara as informações armazenadas pelo squid™³ sobre as URL's (Uniform Resource Locator) requisitadas com um

² Ferramenta com livre distribuição e código-fonte aberto (squidtaild@usa.net)

³ Servidor proxy que gerencia a requisição de URL's em uma rede

glossário de palavras proibidas, previamente definidas, para identificar *sites* com temas censurados. Porém, o sistema não armazena informações em um banco de dados como também não possui dados sobre data e horário da ocorrência de cada item no log. As linguagens escolhidas foram C++ e JAVA™. A razão para a escolha da linguagem JAVA concentra-se em sua capacidade de possuir independência quanto à plataforma a ser utilizada; em contrapartida, necessita de muitos recursos do sistema operacional, visto que sua execução só é possível através de um interpretador. Para a escolha da linguagem C++, foi considerada a característica de ser nativa ao Linux, ou seja, não precisa de um interpretador para ser executada, diminuindo, assim, os recursos utilizados pelo sistema operacional. Todavia, é necessário um arquivo executável diferente para plataformas heterogêneas. Para a elaboração do modelo do projeto foi escolhida a linguagem de modelagem UML (Booch G., 2000)⁴ (Unified Modeling Language), que é muito flexível quanto à elaboração de modelos sem criar ambigüidades e, por ser bem difundida, possui uma vasta documentação disponível na Internet, em livros e através do próprio *site* do fabricante. O squid armazena em um arquivo informações sobre todas requisições de URL's feitas à INPANet. No arquivo contém informações sobre quem solicitou e qual a URL requisitada. Por tais razões, a solução adequada inclui:

- Um programa responsável pela leitura do log do squid e atualização do banco de dados com as informações adquiridas. O sistema implementado é chamado *cachefind*;
- O banco de dados para armazenar as URL's acessadas;
- Um programa para administrar tal banco de dados chamado *URLadmin*.

A Figura 1 apresenta o uso desses componentes na INPANet.

O *cachefind* e o banco de dados foram modelados sob critérios da UML e possuem 8 diagramas descrevendo seus funcionamentos. Os termos e conceitos utilizados pertencem à UML. O *cachefind* foi implementado através da aplicação KDevelop⁵. O sistema operacional que o mesmo é executado é o Linux. Foi escrito na linguagem C++, utilizando o compilador G++ do GNU Project⁶. O compilador G++ possui várias bibliotecas prontas e distribuídas gratuitamente, assim como o próprio compilador. A documentação é vasta, o que facilita seu uso.

⁴ Linguagem gráfica que auxilia a modelagem de um sistema

⁵ Ferramenta que auxilia a implementação e documentação de um programa (<http://www.kdevelop.org>)

⁶ Comunidade que desenvolve programas com código-fonte aberto (<http://www.gnu.org>)

O programa implementado verifica data e horário do servidor local, lê o log gerado pelo squid, interpreta tais dados obtidos e, em seguida, atualiza automaticamente o banco de dados com as informações adquiridas.

O gerenciamento de URL's requer funcionalidade para gerenciar grande quantidade de informações: O banco de dados, por conter informações sobre horários e datas, consideraria cada requisição de URL como uma nova tupla a adicionar aos dados armazenados, e como o número de requisições é muito grande, tornou-se necessário implementar mecanismo capaz de evitar tal crescimento demasiado de volume de dados no banco; antes que uma nova tupla seja criada, é verificado se a URL requisitada não foi anteriormente armazenada com datas e/ou horários diferentes e, caso positivo, apenas incrementará os contadores da tupla encontrada. Além de ser importante dispor de tais algoritmos que utilizam poucos recursos de armazenamento, também é necessário que apresentem altos desempenhos nas consultas e, principalmente, nas inserções de dados, que ocorrerão constantemente.

As implementações definitivas do Banco de Dados e do *URLadmin* não puderam ser concluídas devido a problemas ocorridos com a linguagem JAVA e a comunicação com o Banco de Dados através do JDBC⁷, mas que são passíveis de serem solucionados.

Boosh, G., Rumbaugh, J. e Jacobson, I. 2000. *UML - Guia do Usuário*. Rio de Janeiro, Brasil. Makron Books.

Bortzmeyer, S. *Manual QUICKSTART*. squid v2.3.3.

Bothner, P. *The GNU Iostream Library Reference Manual*. 1993.

Flanagan, D. 2000. *JAVA – O Guia essencial*. Editora Campus.

Jamsa, K. 1998. *Aprendendo C++*. Makron Books.

Jepson, B.; Peckham, J.; Sadasiv, R. 2000. *Database Application Programming with Linux*. Estados Unidos. Wiley Computer Publishing.

Lea, D. 1992. *User's Guide to the GNU C++ Library*. GNU Project.

Navathe E. 2000. *Fundamentals of Database Systems*, 3a edição. Estados Unidos. Addison Wesley.

Pappas, C. H.; Murray W. H. 1990. *Turbo C++ Completo e Total*. São Paulo, Brasil. Makron Books.

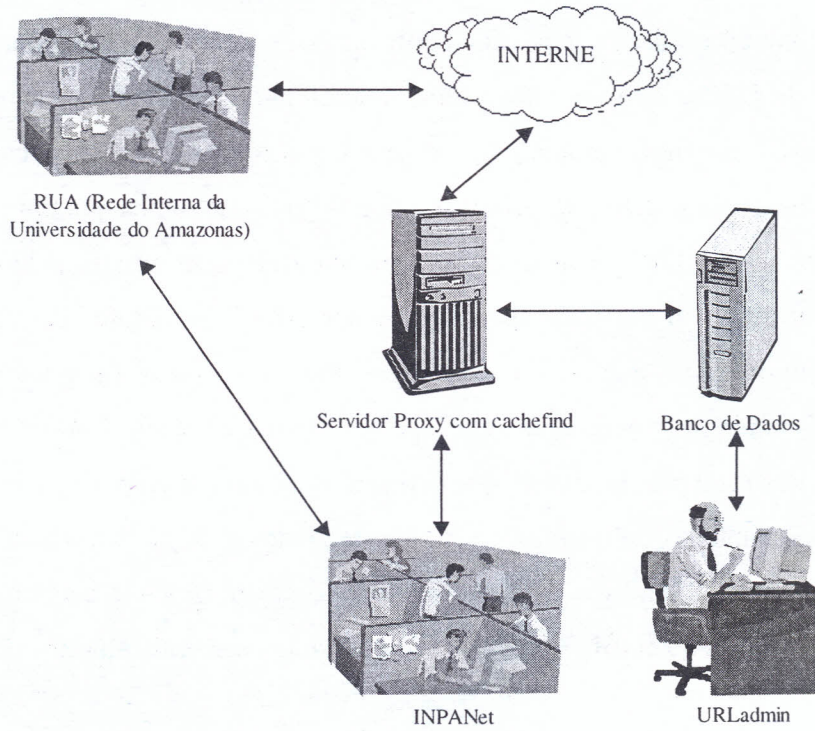


Figura 1: Disposição dos componentes do projeto

⁷ JAVA Database Connectivity