Um servidor para a classificação e filtragem de conteúdo na Internet

Marcos Forte[+] , Wanderley Lopes de Souza, Antonio Francisco do Prado

Departamento de Computação (DC) - Universidade Federal de São Carlos (UFSCar)
Caixa Postal 676 - 13565-905 – São Carlos (SP)

[+] Centro Universitário Fundação Santo André – Av. Príncipe de Gales, 821
09060-650 - Santo André (SP)

{marcos_forte,desouza,prado}@dc.ufscar.br

Abstract *In the last years the impressive growth of the Web – which includes contents rated as improper to some classes of users – has been accompanied by the appearance of new mobile access devices. In this context, one of the greatest challenges is the dynamic content adaptation to permit such devices to access certain contents independently of their original format, together with the offering of a number of added value services such as virus scanning, language translation, and content filtering. This article proposes and implements a content classification and filtering server that is inserted in a content adaptation architecture, which is based on the Internet Content Adaptation Protocol (ICAP).*

Resumo *Nos últimos anos o impressionante crescimento da Web, que inclui conteúdos impróprios para algumas classes de usuários, vem sendo acompanhado pelo aparecimento de novos dispositivos móveis de acesso. Nesse contexto, um dos grandes desafios é a adaptação dinâmica de conteúdo, a fim de que esses dispositivos possam acessar determinado conteúdo independentemente de seu formato original, aliado à oferta de uma variedade de serviços de valor agregado, tais como: escaneamento de vírus, tradução de linguagens e filtragem de conteúdo. Este artigo propõe e implementa um servidor de classificação e filtragem de conteúdo, inserido em uma arquitetura de adaptação de conteúdo, que se apóia no protocolo ICAP.*

1. Introdução

Em 1984 o mecanismo de busca Lycos iniciava a sua atividade com 54.000 páginas cadastradas e em 15 de dezembro de 2004 o mecanismo de busca Google possuía 8.058.044.651 páginas cadastradas. Isso é uma amostra do crescimento exponencial da Internet nas últimas décadas, que ocorreu não só na quantidade, mas também na diversidade, apoiada em novas tecnologias de acesso (e.g., banda larga) que propiciam novos tipos de conteúdo (e.g., vídeo e áudio em tempo real).

Uma vez que o acesso a todo esse volume de informação tornou-se um fator competitivo primordial, empresas e escolas passaram a disponibilizá-lo aos seus empregados e alunos respectivamente. Graças à convergência das tecnologias de computadores, comunicação e eletrônica de consumo esse acesso passou também a ser

realizado via uma grande variedade de dispositivos móveis (e.g., celulares, computadores de mão). Embora esse acesso ubíquo à Internet seja uma fonte para benefícios inegáveis, este pode ser também uma fonte para a distração dos empregados de suas tarefas profissionais e pode disponibilizar conteúdos inapropriados e/ou ofensivos, o que gera a necessidade do seu controle.

A pesquisa e desenvolvimento de ferramentas de classificação e filtragem de conteúdo tiveram um grande impulso nos últimos 5 anos. Não só pelo crescimento da demanda no ambiente corporativo, mas também pelo investimento governamental na área. Em 1999 a União Européia lançou o *Safe Internet Action Plan (SIAP)* [1] que tem como base o desenvolvimento de ferramentas de classificação e filtragem de conteúdo baseadas na cultura local. Os Estados Unidos lançaram em 2000 o *Children's Internet Protection Act (CIPA)* [2], que limita o envio de verbas federais somente a escolas e bibliotecas que utilizem filtros de conteúdo.

Este artigo propõe um servidor de filtragem e classificação dinâmica de conteúdo *Web*, que é um dos componentes de uma arquitetura de adaptação de conteúdo em desenvolvimento [3]. A independência de dispositivos, navegadores e sistemas operacionais é obtida por meio da implementação dessa arquitetura de adaptação de conteúdo junto a um dispositivo de borda. A flexibilidade da arquitetura de adaptação de conteúdo em atender as preferências dos usuários e limitações dos dispositivos de acesso é obtida a partir de uma política de adaptação baseada em perfis e regras. Objetivando uma arquitetura de adaptação de conteúdo aberta o *Internet Content Adaptation Protocol (ICAP)* [4] é empregado, o que permite o uso de outros servidores de adaptação disponíveis no mercado. Para demonstrar a viabilidade desse servidor foram desenvolvidos um protótipo em *C++* e um estudo de caso.

A seqüência deste artigo está estruturada da seguinte forma: a seção 2 aborda o tema classificação e filtragem de conteúdo e fornece uma visão geral do protocolo *ICAP*; a seção 3 discorre sobre os perfis e as regras de adaptação, empregados na política de controle de adaptação; a seção 4 trata da arquitetura de adaptação de conteúdo com destaque para o servidor de classificação e filtragem de conteúdo e de trabalhos correlatos; a seção 5 descreve a implementação desse servidor e um estudo de caso realizado com o mesmo; finalmente a seção 6 apresenta as conclusões relativas a esse trabalho, bem como as recomendações para trabalhos futuros.

2. Classificação e Filtragem de Conteúdo

Para controlar o acesso ao conteúdo indesejado foram desenvolvidos sistemas para classificação e filtragem de conteúdo. Como base para o estudo desses sistemas é necessária à definição de três termos inter-relacionados [5]:

- *rotular* (*labelling*) é o processo que visa descrever um conteúdo associado a um rótulo, sem que seja necessário ao usuário abrir o recipiente para examinar esse conteúdo. Esse rótulo pode ser gerado pelo próprio criador do conteúdo ou por um terceiro;
- *classificar* (*rating*) é o processo que visa atribuir valores a um conteúdo baseado em certas suposições/critérios. Caso o conteúdo disponha de um rótulo, este já possui uma pré-qualificação que pode ser (ou não) aceita pelo filtro;

- *filtrar* (*filtering*) é o processo que visa bloquear (*blocking*) o acesso a um conteúdo a partir da comparação da classificação deste com as definições de conteúdo indesejado pelo sistema.

É importante ressaltar que a classificação e filtragem de conteúdo não se restringem apenas a conteúdos ilegais (e.g., racismos, apologia à violência, pedofilia) ou inapropriados (e.g., pornografia), mas também a conteúdos indesejados numa corporação (e.g., *shopping*, *chats*, *blogs*). Entre as vantagens de se implantar este serviço num ambiente corporativo destacam-se [6,7]: proteção contra a exposição a conteúdo inapropriado, ofensivo ou ilegal que pode levar a uma responsabilidade legal; garantia de obediência às políticas internas de trabalho e sustentação de um ambiente positivo de trabalho; aumento de produtividade, preservação da capacidade de fluxo de dados da rede e melhoria do tempo de resposta, na medida em que restringe o acesso à Internet a conteúdos relativos ao trabalho.

2.1. Métodos de classificação de conteúdo

O método de classificação mais antigo e utilizado baseia-se em coleções proprietárias de *Uniform Resource Locator (URL)*, onde se associa cada *URL* a uma categoria específica de conteúdo. Quando uma página é solicitada, o classificador verifica o seu endereço no banco de dados em busca de sua categoria. Com a definição da categoria o filtro pode bloquear ou liberar o acesso ao site, de acordo com a política de uso da Internet configurada pela organização ou individuo [8]. *URLs* não localizadas no banco de dados geralmente são liberadas, sendo que os filtros podem ser configurados para bloquear o tráfego de sites não classificados.

Esses bancos são regularmente atualizados por pesquisadores, que auxiliados ou não por algoritmos de classificação revisam e categorizam manualmente cada *URL*, sendo que os usuários devem pagar uma taxa periódica de modo a manter esse serviço ativo. Manter esses bancos de dados atualizados é um desafio para os fornecedores de serviços de classificação e filtragem de conteúdo, uma vez que a taxa de criação de novas páginas na Internet é muito maior do que a capacidade destes em classificá-las.

Uma segunda geração de classificadores executa sob demanda a análise e classificação de todo o tráfego *Web* solicitado pelo usuário. Ao ser recebida uma página é analisada e categorizada de acordo com o seu conteúdo, sendo que em função da política de filtragem estabelecida o sistema bloqueia ou libera a página. Entre as diferentes técnicas para essa análise dinâmica de conteúdo destacam-se:

- *palavras chave*, onde a página tem o seu conteúdo rastreado e comparado com palavras chave pré-definidas e pré-classificadas por categoria. Quando o resultado de uma comparação é positivo a categoria da palavra chave é associada ao conteúdo rastreado. Apesar de sua fácil implementação, esse modelo leva a uma alta taxa de bloqueios indevidos de conteúdo [8];

- *análise textual*, onde é realizada uma análise do contexto no qual estão inseridas as palavras chave encontradas numa página. Geralmente há uma fase de aprendizado, onde o sistema é alimentado com exemplos e contra-exemplos da categoria a ser classificada, e uma fase de classificação, onde o sistema usa a base de conhecimento adquirida para classificar um novo conteúdo. Com essa

técnica reduzem-se os erros de classificação nos casos onde uma palavra chave pertence a duas ou mais categorias distintas (e.g., *breast* – pornografia e medicina). As abordagens mais utilizadas são *Perceptron, Naive-Bayes, MC4, Nearest-Neighbor, Rochio Centroid e Support Vector Machine* [9];

- *rótulos*, que são lidos pelo sistema de análise e classificação e onde estão inseridas, pelo produtor de conteúdo, as características do conteúdo da página solicitada. O *World Wide Web Consortium (W3C)* criou a *Platform for Internet Content Selection (PICS)* [10], estabelecendo padrões para formatos de rótulos e métodos de distribuição. Essa plataforma possui uma parte destinada ao produtor de conteúdo *Web*, que deseja ou necessita que seu conteúdo seja visto por um público específico, e uma parte destinada aos produtores de software, que implementam sistemas de classificação baseados em *PICS*, no *browser* utilizado, softwares adicionais, ou no servidor de adaptação de conteúdo.
- *análise de imagens*, onde características genéricas das imagens (e.g., cor, textura, formato) são extraídas e comparadas com imagens pornográficas armazenadas num banco de dados. Atualmente essa técnica ainda está em maturação, consome um grande volume de processamento e apresenta um alto grau de erros de classificação [11].

Devido à complexidade da análise de conteúdo, que se agravou nos últimos anos com o aumento da diversidade de formatos nas páginas *Web*, incluindo o uso de áudio e vídeo de fluxo contínuo, o processo de classificação, independente dos algoritmos utilizados, é passível dos seguintes problemas:

- *under-blocking*, quando o filtro não bloqueia algum conteúdo indesejado. Os motivos podem ser: uma base de dados de *URL* desatualizada ou, no caso das abordagens dinâmicas, uma classificação errada do conteúdo;
- *over-blocking*, quando o filtro bloqueia indevidamente um conteúdo. Geralmente ligado à classificação dinâmica de conteúdo, ocorre sobretudo quando palavras chave são usadas sem análise de contexto. Páginas de educação sexual e de medicina são as mais afetadas por esse problema [12].

2.2. Posicionamento do software de classificação e filtragem de conteúdo

A classificação e filtragem de conteúdo podem ser realizadas localmente [13], ou seja, o software é instalado no equipamento do usuário. Essa abordagem possui várias desvantagens: necessidade de instalação e configuração do software no equipamento do usuário; incompatibilidade com outros programas; necessidade de uma versão do software para cada combinação sistema operacional/dispositivo; limitações de memória e de processamento de dispositivos móveis; pode ser desativado por um usuário experiente. A principal vantagem é a pequena carga de trabalho do filtro, pois atende apenas aos acessos de um único equipamento.

A classificação e filtragem pode ser realizada externamente num *proxy* ou num servidor de adaptação de conteúdo, sendo que o *proxy* assumirá o papel de intermediário. Este posicionamento elimina a instalação de software no equipamento do usuário, independe do sistema operacional e das características do dispositivo de acesso.

A principal desvantagem é ter que dispor de uma capacidade de processamento suficiente para filtrar e classificar conteúdo de vários usuários simultaneamente. Com a arquitetura cliente-servidor propiciada pelo protocolo *ICAP*, essa capacidade pode ser expandida através do uso de vários servidores de filtragem de conteúdo, os quais irão balancear a carga de trabalho demandada pelo *proxy*.

2.3 Protocolo *ICAP*

O protocolo *ICAP* tem como objetivo possibilitar a comunicação entre equipamentos de borda e servidores de adaptação, visando adaptar o conteúdo o mais próximo do cliente, o que propicia uma melhor capacidade de personalização aliada a ganhos de velocidade. Num ambiente *ICAP* o *proxy* exerce o papel de um cliente *ICAP* e está habilitado a enviar requisições *ICAP* para o servidor de adaptação.

O cliente *ICAP* gera uma requisição *ICAP* através de um *Uniform Resource Identifier (URI) ICAP* [4]. A requisição *ICAP* encapsula cabeçalhos *HTTP* e o conteúdo *HTTP*, se este último estiver disponível. O *ICAP* dispõe essencialmente de 2 modos de operação: modificação de requisição (*reqmod*) e modificação de resposta (*respmod*).

No *reqmod* o cliente *ICAP* encapsula uma requisição *HTTP* numa requisição *ICAP* e a envia para o servidor de adaptação *ICAP* [4]. O servidor *ICAP* pode devolver a versão adaptada da requisição. O cliente *ICAP* pode então executar a requisição modificada, contatando o servidor de origem, ou enviar a requisição modificada para outro servidor de adaptação para a realização de outras modificações. Também pode enviar uma resposta *HTTP* ao usuário, provendo-lhe informação útil em caso de erro e/ou respondendo-lhe com um código de erro.

No *respmod* o cliente *ICAP* encapsula a resposta *HTTP*, recebida do servidor de origem, numa requisição *ICAP* e a envia para o servidor de adaptação *ICAP* [4]. O servidor *ICAP* pode responder ao cliente *ICAP* com uma resposta adaptada ou com o retorno de um erro. Finalizando o processo o cliente *ICAP* envia a resposta *HTTP* para o usuário. Um exemplo de aplicação para este modo é a inserção de propaganda num conteúdo obtido de um servidor de origem.

3. Política de adaptação

Um dos aspectos fundamentais de uma arquitetura de adaptação de conteúdo é a definição da política de adaptação, ou seja, quais serviços de adaptação serão oferecidos, quais adaptadores locais ou remotos executarão essas adaptações e quando estas deverão ser solicitadas.

Para que essa decisão seja tomada com eficiência as seguintes informações, relativas ao ambiente de adaptação, são necessárias: características e capacidades dos dispositivos de acesso; dados pessoais e preferências dos usuários; condições da rede de comunicação; características dos conteúdos requisitados; resoluções contratuais entre o provedor de serviços e o usuário final. Essas informações são descritas e armazenadas respectivamente nos seguintes perfis: *dispositivo*, *usuário*, *rede*, *conteúdo* e *Service Level Agreement (SLA).* Para implementar esses perfis, ilustrados na Figura 1, foi utilizado o *Composite Capability / Preference Profile (CC/PP)* [14], uma especificação geral do *W3C*.

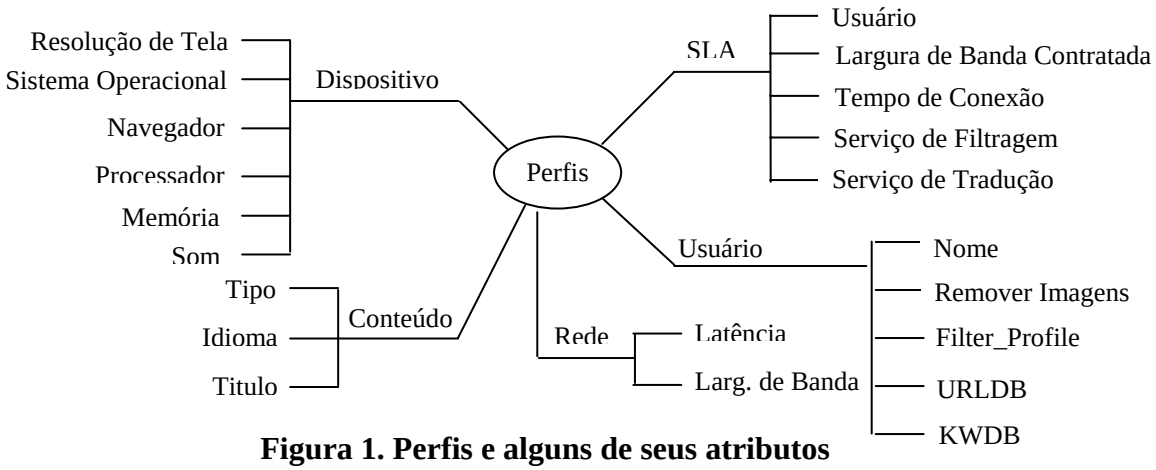

Figura 1. Perfis e alguns de seus atributos

Visando otimizar o processo de tomada de decisão, a política de adaptação considera ainda as regras de adaptação, que indicam as condições que devem ser atendidas para que determinada adaptação seja realizada e as ações a serem executadas.

3.1. Perfis

O perfil de dispositivo contém características de hardware (e.g., resolução de tela, processador) e software (e.g., sistema operacional, navegador) do dispositivo. O conhecimento das capacidades de um dispositivo orienta o processo de adaptação a fim de que apenas as adaptações necessárias sejam aplicadas ao conteúdo (e.g., remoção de som de um conteúdo requisitado por um dispositivo incapaz de reproduzir som).

O perfil de usuário, cujo fragmento de sua especificação é apresentado na Figura 2, contém informações pessoais do usuário e suas preferências de adaptação de conteúdo. Diferentes usuários podem desejar que diferentes adaptações sejam aplicadas a um conteúdo requerido (e.g., enquanto um usuário tem preferência pela retirada de imagens, um outro pode preferir a retirada de som). Adaptações não baseadas nas preferências dos usuários podem ser inconvenientes ou até mesmo indesejadas.

```
<?xml version="1.0"?>
xmlns:ccpp="http://www.w3.org/2000/07/04-ccpp#"
xmlns:usr="http://www.gsdr.br/user-schema-1_0#">
<rdf:Description rdf:ID="UserProfile">
<ccpp:component>
  <rdf:Description rdf:ID="Identification">
  <usr:UserName>mlobato</usr:UserName>
  <usr:Name>Monteiro Lobato</usr:Name>
  <usr:Email>mlobato@gsdr.br</usr:Email>
  <usr:Gender>Male</usr:Gender>
  <usr:Age>21</usr:Age>
  <usr:Occupation>Escritor</usr:Occupation>
  <usr:City>Taubaté</usr:City>
   <usr:State>SP</usr:State>
  <usr:Country>Brazil</usr:Country>
  </rdf:Description>
</ccpp:component>
<ccpp:component>
  <rdf:Description rdf:ID="Preferences">
    <usr:ImageGrayScale>0</usr:ImageGrayScale>
    <usr:ImageReduction>0</usr:ImageReduction>
    <usr:ImgDownResolution>1</usr:ImgDownResolution>
    <usr:ImageRemoval>2</usr:ImageRemoval>
    <usr:VideoGrayScale>0</usr:VideoGrayScale>
    <usr:VideoReduction>0</usr:VideoReduction>
    <usr:VideoRemoval>2</usr:VideoRemoval>
    <usr:SoundDownQuality>0</usr:SoundDownQuality>
    <usr:SoundRemoval>2</usr:SoundRemoval>
    <usr:AttachmentRemoval>2</usr:AttachmentRemoval>
    <usr:Filter_Profile>001</usr:Filter_Profile>
    <usr:UrlDB>001</usr:UrlDB>
    <usr:KWDB>005</usr:KWDB>
    <usr:Append>1</usr:Append>
    <usr:BackgroundRemoval>2</usr:BackgroundRemoval>
    </rdf:Description>
</ccpp:component>
</rdf:Description>
</rdf:RDF>
```

Figura 2. Fragmento da especificação em *CC/PP* de um perfil de usuário.

O perfil de rede é obtido dinamicamente por meio de agentes que monitoram parâmetros da rede de comunicação entre o provedor e o usuário. Parâmetros como latência e largura de banda disponível orientam alguns processos de adaptação (e.g., imagens, vídeo e áudio sob demanda) de modo que o conteúdo adaptado esteja otimizado para as condições da rede de um determinado contexto.

O perfil de conteúdo também é gerado dinamicamente, sendo baseado em características do próprio conteúdo requisitado. A partir de informações extraídas do cabeçalho *HTTP* (e.g., tipo do conteúdo texto/imagem, idioma) e do conjunto de meta-dados do conteúdo, se disponível, são determinadas as alterações necessárias e aplicáveis ao conteúdo.

No perfil *SLA* estão contidas as resoluções contratuais entre o provedor de acesso e usuário final. Atualmente os provedores de acesso oferecem diferentes planos aos seus usuários, que incluem largura de banda, tempo de conexão e vários serviços de valor agregado, permitindo que os usuários escolham o plano mais adequado as suas necessidades.

4. Arquitetura

O servidor de classificação e filtragem de conteúdo, proposto neste artigo, faz parte do projeto de uma arquitetura geral que engloba um conjunto de servidores de adaptação dedicados e um *proxy* de adaptação de conteúdo [3]. O objetivo desse projeto é disponibilizar o acesso ao conteúdo da Internet independentemente do dispositivo, meio de acesso e sistema operacional que o usuário esteja utilizando. Além disso, o conteúdo deve ser adaptado de modo a satisfazer as preferências dos usuários. A Figura 3 ilustra essa arquitetura destacando o servidor de classificação e filtragem.

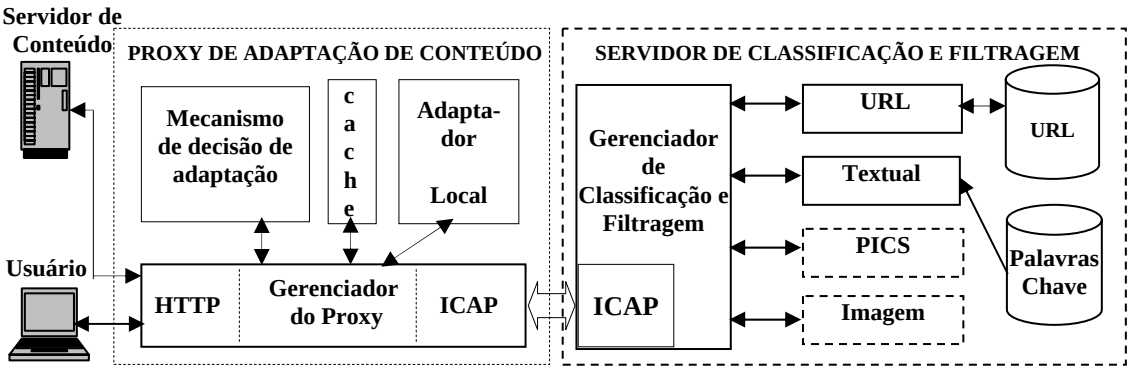

Figura 3. Arquitetura geral com o servidor de classificação e filtragem de conteúdo

Essa arquitetura baseia-se no modelo cliente-servidor, onde o *proxy* captura as requisições do usuário e as respostas do servidor de conteúdo. O mecanismo de decisão de adaptação, que implementa a política de adaptação, determina se existe a necessidade de adaptações, que tipo de adaptações devem ser efetuadas, em que ordem estas serão processadas e onde serão realizados esses serviços.

Caso a política de adaptação defina a necessidade de classificação e filtragem de conteúdo, o *proxy* enviará uma requisição ICAP ao servidor de classificação e filtragem. Para otimizar a performance e aumentar as possibilidades de uso, o servidor pode atuar nos dois modos de operação suportados pelo protocolo ICAP.

O servidor de classificação e filtragem de conteúdo foi projetado de forma modular, permitindo a fácil integração de novos módulos de classificação. O modulo gerenciador de classificação e filtragem gerencia todos os módulos de classificação, a comunicação com o *proxy* de adaptação de conteúdo, utilizando o *ICAP*, incluindo o *parser* dos cabeçalhos *ICAP* e *HTTP*, e filtra o conteúdo a partir das informações enviadas pelos módulos de classificação. Os módulos de classificação *PICS* e de imagem não fazem parte da atual implementação podendo ser objetos de trabalhos futuros.

4.1. Uso de perfis e regras na arquitetura

Para exemplificar a utilização de perfis e regras, é apresentada na Figura 4 a seqüência de uma adaptação de conteúdo, que usa o servidor de classificação e filtragem. A partir de uma requisição *HTTP* do usuário (1), o provedor de acesso envia ao *proxy* de adaptação a requisição *HTTP* juntamente com a identificação do usuário (2). O mecanismo de decisão de adaptação obtém de sua base de dados os perfis de usuário e *SLA* (3) e verifica que o usuário optou por um serviço de filtragem mais completo, o qual necessita do conteúdo requisitado e do perfil de conteúdo. O *proxy*, não tendo localizado esse conteúdo no seu cache, envia uma requisição ao servidor *Web* de origem (4) e, ao receber a resposta (5), cria dinamicamente o perfil de conteúdo. Sendo o conteúdo solicitado do tipo texto, todos os requisitos da regra do servidor de classificação e filtragem são atendidos. O mecanismo de decisão cria então uma requisição *ICAP*, anexando as preferências do usuário (destacadas na Figura 4 em negrito) e o conteúdo recebido do servidor *Web*, e as envia através do *proxy* ao servidor de adaptação (6). Este último executa a adaptação solicitada, devolve o resultado ao *proxy* (7), que por sua vez o encaminha ao usuário (8).

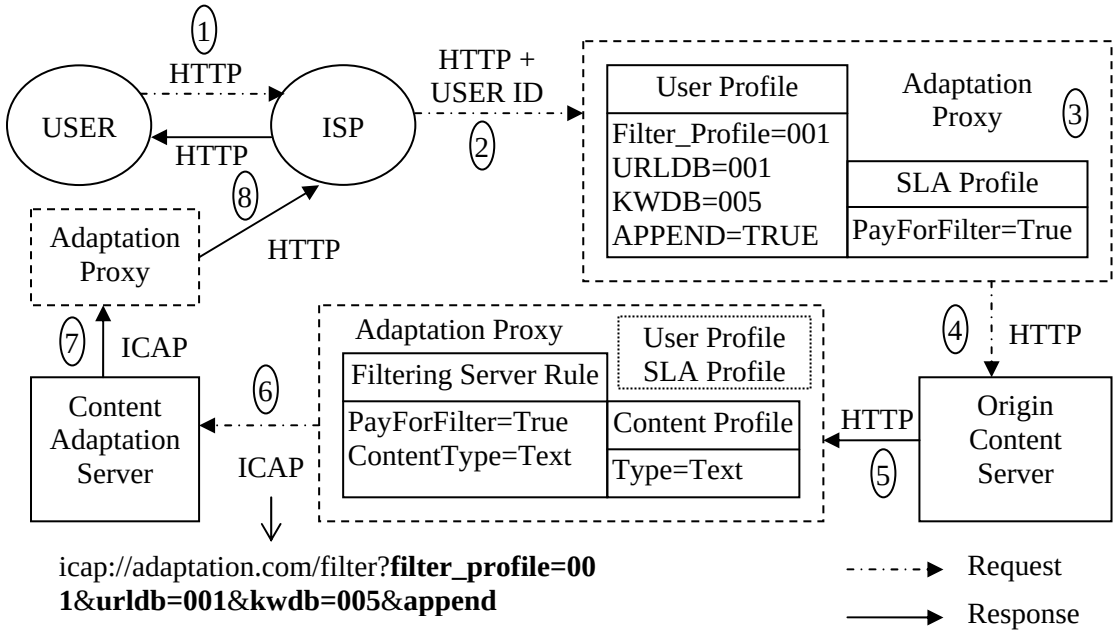

Figura 4. Seqüência de uma adaptação de conteúdo

Uma requisição *ICAP* encapsula o cabeçalho *ICAP*, o cabeçalho de requisição *HTTP*, o cabeçalho de resposta *HTTP* e o corpo da página solicitada, sendo que estes dois últimos apenas quando se opera em *respmod*. Quando esta requisição *ICAP* chega

ao servidor de classificação e filtragem de conteúdo, são extraídas do cabeçalho *ICAP* informações que definem as categorias de conteúdo a serem bloqueadas (e.g., filter_profile=001), a base de dados de *URLs* e domínios categorizados que serão utilizados (e.g., urldb=001) e, caso a adaptação esteja ocorrendo em *respmod*, a base de dados de palavras chave (e.g., kwdb=005). O domínio (*domain*) e a *URL* da página requisitada pelo usuário são extraídos do cabeçalho *HTTP* de requisição. Caso se esteja operando em *respmod* será extraído o conteúdo requisitado (*body*) possibilitando a classificação por palavras chave. A partir destas informações é realizado o processo de classificação e filtragem de conteúdo cujo modelo de estados é apresentado na Figura 5.

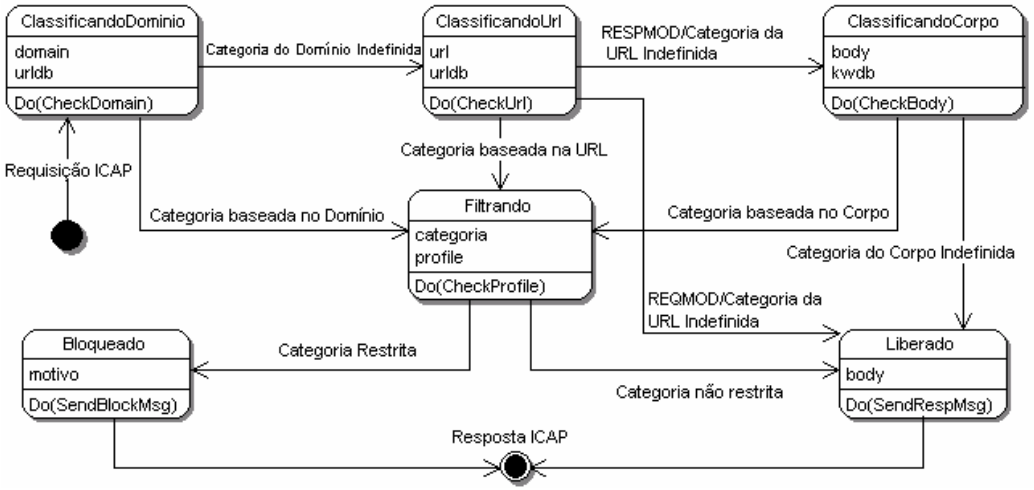

Figura 5. Modelo de estados de uma requisição ICAP para classificação e filtragem de conteúdo

4.2. Trabalhos correlatos

A comunidade européia [1] realizou nos últimos 5 anos grandes investimentos na pesquisa, desenvolvimento e divulgação de ferramentas para controle de conteúdo, baseadas na cultura européia. Um dos objetivos era o desenvolvimento de filtros que operam com conteúdos de idiomas múltiplos (e.g., Italiano, Francês, Espanhol, Alemão, Grego). Deste plano de ação surgiram 13 projetos dentre os quais, pelas qualidades técnica e documental, destacaram-se:

- ***P**ublic **O**pen-source **E**nvironment for a **S**afer **I**nternet **A**ccess (POESIA)* [15], que tem como destaques o uso de filtros textuais, que empregam técnicas de *Natural Language Processing (NLP)* para a análise lingüística de conteúdo, a identificação de rótulos *PICS* e um filtro de imagens, baseado na detecção de tons de pele e utilizado na identificação de imagens pornográficas. Na última fase desse projeto foi definido o emprego do protocolo *ICAP* nessa arquitetura;
- ***S**olution for **I**nternet Combined **F**il**T**ering (SIFT)* [16], que é um projeto em parceria com a *Internet Content Rating Association (ICRA)* [17], no qual foi desenvolvida uma solução modular, denominada *ICRAplus* [18]. Esta possibilita a conexão de filtros de vários fornecedores, onde a decisão de bloqueio é tomada a partir da combinação dos resultados desses filtros. Dentre os filtros opcionais destaca-se o *FilterX*, que usa um agente de inteligência artificial treinado para a

classificação de conteúdo de cunho sexual e que foi desenvolvido pelo *National Centre for Scientific Research "Demokritos"* da Grécia.

Também merecem destaque os softwares de código aberto *Dansguardian* [19] e *SquidGuard* [20], devido as suas otimizações de código e sua base de dados com mais de 600000 endereços pornográficos disponibilizada por este último. Cabe citar também o artigo da *IMimic Network*, onde é apresentada uma *API* [21] que adiciona a funcionalidade de adaptação de conteúdo a um *proxy*, sendo que essa adaptação pode ser realizada no próprio equipamento do *proxy* e/ou num servidor de adaptação externo via o protocolo *ICAP*.

Apesar de todos os trabalhos destacados nesta seção, este artigo vem agregar o uso de perfis, personalizando o serviço prestado pelo servidor e preservando as informações do usuário. Para ilustrar as vantagens da arquitetura e servidor propostos, pode-se imaginar um cenário no qual uma empresa presta serviços de classificação e filtragem de conteúdo para terceiros e onde: bases de dados de *URLs* específicas são disponibilizadas para cada cliente, definidas pelo cabeçalho *ICAP*; os dados dos usuários, compostos pelos seus perfis, permanecem armazenados num único local, ou seja, no seu respectivo *proxy*; a partir de uma única requisição podem ser realizadas mais de uma adaptação (e.g., um usuário que acessa a *Web* via um dispositivo móvel, além da classificação e filtragem de conteúdo tem a página adaptada em função das capacidades desse dispositivo); é possível uma regionalização e adaptação dos filtros à cultura e ao idioma locais e uma análise de outros tipos de rótulos de conteúdo, além do *PICS*, que são mais usados localmente.

4.3 Implementação

O servidor de classificação e filtragem de conteúdo foi implementado em *C++*. Foram pesquisadas várias bibliotecas e classes disponíveis publicamente, objetivando o reuso de código. Após vários testes foram incorporados ao código do servidor os softwares: *C++ Socket Class for Windows* [22]; *CppSQLite - C++ Wrapper for SQLite* [23]; e o banco de dados *SQLITE 3.07* [24].

Para acelerar a classificação, principalmente quando se opera em *respmod*, foi adicionada a categoria de sites confiáveis (*whitelist*), evitando assim a verificação de palavras chave no conteúdo obtido em endereços conhecidos (e.g., *Google*). A classificação baseada em endereço é realizada com no máximo 2 consultas à base de dados: a primeira verifica se o domínio ou o endereço *IP* estão registrados na base de dados; em caso negativo, a segunda efetua a pesquisa com base na *URL*. Quando o registro de um endereço é encontrado na base de dados, este tem associado à categoria a qual pertence. Com base nessa categoria o filtro verifica se ela pertence ao conjunto de categorias restritas definido pelo usuário (*Filter_Profile*). Caso o endereço não tenha sido encontrado na base de dados é realizada a análise do conteúdo, utilizando-se as palavras chave que estão armazenadas no banco de dados. Este é um processo que demanda uma quantidade maior de processamento e está mais sujeito a falhas.

A maior parte dos softwares de classificação e filtragem de conteúdo opera com bases de dados estáticas, sendo que qualquer atualização da base requer o reinício do software, ou que seja recarregada toda a base de dados. Para evitar esse problema foi utilizado na implementação um banco de dados relacional, que permite atualizações na base de dados sem qualquer interrupção do serviço.

Para manter o software compatível com as listas categorizadas de endereços disponíveis no mercado, comumente chamadas de *blacklists*, foi desenvolvido um módulo externo que importa as listas compatíveis com o *SquidGuard* [19] e *DansGuardiam* [20] e que estão disponíveis em *http://urlblacklist.com/*.

5. Estudo de caso

Para testar a performance e a capacidade de carga do servidor de classificação e filtragem de conteúdo uma configuração envolvendo três computadores foi implementada, o primeiro executando *Linux Fedora Core 2* (2Ghz – 256MB) e os demais *Windows 2000* (700Mhz – 256MB). Para evitar que alterações no tempo de resposta dos servidores de origem (incluindo variações no *throughput* da Internet) afetassem o resultado final, um servidor *Apache 2* foi instalado, empregando-se o recurso *hosts virtuais*. Isto permitiu a clonagem das páginas testadas, restringindo o fluxo de dados aos computadores envolvidos no teste. Na implementação do *proxy* de adaptação de conteúdo foi utilizado o software desenvolvido em [3], instalado na plataforma *Linux*, sendo adicionados perfis e regras específicos do servidor de filtragem e classificação de conteúdo.

Numa plataforma *Windows 2000* foram instalados o servidor de classificação e filtragem e a sua base de dados contendo 651620 endereços categorizados e 29 palavras chave. Para realizar os testes de carga foi utilizado, na outra plataforma *Windows 2000*, o *Webserver Stress Tool v6.0 – Enterprise Edition*, já que este permite o uso do *proxy* em testes de carga.

Cada teste consistiu em acessar 5 *links* pré-definidos durante 5 min: dois *links* não eram bloqueados pelo filtro, um era bloqueado pelo seu endereço de domínio, um outro pelo seu endereço de *URL* e o ultimo por conter palavra chave restrita. Foi aplicada uma carga progressiva de usuários, onde um usuário era adicionado a cada 1,5s até o limite de 200 usuários, com cada usuário clicando num link a cada 5 s.

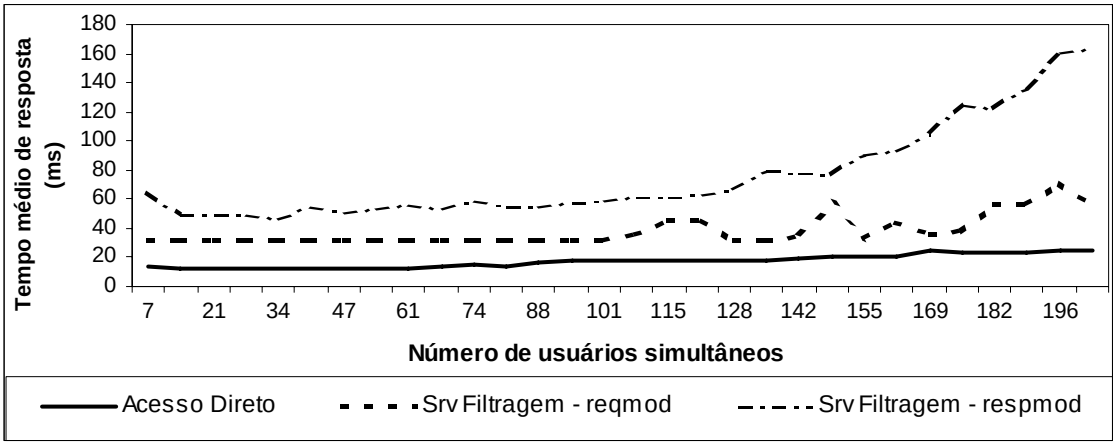

Figura 6. Tempo Médio de Resposta x Número de Usuários

Foram definidos três cenários de teste: acesso direto ao servidor *Web* interno, obtendo um *Tempo Médio de Resposta (TMR)*, de 20 ms; acesso usando o servidor de classificação e filtragem de conteúdo no modo *reqmod* , TMR de 42 ms; por último foi realizado o teste usando o servidor no modo *respmod* habilitando a classificação por palavra chave, TMR de 91 ms. A Figura 6 apresenta os resultados destes cenários.

A diferença dos tempos de resposta entre os modos *reqmod* e *respmod* é devida à adição da rotina de classificação do conteúdo *HTTP* por meio de palavras chaves (uma grande consumidora de processamento) e ao aumento do fluxo de dados entre o *proxy* e o servidor de adaptação. A Figura 7 ilustra o volume de dados de cada modo.

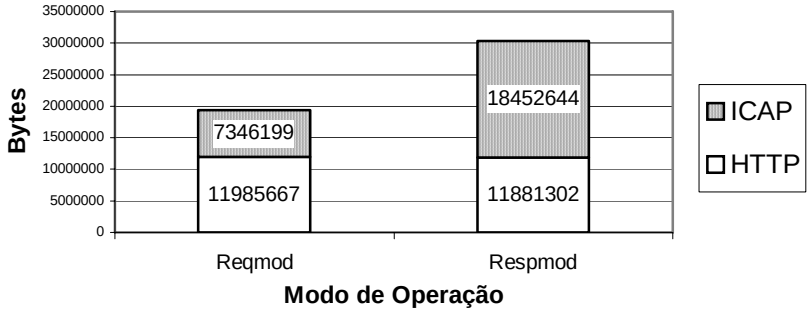

Figura 7. Fluxo de Dados x Modo de Operação

Para realizar o teste de eficiência do classificador de conteúdo, o servidor de adaptação foi instalado em um centro universitário da grande São Paulo e utilizado com o *proxy Squid*, já disponível nesse centro, no qual foi integrado um módulo *ICAP open-source* [25], demonstrando a flexibilidade do servidor em se comunicar com outros clientes *ICAP*. Foram realizados testes em dois ambientes independentes: a rede administrativa, contendo 203 computadores; os laboratórios de informática de uma das faculdades, contendo 135 computadores. Durante um período de 48 horas foram verificadas 1215854 requisições (6,8 GB) da rede administrativa e 409428 requisições (2,8GB) da rede de laboratórios. De modo a causar o menor impacto possível no tempo de resposta dos acessos de funcionários e alunos, incluindo a já saturada rede da instituição, o servidor de adaptação operou apenas no modo *reqmod*. Os resultados obtidos são apresentados na Figura 8.

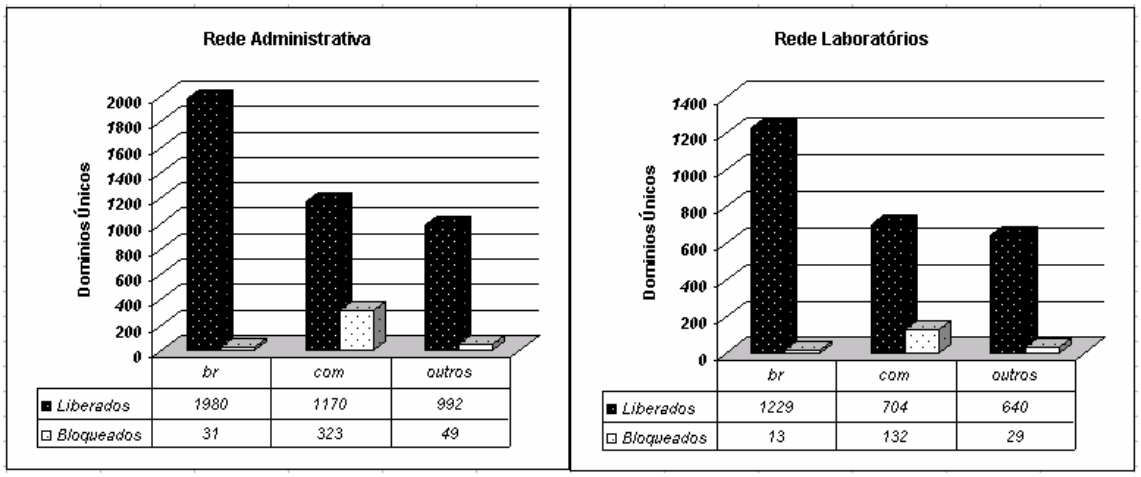

Figura 8. Resultados da Classificação e Filtragem de conteúdo do estudo de caso.

Do total de domínios classificados em ambas as redes 3 foram classificados indevidamente em categorias restritas (*over-blocking*) e 22 não foram classificados e passaram pelo filtro indevidamente (*under-blocking*). Dentre as que passaram pelo filtro, 19 pertenciam a domínios .br, o que demonstra uma menor eficiência das *blacklists* produzidas em outros países. Estes 22 domínios foram testados separadamente com o servidor de classificação e filtragem operando no modo *respmod*.

Nessa nova análise de conteúdo utilizando palavras chave, 16 foram categorizados corretamente diminuindo assim sensivelmente a margem de erro.

6. Conclusão

Baseado num modelo aberto e expansível, o servidor proposto neste artigo tem como objetivo estender as capacidades dos servidores de classificação e filtragem de conteúdo, de modo a atender as demandas de tecnologias de cliente *Web*, cada vez mais ubíquas e buscando a satisfação das preferências do usuário.

O servidor apresenta uma solução flexível, uma vez que a cada requisição vários aspectos do processo de classificação e filtragem de conteúdo podem ser definidos. É importante ressaltar que esta personalização é realizada sem ameaçar a privacidade dos usuários, pois os seus perfis são armazenados apenas no *proxy*, o que permite também a utilização simultânea de um mesmo servidor por vários provedores e/ou empresas.

Várias melhorias podem ser incorporadas a esse servidor: implementação de um classificador textual que utilize processamento de linguagem natural; classificador de imagens; melhor gerenciamento de filas e *threads*; compactação das requisições e respostas ICAP (*gzip*); incorporação de um antivírus; filtragem de conteúdos criptografados (*SSL*); filtragem de email; criação de uma base de endereços categorizados; palavras chaves específicas para as páginas brasileiras;

Esse servidor é apenas um dos componentes da arquitetura de adaptação de conteúdo proposta em [3], a qual depende do desenvolvimento e incorporação de outros tipos de servidores de adaptação, além do aprimoramento do mecanismo que implementa a política de adaptação. Quanto maior a variedade de adaptações disponíveis, mais próximo estará o objetivo de acessar a *Web* a qualquer hora, em qualquer lugar, com qualquer dispositivo.

7. Referências

[1] **Safe Internet Action Plan** - http://europa.eu.int/information_society/programmes/iap/index_en.htm

[2] Electronic Frontier Foundation. **Internet Blocking in Public Schools – A Study on Internet Access in Educational Institution**. 2003. http://www.eff.org

[3] CLAUDINO, A. T. M.; et al. **Adaptação Dinâmica de Páginas Web em Ambiente de Computação Ubíqua.** Workshop Cooperação UFSCar – FSA em Ciência da Computação. 2004

[4] ELSON, J.; CERPA, A. **Internet Content Adaptation Protocol (ICAP)**. IETF Request for Comments #3507. 2003 - http://www.isi.edu/in-notes/rfc3507.txt

[5] **OII Guide to Labelling, Rating and Filtering**. 2002. http://www.difuse.org/oii/en/labels.html

[6] N2H2 CORP. **An Introduction to Filtering Using Sentian Software.** 2002. http://www.n2h2.com

[7] N2H2 CORP. **Managing the Workplace Internet**. 2001. http://www.n2h2.com

[8] ICOGNITO Technologies Ltd. **Dynamic Filtering of Internet Content: An Overview of Next Generation Filtering Technology**. 2002. http://www.icognito.com

[9] GOLLER, Christoph. **Automatic Document Classification: A thorough Evaluation of various Methods**. http://citeseer.ist.psu.edu/context/2496479/0

[10] **Platform for Internet Content Selection -** http://www.w3.org/PICS/

[11] NETPROTECT Consortium. **Report on Filtering Techniques and Approaches**. 2001. http://www.net-protect.org/en/OPT-WP2-D2.3-v1.0.pdf

[12] The Henry J. Kaiser Family Foundation – **See No Evil: How Internet Filters Affect the Search for Online Health Information –** http://www. kff.org

[13] CSIRO Mathematical and Information Sciences. **Effectiveness of Internet Filtering Software Products. 2001.** http://www.netalert.net.au/00379-CSIRO-Filter-Report.pdf

[14] **Composite Capability/Preference Profiles** (**CC/PP**): Structure and Vocabularies 1.0. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/

[15] **POESIA** - http://www.poesia-filter.org

[16] **SIFT** - http://www.sift-platform.org/

[17] **Content Rating Association ICRA** - http://www.icra.org

[18] **ICRAplus** – http://www.icra.org/icraplus/

[19] **Dansguardian** – http://www.dansguardiam.org

[20] **SquidGuard** – http://www.squidguard.org

[21] PAI, Vivek S. **A Flexible and Efficient Programming Interface for a Customizable Proxy Cache –** http://www.cs.princeton.edu/~vivek/usits03/

[22] NYFFENEGGER, René. **C++ Socket Class for Windows**. http://www.adp-gmbh.ch/win/misc/sockets.html

[23] GROVES, Rob. **CppSQLite - C++ Wrapper for SQLite** http://www.codeproject.com/database/CppSQLite.asp

[24] **SQlite** - http://www.sqlite.org/

[25] **Squid ICAP Client Development** - http://www.webwasher.com/squid-icap/