

Sistema de Gerenciamento de Pacotes para Ambiente Heterogêneo e Distribuído – SGphd

Fábio Vieira Mello, Guilherme Bertoni Machado

Curso de Análise e Desenvolvimento de Sistemas
Faculdade de Tecnologia Senac RS (FATEC/RS)
Porto Alegre – RS – Brasil

fmello@terra.com.br, gb.machado@sinprors.org.br

***Abstract.** The simple task of manually updating a file in an application server is fairly easy, but performed the same activity on hundreds of servers became extremely costly and may be even impossible in some cases, besides being susceptible to failure, which could generate a disabling of the service. This paper describes a system that was developed to provide a package management tool in a distributed and heterogeneous environment - SGphd.*

***Resumo.** A simples tarefa de atualizar manualmente um arquivo em uma aplicação em servidor é relativamente fácil, porém realizar a mesma atividade em centenas de servidores acaba sendo extremamente custosa, podendo ser até inviável em algumas vezes, além de ser suscetível a falhas, o que poderia gerar uma indisponibilização do respectivo serviço. Este trabalho descreve um sistema que foi desenvolvido para prover uma ferramenta de gerenciamento de pacotes em um ambiente distribuído e heterogêneo - SGphd.*

1. Introdução

Com a crescente demanda de aplicações web, onde é oferecida ao usuário uma grande diversidade de conteúdos, incluindo uma gama de tipos de conteúdos entre estáticos, dinâmicos, download de arquivos e *streaming* de áudio e vídeo, requerendo cada vez mais uma maior utilização de banda de rede e maior disponibilidade.

De um lado temos estes usuários a experimentar os novos conteúdos, e que para solucionar esta demanda por banda basta contatar a operadora e mudar para um plano de banda larga que ofereça mais. Do outro temos as empresas que provêm conteúdos, que estão em uma corrida constante para conseguir atender as demandas dos clientes.

Porém acaba sendo um grande desafio para as empresas conseguirem acompanhar o crescimento da demanda, mantendo os custos e a qualidade de entrega, tanto para a empresa de mídia ou provedor, quanto para as de telecomunicações, o que faz com que sejam buscadas novas alternativas para atenuar o problema. Uma das formas de atacar o problema é utilizar uma CDN (*Content Delivery Network* – Rede de Entrega de Conteúdo).

A CDN (CDN, 2009) é uma infra-estrutura que permite entregar conteúdo ao usuário de forma distribuída, reduzindo o impacto nas aplicações que exigem qualidade de entrega (tais como: menor *buffering*, menor *jitter* e menor latência), como no caso dos sites de conteúdo *streaming*, portais e conteúdos diversos, gerando uma economia na ocupação de *links*.

Conseqüentemente administrar um ambiente distribuído e heterogêneo se torna uma atividade custosa, pois em função da quantidade de servidores e da diversidade de sistemas, tarefas simples de ajustes das configurações da aplicação, bem como tarefas de administração dos serviços, tornam-se complexas.

Esses fatos levam ao problema abordado neste trabalho, que é: como gerenciar as configurações, bem como o conteúdo a ser disponibilizado por esta CDN? Hoje, mais do que nunca, é preciso simplificar e centralizar o gerenciamento desses ambientes de sistemas para garantir a eficiência operacional do mesmo (GEYER, 2001).

Portanto, surge a necessidade da criação de uma ferramenta de gerenciamento de pacotes para ambiente distribuído e heterogêneo, o qual permita manter em uma base de dados diversas informações sobre a infra-estrutura, tais como: os servidores que a compõe, os pacotes instalados e suas respectivas versões.

Embora a idéia de um sistema de gerenciamento de pacotes para ambiente distribuído seja simples, sua exploração apresenta alguns desafios, principalmente quanto a manter a facilidade de distribuir pacotes.

Sendo assim o presente artigo se encontra organizado como segue: a seção 2 traz considerações sobre os ambientes que possuem necessidades a qual o sistema se aplica, bem como apresenta alguns conceitos que envolvem o escopo do trabalho desenvolvido. Na seção 3 são apresentados os aspectos que motivaram o desenvolvimento. A seção 4 descreve a solução e seus componentes que foram desenvolvidos. Dando continuidade, a seção 5 apresenta um caso de uso e resultados obtidos com a utilização do sistema. Por fim, a seção 6 expõe algumas considerações finais sobre o trabalho.

2. Conceitos

A fim de uma melhor compreensão, cabe esclarecer alguns conceitos envolvidos neste trabalho.

2.1. Ambiente distribuído

Um ambiente distribuído (REIS, 2008) é um conjunto de processadores interligados por uma rede de interconexão e sem memória compartilhada. A ausência de memória compartilhada exige que a interação entre processadores ocorra de uma forma distinta do ambiente centralizado: ao invés de variáveis ou arquivos compartilhados utiliza-se troca de mensagens. Um sistema distribuído é um sistema projetado para executar em um ambiente distribuído de forma transparente ao usuário.

2.2. Plataforma Heterogênea

Heterogêneo é um adjetivo usado para descrever um objeto ou sistema consistido de múltiplos itens tendo uma grande quantidade de variações de estruturas. Ou seja, o antônimo de Homogêneo, que significa múltiplos itens idênticos.

Em computação o termo refere-se a uma variedade de sistemas eletrônicos que utilizam vários de tipo de unidades computacionais. Em geral uma plataforma computacional heterogênea consiste de diversos processadores com diferentes arquiteturas, requerendo sistemas operacionais específicos para sua plataforma. Como por exemplo, Windows, Linux e MAC OS.

3. A Motivação

Conforme descrito na introdução deste trabalho, a necessidade de se administrar ambientes distribuídos e heterogêneos onde tarefas simples de ajustes das configurações da aplicação, bem como tarefas de administração dos serviços, é um problema real que se tornam custosas em função da quantidade de servidores e da diversidade de sistemas. Para tal faz-se necessário a utilização de alguma ferramenta que auxilie nestas tarefas.

Dentre os softwares encontrados no mercado, sejam vendidos por um fornecedor, ou disponibilizados através de site de projetos *open source* podemos citar os seguintes por realizarem tarefas similares a ferramenta proposta:

- Puppet (REDUCTIVE LABS, 2009) é um projeto *open source* voltado a plataformas para sistemas Unix, isso inclui distribuições Linux, FreeBSD, Solaris, MacOS, entre outras. Tem a função principal de disponibilizar configurações a um ambiente distribuído, bem como manter uma base de inventário, com a possibilidade de aplicar ações por determinado perfil.
- Tivoli Configuration Manager (IBM, 2009) é um software comercial da IBM, o qual permite o gerenciamento de plataformas heterogêneas.
- System Center Configuration Manager (MICROSOFT, 2009) é um software comercial da Microsoft que permite gerenciar os ambientes distribuídos, está fortemente dependente de outro produto da empresa, o Active Directory. Este software permite a integração com ambientes heterogêneos, porém é necessária a aquisição de um módulo adicional chamado Extend System Center Configuration Manager (QUEST SOFTWARE, 2009), fabricado pela empresa Quest Software Inc.

Um dos fatores que influenciou na decisão para o desenvolvimento de uma solução própria foi à inexistência, até o momento do início do projeto, de uma ferramenta multiplataforma *open source* disponibilizada sob a licença GPL. Isso porque os softwares encontrados ou eram vendidos como módulo de outro software, o que não atende aos requisitos deste projeto, em função da necessidade de instalar o módulo principal para só então poder usufruir do módulo desejado, ou eram softwares específicos somente para determinada plataforma.

4. O Sistema

O Sistema de Gerenciamento de Pacotes para Ambiente Heterogêneo e Distribuído – SGphd é um sistema baseado na arquitetura cliente-servidor, que mantém uma base de dados com as informações sobre ambiente e sobre os *Jobs* (tarefas de distribuição de pacotes), bem como dos pacotes, sites e informações dos usuários.

Na figura 1 são apresentados os componentes do sistema, sendo eles: O módulo de Administração via Web, o Webservice, o repositório de pacotes e o módulo do cliente, que é instalado em cada servidor a ser gerenciado.

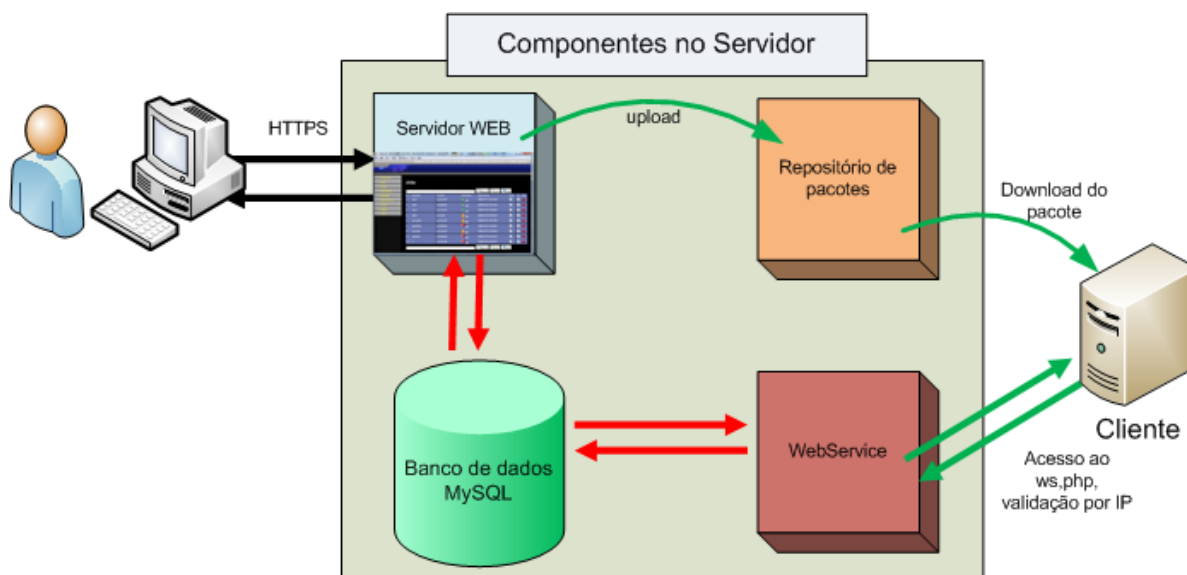


Figura 1. Componentes do Sistema

4.1. Módulo de Administração

O módulo de Administração é a interface de interação com o usuário através de um navegador, utilizando para isso uma conexão segura sobre o protocolo SSL, permitindo assim uma maior segurança.

4.1.1. Sobre o funcionamento

O módulo de administração é acessado somente através de uma conexão segura por SSL, isto é, mesmo que o usuário não acessar por HTTPS, ele será redirecionado para uma sessão SSL. O acesso é autenticado na aplicação pelo *login* do usuário com distinção por perfil.

O sistema possui dois perfis: Operador - que possui acesso somente de leitura, podendo realizar somente consultas sobre *Jobs*, pacotes, servidores e *sites*. E o perfil Administrador, que além de realizar consultas sobre *Jobs*, Pacotes, Servidores e *Sites*, possui direito de realizar inclusão, consulta, alteração e remoção dos itens, ou seja, possui acesso irrestrito, inclusive para manter os Usuários.

O módulo de administração possui um menu com as seguintes opções: Usuários, *Jobs* (figura 2), Sites, Pacotes, Servidores, Tools, Sobre e Sair.

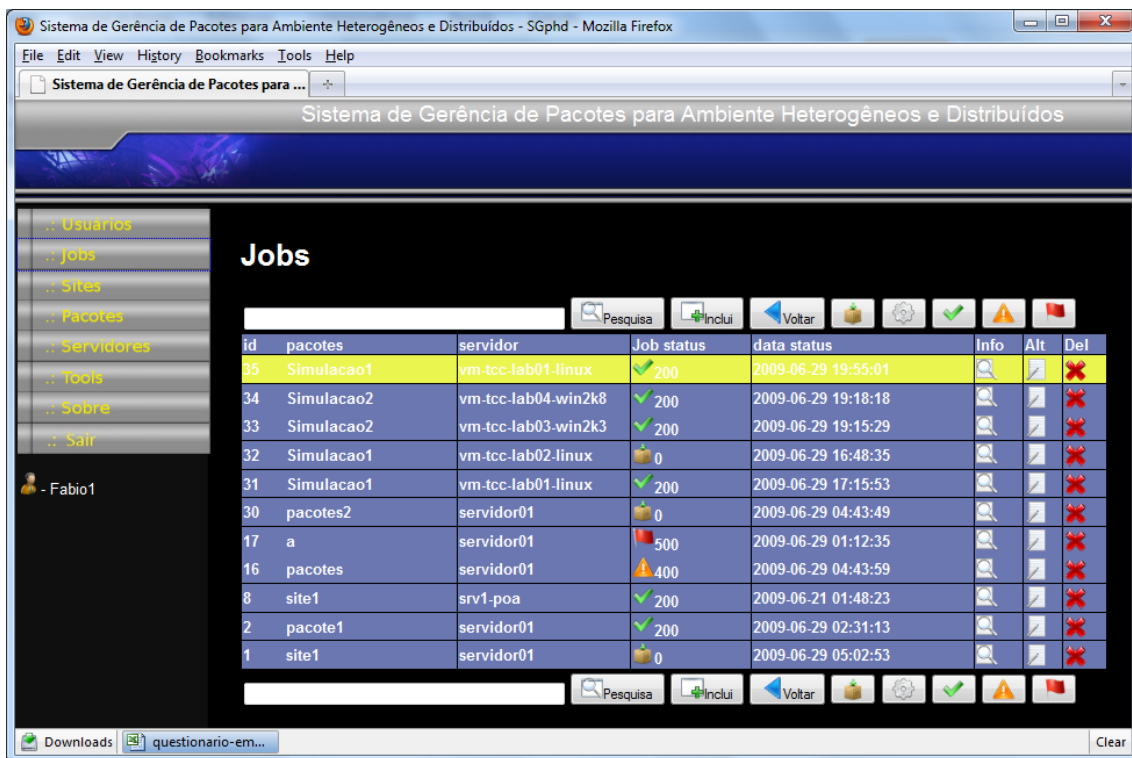


Figura 2. Tela do Sistema

A tela principal é composta pelo cabeçalho, menu lateral a esquerda e painel principal, o qual apresenta as respectivas telas referente aos itens do menu, onde cada uma delas é composta por uma tabela contendo os últimos itens inseridos e os botões de ações, que são: pesquisa, incluir, voltar e as consultas por *status*, conforme a tabela 1.

Tabela 1 – Códigos de Status do Jobs

Código	Ícone	Mensagem	Descrição
0		Disponível	<i>Jobs</i> recém criado aguardando a conexão do cliente;
100		Processando	<i>Jobs</i> que estão em processamento (<i>download</i> , executando pré-script, descompactação, cópia e executando o pós-script);
200		Ok	<i>Jobs</i> concluídos com sucesso;
400		Erro temporário	Erro temporário (<i>download</i>), perda de conectividade temporária. Recuperáveis.
500		Erro permanente	Erros no processamento, que irão requer a intervenção manual.

A partir da barra de ferramentas é possível realizar pesquisas por aproximação nos principais campos das tabelas, além disso, são disponibilizados dois botões: o Incluir caso o usuário que está logado possua o perfil de Administrador e o botão Voltar que retorna a tela anterior, desta forma permitindo maior agilidade para o usuário.

4.1.2. Sobre a implementação

A implementação do módulo de administração do sistema foram utilizadas as seguintes tecnologias: Gnu Linux Debian 4.0 como sistema operacional do servidor, juntamente com Apache Web Server 2.0, com os módulos dinâmicos de SSL e PHP 5 com o respectivo módulo de MySQL. O gerenciador do Banco de dados foi utilizado o MySQL 5 (DALL'OGGIO, 2007).

No desenvolvimento da aplicação foram utilizadas as seguintes ferramentas: Notepad++ e Vi como editores de texto, DBDesigner para fazer a modelagem do banco, Gimp para editar as imagens, Mozilla Firefox como navegador e Firebug para auxiliar na depuração de erros. A interface da aplicação foi desenvolvida em PHP 5, juntamente com Javascript, Ajax e CSS permitindo assim maior flexibilidade na formação do conteúdo.

4.2. Repositório de pacotes:

O repositório é o local publicado por um servidor Web, onde ficarão disponíveis externamente todos os pacotes distribuídos. Ao cadastrar um novo pacote no sistema, uma das etapas obrigatórias é o *upload* do arquivo, gerando assim a URL do pacote. A área de disco utilizada pelo repositório será proporcional ao volume de pacotes publicados e ao tamanho dos mesmos.

4.3. Webservice

WebService (WS) é o componente da plataforma utilizado exclusivamente para responder as consultas dos servidores clientes. O acesso ao WS fica protegido por dois mecanismos: A “*secret*”, que é uma senha estática, contida no arquivo de configuração do cliente, o qual é enviado juntamente com os demais campos da requisição do cliente ao WS e pela validação do IP da conexão (REMOTE_ADDR) com as informações do cliente cadastrado no banco. A figura 3 ilustra a comunicação entre os clientes e o Webservice.

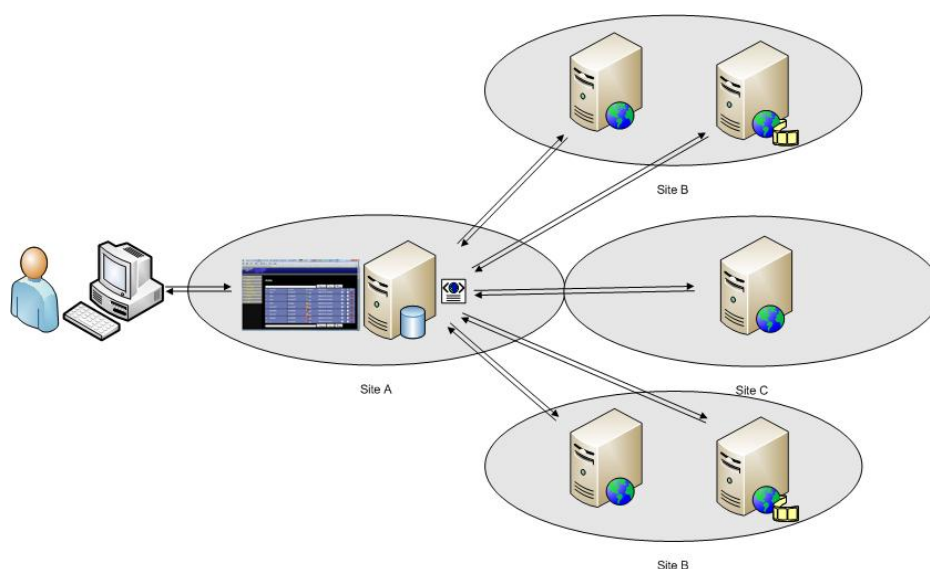


Figura 3. Comunicação entre os clientes e o Webservice

Em função dos clientes serem servidores e estarem na Internet, os mesmos utilizam IP fixo. Em caso de alteração de IP, deverá ser alterado o cadastro do servidor, pois se os dados do cliente não for atualizado no sistema os *Jobs* pendentes para este servidor ficarão com status 500 (erro permanente).

4.4. O cliente

O módulo Cliente é a parte do sistema responsável por buscar as informações dos *Jobs* no Webservice e realizar o *download* dos pacotes em intervalos de tempos definidos pelos administradores do ambiente. Também é responsável por executar os comandos antes (pré-script) e depois (pós-script) do processamento do *Job*.

O cliente foi desenvolvido em Perl, utilizando principalmente as seguintes classes XML::Simple e LWP::Simple, permitindo assim a independência de plataforma operacional, uma vez que existe interpretador Perl para todos os sistemas operacionais.

O cliente tem a possibilidade de rodar em modo *verbose*, exibindo todas as informações para *debug* na tela (ou na saída padrão).

4.4.1. Sobre o funcionamento

O cliente será executado a partir do sistema de agendamento do servidor, no caso de Linux: cron ou de Windows: gerenciador de tarefas ou *schedule task*.

Ao iniciar a execução, o cliente carrega as informações básicas a partir do arquivo config.xml, que contém os seguintes campos: ID do cliente a ser informado durante a instalação. Url_Ws, que é a url do servidor web com o Webservice (ws.php) e o campo *Secret*, que é a senha estática utilizada pelo cliente para conectar ao servidor, sendo uma medida de segurança a mais, além da validação do IP da conexão com as informações do *host* no banco de dados.

O cliente irá obter do WS os registros dos *Jobs* disponíveis (jobsstatus=0) para o seu id a partir da url que é formada pelos dados do arquivo de configuração.

Como este projeto tinha como base a premissa de que todo o sistema seria preferencialmente software livre, foi utilizada a linguagem Perl para a criação do cliente, o nos permitiu portar o cliente para outras plataformas com poucas alterações. No cliente para Windows foi necessário instalar o Strawberry Perl (PERL, 2009) versão 5.10, que é uma versão de Perl para Windows, o que permitiu a portabilidade do cliente.

Como um dos requisitos era permitir a execução de código remoto, foram adicionados dois campos, com as seguintes funções:

- pré-script: é o campo da tabela pacotes que permite que antes de aplicar a atualização de um pacotes seja executado o seu conteúdo. Pode ser utilizado para executar um script que faça backup antes de iniciar a atualização ou mesmo enviar um *e-mail* e até gravar em um *log*.
- pós-script: é o campo da tabela pacotes que tem o mesmo propósito do pré-script, devendo ser executado imediatamente ao processamento do pacote.

A figura 4 ilustra um arquivo de configuração do cliente.

```
<?xml version="1.0" encoding="UTF-8" ?>
<config>
  <servidor_id>3</servidor_id>
  <secret>aaa123</secret>
  <url_ws>http://servidor/principal</url_ws>
</config>
```

Figura 4. Arquivo config.xml

Primeira chamada ao WS:

Através de uma chamada ao método GET da classe LWP::Simple (módulo PERL para ler o conteúdo de um documento via Internet), o cliente obtém os dados em XML a partir da url abaixo:

```
http://servidor/ws.php?secret=aaa123&tipo=1&servidor=3
```

Com as informações recebidas em XML do WS, serão realizadas sucessivas chamadas (ilustrada no próximo item) ao WS, para obter os dados de cada *Job* disponível para o cliente, no qual a resposta contém entre outras informações a URL do arquivo do pacote no repositório.

Exemplo de uma segunda chamada do WS:

```
http://servidor/principal/ws.php?secret=aaa123&tipo=2&pacote=14
```

Uma vez feito o download do arquivo, o comando descrito nos campo *prescript* será executado, seguido da descompactação do arquivo do pacote na localização destino, e logo após é feita a execução do comando do campo *posscript*. Por fim, é acessado novamente o WS enviando-o status do *Job*.

5. Testes

Para validação da ferramenta foram realizados testes, tanto em um ambiente simulado como em um ambiente de produção, a fim de comprovar a eficácia do Sistema de Gerenciamento de Pacotes para Ambiente Heterogêneo e Distribuído – SGphd.

Foram realizados testes em ambiente simulado composto por cinco servidores, onde um deles (com SO Linux Debian) era responsável pelo gerenciamento e os outros quatro servidores eram clientes, sendo que esses possuíam versões diferentes de SOs (Windows 2003, Windows 2008, Linux Debian, Linux Fedora).

O experimento consistia em atualizar uma aplicação (IIS - Windows e Apache - Linux) em todos os servidores clientes. Para tanto foi necessário instalar o software cliente em cada servidor, na respectiva versão de ambiente operacional (Windows e Linux). No sistema de gerência foram cadastrados os servidores com as respectivas informações. (IP, localização, sistema operacional, etc).

Após a geração de um arquivo zip, que determinava o conteúdo a ser distribuído pelo sistema em forma de 'pacote', bem como feito o cadastro e o *upload* deste pacote no gerenciador, o SGphd pode criar a tarefa (*job*) para distribuição do pacote., além de checar o progresso da distribuição, através da sua interface (status dos *jobs*).

O ambiente de produção do teste era formado por três servidores reais, distribuídos em três localidades do Estado do Rio Grande do Sul. Estes servidores possuem o SO Linux Debian e funcionam como *firewall* e como *proxy squid*, controlando o acesso Internet, a partir das estações da rede local.

O objetivo do teste era determinar se a aplicação em um ambiente real iria realmente funcionar igual aos testes no ambiente de desenvolvimento e verificar qual o comportamento da aplicação. O teste consistia em enviar um pacote para atualização das configurações nos três servidores. Ou seja, o pacote deveria atualizar o arquivo de configuração do *squid* que contém a url dos *sites* a serem bloqueados.

```
Arquivo: sites-bloqueados.txt em /etc/squid/regras/  
pre-script: faz-bkp-conf.sh  
pos-script: restart-squid.sh
```

Executando os Testes

Para a realização dos testes foi necessário prover uma infra-estrutura real que permitisse a instalação da aplicação SGphd, ou seja, as máquinas envolvidas deveriam estar na Internet e com IP fixo. Depois de instalado e realizado os testes com o acesso a interface do módulo de administração e o acesso ao Webservice (ws.php) a partir de um navegador, foi procedida à instalação dos clientes. A instalação manual consiste em criar um diretório para o cliente, copiar os arquivos cliente.pl e config.xml para este diretório, e atribuir direito de execução ao arquivo cliente.pl, através do comando: `chmod a+x cliente.pl`

Também foi necessário instalar a libxml-simple-perl (XML::Simple) com suas respectivas dependências, concluindo a instalação do cliente foi necessário o cadastramento do servidor no Módulo de Administração, bastando apenas criar o pacote com o arquivo a ser atualizado, bem como os comandos para pré-script e pós-script.

Com isto concluído, bastou apenas criar o *Job*, relacionando os servidores e o pacote que se desejava atualizar e confirmar a criação do *Job*. O cliente foi configurado para executar a cada hora cheia, ou seja, de uma em uma hora foi verificado se existe alguma tarefa, caso houvesse, esta era executada, atualizando o *jobstatus* no banco de dados através do WS.

Todos os *downloads* foram executados com sucesso na primeira hora cheia, e levaram menos de um minuto para concluir todo o processo. Pelo meio deste teste ficou comprovado que a atualização desta forma acaba sendo muitas vezes mais rápida que o processo manual, além de se ter o controle exato de qual versão de arquivos (pacotes) estão sendo utilizadas.

6. Conclusões

Este trabalho descreveu o Sistema de Gerência de Pacotes para Ambiente Heterogêneo e Distribuído, uma aplicação para distribuir conteúdo e gerenciar aplicações em *hosts* separados geograficamente e em larga escala. Em função da diversidade de sistemas, aplicações, formas de acesso e localização, estas tarefas tendem a dificultar o trabalho do administrador de redes/sistemas em função das diversas ações a serem realizadas em ambientes heterogêneos.

Este sistema de gerenciamento de pacotes foi concebido a partir da real necessidade de manter diversos ambientes, pois para tal se fez necessário uma ferramenta de auxílio à administração destes ambientes. As soluções de gerência comerciais utilizam tecnologias proprietárias e protegidas por *copyright*, não permitindo a adequação por parte dos usuários, ou então somente através da aquisição de customizações sendo o custo bastante elevado.

Comprovadamente os ganhos com esta ferramenta são inúmeros, tanto na forma de diminuir o tempo gasto pela equipe de administração, quanto no custo indireto para manter a plataforma, liberando assim o profissional de suporte, para desempenhar tarefas inerentes ao negócio e não tão somente de manter a atual infra-estrutura. A padronização dos processos de instalação de pacotes irá ocorrer de forma mais eficiente, minimizando eventuais erros dos processos manuais.

Este sistema tem grandes perspectivas, embora ainda existam algumas melhorias que somente foram descobertas que eram necessárias com a utilização em ambientes reais. Tais como: criação de múltiplas filas de *Jobs* por severidade, implementação de controles de *timeout* mais eficientes, criação de perfil de servidor ou aplicação a fim de poder realizar o processamento de todos os *Jobs*, e histórico do perfil para uma novo servidor que for adicionado ao sistema.

Acredita-se que o Sistema de Gerenciamento de Pacotes para Ambiente Heterogêneo e Distribuído – SGphd, será uma ótimo auxílio aos administradores de ambientes distribuídos.

6.1 Limitações do Sistema

Pela arquitetura do sistema, os componentes podem ser escaláveis, permitindo ampliar a plataforma que entrega o sistema de gerência em WS, Interface WEB, Banco e Repositório. E estes, por sua vez, poderem ser clusterizados.

Em relação ao comportamento do SGphd certas limitações devem ser consideradas:

- A quantidade de conexões simultâneas ao módulo de gerência está diretamente relacionada ao hardware utilizado, limites da pilha TCP do sistema operacional e *throughput* de rede.
- A capacidade de entrega de pacotes aos clientes está diretamente relacionada ao tamanho do pacote, *throughput* de rede e limites da pilha TCP do sistema operacional.
- A capacidade de processamento do cliente depende diretamente dos recursos dos clientes para processar os comando pré-script e pós-script.

- O sistema ainda não possui um mecanismo de priorização dos *jobs* e também não possui parâmetros para impor limites ao sistema evitando esgotamento de recursos.

Outro item que deve ser considerado como fator limitante do sistema é que não foi realizada nenhuma comparação com as ferramentas citadas na sessão três, uma vez que o escopo do trabalho era apresentar uma alternativa livre para solucionar um problema real de distribuição de pacotes, evitando a utilização de softwares comerciais.

Referências

CDN. Content Delivery Network. 2009. Wikipédia, A Enciclopédia Livre. Disponível em: <http://en.wikipedia.org/wiki/Content_delivery_network> Acesso em 19 jun. 2009.

DALL'OGGIO, Pablo. PHP: Programando com Orientação a objetos. São Paulo: Novatec, 2007.

GEYER, Cláudio F. R. et al. Sistemas Distribuídos. In: Escola Regional de Alto Desempenho - ERAD, 2001, Gramado, RS. ERAD 2001 1o Escola Regional de Alto Desempenho. Porto Alegre, RS: SBC/UFRGS, 2001. Disponível em: <<http://www.inf.ufrgs.br/~kayser/papers/erad2001/eradPalestraSisDistribuidosvf.pdf>> Acesso em 20 jun. 2009.

IBM. Tivoli Configuration Manager. Disponível em: <<http://www-01.ibm.com/software/tivoli/>> Acesso em 20 jun. 2009.

MICROSOFT. System Center Configuration Manager. <<http://www.microsoft.com/systemcenter/configurationmanager/en/us/default.aspx>> Acesso em 20 jun. 2009.

PERL. Strawberry Perl. Disponível em: <<http://strawberryperl.com/>> Acesso em 20 jun. 2009.

QUEST SOFTWARE. Extend System Center Configuration Manager. Disponível em: <<http://www.quest.com/>> Acesso em 20 jun. 2009.

REDUCTIVE LABS. Puppet, the configuration management solution. Disponível em: <<http://reductivelabs.com/products/puppet/>> Acesso em 20 jun. 2009.

REIS, Valéria Quadros dos; CERQUEIRA, Renato Fontoura de Gusmão. Gerenciamento de Recursos em Ambientes Distribuídos: uma Visão do Escalonamento de Processos. Monografia em Ciência da Computação, 2008. Disponível em: <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/08_37_reis.pdf> Acesso em 20 jun 2009.