

Redes de Computadores II

Aula 03 - Web: Protocolo HTTP e servidor Apache – Parte II

Apresentação

Nesta segunda parte do conteúdo relacionada a Web estudaremos um dos principais elementos da arquitetura necessária para o funcionamento do protocolo de comunicação HTTP: o servidor *web*.

Estudaremos o servidor *web* mais popular hoje em dia: o servidor HTTP **Apache**. Em relação a esta plataforma aprenderemos a instala-lo e configura-lo, bem como adicionar e remover funcionalidades ao servidor (por exemplo, suporte a PHP).

Objetivos

Após estudar o conteúdo desta aula, você será capaz de:

- Instalar e configurar o servidor HTTP Apache.
- Adicionar e remover funcionalidades ao servidor HTTP Apache (Ex: suporte a PHP).

Servidor HTTP Apache

O servidor HTTP Apache é, juntamente com o Linux, um dos softwares livres que mais obtiveram sucesso. Ele foi criado em 1995 por um funcionário do NCSA (*National Center for Supercomputing Applications*), sendo atualmente, segundo dados da NetCraft de Agosto de 2012, utilizados em aproximadamente 60% dos servidores HTTP do mundo. Por ser um *software* livre, seu código fonte pode ser baixado diretamente do seu site, o <http://httpd.apache.org/>. Contudo, praticamente todas as distribuições GNU/Linux disponibilizam pacotes pré-configurados, de forma a facilitar sua instalação e uso. Dentre outras características do Apache, podemos destacar:

Estudaremos o servidor *web* mais popular hoje em dia: o servidor HTTP **Apache**. Em relação a esta plataforma aprenderemos a instala-lo e configura-lo, bem como adicionar e remover funcionalidades ao servidor (por exemplo, suporte a PHP).

- Implementa 100% do padrão HTTP/1.1;
- Disponível para diversos SO's, além do Linux, para o qual foi criado.
- Bastante estável e seguro;
- Suporte ao protocolo HTTPS;
- Facilmente extensível (PHP, JSP, Perl, Pythonetc) por meio de módulos extras;
- Suporte a VirtualHosts, possibilitando a hospedagem de várias páginas no mesmo servidor.



Vídeo 01 - Web



Vídeo 02 - Apache

Instalando e iniciando o servidor Apache

O Ubuntu disponibiliza diversos pacotes para instalação do Apache e de seus módulos opcionais. Dessa forma, o administrador pode instalar um ambiente que atenda idealmente as suas necessidades. Num primeiro cenário, um administrador poderia instalar apenas o servidor Apache base, para disponibilizar páginas HTML estáticas. Assim que aparecesse a demanda, poder-se-ia adicionar um módulo, para, por exemplo, interpretar páginas escritas na linguagem PHP. A listagem de pacotes relacionados ao Apache e disponíveis para instalação pode ser obtida com o comando:

```
apt-cache search apache2
```

Dentre a grande quantidade de pacotes listados, os principais estão destacados no Quadro1.

Nome do Pacote	Descrição
apache2	Servidor HTTP base
apache2-doc	Documentação
apache2-utils	Programas utilitários
libapache2-mod-*	Módulos para adição de funcionalidades ou características

Quadro 1 - Pacotes importantes do servidor web Apache

Para instalar o servidor HTTP básico, basta executar o comando:

```
apt-get install apache2
```

Após a instalação, o servidor já irá iniciar a sua execução com um conjunto de configurações padrão. Contudo, é altamente recomendável que se personalize essa configuração, para que sejam atendidas as necessidades particulares de cada servidor. Sempre que for feita alguma alteração na configuração de um servidor Apache, deve-se reiniciá-lo por meio do comando:

```
/etc/init.d/apache2 restart
```

Configurando o servidor Apache

A configuração do servidor HTTP Apache é distribuída em uma série de arquivos, conforme mostrado no Quadro 2. Cada um deles tem um propósito específico. O (*) nas linhas 2 e 3 da Tabela 2 indicam que podem existir vários arquivos dentro de cada uma dessas pastas.

Arquivo de configuração	Finalidade
/etc/apache2/apache2.conf	Configurações gerais do servidor
/etc/apache2/mods-enable/*	Configuração de módulos extras
/etc/apache2/sites-enable/*	Configuração dos sites hospedados no servidor

Quadro 2 - Configuração dos sites hospedados no servidor

Em qualquer arquivo, a configuração é realizada dando-se valores para uma série de "diretivas" que compõem os arquivos de configuração. As principais diretivas, que podem ser vistas no Quadro 3, vêm com valores padrão, de modo a

permitir o funcionamento inicial do servidor.

Diretiva	Significado
Port 80	Porta que o servidor escuta
ServerRoot /etc/apache2	Diretório base do Apache
User www-data	Usuário que irá executar o Apache
Group www-data	Grupo que irá executar o Apache
ServerAdmin root@localhost	E-mail do administrador
DocumentRoot "/var/www"	Diretório padrão das páginas WEB
UserDir public_html	Diretório para páginas pessoais de usuários
DirectoryIndex index.html	Arquivos iniciais das páginas
ErrorLog logs/error_log	Registro (log) de erros
CustomLog logs/access_log combined	Registro (log) de acessos
MinSpareServers 8	Número mínimo de servidores "esperando" clientes
MaxSpareServers 20	Número máximo de servidores "esperando" clientes
MaxClients 150	Limite de clientes simultâneos

Quadro 3 - Diretivas de configuração do Apache.

Não se preocupe, a princípio, em aprender o que cada uma significa. As duas diretivas mais importantes nesse instante são: `DocumentRoot` e `DirectoryIndex`. A primeira diz a partir de qual pasta do servidor os arquivos e pastas contidos na URL solicitada serão pesquisados. Desse modo, assumindo o valor padrão (`/var/www`), se a URL solicitada pelo navegador fosse [HTTP://start.ubuntu.com/9.10/index.html](http://start.ubuntu.com/9.10/index.html), o arquivo `index.html`, seria procurado na pasta `/var/www/9.10`.

A diretiva `DirectoryIndex` informa qual arquivo deve ser fornecido ao usuário caso ele não informe nenhum nome de arquivo. Portanto, se a URL solicitada pelo navegador fosse [HTTP://start.ubuntu.com/12.04/](http://start.ubuntu.com/12.04/), o servidor retornaria o conteúdo do arquivo `index.html` da pasta `/var/www/12.04`, porque `index.html` é o nome do arquivo padrão. Pelo mesmo raciocínio, se a URL solicitada pelo navegador fosse apenas [HTTP://start.ubuntu.com](http://start.ubuntu.com), o servidor retornaria o conteúdo do arquivo `index.html` da pasta `/var/www`.

Atividade 01

1. Onde deveria ficar o arquivo `index.php` do site <http://www.moodle.imd.ufrn.br/>?
2. Onde deveria ficar o arquivo `index.html` do usuário aluno no mesmo servidor?

Disponibilizando uma página web no Apache

Vamos agora ver como configurar o Apache para “hospedar” uma página web acessível pelo endereço `www.metropole1.edu.br`. Suponha que esse nome é do servidor que possui o endereço IP `10.1.1.2`.

O primeiro passo, assumindo que não pretendemos alterar as configurações do arquivo geral do servidor, consiste em criar um arquivo com o nome *metropole*, na pasta `/etc/apache2/sites-available/`. O conteúdo desse arquivo é mostrado na Figura 5. Os textos após o caractere `#` são comentários e poderiam ser suprimidos.

```
1 <VirtualHost *:80> # inicia a configuração do site
2   ServerAdmin webmaster@metropole1.edu.br # mail do administrador
3   DocumentRoot /var/www/metropole1 # Diretório para colocar os arquivos
4   ServerName www.metropole1.edu.br # Endereço do site
5   ErrorLog /var/log/apache2/metropole1-error.log # LOG de erro
6   CustomLog /var/log/apache2/metropole1-access.log combined # LOG de acesso
7 </VirtualHost> # finaliza a configuração do site
```

Figura 1 - Arquivo de configuração de um site web.

Vamos agora habilitar o site. Isso é feito com o comando a seguir. Mas observe que o site ainda não entra em execução. Isso só será feito quando reiniciarmos o Apache.

```
a2ensite metropole
```

O endereço <www.metropole1.edu.br> deve ser um nome válido, ou seja, existente na Internet. A maneira mais correta de tornar esse nome publicamente visível seria configurando o mesmo no serviço de DNS. Nesse nosso exemplo, para efeito de simplificação, tornaremos esse nome válido apenas na máquina onde estamos realizando as configurações. Para tanto, editamos o arquivo de nome `/etc/hosts` e inserimos a linha seguinte:

```
1 10.1.1.2 www.metropole1.edu.br metropole1
```

Figura 2 - Linha a ser inserida no arquivo hosts

Nesse ponto, reiniciamos o servidor Apache executando, como *root*, o comando:

```
/etc/init.d/apache2 restart
```

Para disponibilizar a página web, criamos um arquivo HTML simples e salvamos o mesmo com o nome `index.html` no diretório `/var/www/metropole1`. Após a cópia, podemos acessar a página com o navegador Firefox, digitando o seu endereço www.metropole1.edu.br.

Veja que como não informamos na URL o nome do arquivo, nem o nome de nenhuma pasta, será mostrado o arquivo `index.html` da pasta `/var/www/metropole1` do nosso servidor, pois esses são os valores padrão do arquivo de configuração do Apache.

O Apache já vem com um site padrão habilitado. Se quiser, pode usar o arquivo de configuração desse site que se chama `default` e está na pasta `/etc/apache2/sites-available`.

Atividade 02

1. Fazendo uma analogia ao comando `/etc/init.d/apache2 restart`, qual seria o comando para parar o servidor e qual seria o comando para iniciar o serviço?

Incluindo funcionalidades extras

Vamos agora incluir a possibilidade de utilização de páginas escritas na linguagem PHP no site previamente configurado. Para tanto, basta instalar o módulo do Apache que possibilita o suporte a essa linguagem, o que pode ser feito por meio do comando:

```
apt-get install php5 libapache2-mod-php5
```

Após a instalação do novo módulo, reinicie o servidor Apache executando, como *root*, o comando:

```
/etc/init.d/apache2 restart
```

```
1 <?php
2     Echo "Hello, World!";
3 ?>
```

Figura 3 - Arquivo teste.php de exemplo

No Firefox, se quiséssemos testar se o PHP está funcionando pelo Apache, podemos acessar o código PHP da Figura 7 através do endereço <http://www.metropole1.edu.br/teste.php>

Atividade 03

1. Se o parâmetro *DocumentRoot* do servidor web www.metropole1.edu.br possui o valor */dados/www2*, qual seria a pasta onde o arquivo *listagem.html* seria pesquisado quando um cliente tentasse acessar a URL <http://www.metropole1.edu.br/arquivos/listagem.html?>

Criando vários domínios no mesmo servidor web Apache

O Apache pode ser facilmente configurado para hospedar diversos sites distintos em um mesmo servidor.

Tomando como base a configuração já existente, bastaria alterar o arquivo */etc/apache2/sites-enable/metropole* e inserir as configurações referentes ao novo site.

A seguir vemos uma nova versão desse arquivo, estando agora o Apache configurado para hospedar os sites www.metropole1.edu.br e www.metropole2.edu.br.

```
1 <VirtualHost *:80> # inicia a configuração do site
2   ServerAdmin webmaster@metropole1.edu.br # mail do administrador
3   DocumentRoot /var/www/metropole1 # Diretório para colocar os arquivos
4   ServerName www.metropole1.edu.br # Endereço do site
5   ErrorLog /var/log/apache2/metropole1-error.log # LOG de erro
6   CustomLog /var/log/apache2/metropole1-access.log combined # LOG de acesso
7 </VirtualHost> # finaliza a configuração do site
8
9 <VirtualHost *:80> # inicia a configuração do site
10  ServerAdmin webmaster@metropole2.edu.br # mail do administrador
11  DocumentRoot /var/www/metropole2 # Diretório para colocar os arquivos
12  ServerName www.metropole2.edu.br # Endereço do site
13  ErrorLog /var/log/apache2/metropole2-error.log # LOG de erro
14  CustomLog /var/log/apache2/metropole2-access.log combined # LOG de acesso
15 </VirtualHost> # finaliza a configuração do site
```

Figura 4 - Arquivo de configuração do Apache para dois sites

Para testarmos as novas configurações, inserirmos uma nova linha no arquivo `/etc/hosts`, semelhante ao que foi configurado do exercício anterior, e reiniciamos o servidor Apache. A seguir, mostramos as duas linhas que foram inseridas.

```
1 10.1.1.2 www.metropole1.edu.br metropole1
2 10.1.1.2 www.metropole2.edu.br metropole2
```

Figura 5 - Linhas adicionadas no arquivo hosts para nosso exemplo

Lembre-se que em uma rede real você não iria utilizar o arquivo hosts para associar os endereços IP aos nomes. Isso seria feito no servidor de DNS.

Pronto! Se acessarmos <http://www.metropole1.edu.br> estaremos acessando as páginas na pasta `/var/www/metropole1`; e se acessarmos <http://www.metropole2.edu.br>, estaremos acessando as páginas na pasta `/var/www/metropole2`.

Também seria possível separar a configuração dos dois sites (metropole1 e metropole2) em dois arquivos separados. Nesse caso, utilizaríamos o comando `a2ensite` para cada um deles.

Veja aqui a explicação sobre apache.



Vídeo 03 - Apache

Arquivos de log

Uma coisa muito importante na maioria das empresas é analisar o log do servidor Web. Com as informações de log é possível gerar relatórios indicando, por exemplo, quais páginas do site foram acessadas, a que horário, e por quem.

O arquivo com as informações dos acessos se chama `access.log` e fica na pasta `/var/log/apache2`. Existe também um arquivo que mostra as requisições que geraram erro. Esse arquivo se chama `error.log` e também fica na pasta `/var/log/apache2`.

Essa análise pode ser feita de forma mais simples utilizando-se programas que leem os arquivos de log e geram relatórios.

Resumo

Nesta aula, aprendemos a instalar e configurar um servidor Apache. Vimos como configurar um único site no Apache. Depois vimos como ativar o suporte a PHP, como criar mais de um site na mesma máquina, e finalmente onde verificar os logs do Apache.

Referências

APACHE HTTP Server. Disponível em: <<http://httpd.apache.org/>>. Acesso em: 29 set. 2010.

KUROSE, J.; ROSS, K. **Redes de computadores e a internet**. 5. ed. São Paulo: Addison Wesley, 2010.

NETCRAFT. Disponível em: <<http://netcraft.com>>. Acesso em: 29 set. 2010.

RFC 1945. Disponível em: <<http://www.ietf.org/rfc/rfc1945.txt>>. Acesso em: 29 set. 2010.

RFC 2616. Disponível em: <<http://www.ietf.org/rfc/rfc2616.txt>>. Acesso em: 29 set. 2010.