

Administração de Redes de Computadores

Apache: servidor de HTTP/WWW

Alexssandro C. Antunes
(Alexssandro.Antunes@unisul.br)



Servidor Web

A internet hoje é a forma mais prática e rápida de se comunicar com o mundo. Grande parte das informações são transmitidas através de páginas na web.



O surgimento da World Wide Web

- Antes do surgimento da WWW, havia uma rede para fins militares; meados da década de 70;
- Começou em 1989 no CERN (Centro Europeu para pesquisa nuclear);
- Integrava pesquisadores de vários países;
- A Web nasceu da necessidade de troca de informações entre estes pesquisadores;



O surgimento da World Wide Web

(cont.)

- Tim Berners-Lee, propôs uma teia de documentos vinculados (março/1989);
- O primeiro protótipo (modo texto) já era operacional um ano e meio depois;
- A primeira amostra em público foi na Conferência Hypertext'91 (San Antonio - Texas);
- Lançamento da primeira interface gráfica (MOSAIC - fev/1993);



O surgimento da World Wide Web

(cont.)

- Marc Andersen, criador do MOSAIC, deixou a NCSA (National Center for Supercomputing Applications), para fundar sua própria companhia, a Netscape Communications Corporation; objetivo: desenvolver clientes, servidores e outros aplicativos para Web;
- A Web é basicamente um sistema Cliente-Servidor;



Tarefas de um servidor Web

- Estabelecer conexão com o cliente, através do protocolo HTTP;
- Fornecer documentos solicitados pelos clientes;
- Executar programas (C, scripts, perl, etc);
- Receber chamadas a programas e scripts e efetuar a passagem de parâmetros, recuperar os resultados e repassá-los;

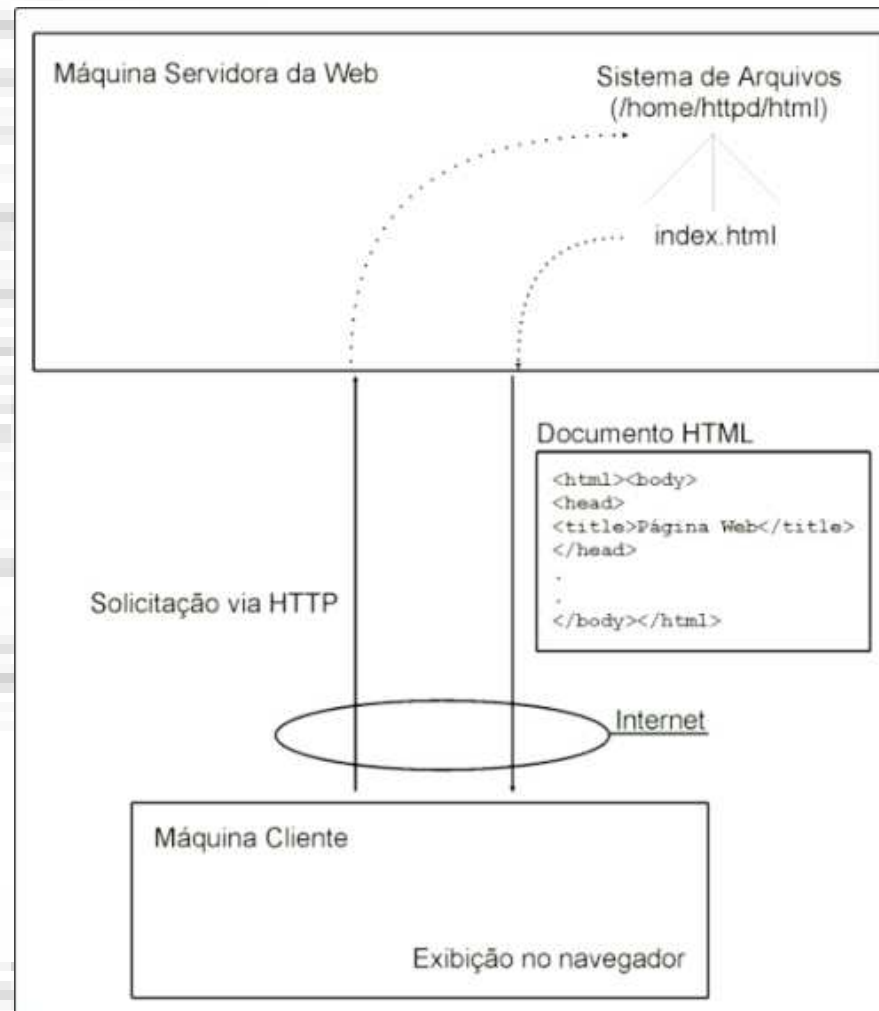


Tarefas de um servidor Web (cont)

- Controlar acesso a disco e arquivos;
- Mapear os endereços lógicos referenciados nas URLs;
- Atualizar as informações nos arquivos de log de acessos e erros;
- Aceitar e responder requisições dos métodos de solicitação de informações embutidos no protocolo HTTP (GET, PUT, POST E HEAD, DELETE, LINK, UNLINK);



Modelo de solicitação de HTTP



O conceito inicial (HTML estático)

O modelo de HTML estática possui várias vantagens, mas certos empecilhos (desvantagens) contribuíram para que outras tecnologias fizessem parte do crescimento da Web.

Algumas desvantagens do HTML estático:

- Difícil controle de layout e design;
- Não possui interatividade;
- Não suporta alterações rápidas de conteúdo ou personalização;



Um novo conceito Common Gateway Interface (CGI)

Devido a estas limitações, inúmeras tecnologias adicionais foram desenvolvidas. Talvez o avanço mais importante tenha sido o conceito de conteúdo dinâmico, ou seja, páginas Web criadas em resposta a solicitações de usuário



Outras tecnologias

- Javascript
- miniaplicativos Java (atuando no lado do cliente)
- XML - Extensible Markup Language
- ASP - Active Server Pages
- PHP - Personal Home Page
- JSP - Java Server Pages
- Cold Fusion

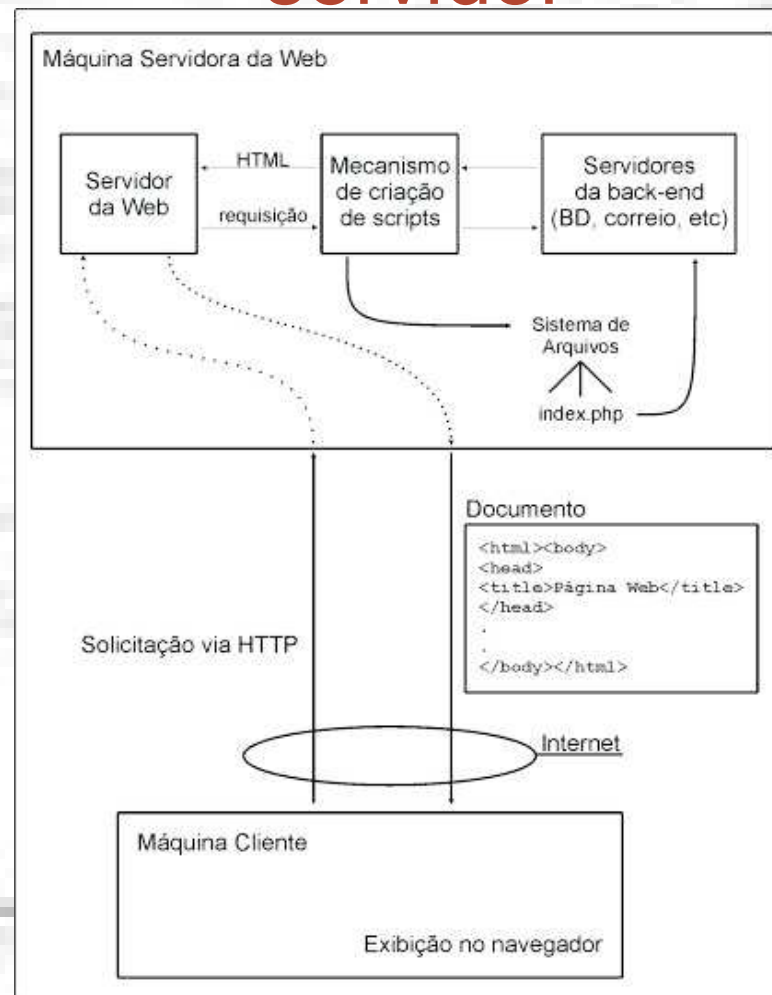


Um novo conceito Common Gateway Interface (CGI) (cont)

Ele basicamente define como um servidor Web executa programas localmente e os transmite ao browser do usuário que está requisitando o conteúdo dinâmico.



Representação esquemática da criação de scripts do fluxo de dados do lado servidor



HTTP (*HyperText Transfer Protocol*)

O protocolo HTTP (*Hypertext Transfer Protocol*) é um protocolo do nível de aplicação que possui objetividade e rapidez necessárias para suportar sistemas de informação distribuídos, cooperativos e de hipermídia.



HTTP (*HyperText Transfer Protocol*) (cont)

O HTTP ou *Hypertext Transfer Protocol* é o protocolo usado pelos browsers para se comunicar com os hosts, ou seja, “especular” por um determinado arquivo em uma determinada URL, ou enviar/postar dados para o servidor.



Principais características do HTTP

- Permite busca de informação e atualização (update)
- Comunicação entre os agentes usuários e gateways, permitindo acesso à hipermídia e a diversos protocolos do mundo Internet, tais como, SMTP, NNTP, FTP, Gopher, WAIS.



Principais características do HTTP

(cont)

- Pode ser implementado em cima de qualquer protocolo Internet.
- As mensagens são enviadas em um formato semelhante aos utilizados pelo correio eletrônico da Internet e pelo MIME (Multipurpose Internet Mail Extensions).



Principais características do HTTP

(cont)

- Obedece ao paradigma de pedido/resposta: um cliente estabelece uma conexão com um servidor e envia um pedido a este, o qual o analisa e responde. A conexão deve ser estabelecida antes de cada pedido de cliente e encerrada após a resposta.



Principais métodos do HTTP

GET : O método GET é o método mais comum e o mais utilizado. É através do método GET que os browsers solicitam as páginas ou documentos. Neste tipo de método, você será o cliente (browser), o qual fará uma requisição a um servidor Web(host).

POST

HEAD



Simulação de um browser através de um cliente para o protocolo TELNET de login remoto

```
bash-2.05b# telnet 192.168.7.9 80
```

```
Trying 192.168.7.9...
```

```
Connected to 192.168.7.9.
```

```
Escape character is '^['.
```

```
GET /index.html
```

```
....
```

```
....
```



Simulação de um browser através de um cliente para o protocolo TELNET de login remoto (cont)

Os servidores Web possuem processos servidores que escutam a porta 80 (TCP), os quais ficam aguardando por conexões clientes (normalmente browsers).

Após o estabelecimento da conexão, o cliente envia uma solicitação e o servidor envia uma resposta. A conexão é então liberada.



HTTPS

Um avanço importante do protocolo HTTP foi a introdução do HTTP seguro (HTTPS). Este protocolo permite comunicação segura entre o browser e o servidor.

Basicamente, isto significa que é seguro para usuário e servidor transmitir dados importantes de um para outro através do que pode ser considerada uma rede insegura.



Apache (instalação)

Instalando o Apache:

```
[root@localhost http]# rpm -ivh httpd-2.0.40-8.i386.rpm
```

```
Preparing packages for installation...
```

```
httpd-2.0.40-8
```

Site:

www.apache.org



Layout dos Arquivos do Servidor Web Apache

/etc/httpd/

drwxr-xr-x	2	root	root	4096	Jul 22 15:45	conf
drwxr-xr-x	2	root	root	4096	Jul 22 15:45	conf.d
lrwxrwxrwx	1	root	root	19	Jul 22 15:45	logs -> ../../var/log/httpd
lrwxrwxrwx	1	root	root	27	Jul 22 15:45	modules -> ../../usr/lib/httpd/modules
lrwxrwxrwx	1	root	root	13	Jul 22 15:45	run -> ../../var/run

/var/www/

drwxr-xr-x	7	root	root	4096	Jul 22 17:24	.
drwxr-xr-x	19	root	root	4096	Jul 22 15:45	..
drwxr-xr-x	2	root	root	4096	Sep 4 2002	cgi-bin
drwxr-xr-x	3	root	root	4096	Jul 22 15:45	error
drwxr-xr-x	2	root	root	4096	Jul 22 16:29	html
drwxr-xr-x	3	root	root	4096	Jul 22 15:45	icons
drwxr-xr-x	13	root	root	4096	Jul 22 17:24	manual



Layout dos Arquivos do Servidor Web Apache

Estrutura de diretórios

- /etc/httpd/conf - Contém todos os arquivos de configuração do apache.
- /var/www/ - Contém os ícones do servidor, programas CGI, arquivos HTML.
- /var/log/httpd - Contém os arquivos de log do servidor.



Configurando o Apache

O arquivo `/etc/httpd/conf/httpd.conf`:

O arquivo `httpd.conf` é um arquivo extenso, legível e muito bem comentado. O Apache apresenta ao administrador exatamente o que pode estar errado quando se tenta iniciar o serviço.



Configurando o Apache (cont.)

Após a instalação, o Apache já pode ser inicializado e está apto a servir requisições de forma adequada.

Entretanto, por questões de segurança, há algumas diretivas que precisam ser ajustadas. Procure no arquivo de configuração (</etc/httpd/conf/httpd.conf>) e altere-as.



Diretivas do httpd.conf

Segue abaixo algumas destas diretivas:

- **Servername**
- **DocumentRoot**
- **UserDir**
- **DirectoryIndex**
- **AccessFileName**
- **ScriptAlias**



Parâmetros de Configuração httpd.conf

- **Servertype** - Modo de Execução do Apache. inetd ou standalone. standalone aconselhável.
- **ServerRoot** - Diretório contendo arquivos de configuração, log e erros.
- **Port** - A porta que o servidor "escutará".
- **User** - O nome e/ou número do usuário com o qual o servidor httpd rodará.
- **Group** - O nome e/ou número do grupo com o qual o servidor httpd rodará.



Parâmetros de Configuração httpd.conf

- **ServerAdmin** - E-mail do administrador do site.
- **ServerName** – Nome do servidor web.
- **DocumentRoot** - Diretório que contém as paginas web do servidor.
- **UserDir** - Nome do diretório que cada usuário deve possuir para disponibilizar paginas web.
- **DirectoryIndex** - Página inicial de um diretório web.
- **StartServers** - Número de servidores executados inicialmente.



Parâmetros de Configuração httpd.conf

- **Alias** – Permite mapear um caminho para qualquer localização no sistema de arquivos.
- **Redirect** – Redireciona uma URL requisitada para outra URL qualquer.
- **ErrorLog** - Arquivo padrão de erros.
- **TransferLog** – Indica o nome do arquivo de log para onde as informações de log serão enviadas.
- **FancyIndexing** – Cria uma página web para um diretório que não contenha um arquivo index.html contendo a listagem do diretório.



Parâmetros de Configuração httpd.conf

- **AccessFileName** – Especifica o nome do arquivo de controle de acesso por diretório.
- **ErrorDocument** – Permite especificar um arquivo ou URL para um erro gerado pelo servidor.
- **<Directory></Directory>** - Permite especificar diretivas que se aplicam somente ao diretório especificado.
- **ScriptAlias** – Controla que diretórios possuem scripts CGI.



Parâmetros de Configuração Hosts Virtuais

- **Tipos**

- Baseado em endereços - Única máquina com IP'S diferente.
- Baseado em nomes - Único IP para vários hosts virtuais

- **<VirtualHost>...</VirtualHost>**

- **NameVirtualHost <IP>**

- .



Parâmetros de Configuração

Hosts Virtuais x DNS

- **Exemplo de Configuração no Servidor Web Apache ([httpd.conf](http://httpd.apache.org/docs/2.4/conf.html))**
 - `ServerName ns.redes.com.br:80`
 - `NameVirtualHost *:80`
 - `<VirtualHost *:80>`
 - `DocumentRoot /var/www/redes`
 - `ServerName www.redes.com.br`
 - `</VirtualHost>`
 - `<VirtualHost *:80>`
 - `DocumentRoot /var/www/admredes`
 - `ServerName www.adm.redes.com.br`
 - `</VirtualHost>`



Parâmetros de Configuração

Restrições de Acesso

- **AllowOverride** – Permite especificar que diretivas declaradas no arquivo de controle de acesso podem se sobrepor a diretivas encontradas anteriormente no arquivo de configuração.
 - All – Permite o uso de todas as diretivas.
 - AuthConfig – Permite o uso de diretivas de autenticação.
 - Limit – Permite o uso de diretivas de controle de acesso por host.
 - .



Parâmetros de Configuração

Restrições de Acesso

- **allow** – Permite definir uma lista de hosts que terão acesso liberado a um determinado diretório.
- **deny** – Oposto de allow
- **order** – Permite especificar a ordem em que as diretivas de controle de acesso (allow/deny) serem lidas.
- **require** – Permite especificar que usuários ou grupos podem acessar um diretório.
- **AuthName** – Rótulo da janela de autenticação.
- **AuthType** – Tipo de autenticação.



Parâmetros de Configuração

Restrições de Acesso

- **AuthUserFile** – Nome do arquivo contendo o nome e a senha de usuários que podem acessar o diretório.
- **AuthGroupFile** - Nome do arquivo contendo os nomes dos grupos que podem acessar o diretório.



Exemplo de Configuração Servidor Web - Autenticação

- No arquivo `httpd.conf` (`AccessFileName .htaccess`):

```
DocumentRoot "/var/www/html"
```

```
<Directory "/home/seguro/public_html">
```

```
AllowOverride AuthConfig
```

```
</Directory>
```

- Conteúdo do arquivo `.htaccess` criado no diretório (`/home/seguro/public_html`):

- `AuthType Basic`
- `AuthName "Acesso ao diretório home do user seguro"`
- `AuthUserFile /etc/httpd/conf/seguro/users` e/ou `AuthUserFile /etc/httpd/conf/seguro/httpd.passwd`
- `AuthGroupFile /etc/httpd/conf/seguro/groups`
- `Require group adm` e/ou `Require user seguro`



Exemplo de Configuração Servidor Web - Autenticação

- **Para criar um novo arquivo:**

```
htpasswd -c <nome_arquivo> <usuário>
```

Ex: # **htpasswd -c httpd.passwd seguro**

- **Exemplo de arquivo de users:**

```
fulano:15M3ueyCXI3ZI
```

```
wesley:..QfIx/rhzenhA
```

- **O arquivo de grupos:**

```
adm:wesley
```

```
all:wesley fulano
```

```
staff:fulano
```



Apache (inicializando o daemon)

Inicializando o Servidor Web Apache:

O **apachectl** é um *front end* para o servidor Apache HyperText Transfer Protocol (HTTP). Essa ferramenta foi projetada para ajudar o administrador a controlar a funcionalidade do *daemon* httpd.

```
/etc/rc.d/init.d/httpd start
```

```
/etc/rc.d/init.d/httpd status
```

```
/etc/rc.d/init.d/httpd stop
```



Apache (Verificando o *status* dos processos)

Para verificar os status do processo responsável pelo servidor Web, utiliza-se o comando *ps*.

```
[root@localhost http]# ps -ax | grep httpd
1313 ?    S    0:00 /usr/sbin/httpd -k start
1314 ?    S    0:00 /usr/sbin/httpd -k start
1315 ?    S    0:00 /usr/sbin/httpd -k start
1316 ?    S    0:00 /usr/sbin/httpd -k start
1317 ?    S    0:00 /usr/sbin/httpd -k start
1318 ?    S    0:00 /usr/sbin/httpd -k start
1319 ?    S    0:00 /usr/sbin/httpd -k start
1320 ?    S    0:00 /usr/sbin/httpd -k start
1321 ?    S    0:00 /usr/sbin/httpd -k start
1325 pts/1 S    0:00 grep httpd
```



Fim

