

Análise de tráfego HTTP

1. O objetivo deste trabalho é sensibilizar o estudante para a utilização de ferramentas baseadas na WEB para a análise da utilização de recursos de servidores e *proxies* HTTP.
2. Propõe-se a avaliação de duas ferramentas de uso gratuito: o “Webalizer” e o “AWStats”. Ambas deverão permitir realizar a análise estatística e a produção de relatórios de utilização do serviço HTTP, a partir dos ficheiros de log do servidor.
3. Para objeto de análise sugere-se a utilização do “Apache” e do “Squid” como servidor HTTP e proxy HTTP, respetivamente.

Apache

Configure um servidor Apache para disponibilizar dois sites Web com atualização dinâmica de conteúdos. Eventualmente, se tiver esse conhecimento prévio, pode criar dois “*virtual hosts*” (VH) para simular acessos a sites diferentes (veja o código de exemplo em anexo), contendo um site a lista dos processos e carga do sistema, e o outro site a lista de sessões TCP/IP estabelecidas e a tabela de ARP.

Os ficheiros de log do Apache deverão estar em `/var/log/httpd`, devendo ser feito o *logging* separado em diferentes ficheiros de log, no caso de termos mais do que um VH.

Proxy SQUID

Configure o servidor proxy SQUID num dos servidores que tem disponíveis. Usando um cliente HTTP configure o acesso periódico via o proxy ao(s) site(s) que instalou no Apache. Verifique que nos logs do SQUID estão registados os acessos efetuados.

Os ficheiros de log do Squid deverão estar em `/var/log/squid`.

Para simplificar, é aconselhado alterar a opção “`logformat`” no ficheiro de configuração do SQUID, para gerar registos de log no formato semelhante ao servidor Apache.

Alternativamente, pode-se manter o formato nativo dos registos de log do SQUID e usar o programa `squid2common.pl` para converter os logs do proxy SQUID em dados no formato do servidor do Apache.

Quando se executa o programa `squid2common.pl` são criados dois ficheiros: `cache.convert` e `proxy.convert`. O primeiro contém os logs de acesso à cache e o segundo os logs de acesso ao proxy.

Webalizer

Instale a ferramenta Webalizer. As configurações do Webalizer estão num ficheiro de configuração único chamado `/etc/webalizer.conf`. Pode também colocar um ficheiro de

configuração em cada directório onde executa o programa webalizer e dessa forma analisar diferentes ficheiros de log: cada site com o seu ficheiro de log.

AWStats

Instale a ferramenta AWStats. As configurações do AWStats estão num ficheiro de configuração residente no directório em `/etc/awstats/`, podendo também acrescentar um ficheiro de configuração, no mesmo directório, para cada servidor que pretenda analisar.

Resultados

Apresente a análise comparativa detalhada das duas ferramentas, em termos funcionais e, se possível, também quanto ao desempenho.

Existem outras ferramentas para este tipo de análise dos ficheiros de log, tal como o “W3Perl”, pelo que deverá ser feita a análise comparativa desta ferramenta com as anteriormente avaliadas.

Crontab

Estas ferramentas de análise de logs devem correr periodicamente, pelo menos uma vez por dia, de forma a recolher os dados estatísticos e apresentá-los de uma forma amigável.

Para as executar de forma periódica pode-se usar o `crond`, que lê o ficheiro de configuração `/etc/crontab` onde se programam as horas a que devem correr.

Estas deverão ser executadas sempre antes da rotação de logs, que também é programada no `/etc/crontab` ou usando a ferramenta `logrotate`.

Consulta recomendada:

- <http://httpd.apache.org>
- <http://www.squid-cache.org>
- <http://www.webalizer.org>
- <http://www.awstats.org>
- <http://www.w3perl.com>

Elabore um pequeno relatório com as respostas às perguntas acima e envie-o por e-mail até ao dia anterior da próxima aula prática, para [<joao.neves@fe.up.pt>](mailto:joao.neves@fe.up.pt).

Anexos

Exemplo de código para o *Site 1*:

```
<html>
<meta http-equiv="refresh" content="<?php
    $hora = date("H");
    if ($hora > 9 && $hora < 12)
        echo rand(1, 5);
    elseif ($hora >= 12 && $hora < 14)
        echo rand(20, 240);
    elseif ($hora >= 14 && $hora < 18)
        echo rand(1, 4);
    elseif ($hora >= 18 && $hora < 20)
        echo rand(60, 480);
    else
        echo rand(2000, 4000);
?>">

<body>
<?php
    $f = fopen("/proc/net/netstat", 'r');
    while (!feof($f)) {
        echo "<pre>" . fgets($f) . "</pre>";
    }
    fclose($f);
?>
</body>
</html>
```

Exemplo de código para o *Site 2*:

```
<html>
<meta http-equiv="refresh" content="<?php
    $hora=date("H");
    if($hora>8 && $hora<=18) {
        echo rand(1,60);
    } else if($hora>18 && $hora<=23) {
        echo rand(60,120);
    } else if($hora>23 && $hora<=8) {
        echo rand(500,900);
    }
?>">

<body>
<?php
    $output = shell_exec('netstat -n');
    echo "<pre>".$output."</pre>";
    $output = shell_exec('cat /proc/net/arp');
    echo "<pre>".$output."</pre>";
?>
</body>
</html>
```