Programación Paralela
Año 2014/2015
Grado en Matemáticas
Facultad de CC. Matemáticas

Práctica 2

Análisis de ficheros log de acceso a páginas web

12 de mayo de 2015

Toda la actividad que realizan los usuario en un servidor web queda reflejada en sus ficheros *log*. Estos ficheros guardan información textual como la que aparece en la figura siguiente:

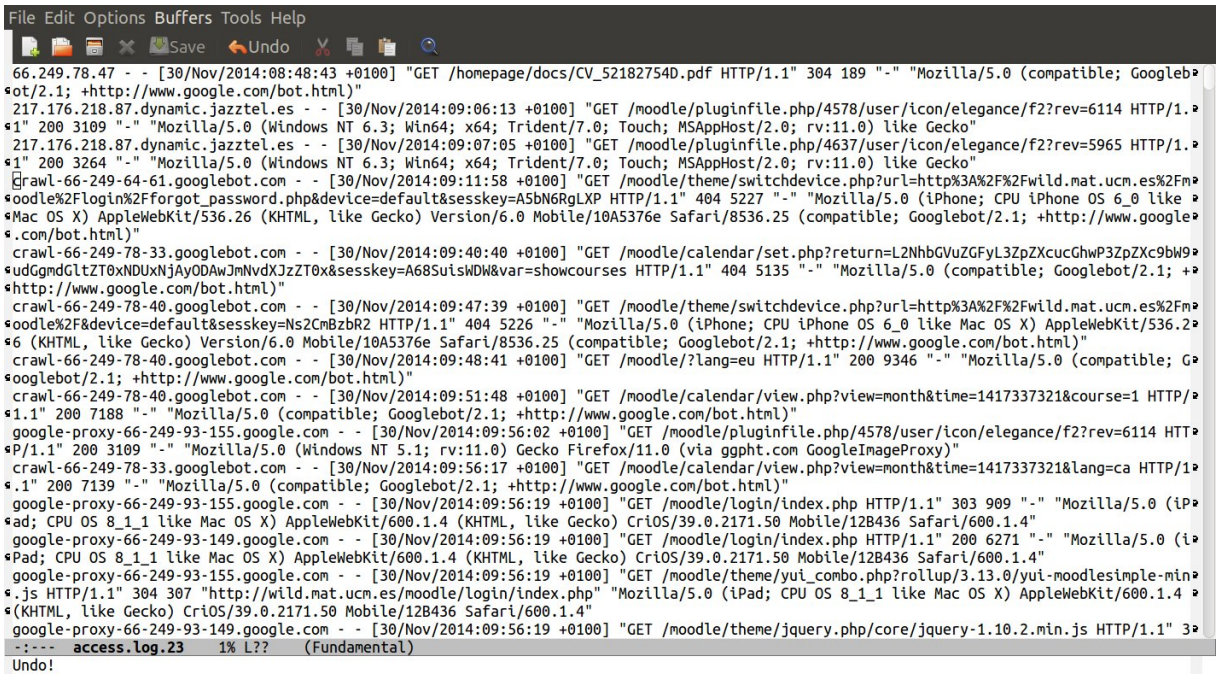

Figura 1: Fragmento de un fichero log.

Mientras que un libro suele tener entre 4000 y 6000 líneas, en un solo día, un servidor apache de un sitio 'modesto' como es `wild.mat.ucm.es`, puede general decenas de miles de líneas. De forma simplificada, podemos decir que por cada petición que hace un usuario del servidor, se registra una línea que contiene toda la información relacionada con la petición.

Los ficheros log tienen un formato muy concreto y son muy fácilmente legibles (aunque mirando la imagen anterior cueste creerlo!). Miremos una única línea:

```
147.96.18.203 - - [27/Apr/2015:10:15:32 +0200] "GET /moodle/ HTTP/1.1"200
8944 Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:34.0) Gecko/20100101 Firefox/34.0"
```

La información se estructura por columnas separadas por espacios en blanco. Vamos a comentar algunas columnas que serán de nuestro interés, para una información más

detallada puede consultarse la documentación oficial (`http://httpd.apache.org/docs/1.3/logs.html#accesslog`).

- En la primera columna aparece la dirección ip del cliente, lo que llamaremos usuarios. En el ejemplo anterior `147.96.18.203`.
- En la cuarta columna aparece la fecha y la hora en la que la petición de página se ha realizado, `[27/Apr/2015:10:15:32 +0200]`.
- En la quinta columna aparece la petición de página concreta que el usuario ha hecho, `"GET /moodle/ HTTP/1.1"`. De esta petición lo que más nos interesa es la página pedida, en este caso `/moodle/`.
- En la sexta columna aparece un código de respuesta, que indica si se ha podido responder a la petición del usuario, si la página no existe, etc. (véase, por ejemplo, `http://es.wikipedia.org/wiki/Anexo:C%C3%B3digos_de_estado_HTTP`). En el ejemplo anterior el código de respuesta es 200, que indica que la página solicitada ha sido enviada sin ningún problema.

El análisis de log es una actividad que interesa a muy diversos niveles: identificar las páginas más visitadas, la carga del servidor en las distintas horas del día, los navegadores utilizados por los usuarios... Vamos a hacerte algunas propuestas concretas:

1 Sesiones

Decimos que una *sesión de usuario* es la actividad que ha realizado de forma continuada. Diremos que dos peticiones de un usuario están en diferentes sesiones si difieren en más de un determinado tiempo T. Este tiempo suele estar entre media hora o una hora, pero queremos que sea un parámetro de nuestra solución para poder cambiarlo en cualquier momento.

Escribe un programa map-reduce que, a partir de unos ficheros log y un tiempo de cierre de sesión T, devuelva el número total de sesiones de cada usuario. Utiliza librerías externas que te permitan manejar con facilidad las fechas que hay en el log.

2 Comportamiento de los usuarios

Supongamos que estamos interesados en analizar el *comportamiento* de los usuarios en cada sesión. Para cada usuario, nos gustaría saber qué grupos de páginas visita, con que frecuencia se conecta, si tiene un patrón definido de navegación, si este patrón es parecido al de otros usuarios... Realmente no hace falta mucha imaginación para pensar posibles aplicaciones de un análisis de estos resultados.

El comportamiento de un usuario puede definirse con muy diversos niveles de detalle. Nosotros vamos a considerar simplemente que el comportamiento durante una sesión es el conjunto de páginas web que solicita.

Escribe un programa map-reduce que, a partir de unos ficheros log, devuelva los comportamientos de cada usuario (el conjunto de páginas visitadas) y las veces que cada comportamiento se ha repetido.

3 Comportamientos parecidos entre usuarios

Puede ser interesante saber qué usuarios tienen comportamientos parecidos.

Escribe un programa map-reduce que, a partir de unos ficheros log, devuelva para cada comportamiento detectado, la lista de usuarios que han seguido ese comportamiento en una sesión.

4 Integración

Por último, queremos integrar los datos que hemos calculado de forma independiente en los apartados anteriores.

Escribe un programa map-reduce que, a partir de unos fichreros log, devuelva las tuplas:

```
user, t_s, behaviour, n_b, user_list
```

- `user` es la ip de un usuario
- `t_s` es el número total de sesiones de dicho usuario
- `behaviour` es uno de los posibles comportamientos del usuario (conjunto de páginas)
- `n_b` es el número de veces que el usuario `user` repite el comportamiento `behaviour`
- `user_list` es la lista de usuarios que también han seguido en alguna ocasión el comportamiento `behaviour`.