

A. Contextualización

Después del duro trabajo y la frenética carrera por entregar a tiempo el analizador de logs, la empresa contratante se niega a pagarnos, a menos que le demos plenas garantías de que funciona correctamente, y que es más rápido y barato que el programa de la competencia (<http://www.mrunix.net/webalizer/>) que es gratis y tiene gráficos en color¹. Por lo tanto, deberemos comprobar que no se pierden peticiones en las consultas, y medir la eficiencia de los tiempos de consulta, de carga y el uso de memoria.

En definitiva, se nos pide un **informe técnico** del programa desarrollado en el que se analice en profundidad el tiempo y la memoria consumidos, y su variación en función de los factores más relevantes, como el número de peticiones, y el número de clientes, ficheros y días distintos. Para la elaboración del informe será necesario hacer un estudio experimental exhaustivo y riguroso, complementado con un estudio teórico adecuado.

B. Enunciado del problema

El informe técnico se compone de dos partes: (1) verificación y pruebas, y (2) análisis de eficiencia. El objetivo de la primera es garantizar que el analizador de logs obtiene resultados correctos en todos los casos. La segunda tiene un mayor peso, y su propósito es medir el consumo de recursos de tiempo y memoria.

B.1. Verificación y pruebas

Para comprobar el correcto funcionamiento de un programa no basta con ejecutarlo y ver que no se cuelga. Es necesario diseñar un conjunto suficientemente amplio y variado de casos de prueba, establecer de antemano los resultados que debe producir una ejecución correcta de los mismos, ejecutar el programa con los casos de prueba y contrastar los resultados obtenidos con las salidas esperadas.

- **Diseño de casos de prueba.** Los casos de prueba diseñados deben tener tamaños y condiciones de uso más o menos *realistas*, dentro de nuestras posibilidades. A modo orientativo, el programa debería poder cargar sin problema todos los ficheros de ejemplo disponibles (el máximo incluido es de 16 Mbytes). Los casos deben presentar variabilidad en diferentes aspectos: distintos tipos de operaciones (selecciones, listados, agrupaciones), distintos tipos de criterios en las consultas, carga incremental, distintos tamaños de las consultas (en cuanto al número de resultados esperados), etc.

Cada caso de prueba consta de dos elementos: (a) el fichero o conjunto de ficheros de logs utilizados, y (b) el **fichero de entrada al analizador**. El fichero de entrada al analizador contiene todas las instrucciones a ser ejecutadas por el programa, empezando por la carga de un fichero de log y siguiendo con la realización de consultas. Por ejemplo, podría contener lo siguiente:

```
load access_log.1
select cliente 155.54.12.200
selectand fichero /contest/
select codigo 404
group dia
quit
```

¹ Ver otros analizadores en: http://tu cows.arrakis.es/log95_default.html
y en: http://arrakis.linux.tu cows.com/log_analyzers_default.html

Los ficheros de log usados pueden ser los entregados a los alumnos como ejemplo, u otros creados mediante un programa que genere ficheros de logs “artificiales”. En este último caso, el programa puede controlar el tipo y contenido de las peticiones generadas, para conocer de antemano los resultados de las consultas.

- **Resultados esperados de los casos de prueba.** Los resultados esperados hacen referencia al número de peticiones que, *teóricamente*, deberían ser obtenidas como respuesta de las consultas (comandos *select*, *selectand*, *selector* y *group*), y las peticiones en sí de la selección actual (comandos *list* y *listall*).

Se deja libertad para decidir cómo encontrar las respuestas teóricas de cada caso de prueba. Por ejemplo, se puede hacer manualmente, usando comandos de Linux que resuelvan el problema (como `grep`), o controlando de antemano los contenidos de los ficheros de log. En cualquier caso, es necesario que los resultados se establezcan **antes** de usar el analizador implementado.

- **Ejecución y contraste de resultados.** Una vez disponemos de los casos de prueba y de los resultados teóricos esperados, el siguiente paso es ejecutar el programa con los casos definidos y comprobar que obtiene los valores correctos. Cada caso de prueba se debe ejecutar haciendo uso de una redirección de la entrada (usando el fichero de entrada al analizador) y enviando el resultado a otro fichero para ser comprobado. Por ejemplo:

```
>> ./analizador <entrada1.txt >salida1.txt
```

La verificación de los resultados se puede realizar manualmente (contrastando a ojo los valores esperados y los obtenidos) o implementando algún programa que automatice el proceso. Se valorará el grado de automatización conseguido.

En resumen, la labor a realizar por los alumnos consiste en establecer y documentar el mecanismo de verificación (es decir, determinar los aspectos no definidos arriba), aplicarlo al analizador de logs y obtener las conclusiones oportunas. En caso de encontrar errores no justificables, deben ser corregidos y documentados.

Ojo: conviene ser sincero en el proceso de verificación, ya que *la empresa* usará sus propios casos de prueba, también con sus resultados teóricos esperados.

B.2. Análisis de eficiencia

En el análisis de eficiencia se seguirá una filosofía similar a la usada en la fase de verificación, en cuanto a la creación y uso de casos de prueba (con ficheros de logs y ficheros de entrada al analizador). Pero ahora el número de peticiones resultantes es irrelevante, y lo que nos interesa son los **tiempos de consulta**, el **tiempo de carga** y el tamaño de la **memoria ocupada**.

Se debe estudiar la variación de esos **recursos** en función de los siguientes **factores**:

- Número de peticiones existentes (lo llamaremos **p**).
- Número de clientes distintos (**c**).
- Número de ficheros distintos (**f**).
- Número de días distintos (**d**), es decir, el periodo de tiempo al que corresponden las peticiones.
- Opcionalmente, en función de cualquier otro factor que los alumnos consideren relevante en su programa.

De manera simplificada, el estudio consistirá en analizar cómo varían los recursos cambiando un factor y manteniendo fijos los demás. Y así para cada factor de interés. En concreto, el análisis constará de las siguientes fases:

- **Diseño del proceso.** Determinar los objetivos a analizar: recursos a considerar, factores relevantes en cada recurso, intervalos de análisis para cada factor y recurso.

Después, especificar los tamaños concretos y el número de pruebas por tamaño, para conseguir un análisis fiable y preciso. Se entiende que cuanto mayor sea el número de pruebas más completo y fiable será el estudio. Establecer las condiciones en las que se deben aplicar los casos de prueba.

- **Estudio teórico.** Estudiar de forma teórica las funciones que indican la variación de los recursos en función de los diferentes factores, siempre que sea posible. Ojo, no se trata de hacer un estudio del código línea por línea, sino de forma más global y abstracta, teniendo en cuenta las estructuras de datos y los algoritmos que intervienen en cada operación. El estudio teórico debe conducir a la obtención de los órdenes de complejidad de los recursos en función de los factores considerados. Usar las notaciones asintóticas adecuadas (O , Ω , Θ , o-pequeña). Estos órdenes se usarán en el contraste teórico/experimental.
- **Creación de los casos de prueba.** Definir un método adecuado para crear los diferentes casos de prueba, con los tamaños y variaciones establecidos en el primer punto. Asimismo, se deberá definir un método para medir los resultados obtenidos. Se recomienda muy encarecidamente implementar un programa que automatice el proceso de extracción de tiempos y uso de memoria. El mismo programa podrá calcular valores promedios y escribir los resultados en otro fichero.
- **Análisis estadístico y contraste teórico/experimental.** Utilizando los amplios conocimientos del alumno en estadística, se analizarán los resultados obtenidos. Sugerimos las siguientes técnicas: representaciones gráficas (que muestren la información relevante), ajuste de regresión (usando las funciones teóricas, para ajustar una muestra de puntos a cada función con forma conocida), bondad de ajuste y tests de aleatoriedad (para verificar la bondad de los ajustes estadísticos realizados). Para ello, se usarán las herramientas estadísticas/matemáticas adecuadas que dominen los alumnos. Además del análisis, se deben mostrar tabularmente los valores obtenidos y usados (es decir, los datos *en crudo*).
- **Comparación y conclusiones.** Decir que un programa es más o menos eficiente requiere comparar distintas soluciones a un mismo problema. En este punto se trata de hacer una comparativa entre el analizador implementado y algunos de los programas de la competencia (el indicado en el apartado A u otro similar). Obviamente la comparativa será imprecisa, puesto que las medidas de tiempo y memoria no son exactas, y los programas no hacen lo mismo. No obstante, dentro de esto, se intentará hacer la comparación en las condiciones lo más objetivas posible. Se pide, al menos, seleccionar un programa analizador de logs, y comparar los tiempos de carga y la memoria consumida para algunos ejemplos de entrada. Por último, obtener las conclusiones más relevantes del estudio realizado, en cuanto a: eficiencia obtenida, puntos fuertes y débiles del analizador, qué versión funciona mejor y en qué condiciones, fiabilidad del estudio, resultados del estudio de eficiencia y del contraste teórico/experimental, etc.

Se deben documentar las decisiones tomadas en cada una de las diferentes fases, los programas creados y los resultados obtenidos. En cualquier caso, se debe intentar que la carga de trabajo de la práctica se ajuste a la racionalidad de créditos de la asignatura.

C. Memoria de la práctica

La memoria de la práctica deberá contener obligatoriamente los siguientes apartados.

C.1. Portada

Nombre de los alumnos, número de prácticas y **e-mail** de cada uno. Dentro de la memoria deberá aparecer el login y el password de la cuenta en la que están los ficheros de la práctica (incluida la memoria de prácticas, en formato DOC, PS o PDF).

C.2. Verificación y pruebas

Verificación de que el analizador de logs implementado funciona correctamente:

- Diseño de casos de prueba. Cuestiones a considerar, tipos de comandos, criterios y casos, condiciones de ejecución y mecanismo de creación de los casos.
- Resultados esperados de los casos de prueba. Mecanismo de obtención de los resultados teóricos esperados.
- Ejecución y contraste de resultados. Mecanismo de comparación entre los resultados teóricos y los valores obtenidos.
- Conclusiones. Determinar la corrección o no del analizador. En caso contrario hacer las modificaciones oportunas.

En cualquier caso, no se deberán incluir en la memoria documentos que ocupen un gran tamaño y no sean relevantes o hayan sido generados de manera más o menos automática. En su lugar, se deberán hacer las referencias oportunas al directorio de prácticas donde están disponibles.

C.3. Proceso de análisis de eficiencia

Establecimiento del método para la realización del análisis, incluye:

- Determinación de los objetivos del análisis: recursos a medir, factores relevantes a considerar, e intervalos de análisis para cada factor y recurso.
- Análisis teórico de la eficiencia. Análisis teórico de eficiencia de los algoritmos y estructuras de datos usados en el programa. Obtener los órdenes de complejidad teóricos para cada recurso, en función de los factores.
- Diseño de casos de prueba. Describir los casos de prueba que se van a aplicar: número de casos, tamaños a utilizar en cada caso y número y tipo de comandos por caso. Indicar claramente las condiciones para la ejecución de los casos (esto es, los factores externos: procesador, memoria, SO, etc.).
- Diseño del método de ejecución de las pruebas. Ejecución manual o automática. En el segundo caso, describir los programas diseñados para hacer las pruebas. Detallar las herramientas matemáticas y estadísticas usadas.

C.4. Listado del código

En caso de haber creado algún programa para la ejecución automática o semiautomática de las pruebas, tanto del apartado C.2 como del C.3, incluir aquí el listado.

C.5. Resultados y estudio estadístico

Valores resultantes de la ejecución de los casos del apartado C.3 y análisis de los mismos. Debe contener los siguientes puntos:

- Tablas de resultados obtenidos. Tiempos y memoria usada para cada caso de prueba, así como otra información relevante en caso necesario.
- Representación gráfica de los resultados. Deben haber gráficos representativos para cada una de las condiciones o parámetros de tamaño relevantes.
- Ajuste estadístico. Ajuste de regresión de los resultados obtenidos a las funciones teóricas previstas. Estimación de la bondad del ajuste realizado (mediante el error de ajuste o tests de aleatoriedad). Se pide que haya como mínimo, por lo menos, un ajuste de regresión.

- Comparación de resultados. Seleccionar un analizador de logs *externo*. Documentar las pruebas comparativas realizadas. Comparar los resultados obtenidos con el programa propio y con el otro analizador.
- Conclusiones. Conclusiones más relevantes del análisis.

C.5. Informe de desarrollo

Debe contener lo siguiente:

- Organización temporal de las fases de resolución para esta práctica en concreto.
- Cómo ha sido la coordinación y el reparto del trabajo entre los miembros del grupo. Concretar personas y tareas realizadas.
- Tablas de dedicación personal en las distintas fases del trabajo. Se utilizarán tablas como las explicadas en las páginas 37 y 350 del texto guía, rellenas con el mayor rigor posible.
- Conclusiones y valoraciones personales.

D. Evaluación de la práctica

D.1. Obligatorio

Para aprobar la práctica se requiere que:

- Los programas que sean desarrollados se puedan compilar sin errores en las máquinas del laboratorio de prácticas, en la fecha y hora en la que se realice la entrevista con los alumnos, y deben funcionar correctamente.
- La memoria de la práctica debe contener todos los puntos indicados en el apartado C. La memoria debe ser entregada en el plazo que se establezca.
- Todos los datos (tiempos y uso de memoria) deben ser ciertos. La invención o manipulación de los mismos puede suponer no superar la práctica.

D.2. Criterios de valoración

La práctica se puntuará de acuerdo con los siguientes criterios:

- Rigor matemático y estadístico, tanto en el estudio teórico, como en el diseño de casos y en el análisis de resultados.
- Utilización adecuada de las herramientas matemáticas y estadísticas.
- Grado de automatización del proceso. Se valorará positivamente la implementación de programas que automaticen las pruebas, frente a la ejecución manual.
- Obtención de conclusiones relevantes y acertadas.

D.3. Otras cuestiones

La práctica se deberá realizar en grupos de dos o tres alumnos, en los mismos grupos que los creados para la primera práctica.

Los profesores de prácticas dejarán en la página web de la asignatura (<http://dis.um.es/~ginesgm/aed.html>) un programa generador de ficheros de log “artificiales”. Se permite usar y modificar este programa para obtener los casos de prueba.

Todos los ficheros generados y la memoria de prácticas deben de estar accesibles en un subdirectorio **practica2** dentro de la cuenta del grupo de prácticas.

Se establece como fecha tope de entrega de esta práctica el viernes 29 de abril de 2005.