Un conjunto de herramientas didácticas sencillas para un curso introductorio sobre modelado y evaluación de computadores

Xavier Molero, Vicente Santonja
Departament d'Informàtica de Sistemes i Computadors
Universitat Politécnica de València
Camí de Vera, s/n. 46022 València
e-mail: {xmolero|visan}@disca.upv.es

Resumen

El presente artículo describe de manera concisa un conjunto de pequeñas herramientas didácticas implementadas mediante programas de fácil uso. Estos programas tratan sobre diversos aspectos de carácter básico en materias sobre modelado y evaluación de sistemas informáticos. El principal objetivo perseguido con el diseño de las herramientas expuestas en este trabajo es que sean fácilmente utilizables por los alumnos que cursan asignaturas relacionadas con esta temática.

1. Introducción

La evaluación de prestaciones o rendimiento (*performance*) de los computadores tiene una gran influencia en el diseño, desarrollo, configuración y sintonización de los sistemas informáticos. El marco que actualmente ofrece el plan de estudios de nuestra Escuela Universitaria de Informática para el tratamiento de la temática relacionada con la evaluación de prestaciones viene representado por la asignatura Evaluación de Sistemas Informáticos (ESI).

La asignatura ESI tiene carácter optativo y se imparte en el primer cuatrimestre del tercer curso. La distribución de créditos es de 3 para las sesiones de teoría y 3 para las de laboratorio, lo que le confiere un aspecto muy práctico. La parte teórica se trata en cuatro temas:

- Introducción a la evaluación del rendimiento
- Monitorización de sistemas informáticos
- Comparación del rendimiento de sistemas: referenciación (*benchmarking*)
- Técnicas analíticas: análisis operacional. Detección de cuellos de botella

Las sesiones de laboratorio inciden en los aspectos más prácticos de la parte teórica: análisis de la configuración de un computador mediante el programa Sandra [6], herramientas básicas de monitorización de sistemas Unix [4,7], resumen y comparación de rendimientos [2,3], diseño de programas de prueba (*benchmarks*) [2,4] y modelado mediante redes de colas de espera [1,2] con el lenguaje de especificación de modelos QNAP2 [5].

2. La necesidad de herramientas prácticas

Dentro del apartado práctico expuesto en el punto anterior, y tras una dilatada experiencia en la impartición de la asignatura ESI, se ha puesto de manifiesto la gran utilidad de disponer de un conjunto reducido de pequeñas herramientas con las cuales el alumno pueda aplicar algunas de las técnicas más usuales que aparecen en el campo de la evaluación de prestaciones.

El hecho más importante no es que estas herramientas tengan una interfaz visual destacada, sino que lo realmente interesante es poder resolver pequeños problemas mediante herramientas diseñadas *ad hoc* y que sean fácilmente portables de una plataforma a otra. Ello ha hecho que se escoja C (en su versión ANSI) como lenguaje de implementación, y que los parámetros se especifiquen en la línea de órdenes del sistema operativo. Por otro lado, se ha incluido una

pequeña ayuda de manejo de cada herramienta, la cual se puede obtener sin más que ejecutar el programa sin ningún parámetro.

3. Breve descripción de las herramientas

A continuación se presenta brevemente un conjunto de cuatro programas de manejo muy sencillo para su aplicación en algunos de los temas básicos de la evaluación de los sistemas informáticos.

Herramienta `amdahl`. Calcula, empleando la ley de Amdahl [3], la aceleración global de un sistema (*speedup*) después de sustituir un componente del mismo por uno k veces más rápido, el cual se utiliza durante una fracción de tiempo f. Esta aceleración se calcula mediante la fórmula:

$$A = \frac{1}{1 - f + \frac{f}{k}}$$

Asimismo, también se calcula, utilizando la misma expresión, el aumento del coste del sistema a partir del coste total C del mismo, el del componente que se reemplaza Cr y el del componente nuevo Cn. La Figura 1 muestra un ejemplo sencillo de uso donde se puede ver el valor que toma cada parámetro de entrada al programa.

```
amdahl   f  k       C     Cr     Cn

amdahl 0.8  3  150000 56000 90000

      Aceleración del sistema: 2.14
      Incremento del coste:    1.23
```

Figura 1: Ejemplo de uso del programa `amdahl`

Herramienta `zerotest`. Esta herramienta resulta muy útil para comparar el rendimiento de dos computadores [2,3]. Dados un total de n programas de prueba, este programa analiza los tiempos de ejecución de dichos programas en dos computadores, y calcula la significación estadística de las diferencias observadas. Si el intervalo de confianza calculado con un nivel de confianza del 95% para estas diferencias incluye el cero entonces no hay una diferencia significativa de rendimientos. Un ejemplo de aplicación de esta herramienta para el caso de 4 programas de prueba se muestra en la Figura 2.

```
zerotest n    x1   x2   x3   x4   y1   y2   y3   y4

zerotest 4  23.3 11.8 14.8 87.2 26.9 14.1 13.7 98.6

 Media aritmética diferencias:      -4.05
 Desviacion típica diferencias:      5.29
 Intervalo de confianza diferencias: [-12.46,4.36]
 Análisis: no hay diferencias significativas
```

Figura 2: Ejemplo de uso del programa `zerotest`

Herramienta `merrill`. Esta herramienta calcula el índice de Merrill de un gráfico de Kiviatt [2]. La forma de este tipo de gráficos es la de un polígono donde la longitud de sus lados está determinado por el valor de 8 índices x_i de rendimiento, la mitad buenos y la otra mitad malos, dispuestos alternativamente. Los índices pueden representar la utilización de determinados componentes del sistema (como procesador, entrada/salida, red, etcétera). El índice de Merrill varía entre 0 y 100 y se calcula según la fórmula:

$$Q = \sqrt{\frac{1}{2n}\sum_{i=1}^{n}(x_{2i-1} + x_{2i+1})(100 - x_{2i})}$$

Cuanto mayor sea el valor de este índice Q mejor será el rendimiento del sistema informático, ya que el gráfico de Kiviatt asociado se parecerá más a una estrella. Para comparar el rendimiento de dos computadores basta comparar sus índices de Merrill teniendo en cuenta que sólo es significativa la parte entera de este índice. La Figura 3 muestra un ejemplo de resolución.

```
merrill  x1 x2 x3 x4 x5 x6 x7 x8

merrill 100 60 40  0 40  0 40 60

  Cuantificación de Merrill (0..100): 58.31
```

Figura 3: Ejemplo de uso del programa `merrill`

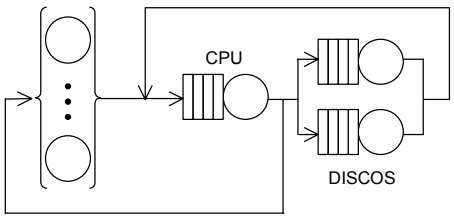
Herramienta `solvenet`. Esta herramienta es, con mucho, la más compleja de todas las presentadas hasta ahora. El objetivo fundamental es resolver modelos sencillos de redes de colas de espera [1,2] mediante la aplicación del algoritmo del valor medio o los algoritmos para redes de colas abiertas. La herramienta está pensada como alternativa al programa QNAP2 en los casos de redes más simples. Para facilitar la lectura de resultados se ha optado por utilizar también una interfaz similar a la empleada por QNAP2, además de aportar información adicional sobre cuellos de botella y asíntotas del rendimiento que resultan muy útiles para el análisis de las prestaciones del sistema. La Figura 4 muestra cómo se especifican los parámetros de entrada.

```
solvenet [0|1] [lambda| N Z] tcpu nio Sio1 Vio1...Sion Vion

  red:     0 (abierta) 1 (cerrada)
  lambda:  tasa de llegada(sólo para redes abiertas)
  N:       número de trabajos (sólo para redes cerradas)
  Z:       tiempo de reflexión(sólo para redes cerradas)
  tcpu:    tiempo de servicio de la CPU
  nio:     número de dispositivos de I/O
  Sio:     tiempo de servicio del dispositivo de I/O nº i
  Vio:     razón de visita o probabilidad de encaminamiento
           del dispositivo de I/O nº i
```

Figura 4: Parámetros de entrada al programa `solvenet`

Para ilustrar el funcionamiento de este programa, se ha considerado la red de colas mostrada en la Figura 5. En esta figura también se especifican los valores de los tiempos de servicio S_i de las estaciones así como la razón de visita V_i.



Z = 25 $S_{CPU} = 0.05$, $V_{CPU} = 31$
N = 15 $S_{D1} = 0.03$, $V_{D1} = 20$
$S_{D2} = 0.05$, $V_{D2} = 10$

Figura 5: Ejemplo de red de colas cerrada

La información aportada por el programa `solvenet` se refleja en la Figura 6 de la página siguiente. Como se puede apreciar, el programa aporta tanto información referida a cada estación de servicio del modelo como del sistema en su conjunto. La información de cada estación, expresada en valores medios, incluye la utilización, el número de clientes en toda la estación así como en la cola de espera, el tiempo de respuesta, la productividad y la demanda de servicio.

En cuanto a la información referida al sistema informático completo, el programa calcula, a diferencia del programa QNAP2 que sólo da información por estación de servicio, el número de trabajos en el sistema central y en reflexión, el tiempo de respuesta y su valor mínimo, la productividad y su valor máximo. Por otro lado, también indica los límites asintóticos optimistas de la productividad y del tiempo de respuesta del sistema en función del número de trabajos.

4. Conclusión

En este trabajo se ha presentado un conjunto de herramientas sencillas diseñadas *ad hoc* y destinadas fundamentalmente a ser utilizadas en un curso básico de modelado y evaluación de computadores.

El hecho de estar diseñadas en lenguaje C y disponer de parámetros indicados en la línea de órdenes permiten que los alumnos puedan utilizarlas fácilmente. En la actualidad se está trabajando en otra serie de herramientas adicionales para tratar aspectos relacionados con la monitorización de sistemas Unix.

Referencias

[1] J. Cady and B. Howarth. *Computer systems performance, management and capacity planning*. Prentice may, 1990.

[2] R. Jain. *The art of computer system performance analysis. Techniques for experimental design, measurement, simulation and modeling*. John Wiley & Sons, 1991.

[3] D.J. Lilja. *Measuring computer performance. A practitioner's guide*. Cambridge University Press, 2000.

[4] M. Loukides. *System performance tuning*. O'Reilly & Associates, Inc., 1992
[5] QNAP2. *Reference manual* (Tomes I and II). Simulog, 1996.
[6] Sandra: *The System ANalyser, Diagnostic and Reporting Assistant program*. Versión de uso gratuito disponible en la dirección web www.sisoftware.co.uk/sandra
[7] Varios autores. *Unix performance tuning*. R&D Books, 1997.

```
QUEUEING NETWORK: CLOSED
Mean Value Analysis Algorithm

*********************************************************
*   NAME    *   UTIL    *  CUST NB * RESPONSE *  THRUPUT *
*********************************************************
*           *           *          *          *          *
* CPU       *    0.7534*    2.1342*    0.1416*   15.0687*
*           *           *          *          *          *
* I/O  1    *    0.2917*    0.3996*    0.0411*    9.7217*
*           *           *          *          *          *
* I/O  2    *    0.2430*    0.3141*    0.0646*    4.8609*
*           *           *          *          *          *
*********************************************************

*********************************************************
*   NAME    *  VISIT    *  SERVICE *  DEMAND  *  CUST NQ *
*********************************************************
*           *           *          *          *          *
* CPU       *   31.0000*    0.0500*    1.5500*    1.3808*
*           *           *          *          *          *
* I/O  1    *   20.0000*    0.0300*    0.6000*    0.1079*
*           *           *          *          *          *
* I/O  2    *   10.0000*    0.0500*    0.5000*    0.0710*
*           *           *          *          *          *
*********************************************************

      ***********************************
      *         SYSTEM VARIABLES        *
      ***********************************
      *                         *       *
      * WORKING CUSTOMERS       *    2.8479*
      * THINKING CUSTOMERS      *   12.1521*
      * ALL CUSTOMERS           *        15*
      * SATURATION POINT        *        18*
      *                         *       *
      * RESPONSE TIME           *    5.8588*
      * MINIMUM RESPONSE TIME   *    2.6500*
      *                         *       *
      * THROUGHPUT              *    0.4861*
      * MAXIMUM THROUGHPUT      *    0.6452*
      *                         *       *
      ***********************************
```

Figura 6. Resultados generados por el programa `solvenet` para el caso de red de colas de la Figura 5.