TPCC-UVA: Implementación del Benchmark TPC-C

Julio A. Hernández, Eduardo Hernández, Diego R. Llanos

Resumen— **Los benchmarks de transacciones sobre una base de datos son herramientas muy útiles para medir el rendimiento global de un sistema. Los resultados de estos benchmarks son de gran utilidad para los clientes que buscan comparar el rendimiento de diferentes máquinas en aplicaciones de gestión. El benchmark TPC-C, del *Transaction Procesing Performance Council* (TPC) es uno de los más utilizados por los fabricantes. Se trata de un benchmark cuyas especificaciones son de dominio público, pero del que no se conocen hasta la fecha implementaciones de libre distribución. Se presenta en este artículo una implementación de TPC-C versión 5, denominada TPCC-UVA, cuyo diseño permite que sea portable a cualquier entorno UNIX/Linux compatible con el estándar System V, permitiendo hacer comparativas entre diferentes sistemas.**

La implementación del benchmark cumple con los requisitos del TPC-C en lo que respecta a rendimiento, medido en transacciones por minuto (tpmC). Dado que se trata de una implementación pensada fundamentalmente para un uso no comercial, no se ha utilizado ésta para cálculos de tipo precio/rendimiento (price/tpmC). Por el mismo motivo, se ha utilizado un monitor de transacciones hecho a medida, lo que impide comparar los resultados obtenidos con TPCC-UVA con resultados obtenidos a través de otras implementaciones del estndar TPC-C.

Palabras clave— **Benchmarks, TPC-C, medición de rendimientos.**

I. Introducción

EN la actualidad existen multitud de benchmarks destinados a medir el rendimiento de arquitecturas en paralelo. Estos benchmarks se diferencian por los dominios de aplicación que simulan, así como por sus características de ejecución.

Por un lado están los benchmark de cálculo numérico puro. Un ejemplo de esto son los *Nas Parallel Benchmarks* [4] que consisten en una serie de programas de cálculo que intentan reflejar aspectos encontrados frecuentemente en aplicaciones de cálculo en paralelo, como por ejemplo la resolución de ecuaciones de FFT en tres dimensiones. Otro ejemplo son los benchmarks SPEC [3] que se componen de un serie de programas en aritmética entera y de coma flotante.

Este tipo de benchmarks solamente proporciona información relevante de una parte de la arquitectura sometida a examen. Sus resultados suministran mucha información en cuanto a rendimientos de CPU, pero no reflejan otros aspectos importantes de la arquitectura, como pueden ser la velocidad de acceso a los dispositivos de almacenamiento, de entrada-salida, etcétera.

Departamento de Informática, Universidad de Valladolid, Edificio de Tecnologías de la Información y las Telecomunicaciones, Campus Miguel Delibes, 47011 Valladolid (España). e-mail: diego@infor.uva.es

Una de las posibilidades para medir el rendimiento del sistema de una forma global son los benchmarks de bases de datos, ya que presentan una carga de trabajo que incluye los aspectos mencionados anteriormente.

Otro motivo que hace interesantes a los benchmarks de bases de datos es que la información que proporcionan resulta muy relevante en los sectores comerciales. Sectores como el comercial o el financiero se presentan como campos en los que las tecnologías de computación en paralelo juegan un papel muy importante. Esto hace que las compañías prefieran este tipo de benchmarks a la hora de decidir qué máquina es la mejor para sus propósitos. El motivo principal es que estos benchmarks presentan una carga de trabajo que simula un entorno de aplicación muy similar al de esas empresas.

El benchmark TPC-C versión 5 [1] del *Transaction Processing Performance Council* (TPC) [2] se basa en la ejecución de una mezcla de transacciones interactivas, y simula características propias de un entorno OLTP tales como el encolado de transacciones o su cancelación (*rollback*). La medida de rendimiento se expresa en transacciones por minuto (tpmC). Este benchmark incluye unos criterios de escalado que hacen que el problema sea mucho mas realista: el tamaño de la base de datos y el número de terminales que ejecutan transacciones se ajustan a la capacidad de la máquina a medir, lo que da lugar a una mayor o menor tasa de tpmC.

TPC-C requiere de una serie de emuladores de terminal remoto que simulen una población de usuarios, con sus terminales, que ejecutan transacciones contra una base de datos. Este benchmark simula la actividad de un proveedor mayorista con una serie de distritos de ventas y almacenes distribuidos geográficamente. La carga de trabajo que implementa el TPC-C consiste en órdenes de cliente, ejecución de pagos, ejecución de consultas, repartos y chequeos de inventario, lo que da lugar a cinco tipos de transacciones con distintos grados de complejidad.

Otras características del TPC-C son la complejidad de la estructura de la base de datos, un conjunto de requisitos realistas para las pantallas de E/S de los terminales, los accesos a la base de datos a través de llaves primarias así como de llaves no primarias, la completa transparencia en las particiones de datos, o la ya mencionada cancelación de transacciones.

Debido a todas la características anteriores, cada vez más consumidores solicitan a los fabricantes de sistemas informáticos resultados de rendimiento expresados según el estándar TPC-C. Como consecuencia, el benchmark TPC-C es uno de los estándares

más utilizados en la actualidad por dichos fabricantes.

Las especificaciones del benchmark TPC-C son de dominio público. A pesar de ello, no tenemos noticia hasta la fecha de ninguna implementación al alcance de cualquier usuario. Las implementaciones existentes son propiedad de las empresas que conforman el TPC, que las utilizan para la promoción de sus productos.

La filosofía del proyecto es construir una herramienta que permita a cualquier usuario realizar cálculos de rendimiento en diferentes máquinas y comparar resultados. Con este fin, se ha desarrollado una implementación de libre distribución del benchmark TPC-C denominada TPCC-UVA [8]. Se trata de una implementación completa del benchmark en lo que se refiere al cálculo de las transacciones por minuto (tpmC), utilizada para medir y comparar el rendimiento entre sistemas. Dado que se trata de una implementación sin fines comerciales, no se ofrece soporte para el cálculo de parámetros de coste/rendimiento (price/tpmC).

El resto del documento se organiza como sigue. En la sección II se describe brevemente cada uno de los módulos que componen el benchmark. En la sección III se explican los mecanismos de comunicación entre los módulos involucrados en el test. En la sección IV se analizarán algunos resultados obtenidos con el benchmark. En la sección V se exponen las limitaciones de esta implementación del TPC-C. Finalmente, en la sección VI se enumeran nuestras conclusiones.

II. Esquema del Benchmark

Como se ha dicho anteriormente, el benchmark TPC-C ejecuta un test de rendimiento cuya carga de trabajo viene determinada por la actividad de un conjunto de terminales que solicitan la ejecución de un conjunto de transacciones contra una base de datos.

Los cinco tipos de transacciones que se definen son:

New-Order Introduce una orden completa a través de una única transacción de base de datos.

Payment Realiza el pago de un cliente, actualizando su balance actual.

Order-Status Comprueba el estado de la última orden del cliente.

Delivery Registra el reparto de una orden pendiente para cada distrito.

Stock-Level Realiza un recuento de los artículos cuyas existencias son menores que un umbral dado.

El número de transacciones de tipo New-Order por minuto es lo que determina la tasa de rendimiento tpmC.

Cuando un terminal envía la solicitud de ejecución de una transacción, espera a recibir los resultados, salvo en el caso de la transacción Delivery, que simula un tipo de transacción no interactiva.

La estructura de la base de datos contra la que se ejecutan las transacciones se compone de nueve tablas con distintas características en cuanto a su esquema y cardinalidad.

La implementación del benchmark se compone de cinco módulos que interactúan entre sí a fin de realizar todas las funciones necesarias para la medición del rendimiento. Estos módulos son:

- Controlador del Benchmark.
- Conjunto de Emuladores de Terminal Remoto (ETR).
- Monitor de Transacciones (MT).
- Controlador de Checkpoints.
- Controlador de Limpiezas.

Se ha utilizado postgreSQL-7.1.3 [6] para mantener la base de datos y ejecutar las transacciones solicitadas por los terminales. El esquema general del benchmark se describe a continuación a partir de la figura 1.

A. Controlador del Benchmark

Proporciona la interfaz con el usuario para el acceso a las funciones del benchmark. Además, realiza las operaciones necesarias para llevar a cabo esas funciones, que pasamos a describir.

A.1 Población inicial de la base de datos

Se crea una nueva base de datos para el test, compuesta de las nueve tablas definidas en el estándar TPC-C con los requisitos de escalamiento y población requeridos. Además se incluyen, en la base de datos, mecanismos que aseguran la integridad referencial (*Primary keys* y *Foreign keys*).

A.2 Restauración de una base de datos existente

Se eliminan las modificaciones producidas por un test en la base de datos. El objetivo de esta función es obtener una base de datos con los requisitos necesarios para realizar un nuevo test, evitando la necesidad de crear una base de datos nueva debido al tiempo que conlleva dicha operación.

A.3 Ejecución del test

Se ponen en funcionamiento las partes del benchmark utilizadas para la ejecución del test de rendimiento, considerando los parámetros seleccionados por el usuario: número de almacenes, periodo de rampa, periodo de medida y la configuración del Controlador de Limpiezas.

El test de rendimiento se compone de tres periodos. Al primero se le denomina *periodo de rampa*, y permite que la medida de rendimientos se realicen a partir de que el rendimiento del sistema haya alcanzado un estado estable. Al segundo se le denomina *periodo de medida*, y es en el que se realiza la medida de rendimiento. El tercero se denomina *periodo de fin de test*, y es el tiempo que el Controlador del Benchmark necesita para desactivar todos los procesos involucrados.

El controlador del benchmark se encarga de activar el Monitor de Transacciones, diez Emuladores de Terminal Remoto por cada almacén seleccionado

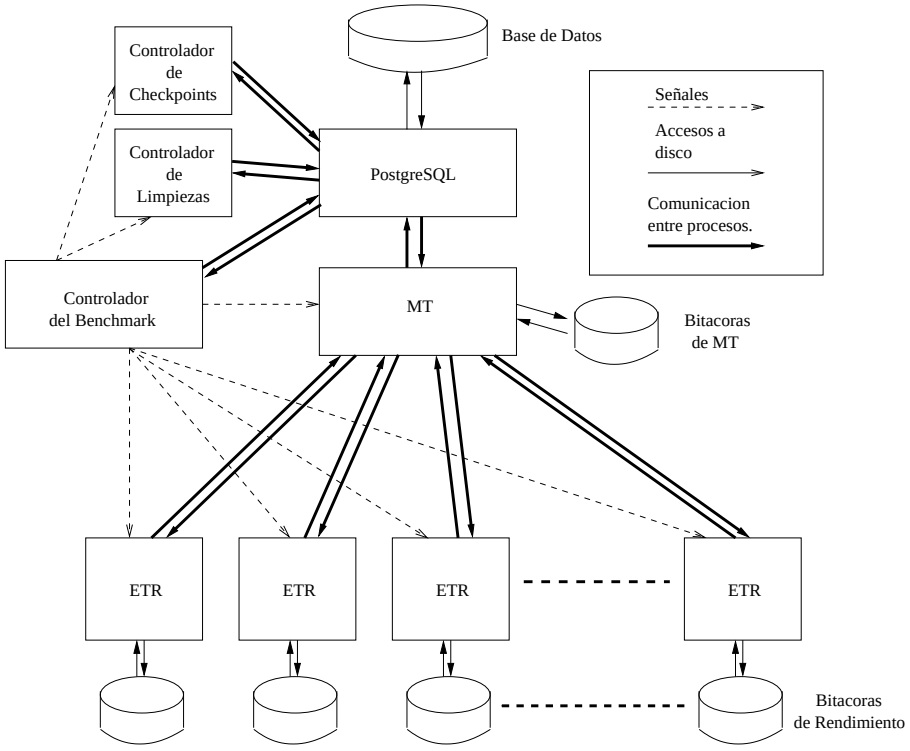

Fig. 1. Esquema General del benchmark

por el usuario, el Controlador de Checkpoints y en su caso el Controlador de Limpiezas. Además se realiza la temporización de los periodos de los que se compone el test, informando a los módulos involucrados en la medición del rendimiento, mediante envío de señales, de la expiración del periodo de test para provocar su desactivación.

A.4 Comprobación de la consistencia de la base de datos

Se comprueba que la base de datos cumple con las condiciones de consistencia especificadas en el estándar TPC-C.

A.5 Eliminación de la base de datos existente

Se procede a la eliminación de la base de datos, en el caso de que exista.

A.6 Recuento de resultados del último test realizado

Se realiza la lectura y el tratamiento de los ficheros de bitácora producidos por el conjunto de ETRs y el MT que contienen la información necesaria para determinar el rendimiento alcanzado por la máquina.

El conjunto de datos sobre el test que ofrece el Controlador del Benchmark se puede dividir en dos grupos claramente diferenciados. Por un lado se encuentra el número de transacciones de tipo New-Order ejecutadas por minuto (tpmC) y el tiempo de respuesta de las transacciones ejecutadas. Este grupo determina el rendimiento del sistema. Por otro lado se encuentran los datos que determinan que el test se ha realizado según las especificaciones, tales como los tiempos de espera de los terminales o los porcentajes de cada transacción dentro de la mezcla de transacciones.

B. Conjunto de Emuladores de Terminal Remoto (ETR)

Realizan las siguientes funciones:

- Emulación de usuario: se simula la entrada de información de un usuario: elección de transacciones y generación de datos para las transacciones. Además se simulan dos tiempos de espera de usuario: tiempo de teclado y tiempo de pensar.
- Emulación de terminal: muestra de los datos introducidos por el usuario emulado y los datos resultantes de la ejecución de la transacción.
- Envío de transacciones y recepción de resultados a través de los mecanismos de comunicación con el Monitor de Transacciones.
- Medida del tiempo de respuesta de transacción y su registro en una bitácora. En esta bitácora se registran datos adicionales necesarios para el cálculo del rendimiento.

El ETR recibe la señal de expiración del periodo de medida enviada por el controlador del benchmark para la desactivación de las funciones.

C. Monitor de Transacciones (MT)

El Monitor de Transacciones actúa como mediador entre el conjunto de terminales y el motor de la base de datos, encargándose de la recepción de las transacciones enviadas por los terminales y la ejecución de las mismas según su orden de llegada. El MT, además, registra en las bitácoras correspondientes los datos resultantes de la ejecución aplazada de

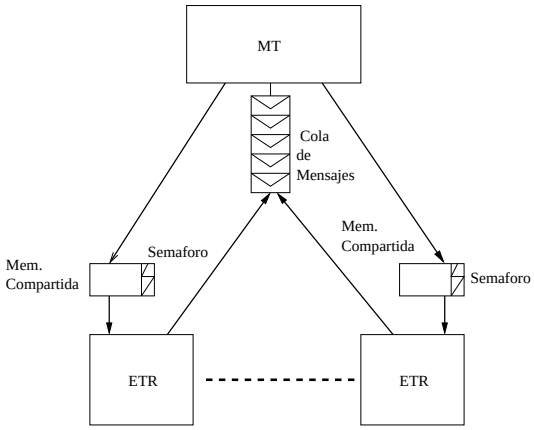

Fig. 2. Comunicación ETR - MT

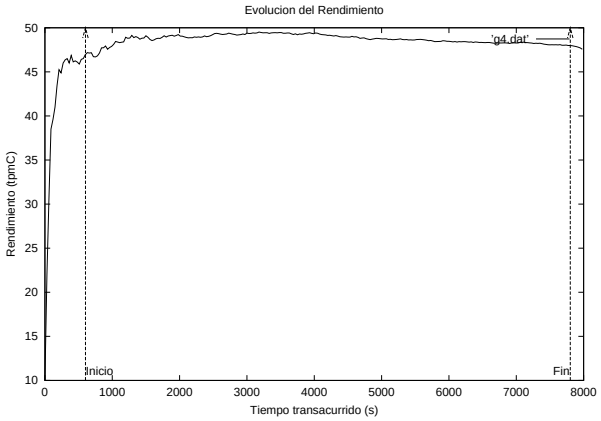

Fig. 3. Evolución del rendimiento. Test 1.

la transacción Delivery y los datos acerca de posibles errores en la ejecución de las transacciones.

Tras recibir la señal de expiración del periodo de medida enviada por el controlador del benchmark, el MT procede a su desactivación.

D. Controlador de Checkpoints

Es el encargado de la realización de checkpoints periódicos en la base de datos y del registro de los sellos de hora de comienzo y fin de los checkpoints. El primer checkpoint se realiza inmediatamente después de su activación, que se produce al comienzo del intervalo de medida.

E. Controlador de Limpiezas

El Controlador de Limpiezas se encarga de eliminar el efecto producido por la continua ejecución de operaciones en la base de datos. Por cada ejecución, postgreSQL mantiene información residual que relentiza dichas operaciones. Por ello será necesario efectuar limpiezas periódicas para evitar que decaiga el rendimiento. Volveremos sobre este problema en la sección V. El Controlador de Limpiezas se encarga de realizar dichas limpiezas periódicamente sobre la base de datos, ejecutando el comando `vacuum` [7] de postgreSQL. El usuario puede elegir si desea realizar estas limpiezas periódicas. Si lo hace puede configurar el intervalo entre ellas así como su número máximo.

III. Mecanismos de Comunicación entre el conjunto de ETRs y el TM

La comunicación se establece utilizando los mecanismos de comunicación entre procesos IPC del UNIX System V: semáforos, memoria compartida y colas de mensajes [5].

Como se puede apreciar en la figura 2, un ETR se comunica con el MT a través de dos canales unidireccionales: la memoria compartida y la cola de mensajes, que están sincronizados por medio de un semáforo. Los mensajes de información que tienen como destino el MT se envían a través de la cola de mensajes, mientras que los mensajes que tienen como destino el ETR se envían a través de memoria compartida.

El ETR envía un nuevo mensaje a través de la cola de mensajes, y cierra el semáforo de sincronización. Cuando el MT recibe el mensaje y termina de procesarlo, abre el semáforo permitiendo al ETR el acceso a los resultados.

IV. Resultados del benchmark TPCC-UVA

A continuación se incluyen los resultados de dos tests ejecutando, en dos sistemas diferentes, el benchmark TPCC-UVA. Debido a la extensión de los datos que ofrece este benchmark, únicamente se muestran los más relevantes en lo que al rendimiento del sistema se refiere.

A. Test 1

Las características del sistema donde se ha ejecutado el test son: procesador AMD Athlon k7-1GHz, 256 Mb de memoria RAM DIMM-133Mhz, memoria cache L1 128Kb-L2 512kb, disco duro 40Gb - 7200 rpm, sistema operativo Red Hat Linux 7.2, versión del kernel 2.4.7-10, compilador gcc-2.96, preprocesador SQL ecpg-2.8, motor de base de datos postgreSQL-7.1.3.

El test se realizó con una carga de 4 almacenes, con periodos de rampa y de medida de 10 y 120 minutos respectivamente y sin realizar limpiezas en la base de datos.

El rendimiento máximo medido en el sistema con la configuración anterior y cumpliendo con los requisitos que especifica el estándar TPC-C en cuanto a la correcta ejecución de la carga de trabajo, fue de 49,000 tpmC. La figura 3 muestra una gráfica con la evolución del rendimiento a lo largo del test.

En esta gráfica se puede apreciar cómo el intervalo de medida comenzó antes de que el rendimiento se estabilizara completamente, por lo que el tiempo de rampa debería haber sido mayor. Además se puede comprobar que, a mitad de la ejecución del test, el rendimiento comienza a descender levemente. El descenso del rendimiento es causado por el motor de base de datos. Describiremos esta circunstancia en la sección V.

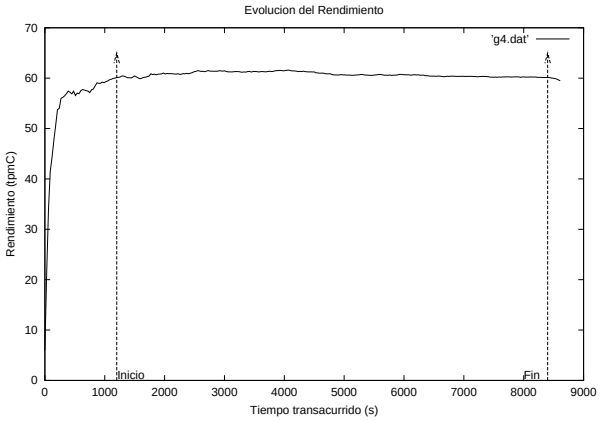

Fig. 4. Evolución del rendimiento. Test 2.

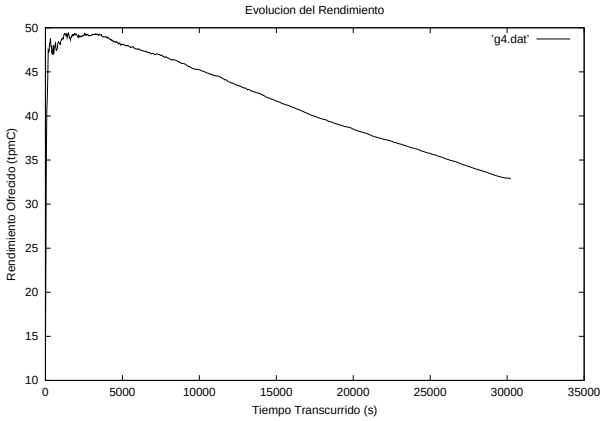

Fig. 5. Test de ocho horas sin limpiezas en la base de datos.

B. Test 2

Las caracteristicas del sistema donde se ha ejecutado son: procesador AMD Athlon XP-1,9GHz, 512 Mb de memoria RAM DDR-333Mhz, memoria cache L1 128Kb-L2 512kb, disco duro 60Gb - 7200 rpm, sistema operativo Red Hat Linux 7.2, versión del kernel 2.4.7-10, compilador gcc-2.96, preprocesador SQL ecpg-2.8, motor de base de datos postgreSQL-7.1.3.

El test se realizó con una carga de 5 almacenes, con periodos de rampa y de medida de 20 y 120 minutos respectivamente y sin realizar limpiezas en la base de datos.

El rendimiento máximo medido en el sistema con la configuración anterior y cumpliendo con los requisitos que especifica el estándar TPC-C en cuanto a la correcta ejecución de la carga de trabajo, fue de 61,375 tpmC. En la figura 4 se muestra una gráfica con la evolución del rendimiento a lo largo del test.

Cabe destacar que, según indica la gráfica, el periodo de medida comenzó tras alcanzar el estado estable. Se puede comprobar que la curva es más plana que la del test anterior, de lo que se deduce que en este sistema postgreSQL mantiene mejor el rendimiento.

V. Limitaciones de la implementación

La actual implementación del benchmark presenta limitaciones en cuanto al cumplimiento de los tiempos de respuesta máximos de las transacciones, especificados en el estándar del TPC-C. Debido a que el Monitor de Transacciones ejecuta las transacciones en serie según el orden de llegada, cada transacción espera en la cola el tiempo de ejecución de todas las que están por delante, a lo que hay que sumar el suyo propio. Debido a esto, el tiempo de respuesta no sólo se ve afectado por el número de terminales existentes, sino también por la espera de cada transacción en la cola.

La solución a esto pasa por desarrollar un Monitor de Transacciones que sea capaz de paralelizar dichas transacciones, cumpliendo los requisitos de aislamiento especificados en el estándar, o bien utilizar un monitor comercial, lo que impediría la libre distribución de este software. Debido a esto, los resultados obtenidos a través de la ejecución de TPCC-UVA sólo son válidos al objeto de comparar el rendimiento de diferentes sistemas en los que se ha ejecutado este software. En consecuencia, no debe utilizarse la medida de transacciones por minuto (tpmC) obtenida a través de TPCC-UVA para efectuar comparaciones con tpmC obtenidos a través de otras implementaciones.

Otra limitación que presenta esta implementación del benchmark se debe a que postgreSQL no es capaz de mantener el rendimiento en periodos de ejecución mayores de tres horas, lo que obliga, como se ha mencionado, a realizar limpiezas periódicas en la base de datos a lo largo del test. Estas limpiezas periódicas tienen un fuerte impacto sobre la evolución del rendimiento, aunque cabe destacar que a partir de un cierto número de limpiezas, el rendimiento se estabiliza notablemente.

Un ejemplo de lo anterior se muestra en las figuras 5 y 6, que representan la evolución del rendimiento en un test de ocho horas con cuatro almacenes en un PC Athlon k7, con las características descritas en la sección IV.A. La figura 5 se corresponde con un test en el que no se han realizado limpiezas en la base de datos. En ella se puede apreciar cómo disminuye progresivamente el rendimiento. El rendimiento obtenido fue de 33,027 tpmC. La figura 6 se corresponde con un test en el que se han realizado limpiezas cada hora. El rendimiento aumentó a 43,617 tpmC. Se puede observar cómo ha aumentado el rendimiento final además de cómo se ha establizado su evolución, frenándose su caída. Sin embargo se puede ver cómo cada limpieza provoca una perturbación en la curva del rendimiento.

En consecuencia, para realizar comparativas entre distintas máquinas se debe buscar una configuración para la cual la evolución del rendimiento durante el periodo de medida sea estable, y comparar únicamente los resultados obtenidos durante el periodo de medida, que deberá ser el mismo para ambas pruebas.

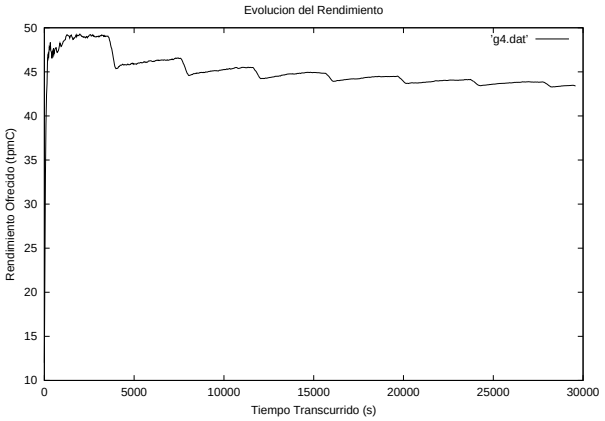

Fig. 6. Test de ocho horas con limpiezas en la base de datos.

VI. Conclusiones

El benchmark TPC-C constituye una herramienta muy útil para el cálculo de rendimientos de un sistema a través de la ejecución de un conjunto de transacciones sobre una base de datos. En este trabajo se presenta una implementación de dominio público de dicho benchmark. Para su funcionamiento utiliza el motor de base de datos postgreSQL-7.1.3. Se trata de una implementación que cumple los requisitos especificados en el estándar, a excepción de la utilización de un Monitor de Transacciones disponible comercialmente, ya que se utiliza uno hecho a medida. Pese a ello, creemos que se trata de una herramienta válida para la comparación entre sistemas existentes o para su ejecución a través de mecanismos de simulación.

Esta implementación se distribuye gratuitamente bajo la licencia GPL, y puede obtenerse junto con los manuales de uso y documentos relacionados en la siguiente página web:

`http://www.infor.uva.es/~diego/tpcc.html`

Referencias

[1] TPC Benchmark TM C. Standard Specification Revision 5.0. 6 Febrero 2001. http://www.tpc.org/tpcc/default.asp. Fecha de acceso: Mayo 2002.

[2] Transaction Processing Performance Council. http://www.tpc.org. Fecha de acceso: Mayo 2002.

[3] Standard Performance Evaluation Corporacion. http://www.specbench.org. Fecha de acceso: Mayo 2002.

[4] NAS Parallel Benchmark. http://science.nas.nasa.gov/Software/NPB. Fecha de acceso: Mayo 2002.

[5] Advanced Programing in the UNIX Environment. W. Richard Stevens. Ed. Addison - Wesley 1993.

[6] PostgreSQL. http://www.postgresql.org. Fecha de aceso: Mayo 2002.

[7] PostgreSQL 7.1 Reference Manual. The postgreSQL Global Development Group. Disponible en: ftp.es.postgresql.org/postgresql/doc/7.1/reference.pdf. Fecha de acceso: Abril 2002.

[8] Implementación en C del benchmark de transacciones distribuidas TPCC. Eduardo Hernández Perdiguero, Julio A. Hernández Gonzalo, Proyecto de Fin de Carrera, I. T. Telecomunicación. Escuela Universitaria Politécnica de Valladolid, julio 2002.