



Universitas, Revista de Ciencias Sociales
y Humanas

ISSN: 1390-3837

trubio@ups.edu.ec

Universidad Politécnica Salesiana
Ecuador

Jiménez Govea, Aarón; Mireles Mercado, Marisela; Torres López, Laura
Sistema gestor con características adaptativas, para el control en la ejecución de
aplicaciones de usuario que solicitan una alta demanda de tiempo procesador
Universitas, Revista de Ciencias Sociales y Humanas, núm. 21, julio-diciembre, 2014, pp.
191-207
Universidad Politécnica Salesiana
Cuenca, Ecuador

Disponible en: <http://www.redalyc.org/articulo.oa?id=476147261010>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Sistema gestorador con características adaptativas, para el control en la ejecución de aplicaciones de usuario que solicitan una alta demanda de tiempo procesador

*Management systems with adaptive characteristics to control
and execute user applications that require high processing times*

Aarón Jiménez Govea,¹ Marisela Mireles Mercado,² Laura Torres López³
aaron@cencar.udg.mx marisela.mireles@gmail.com ltorres@redudg.udg.mx

Resumen

En el campo computacional se han hecho avances relacionados con el ahorro de energía, poniendo especial atención en el diseño e implementación de componentes electrónicos con el objetivo de reducir su consumo. Sin embargo, existen casos donde se requiere un mayor gasto de energía, un ejemplo de ello son las aplicaciones que solicitan una alta demanda de tiempo procesador, algunas se asocian con el modelaje de imágenes (renderizado), cuya ejecución se prolonga por minutos o incluso días, generando un sobrecalentamiento en el procesador, lo que podría provocar un daño grave en el circuito o en algún otro componente del sistema. Una alternativa a tal problemática radica en generar una estrategia para la gestión de los procesos de usuario. El presente trabajo propone la optimización del manejo energético por medio de una estrategia de ejecución para los procesos de usuario, aplicando un manejo adaptativo, gestionando la asignación del porcentaje de uso del procesador con la finalidad de priorizar o restringir la ejecución de las solicitudes hechas por los usuarios.

Forma sugerida de citar: Jiménez Govea, A., Mireles Mercado, M., & Torres López, L. (2014). Sistema gestorador con características adaptativas, para el control en la ejecución de aplicaciones de usuario que solicitan una alta demanda de tiempo procesador. *Universitas*, XII(2), pp. 191-207. Quito: Editorial Abya Yala/Universidad Politécnica Salesiana.

- 1 Ingeniero en Computación, egresado de la Universidad de Guadalajara del estado de Jalisco en 1998. Obtuvo el grado de Maestría en Sistemas de Información en 2003, actualmente estudiante del Doctorado en Tecnologías de Información de la Universidad de Guadalajara, Jalisco, México.
- 2 Ingeniera en Computación, egresada de la Universidad de Guadalajara del estado de Jalisco en 1998. Obtuvo el grado de maestría en Sistemas de Información en 2006, actualmente estudiante del Doctorado en Tecnologías de Información en la Universidad de Guadalajara, Jalisco, México.
- 3 Ingeniera en Sistemas Computacionales egresada del ITESO en el estado de Jalisco en 1991. Obtuvo el grado de Maestría en Comercio Electrónico en 2005 por el ITESM Campus Guadalajara y el Doctorado en Tecnologías de Información con Especialidad en Sistemas Distribuidos en 2013 por la Universidad de Guadalajara, Jalisco, México.

Palabras claves

Gestión de procesos computacionales, asignación de tiempo procesador, ejecución de procesos de usuario, manejo adaptativo, consumo energético, optimización de energía.

Abstract

Progress has been made in the field of information systems when it comes to energy savings, particularly around the design and implementation of electronic components meant to reduce energy consumption. However, there are times when more energy is needed, associated with image modelling (rendering) that can take minutes and even days to process, which overheats processors and can severely damage circuits or other components. An alternative to this problem is coming up with a strategy for user's processes management. We propose energy optimization through execution strategies for user processes by applying an adaptive system management and assigning to it a percentage of the processor's use, thus prioritizing and restricting the execution of requests made by users.

Keywords

Computer processes management, processing time optimization, execution of user processes, adaptive management, energy consumption, energy optimization.

Introducción

Un aspecto que recientemente ha tomado una especial relevancia a nivel mundial es el desarrollo de estrategias relacionadas con el ahorro de energía. En particular en el área computacional se han concretado avances en el diseño e implementación de componentes electrónicos con el objetivo de reducir el consumo de energía. En el campo de la computación, existe un especial interés en optimizar el consumo de energía de microcomponentes electrónicos. Sin embargo, se presentan situaciones en las que una excesiva carga de trabajo acarrea consigo un incremento en el consumo de energía, como ejemplo citaremos el caso de un microprocesador, el cual sin duda es un componente básico y representativo de un sistema electrónico, usado en los teléfonos celulares, sistemas computacionales y todo aquel dispositivo que requiera de una administración en su operación. Al presentarse un incremento en el porcentaje de uso del procesador también se detecta un aumento en la temperatura, lo cual se

traduce en una mayor disipación de calor y consumo de energía, por lo que una alternativa a tal problemática radica en generar una estrategia para la gestión de los procesos de usuario (Stallings, 2006, Tanenbaum, 2001, 2008), con la intención de reducir por lapsos de tiempo la carga de trabajo del procesador, acción que traería como consecuencia bajar la temperatura del CPU y al disminuir la disipación de calor se tiene una optimización del manejo energético.

Descripción del problema

En ambientes computacionales es frecuente encontrar escenarios donde se involucran tareas que compiten por obtener tiempo de ejecución, pero desde el punto de vista de un usuario, se percibe un aparente reparto inequitativo de recursos, ya que la ejecución de una aplicación toma un carácter prioritario para el interesado, sin embargo es responsabilidad del sistema el otorgar los turnos de ejecución, por lo que un proceso podría ser forzado a esperar, el caso contrario ocurre cuando una tarea monopoliza el uso del procesador (Tanenbaum, 2008, Stallings, 2005, Tanenbaum, 2007, Dhamdhere, 2012). Para el desarrollo de esta investigación abordaremos la segunda posibilidad. Existen situaciones en las cuales los procesos debido a su naturaleza requieren de una alta demanda de tiempo procesador, un ejemplo de ello son las aplicaciones usadas para el diseño y modelaje de imágenes, donde lógicamente se involucran gráficos y con la finalidad de mejorar la calidad, se realiza un proceso de renderizado, demandando prácticamente la totalidad del porcentaje de tiempo procesador (100%), tal situación puede prolongarse desde un par de minutos hasta incluso días y aún cuando un equipo de cómputo cuente con dispositivos auxiliares como son las tarjetas gráficas, usadas para aligerar la carga del procesador, es incuestionable que se requerirá en gran magnitud de la atención del CPU.

Adicionalmente existe un riesgo que se deriva de la situación mencionada, la ejecución de un proceso durante tiempo prolongado genera calentamiento en la unidad de procesamiento, lo que podría provocar un daño de consideración en el circuito o en algún otro componente del sistema. La problemática se centra en evitar que un proceso de usuario durante su ciclo de ejecución provoque una falla en el hardware, por lo que un oportuno control en el cambio de ejecución de los procesos podría evitar una avería en el sistema y al reducir

la carga de trabajo del procesador se logra una estrategia de ahorro de energía, desafortunadamente no existe una alternativa que resulte práctica de aplicar por parte de un usuario con o sin privilegios (Perry, Porter, & Votta, 2000).

El presente trabajo pretende darle solución a la problemática de reasignación de procesos para hacer eficiente el uso energético de la unidad de procesamiento y a su vez aliviar el calentamiento derivado de un procesamiento excesivo, al enfocarse en realizar un control en la ejecución de los procesos de usuario, con base en un monitoreo constante del sistema, con la finalidad de detectar a algún proceso cuyo tiempo de ejecución se extienda por un lapso considerable, representando una amenaza para el sistema, por lo que se procederá a otorgar de forma temporal privilegios, con la intención de poder reasignar y reducir el porcentaje de uso de CPU (Microsoft, Overview of Windows performance monitor; Brown y G. Comunity, 2009).

Estructura del documento

La siguiente sección presenta los trabajos relacionados con la problemática, así como la forma en que lo resuelve. Inmediatamente después se abordara la propuesta que sustenta este trabajo de investigación, explicando la arquitectura, funcionamiento general, así como los principales componentes del prototipo MALABARES. En seguida se encuentra la sección de pruebas, en la cual se plantea un escenario y cómo se hizo frente para darle solución, finalmente en el apartado de conclusiones, se presentan las aportaciones, limitaciones y se plantea la factibilidad de trabajos e investigación en un futuro.

Trabajos relacionados

En la presente sección se abordan posibles soluciones a la cuestión descrita en el párrafo anterior. En primer lugar es necesario mencionar que el hardware a manera de protección, al detectar un incremento considerable en la temperatura del sistema, simplemente se reinicia, dando un respiro a los componentes del equipo pero con la enorme desventaja de perder el trabajo que se estaba ejecutando.

Soporte del Sistema Operativo

Una alternativa que tiene un usuario con privilegios básicos, es que los sistemas operativos más conocidos comercialmente como Linux y Windows, cuentan con la posibilidad de cambiar algunos parámetros previamente establecidos por el sistema, aplicando comandos o bien por medio de herramientas gráficas⁴. Con su uso es factible hacer cambios en la prioridad de la ejecución de los procesos, ejemplo, Alta, Normal y Baja, sin embargo el resultado es poco significativo ya que un proceso que pasa de prioridad alta a baja, tan sólo disminuye en un par de puntos porcentuales en cuanto al uso del procesador, es decir si el proceso hacía uso de CPU del 100% con dicho cambio solamente bajaría a 99% o quizá 98% a lo sumo, lo cual por supuesto no reflejaría un gran avance en cuanto a la ejecución de la aplicación o proceso que se pretende controlar, de igual forma ocurre al intentar priorizar la ejecución de dicha tarea (Pery, Porter, & Votta, 2000; Microsoft, Administrador de tareas de Windows). Una desventaja adicional es que no todas las herramientas de software permiten realizar cambios de los parámetros, por lo que sólo en algunos casos será posible influir sobre el sistema, mediante la modificación de los valores y porcentajes asignados a cada proceso (Perry, Porter, & Votta, 2000; Microsoft, Administrador de tareas de Windows; Red hat enterprise linux 3).

Ambientes virtualizados

VMWare DRS es una aplicación desarrollada por VMware; entre los aspectos más relevantes que presenta, está el hecho de realizar un equilibrio de los recursos existentes en hardware, así como de llevar a cabo la asignación de los mismos de forma dinámica y repartirlos entre un grupo de recursos lógicos (máquinas virtuales). También, es responsable de realizar un monitoreo constante de los recursos, con la intención de asignarlos a las máquinas virtuales a

4 Microsoft, Administrador de tareas de windows, <http://support.microsoft.com/kb/323527/es> Linux commands. <http://www.linux.org/>
Red hat enterprise linux 3: Introduction to system administration, chapter 2. Resource monitoring. https://access.redhat.com/site/documentation/en-/Red_Hat_Enterprise_Linux/3/html/Introduction_to_System_Administration/s1-resource-what-to-monitor.html

fin de cubrir la demanda hecha por los procesos que se ejecutan en cada una de las instancias virtuales (V. Community, 2006). Esta herramienta da la flexibilidad de que los usuarios puedan definir sus propias reglas y políticas acerca de la compartición de recursos a través de las máquinas virtuales, además de priorizar el uso que tendrán entre las diferentes máquinas virtuales (V. Community, 2006, 2008). Si bien se puede priorizar a los recursos, con lo que se benefician a las aplicaciones consideradas más importantes, no existe una posibilidad para el caso contrario, es decir, reducir la atención del procesador hacia un proceso, finalmente, quizá la mayor desventaja de ésta propuesta radica en el hecho que sólo es posible gestionar los recursos en su ambiente virtual.

Por último, citaremos un tercer caso, es un trabajo que hace un control adaptativo de recursos virtualizados en ambientes computacionales (*adaptive control of virtualized resources in utility computing environments*), La propuesta se centra en el caso del funcionamiento de un centro de datos, donde múltiples aplicaciones de varios niveles comparten un repositorio común de recursos del servidor y los niveles para cada aplicación se encuentran alojados en una máquina virtual. La propuesta explota las ventajas brindadas por las técnicas de virtualización, para ello usa al hipervisor de la máquina virtual para determinar el consumo de uso del procesador por parte de cada una de las máquinas virtuales y dicha información será transmitida al sistema de control (controlador de recursos), responsable de ajustar dinámicamente las acciones para compartir los recursos para los niveles individuales, cuidando los parámetros de calidad de servicio de la aplicación (Padala, Pradeep, Shin, Kang, G., et al, 2007) La principal ventaja de esta propuesta es que su comportamiento se basa en técnicas para realizar los ajustes de adaptabilidad en un eficiente y constante monitoreo del sistema, poniendo especial atención en los recursos cuya demanda es variante. Sin embargo se tiene nuevamente la desventaja de que su campo de acción es exclusivo de entornos virtuales y el hecho de que en ningún momento se hace aborda la posibilidad de beneficiar o restringir alguna tarea con un recurso en particular, ya que precisamente carece de un nivel alto en prioridad.

Propuesta-El prototipo MALABARES

MALABARES se diseñó con la intención de realizar pruebas relacionadas con esta investigación y toma su nombre del típico espectáculo que se puede

presenciar en un circo y que consiste en la habilidad de equilibrar diferentes objetos lanzándolos por el aire y recuperándolos sin que caigan al suelo. La analogía con la propuesta presentada en este trabajo radica en el hecho de que su principal característica es la de intentar distribuir recursos críticos entre los procesos de usuario.

Arquitectura

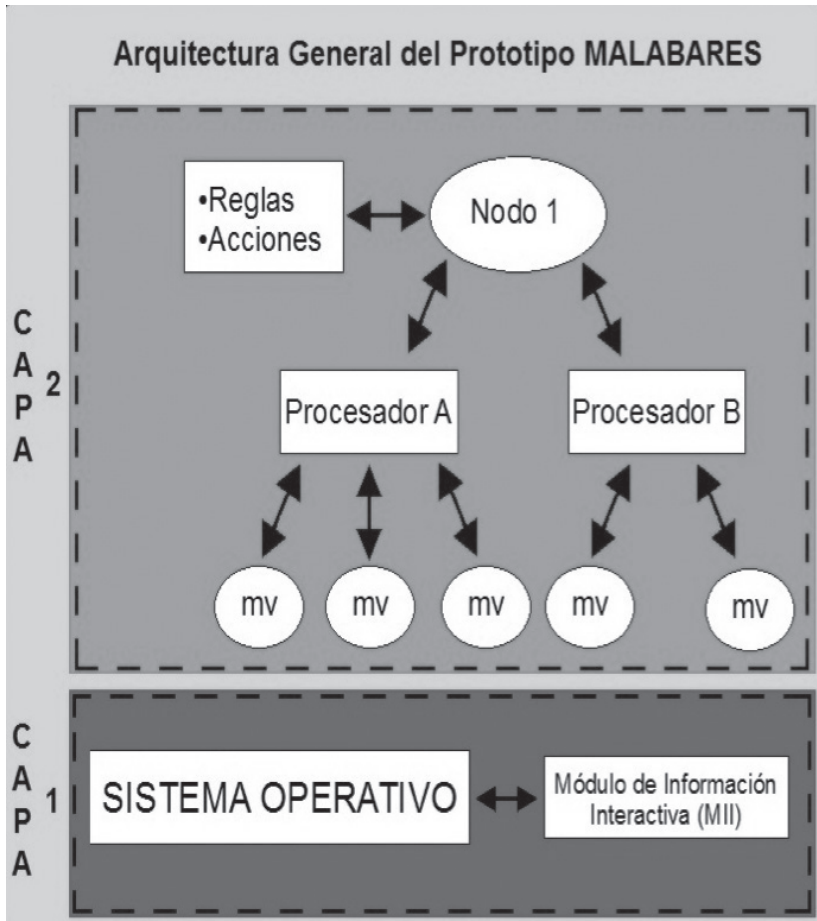
En las siguientes secciones se describirán los principales elementos que conforman la arquitectura del prototipo.

MALABARES se puede definir como un sistema con características adaptativas [16, 17], ya que realiza un monitoreo frecuente del estado del sistema, por lo que una parte fundamental del funcionamiento del sistema se centra en realizar una revisión constante, para conocer el estado que guardan los procesos que son ejecutados, determinando el consumo y la demanda de los recursos, lo que incluye también la detección de cambios en el contexto de la ejecución, esta tarea es realizada por un componente de MALABARES llamado Módulo de Información Interactiva (MII), el cual mantendrá una frecuente comunicación con el sistema operativo residente con la finalidad de obtener los datos necesarios. Un esquema general de la arquitectura de MALABARES se detalla en la figura 1.

Módulo de Información Interactiva (MII)

Su labor primordial es monitorear el estado del sistema, recopilando información sobre los procesos que residen en el sistema, por lo que su trabajo inicia estableciendo comunicación con el sistema operativo, con la finalidad de obtener dichos datos. Esto se logra por medio de un programa codificado en C, que se auxilia de instrucciones contenidas en el API Win32. Los datos recabados son útiles para que el módulo monitor realice su función. En la figura 2, se esquematiza la función de este módulo.

Figura 1
Esquema del funcionamiento general
del prototipo MALABARES



Monitor-MALABARES

Este componente es probablemente la pieza más importante del prototipo, se puede considerar como el núcleo de MALABARES. Su labor consiste en

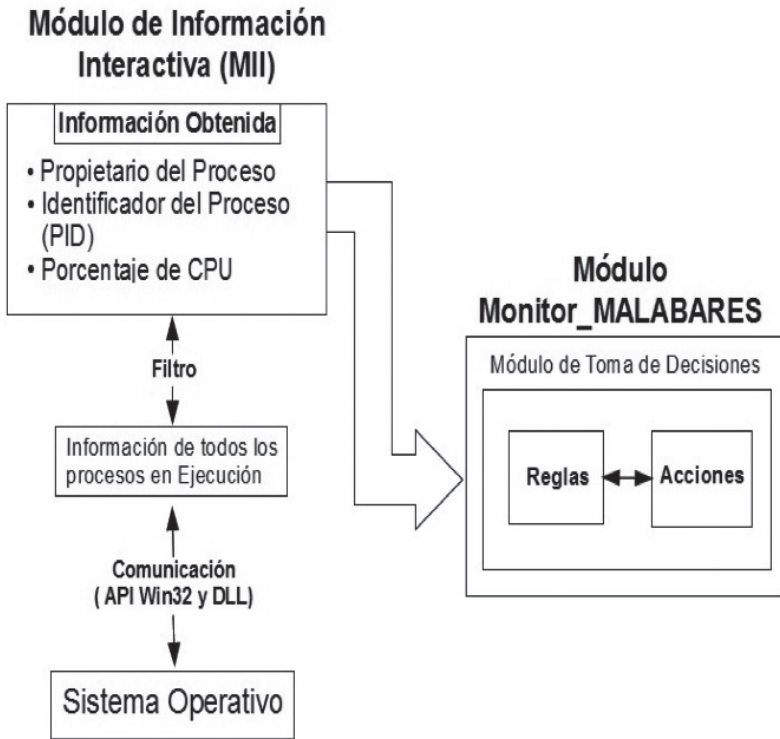
recopilar la información proporcionada por el MII, lo que incluye una estrategia para hacer frente a las situaciones que indican que algún proceso perteneciente a un usuario común (sin privilegios) de entre todos los que son administrados de forma local, está haciendo una demanda excesiva por algún recurso, lo que sin duda pondría al Monitor-MALABARES en un estado de atención ya que puede darse la situación, donde sea indispensable que otro(s) procesos requieran hacer uso del mismo recurso.

La pieza responsable de atender las situaciones consideradas como especiales en el sistema se denomina módulo de Toma de Decisiones (MTD), se compone de dos partes: reglas y acciones. Las reglas, trabajan con la información proporcionada por la MII y opera con base en un conjunto de criterios establecidos para el manejo de los recursos críticos. Cuando una regla es aplicada, de forma inmediata se emite una acción, la cual ejecuta una instrucción que realiza los cambios solicitados, ejemplo, modificar algunos de los valores del proceso que fueron preestablecidos por el sistema operativo, como es el porcentaje de CPU asignado.

MALABARES como un sistema con características adaptativas

Entre las principales características de un sistema adaptativo, se encuentra la cualidad de detectar cambios en el contexto de la ejecución y con base en ello tiene la capacidad de acoplarse lo mejor posible a las nuevas circunstancias. La habilidad de ajustarse a tales alteraciones, tiene como fundamento el hecho de que la reacción presentada a los cambios depende directamente del conocimiento que se tenga de su ambiente, por lo cual puede deducirse que al establecer un proceso eficiente y constante de monitoreo, será posible usar la información para aplicar criterios que modifiquen el comportamiento del sistema (David, 2005; Padala, Pradeep, Shin, Kang, G., et al, 2007; Sodan, 2009). MALABARES adquiere esta cualidad al estar solicitando frecuentemente información al sistema operativo, sobre el estado de los procesos en ejecución.

Figura 2
Módulo de MALABARES



Control del porcentaje de uso del CPU

Mediante técnicas de programación, es viable modificar el orden de ejecución establecido por el planificador de tareas del sistema operativo, forzando a un proceso a renunciar a la atención del procesador por intervalos de tiempo, reduciendo el riesgo de una falla en los componentes del sistema. Es importante resaltar, de que el procedimiento usado no se fundamenta en el hecho de que los procesos son forzados a entrar en un ciclo de espera obligatoria (dormir), ya que el resultado de aplicar ésta opción es el desperdicio de miles de ciclos máquina, es decir la tarea sigue en ejecución, pero el procesador no realiza ac-

ción alguna, afectando el rendimiento general del sistema (Tanenbaum, 2008, Dhamdhere, 2012). Por otra parte, nuestra propuesta consiste en retirar de forma momentánea al proceso de ejecución y una vez que se haya cumplido cierta condición estará listo para recibir nuevamente la atención del procesador.

Con base en la siguiente fórmula es factible realizar el cálculo correspondiente a la cantidad de porcentaje de CPU usado por los procesos de usuario y a su vez determinar el lapso de tiempo en que el proceso no tendrá la atención del procesador.

$$\text{Porcentaje_CPU} = \frac{\text{TotalTimeProc} * 100}{\text{TotalTimeProcSys}}$$

Dónde:

TotalTimeProc= Tiempo total de procesamiento (todos los procesos en ejecución son considerados).

TotalTimeProcSys= Tiempo total usado por los procesos del sistema.

Pruebas (Escenario)

Para la descripción de la situación, retomaremos algunos de los aspectos mencionados en los primeros párrafos de este documento, en particular el hecho de que existen aplicaciones que solicitan una alta demanda de tiempo procesador, las cuales en algunos casos se relacionan con el modelaje de imágenes y el proceso para mejorar la calidad de las mismas (renderizado), prolongando su ejecución por períodos que pueden resultar muy variables. Aún cuando un equipo de cómputo cuente con dispositivos auxiliares como son las tarjetas gráficas, usadas para aligerar la carga del procesador, es incuestionable que se requerirá en gran magnitud de la atención del CPU (Buyya, Abramson, & Giddy, 2000). Sin embargo, existe un riesgo adicional que con lleva ésta situación, la ejecución de un proceso durante un tiempo prolongado genera un sobrecalentamiento en el procesador, lo que podría provocar un daño de consideración en el circuito o en algún otro componente del sistema.

Realizar esta prueba implicó generar intencionalmente una mayor exigencia para el procesador, por lo que se ejecuto una aplicación llamada *google sket-*

chup 8, usada para el diseño. Una de las opciones contenidas en el programa permite renderizar una imagen y como se ha hecho mención anteriormente, éste tipo de aplicaciones requiere de un gran poder de cómputo.

La prueba se compone de tres diferentes etapas, en la fase inicial se realizó un monitoreo del porcentaje de uso del procesador (CPU), el cual se encontraba en cero ya que no había demanda por parte de ninguna aplicación, usando el software PC Health Monitor de Toshiba, se revisó la temperatura del procesador (T_Proc) la cual se encontraba en 41% y finalmente se revisó la velocidad del ventilador (Vvent). En la segunda etapa se procedió a ejecutar la aplicación y de forma casi inmediata los parámetros monitoreados se incrementaron de forma considerable, todos los valores correspondientes a cada una de las etapas se muestran en la tabla 1.

Tabla 1
Parámetros monitoreados

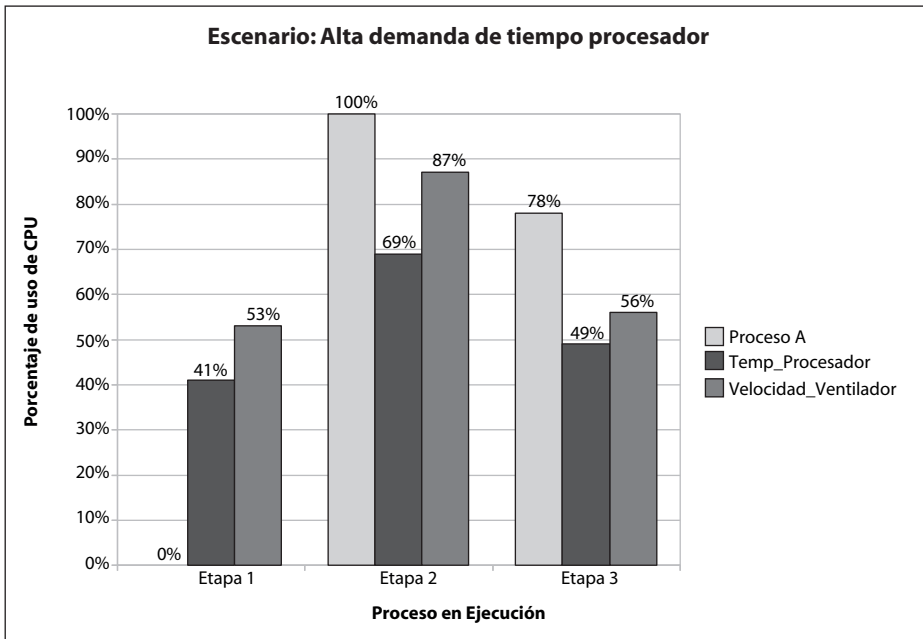
Parámetros	Etapas 1	Etapas 2	Etapas 3
CPU	0 %	100%	78%
T_Proc	41%	69%	49%
Vvent	53%	87%	56%

El prototipo MALABARES funciona con base en reglas y acciones, por lo que a continuación se describen brevemente en qué consisten:

Regla: Al monitorear un proceso que lleva más de diez minutos en ejecución demandando el cien por ciento del procesador, se establece una alerta.

Acción: MALABARES toma la decisión de que al proceso detectado se le modificara el porcentaje de CPU asignado, reduciéndolo durante intervalos de un minuto por cada diez de ejecución, la consecuencia es que la tarea continuará hasta llegar a un buen término sin poner en riesgo a los componentes del equipo de cómputo. Los valores de la etapa final, fueron obtenidos después de aplicar la acción, con lo que se logró reducir los parámetros de la temperatura del procesador así como la velocidad del ventilador, evitando un daño en dichos componentes. El resumen de cada una de las fases se muestra en la gráfica de la figura 3.

Figura 3
Gráfica de los valores obtenidos en la prueba



Finalmente mencionaremos que las características de esta misma prueba se replicaron en un ambiente virtualizado usando el software de VirtualBox, se hicieron algunas pequeñas variantes como el hecho de ejecutar Windows Server 2008 como sistema operativo residente, obteniendo en todas las ocasiones el mismo resultado, disminuir la temperatura del procesador.

Descripción del equipo usado para las pruebas

En la fase de pruebas se usaron tres tipos diferentes de equipos de cómputo con variantes en cuanto a sus capacidades (arquitectura), configuraciones y velocidad de procesamiento; entre ellas se incluyen máquinas con un solo procesador y modelos de Dual Core, los tres tipos de equipos únicamente eran diferentes en cuanto a hardware, el software usado es el mismo en todos los casos.

a) Hardware

Equipo 1: AMD Athlon II P320 Dual Core. 4 Gb en RAM.

Equipo 2: Intel Core i3. 4 Gb en RAM.

Equipo 3: Intel Core i7. 8 Gb en RAM.

b) Software

Se usaron tres ediciones diferentes de la familia de Microsoft Windows: Windows XP, Windows professional 7 32 bit y Windows Server 2008 SP2 (x86-x64).

c) Lenguajes de programación

Para la implementación del prototipo se recurrió a tres diferentes lenguajes de programación, todos ellos con la característica de ser software libre o haber sido liberado con fines académicos: Borland C 5.5. y Rutinas en ensamblador.

d) Entorno virtual

Además se instalaron en momentos no consecutivos cronológicamente dos versiones diferentes del software VirtualBox, para hacer pruebas que involucran a los ambientes virtualizados: VirtualBox 4.1.8 y VirtualBox 4.3.6

e) Herramientas de monitoreo

Con la finalidad de comprobar y cotejar que los valores mostrados por MALABARES sean correctos se usaron dos de las herramientas que forman parte del sistema operativo Windows: Administrador de Tareas, Monitor de Recursos y PC Health Monitor.

Conclusiones

Como parte de los resultados obtenidos en el presente trabajo de investigación, se diseñó e implementó una estrategia que basa su funcionamiento en la

modificación del porcentaje de CPU asignado a los procesos de usuario, permitiendo que la ejecución de una aplicación o proceso, se limite o se favorezca, con lo cual se instauró un control sobre la carga de trabajo del procesador, evitando el sobrecalentamiento del mismo, previniendo con ello un posible daño que incluso pudiese afectar a otros componentes del sistema.

Sin embargo, como principal limitación del prototipo es que al momento su entorno de ejecución se limita solamente a la familia de los sistemas operativos Windows. Por otra parte se tiene una restricción más, en la prueba se logró de forma inmediata disminuir en un 20% la temperatura del procesador, pero no se ha profundizado en obtener una medición sobre cantidades de ahorro de energía, por lo que se puede considerar en una etapa posterior.

Trabajo futuro

Un reto muy importante para ampliar los horizontes del prototipo consistiría en lograr la implementación de MALABARES en sistemas operativos compatibles con Unix, aunque existen similitudes con Windows el manejo y obtención de información de los procesos es diferente a como actualmente está implementado.

Otro aspecto a considerar es que con base en la disminución de tiempo CPU, es factible gestionar otros recursos del sistema, ejemplo, se puede evitar la saturación del canal de ancho de banda al establecer un control en los procesos de usuario que realizan una gran cantidad de operaciones de Entrada/Salida, reduciendo el flujo de datos enviado a otras máquinas conectadas en red.

La implementación de la estrategia para disminuir la temperatura del CPU por medio de la descarga de trabajo del procesador, se puede considerar como punto de partida para nuevas investigaciones en diferentes áreas que involucren conceptos como el balanceo de cargas o variantes para disminuir el consumo de energía (*Green computing*).

Por último se mencionará un caso de estudio que pudiese compaginarse favorablemente con el prototipo descrito en este trabajo de investigación⁵. Intel

5 Intel-galileo. <http://arduino.cc/en/ArduinoCertified/IntelGalileo>
Intel-galileo. <http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html>.

Galileo es esencialmente un micro-controlador multipropósito desarrollado por la compañía Intel basado en el procesador de aplicaciones de 32-bits, *Quark SoC X1000*, el cual es compatible con la arquitectura X86, lo que permite una gran flexibilidad en términos de integración con máquinas que comparten esta tecnología. El principal beneficio de implementar MALABARES sobre Galileo es que se pudiesen administrar aplicaciones cuya naturaleza requiera de un control durante su ejecución. Si partimos del hecho de que se tendría una unidad de procesamiento, también sería posible lograr el dominio de los componentes involucrados con la intención de optimizar el consumo de energía.

Bibliografía

- Brown, D., y G. Community
2009 *Evaluating performance*. Technical report, GemFire Enterprise.
- Buyya Rajkumar, Abramson, David, & Giddy, Jonathan
2000 Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid. *Proceedings. The Fourth International Conference/ Exhibition*, 283-289 vol.1. High Performance Computing in the Asia-Pacific Region, IEEE Xplore, 2000.
- David, P.C.
2005 *Développement de composants Fractal adaptatifs: un langage dédié à l'aspect d'adaptation* (PhD thesis, École Mines de Nantes).
- Dhamdhere, D. M.
2012 *Sistemas operativos, un enfoque basado en conceptos*. 2da. Edición, Mc Graw Hill.
- Intel-galileo. <http://arduino.cc/en/ArduinoCertified/IntelGalileo>.
- Intel-galileo. <http://www.intel.com/content/www/us/en/do-it-yourself/galileo-maker-quark-board.html>.
- Linux commands. <http://www.linux.org/>
- Microsoft Administrador de tareas de windows, <http://support.microsoft.com/kb/323527/es>.
- Microsoft, Overview of windows performance monitor, <http://technet.microsoft.com/en-us/library/cc749154.aspx>
- Padala, Pradeep, Shin, Kang, G., et al.
2007 Adaptive control of virtualized resources in utility computing environments. *Proceeding EuroSys '07 Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems*, 289-302. ACM.

- Perry, D. E., Porter, A. A., & Votta, L. G.
2000 Empirical studies of software engineering: a roadmap. *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, 345-355, New York: ACM.
- Red hat enterprise linux 3: Introduction to system administration, chapter 2. Resource monitoring. [https://access.redhat.com/site/documentation/en-/ Red_Hat_Enterprise_Linux/3/html/Introduction_to_System_Administration/s1-resource-what-to-monitor.html](https://access.redhat.com/site/documentation/en-/Red_Hat_Enterprise_Linux/3/html/Introduction_to_System_Administration/s1-resource-what-to-monitor.html)
- Sodan, Angela C.
2009 Adaptive scheduling for qos virtual machines under different resource availability, performance effects and predictability. *Lecture Notes in Computer Science* 5798, 259-279. Springer verlag.
- Stallings, W.
2005 *Operating Systems: Internals and Design Principles*. 5ta. Edición, Wiley.
2006 *Organización y arquitectura de computadores*. 5ta. Edición, Prentice Hall.
- Tanenbaum, A. S.
2001 *Organización de computadoras: un enfoque estructurado*. 4ta Edición, Pearson.
2007 *Distributed Systems: Principles and Paradigms*. 2da. Edición, Prentice Hall.
2008 *Modern Operating Systems*. 3era. Edición, Prentice Hall.
- V. Community
2006 *Vmware infrastructure and architecture*. Technical report, EMC Corporation.
2008 *Vmware drs (distributed resource scheduler)*. Technical report, EMC Corporation.

Fecha de recepción: diciembre 5/2014; fecha de aceptación: enero 6/2015