Sincronización de Relojes en Ambientes Distribuidos

Fernando L. Romero, Walter Aróztegui, Fernando G. Tinetti[1]

Instituto de Investigación en Informática LIDI (III-LIDI)
Facultad de Informática – UNLP

Centro de Técnicas Analógico-Digitales (CeTAD)
Facultad de Ingeniería - UNLP
fromero@lidi.info.unlp.edu.ar, waroz@graffiti.net, fernando@info.unlp.edu.ar

CONTEXTO

Esta línea de Investigación forma parte de dos de los Subproyectos dentro del Proyecto "Sistemas Distribuidos y Paralelos" acreditado por la UNLP y de proyectos específicos apoyados por CyTED, CIC, Agencia e IBM.

RESUMEN

Esta línea de investigación se orienta a resolver el problema de sincronización de tiempo en ambientes distribuidos. El objetivo inicial de la sincronización de relojes es la estimación de rendimiento a partir de experimentos con la mínima instrumentación (y su consecuente interferencia) posible.

El algoritmo básico sigue las estrategias clásicas de sincronización de tiempo en ambientes distribuidos [1] [3] [11]. Sin embargo, se adaptan (al menos en principio) al entorno de un cluster o, al menos, de una red de interconexión sobre la que se tiene acceso exclusivo (o al menos controlado) para todas las comunicaciones entre las computadoras que se sincronizan. Este ambiente es específicamente el de los entornos de cómputo paralelo en clusters.

Keywords: *Sincronización de Procesos, Relojes Distribuidos, Rendimiento e Instrumentación, Sistemas Paralelos y Distribuidos, Paralelismo en Clusters e Interclusters, Sincronización Interna y Externa* .

1. INTRODUCCION

Siempre ha sido necesario disponer de un sistema capaz de proporcionar una referencia de tiempo en los sistemas de cómputo [19] [21]. Dicha referencia es esencial para resolver problemas tales como el ordenamiento de eventos (ej: envío y recepción de correo electrónico, eventos dentro de las transacciones, inicio de procesos en tiempo real, etc.).

También cobra importancia la medición de tiempos en la optimización del rendimiento tanto en sistemas de cómputo monoprocesador como en sistemas paralelos y distribuidos [2] [4] [7] [12] [13]. En todos los casos, las aplicaciones con fuertes requerimientos de cómputo o procesamiento son las que también requieren la optimización para el máximo aprovechamiento del hardware disponible. La relación es bastante directa: a partir de la monitorización de los tiempos de ejecución se pueden analizar los problemas de rendimiento e intentar solucionarlos [9].

Aún en el caso de las mediciones en una misma computadora (usualmente en el contexto de un sistema con un único procesador), es deseable que el registro de tiempos no influya en el tiempo de ejecución de la misma. En todos los casos, se necesitan resoluciones de reloj acordes a los tiempos que se deben medir en las aplicaciones.

[1] Investigador Asistente CICPBA

Usualmente, los métodos provistos por el sistema operativo no son apropiados [20] para todos los casos. Por otro lado, los métodos y/o herramientas provistas por los lenguajes dependen del sistema operativo y, por lo tanto, resultan inadecuados también.

En el caso de procesamiento distribuido, con un programa que ejecuta procesos en diferentes computadoras o en los que el tiempo de las comunicaciones es importante, la tarea de medir intervalos de tiempo conlleva la necesidad de sincronizar los relojes de las diferentes computadoras que se utilizan [5] [16] [17]. Sería deseable que esta tarea de sincronización se lleve a cabo fuera del tiempo en que se ejecute el programa que se está monitorizando, y conociendo el tipo (o al menos magnitud) de error con que se sincroniza.

Por otro lado, es deseable que dicha sincronización se lleve a cabo sin la necesidad de incluir hardware adicional al del sistema. Esto implica que todo lo referente a las comunicaciones deberá utilizar la red de interconexión entre computadoras existente y el sistema de medición existente en cada sistema de cómputo.

Básicamente, se desea contar con una herramienta de instrumentación para programas paralelos que:

Pueda ser usada inicialmente en un cluster de PC's, con la posibilidad de ser extendido a clusters en general y luego en plataformas distribuidas aún más generales.

Sea de alta resolución, es decir que se pueda utilizar para medir tiempos cortos, del orden de microsegundos.

Que no altere el funcionamiento de la aplicación bajo prueba, o que la alteración sea mínima y conocida por la aplicación.

Utilice en forma predecible la red de interconexión. Más específicamente, se puedan determinar, desde la aplicación, los intervalos de tiempo en los cuales se utilizará la red. De esta forma, se puede *desacoplar* el uso de la red de interconexión, ya que habrá intervalos de tiempo usados para la sincronización e intervalos de tiempo utilizados para la ejecución de programas paralelos.

2. LINEAS DE INVESTIGACION Y DESARROLLO

Inicialmente, se estudian tanto los algoritmos básicos como las implementaciones existentes. En este sentido, se cuenta con una amplia cantidad de información tanto de los algoritmos como de las implementaciones. Como requisito previo, normalmente se establece que cada computadora cuente con un oscilador físico de frecuencia más o menos constante. A partir de este oscilador físico se derivan los relojes lógicos que son los que se sincronizan [8] [10] [15]. En todos los casos, lo que se tiende a resolver son las diferencias de (Fig. 1):

1. Referencia fija en el tiempo a partir de la cual se contabiliza el tiempo en cada computadora. Aunque normalmente es constante, es relativamente difícil establecer con precisión su valor.
2. Frecuencia entre los relojes de las computadoras que se sincronizan. En algunos casos, se suelen incluir en este punto las diferencias de las variaciones de los relojes, dado que los relojes no necesariamente tienen frecuencia constante a lo largo del tiempo.

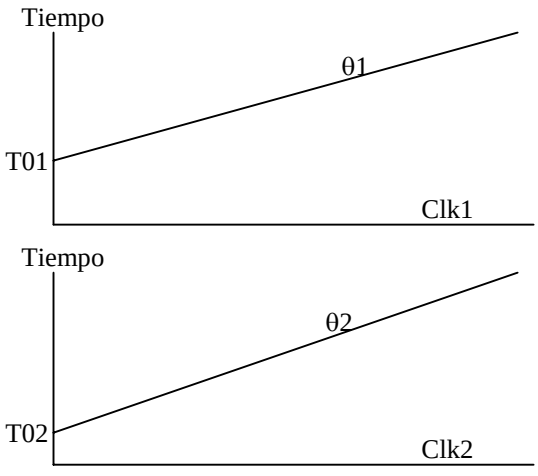

Figura 1: Relojes de Dos Computadoras.

La Fig. 1 muestra dos computadoras, cada una de ellas con su oscilador, que a su vez tiene sus propias características. El instante de tiempo T01 es la referencia al tiempo real en una computadora y T02 es el equivalente en la otra computadora. La pendiente θ1 corresponde al oscilador de una computadora y θ2 es la correspondiente pendiente de la otra.

Una de las formas más sencillas e intuitivas de sincronizar dos computadoras consiste en determinar el reloj de una de ellas en función del reloj de la otra teniendo en cuenta algún tipo de interconexión entre ellas. La Fig. 2 muestra esquemáticamente esta forma de sincronización, donde el reloj de una computadora se puede establecer a partir del reloj de la otra teniendo en cuenta el tiempo de las comunicaciones de sincronización entre las dos.

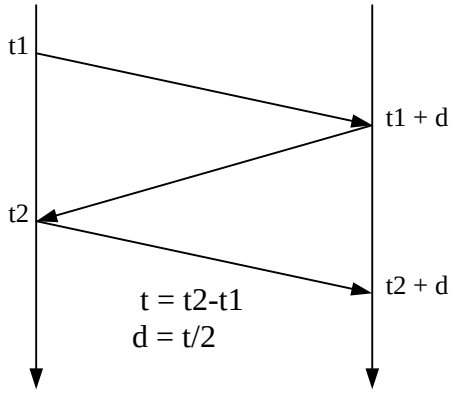

Figura 2: Sincronización Básica.

Sin embargo, como paso previo se deben investigar las formas de hacer referencias al reloj físico y/o a los relojes lógicos con la mínima sobrecarga. La idea básica para disminuir la sobrecarga normalmente se basa en evitar llamadas al sistema operativo para, al menos, evitar el consiguiente cambio de contexto. También es necesario investigar en particular hasta qué punto las referencias a los relojes/osciladores físicos son portables o al menos tienen funciones u operaciones equivalentes en diferentes plataformas de hardware.

Paralelamente a lo anterior, al menos en principio, es necesario analizar y, más específicamente cuantificar las características de NTP (Network Time Protocol). Esta cuantificación se refiere principalmente a:

- Resolución posible del reloj sincronizado. Aunque NTP es paramétrico esto no significa *a priori* que se puede obtener una resolución arbitraria cualquiera.
- Relación entre la resolución definida y la sobrecarga en la red de comunicaciones.
- Relación entre la resolución definida y la sobrecarga en la pila de protocolos TCP/UDP/IP.
- Relación entre la resolución definida y la sobrecarga en el uso de CPU y la jerarquía de memoria.

Todo este análisis es básicamente experimental [14] [17] [18], dado que el funcionamiento de NTP se define por un lado paramétricamente y por el otro con el comportamiento o rendimiento de la red de interconexión.

Una vez realizada la sincronización mínima entre las computadoras se debe investigar el comportamiento de la misma en términos de escalabilidad. Usualmente la sincronización se da entre dos máquinas y en el caso particular de NTP se lleva a cabo con el modelo cliente/servidor. Es claro que cualquier tipo de centralización (en el servidor, por ejemplo) tiene sus inconvenientes de escalabilidad y al menos debería ser posible su cuantificación. En este contexto específico es muy interesante la posibilidad de sincronización utilizando mensajes broadcasts con su consiguiente ahorro de comunicaciones punto a punto.

Como extensiones futuras, siempre es deseable la sincronización externa de los relojes [8]. Este paso está muy ligado también a la posibilidad de utilizar más de un cluster de computadoras para cómputo paralelo y en este contexto la sincronización de los relojes va más allá del análisis de rendimiento con el objetivo de optimizarlo. Las alternativas en este contexto están abiertas para varias posibilidades. Entre otras alternativas: una nivel jerárquico superior a to-

dos los clusters, tomar un cluster como el de mayor jerarquía, sincronizar con algún tipo de estrategia de tiempo real, etc.

3. RESULTADOS OBTENIDOS/ESPERADOS

Inicialmente, la idea es contar con una biblioteca mínima en cuanto a la medición de tiempos de cómputo en un mismo sistema (usualmente en una máquina monoprocesador). Ya se cuenta con una, para el sistema operativo Linux en PCs (procesadores Intel y compatibles) en las cuales se puede hacer referencia al contador de ciclos del oscilador. Entre las tareas pendientes se tienen la cuantificación de la sobrecarga y posibilidad de extensión a otros sistemas operativos y/o plataformas de hardware. En lo referente a la sobrecarga, los datos preliminares muestran resultados altamente satisfactorios, dado que la sobrecarga no excede las decenas de ciclos de reloj. En este sentido, ya es posible la instrumentación de código y el análisis de las aplicaciones que se resuelven en un mismo sistema o máquina. La precisión es satisfactoria: del orden de los microsegundos.

Se está desarrollando software (básicamente una biblioteca de una cantidad reducida de funciones) para instrumentación de programas paralelos que se ejecuten en clusters de PCs. Esta biblioteca tiene los lineamientos dados antes, con énfasis en la resolución del orden de los microsegundos, la mínima sobrecarga de procesamiento y el desacople de la red de interconexión.

Además, se están llevando a cabo pruebas sobre los sistemas de relojes distribuidos empleados en este momento tales como NTP (Network Time Protocol) y DTS (Distributed Time Service), con el fin de analizar sus ventajas y desventajas. También un análisis de los sistemas que utilizan hardware especializado, como para tener un marco de referencia más amplio [6]. En todos estos casos, la idea final es contar con un conjunto de programas del estilo de los *benchmarks* para que provean automáticamente la caracterización del sistema de sincronización elegido/utilizado. Como mínimo, el objetivo es contar con una metodología de caracterización de las herramientas, bibliotecas, y/o protocolos de sincronización que incluso pueda ser aplicada a la nueva biblioteca que se desarrolle, mencionada antes.

Como paso posterior, se espera extender la biblioteca para su uso en Internet y múltiples clusters para cómputo paralelo [22]. Quizás en este punto se deban redefinir algunas características de la herramienta o biblioteca, tal como la capa de transporte de los mensajes con los cuales se implementa la sincronización.

4. FORMACION DE RECURSOS HUMANOS

En esta línea de I/D existe cooperación a nivel nacional e internacional. Inicialmente se tiene una posible tesis de maestría y está abierta la posibilidad para varias Tesinas de Grado de Licenciatura.

5. BIBLIOGRAFIA

[1] Coulouris G., Dollimore J., Kinberg T., “Sistemas Distribuidos. Conceptos y Diseño”, 3ª edición. Pearson Educación, 2001. ISBN: 84-7829-049-4.

[2] Cristian F. "Synchronous and Asynchronous Group Communication", Comm. ACM, Vol.39, No. 4, April 1996, pp.88-97.

[3] Cristian F. "Probabilistic Clock Synchronization". *Distributed Computing*, 3: 146–158, 1989.

[4] Cristian F., C. Fetzer. "The Timed Asynchronous Distributed System Model" IEEE Transactions on Parallel and Distributed systems, June 1999, pp. 603-618.

[5] Cristian F., Fetzer C., "The Time Asyn-

chronous Distributed System Model", IEEE Transactions on Parallel Systems, June 1999, pp. 603-618.

[6] Elson K. J., Romer K., "Wireless Sensor Networks: A New Regime for Time Synchronization in Distributed Systems", Proceedings of the First Wirkshop on Hot Topics In Networks (HotNets-1), Princeton, New Jersey, October 2002.

[7] Elson K. J., Girod L., Estrin D., "Fine-Grained Network Time Synchronization using Reference Broadcasts", Proceedings of fifth symposium on Operating System Design and Implementation. December 2002.

[8] Fetzer C., Christian F., "Integrating External and Internal Clock Synchronization", June 1996.

[9] Grove D. A.. "Performance Modelling of message-passing parallel programs", May 2003.

[10] Kim K. H., Im C., Athreya P, "Realization of a Distributed OS Component for Internal Clock Synchronization in a LAN Environment".Proc. ISORC 2002, IEEE 5th Int'l Symp on Objet-oriented Real-time distributed Computing, Washington, D.C., April 2002, pp. 263-270.

[11] Mills D. L. "Measured performance of the Network Time Protocol in the Internet System". ACM Computer Communication Review 20, Jan. 1990. pp. 65-75.

[12] Mills D. L., "Modelling and analysis of computer network clocks", Electrical Engineering Department Report 92-5-2, University of Delaware, May 1992.

[13] Mills, D. L. "Improved algorithms for synchronizing computer network clocks", IEEE/ACM Transactions on Networks June 1995.

[14] Mills D. L. "Measured Performance of the Network Time Protocol in the Internet System", ACM Computer Review 20, 1, January 1990, pp.65-75.

[15] Mills D.L., "Internet time Synchronization: the Network Time Protocol", IEEE trans. Communications COM39, October 1991, pp. 1482-1493.

[16] Mills D.L., "Network Time Protocol (Version 3) specification, implementation and analysis", DARPA Networking Group Report RFC-1305, University od Delaware, March 1992.

[17] Mills D. L., "A Brief History of NTP Time: Confessions of an Internet Timekeeper". ACM Computer Communications Review 33, 2 (April 2003), pp 9-22.

[18] Mills, D.L. "Unix kernel modifications for precision time synchronization". Electrical Engineering Department Report 94-10-1, University of Delaware, October 1994.

[19] Mills, D.L, Kamp P.-H., "The Nanokernel", Proc. Precision Time and Time Interval (PTTI) Applications and Planning Meeting (Reston VA, November 2000).

[20] Rubini A., Corbet J., "Linux Device Drivers 2nd Edition" ISBN 0-59600-008-1. June 2001.

[21] Work P., Nguyen K., "Measure Code Sections Using The Enhanced Timer", http://www.intel.com.ar, October 2005.

[22] Zhao Y., Zhou W., Huang J, Yu S., "Self-Adaptive Clock Synchronization for Computational Grid", Journal of Computer Science and Technology, 2003 Volume: 18 Issue: 4 pp. 434 – 441.