

John A. Stankovic. University of Massachusetts

Misconceptions About Real-Time Computing: A Serious Problem for Next-Generation Systems

CARLOS MARTÍN FLORES GONZÁLEZ, Carné: 2015183528

Instituto Tecnológico de Costa Rica
Maestría en Computación
Sistemas Operativos Avanzados
Profesor: Francisco Torres Rojas, Ph.D

¿Qué es computación en tiempo real? En computación en tiempo real la correctitud del sistema depende no sólo de los resultados lógicos de la computación si también en el tiempo que los resultados fueron producidos. Ejemplos: sistemas para control de experimentos, control de motores de automóviles, plantas de energía, sistemas de control de vuelo, naves espaciales y robótica. Los sistemas en tiempo real más complejos son costosos de construir y son verificados por medio de técnicas *ah hoc* o con simulaciones costosas. Los diferentes componentes en un sistema son difíciles de integrar y esto agrega más costo al sistema. Dos grandes fuerzas están empujando a los sistemas en tiempo real a una nueva generación: la necesidad de inteligencia artificial y el rápido avance del hardware. Estas fuerzas están incrementando la dificultad de los problemas científicos y de ingeniería que afrontan los sistemas en tiempo real.

Conceptos erróneos comunes. **1. No hay ciencia en el diseño de sistemas en tiempo real:** Es cierto que el diseño de sistemas en tiempo real es mayoritariamente *ah hoc*. Esto no significa, sin embargo, que un enfoque científico no es posible. Investigaciones están iniciando con la inclusión de una métrica de tiempo para la especificación de un sistema y teoría semántica para lenguajes de programación en tiempo real. **2. Los avances en hardware de supercomputadoras van a hacerse cargo de los requerimientos de tiempo real:** avances en el diseño de supercomputadoras probablemente va a aprovecharse de procesadores en paralelo y de la mejora del *throughput* del sistema, pero esto no significa que las restricciones de tiempo se vayan a cumplir automáticamente. A menos que la arquitectura de la computadora sea modificada para coincidir con la aplicación, los procesadores y sus subsistemas de comunicación podrían no estar en la capacidad de manejar toda la carga de tareas y tráfico de tiempo crítico. De hecho, los problemas de *real-time task and-communication scheduling* probablemente serán peores conforme más hardware sea usado. **3. Computación en tiempo real es equivalente a computación rápida:** El objetivo de la computación rápida es la de minimizar el tiempo de respuesta promedio de un conjunto de tareas. Sin embargo, el objetivo en computación en tiempo real es cumplir con requisitos de tiempo individuales de cada tarea. En lugar de ser rápido¹, la propiedad más importante de un sistema en tiempo real debería de ser la predictibilidad, esto es, el comportamiento funcional y temporal debería ser tan determinístico como fuera necesario para satisfacer la especificación de un sistema. La computación rápida sirve de ayuda para cumplir con las especificaciones de tiempo, pero la computación rápida por sí sola no garantiza predictibilidad. Tal vez la mejor respuesta para aquellos quienes dicen que dicen que computación en tiempo real es equivalente a computación rápida es formulando la siguiente pregunta: Dado un conjunto de requisitos de

¹El cual, de todos modos es un término relativo.

tiempo real demandantes y una implementación usando el mejor hardware posible, cómo se puede mostrar que el comportamiento relacionados con el tiempo se logran? La respuesta no es “por medio de pruebas”. Se sabe que un procedimiento de verificación formal acoplado con pruebas podría mejorar substancialmente lo que se sabe en la actualidad. **4. La programación en tiempo real es codificación en ensamblador, programación de prioridad de interrupciones y escritura de controladores para dispositivos:** un objetivo primario de la investigación en tiempo real es de hecho automatizar, por medio de transformaciones de optimización y de teoría de *scheduling*, la síntesis de código altamente eficiente y *schedulers* de recursos personalizados a partir de especificaciones con restricciones temporales. Programación en lenguaje ensamblador, programación de interruptores y escritura de controladores aunque son aspectos de computación en tiempo real, no representan inquietudes científicas, solamente su automatización. **5. Investigación en sistemas en tiempo real es ingeniería del rendimiento:** Aunque muchos aspectos en la investigación de sistemas en tiempo real tienen que ver con rendimiento, un diseño apropiado de un sistema en tiempo real requiere soluciones a otras muchas preguntas interesantes: especificación y verificación del comportamiento temporal y semántica para lenguajes de programación que tome en cuenta tiempo. **6. Los problemas en sistemas en tiempo real han sido resueltos en otras áreas de la computación o investigación de operaciones:** existen problemas únicos en sistemas en tiempo real que no han sido resueltos en otras áreas. **7. No es significativo hablar acerca de garantizar el rendimiento en tiempo real porque no se puede garantizar que el hardware no va a fallar o que el software está libre de errores o que las condiciones operacionales reales no van a violar los límites del diseño especificado:** El hecho que el hardware/software no funcione correctamente o que las condiciones operacionales impuestas por el mundo externo puedan exceder los límites del diseño del sistema, no le da al diseñador licencia para incrementar las posibilidades de fallo al no intentar asignar los recursos cuidadosamente para que se puedan cumplir los requisitos de tiempo críticos. Ciertamente no se puede garantizar lo que está fuera de nuestro control pero se debe garantizar lo que se pueda. **8. Los sistemas en tiempo real funcionan en un ambiente estático:** un problema que experimenta la industria es cómo reconfigurar los sistemas para acomodar los cambios en los requerimientos y así crear una mínima interrupción en la operación que se lleva a cabo. No es común que algunos hardware de tiempo real lleven en el campo 15 o más años, de allí que, una metodología de diseño para tales sistemas no debe asumir un ambiente estático.

Los retos de la computación de tiempo real: **1. Especificación y verificación:** (1) el reto fundamental en la especificación y la verificación de sistemas en tiempo real es cómo incorporar la métrica de tiempo. Métodos deben ser buscados para incluir restricciones de tiempo en especificaciones y para establecer un sistema que satisfaga tales especificaciones. Los enfoques usuales para especificar el comportamiento de sistemas de computación implican enumeración de eventos o acciones en las que el sistema participa y describiendo los órdenes en que estos puedan ocurrir. No está claro como extender tales enfoques para incluir restricciones de tiempo. Tampoco está claro cómo extender la notación de la programación para permitir al programador a especificar cómputos que están restringidos por el tiempo. (2) Deducir las propiedades del sistema completo a partir de sus partes y la forma en que estas partes están combinadas, se debe caracterizar una forma de componer las restricciones de tiempo y las propiedades de las partes para sintetizarlos en un todo. (3) Otro problema que se enfrenta es el de tratar con la explosión de estados que se encuentran en técnicas de verificación. Cientos o a veces miles de estados son usualmente requeridos para expresar formalmente el estado de un sistema relativamente simple. **2. Teoría de scheduling en tiempo real:** satisfacer los requerimientos de tiempo en sistemas en tiempo real demanda la planificación de recursos del sistema de acuerdo a algún algoritmo bien conocido para que el comportamiento temporal del sistema sea entendible, predecible y mantenible. Los sistemas son altamente dinámicos, requieren de algoritmos de *scheduling* adaptativos. Tales algoritmos deben basarse en heurísticas, debido que estos problemas de *scheduling* son NP-hard. **3. Sistemas operativos de tiempo real:** el sistema operativo debe proporcionar soporte básico para garantizar las restricciones de tiempo real. Dado que un sistema

es distribuido, se enfrenta a un complicado problema de análisis temporal *end-to-end*: restricciones de tiempo son aplicadas a colecciones de tareas cooperativas y no solo a tareas individuales. Para desarrollar la próxima generación de sistemas operativos distribuidos en tiempo real adecuados para aplicaciones complicadas se requiere al menos de 3 innovaciones científicas: (1) la dimensión de tiempo debe ser elevada como un principio central del sistema. (2) Los paradigmas básicos que se encuentran en los sistemas operativos distribuidos actuales deben cambiar. (3) Un enfoque de asignación de recursos altamente integrado y restringido en tiempo. **4. Lenguajes de programación en tiempo real y metodologías:** (1) Soporte para gestión de tiempo. (1.1) Las características del lenguaje deberían de soportar la definición de restricciones de tiempo. (1.2) El ambiente de programación debería de proporcionar al programador con las primitivas para controlar y mantener registro de los recursos utilizados por los módulos de software. (1.3) Las características del lenguaje deberían de soportar el uso de algoritmos de *scheduling*. (2) Verificación de *scheduling*. Dada un conjunto de algoritmos de *scheduling* bien conocidos, el análisis de *schedulability* permitirá saber si requisitos de tiempo puede ser cumplidos. (3) Módulos de software de tiempo real reusables. (4) Soporte para programas distribuidos y tolerancia a fallos. **5. Bases de datos de tiempo real:** El reto fundamental de las bases de datos de tiempo real para ser la creación de una teoría unificada que proporcione un protocolo de control de concurrencia en tiempo real que maximice ambos la concurrencia y la utilización de los recursos con respecto a 3 restricciones al mismo tiempo: consistencia de datos, exactitud de las transacciones y *deadlines* de las transacciones. **6. Inteligencia Artificial:** la investigación en inteligencia artificial de tiempo real se centra en razonamientos sobre procesos restringidos en tiempo, usando conocimiento heurístico para planificar esos procesos. Una consideración clave en la resolución robusta de problemas es la de proporcionar la mejor solución dentro de una determinada restricción dinámica de tiempo. Mucho de los problemas actuales en sistemas en tiempo real son aplicables a tales sistemas. **7. Tolerancia a fallos:** ciertos problemas en investigación son importantes para lograr sistemas en tiempo real tolerantes a fallos y confiables: (7.1) La especificación formal del requerimiento de confiabilidad y el impacto en las restricciones de tiempo. (7.2) El manejo de errores es usualmente compuesto por una secuencia de pasos: detección del error, localización de la falla, reconfiguración del sistema y recuperación. Todos estos pasos deben ser diseñados y analizados en el contexto de rendimiento combinado y confiabilidad. (7.3) Los efectos de las cargas de trabajo en la tolerancia fallos no ha sido abordado adecuadamente. **Arquitectura de sistemas en tiempo real:** Muchos sistemas en tiempo real pueden ser vistos como una tubería con tres etapas: adquisición de datos de los sensores, procesamiento de datos y salida a los actuadores/monitores. La próxima generación de sistemas será usualmente distribuido tal que cada nodo pueda ser un multiprocesador. Una arquitectura de un sistema en tiempo real debe ser diseñada para soportar estos componentes con fidelidad. Algunos temas abiertos a investigación en arquitecturas en tiempo real son: (8.1) Topología de interconexión para procesadores e I/O. (8.2) Comunicaciones rápidas, confiables y restringidas en tiempo. (8.3) Soporte arquitectural para manejo de errores. (8.4) Soporte arquitectural para algoritmos de *scheduling*. (8.5) Soporte arquitectural para sistemas operativos de tiempo real. (8.6) Soporte arquitectural para características de los lenguajes en tiempo real. **9. Comunicaciones en tiempo real:**

1. ¿CUÁL ES EL PROBLEMA QUE PLANTEA EL *PAPER*?
2. ¿POR QUÉ EL PROBLEMA ES INTERESANTE O IMPORTANTE?
3. ¿QUÉ OTRAS SOLUCIONES SE HAN INTENTADO PARA RESOLVER ESTE PROBLEMA?
4. ¿CUÁL ES LA SOLUCIÓN PROPUESTA POR LOS AUTORES?
5. ¿QUÉ TAN EXITOSA ES ESTA SOLUCIÓN?