

Desarrollo de Sistemas de Scheduling para la Nube

Daniel Díaz, Francisco Ibañez, Sandra Oviedo, Nancy Alonso, Juan Manuel Cuneo

LISI- Instituto de Informática – Dpto. de Informática
FCEFYN - Universidad Nacional de San Juan
CUIM – Av. Ignacio de la Roza 590 (O), Rivadavia – J5402DCS San Juan
{ fibanez, ddiaz, soviedo, nalonso, jmcuneo}@iinfo.unsj.edu.ar}

Resumen

Los sistemas de scheduling son el corazón de todo sistema de planificación de la producción. Las nuevas tecnologías traen consigo amenazas y oportunidades para la industria, en este contexto, los sistemas de scheduling no escapan a esta realidad. El objetivo de este proyecto es investigar sobre métodos y técnicas en pos de integrar las tecnologías de cloud computing, web de tiempo real, software product lines y de los solucionadores para optimización en el desarrollo de sistemas de scheduling de producción.

Palabras clave:

Scheduling de producción, cloud computing, web de tiempo real

Contexto

Este trabajo se desarrolla en el marco del proyecto denominado “Desarrollo de sistemas de scheduling producción utilizando computación en la nube, web de tiempo real, y línea de productos de software”. La unidad ejecutora es el Laboratorio de Informática Aplicada a la Innovación del Instituto de informática de la Facultad de Ciencias Exactas Físicas y Naturales de la Universidad Nacional de San Juan.

Introducción

Entre las últimas tecnologías que se han instalado en el mercado, la computación en la nube (cloud computing) es una tecnología que está permitiendo que las pequeñas y medianas empresas utilicen los software que en la década pasada estaba solo permitidos, por sus costos de licenciamiento, a las grandes empresas. Las herramientas informáticas que permiten solucionar problemas de scheduling se denominan “optimization solvers”, son plataformas que permiten modelar, diseñar e implementar soluciones para problemas de optimización. Los problemas de scheduling son un caso particular de los problemas de optimización. Muchas de estas herramientas están desplegadas en la nube como servicios. Web de tiempo real se refiere a la capacidad que tiene una aplicación web de comunicar diferentes usuarios o dispositivos de manera tal que la información está disponible tan pronto es publicada por su autor, es decir no requiere que cada participante observe periódicamente la fuente para ver si existen información nueva o modificada.

Software product lines es una tecnología que promueve la reutilización del software para el desarrollo de familias

de productos, recientemente estas técnicas se están aplicando a la construcción de aplicaciones que se ejecutan en el modelo de cloud computing, dicha tecnologías se conocen como cloud-based software product lines.

El resultado de la integración de estas tecnologías debe permitir disponer de sistemas scheduling que se ejecuten en la nube, hagan uso de la web de tiempo real, y aprovechen todos los beneficios que una línea de producto de software brinda en cuanto a customización y reducción de costos de implementación y mantenimiento.

Sistemas de scheduling en dominios industriales

Los problemas de Scheduling aparecen en diferentes dominios, este proyecto trata con problemas de scheduling que surgen en el dominio industrial y que en la industria que se denominan *production scheduling* o *manufacturing scheduling*.

La definición más clásica de la palabra scheduling dice : “Scheduling es el problema de asignar recursos limitados a tareas en el tiempo con el objeto de optimizar uno o más objetivos” [1]. En la industria, un sistema de scheduling es el corazón del sistema de planificación y control de la producción. Normalmente el sistema de scheduling interactúa con otros sistemas de fabricación, tales como ERP (Enterprise Resource Planning) y MES (Manufacturing Execution System).

Existe una literatura muy extensa sobre scheduling y se ha escrito mucho sobre modelos y algoritmos que resuelven problemas de scheduling. Sin embargo, poco son los artículos que tratan de cómo traer estos modelos y algoritmos a implementaciones reales. Esto se conoce como el “hueco o gap” entre la teoría y la práctica de scheduling [2]. Para cerrar este hueco entre la teoría y la práctica los

modelos y algoritmos de scheduling se deben implementar en una pieza de software que satisfaga las necesidades que tiene una empresa en el área de scheduling. Esto implica llevar a cabo un proceso de desarrollo de software para obtener un producto final, es decir un sistema de scheduling y que ayude a la toma de decisión. Según Yen y Pinedo [3] un sistema de scheduling se compone tres módulos: (1) módulos de base datos y base de conocimiento, (2) módulos del motor de scheduling y (3) módulo de interface de usuario. La parte esencial de todo sistema de scheduling es el desarrollo del motor de scheduling, es donde se vinculan las necesidades que tiene la empresa con los modelos y algoritmos de scheduling.

Debido a los avances tecnológicos y evolución del mercado nuevas necesidades por parte de la empresa han surgido. McKay et al. [4] describen un conjunto de necesidades o requisitos que debe reunir un sistema de scheduling, algunos de los cuales se refieren a los algoritmos de secuenciación y al uso del sistema. Con respecto a los algoritmos de secuenciación, éstos deben ser flexibles y configurables. Deben ser capaces de manejar diversos criterios (scheduling multicriterio). A estos requisitos se les debe agregar otros tales como facilitar el mantenimiento y la reducción de costos de desarrollo.

Por último , [5] intenta integrar esta definiciones definiendo a un manufacturing scheduling como el proceso de toma de decisión que consiste de asignar un conjunto de operaciones/tareas requeridas para manufactura un conjunto de productos con los recursos existentes en una planta, como así también los tiempos necesario para iniciar estas operaciones o tareas. Un Schedule o plan se define como un conjunto específico de asignaciones de

operaciones o tareas a los recursos sobre una escala temporal.

Evolución de los sistemas de scheduling

La investigación en el diseño y desarrollo de sistemas de scheduling se puede clasificar en 5 generaciones. Inicialmente, esta clase de problemas fue abordada por el área de la ingeniería industrial, y posteriormente por el dominio de la investigación operativa. En los comienzos del siglo pasado, Henry L. Gantt, fue uno de los primeros en proponer un sistema de scheduling: daily balance “el equilibrio diario” que consiste en un método de programar y de registrar el trabajo. En 1960 se desarrolló Planalog Control Board, una herramienta que constaba de una pizarra y entre otras cosas podía forzar restricciones de precedencia. Los primeros sistemas de scheduling computarizados nacieron con el MRP (Material Requirement Planning). En la segunda generación surgen los ISS (Intelligent Scheduling Systems), tales como ISIS y OPIS de [6]. Esta fue una etapa donde el problema de scheduling se comenzó a abordar con técnicas de inteligencia artificial tales como algoritmos genéticos, simulated annealing, redes neuronales y constraint satisfaction techniques. En esta generación aparecen los sistemas de scheduling que se construyen utilizando técnicas de KBS (Knowledge Based Systems). La tercera generación se inició con el surgimiento del paradigma orientado, dando origen a sistemas de scheduling desarrollados con esta tecnología. Esto permitió el desarrollo de frameworks especializados en scheduling, los cuales encapsulan técnicas de inteligencia artificial [7]. La cuarta generación de sistemas de scheduling es la que se integra en lo que se conoce

como Advanced Planning and Scheduling Systems [8]. La quinta generación está surgiendo actualmente, está basada sobre la reutilización sistemática del software, la computación en la nube y la web de tiempo real.

Computación en la nube.

Según [9] Cloud Computing es un modelo para habilitar acceso conveniente por demanda a un conjunto compartido de recursos computacionales configurables, por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente provisionados y liberados con un esfuerzo mínimo de administración o de interacción con el proveedor de servicios. La característica principal de este modelo es que el software se despliega como servicios que están alojados en la nube, aquí los usuarios pagan por lo que usan, es decir son servicios elásticos. Esta característica permite una escalabilidad flexible de un sistema basado en el modelo de cloud computing, es decir un sistema puede atender a un solo usuario en un momento y a miles de usuarios en otro momento, los costes de la atención varían en función de la cantidad de usuarios que hay para un momento dado. Normalmente en este modelo atender pocos usuarios resulta mucho más económico que comprar las licencias de un software.

Programación web en tiempo real

La web en tiempo real es la web en la que los datos se entregan a sus destinatarios (sean humanos o máquinas) en tiempo real o casi real, tan pronto como estén disponibles. Su alcance se extiende desde las finanzas y herramientas médicas, a los servicios de redes sociales y medios de comunicación. Los principales requisitos y capacidades de aplicaciones en tiempo real son

interactividad para presentar los datos en tiempo real a los clientes, capacidad para mantener una conexión permanente con el servidor, protocolo de transmisión de datos más rápida, capacidad para integrar la lógica del cliente para reducir la carga del servidor, capacidad para enviar datos de múltiples fuentes, incluyendo los cambios de otros usuarios [10].

Meteor es un framework de JavaScript que tiene como objetivo automatizar y simplificar el desarrollo de aplicaciones web que actúan en tiempo real. Usa la comunicación en tiempo real mediante un protocolo llamado Distributed Data Protocol (DDP), que es admitido en navegadores nuevos que usan WebSockets o en navegadores anteriores que usan Asynchronous JavaScript + XML (Ajax) de sondeo largo. En ambos casos, la comunicación navegador-servidor es transparente [11].

Líneas de producto de software

La Línea de Productos de Software (LPS) se refiere a técnicas de ingeniería para crear sistemas de software que comparten un conjunto común y gestionado de características que satisfacen las necesidades específicas de un segmento de mercado y son desarrollados a partir de un conjunto compartido de activos de software, usando un medio común de producción [12-14]. Los beneficios que se tienen al hacer uso de la LPS es la entrega de productos de software de una manera más rápida, económica y con una mejor calidad.

El Desarrollo de Software Orientado a Características (DSOC) es un paradigma para la construcción, la personalización, y la síntesis de Sistemas de Software. Una característica es un aspecto de un sistema de software distintivo o visible para el usuario. La idea básica de DSOC es descomponer los sistemas de software en características con el fin de proporcionar

opciones de configuración y facilitar la generación de sistemas de software basado en una selección de características [15]. De esta manera, una línea de productos de software se refiere a un conjunto de sistemas de software que pueden ser obtenidos a partir de un determinado conjunto de características. Un determinado código (o parte) es mapeado en una característica con el objeto de construir un sistema de software a medida con la única acción de seleccionar las características que se desean de este software. FeatureIDE es un marco de trabajo para el Desarrollo de Software Orientado a Características (DSOC) basado en Eclipse. Según [16], FeatureIDE se enfoca en todo el proceso de desarrollo e incorpora herramientas para la implementación de Líneas de Producto de Software en un Entorno de Desarrollo Integrado. FeatureIDE soporta la implementación de varias técnicas como la programación orientada a características, la programación orientada a aspectos y preprocesadores.

Resultados y Objetivos

El objetivo es investigar sobre métodos y técnicas en pos de integrar las más recientes tecnologías de cloud computing, web de tiempo real, software product lines y de los solucionadores para optimización en el desarrollo de sistemas de scheduling de producción.

Estas nuevas tecnologías presentan nuevos desafíos y oportunidades para los sistemas de toma de decisión y optimización en particular para los sistemas de planificación y scheduling de la producción. La toma de decisión y optimización es una línea de investigación en la que el grupo de investigación cuenta con mucha experiencia. Por otro lado, nuestro grupo de investigación en los últimos años

también ha estado trabajando en otros proyectos relacionados con estas tecnologías desde que las mismas se consideraban emergentes. Creemos que las tecnologías ya están lo suficientemente maduras como para ser incorporadas a los sistemas de toma de decisión y optimización.

Formación de Recursos Humanos

El equipo de trabajo que lleva adelante este proyecto se compone de

- 5 docentes investigadores,
- 3 tesistas de grado en período de finalización.
- 2 tesistas de grado en período iniciación.
- 2 tesistas de posgrado (maestría) iniciando sus trabajos.

Referencias

- [1] K. R. Barker, *Elements of sequencing and scheduling*. New York: John Wiley and Sons, 1974.
- [2] B. L. MacCarthy and J. Liu, "Addressing the gap in scheduling research: A review of optimization and heuristic methods in production scheduling," *International Journal of Production Research*, vol. 31, pp. 59-79, 1993.
- [3] B. P.-C. Yen and M. Pinedo, "On the design and development of scheduling systems," in *Fourth International Conference on Computer Integrated Manufacturing and Automation Technology*, 1994, pp. 197 - 204.
- [4] K. McKay, M. L. Pinedo, and S. Webster, "Practice-Focused Research issues for scheduling systems," *Production and Operations Management*, vol. 11, pp. 249-258, 2002.
- [5] J. M. Framinan, R. Leisten, and R. R. García, "Manufacturing scheduling systems," *An integrated view on Models, Methods and Tools*, pp. 51-63, 2014.
- [6] F. S. Smith, *OPIS: A Methodology and architecture for reactive scheduling*: Morgan Kaufmann., 1994.
- [7] C. Le Pape, "Implementation of Resource Constraints in ILOG SCHEDULE: A Library for the Development of Constraint-Based Scheduling Systems," *Intelligent Systems Engineering* 3, vol. 3, pp. 55-66., 1994.
- [8] M. L. Entrup, *Advanced Planning and Scheduling Systems: Advanced Planning in Fresh Food Industries: Integrating Shelf Life into Production Planning*. Physica-Verlag HD, 2005.
- [9] N. Antonopoulos and L. Gillam, *Cloud computing*: Springer, 2010.
- [10] D. Berube, "Desarrollo de sitios web en tiempo real " *IBM-Developers Works*, vol. <http://www.ibm.com/developerworks/ssa/library/wa-meteor/wa-meteor-pdf.pdf>, 2013.
- [11] F. Vogesteller, "Building Single-page Web Apps with Meteor," *Packt Publishing*, 2015.
- [12] J. Greenfield and K. Short, "Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools," *Wiley*, 2004.
- [13] K. Pohl, G. Böckle, and F. J. von der Linden, "Software Product Line Engineering: Foundations, Principles and Techniques," *Springer* 2005.
- [14] D. M. Weiss and C. T. R. Lai, "Software Product-Line Engineering - A Family-Based Software Development Process.," *Addison-Wesley. Massachusetts*, 1999.
- [15] K. Czarnecki, "Overview of Generative Software Development," *Unconventional Programming Paradigms*, 2005.
- [16] C. Kastner, T. Thum, G. Saake, and J. Feigenspan, "FeatureIDE: A tool framework for feature-oriented software development " *IEEE 31st ICSE* 2009.