

Una herramienta para evaluar el rendimiento de aplicaciones intensivas en datos*

Abel Gómez and José Merseguer

Departamento de Informática e Ingeniería de Sistemas
Universidad de Zaragoza, Spain
{abel.gomez|jmerse}@unizar.es

Resumen Las aplicaciones intensivas en datos (AID) que usan tecnologías de *Big Data* se están convirtiendo en una parte importante del mercado de desarrollo de software. Sin embargo, las técnicas —y su automatización— para el asesoramiento de la calidad para este tipo de aplicaciones es claramente insuficiente. El proyecto DICE H2020 tiene como objetivo definir metodologías y crear herramientas para desarrollar y monitorizar AID mediante técnicas de ingeniería dirigida por modelos. En este artículo presentamos un componente clave del proyecto DICE: su herramienta de simulación. Esta herramienta es capaz de evaluar el rendimiento de AID simulando su comportamiento mediante modelos de redes de Petri. Como complemento, existe a disposición un vídeo mostrando la herramienta en <http://tiny.cc/z1qzay>.

Keywords: Aplicaciones Intensivas en Datos (AID), Lenguaje de Modelado Unificado (UML), Redes de Petri (RdP)

1. Introducción

La creciente disponibilidad de infraestructuras en la nube, así como clusters distribuidos o nuevos modelos de programación (como *MapReduce*), están teniendo un fuerte impacto en el mercado del desarrollo de software y en las llamadas *aplicaciones intensivas en datos* (AID). En esta nueva realidad, es necesario proporcionar metodologías que, siendo capaces de lidiar con la creciente complejidad del software y su despliegue, permitan incrementar la productividad.

El proyecto DICE [2] aspira a definir este nuevo marco de trabajo para AID que permita explotar las tecnologías de *Big Data* en nubes públicas o privadas. DICE propone técnicas, metodologías y herramientas para diseñar, desplegar, monitorizar y refactorizar AID. En lo que concierne al diseño de las aplicaciones el Lenguaje de Modelado Unificado (UML) es fundamental en DICE. Aquellas características inherentes a una AID que no están contempladas en la propia especificación de UML (e.g., tecnología de *Big Data* a utilizar) se definen usando un nuevo perfil proporcionado por DICE [4]. Un aspecto clave del proyecto DICE es su ecosistema de herramientas para simular, analizar y optimizar las

* Trabajo financiado por la UE bajo el programa H2020 (n^o 644869 – DICE), el MICINN (ref. TIN2013-46238-C4-1-R) y el Gobierno de Aragón (ref. T94 – DisCo)

aplicaciones modeladas en DICE. Una de estas herramientas es la llamada *DICE Simulation Tool*, que permite evaluar los requisitos de rendimiento de una AID usando las redes de Petri [1] (RdP). A continuación, la sección 2 introduce las bases sobre las que se construye la herramienta de simulación con un sencillo ejemplo, mientras que la sección 3 describe la arquitectura de dicha herramienta. Finalmente, la sección 4 cierra el artículo.

2. Un ejemplo ilustrativo

La figura 1a muestra un diagrama de actividades de UML sencillo. El diagrama consta de un único nodo inicial (*Inicio*), un único nodo final (*Fin*) y dos acciones que se ejecutan de forma alternativa (*A1* y *A2*). Se puede observar que se han aplicado los estereotipos *GaWorkloadEvent* y *GaStep*. Estos estereotipos, definidos en el profile MARTE [6], son importados por el perfil de DICE para especificar tanto la carga del sistema como los requisitos de demanda de servicio. La aplicación de *GaWorkloadEvent* al nodo inicial marca una carga del sistema cerrada con una población inicial de 3 trabajos, mientras que, la aplicación de *GaStep* a los flujos que parten del nodo de decisión indican que las probabilidades de que se ejecuten las actividades *A1* o *A2* son $\$p1$ y $\$p2$ respectivamente. Respecto a $\$p1$ y $\$p2$, éstas son variables que pueden configurarse posteriormente en el momento de evaluar el sistema (sec. 3). Finalmente, el estereotipo *GaStep* aplicado a *A1* y *A2* especifica que, por término medio, la actividad *A1* requerirá un segundo de uso de CPU para su ejecución y la actividad *A2* requerirá dos.

Si bien el modelo UML es óptimo para que el ingeniero especifique tanto el *workflow* de la AID como sus características inherentes, no lo es para llevar a cabo una evaluación y asesoramiento de las propiedades cuantitativas de la AID. Es por ello que es necesario transformar el modelo UML en una RdP. Las RdP estocásticas son un formalismo gráfico que posibilita el modelado, análisis y evaluación de los sistemas. La figura 1b muestra la correspondiente RdP que permitirá evaluar el rendimiento del sistema especificado en la figura 1a. En ella, los nodos circulares representan *lugares*, los nodos rectangulares con relleno negro *transiciones inmediatas*, y los nodos rectangulares con relleno blanco *transiciones con tiempo*. Se puede observar que el lugar llamado *Inicio* contiene tres *tokens* (como indica la población inicial especificada en el modelo UML); las transiciones inmediatas t_2 y t_3 especifican un peso de $\$p1$ y $\$p2$ respectivamen-

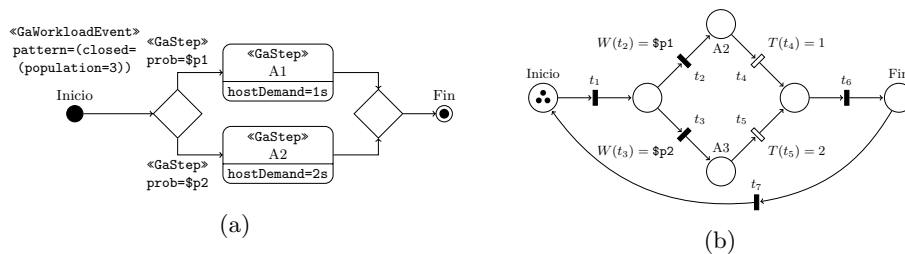


Figura 1: Ejemplo de un diagrama de actividad y su correspondiente red de Petri

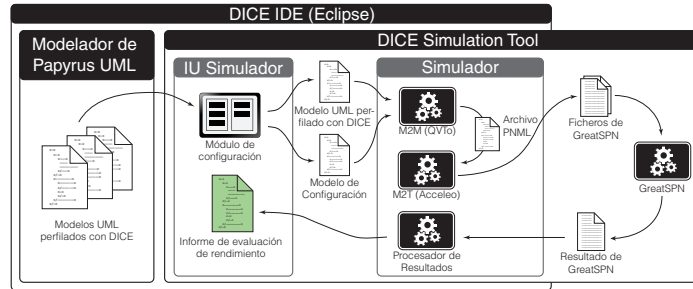


Figura 2: Arquitectura de alto nivel de la herramienta

te; y las transiciones con tiempo t_4 y t_5 especifican un tiempo medio de disparo de 1 y 2 unidades de tiempo respectivamente. El análisis de esta red de Petri permitirá razonar sobre el rendimiento teórico del sistema modelado.

3. La herramienta de simulación de DICE

La herramienta de simulación de DICE permite no sólo automatizar la transformación de un diagrama UML a su modelo formal, sino también controlar el proceso de simulación de forma transparente para el usuario. Además esta herramienta implementa el *Profile* de DICE y en un futuro tendrá capacidades para redefinir el diseño de la aplicación en función de las ejecuciones reales de la misma. La figura 2 muestra, de forma simplificada, su arquitectura.

El *DICE-IDE* es el entorno de desarrollo de DICE. Está construido sobre Eclipse [9] y en él se integran todos los componentes proporcionados por el proyecto. Como paso previo a una evaluación de rendimiento, se debe modelar una AID con UML y el perfil de DICE. Para realizar el modelado, DICE emplea *Papyrus UML* [8] (fig. 3 izquierda). La interfaz de usuario del simulador (*IU Simulador*) es un *plug-in* de Eclipse que se integra perfectamente en el DICE-IDE y proporciona una fachada sencilla que interactúa con las herramientas de análisis subyacentes de forma transparente. El *módulo de configuración* permite

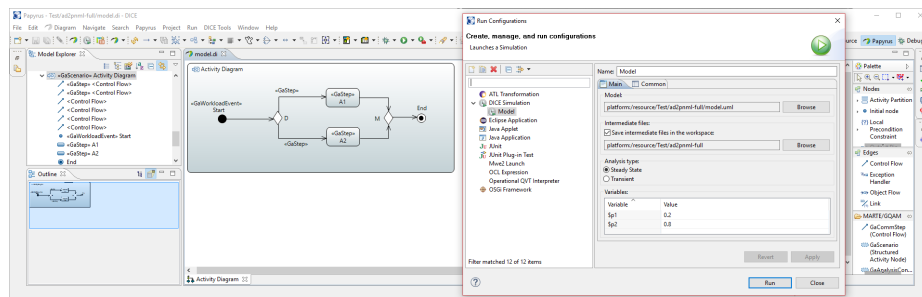


Figura 3: El entorno de desarrollo de DICE y el módulo de configuración

al ingeniero instanciar los parámetros de la AID especificados mediante el profile de DICE, como por ejemplo la carga de trabajo. Como resultado de una configuración, el *Simulador* recibe dos ficheros: el modelo perfilado con DICE (modelo a analizar) y el *Modelo de configuración*.

El *Simulador* es el componente que orquesta la interacción entre todas las herramientas involucradas. Para ello, ejecuta los siguientes pasos: (i) transforma el modelo UML perfilado en una RdP según el estándar PNML [5] empleando una transformación modelo-a-modelo (M2M); (ii) convierte la RdP estándar al formato específico de la herramienta de análisis (*GreatSPN* [3]), con una transformación modelo-a-texto (M2T); (iii) ejecuta el análisis empleando *GreatSPN*; y (iv) construye un *informe de evaluación de rendimiento* independiente de la herramienta a partir del resultado proporcionado por la herramienta de análisis. Dicho *informe de evaluación de rendimiento* se muestra al usuario en la IU del simulador empleando los conceptos definidos en el modelo UML inicial.

4. Conclusiones

En este artículo hemos presentado la *herramienta de simulación* del proyecto *DICE*. La herramienta es capaz de proporcionar un *informe de evaluación del rendimiento* de una AID a partir de un modelo UML inicial anotado con el perfil de DICE [4]. En su estado actual, el prototipo cubre todos los pasos del flujo de trabajo de la simulación. La herramienta se integra completamente en el entorno de desarrollo de DICE y proporciona una interfaz amigable que oculta al usuario los detalles de la herramienta de análisis subyacente. La herramienta (<http://tiny.cc/z1qzay>), se ha publicado bajo una licencia de código abierto [7].

Referencias

1. Ajmone-Marsan, M., Balbo, G., Conte, G., Donatelli, S., Franceschinis, G.: Modeling with Generalized Stochastic Petri Nets. John Wiley and Sons (1994)
2. Casale, G., et al.: DICE: Quality-driven Development of Data-intensive Cloud Applications. In: Proceedings of the Seventh International Workshop on Modeling in Software Engineering. pp. 78–83. IEEE Press, NJ, USA (2015)
3. Dipartimento di informatica, Università di Torino: GRaphical Editor and Analyzer for Timed and Stochastic Petri Nets (2016), URL: <http://www.di.unito.it/~greatspn/index.html>
4. Gómez, A., Merseguer, J., Di Nito, E., Tamburri, D.A.: Towards a UML Profile for Data Intensive Applications. In: Proceedings of the 2nd International Workshop on Quality-aware DevOps. pp. 18–23. ACM (2016), DOI: 10.1145/2945408.2945412
5. ISO: Systems and software engineering – High-level Petri nets – Part 2: Transfer format. ISO/IEC 15909-2:2011, Geneva, Switzerland (2008)
6. OMG: UML Profile for MARTE: Modeling and Analysis of Real-time Embedded Systems, Version 1.1 (June 2011)
7. The DICE Consortium: DICE Simulation Repository (2016), URL: <https://github.com/dice-project/DICE-Simulation>
8. The Eclipse Foundation: Papyrus (2016), URL: <https://eclipse.org/papyrus/>
9. The Eclipse Foundation: Website (2016), URL: <http://www.eclipse.org/>