

Propuesta de investigación: Marco Teórico Inicial y Objetivos.

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Maestría en Computación
MC-7205 Tema Selecto de Investigación

CARLOS MARTÍN FLORES GONZÁLEZ, Carné: 2015183528

ÍNDICE

Índice	1
1. The Palladio component model for model-driven performance prediction[1]	2
2. Performance-oriented DevOps: a research agenda[2]	3
3. Performance Engineering for Microservices: Research Challenges and Directions[3]	4
3.1. Retos de investigación	4
3.1.1. Pruebas de rendimiento	4
3.1.2. Monitoreo	4
3.1.3. Modelado del rendimiento	4
4. An industrial case study of performance and cost design space exploration[4]	5
5. Sobre la propuesta	6
6. Objetivos	6
6.1. Objetivo General	6
6.2. Objetivos Específicos	6
Referencias	6
A.	6

1. THE PALLADIO COMPONENT MODEL FOR MODEL-DRIVEN PERFORMANCE PREDICTION[1]

El el desarrollo de software basado en componentes (CBSE, por sus siglas en inglés) la idea central es construir sistemas de software complejo uniendo componentes básicos. El objetivo inicial de CBSE fue incrementar el nivel de reutilización. Sin embargo, estructuras compuestas también pueden aumentar el nivel de predictibilidad del sistema durante etapas tempranas de desarrollo, esto porque modelos certificados de componentes individuales puee ser compuestos, permitiendo a los arquitectos de software razonar sobre la estructura compuesta. Esto es importante para las propiedades funcionales, pero también para las propiedades extra funcionales como rendimiento y confiabilidad.

Los métodos para predicción de rendimiento y confiabilidad de sistemas de software en general son limitados y raramente usados en la industria. Este reto es aún mayor en CBSE debido a que varios roles independientes están involucrados en la creación del sistema. Muchos métodos existentes para predicción de CBSE requieren que los arquitectos de software modelen el sistema basados en especificaciones de un solo componente. Luego se asume que el arquitecto proporcionará información faltante. Otros enfoques dejan de lado factores que afectan el rendimiento percibido de un componente de software tal y como la influencia de servicios externos, cambio de recursos de su ambiente o diferentes parámetros de entrada. Sin embargo, para predicciones precisas, todas estas dependencias tiene que hacerse explícitas en la especificación del componente.

Con *Palladio*¹ *Component Model* (PCM) un meta-modelo provee la especificación de información relevante del rendimiento de una arquitectura basada en componentes. El diseño y análisis de estos meta-modelos son los primeros que explícitamente incluyen todos los factores que influncian el rendimiento de un componente de software como lo son: la implementación, rendimiento de servicios externos, rendimiento del ambiente de ejecución y el perfil de uso. El modelo es diseñado con especial énfasis en la predicción de atributos de calidad de servicio como el rendimiento y la confiabilidad.

En el proceso de CBSE(figura 1) los autores distinguen cuatro tipos de roles involucrados en la producción de artefactos de un sistema de software: *desarrolladores de componentes* especifican e implementan los componentes, *arquitectos de software* ensamblan componentes para construir aplicaciones, *system deployers* modela los recursos del ambiente y luego su asignación, *expertos del dominio de negocio* que están familiarizados con los usuarios del sistema y proporcionan modelos de su uso. El modelo completo del sistema puede ser derivado a partir de modelos parciales especificados por cada uno de los roles para que luego las propiedades extra funcionales puedan ser predecidas.

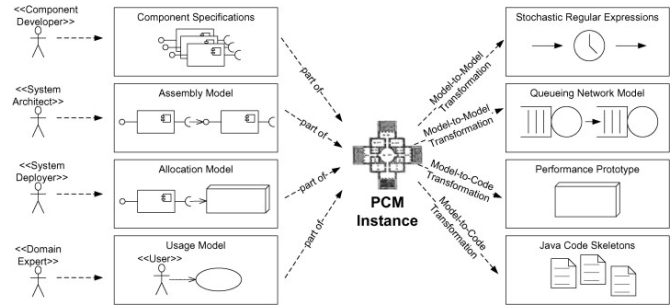


Fig. 1. Proceso CBSE propuesto por PCM

¹Toma su nombre del arquitecto renacentista Andrea Palladio (1508 - 1580) quien de cierta forma, intentó predecir el impacto estético de sus construcciones por adelantado.

2. PERFORMANCE-ORIENTED DEVOPS: A RESEARCH AGENDA[2]

3. PERFORMANCE ENGINEERING FOR MICROSERVICES: RESEARCH CHALLENGES AND DIRECTIONS[3]

Los microservicios complementan enfoques como DevOps y entrega continua(CD por sus siglas en inglés) en relación en términos de arquitectura de software. Junto con este estilo de arquitectura, otras tecnologías importantes para *deployment* como, virtualización basada en contenedores y soluciones de orquestación de contenedores han emergido. Estas tecnologías permiten explotar plataformas en la nube, proporcionando altos niveles de escalabilidad, disponibilidad y portabilidad para microservicios. A pesar del hecho de que es una necesidad inherente contar con escalabilidad y elasticidad, la ingeniería de rendimiento para microservicios hasta ahora ha tenido poca atención por parte de las comunidades tanto de microservicios como de comunidades de investigación de ingeniería de rendimiento. Un gran cuerpo de conocimiento y buenas prácticas para ingeniería de rendimiento para desarrollo tradicional de software y arquitecturas está disponible. Sin embargo, sus aplicaciones en DevOps imponen tanto retos como oportunidades.

3.1. Retos de investigación

3.1.1. Pruebas de rendimiento.

- Reemplazar y compensar pruebas extensivas de integración y de sistema por un control detallado de los entornos de producción
- Alinear las pruebas de rendimiento y pruebas de regresión de rendimiento con prácticas de CD, por ejemplo, acelerar las estas de pruebas correspondientes.
- Selección dinámica y semi automática de pruebas de rendimiento

3.1.2. Monitoreo.

- Instrumentación para monitoreo distribuido arquitecturas microservicios políglotas
- Métricas adicionales para monitorear microservicios
- Técnicas de detección precisa de anomalías en arquitecturas de microservicios

3.1.3. *Modelado del rendimiento.* De momento no existen enfoques para modelado del rendimiento de microservicios

- Adoptar modelos de rendimiento para casos de uso como planeamiento de capacidad, confiabilidad y resiliencia
- Buscar abstracciones de modelado apropiadas
- Extracción automática de modelos de rendimiento
- Aprender del comportamiento de la infraestructura e integrarlo en los modelos de rendimiento

4. **AN INDUSTRIAL CASE STUDY OF PERFORMANCE AND COST DESIGN SPACE EXPLORATION**^[4]

5. SOBRE LA PROPUESTA

6. OBJETIVOS

6.1. Objetivo General

6.2. Objetivos Específicos

REFERENCIAS

- [1] Stephen Becker, Heiko Koziulek and Ralf Reussner. *The Palladio component model for model-driven prediction*. Journal of Systems and Software 82:3-22. Elsevier Science Inc. 2009. DOI: <http://dx.doi.org/10.1016/j.jss.2008.03.066>
- [2] Andreas Brunnert, André van Hoorn, Felix Willnecker, Alexandru Danciu, Wilhelm Hasselbring, Christoph Heger, Nikolas Roman Herbst, Pooyan Jamshidi, Reiner Jung, Jóakim von Kistowski, Anne Koziulek, Johannes Kroß, Simon Spinner, Christian Vögele, Jürgen Walter, and Alexander Wert. *Performance-oriented devops: A research agenda*. Technical Report SPEC-RG-2015-01, SPEC Research Group - DevOps Performance Working Group, Standard Performance Evaluation Corporation (SPEC), 2015. <http://arxiv.org/abs/1508.04752>
- [3] Robert Heinrich, André van Hoorn, Holger Knoche, Fei Li, Lucy Ellen Lwakatare, Claus Pahl, Stefan Schulte, and Johannes Wettinger. *Performance engineering for microservices: Research challenges and directions*. In Companion Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering, 2017, pages 223-226. DOI: <https://doi.org/10.1145/3053600.3053653>
- [4] Thijmen de Gooijer, Anton Jansen, Heiko Koziulek, and Anne Koziulek. 2012. *An industrial case study of performance and cost design space exploration*. In Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering (ICPE '12). ACM, New York, NY, USA, 205-216. DOI: <https://ezproxy.itcr.ac.cr:2878/10.1145/2188286.2188319>

A.