

# MODELADO Y SIMULACIÓN DE FUNCIONES EN LA NUBE EN PLATAFORMAS *Function-as-a-Service (FaaS)*

**CARLOS MARTÍN FLORES GONZÁLEZ**

[martin.flores@computer.org](mailto:martin.flores@computer.org)

INSTITUTO TECNOLÓGICO DE COSTA RICA  
ESCUELA DE COMPUTACIÓN  
MAESTRÍA EN COMPUTACIÓN  
MC-7202 SEMINARIO DE INVESTIGACIÓN 1  
28 DE NOVIEMBRE, 2018

## AGENDA

- *Serverless* y *FaaS*
- Problema
- Justificación
- Antecedentes
- Objetivo general
- Objetivos específicos
- Alcance
- Entregables
- Metodología
- Conclusión

## *Serverless*

- *Serverless* se utiliza para describir un modelo de programación y una arquitectura en donde fragmentos de código son ejecutados en la nube sin ningún control sobre los recursos en donde el código se ejecuta.
- Esto de ninguna manera es una indicación de que no hay servidores, sino simplemente que el desarrollador delega la mayoría de aspectos operacionales al proveedor de servicios en la nube.
- A la versión de *Serverless* que utiliza explícitamente funciones como unidad de instalación se le conoce como **Function-as-a-Service (FaaS)**.

## FaaS: Function-as-a-Service

- Los servicios de funciones en la nube *Function-as-a-Service*, permiten a los desarrolladores instalar código en una plataforma de servicios en la nube en forma de función y, la infraestructura de la plataforma, es responsable de la ejecución, el aprovisionamiento de recursos, monitoreo y el escalamiento automático del entorno de ejecución.
- **Código en forma de función:** pequeño, sin estado, trabaja bajo demanda (se cobra solamente el tiempo que duró su ejecución) y con una sola responsabilidad funcional

---

---

---

---

---

---

---

## PROBLEMA

- Se carece de modelos de rendimiento que contribuyan a caracterizar el comportamiento de funciones en la nube distintas cargas de trabajo. Contar con tales modelos permitiría validar si las funciones pueden cumplir criterios de calidad de servicio.
- En las plataformas *FaaS* la infraestructura tecnológica subyacente se oculta por completo de los desarrolladores y diseñadores.
- La influencia de esta infraestructura y su configuración es vital para obtener predicciones significativas del comportamiento de una función sin esto, se puede conducir a la generación de predicciones erróneas con respecto del rendimiento de una función.

---

---

---

---

---

---

---

## JUSTIFICACIÓN INNOVACIÓN

- Proponer un método mediante el cual se pueden obtener estimaciones del rendimiento de una función en la nube.
- Realizar modelado y simulación basados en componentes para caracterizar el rendimiento de funciones en la nube sobre plataformas *FaaS*.
- Proporcionar, a partir de lo anterior, un modelo del rendimiento de una función en la nube.

---

---

---

---

---

---

---

## JUSTIFICACIÓN

### IMPACTO

- La adopción de tecnologías *serverless* en la industria, viene en franco aumento en los últimos 3 años.
- Al ser aún reciente, también se reportan problemas para monitorear, depurar, desarrollar y entender el modelo *FaaS*.
- Analizar el rendimiento de una función en la nube y obtener un modelo de este, contribuiría a tener un mejor entendimiento de esta tecnología:
  - Permitiría tener mayor control sobre los cambios en una función para cumplir con requerimientos de calidad de servicio.
  - Generar menores costos por el uso de la plataforma *FaaS*.

---

---

---

---

---

---

---

---

## JUSTIFICACIÓN

### PROFUNDIDAD - ACTIVIDADES PLANTEADAS

- Diseño e implementación de un caso de uso de referencia de una función en la nube.
- Obtención un modelo:
  - Realizar pruebas de carga sobre la función en la nube seleccionada.
  - Obtener métricas asociadas al rendimiento a partir de bitácoras de ejecución de la función.
  - Utilizar las métricas obtenidas para suministrarlas como entrada a una herramienta de extracción de modelos de rendimiento. La herramienta generará como resultado un modelo de rendimiento.
- Diseñar experimentos y ejecutar simulaciones sobre el modelo obtenido con el fin de validar las estimaciones o determinar si es necesario calibrar el modelo.

---

---

---

---

---

---

---

---

## INGENIERÍA DE RENDIMIENTO DE SOFTWARE (SPE)

- Woodside et al. "*Ingeniería de rendimiento del software representa toda la colección de actividades de ingeniería de software y análisis relacionados utilizados a través del ciclo de desarrollo de software que están dirigidos a cumplir con los requerimientos de rendimiento*"
- Dividido en dos grandes categorías: basadas en mediciones y basados en modelos
- Otros autores consideran que ambas categorías no son mutuamente excluyentes sino todo lo contrario

---

---

---

---

---

---

---

---

### ANTECEDENTES: SPE EN APLICACIONES EN LA NUBE

- Se reporta que el monitoreo, pruebas y modelado del rendimiento como las tres áreas en donde se carece de investigación y desarrollo de SPE y microservicios.
- Falta de investigación empírica sobre diseño, desarrollo y evaluación de aplicaciones de microservicios. Se cuenta con muy pocas aplicaciones y arquitecturas de referencia, así como de cargas de trabajo que contribuyan a caracterizar el comportamiento las mismas.
- A pesar del alto nivel de adopción de prácticas de integración continua y entrega continua en ingeniería de software, ninguna toma en cuenta aspectos relacionados con el rendimiento.

---

---

---

---

---

---

### ANTECEDENTES: SPE EN *FaaS* (1)

- Los investigadores han empezado a describir y analizar *FaaS* a través de encuestas y experimentos, y también por análisis económicos, sin embargo se reporta que aún no se conoce mucho acerca de SPE en *FaaS*.
- Se menciona que al igual que con microservicios, *FaaS* también requiere de nuevas estrategias de modelado para capturar el comportamiento del código bajo estas infraestructuras. Los modelos de rendimiento tradicionales basados en en la noción de máquinas independientes podría ser inadecuado.

---

---

---

---

---

---

### ANTECEDENTES: SPE EN *FaaS* (2)

- SPEC RG Cloud: uno de los principales desafíos de plataformas *FaaS* es la predicción del rendimiento.
- Se indica que la aplicación de modelos de rendimiento de sistemas de software tradicionales en *FaaS* trae nuevos retos como la brecha de información (B.I.).
- B.I.: El usuario de *FaaS* no está consciente de los recursos de hardware, mientras que por otro lado, la plataforma de *FaaS* no tiene información acerca de los detalles de implementación

---

---

---

---

---

---

### ANTECEDENTES: SPE EN *FaaS* (3)

- Tal y como en las aplicaciones tradicionales, la entrada (tamaño, estructura y contenido) influyen en el rendimiento de una función, el hecho de tener una infraestructura oculta hace necesario encontrar nuevos modelos que logren predecir de forma precisa el rendimiento de una función.
- Técnicas de modelado desarrolladas para sistemas de software se podrían aprovechar para *FaaS*, como por ejemplo modelado y simulación de arquitecturas de software basados en componentes.

---

---

---

---

---

---

### OBJETIVO GENERAL

- Diseñar un método para modelar el rendimiento de funciones en la nube alojadas en plataformas *Function-as-a-Service* por medio del modelado y la simulación basados en componentes, con el fin de evaluar los factores que pueden influir en su comportamiento.

---

---

---

---

---

---

### OBJETIVOS ESPECÍFICOS (1)

- 1.Revisar el estado del arte de trabajos relacionados con enfoques de predicción y medición del rendimiento en sistemas de software como servicio.
- 2.Sintetizar un caso de uso de una función en la nube considerada como de referencia, con el propósito de analizar su comportamiento.
- 3.Elaborar, conforme a un diseño experimental, pruebas de rendimiento sobre el caso de uso seleccionado a fin de obtener datos base.

---

---

---

---

---

---

## OBJETIVOS ESPECÍFICOS (2)

4. Analizar los datos experimentales mediante herramientas de extracción de modelos de rendimiento de software.
5. Proponer y validar modelos de rendimiento a partir de los experimentos realizados.
6. Formular una guía metodológica para dar a conocer aspectos de rendimiento en funciones en la nube a partir de la experiencia obtenida.

---

---

---

---

---

---

## ALCANCE

### LO QUE LA PROPUESTA COMPRENDE

1. El estudio de factores que pueden influir en el rendimiento de un caso de uso de una función en la nube bajo una plataforma *FaaS*. A partir de la experiencia generada y los datos recolectados, este trabajo podría ser replicado con otras funciones en otras plataformas *FaaS*.
2. La propuesta de un modelo de rendimiento que permita la simulación de la ejecución de una función en la nube y obtener, a partir de esta, estimaciones del tiempo de respuesta de la ejecución de la función.

---

---

---

---

---

---

## ALCANCE

### LO QUE LA PROPUESTA **NO** COMPRENDE

1. El modelado o simulación de toda una plataforma *FaaS*.
2. La realización de un modelo o estudio de los aspectos económicos asociados a la ejecución de una función en la nube.
3. Estudiar el rendimiento de funciones en la nube que mantengan alguna relación o dependencias entre sí (orquestración de funciones).
4. Experimentar con múltiples plataformas *FaaS* ni con diversos lenguajes de programación.

---

---

---

---

---

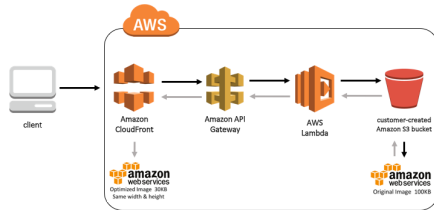
---

## ENTREGABLES

- 1.Revisión de literatura (Obj 1)
- 2.Implementación de caso de uso de función en la nube (Obj 2)
- 3.Pruebas sobre el caso de uso (Obj 3)
- 4.Modelado y análisis del rendimiento de la función (Obj 4)
- 5.Modelo de rendimiento de la función en la nube (Obj 5)
- 6.Guía Metodológica (Obj 6)

## METODOLOGÍA

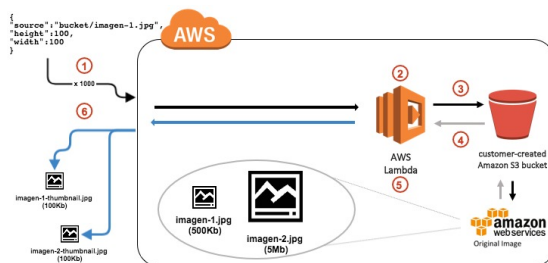
### ETAPA 1: CASO DE USO - MANEJADOR DE IMÁGENES



- Uno de los casos de uso más comunes identificados por Amazon en su plataforma AWS Lambda.
- Específico, popular, replicable en otros proveedores, lenguajes y plataformas de programación.

## METODOLOGÍA

### ETAPA 1: MANEJADOR DE IMÁGENES PARA SPE



## METODOLOGÍA

### ETAPA 2: MODELO DE RENDIMIENTO A PARTIR DE LA FUNCIÓN

Walter et al. Enfoque de ingeniería de rendimiento declarativo

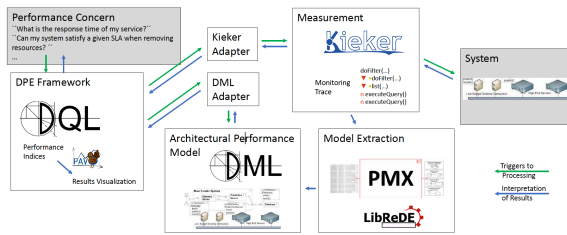
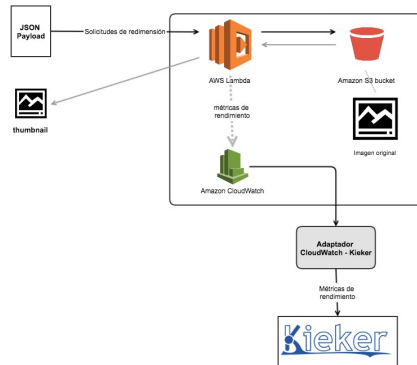


Figure 1: Overview of Tools for Declarative Performance Engineering

## METODOLOGÍA

### ETAPA 2: MODELO DE RENDIMIENTO A PARTIR DE FUNCIÓN



## METODOLOGÍA

### ETAPA 3: EJECUCIÓN DE SIMULACIONES

- *Palladio Workbench* es la herramienta que permite crear y simular modelos basados en *Palladio Component Model (PCM)*
- Cuando se haya derivado un modelo de rendimiento de la función, se procederá a cargar este modelo en *Palladio Workbench* para modificar y calibrar el modelo
- Las herramientas de simulación que vienen con *Palladio Workbench*, se utilizarán para elaborar experimentos para evaluar el grado de coincidencia de los resultados de la simulación con los obtenidos al exponer la función a cargas de trabajo reales.



## CONCLUSIONES

- *FaaS* ejecuta fragmentos de código en forma de función en la nube y el proveedor de servicios controla los aspectos operacionales de la ejecución.
- Se carece de modelos de rendimiento que contribuyan a caracterizar el comportamiento de funciones en la nube distintas cargas de trabajo.
- Se propone realizar modelado y simulación basados en componentes para caracterizar el rendimiento de funciones en la nube sobre plataformas *FaaS*.

## CONCLUSIONES (2)

- El enfoque de ingeniería de rendimiento declarativo se presenta como una alternativa para extraer un modelo de rendimiento a partir de una función en la nube.
- *Palladio Component Model*, permite el modelado y simulación de arquitecturas de software basadas en componentes. Experimentos y simulaciones sobre el modelo obtenido podrán ser ejecutados para evaluar si logran caracterizar el comportamiento de una función.