

Perses: Un framework para evaluar la Calidad de Servicio en aplicaciones móviles distribuidas

Sergio Laso¹, Javier Berrocal¹, Pablo Fernández², Antonio Ruiz-Cortés² and Juan M. Murillo¹

¹ University of Extremadura, Spain
{slasom, jberolm, juanmamu}@unex.es

² University of Sevilla, Spain
{pablofm, aruiz}@us.es

Abstract. Las crecientes capacidades de los dispositivos finales han llevado al despliegue masivo de aplicaciones móviles distribuidas. El éxito de estas aplicaciones depende en gran medida de la Calidad del Servicio (QoS) que ofrecen. Esta calidad es especialmente difícil de evaluar debido al gran número de entidades implicadas y a su heterogeneidad. Las herramientas actuales de evaluación no dan soporte a este tipo de aplicaciones, suelen centrarse en la evaluación de la QoS proporcionada por una sola entidad. Sin embargo, la QoS de las aplicaciones distribuidas no sólo depende de la QoS de cada entidad, también hay que evaluar las interacciones entre las entidades. En este artículo se presenta un framework, denominado Perses, para lanzar entornos virtuales que permitan simular y evaluar la ejecución de aplicaciones móviles distribuidas. Esta simulación proporciona resultados de la QoS alcanzada. Además, el framework se ha integrado en una metodología DevOps para automatizar su ejecución.

Vídeo de presentación— https://youtu.be/wpIApe_sPFE

Keywords: Aplicaciones Móviles Distribuidas; Calidad de Servicio; DevOps; Entorno Virtual

1 Introducción

El éxito o el fracaso de cualquier aplicación móvil depende de dimensiones, como la publicidad, la viralidad o la Calidad del Servicio (QoS) proporcionada [7]. Una mala calidad llevará a los usuarios a rechazar rápidamente la aplicación.

Estas aplicaciones se desarrollan normalmente utilizando un estilo arquitectónico cliente-servidor. Los componentes informáticos más exigentes se despliegan normalmente en entornos cloud debido a su gran capacidad de computación y almacenamiento. Los componentes básicos se despliegan en el lado del cliente, ya que sus requisitos de computación suelen ser bajos y pueden ser ejecutados por casi cualquier dispositivo móvil.

Durante los últimos años, las capacidades de computación y almacenamiento de los dispositivos móviles han aumentado significativamente [3], provocando la aparición de nuevos paradigmas centrados en su explotación, como los Microservicios Humanos [8] o Mist Computing [11]. Estos paradigmas permiten a los

desarrolladores cargar algunas funcionalidades de computación en los dispositivos móviles para mejorar la QoS y ser más competitivos a nivel empresarial.

En las aplicaciones distribuidas, tanto el cloud como el lado cliente/móvil tienen componentes de computación y almacenamiento. Por lo tanto, la evaluación de la QoS debe combinar la evaluación conjunta del lado del servidor y del lado móvil, así como de la interacción entre ambos. Sólo la evaluación conjunta de las tres dimensiones permitiría predecir con suficiente precisión la QoS esperada. Además, suele tratarse de entornos heterogéneos (diferentes dispositivos móviles), siendo aún más difícil predecir la QoS esperada.

Actualmente, existen diferentes plataformas que permiten evaluar el comportamiento de aplicaciones móviles no distribuidas en entornos heterogéneos. AWS Device Farm [10] y Azure App Center Test [1] permiten a los desarrolladores probar diferentes dispositivos con diferentes configuraciones. Sin embargo, se centran en la evaluación de tests de interfaz de usuario (mediante Appium³, Espresso⁴, etc.). Por lo tanto, no están diseñadas para medir los atributos de QoS.

En este artículo, presentamos un framework, denominado Perses [9], para evaluar la QoS de las aplicaciones móviles distribuidas. Para ello, permite definir y desplegar entornos virtuales personalizados, con múltiples dispositivos virtuales heterogéneos, en los que los desarrolladores pueden simular el despliegue de la aplicación móvil distribuida y lanzar tests de rendimiento para evaluar diferentes atributos de la QoS. Además, este framework se ha integrado completamente en una metodología DevOps [2], reduciendo el esfuerzo de evaluación de la QoS antes de desplegar la aplicación en producción.

El resto del artículo se estructura de la siguiente forma: La Sección 2 explica las características de Perses y su arquitectura. La Subsección 2.1 describe un caso de uso para la demo. Finalmente, la Sección 3 presenta las conclusiones.

2 Perses

Perses es un framework que permite a los desarrolladores desplegar fácilmente un entorno virtual con múltiples dispositivos virtuales heterogéneos para evaluar la QoS de una aplicación distribuida. Para utilizarlo, sólo hay que definir un *fichero de configuración* con las características del entorno, los tests a ejecutar y los umbrales deseados de los atributos de calidad.

Perses está totalmente integrado en una metodología DevOps, siendo capaz de automatizar todos los diferentes pasos que se deben ejecutar durante el proceso de evaluación. Además es totalmente escalable, permitiendo evaluar aplicaciones con un gran número de dispositivos virtualizados.

La Figura 1 muestra un diagrama general de la arquitectura de Perses. Perses consta de dos componentes: *Perses Launcher*, centrado en la definición y configuración del entorno virtual, y *Perses Virtual Environment*, encargado de crear el entorno heterogéneo y ejecutar los tests definidos. Cada componente tiene diferentes módulos que se explican a continuación:

³ <http://appium.io/>

⁴ <https://developer.android.com/training/testing/espresso>

- **Setup:** este módulo se encarga de validar y extraer las características del *fichero de configuración* para los demás módulos.
- **Deployment:** este módulo toma como entrada las características obtenidas en el módulo anterior y se encarga de crear y desplegar todo el entorno virtual. Este módulo tiene una capa de abstracción que encapsula Terraform [6]. Además, orquesta el entorno virtual instalando los recursos necesarios, creando los dispositivos móviles virtuales y desplegando la aplicación.
- **Tests Execution:** este módulo se encarga de ejecutar y analizar los tests definidos en el *fichero de configuración*. Perses permite ejecutar tests de rendimiento y de interfaz de usuario. Los tests de rendimiento evalúan la QoS de la aplicación utilizando APIPecker [4]. Los tests de interfaz de usuario permiten ejecutar tests desarrollados con Espresso.
- **CI Manager:** este módulo se encarga de la integración con DevOps. Automatiza la ejecución de los diferentes módulos. Esta automatización se realiza a través de un flujo de trabajo definido con GitHub Actions [5].

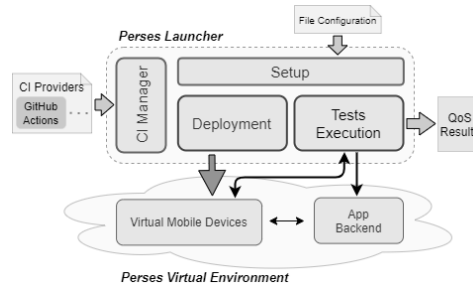


Fig. 1. Arquitectura de Perses.

2.1 Caso de uso de la demo

En esta demostración, mostraremos a los asistentes cómo funciona el marco Perses evaluando la QoS de una aplicación móvil distribuida. Para ello, evaluaremos una aplicación para ayudar a frenar la pandemia del Covid-19. Esta aplicación proporciona el riesgo de contagio de una persona.

La aplicación se compone de un servidor y el lado móvil. El lado del servidor es el punto de acceso para que los usuarios obtengan el porcentaje de riesgo y, por lo tanto, también es responsable de calcular este riesgo. El lado móvil se encarga de guardar la localización de cada usuario y proporcionar al cloud los lugares/regiones en los que ha estado.

3 Conclusiones

La QoS es una de las dimensiones clave para el éxito o el fracaso de cualquier aplicación. En los últimos años, las capacidades de los dispositivos móviles han

aumentado considerablemente, dando lugar a aplicaciones con computación distribuida en las que tanto el cloud como el lado móvil pueden tener componentes de computación y almacenamiento con el objetivo de mejorar aún más la QoS. Sin embargo, esta calidad depende de diferentes entidades y sus interacciones, que deben ser analizadas cuidadosamente.

En este trabajo, presentamos un framework para el despliegue de un entorno virtual personalizado para evaluar la QoS de las aplicaciones móviles distribuidas. Actualmente, trabajamos en el análisis del impacto del uso.

Agradecimientos

Este trabajo ha sido parcialmente financiado por los proyectos RTI2018-094591-B-I00 (MCI/AEI/FEDER,UE), RTI2018-101204-B-C21, el Proyecto 4IE+ (0499-4IE-PLUS-4-E) financiado por Interreg V-A España-Portugal (POCTEP) 2014-2020, por la red RCIS (TIN2016-81978-REDT), por la Consejería de Economía, Ciencia y Agenda Digital de la Junta de Extremadura (GR18112, IB18030), por la Junta de Andalucía (US-1264651) y el Fondo Europeo de Desarrollo Regional.

References

1. Azure, M.: App center test. <https://docs.microsoft.com/en-us/appcenter/test-cloud/>, (Accessed on 28/10/2020)
2. Bass, L., Weber, I., Zhu, L.: DevOps: A software architect's perspective. Addison-Wesley Professional (2015)
3. Berrocal, J., Garcia-Alonso, J., Vicente-Chicote, C., Hernández, J., Mikkonen, T., Canal, C., Murillo, J.M.: Early analysis of resource consumption patterns in mobile applications. *Pervasive and Mobile Computing* **35**, 32 – 50 (2017). <https://doi.org/https://doi.org/10.1016/j.pmcj.2016.06.011>, <http://www.sciencedirect.com/science/article/pii/S1574119216300797>
4. Fernandez, P.: API Pecker. <https://www.npmjs.com/package/apipecker>, (Accessed on 28/10/2020)
5. Github: Github actions. <https://github.com/features/actions>, (Accessed on 28/10/2020)
6. HashiCorp: Terraform. <https://www.terraform.io/>, (Accessed on 28/10/2020)
7. Kim, H.J., Lee, D.H., Lee, J.M., Lee, K.H., Lyu, W., Choi, S.G.: The qoe evaluation method through the qos-qoe correlation model. In: 2008 Fourth International Conference on Networked Computing and Advanced Information Management. vol. 2, pp. 719–725 (2008). <https://doi.org/10.1109/NCM.2008.202>
8. Laso, S., Berrocal, J., García-Alonso, J., Canal, C., Manuel Murillo, J.: Human microservices: A framework for turning humans into service providers. *Software: Practice and Experience* (2021)
9. Perses-org: Perses. <https://github.com/perses-org/perses>, note = (Accessed on 28/10/2020)
10. Service, A.W.: Aws device farm. <https://aws.amazon.com/device-farm/>, (Accessed on 28/10/2020)
11. Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., Jue, J.P.: All one needs to know about fog computing and related edge computing paradigms: A complete survey. *Journal of Systems Architecture* (2019)