

Computación Serverless para tratamiento de datos provenientes de dispositivos de IoT

Nelson Rodríguez¹, María Murazzo¹, Diego Medel¹, Lorena Parra¹, Ana Laura Molina¹, Federico Sánchez¹, Fabiana Piccoli², Miguel Méndez Garabetti³, Daniel Arias Figueroa⁴, Martín Tello⁵, Cynthia Aguilera⁶, Hernán Atencio⁶

¹ Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.

² Departamento de Informática - F.C.F.M. y N – UNSL

³ Instituto de Investigaciones, Facultad de Informática y Diseño, Universidad Champagnat

⁴ Consejo de Investigación, FCE, Universidad Nacional De Salta

⁵ Alumno con Beca de CIN – F.C.E.F. y N. – U.N.S.J.

⁶ Alumno Avanzado Licenciatura en Sistemas de Información y Cs. de la Computación - F.C.E.F. y N. - U.N.S.J.

Complejo Islas Malvinas. Cereceto y Meglioli. 5400. Rivadavia. San Juan, 0264 4234129

nelson@iinfo.unsj.edu.ar, marite@unsj-cuim.edu.ar, vdiego.unsj@hotmail.com, lorenaparra152@yahoo.com.ar, almm95@gmail.com, fgsanchez@unsj-cuim.edu.ar, mpiccoli@unsl.edu.ar, mendez-garabettimiguel@uch.edu.ar, daaf@cidia.unsa.edu.ar, martinl.tello@gmail.com, cynaguilera95@gmail.com, hernan.atencio.98@gmail.com

Resumen

Cloud Computing se ha consolidado como una arquitectura que permite la provisión de recursos bajo demanda y de forma elástica; es ampliamente utilizada y ha demostrado su efectividad. Por otro lado Internet de las cosas ha comenzado a tener una gran adopción fruto de la emergente industria 4.0 y de las smart cities entre otras aplicaciones Serverless en un nuevo paradigma de desarrollo de software en el Cloud, que surge como una evolución del mismo, partiendo de una arquitectura monolítica, pasando por máquinas virtuales, luego contenedores y por último serverless, que en algunos casos se conoce como función como servicios. La misma está enfocada en proveer una arquitectura que permita la ejecución de funciones arbitrarias con mínima sobrecarga en la administración del servidor y soportada bajo la programación orientada a eventos. La cantidad y variedad de datos provenientes del Edge, necesitan ser procesados y en algunos casos en tiempo real. Estrategias como CloudIoT o Fog computing

se han propuestos pero debido a los costos de los dispositivos fog enabled o a la latencia hacia el Cloud, estas soluciones son aplicables solo en determinados casos. Serverless provee la ejecución individual de funciones Cloud y es adecuada para el procesamiento de datos (ETL), procesamiento de datos provenientes de dispositivos de IoT (donde claramente favorece la escalabilidad) y computación científica. Esta unión de tecnologías presenta varios desafíos y oportunidades a resolver, entre ellas la redefinición de DevOps para serverless (NoOps), testing e interoperabilidad, performance, seguridad y análisis de costo entre otras. La evaluación de la eficiencia de las arquitecturas serverless y el estudio detallado de las mismas, permitirá plantear nuevos casos de uso, proponer estrategias para el procesamiento IoT y redefinir el proceso de desarrollo y operación

Palabras clave: *Serverless Computing, Edge Computing, IoT, Cloud Computing, Distributed Computing,*

Contexto

El presente trabajo se encuadra dentro del área de I/D Procesamiento Distribuido y Paralelo y es un resultado del proyecto de investigación: Computación Serverless para el tratamiento de datos provenientes de dispositivos de IoT, que está en proceso de evaluación para el período 2020-2021. Asimismo el grupo de investigación viene trabajando en proyectos relacionados con la computación distribuida y de alta performance desde hace más de 19 años. Como continuación del proyecto anterior: Orquestación de Servicios para la Continuidad Edge al Cloud, se sigue trabajando con los investigadores de otras universidades, lo cual favorece notablemente a todos las instituciones participantes.

Introducción

La computación serverless es un paradigma emergente en el que las aplicaciones de software se descomponen en múltiples funciones independientes sin estado [1] [2]. Las funciones solo se ejecutan en respuesta a acciones desencadenantes (como interacciones de usuario, eventos de mensajería o cambios en la base de datos), y se pueden escalar de forma independiente.

Por lo tanto, la computación serverless, también es a veces denominada función como servicio (FaaS). En este enfoque, casi todas las preocupaciones operativas son abstraídas lejos de los desarrolladores. De hecho, los desarrolladores simplemente escriben código e implementan sus funciones en una plataforma sin servidor. La plataforma se encarga de la ejecución de la función, el almacenamiento y la infraestructura de contenedor, redes y tolerancia a fallas. Adicionalmente, la plataforma serverless se encarga de escalar las funciones según la demanda real.

La computación serverless ha sido identificada como un enfoque prometedor para varias aplicaciones, como el análisis de datos en el borde de la red [3]. En consecuencia, una plataforma sin servidor maneja el ciclo de vida, la ejecución y escalada de las funciones reales; estas necesitan correr solo cuando son invocadas o activadas por un evento. Por lo tanto, el mayor beneficio de la computación

serverless son las pocas preocupaciones operativas y de gestión y la utilización eficiente de los recursos [4].

Un modelo basado en funciones es particularmente adecuado para ráfagas, uso de CPU intensivo, cargas de trabajo granulares. Actualmente, los casos de uso de FaaS varían ampliamente, incluido el procesamiento de datos, el procesamiento de flujo, la computación de borde (IoT) y la computación científica [5] [6]. Con la continua experimentación en torno a FaaS, es probable que otros casos de uso surjan en un futuro.

En los últimos años, el paradigma de la función como servicio ha revolucionado la forma en que resuelve el procesamiento de eventos distribuidos. Los desarrolladores pueden implementar y ejecutar funciones controladas por eventos en el Cloud que se escalan bajo demanda. La ventaja de la alta disponibilidad y poca administración, son muy valoradas por la conducción de las empresas.

El término "sin servidor" es confuso ya que hay tanto hardware de servidor como procesos de servidor que se ejecutan, pero la diferencia en comparación con los enfoques normales es que la organización que construye y admite una aplicación "sin servidor" no se ocupa de ese hardware o esos procesos, debido a que están subcontratando esta responsabilidad.

Los primeros usos del término aparecieron en 2012, incluido el artículo de Ken Fromm [7]. El término se hizo más popular en 2015, luego del lanzamiento de AWS Lambda, y creció en popularidad después del lanzamiento de API Gateway de Amazon en julio de 2015. En octubre de ese año hubo una charla titulada "The Serverless Company using AWS Lambda", en referencia a PlayOn! Deportes. Hacia fines de año, el proyecto de código abierto "JavaScript Amazon Web Services (JAWS)" se renombró a Serverless Framework, continuando la tendencia.

A mediados de 2016, Serverless se había convertido en un nombre dominante para esta área, dando paso al nacimiento de la serie de conferencias Serverless, y varios proveedores Cloud adoptaron el término en todo, desde marketing de productos hasta descripciones de puestos de trabajo.

La Computación Serverless

Existen varias definiciones y todas son muy similares. Se toma de referencia la del grupo de investigación The SPEC Cloud:

La computación serverless es una forma de computación en el cloud que permite a los usuarios ejecutar eventos y aplicaciones facturadas de forma granular, sin tener que abordar la lógica operativa [8].

Esta definición coloca al servidor como una abstracción, que se superpone parcialmente con la plataforma como servicio (PaaS). Los desarrolladores se centran en abstracciones de alto nivel (por ejemplo, funciones, consultas y eventos) y en crear aplicaciones que los operadores de infraestructura asignan a recursos concretos y servicios de soporte. Esto efectivamente separa las preocupaciones, con los desarrolladores enfocándose en la lógica empresarial y en las formas de interconectar elementos de la lógica empresarial en flujos de trabajo complejos. Mientras tanto, los proveedores de servicios se aseguran de que las aplicaciones están alojadas en contenedores, desplegadas, aprovisionadas y disponibles bajo demanda, mientras se factura al usuario solo por los recursos utilizados [9].

La computación serverless se puede identificar como resultado de la unión de Cloud y Microservices Architecture. Pero la evolución de Cloud ha pasado por varias etapas.

Surgió como una evolución de todas las tecnologías Cloud. Se investigaron muchas vías diferentes para encontrar formas de simplificar la gestión de aplicaciones a escala, entornos virtuales, hosts virtuales dentro de máquinas virtuales, contenedores, por nombrar algunos. Con la aparición de Docker, los contenedores comenzaron a ser tenidos en cuenta y comenzaron a ofrecerse plataformas que permitían desplegar contenedores.

De estos microservicios, surgieron nuevas ofertas de "nivel de servicio" de alto nivel como Backend-as-a-Service (BaaS) y Function-as-a-Service (FaaS). Estos representaban los albores de la computación serverless.

La determinación exacta de lo que es "sin servidor" está en debate, pero se puede definir esto como: cualquier aplicación o servicio para

el cual la empresa solo proporciona lógica de aplicación, que se ejecuta en una infraestructura que no administra. Los paradigmas claves detrás de la arquitectura empresarial y las aplicaciones que admite están cambiando rápidamente, presentando nuevas oportunidades de escalabilidad y agilidad.

Adoptar nuevos paradigmas será un desafío en el contexto de arquitecturas heredadas monolíticas y prácticas de TI arraigadas. Pero con la computación serverless puede mejorar enormemente la capacidad de su organización para reaccionar ante los cambios inevitables pero imprevisibles que aún están por venir.

La computación serverless es adecuada para aplicaciones de corta duración sin estado dirigidas por eventos, por ejemplo microservicios, backends IoT móviles, procesamiento de flujo modesto, bots e integración de servicios. No son indicados para cálculo numéricos de larga duración con estado como son: bases de datos, entrenamiento de aprendizaje profundo, análisis de flujo de servicio pesado, simulación numérica y vídeo transmitido en vivo [9].

La computación serverless se está utilizando para admitir una amplia gama de aplicaciones. Desde una perspectiva funcional, serverless y las arquitecturas más tradicionales pueden usarse indistintamente. La determinación de cuándo usar serverless estará influenciada por otros requisitos no funcionales como la cantidad de control sobre las operaciones requeridas y el costo y las características de la carga de trabajo de la aplicación, por ejemplo. Para los proveedores, la computación serverless promueve el crecimiento empresarial, ya que facilita la programación, ayuda a atraer nuevos clientes y ayuda a los clientes a hacer un mayor uso de lo que ofrece el cloud. Por ejemplo, encuestas realizadas sobre el uso de tecnología indicaron que: el 41% de los encuestados usa tecnología serverless, mientras que el 31% no. Un 28% adicional tiene planes de usarlo dentro del próximo año. De los encuestados indicaron que el 70% utiliza AWS Lambda, seguido por Apache OpenWhisk (12%), Azure Functions (12%) y Google Cloud Functions (13%) [10].

La Arquitectura Serverless está estructurada de forma análoga al cloud tradicional, pero incluye una capa intermedia que se encuentra entre las aplicaciones y la plataforma base del cloud, lo que simplifica la programación. Las FasS en el cloud proporcionan información general, computan y se complementan con un ecosistema de ofertas especializadas de BaaS como almacenamiento de objetos, bases de datos o mensajes, entre otros. Por ejemplo, una aplicación serverless en AWS podría usar Lambda con S3 (almacenamiento de objetos) y DynamoDB (base de datos NoSQL), mientras que una aplicación en el Cloud de Google podría usar Cloud Functions con Cloud Firestore (base de datos de back-end móvil) y Cloud Pub / Sub (mensajería). Serverless también incluye servicios de big data como AWS Athena y Google BigQuery. La plataforma base del cloud subyacente incluye máquinas virtuales, privadas, redes privadas, almacenamiento virtualizado, gestión de identidad y acceso, así como facturación y seguimiento [11].

En el modelo tradicional, las aplicaciones consisten en máquinas virtuales grandes y multifuncionales con minutos de tiempos de aprovisionamiento. Estas máquinas virtuales actúan como cajas negras y son difíciles de modelar y predecir para los operadores. Aunque las aplicaciones suelen estar bloqueadas por uno de los recursos en una parte de la aplicación, los operadores sólo pueden escalar la aplicación completa para resolver el cuello de botella. Es posible eliminar algunos de estos problemas, pero requiere que el usuario vuelva a diseñar la aplicación, típicamente como microservicios. Desde la perspectiva de un proveedor, la computación serverless brinda una oportunidad para controlar toda la pila de desarrollo, reducir los costos operativos mediante una optimización eficiente y gestión de recursos en el cloud y la habilitación de un ecosistema sin servidor que fomenta el despliegue de servicios adicionales [12].

Si bien las primeras ventajas de serverless eran la económica y la sencillez en el desarrollo de aplicaciones, posteriormente surgieron nuevos beneficios como son: la escalabilidad, el

código puede ejecutarse desde cualquier lugar y se adecúa al desarrollo de productos ágiles.

Por supuesto que existen desventajas como son: la depuración puede resultar más difícil de llevar a cabo, existe pérdida del control operativo, surgen nuevos problemas al utilizar APIs de terceros, nuevos riesgos de seguridad aparecen y los desarrolladores necesitan de un cambio de mentalidad, entre otras.

Líneas de Investigación, Desarrollo e Innovación

El tipo de investigación que se usará para llevar a cabo este proyecto comenzará con una investigación documental que permita definir las bases de conocimiento sobre la “Computación Serverless” y su convergencia al IoT. Esto será posible mediante la confección de revisiones sistemáticas y mapeos sistemáticos de la literatura a fin de definir el problema científico a solucionar.

Posteriormente la investigación se conducirá de forma hipotética mixta (experimental y deductiva). Esta forma de trabajo permitirá la selección de una plataforma de despliegue serverless sobre la cual se realizará el tratamiento de datos provenientes de los dispositivos IoT. Realizada esta selección, se fijará como principal variable el grado de escalabilidad horizontal lograda con la solución desarrollada.

Para lograr esto, se usará un método empírico analítico, que permita determinar si mediante el uso de la arquitectura serverless, se logra mejor escalabilidad en el tratamiento de datos provenientes de los dispositivos IoT.

Resultados y Objetivos

Resultados Obtenidos

Durante los últimos doce años se trabajó en el área de Computación de Altas Prestaciones y distribuidas, en particular sobre análisis de diversas arquitecturas paralelas y distribuidas, tales como: Cloud Computing, Cluster de commodity, arquitecturas distribuidas y paralelas de bajo costo y fog computing. Dicha experiencia impulsó esta línea de investigación. El grupo ha realizado varias

publicaciones durante el último año: doce trabajos de investigación en Congresos y Jornadas, se realizaron tres publicaciones en revistas científicas y se transfirieron los resultados mediante conferencias en eventos científicos.

Se han aprobado siete tesinas de grado, se incorporó un becario de investigación categoría alumno y otra beca está en evaluación.

Objetivos

El objetivo del grupo de investigación es Evaluar la eficiencia de las arquitecturas serverless para el tratamiento de datos proveniente de dispositivos de IoT, analizando variadas estrategias y evaluando cómo se comportan los parámetros de desempeño comparado con el modelo de cloud tradicional.

Formación de Recursos Humanos

El equipo de trabajo está compuesto por los nueve investigadores que figuran en este trabajo de las universidades Nacional de San Luis, Champagnat, Nacional de Salta y San Juan, un becario CIN y cinco alumnos.

Se está desarrollando una tesis doctoral sobre paralelismo híbrido y Big Data, seis tesinas de grado en el área de Serverless computing, Concurrencia y Computación distribuida y una tesis de maestría en áreas afines. Además se espera aumentar el número de publicaciones. Por otro lado también se prevé la divulgación de varios temas investigados por medio de cursos de postgrado y actualización o publicaciones de divulgación y asesoramiento a empresas y otros organismos del estado.

Referencias

[1] Adzic, G., & Chatley, R. (2017). Serverless computing: economic and architectural impact. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (pp. 884-889). ACM.

[2] Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V. & Suter, P. (2017). Serverless computing: Current trends and open problems. In Research Advances in Cloud Computing (pp. 1-20). Springer, Singapore.

[3] Nastic, S., Rausch, T., Scekcic, O., Dustdar, S., Gusev, M., Koteska, B. & Prodan, R. (2017). A serverless real-time data analytics platform for edge computing. *IEEE Internet Computing*, 21(4), 64-71.

[4] Mohanty, S. K., Premsankar, G., & Di Francesco, M. (2018). An Evaluation of Open Source Serverless Computing Frameworks. In *CloudCom* (pp. 115-120).

[5] Gottlieb, N. (2016). State of the Serverless Community Survey Results. <https://serverless.com/blog/state-of-serverless-community/>.

[6] Jonas, E., Pu, Q., Venkataraman, S., Stoica, I., & Recht, B. (2017). Occupy the cloud: Distributed computing for the 99%. In *Proceedings of the 2017 Symposium on Cloud Computing* (pp. 445-451). ACM.

[7] Fromm, K. (2012). <https://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>

[8] Van Eyk, E., Iosup, A., Seif, S., & Thömmes, M. (2017). The SPEC cloud group's research vision on FaaS and serverless architectures. In *Proceedings of the 2nd International Workshop on Serverless Computing* (pp. 1-4). ACM.

[9] Van Eyk, E., Toader, L., Talluri, S., Versluis, L., Uță, A., Iosup, A. (2018). Serverless is more: From paas to present cloud computing. *IEEE Internet Computing*, 22(5), 8-17.

[10] Conway, S. (2017) Cloud Native Technologies Are Scaling Production Applications. CNCF website: <https://www.cncf.io/blog/2017/12/06/cloud-native-technologies-scaling-production-applications..>

[11] Jonas, E., Schleier-Smith, J., Sreekanti, V., Tsai, C. C., Khandelwal, A., Pu, Q., ... & Gonzalez, J. E. (2019). Cloud programming simplified: a Berkeley view on serverless computing. arXiv preprint arXiv:1902.03383.

[12] Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2017). Serverless programming (function as a service). In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (pp. 2658-2659). IEEE. DOI 10.1109/ICDCS.2017.305.