

Optimización en el desarrollo de aplicaciones Serverless en entornos distribuidos

Nelson Rodríguez¹, María Murazzo¹, Diego Medel¹, Daniel Arias Figueroa², Lorena Parra¹, Ana Laura Molina¹, Federico Sánchez¹, Adriana Martín¹, Hernán Atencio³, Martín Gómez³

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

² Consejo de Investigación, FCE, Universidad Nacional De Salta

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, daaf@cidia.unsa.edu.ar, lorenaparra152@yahoo.com.ar, almm95@gmail.com, fgsanchez@unsj-cuim.edu.ar, adrianamartin1@gmail.com, hernan.atencio.98@gmail.com, martinsj0811@gmail.com

Resumen

Los últimos años han sido testigos de avances sin precedentes en el campo de Cloud Computing. Este modelo se refiere al acceso de red ubicuo, conveniente y bajo demanda a un grupo compartido de recursos informáticos. Como una evolución constante, aparece Serverless como un nuevo paradigma de desarrollo de software en el Cloud, que surge a partir de una arquitectura monolítica y luego pasando por máquinas virtuales y contenedores para finalizar en serverless, que en algunos casos se conoce como función como servicios, aunque también incluye Backend como servicio, lo cual en general son servicios de terceros disponibles en el Cloud. La misma está enfocada en proveer una arquitectura que permite 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 y por lo tanto deben resolverse varios desafíos y problemas abiertos. Debido a que las investigaciones apuntan a diversas áreas como son: arquitectura y hardware, administración de recursos, modelo de negocios y desarrollo

de aplicaciones. Debido a que las aplicaciones serverless son altamente distribuidas, donde cada función realiza una tarea enfocada y depende de un sinnúmero de otros servicios para el resto., el objetivo de esta línea de investigación es resolver problemáticas vinculadas al desarrollo, que generen alta latencia, dificulten el paralelismo de funciones serverless o no se adapten a los principios DevOps, entre otros.

Palabras clave: *Serverless Computing, Distributed Computing, FaaS, , Cloud Computing, Serverless Programming*

Contexto

El presente trabajo se encuadra dentro del área de I/D Procesamiento Distribuido y Paralelo y es una de las líneas de investigación del proyecto: Computación Serverless para el tratamiento de datos provenientes de dispositivos de IoT, cuya propuesta ha sido aprobada y está en desarrollo para el período 2020-2021. Asimismo el grupo de investigadores viene trabajando en proyectos relacionados con la computación distribuida y de alta performance desde hace más de 20 años. Como continuación del proyecto

anterior: Orquestación de Servicios para la Continuidad Edge al Cloud, se sigue trabajando con investigadores de otras universidades, lo cual favorece notablemente a todos las instituciones participantes.

Introducción

La computación Serverless es una tecnología emergente con un impacto creciente en nuestra sociedad y una mayor adopción tanto por parte de la academia como de la industria [1]. Es un paradigma emergente en el que las aplicaciones de software se descomponen en múltiples funciones independientes sin estado [2] [3]. Las funciones se ejecutan en contenedores casi sin estados y solo en respuesta a acciones desencadenantes (como interacciones de usuario, eventos de mensajería o cambios en la base de datos), se pueden escalar de forma independiente y pueden ser efímeras (pueden durar una invocación) y están completamente administrados por el proveedor de Cloud.

Debido a que la implementación se realiza mediante la plataforma de funciones, la computación serverless, también es denominada función como servicio (FaaS). En este enfoque, casi todas las preocupaciones operativas son abstraídas lejos de los desarrolladores. Los cuales en principio 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, también se encarga de escalar las funciones según la demanda real.

En la mayoría de los casos, se puede escribir funciones en el lenguaje favorito del programador (Node.js, Python, Go, Java y más) y utilizar herramientas de contenedor y serverless, como AWS SAM o la CLI de Docker, para compilar, probar e implementar las funciones.

La computación serverless ha sido identificada como un enfoque prometedor para varias aplicaciones, como el análisis de datos en el borde de la red [4]. 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 lo tanto, el mayor beneficio de esta computación son las pocas preocupaciones operativas y de gestión y la utilización eficiente de los recursos [5].

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 [6] [7]. 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 [8]. 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. Posteriormente 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.

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 [9].

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 [10].

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.

Se investigaron muchas vías diferentes para simplificar la gestión de aplicaciones a escala, entornos virtuales, hosts virtuales dentro de máquinas virtuales, contenedores, por nombrar algunos. Con la aparición de los contenedores 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.

Optimización del desarrollo Serverless

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 [10].

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 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 [11].

Si bien las primeras ventajas 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.

El desarrollo de aplicaciones presenta una gran cantidad de problemas a resolver. De los cuales se van a enunciar algunos de ellos y se espera que sean resueltos o minimizados por los trabajos del grupo de investigación.

Si bien los frameworks serverless han permitido a los desarrolladores crear e implementar aplicaciones más rápidamente, también traen un conjunto de desafíos de depuración en comparación con los frameworks tradicionales.

Además, debido a que la arquitectura sin servidor todavía está en sus etapas formativas, hay enfoques y herramientas en evolución que se integran sin problemas con una plataforma FaaS, al tiempo que ayudan con la recopilación constante de datos [12].

En cuanto a la configuración, existen muchas dificultades para realizar la supervisión de

sistemas distribuidos dinámicamente para identificar los posibles cuellos de botella que puedan ocurrir.

También se produce un aumento de la latencia, que es causada por comunicación entre intercomponentes complejos

Aparece la necesidad de integrar rápida y eficientemente los microservicios asociados. Pueden ocurrir errores en el Cloud o en el Edge, pero en el borde, sin embargo, las técnicas de rollback de falla son muy limitadas y deben lograrse soluciones más robustas que para lograr el nivel deseado de resiliencia [13].

Otro problema es la gestión eficiente de arranques en frío, dado que se produce un retraso introducido a medida que la infraestructura subyacente aprovisiona dinámicamente nuevos recursos en tiempo de ejecución y además puede generar, si la cantidad de funciones es elevada, latencias adicionales y errores de sistema.

Por otro lado no existen herramientas de seguimiento y depuración adecuadas, tampoco hay expertización en este tipo de programación para atender la refactorización de los sistemas existentes, la capacidad de integrar y componer, gestionar y mantener, diseñar funciones con estado y sin estado, la gestión de simultaneidad, la granularidad de código de forma optimizada, diseño del sistema de recuperación y adaptación de los principios de DevOps entre otras. [14].

Líneas de Investigación, Desarrollo e Innovación

El tipo de investigación que se está utilizando para llevar a cabo esta línea de trabajo ha comenzado hace un año con una análisis documental que permitió definir las bases de conocimiento sobre la “Computación Serverless” y su convergencia al Edge. Esto posibilitó mediante la confección de revisiones sistemáticas y mapeos sistemáticos de la literatura, encontrar cuales son los problemas científicos y desafíos que se van a tratar de 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 del Edge y la comunicación entre las diferentes funciones, debido a que en entornos altamente distribuidos la comunicación entre los componentes es sumamente importante para evaluar. Posteriormente, se seleccionarán los problemas a resolver, se fijará el orden y prioridad en analizar e investigar cada uno de ellos. El método a utilizar será empírico analítico, pero puede variar en función de cada problema en particular.

Resultados y Objetivos

Resultados Obtenidos

Durante los últimos trece 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. En el último año se inició un proyecto sobre Srverless Computing, cuya experiencia impulsó la línea de investigación del presente trabajo. El grupo ha realizado varias publicaciones en esta área: seis 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 tres 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 analizar y resolver los diferentes problemas y retos que surgen y que dificultan la eficiencia a nivel del desarrollo (en particular programación) de aplicaciones serverless, teniendo en cuenta entorno distribuidos como Edge a Cloud o datos provenientes de dispositivos IoT. De esta forma poder optimizar el desarrollo aplicando variadas estrategias y evaluando cómo se comportan los parámetros de desempeño y eficiencia.

Formación de Recursos Humanos

El equipo de trabajo de esta línea de investigación está compuesto de ocho investigadores que figuran en este trabajo de las universidades Nacional de San Juan y Nacional de Salta y dos alumnos de grado. Además, el proyecto marco donde se está desarrollando esta propuesta incluye a tres investigadores más de la Nacional de San Luis, de la Universidad Champagnat y de la Universidad Nacional de San Juan y tres alumnos de grado.

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] IDC, FutureScape: Worldwide IT Industry 2019 Predictions," <https://www.idc.com/getdoc.jsp?containerId=US44403818>, Oct 2018.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] Mohanty, S. K., Premsankar, G., & Di Francesco, M. (2018). An Evaluation of Open Source Serverless Computing Frameworks. In CloudCom (pp. 115-120).
- [6] Gottlieb, N. (2016). State of the Serverless Community Survey Results. <https://serverless.com/blog/state-of-serverless-community/>.
- [7] 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.
- [8] Fromm, K. (2012). <https://readwrite.com/2012/10/15/why-the-future-of-software-and-apps-is-serverless/>
- [9] 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.
- [10] 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.
- [11] 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.
- [12] Rodríguez N., et al (2020). Computación Serverless para tratamiento de datos provenientes de dispositivos de IoT. WICC 2020.
- [13] Aslanpour M. and Toosi A. et al. (2021). Serverless Edge Computing: Vision and Challenges. ACSW '21, February 1–5, 2021, Dunedin, New Zealand.
- [14] Arokia Paul Rajan R (2020) A review on serverless architectures - function as a service (FaaS) in cloud computing. TELKOMNIKA Telecommunication, Computing, Electronics and Control. Vol. 18, No. 1, February 2020, pp. 530~537