Procesamiento de datos masivos en un cloud público

Maria Murazzo, Nelson Rodriguez, Pablo Gomez, Miguel Guevara

Universidad Nacional de San Juan, Facultad de Ciencias Exactas, Físicas y Naturales,
Departamento de Informática
Complejo Universitario Islas Malvinas (CUIM), Rivadavia, San Juan, Argentina

marite@unsj-cuim.edu.ar, nelson@iinfo.unsj.edu.ar, pablo6189@gmail.com, migueljoseguevaratencio@gmail.com,

Abstract

Los avances tecnológicos han permitido que se generen grandes cantidades de datos, los cuales necesitan ser almacenados y procesados de manera eficiente. Surge así el paradigma Big Data, donde el principal requerimiento no solo es la capacidad de cómputo, sino el manejo en un tiempo razonable de ingentes cantidades de datos. En este contexto, Cloud ha emergido como una infraestructura que elimina la necesidad de mantener hardware costoso mediante la abstracción de recursos físicos que ofrece la virtualización. Esto genera una plataforma de recursos orquestados y bajo demanda que favorecen el almacenamiento y procesamiento de grandes volúmenes de datos. Es así que la conjunción del cloud y el big data se ha convertido en parte fundamental de la infraestructura de TI de cualquier organización que desee hacer un aprovechamiento eficiente de los datos con los que cuente. En función de esto, el presente trabajo presenta las lineas de investigación que el grupo de trabajo esta desarrollando con el objeto de obtener soluciones a los problemas de datos masivos.

Keywords: Datos Masivos, Cloud Computing, Google Cloud Platform, New SQL, Spanner

1. Introducción

Con el uso masivo de Internet, se está en presencia de un fenómeno donde el crecimiento del volumen de datos capturados y almacenados y la creciente variación en los tipos de datos, hace que las técnicas tradicionales para el procesamiento, análisis y obtención de información útil deban ser redefinidas para formular nuevas metodologías.

Frente a esta problemática se ha popularizado el término Big Data [1], el cual es usado para describir grandes conjuntos de datos (data set), que exhiben las propiedades de variedad, volumen, velocidad, variabilidad, valor y complejidad. Tratar con grandes volúmenes de datos, es un área de investigación focalizada en recolectar, examinar y procesar grandes conjuntos de datos con el objeto de descubrir patrones, correlaciones y extraer información de ellos. Estos aspectos hacen que los sistemas de cómputo convencionales sean muchas veces inapropiados para lograr un procesamiento adecuado, por lo que una alternativa puede ser considerar técnicas de computación de alta prestaciones con el fin de aumentar la velocidad de procesamiento [2].

HPC [3] es la evolución de los sistemas de cómputo convencional, los cuales permiten realizar operaciones de cómputo intensivo y mejorar la velocidad de procesamiento; involucrando diferentes tecnologías tal como los sistemas distribuidos y los sistemas paralelos. Estos entornos son ideales para resolver aplicaciones científicas, computacional-mente costosas con manejo de grandes cantidades de datos, a fin de lograr resultados en menor tiempo. La conjunción de Big Data y HPC se enfoca en la paralelización del problema mediante la distribución de los datos y la delegación del cómputo en los nodos con capacidad de procesamiento de la arquitectura.

El desafío se centra en cómo se aprovecha al máximo el potencial de la arquitectura física existente, con el fin de mejorar los tiempos de procesamiento en los algoritmos. Este objetivo se puede alcanzar mediante la implementación de una plataforma con mayor potencia de cálculo como las supercomputadoras, pero los costos de estos equipos son elevados, lo que dificulta su acceso a la comunidades científicas.

Para resolver los problemas de costo, la computación distribuida [4] es un modelo destinado a resolver problemas de cómputo masivo utilizando un gran número de computadoras organizadas sobre una infraestructura de comunicaciones distribuida. De esta manera es posible compartir recursos heterogéneos, basados en distintas plataformas, arquitecturas y lenguajes de programación, situados en distintos lugares y pertenecientes a diferentes dominios de administración sobre una red que utiliza estándares abiertos .

En función de la problemática para la cual se decide montar una arquitectura Distribuida, existen diferentes tipos de sistemas distribuidos: sistemas de cómputo distribuido, sistemas de almacenamiento distribuido y sistemas ubicuos distribuidos [5]. Para el caso de este trabajo, las investigaciones se han enfocado en los sistemas de cómputo distribuidos,

los cuales permiten realizar de manera más eficiente tareas de computación de alto rendimiento basadas en el modelo de memoria distribuida.

El propósito que se persigue es analizar, diseñar e implementar una solución computacionalmente eficiente a problemas de datos masivos mediante la aplicación de modelos de programación y técnicas de Computación de Alto Desempeño en ambientes distribuidos.

Este trabajo se organiza de la siguiente manera: en la próxima sección se explican las generalidades de las arquitecturas distribuidas. En la Sección 3 se explican las generalidades del big data como servicio. En la Sección 4 se definen los escenarios de trabajos para abordar la problemática de los datos masivos sobre cloud. En la sección final se abordan las conclusiones y futuros trabajos.

2. Arquitecturas Distribuidas

Cuando se realizan operaciones que demandan una gran cantidad de cómputo, las solución secuenciales, se convierten en una opción computacionalmente costosa; es por ello que es necesario migrar a entornos con mayor capacidad de procesamiento. Esta migración permitirá mejorar los tiempos de respuesta y, aumentar la escalabilidad y la eficiencia.

Para lograr esto, una opción es realizar una implementación que permita distribuir los datos y paralelizar el cómputo sobre una arquitectura distribuida. A tales efectos, las arquitecturas distribuidas [6], tales como cluster y cloud, proveen una infraestructura que favorece de manera eficiente y escalable el procesamiento sobre grandes cantidades de datos.

2.1 Cloud Computing

Hasta no hace mucho, el método preferido para incrementar la capacidad de procesamiento en ambientes distribuidos han sido los cluster [7]. Sin embargo, este tipo de arquitecturas presentan como principal problema el costo de inversión, mantenimiento y gestión de la infraestructura de hardware y software. Una alternativa a los cluster, es el cloud, el cual permite contar con una cantidad de recursos computacionales virtualmente infinitos, administrados por terceros y accedidos bajo demanda pagando por el uso.

Según [8], cloud es un modelo de prestación de servicios informáticos cuya principal orientación es la escalabilidad. Esto es, que desde el punto de vista de los usuarios, los servicios son elásticos; pueden crecer o recuperar su tamaño original de manera rápida y sencilla. Esta orientación permite que los usuarios que acceden a los servicios, perciban que todo funciona de manera simple y rápida, dando como resultado una experiencia más gratificante.

Gracias a estas características, cloud se ha convertido en una tecnología centrada en ofrecer cualquier recurso (bases de datos, red, procesador, etc.) y ofrecerlo como un servicio (AaaS, Anything as a Service) bajo demanda.

Uno de los servicios que es capaz de proveer el cloud son los cluster virtuales o CaaS (Cluster as a Servise). CaaS es un modelo híbrido que se crea combinando cluster y cloud para obtener mayor disponibilidad y rendimiento [9], proporcionando un alto de abstracción. Esto permite que los usuarios sólo reciban una cantidad mínima de datos operativos, ocultando todas las características de hardware y software.

Este tipo de arquitecturas virtuales permiten realizar el procesamiento de grandes cantidades de datos mediante framework que implementen el paradigma de programación distribuida, y de esta manera lograr optimización de recursos virtualizados, mediante la distribución de la carga y la paralelización de las tareas.

3. Big Data en el Cloud

La importancia y valor que los datos tienen para las organizaciones crece día a día debido a que gracias a un adecuado análisis de ellos es posible mejorar el desempeño del sistema, guiar la toma de decisiones, evaluar el riesgo, recortar costos, etc.

Con el incremento en la cantidad de datos, las tareas de administración, gestión y análisis de estos se convierte en un problema difícil de resolver con las técnicas, metodologías y herramientas tradicionales. Por esta razón, se hace necesario cambiar el paradigma con el cual los datos son tratados con el objeto de lograr un procesamiento eficiente y eficaz de ellos [10].

Tradicionalmente, la administración de los datos se ha realizado mediante bases de datos, sin embargo, las soluciones construidos alrededor de este paradigma, son incapaces de proporcionar tiempos de respuesta razonables en el manejo de grandes volúmenes de datos.

En este contexto, es necesario asegurar obtener respuestas en tiempo real o casi en tiempo real. Ante esta problemática es necesario desarrollar soluciones que permitan la gestión eficaz de grandes cantidades de datos dentro de un tiempo de procesamiento aceptable; lo cual es una tarea crítica.

Big Data-as-a-Service (BDaaS) [11]12] involucra técnicas de almacenamiento, administración, procesamiento y análisis de grandes volúmenes de datos sobre plataformas cloud

mediante APIś programables que permitan una adecuada visualización de los datos y su análisis, con el objeto de lograr eficiencia, reducir costos y lograr integración con aplicaciones ya existentes.

BDaaS provee diferentes niveles de servicios que incluyen IaaS, PaaS y SaaS, los cuales pueden ser usados e integrados a otros sistemas vía contenedores, tales como Docker [13]. De esta manera se encapsulas las características técnicas, lo que hace la gestión del big data transparente a los usuarios.

En la figura 1, se puede ver la arquitectura de BDaaS, la cual esta formada de tres capas: Big Data Infrastructure-as-a-Service (BDIaaS), Big Data Platform-as-a-Service (BDPaaS) y Big Data Analytics Software-as-a-Service (BDSaaS).

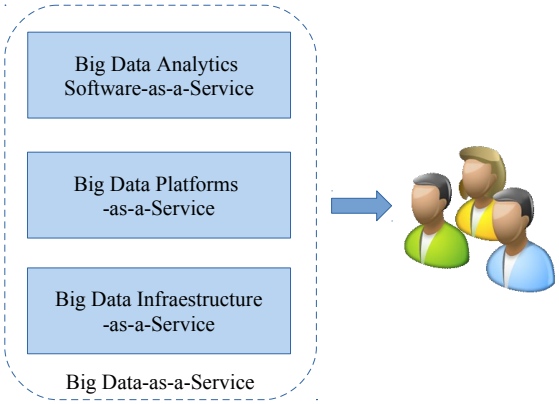

Figura 1: Arquitectura de Big Data

BDIaaS [14], se monta sobre el IaaS e incluye Storage-as-a-Service y Computing-as-a-Service, con el objeto de almacenar y procesar sobre un IaaS en el cloud los datos.

BDPaaS [15], brinda a los usuarios la posibilidad de acceder, analizar y crear aplicaciones para los datos. En esta capa involucra diferentes formas de almacenar y administrar los datos e incluye cloud storage, Data-as-a-Service (DaaS), y Database-as-a-Service (DbaaS)

BDSaaS [16], se encarga de explotar los datos estructurados y no estructurados para ofrecer resultados en tiempo real con el objetivo de permitirle al usuario la toma de decisiones. Esta capa involucra las interfaces gráficas web, apps móviles, machine learning y los ecosistemas distribuidos como Haddop [17] y Spark [18].

Esta arquitectura ofrece un marco de referencia para el desarrollo de soluciones capaces de explotar una gran cantidad de datos. Para lograr esto de manera óptima, se debe proveer acceso a todas las capas de forma interrelacionada para que el proceso de administración, gestión y explotación sea transparente al usuario final.

4. Plataforma de Trabajo

Con el objetivo de usar el cloud como plataforma para trabajar con BDaaS, se ha seleccionado Google Cloud Platform - GCP (cloud.google.com), esta plataforma es un conjunto de recursos físicos y lógicos contenidos en los datacenter de Google. Esta distribución de recursos provee importantes beneficios tales como redundancia en caso de fallas y reducción de la latencia por la selección de ellos en función a la proximidad al usuario.

Los servicios que ofrece GCP permite que se acceda a los recursos físicos mediante invocación de servicios, los cuales pueden ser combinados para armar la infraestructura necesaria.

En la figura 2, se puede ver la consola de trabajo de GCP, la cual provee una interface gráfica mediante la cual es posible administrar los proyectos y recursos.

Los servicios a los cuales se puede acceder en GCP se clasifican en: *Computing Services*, *Storage Services*, *Networking Services*, *Big Data Services*. De todos los servicios ofrecidos se hará referencia a los usados en este trabajo.

- ***Computing Services:*** *App Engine*, un PaaS que provee un SDK para el desarrollo de aplicaciones; *Container Engine*, que provee un hibrido PaaS/IaaS con soporte a clusters virtuales basados en Kubernetes (CaaS, Cluster as a Service) y, *Virtual machines*, que son instancias de maquinas físicas.
- ***Storage Services:*** *Cloud SQL*, permite manipular bases de datos SQL (MySQL o PostgreSQL) y *Cloud Spanner*, permite administrar bases de datos relacionales mission-critical.
- ***Big Data Services:*** *Cloud Dataflow*, proporciona un SDK para administrar grandes cantidades de datos batch y streaming.

Mediante estos servicios se implementarán las soluciones de BDaaS capaces de obtener información en un menor tiempo que con las plataformas tradicionales.

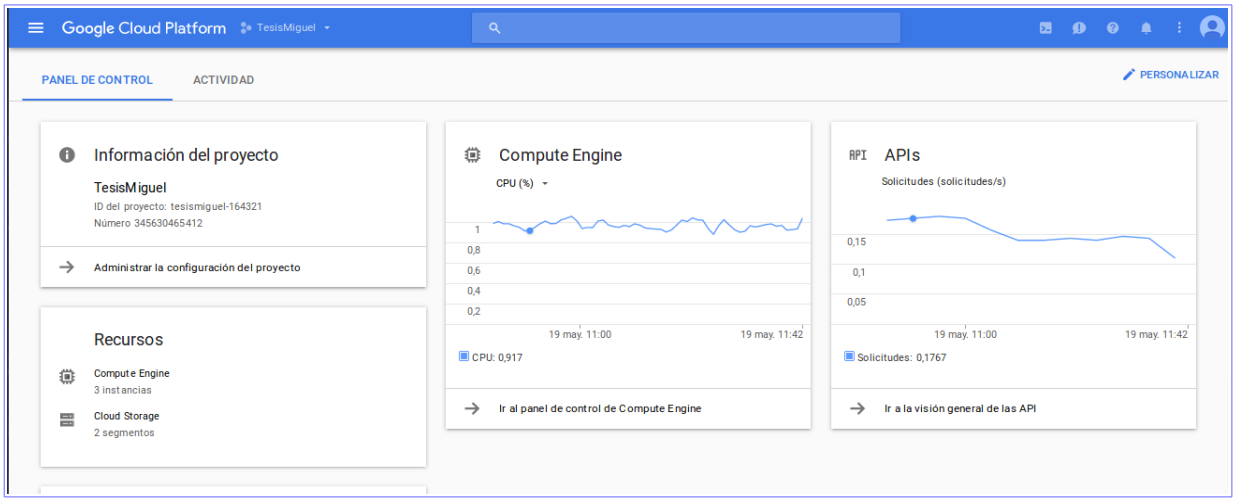

Figura 2: Consola de Google Cloud Platform

5. Caso de Estudio

Con la popularización de aplicaciones basadas en la Web, tales como juegos de multijugadores, sitios de redes sociales, redes de juego en línea, etc.; el número total de interacciones por segundo crece exponencialmente. Además, el crecimiento de la telefonía inteligentes ha creado un mercado para las aplicaciones que utilizan el teléfono como un sensor geográfico y prestan servicios basados en la localización. Conjuntamente con estas necesidades, aparece la necesidad de resolver consultas en tiempo real.

Estos requerimientos en convergencia con la gran cantidades de datos que se maneja, han generado un importante crecimiento en los requisitos de transaccionales. Atendiendo a estas problemáticas, han surgido nuevos tipos de bases de datos para suplir estas necesidades.

NewSQL [19] es una clase de gestores de bases de datos relacionales que proporcionan el mismo rendimiento que los sistemas NoSQL (no sólo SQL) y proporcionan a los administradores garantías de rendimiento *ACID (Atomicity, Consistency, Isolation, Durability)* [20].

5.1 Spanner

Spanner [21] es una base de datos escalable, distribuida globalmente, diseñada, construida y puesta en marcha en Google. En el nivel más alto de abstracción, "es una base de datos que fragmenta y distribuye datos entre varios conjuntos de nodos usando algoritmo de Paxos [22], en los datacenter de Google, dispersos geográficamente. Se utiliza replicación para tener disponibilidad global y el uso de datos locales dependiendo de la zona geográfica; los clientes son automáticamente conmutados entre réplicas.

Una característica de Spanner, que permite escalabilidad es que automáticamente fragmentara datos entre máquinas cuando el número de data o de servidores cambia, y automáticamente migra datos entre máquinas (incluso entre centros de datos) para balancear la carga y para responder a fallas. Esta forma de trabajo, permite escalado a millones de máquinas entre cientos de centros de datos y trillones de filas de bases de datos.

En Spanner los datos se almacenan en tablas con esquemas semi relacionales; estos datos están versionados y cada versión tiene automáticamente una marca de tiempo que incluye el momento de su commit; las versiones antiguas de datos están sujetas a políticas configurables de recolectores de basura. Un aspecto interesante es que Spanner tiene soporte para transacciones de propósito general y provee un lenguaje basado en SQL.

5.1.1 Implementación de Spanner

Spanner se organiza como un conjunto de zonas, las cuales se consideran conjunto de ubicaciones a través de las cuales los datos se pueden replicar. Las zonas se pueden agregar o quitar de un sistema que está en funcionamiento a medida que nuevos datacenters entran en servicio y otros se cierran. Las zonas también son la unidad de aislamiento físico: pueden existir una o más zonas en un centro de datos.

Una zona tiene un "*maestro de zona*" (zonemaster) y múltiples *servidores de Spanner* (spanservers). Los zonemasters asignan datos a los spanservers, y estos últimos entregan los datos a los clientes. El *driver de ubicación* (placement driver) maneja el movimiento de datos a través de las zonas en cuestión de minutos. El mismo se comunica periódicamente con los spanservers para encontrar datos que necesitan ser movidos, ya sea para

satisfacer restricciones actualizadas de replicación o para balanceo de carga.

Cada spanserver es responsable de múltiples instancias de datos, las cuales se almacenan en el sistema de archivos distribuido Colossus [23].

La creación de una base de datos en Spanner requiere que primero se cree una instancia, la cual será el contenedor para las bases de datos. La configuración de la instancia es simple, se puede ver en la figura 3, solo se debe especificar el nombre de la misma, un identificador, el número de nodos asignados y la ubicación de los mismos. En la figura 4, se puede ver la instancia de Spanner ya creada.

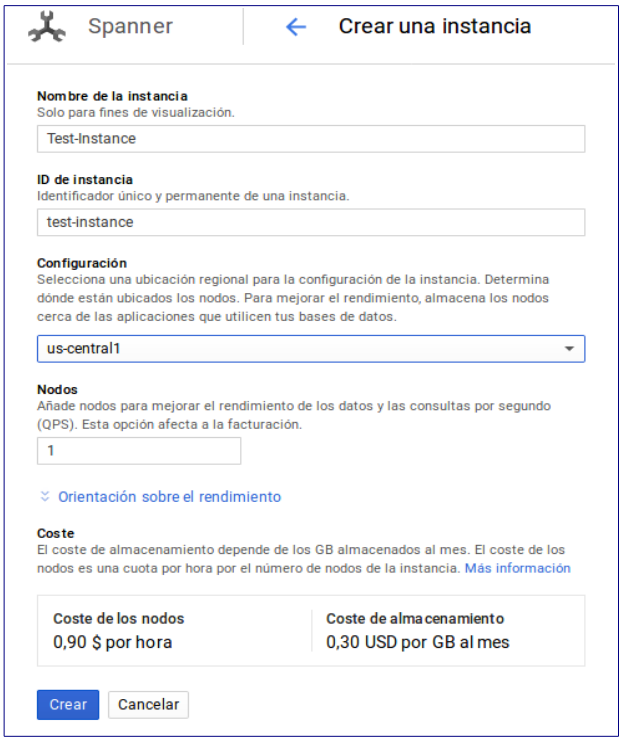

Figura 3: Creación de una Instancia de Spanner

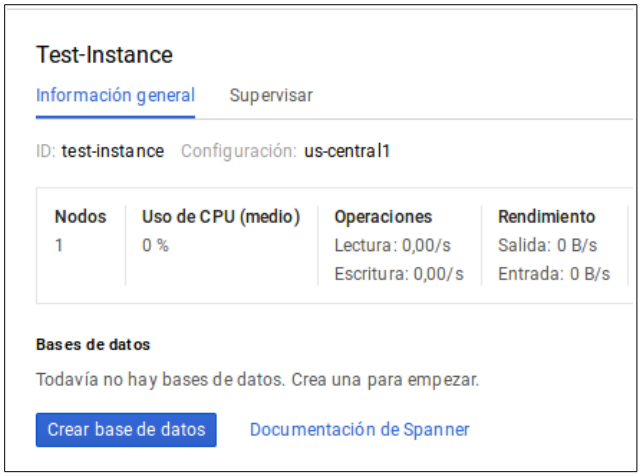

Figura 4: Instancia de Spanner ya creada

Una vez creada la instancia se puede crear la base de datos con sus respectivas tablas. De esta manera ya se pueden consultar los datos mediante SQL. En la figura 5 se puede ver la realización de una consulta utilizando la consola de Spanner, esta permite no solo ver el resultado de la consulta, sino también visualizar una explicación de cada consulta que se realizó, tal como se muestra en la figura 6.

GCP también provee una API para crear un cliente de Spanner disponible en los lenguajes GO, Java, Node.js, Python y REST.

Un aspecto a tener en cuenta que los datos de la base de datos no pueden ser insertados o eliminados desde el entorno de trabajo, por lo cual es necesario la construcción de un cliente para realizar esta tarea.

6. Conclusiones y Futuros Trabajos

El trabajo con grandes volúmenes de datos hace necesario contar con plataformas robustas de procesamiento que permitan un escalado elástico de recursos en la medida que el cómputo lo requiera. Es por ello que trabajar en un ambiente cloud provee de recursos virtualmente infinitos que favorecen este escalado.

Trabajar en un cloud publico presenta como principal ventaja la posibilidad de acceder a una cantidad de recursos, los cuales no están disponibles en los ambientes académicos. En el caso especifico de GCP, ofrece un amplio ecosistema de servicios para desarrollar aplicaciones robustas y escalables. Esta característica es de suma importancia cuando se trabaja con grandes cantidades datos y en especial con transacciones que imponen restricciones de tiempo en su ejecución. A tales efectos el paradigma NewSQL, y en particular Spanner provisto por los servicios de Storage proveen escalabilidad horizontal, alto grado de consistencia y replicación en cientos de datacenter, permitiendo un alto grado de tolerancia a fallos y disponibilidad de los datos.

Los trabajos realizados hasta el momento han cubierto las investigaciones sobre el entorno de trabajo de Spanner y la creación de instancias para trabajar con bases de datos SQL.

Aspectos que restan evaluar es la performance de este tipo de soluciones contra las implementaciones SQL y/o NoSQL. Además, resta investigar los métodos necesarios para migrar de forma transparente bases de datos SQL y/o NoSQL a Spanner.

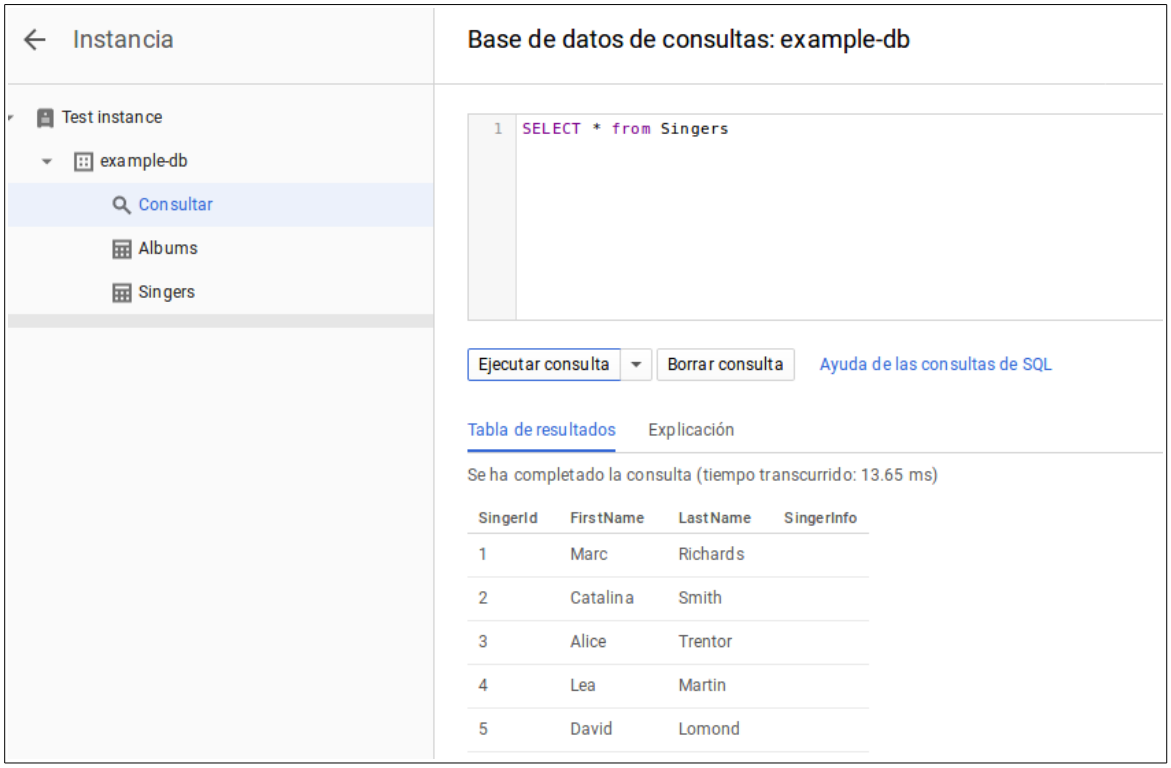

Figura 5: Consulta en la Consola de Spanner

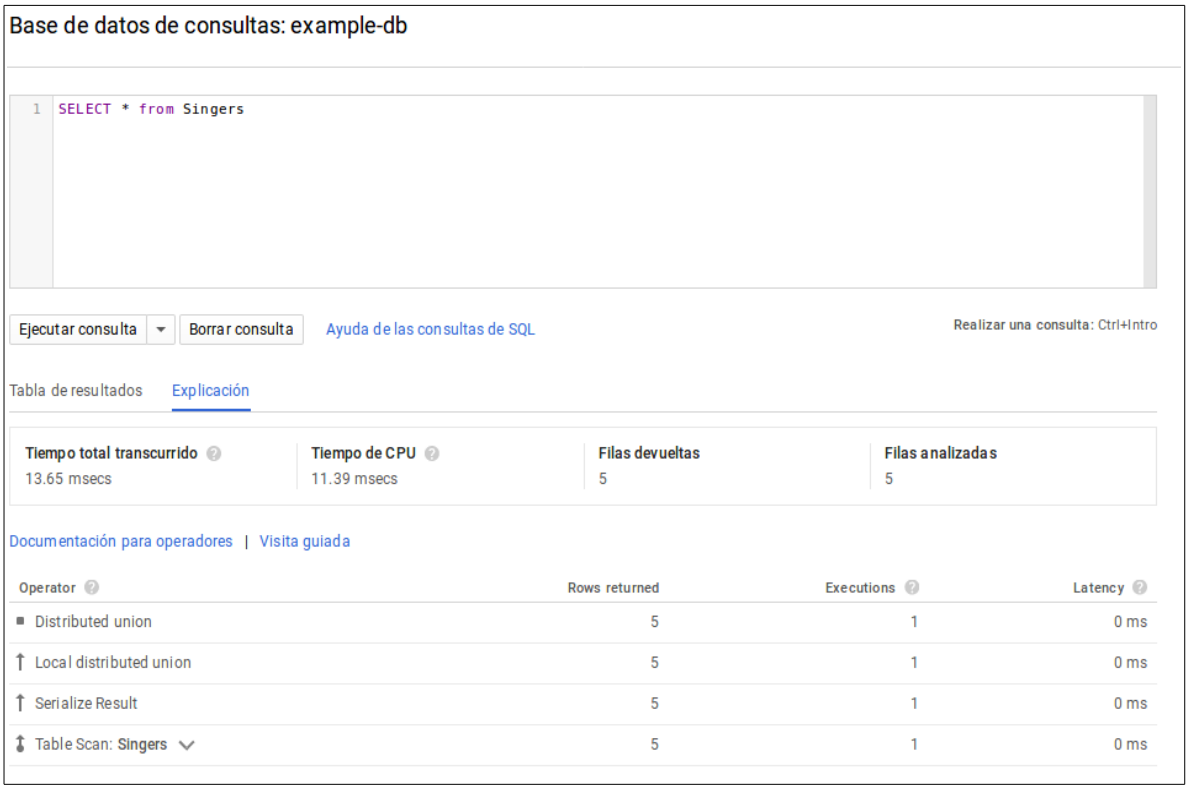

Figura 6: Detalles de la Consulta en la Consola de Spanner

7. Bibliografía

[1] Y. Zhai, Y.-S. Ong, and I. W. Tsang, “The Emerging "Big Dimensionality",” *IEEE Comput. Intell. Mag.*, vol. 9, no. 3, pp. 14–26, Aug. 2014.

[2] Y. You, S. L. Song, H. Fu, A. Marquez, M. M. Dehnavi, K. Barker, K. W. Cameron, A. P. Randles, and G. Yang, “MIC-SVM: Designing a Highly Efficient Support Vector Machine for Advanced Modern Multi-core and Many-Core Architectures,” in *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, 2014, pp. 809–818.

[3] C. J. Barrios Hernández, I. Gitler, and J. Klapp, Eds., *High Performance Computing*, vol. 697. Cham: Springer International Publishing, 2017.

[4] D. B. Kahanwal and D. T. P. Singh, “The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle,” Nov. 2013.

[5] A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms, 2/E*. 2007.

[6] F. J. Seinstra, J. Maassen, R. V. Van Nieuwpoort, N. Drost, T. Van Kessel, B. Van Werkhoven, J. Urbani, C. Jacobs, T. Kielmann, and H. E. Bal, “Jungle Computing: Distributed Supercomputing beyond Clusters, Grids, and Clouds,” 2015.

[7] M. Baker, “Cluster Computing White Paper,” Apr. 2000.

[8] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” 2011.

[9] M. A. Khoshkholghi, A. Abdullah, R. Latip, S. Subramaniam, and M. Othman, “Institute of Advanced Engineering and Science Cluster as a Service for Disaster Recovery in Intercloud Systems: Design and Modeling,” *Int. J. Cloud Comput. Serv. Sci. J.*, vol. 3, no. 3, pp. 2089–3337, 2014.

[10] D. Agrawal, S. Das, and A. El Abbadi, “Big Data and Cloud Computing: Current State and Future Opportunities *.”

[11] Xinhua E, Jing Han, Yasong Wang, and Lianru Liu, “Big Data-as-a-Service: Definition and architecture,” in *2013 15th IEEE International Conference on Communication Technology*, 2013, pp. 738–742.

[12] Z. Zheng, J. Zhu, and M. R. Lyu, “Service-Generated Big Data and Big Data-as-a-Service: An Overview,” in *2013 IEEE International Congress on Big Data*, 2013, pp. 403–410.

[13] www.docker.com, “What is Docker?” [Online]. Available: https://www.docker.com/what-docker. [Accessed: 19-May-2017].

[14] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, “Addressing big data issues in Scientific Data Infrastructure,” in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 48–55.

[15] J. Horey, E. Begoli, R. Gunasekaran, S.-H. Lim, and J. Nutaro, “Big Data Platforms as a Service: Challenges and Approach.”

[16] P. Raj and G. C. Deka, *Handbook of research on cloud infrastructures for big data analytics*. .

[17] S. B. S. Joshi, “Apache hadoop performance-tuning methodologies and best practices,” *Proc. third Jt. WOSP/SIPEW Int. Conf. Perform. Eng. - ICPE ’12*, p. 241, 2012.

[18] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, “Spark: Cluster Computing with Working Sets.”

[19] R. Kumar, N. Gupta, H. Maharwal, S. Charu, and K. Yadav, “Critical Analysis of Database Management Using NewSQL,” *Int. J. Comput. Sci. Mob. Comput.*, vol. 35, no. 5, pp. 434–438, 2014.

[20] J. Gray and A. (Andreas) Reuter, *Transaction processing : concepts and techniques*. .

[21] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, W. Hsieh, S. Kanthak, E. Kogan, H. Li, A. Lloyd, S. Melnik, D. Mwaura, D. Nagle, S. Quinlan, R. Rao, L. Rolig, Y. Saito, M. Szymaniak, C. Taylor, R. Wang, and D. Woodford, “Spanner: Google’s Globally Distributed Database,” *ACM Trans. Comput. Syst.*, vol. 31, no. 3, p. 8, 2013.

[22] J. Gray and L. Lamport, “Consensus on transaction commit,” *ACM Trans. Database Syst.*, vol. 31, no. 1, pp. 133–160, Mar. 2006.

[23] K. B. Maniar and C. B. Khatri, “Data Science: Bigtable, MapReduce and Google File System,” *Int. J. Comput. Trends Technol.*, vol. 16, no. 3, 2014.