Consistencia bajo replicación en bases de datos distribuidas

Alejandro Z. Tomsic, Marc Shapiro
Sorbonne Universites, UPMC Univ. Paris 06,
CNRS, Inria, LIP6

Descripción del curso

Hoy en día, existen aplicaciones web interactivas que hacen uso intensivo de datos para servir millones de consultas por segundo a usuarios provenientes de todos los rincones del mundo. Las bases de datos soportando estas consultas son un componente fundamental en la construcción de tales aplicaciones, ya que su buen diseño es un factor determinante en su rendimiento, particularmente en términos de latencia percibida por el usuario y throughput, esto es, cantidad de pedidos servidos por segundo. La latencia impacta directamente en la experiencia de usuario y, consecuentemente, en las ganancias netas de los proveedores de las aplicaciones. El throughput, en el costo total de propiedad y de operación, ya que determina cuál será el soporte de hardware necesario para poder cumplir con los requisitos del servicio.

La replicación de datos en diferentes zonas geográficas es una técnica muy utilizada al construir bases de datos a esta escala, ya que aumenta su capacidad de rendimiento, permite servir consultas a usuarios con baja latencia, desde la copia más próxima, y mejora su disponibilidad, ya que una consulta fallida puede ser redirigida a otra copia. Sin embargo, la replicación geográfica hace complicada la tarea de implementar las tradicionales propiedades ACID (atomicidad, consistencia, aislamiento y durabilidad) que proveen un modelo simple sobre el cual desarrollar aplicaciones. En particular, implementar modelos clásicos de consistencia (C) y aislamiento (I) tiene un costo prohibitivo en términos de overhead de comunicación y almacenamiento, que genera alta latencia y graves problemas de disponibilidad frente a tipos frecuentes de fallas. Este compromiso entre programabilidad y alta disponibilidad (y eficiencia) representa un problema fundamental en el diseño de bases de datos distribuidas.

Este curso de corta duración tiene como objetivo el de familiarizar a sus alumnos con esta problemática y el estado del arte en las técnicas utilizadas hoy en el intento de construir bases de datos que ofrezcan modelos razonablemente simples donde programar aplicaciones correctas, y que al mismo tiempo puedan proveer baja latencia, alto throughput y alta disponibilidad.

Requisitos

Conocimientos generales de sistemas operativos, redes, bases de datos y sistemas distribuidos.

Contenido

- Sistemas distribuidos
  - Definición y conceptos básicos
  - Aplicaciones y problemas fundamentales
- Bases de datos distribuidas
  - Alta disponibilidad
  - Particionado de datos
  - Replicación
    - síncrona
    - asíncrona
    - protocolos
  - Modelos de consistencia y niveles de aislamiento
    - El teorema CAP (consistencia, disponibilidad y tolerancia a particiones)
    - Consistencia fuerte
    - Consistencia débil
      - Estructuras de datos replicadas convergentes (CRDTs)
    - Protocolos y su escalabilidad
    - Consistencia mixta basada en invariantes de aplicación
- Nuestra investigación
  - Resultados recientes:
    - Cure [3], protocolo transaccional para bases de datos geográficamente replicadas
    - Antidote [1], base de datos
    - CISE ('couse I'm strong enough)[12], extrayendo requisitos de consistencia de aplicaciones.
    - Swiftcloud [21], replicación con cacheo en usuarios
    - G-DUR [4], combinando implementaciones para construir diferentes modelos de consistencia
  - Trabajos actuales [20, 19]
  - Futuras direcciones

Detalles

- El curso es principalmente teórico y será dictado en castellano.
- Tipo de evaluación: take-home teórico práctico.

References

[1] Antidote, https://github.com/syncfree/antidote.

[2] A. Adya. *Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions*. PhD thesis, Mass. Institute of Technology, Cambridge, MA, USA, Mar. 1999. Appears also as MIT Technical Report MIT/LCS/TR-786.

[3] D. D. Akkoorath, A. Tomsic, M. Bravo, Z. Li, T. Crain, A. Bieniusa, N. Preguiça, and M. Shapiro. Cure: Strong semantics meets high availability and low latency. Research Report RT-474, INRIA ; Paris 6, Feb. 2016.

[4] M. S. Ardekani, P. Sutra, and M. Shapiro. In *Proceedings of the 15th International Middleware Conference*, New York, NY, USA.

[5] M. S. Ardekani, P. Sutra, and M. Shapiro. Non-monotonic snapshot isolation: Scalable and strong consistency for geo-replicated transactional systems. In *Proceedings of the 2013 IEEE 32Nd International Symposium on Reliable Distributed Systems*, SRDS '13, pages 163–172, 2013.

[6] P. Bailis, A. Fekete, M. J. Franklin, A. Ghodsi, J. M. Hellerstein, and I. Stoica. Coordination avoidance in database systems. *Proc. VLDB Endow.*, 8(3):185–196, Nov. 2014.

[7] S. Burckhardt. *Principles of Eventual Consistency*, volume 1 of *Foundations and Trends in Programming Languages*. now publishers, October 2014.

[8] B. Charron-Bost, F. Pedone, and A. Schiper, editors. *Replication: Theory and Practice*. Springer-Verlag, Berlin, Heidelberg, 2010.

[9] 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.*, 31(3):8:1–8:22, Aug. 2013.

[10] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: Amazon's highly available key-value store. In *Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles*, SOSP '07, pages 205–220, 2007.

[11] J. Du, S. Elnikety, and W. Zwaenepoel. Clock-si: Snapshot isolation for partitioned data stores using loosely synchronized clocks. In *Proceedings of the 2013 IEEE 32Nd International Symposium on Reliable Distributed Systems*, SRDS '13, pages 173–184, Washington, DC, USA, 2013. IEEE Computer Society.

[12] A. Gotsman, H. Yang, C. Ferreira, M. Najafzadeh, and M. Shapiro. 'cause i'm strong enough: Reasoning about consistency choices in distributed systems. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL 2016, pages 371–384, New York, NY, USA, 2016. ACM.

[13] R. Guerraoui and L. Rodrigues. *Introduction to Reliable Distributed Programming*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[14] A. Lakshman and P. Malik. Cassandra: A decentralized structured storage system. *SIGOPS Oper. Syst. Rev.*, 44(2):35–40, Apr. 2010.

[15] C. Li, D. Porto, A. Clement, J. Gehrke, N. Preguiça, and R. Rodrigues. Making geo-replicated systems fast as possible, consistent when necessary. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 265–278, Berkeley, CA, USA, 2012. USENIX Association.

[16] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Don't settle for eventual: Scalable causal consistency for wide-area storage with cops. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 401–416, 2011.

[17] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski. Conflict-free replicated data types. In *Stabilization, Safety, and Security of Distributed Systems*, pages 386–400. Springer Berlin Heidelberg, 2011.

[18] Y. Sovran, R. Power, M. K. Aguilera, and J. Li. Transactional storage for geo-replicated systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 385–400, 2011.

[19] A. Tomsic, T. Crain, and M. Shapiro. Scaling geo-replicated databases to the mec environment. In *Reliable Distributed Systems Workshop (SRDSW), 2015 IEEE 34th Symposium on*, pages 74–79, Sept 2015.

[20] A. Z. Tomsic, T. Crain, and M. Shapiro. An empirical perspective on causal consistency. In *Proceedings of the First Workshop on Principles and Practice of Consistency for Distributed Data*, PaPoC '15, pages 2:1–2:3, New York, NY, USA, 2015. ACM.

[21] M. Zawirski, N. Preguiça, S. Duarte, A. Bieniusa, V. Balegas, and M. Shapiro. Write fast, read in the past: Causal consistency for client-side applications. In *Proceedings of the 16th Annual Middleware Conference*, Middleware '15, pages 75–87, New York, NY, USA, 2015. ACM.