Semana de Engenharia 2010
Guimarães, 11 a 15 de Outubro

DataDroplets: A correlation-aware large scale decentralized tuple store

Ricardo Vilaça and Rui Oliveira
CCTC – Departmento de Informática
Universidade do Minho
Braga, Portugal
Email: rmvilaca,rco@di.uminho.pt

KEYWORDS
Peer-to-Peer, DHT, Cloud Computing, Dependability

INTRODUCTION

Storage of digital data has reached unprecedented levels with the ever increasing demand for information in electronic format by individuals and organizations, ranging from the disposal of traditional storage media for music, photos and movies, to the rise of massive applications such as social networking platforms.

Until now, relational database management systems have been the key technology to store and process structured data. However, these systems based on highly centralized and rigid architectures are facing a conundrum: The volume of data currently quadruples every eighteen months while the available performance per processor only doubles in the same time period. This is the breeding ground for a new generation of elastic data management solutions, that can scale both in the sheer volume of data that can be held but also in how required resources can be provisioned dynamically and incrementally.

A renewed approach to the problem emerges through the combination of a new business model with highly decentralized, scalable, and dependable systems capable of storing and processing large volumes of data under the Cloud computing moniker. Regarding structured data management, the first generation Cloud computing services is exemplified by Google's BigTable (Chang et al., 2006), Amazon's Dynamo (DeCandia et al. 2007), Facebook's Cassandra (Lakshman and Malik 2009) and Yahoo's PNUTs (Cooper et al. 2008).

Having all started from similar requirements, aiming at supporting their respective global Web operations, they ended up providing a similar service: A simple tuple store interface, that allows applications to insert, query, and remove individual elements, forfeiting complex relational and processing facilities, and most strikingly, transactional guarantees common in traditional database management systems. By doing so, these services focus on a specific narrow tradeoff between consistency, availability, performance, scale, and cost, that fits tightly their motivating very large application scenarios (Vilaça et al. 2010a).

There are indications that the current tradeoff is much less attractive to common business needs, in which there isn't a large in-house research development team for application customization and maintenance. It is also hard to provide a smooth migration path for existing applications, even when using modern Web-based multi-tier architectures and frameworks. This is a hurdle to adoption of Cloud computing by a wider potential market and thus a limitation to the long term profitability of Cloud computing businesses.

CLOUDER

Clouder (Vilaça and Oliveira 2009) aims at advancing the state of the art by skewing the current tradeoff towards the needs of common business users, thus providing additional consistency guarantees and higher level data processing primitives that ease the migration from current database management systems. This requires determining adequate compromises and abstractions, and then devising mechanisms to realize them. Therefore Clouder provide consistent, conflict-free, data storage, and in-place processing capabilities, while allowing to leverage large computing infrastructures with distributed storage capabilities.

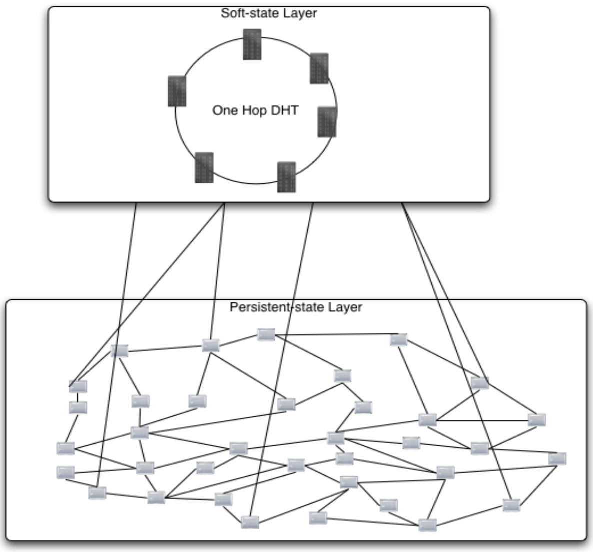

Figure 1: Clouder Architecture

To accomplish these requirements we intend to use an hybrid approach, Figure 1. On one hand the unstructured approach is fit to manage the massive scale of the system, while the structured approach is suited to a smaller scale. The latter would be used to offer an adequate interface to the client, by instance a set of well known contact nodes. Therefore, clients interact with the soft-state layer, DataDroplets, that then use the persistent-layer layer to store data. The former, is responsible for the all access concurrency control, serves as a storage cache managed as allowed by the adopted consistency criteria, and is in charge of data partitioning and mapping into the underlying persistent-state layer. The latter is responsible to massively replicate data ensuring data durability and to realize some simple processing. Each layer tackles different aspects of the system, makes specific assumptions over the computation model and exploits different techniques to data management and propagation. While the soft-state layer is assumed to be run on a subset of tens to hundreds of nodes in a controlled environment with a reasonably stable membership, the persistent-state layer is supported by a very large network with weak assumptions on the nodes and network reliability. These machines are mainly dedicated to common enterprise or academic tasks and aren't functionally or permanently dedicated to Clouder.

DATADROPLETS

DataDroplets (Vilaça et al. 2010b) is a key-value store targeted at supporting very large volumes of data leveraging the individual processing and storage capabilities of a large number of well connected computers. For some applications single tuple and range operations are not enough. These applications have multi-tuple operations that access correlated tuples. Therefore, DataDroplets supports multi-tuple access that allows to efficiently store and retrieve large sets of related data at once. Multi-tuple operations leverage disclosed data relations to manipulate sets of comparable or arbitrarily related elements. Therefore, DataDroplets extend the data model of previous tuple stores with tags that allow to establish arbitrary relations among tuples. In DataDroplets data is organized into disjoint collections of tuples identified by a string. Each tuple is a triple consisting of a unique key drawn from a partially ordered set, a value that is opaque to DataDroplets and a set of free form string tags. It is worth mentioning that the establishment of arbitrary relations among tuples can be done even if they are from different collections.

Current tuple stores offer varying levels of tuple consistency but only PNUTS and Bigtable can offer tuple atomicity. However, in both the burden is left to the application that must deal with multiple tuple's versions. In DataDroplets if an application needs atomic guarantees per tuple, it simply configures synchronous replication and it will obtain it transparently without having to maintain and deal with multiple tuple' versions.

CONCLUSION

Cloud computing and unprecedented large scale applications, most strikingly social networks such as Twitter, challenge tried and tested data management solutions. Their unfitness to cope with the demands of modern applications have led to the emergence of a novel approach: distributed tuple stores.

We present the motivation, goals, tradeoffs and architecture of Clouder and introduce DataDroplets, a distributed tuple store, that aims at shifting the trade-offs established by current tuple stores towards the needs of common business users. It provides additional consistency guarantees and higher level data processing primitives smoothing the migration path for existing applications. Specifically, DataDroplets fits the access patterns required by most current applications, which arbitrarily relate and search data by means of free-form tags.

REFERENCES

Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wal- lach, D. A., Burrows, M., et al. (2006). Bigtable: a dis- tributed storage system for structured data. In Osdi '06: Proceedings of the 7th symposium on operating systems design and implementation (pp. 205–218). Berkeley, CA, USA: USENIX Association.

Cooper, B. F., Ramakrishnan, R., Srivastava, U., Silber- stein, A., Bohannon, P., Jacobsen, H.-A., et al. (2008). PNUTS: Yahoo!'s hosted data serving platform. Proc. VLDB Endow., 1(2), 1277–1288.

DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., et al. (2007). Dynamo: Amazon's highly available key-value store. In Sosp '07: Proceedings of twenty-first acm sigops symposium on op- erating systems principles (pp. 205–220). New York, NY, USA: ACM.

Lakshman, A., and Malik, P. (2009, Ocotber). Cassandra - A Decentralized Structured Storage System. In Sosp workshop on large scale distributed systems and middle- ware (ladis) 2009. Big Sky, MT.

Vilaça, R., Cruz, F., and Oliveira, R. (2010). On the ex- pressiveness and trade-offs of large scale tuple stores (to appear). In On the move to meaningful internet systems: Otm 2010. Springer Berlin.

Vilaca, R., Oliveira, R., and Pereira, J. (2010). A correlation-aware data placement strategy for key- value stores (Tech. Rep. No. DI-CCTC-10-08). CCTC Research Centre, Universidade do Minho. Available from http://gsd.di.uminho.pt/members/rmvilaca/papers/dataDroplets

Vilaça, R., and Oliveira, R. (2009). Clouder: a flexible large scale decentralized object store: architecture overview. In Wdddm '09: Proceedings of the third work- shop on dependable distributed data management (pp. 25–28). New York, NY, USA: ACM.

AUTHOR BIOGRAPHIES

RICARDO VILAÇA was born in Braga, Portugal and went to the Universidade do Minho, where he studied Computer Science and Systems Engineering technology

and obtained his degree in 2005. He worked as a researcher in the IST FP6 project "GORDA - Open Replication of Databases". Currently, he is in the MAP-i Doctoral Programme doing his PhD entitled Large-Scale Decentralized Object Store. His e-mail address is : `rmvilaca@di.uminho.pt` and his Web-page can be found at `http://www.gsd.di.uminho.pt/members/rmvilaca`.