Managing Normalised Data in a Distributed, In-memory, Event Driven Data Store

Benjamin Stopford
Royal Bank of Scotland
90-100 Southwark Street
London
+44 7968 702589
benjamin.stopford@rbs.com

Dave Felcey
Oracle Corporation (UK)
Southgate Centre Two, 321 Wilmslow
Road, Cheadle, Cheshire SK8 3PW
+44 7909 874931
david.felcey@oracle.com

ABSTRACT

The finance industry has a long history of requiring low latency access to large time-sensitive, mission critical data sets. This demand has prompted broad investment in distributed, in-memory technologies. Similar movements away from disk-based storage can be seen in a variety of other industries [DJ09, SZ09] and in the academic database field [MS07]. It is our view that developments in distributed caching and database technologies are converging. For example technologies such as H-Store [RK08], representing a state of the art OLTP database, shares many similarities and some important differences, when compared to the solution presented here.

This talk describes ODC, a distributed, in-memory data store that has emerged from the distributed caching / NoSQL space as an alternative approach to Enterprise Application data storage and processing. We discuss two key characteristics that differentiate our approach:

- The use of a Star Schema to model data in a semi-normalized form across a data grid. This approach avoids distributed transactions when updating dimensions as well as reducing the memory consumption that can be prohibitive in volatile MVCC [PB81] architectures. The result is a store that provides both low latency and high bandwidth access to medium sized data sets.
- Both query and real time, event based views are provided by the ODC which uses messaging as a system of record. This moves the programming model away from the traditional query-only paradigm.

We further demonstrate how analysis of data volatility and volumes allow us to create a storage topology that mitigates network access during the join process, whilst retaining data normalization. Joins run across the various entities stored in the grid, referencing only in-memory data.

The schema is defined using Java object graphs with annotations that specify the storage topology. Data can either be distributed (for large volume, volatile entities, denoted Facts) or replicated (for lower volume, less volatile entities, denoted Dimensions). As the data store and client application use the same language (Java), business processing can easily be moved to Stored Procedure and Trigger equivalents, which can be executed on in-memory data directly in the grid.

ODC uses Oracle Coherence to provide a distributed data structure and durable, horizontally scalable messaging is used as the system of record. Consistency between static and asynchronous views over the data is provided via a variation of the Distributed Snapshot model [DKC85]. We describe how being event-driven enriches the store's programming model on both server and client.

We demonstrate results from the ODC's current implementation, a medium sized in-memory store of around 300 GB. A single request for a fully populated object graph completes in around a millisecond and bulk operations are network limited to the aggregated bandwidth of the cluster.

Some of the restrictions of the model are discussed including the need to simplify the contract between client and store to retain scalability. Finally we reflect back on the convergence between the ODC's approach and other prominent technologies from the database industry.

REFERENCES

[MS07] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, "The end of an architectural era: (it's time for a complete rewrite)," in VLDB '07: Proceedings of the 33rd international conference on Very large data bases, 2007, pp. 1150-1160

[JD09] Jeff Dean "Challenges in Building Large-Scale Information Retrieval Systems" Second ACM International Conference on Web Search and Data Mining, 2009

[SZ09] Shubin Zhang, Jizhong Han, Zhiyong Liu, Kai Wang, Shengzhong Feng, "Accelerating MapReduce with Distributed Memory Cache," icpads, pp.472-478, 2009 15th International Conference on Parallel and Distributed Systems, 2009

[PB81] Philip Bernstein and Nathan Goodman (1981). "Concurrency Control in Distributed Database Systems". ACM Computing Surveys.

[RK08] R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, E. P. C. Jones, S. Madden, M. Stonebraker, Y. Zhang, J. Hugg, and D. J. Abadi, "H-store: a high-performance, distributed main-memory transaction processing system," Proc. VLDB Endow., vol. 1, iss. 2, pp. 1496-1499, 2008

[SH08] Stavros Harizopoulos, Daniel Abadi, Samuel Madden, and Michael Stonebraker, "OLTP Through the Looking Glass, and What We Found There" ACM SIGMOD 2008. [DKC85] D K Chandy, Leslie Lamport, "Distributed snapshots: determining global states of distributed systems", ACM Trans. Comput. Syst., Vol. 3, No. 1. (February 1985), pp. 63-75.