Lessons Learned in Building Real-time Big Data Systems

(Keynote)
Srini V. Srinivasan
Founder and VP Engineering & Operations
Aerospike

srini@aerospike.com

In the Age of the Customer, enterprises must modernize their application infrastructure to use real-time big data to attract, engage and retain consumers across devices, media and channels. Processing massive amounts of data in real-time creates a competitive advantage that has an enormously positive impact on business.

It has been clear now for a long time that lower latency means higher sales for Internet enterprises. In fact, Internet sites routinely lose users to other sites that support lower latency. E.g., Amazon found every 100ms of latency cost them 1% in sales. Google found an extra .5 seconds in search page generation time dropped traffic by 20%[1].

Therefore, predictable low latency is a sure fire way to win in the marketplace. Nowhere has this been more apparent than in the growth of Real-Time Bidding (RTB) systems for delivering digital advertising.

RTB has been effectively used to monetize "long tail" (remnant) inventory and target users across websites and mobile apps, anywhere they might be on the Internet. In fact, RTB has been the key factor driving the enormous growth in digital advertising worldwide. Low latency is a lynchpin of the RTB system, where the entire process from click to view must complete in under 150 milliseconds.

Platform companies realized the critical nature of keeping this contract[2]. At the center of such a business is fast access to data. Note that the user data in an RTB system is changing constantly since the choice of actions at every user visit needs to take into account past behavior of that user. So, such RTB applications need databases that provide predictable sub-millisecond latency for reads in the presence of heavy write load.

Clearly traditional systems are not sufficient for this. It has been known for a while that Database Systems need a complete rewrite[3]. Even most of the first generation NoSQL systems are inadequate. Some of the RTB majors have used custom systems they developed on their own on top of other inadequate systems. In fact, building a fast in-memory system on top of a slow Database could be a "fate worse than death"[4]. The most successful companies use ultra-fast clustered systems[5] or single node systems[6]. These systems work quite well on bare metal[7] or in the cloud[8].

The 20th International Conference on Management of Data (COMAD), 17th-19th Dec, 2014 at Hyderabad, India.

System developers and operators face several issues while deciding to use such a new Database system for their applications:

- From the application point of view, the system needs to be able to deliver extremely low latency for reads in the presence of heavy write load. This is an especially hard problem to solve for traditional databases. In addition, the system must provide support for queries in addition to simple (and fast) key value access.
- It is important that applications work in both cloud based virtual deployments as well as on bare metal data center deployments. Specifically, it is critical that applications work on commodity hardware with no special purpose setup needed for launch.
- As more and more mainstream enterprises move to low latency applications, it is important to avoid sacrificing consistency at the altar of availability[9]. The best systems are those that make judicious choices and provide availability and consistency with high performance in a wide variety of useful scenarios[10]. For example, minimizing network partitions considerably reduces the negative effects of the CAP theorem and it is hard but not impossible to provide ACID support.
- Parallelism is quite powerful both within a node as well as across nodes. Harnessing the best performance and scaling up on one node and scaling out are both important. For example, using a hybrid in-memory system using both DRAM and SSD (Flash), one can run a 14-node cluster using a DRAM/SSD configuration instead of a 186-node cluster using a pure DRAM system. Such a cluster will still provide sub-millisecond latency, but do so at a ten times lower cost than pure DRAM systems.
- Operational excellence is necessary to ensure that a service runs 24X7. All code should be written so that it can run as a service. Extremely high performance (e.g., 1 million TPS per node) provides sufficient headroom for making sure that failures can be handled seamlessly. Additional capacity can also be used to provide better consistency in the presence of failures.
- High performance in a system can be achieved by ensuring that software takes maximum advantage of the performance of hardware. Techniques that are useful: using multiple threads, reference counts to avoid data copies, efficient memory usage (e.g., restricting the index entries to 64 bytes, the same as a cache line), real-time prioritization algorithms to keep the system running smoothly, etc.

To conclude, by making appropriate choices, predictable low latency can co-exist with enough consistency in the vast majority of big data systems. This will enable enterprises to build real-time applications that add to the top line of every Internet enterprise.

Author:

Srini V. Srinivasan, founder and VP Engineering & Operations

Srini brings 20-plus years of experience in designing, developing and operating Web-scale infrastructures, and he holds over a dozen patents in database, Internet, mobile, and distributed system technologies. Srini co-founded Aerospike to solve the scaling problems he experienced with relational databases at Yahoo! where, as senior director of engineering, he had global responsibility for the development, deployment and 24×7 operations of Yahoo!'s mobile products, in use by tens of millions of users. Srini also was chief architect of IBM's DB2 Internet products, and he served as senior architect of digital TV products at Liberate Technologies. Srini has a B.Tech in Computer Science from IIT Madras and a M.S. and PhD in Databases from University of Wisconsin-Madison.

1. REFERENCES

[1] http://glinden.blogspot.com/2006/11/marissa-mayer-at-web-20.html

[2] http://www.adexchanger.com/online-advertising/equinix-seeks-to-speed-rtb-bidding/

[3] Michael Stonebraker, Samuel Madden, Daniel J. Abadi, Stavros Harizopoulos, Nabil Hachem, and Pat Helland. The end of an architectural era: (it's time for a complete rewrite). In *Proceedings of the 33rd International Conference on Very Large DataBases* (VLDB). 2007.

[4] https://gigaom.com/2011/07/07/facebook-trapped-in-mysql-fate-worse-than-death/

[5] http://highscalability.com/blog/2014/5/6/the-quest-for-database-scale-the-1-m-tps-challenge-three-des.html

[6] http://highscalability.com/blog/2014/8/27/the-12m-opssec-redis-cloud-cluster-single-server-unbenchmark.html

[7] http://www.aerospike.com/wp-content/uploads/2013/01/Ultra-High-Performance-NoSQL-Benchmarking.pdf

[8] http://highscalability.com/blog/2014/8/18/1-aerospike-server-x-1-amazon-ec2-instance-1-million-tps-for.html

[9] Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. Dynamo: amazon's highly available key-value store. In *Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles* (SOSP). 2007.

[10] V. Srinivasan and Brian Bulkowski. Citrusleaf: A Real-Time NoSQL DB which Preserves ACID. *Proc. VLDB Endow. (PVLDB) Vol.4(12), 1340-1350.* 2011.