Review: Replication

Louis Rabiet

December 4, 2013

What problem did the papers address? Who is the intended audience?

The common subject for the three papers is replication. The idea is either to be efficient or to be able to resist some failures by duplicating some data in a system where several nodes are used. The papers [1] provides an in-depth analysis for two common replication schema (eager and lazy).

[2] try to describe backup solutions for streaming applications. The problem is very specific to streaming processing since the categorization of the recovery process is only relevant when you are in the context of streaming.

[4] is simply doing a review of the replication schema in the distributed system and databases.

Is it important/interesting? What was the context for the papers?

[1] is not using portioning which the current solution used by most of the databases to scale-up (fairly new databases system will try to distribute uniformly the workload to different machines and will use replication primarily for failure resilience and not that much for performance improvement). With the NoSQL, most of the paper is not relevant since a deadlock is not acceptable while synchronization between several nodes is not primordial (portioning help for that).

Streaming processing (in particular) need to be failure resilient since we will probably have more and more specialized processors available on the market. The precise analysis and description of the differences between the type of recovery is an pointer for programmers that want to develop applications on this kind of architecture.

[4] tries to make a link between two communities that works on similar problems (lot of nodes, lot of failures, performance concerns).

What is the approach used to solve the problems?

Authors of [1] start from the lazy master replication and will improve the schema to allow disconnection of nodes.

Again in [2], an analytic study of the recovery cost is done. They use a simulator to get some results on the recovery time.

What are the possible inefficiencies in this approaches?

- In [1]: any substantial database cannot be replicated in full. The number of objects cannot be fixed and the number of nodes should be parameters (it seems to be doable easily for both).

- In [2]: We can look at the stream on some segment and not just at every output. For example if you can tolerate some error in the stream (t1 t2' t3 is enough to reconstruct t1 t2 t3).
- In [4]: they are not trying to combine both type of replication.

How does the papers support or otherwise justify the conclusions it reaches?

[1] has a very nice way of presenting the analysis of eager and lazy replication schema. The authors shows the scalibility problem of such systems (under certain conditions). The proof is made step-by-step and is very precise. This paper is lacking on the evaluation part of their new replication schema. While it is clear that the system is improved compare to the old schema, an analysis would have been preferred.

[2] is mostly based on analytic study but still employ some simulation to avoid a difficult queue system analysis.

What problems are explicitly or implicitly left as future research questions?

- Is the replication alone allows stability? While it is used in real systems to allow load balancing (between machines that owns the same data), most of the performance is coming from the partitioning part (if you database is big enough, it will not be possible to synchronize everything).
- There is no variation in the number of backups (only one) node. What about optimization on throughput (using inactive secondary nodes if needed) and reducing recovery possibilities?

References

[1] Jim Gray and Pat Helland. The dangers of replication and a solution. In *In Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 173–182, 1996.

[2] Jeong hyon Hwang, Magdalena Balazinska, Er Rasin, Michael Stonebraker, and Stan Zdonik. High-availability algorithms for distributed stream processing. Technical report, 2004.

[3] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. Conflict-free replicated data types. In *Proceedings of the 13th International Conference on Stabilization, Safety, and Security of Distributed Systems*, SSS'11, pages 386–400, Berlin, Heidelberg, 2011. Springer-Verlag.

[4] Matthias Wiesmann, Fernando Pedone, André Schiper, Betina Kemme, and Gustavo Alonso. Understanding replication in databases and distributed systems. pages 264–274, 2000.