Elastic, scalable and self-tuning data replication in the Cloud-TM platform

Paolo Romano
INESC-ID/Instituto Superior Técnico, Lisbon, Portugal

1. INTRODUCTION

Over the last years Cloud Computing has emerged as a disruptive paradigm for the future generation of IT services. Just as the electric grid revolutionized access to electricity one hundred years ago, freeing corporations from having to generate their own power and enabling them to concentrate on their business differentiators, cloud computing is hailed as revolutionizing IT, freeing corporations from large IT capital investments and enabling them to plug into extremely powerful computing resources over the network.

But the promise of elastic computing and infinite scalability, which catalysed much of the recent interest on Cloud Computing, raises also a number of complex issues that challenge the state of the art methodologies and practices in the area of distributed data management.

For several decades, relational databases have represented the indisputable reference solution for transactional data management. Unfortunately, relational databases are known not to be easy to scale out on shared nothing infrastructures [11, 3]. The recent proliferation of a new generation of in-memory, transactional data platforms, often referred to as NoSQL data grids [10, 4], is motivated precisely by the urge for overcoming the scalability and elasticity shortcomings of relational databases.

Clearly, in these in-memory transactional platforms, data replication plays a fundamental role both for performance and fault-tolerance purposes. Replication is a well-known technique and a wide body literature has been developed in this area over the last decades. On the other hand, the cloud's requirements for elasticity and high scalability pose several new challenges, briefly summarized in the following:

Scalability vs consistency: A common trait characterizing the new generation of cloud data platforms is the adoption of a range of weak consistency models, such as eventual consistency [6], restricted transactional semantics (e.g. single object transactions [8], or static transactions [1]), and non-serializable isolation levels [2].

Embracing weak consistency, these platforms have been shown to achieve unprecedented scalability levels [4]. On the down side of the coin, weak consistency models expose additional complexity to application developers, who are required to deal with the idiosyncrasies due to concurrency, network partitioning and/or failures.

Therefore, a crucial research question is whether consistency and scalability are actually two mutual exclusive qualities, or whether, there exist any sweet spot in the trade-off between scalability and consistency that allow to design highly scalable data replication protocols while exposing simple and intuitive consistency semantics.

To scale or not scale, that is the question. Most of existing IaaS and PaaS platforms already allow non-expert users to provision a cluster of any size on the cloud within minutes. This feature gives tremendous power to the average user, while placing a major burden on her shoulders. Previously, the same user would have had to work with system administrators and management personnel to get a cluster provisioned for her needs. However, removing the system administrator and the traditional capacity-planning process from the loop shifts the non-trivial responsibility of determining a good cluster configuration to the non-expert user.

Unfortunately, forecasting the performance of data centric applications while varying the scale of the underlying platform is extremely challenging. In fact, the performance of distributed data management platform, such as the aforementioned NoSQL data grids, tend to exhibit strong non-linear behaviors as the number of nodes in the system grows, as a consequence of the simultaneous, and often inter-dependent, effects of contention affecting both physical (computational, memory, network) and logical (conflicting data accesses by concurrent transactions) resources [7].

Due to these complexities, auto-scaling mechanisms currently offered by commercial cloud support only simple reactive provisioning policies based on user defined thresholds on resource (e.g., CPU or memory) utilization. However, no guarantee is provided on the impact of the auto-scaling policies on key application level performance indicators, such as throughput or response time, which are essential for the definition of any Service Level Agreement.

No-one-size-fits-all solutions: Decades of literature and field experience in the area of data replication have brought to the development of a plethora of approaches for state consistency in distributed platforms, and taught a fundamental, general lesson: no universal, one-size-fits-all solution exists that can achieve optimal efficiency across all possible kinds

(∗) This work has been partially supported by the EU project Cloud-TM, and by FCT (INESC-ID multiannual funding).

EWDCC '12 May 8, Sibiu, Romania

of workloads and for any level of scale of the system.

This issue is is hence particularly exacerbated in Cloud Computing platforms due to the feature that is regarded as one of the key advantages of the cloud: its ability to elastically acquire or release resources, dynamically varying the scale of the platform in real-time to meet the demands of varying workloads. This means that in order to maximize efficiency (i.e. minimizing operational costs, in the pay-for-what-you-use pricing model) data management middleware should be able to adapt their consistency mechanisms in order to ensure optimal performance for every workload and at any scale.

2. THE CLOUD-TM PROJECT

Cloud-TM[1] is an EU project focused on the development of an innovative data-centric platform aimed to facilitate the development and administration of cloud applications.

In this talk I will overview three recent results of the Cloud-TM project that address the above mentioned issues in the area of data replication:

GMU. The first presented result is GMU [9], a genuine partial replication protocol for transactional systems, which exploits an innovative, highly scalable, distributed multi-versioning scheme. GMU never blocks or aborts read-only transactions and spares them from distributed validation schemes, ensuring high performance in presence of read-intensive workloads, as typical of a wide range of real-world applications. Unlike existing multiversion-based solutions, GMU does not rely on a global logical clock, hence avoiding global contention points that would limit system scalability. GMU guarantees the Extended Update Serializability (EUS) isolation level. This consistency criterion is particularly attractive as it is sufficiently strong to ensure correctness even for very demanding applications (such as TPC-C), but is also weak enough to allow efficient and scalable implementations, such as GMU. Further, unlike several relaxed consistency models proposed in literature, EUS has simple and intuitive semantics, thus being an attractive, scalable consistency model for ordinary programmers.

TAS. TAS (Transactional Auto Scaler) [7] is an elastic-scaling system that relies on a novel hybrid analytical/machine-learning-based forecasting methodology in order to predict the performance achievable by transactional applications executing on top of transactional in-memory data stores, in face of changes of the scale of the system.

Applications of TAS range from on-line self-optimization of in-production applications, to the automatic generation of QoS/cost driven elastic scaling policies, and what-if analysis on the scalability of transactional applications.

Polycert. Atomic Broadcast (AB) based certification replication schemes have emerged as a more scalable alternative to classical replication protocols based on active replication or atomic commit protocols. However, as I will show, among the existing AB-based certification protocols, no"one-fits-all" solution exists that achieves optimal performance in presence of heterogeneous workloads. Next, I will present PolyCert [5], a polymorphic data replication protocol that allows for the concurrent co-existence of different AB-based certification protocols, relying on machine-learning techniques to determine the optimal certification scheme on a per transaction basis.

By self-tuning the replication strategy on the basis of current workload, PolyCert can achieve a performance extremely close to that of an optimal non-adaptive protocol in presence of non heterogeneous workloads, and significantly outperform any non-adaptive protocol when used with complex applications generating heterogeneous workloads.

[1] `http://www.cloudtm.eu`

3. REFERENCES

[1] M. K. Aguilera, A. Merchant, M. Shah, A. Veitch, and C. Karamanolis. Sinfonia: a new paradigm for building scalable distributed systems. *SIGOPS Operating Systems Review*, 41:159–174, 2007.

[2] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil. A critique of ANSI SQL isolation levels. In *Proc. of International Conference on Management of Data*, pages 1–10. ACM, 1995.

[3] E. A. Brewer. Towards robust distributed systems (abstract). In *Proc. of the 19th Symposium on Principles of Distributed Computing*. ACM, 2000.

[4] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: a distributed storage system for structured data. In *Proc. of the 7th Symposium on Operating Systems Design and Implementation*, pages 15–15. USENIX Association, 2006.

[5] M. Couceiro, P. Romano, and L. Rodrigues. Polycert: Polymorphic self-optimizing replication for in-memory transactional grids. In *Proc. of 12th Conference on Middleware*, Middleware '11, pages 309–328. Springer, 2011.

[6] 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 *Proc. of Symposium on Operating Systems Principles*, pages 205–220. ACM, 2007.

[7] D. Didona, P. Romano, S. Peluso, and F. Quaglia. Transactional auto scaler: Elastic scaling of nosql transactional data grids. Technical Report 50, INESC-ID, December 2011.

[8] A. Lakshman and P. Malik. Cassandra: a decentralized structured storage system. *SIGOPS Operating Systems Review*, 44:35–40, 2010.

[9] S. Peluso, P. Ruivo, F. Quaglia, and L. Rodrigues. When scalability meets consistency: Genuine multiversion update-serializable partial data replication. In *Proc. of the 32nd International Conference on Distributed Systems*. IEEE Computer Society, 2012.

[10] Red Hat. Infinispan , *http://www.jboss.org/infinispan*.

[11] 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 *Proc. of the 33rd International Conference on Very Large Data Bases*, pages 1150–1160. VLDB Endowment, 2007.