Big Data and Cloud Computing: New Wine or just New Bottles?*

Divyakant Agrawal Sudipto Das Amr El Abbadi
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA 93106-5110, USA
{agrawal, sudipto, amr}@cs.ucsb.edu

1. INTRODUCTION

Cloud computing is an extremely successful paradigm of service oriented computing and has revolutionized the way computing infrastructure is abstracted and used. Three most popular cloud paradigms include: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The concept however can also be extended to Database as a Service and many more. *Elasticity*, *pay-per-use*, *low upfront investment*, *low time to market*, and *transfer of risks* are some of the major enabling features that make cloud computing a ubiquitous paradigm for deploying novel applications which were not economically feasible in a traditional enterprise infrastructure settings. This has seen a proliferation in the number of applications which leverage various cloud platforms, resulting in a tremendous increase in the scale of the data generated as well as consumed by such applications. Scalable database management systems (DBMS) – both for update intensive application workloads, as well as decision support systems for descriptive and deep analytics– are thus a critical part of cloud infrastructures.

The quest for conquering the challenges posed by the management of big data has led to a plethora of systems. This tutorial aims to clarify some of the critical concepts in the design space of big data and cloud computing such as: the appropriate systems for a specific set of application requirements, the research challenges in data management for the cloud, and what is novel in the cloud for database researchers? We also aim to address one basic question: Whether cloud computing poses new challenges in scalable data management or it is just a reincarnation of old problems? We provide a comprehensive background study of state-of-the-art systems for scalable data management and analysis. We also identify the critical aspects in the design of different systems and the applicability and scope of these systems. We further focus on a set of systems which are designed to handle update heavy workloads for supporting internet facing applications. We identify some of the design challenges which application and system designers face in developing and deploying new applications and systems, and expand on some of the major challenges that need to be addressed to ensure the smooth transition of applications from traditional enterprise infrastructures to the next generation of cloud infrastructures. A thorough understanding of current solutions and a precise characterization of the design space are essential for clearing the "cloudy skies of data management" and ensuring the success of DBMSs in the cloud, thus emulating the success enjoyed by relational databases in traditional enterprise settings.

2. TUTORIAL OUTLINE

2.1 Background: Scalable Data Management

In the first part of the tutorial, we outline the features of the cloud that make it attractive for deploying new applications and provide the necessary background by analyzing different scalable data management solutions. Our background study encompasses both classes of systems: (i) for supporting update heavy applications, and (ii) for ad-hoc analytics and decision support. The goal is to provide an elaborate summary of the various approaches for scaling to manage big data, and for supporting both classes of applications. For both class of systems, we survey the state of the art and the recent trends and developments, and outline the interesting design choices as well as the scope of these different systems. For the rest of the tutorial, we focus our attention on the class of systems that are designed to support update heavy web-applications deployed in the cloud. We sub-divide this class into two sub-classes: one where the goal of the system is to support a single large application with large amounts of data (scalable single tenant DBMS); and another where the goal of the system is to support a large number of applications each with a small data footprint (large multitenant DBMS).

2.2 Systems for Update heavy applications

2.2.1 Data Management for Large Applications

In this part of the tutorial, we focus on the design issues in building a DBMS for dealing with applications with single large databases. We refer to this as a *large single tenant system*. Many applications often start small, but with growing popularity, their data footprint continues to grow. As the size of data grows beyond the limits of what can be reasonably served using a single node, system architects have to either rely on expensive commercial solutions or move beyond relation database technology. This has been observed specifically in the modern era of innovative application ideas whose deployment has been made feasible by the cloud economics, and whose popularity often has wide fluctuations. The application servers can easily scale out, but the data management infrastructure often becomes a bottleneck. The lack of cloud features in open source relational DBMSs (RDBMSs) and the high cost as-

*This work is partially funded by NSF grant IIS-0847925.

Proceedings of the VLDB Endowment, Vol. 3, No. 2
Copyright 2010 VLDB Endowment 2150-8097/10/09... $ 10.00.

sociated with enterprise solutions make RDBMSs less attractive for the deployment of large scale applications in the cloud. This has resulted in the popularity of *Key-Value* stores: examples include Bigtable [3], PNUTS [4], Dynamo [8] and their open-source counterparts. These systems have been extensively deployed in various private as well as public and commercial cloud infrastructures. We provide a survey of the popular *Key-Value* stores and crystallize the features and design choices made by these system that have allowed them to scale out to petabytes of data and thousands of concurrent requests – a scale which distributed databases have failed to achieve. We also survey how these principles can be extended beyond *Key-Value* stores for designing scalable systems with a richer set of functionality compared to these *Key-Value* stores [7]. We end this discussion with a survey of some of the current research projects which aim to infuse the cloud features in relational databases to allow the effective deployment of applications which are better expressed and designed using the features supported by RDBMSs. These systems include ElasTraS [5, 6], DBonS3 [2, 9], and Project Deuteronomy [10, 11] to name a few.

2.2.2 *Large Multitenant Databases*

Another important domain for data management in the cloud is the need to support a large number of applications, each of which has a small data footprint [13]. This is referred to as a *large multitenant system*. Database multitenancy is traditionally considered only in the case of SaaS with Salesforce.com being a canonical example [12] where different tenants share the same database tables. But different models of multitenancy are relevant in the context of different cloud paradigms. For instance, a PaaS provider, dealing with a large number of applications with very different schemas, might require a different form of sharing in contrast to the shared table approach used in traditional designs [1, 12]. The goal of this section of the tutorial is to survey the different approaches to multitenancy in a database, and garner the understanding of the requirements and applicability of the different multitenancy models for infusing cloud features into such a system.

2.3 Future of Cloud Data Management

In the concluding part of the tutorial, we identify some of the major open problems that must be addressed to ensure the success of data management systems in the cloud. In summary, a single perfect data management solution for the cloud is yet to be designed. Different systems target different aspects in the design space, and multiple open problems still remain. These challenges include: characterizing the consistency semantics that can be provided at different scales, effective techniques for dealing with the elasticity of cloud infrastructures, designing scalable, elastic, and autonomic multitenant database systems, and last but not the least, ensuring the security and privacy of the data outsourced to the cloud.

3. GOALS OF THE TUTORIAL

3.1 Learning Outcomes

- State-of-the-art scalable data management systems for traditional and cloud computing infrastructures for both update heavy as well as analytical workloads.
- Design choices that have led to the success of some systems, and that limited the success of some other systems.
- Design principles that should be carried over in designing the next generation of data management systems for the cloud.
- Understanding the design space for DBMSs targeted to supporting update intensive workloads for supporting large single tenant systems and large multitenant systems.
- Understanding the different multitenancy models in the database layer and trade-offs associated.
- A list of open research challenges in data management for the cloud that must be addressed to ensure the continued success of DBMSs.

3.2 Intended Audience

This tutorial is intended to benefit researchers and system designers in the broad area of scalable data management for traditional as well as cloud data platforms. A survey of the current systems and an in-depth understanding will is essential for choosing the appropriate system as well as designing an effective system. This tutorial does not require any prior knowledge in scalable data management systems or multitenant systems.

4. BIOGRAPHICAL SKETCHES

Divyakant Agrawal is a Professor of the Department of Computer Science at the University of California, Santa Barbara. Prof. Agrawal's research expertise is in the areas of database systems, distributed computing, data warehousing, and large-scale information systems.
Sudipto Das is a PhD Candidate at the Department of Computer Science, University of California Santa Barbara. His research interests lie in the area of scalable data management in cloud computing infrastructures.
Amr El Abbadi is a Professor and Chair of the Department of Computer Science at the University of California, Santa Barbara. His research interests lie in the broad area of scalable database and distributed systems.

5. REFERENCES

[1] S. Aulbach, D. Jacobs, A. Kemper, and M. Seibold. A comparison of flexible schemas for software as a service. In *SIGMOD*, pages 881–888, 2009.

[2] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska. Building a database on S3. In *SIGMOD*, pages 251–264, 2008.

[3] 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 *OSDI*, pages 205–218, 2006.

[4] B. F. Cooper, R. Ramakrishnan, U. Srivastava, A. Silberstein, P. Bohannon, H.-A. Jacobsen, N. Puz, D. Weaver, and R. Yerneni. PNUTS: Yahoo!'s hosted data serving platform. *Proc. VLDB Endow.*, 1(2):1277–1288, 2008.

[5] S. Das, S. Agarwal, D. Agrawal, and A. El Abbadi. ElasTraS: An Elastic, Scalable, and Self Managing Transactional Database for the Cloud. Technical Report 2010-04, CS, UCSB, 2010.

[6] S. Das, D. Agrawal, and A. El Abbadi. ElasTraS: An Elastic Transactional Data Store in the Cloud. In *USENIX HotCloud*, 2009.

[7] S. Das, D. Agrawal, and A. El Abbadi. G-Store: A Scalable Data Store for Transactional Multi key Access in the Cloud. In *ACM SOCC*, 2010.

[8] 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 *SOSP*, pages 205–220, 2007.

[9] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann. Consistency Rationing in the Cloud: Pay only when it matters. *PVLDB*, 2(1):253–264, 2009.

[10] D. B. Lomet, A. Fekete, G. Weikum, and M. J. Zwilling. Unbundling transaction services in the cloud. In *CIDR Perspectives*, 2009.

[11] D. B. Lomet and M. F. Mokbel. Locking Key Ranges with Unbundled Transaction Services. *PVLDB*, 2(1):265–276, 2009.

[12] C. D. Weissman and S. Bobrowski. The design of the force.com multitenant internet application development platform. In *SIGMOD*, pages 889–896, 2009.

[13] F. Yang, J. Shanmugasundaram, and R. Yerneni. A scalable data platform for a large number of small applications. In *CIDR*, 2009.