The Curriculum Forecast for Portland: Cloudy with a Chance of Data

Michael Grossniklaus
Computer Science Department
Portland State University
Portland, OR 97201
grossniklaus@cs.pdx.edu

David Maier
Computer Science Department
Portland State University
Portland, OR 97201
maier@cs.pdx.edu

ABSTRACT

With the advent of cloud computing, new data management technologies and systems have emerged that differ from existing databases in important ways. As a consequence, universities are currently facing the challenge of integrating these topics into their curriculum in order to prepare students for the changed IT landscape. In this report, we describe the approach we have taken at Portland State University to teach data management in the cloud. We also present our experiences with this effort and give an outlook on how it could be adapted to suit the requirements of other universities.

1. MOTIVATION

Novel cloud data-management systems are different from traditional databases (and each other) in models, languages, consistency guarantees, scalability, and performance characteristics. Therefore, cloud data management has spawned activities in both the database research community and industry. In research, it has lead to renewed interest in shared-nothing architectures and alternative data processing paradigms. In industry, new and existing companies target cloud-based services and infrastructures, both as providers and users. In the Pacific Northwest and Northern California, well-known companies such as Amazon, Google, and Microsoft as well as countless startups are seeking professionals with expertise in cloud computing and cloud data management. As a consequence, we at Portland State University determined that curriculum in this area would be valuable to students entering the job market in the wider Portland area.

The first decision we had to make was whether to integrate these topics into existing undergrad and grad courses, or offer a stand-alone course on cloud data management. When redesigning their curriculum, the University of Washington, for example, opted to teach the MapReduce [4] data processing paradigm in their introductory database course at the undergrad level. In contrast to their approach, we created a dedicated 10-week course for both grad students and advanced undergrads, motivated by at least two reasons. First, due to the timely nature of this topic, we wanted to ensure that all students and, in particular, students close to their graduation get a chance to learn about cloud data management. Extending existing basic courses is therefore not an option as our target group of students will, in all likelihood, have already attended them. Second, the emerging nature of this topic does, in our opinion, not yet justify modifying the existing curriculum, which has been designed to teach established basic knowledge. We believe that a stand-alone course provides a better framework to experiment with the teaching of novel topics and that, once they have matured, blocks from such courses can be integrated into mainstream courses.

The second decision concerned the actual contents of the course. Our main goal was to make the course as self-contained and complete as possible. In terms of self-containedness, we assumed previous knowledge about the design and implementation of databases as well as programming skills, but chose to include an introductory primer on the fundamentals of cloud computing. In terms of completeness, we aimed for a good balance between general cloud data-management principles and actual cloud data-management systems that are currently in use or being developed. Given the diverse and heterogeneous world of cloud data-management systems, where new approaches emerge on a regular basis, finding this balance was challenging. To address this problem, we focused on a few representative systems in the lectures and introduced other related approaches using assignments such as readings, projects, and discussions. Additionally, we emphasized efforts to classify, compare and benchmark the various approaches and systems.

The remainder of this report is organized as follows. Section 2 introduces the structure and content of the curriculum, while Section 3 describes assignments we created to complement the lectures. In Section 4 we report on experiences, then give an outlook on possible adaptations and future editions of the course in Section 5. Finally, resources available to fellow educators are listed in Section 6.

2. CURRICULUM

Apart from an initial cloud-computing primer and a closing look at the user's perspective, we structured the curriculum into two main parts, which respectively discussed novel NoSQL data management systems and efforts to scale traditional SQL databases.

2.1 The Basics

As mentioned, we began our course with a primer that introduced students to cloud-computing principles. In particular, we focused on utility computing in terms of pay-as-you-go models and elastic scalability as major factors that distinguish cloud computing from previous parallel or cluster-based computing paradigms. We also discussed enabling technologies such as virtualization and service-oriented architectures to provide processing power, storage, and software as commodities. This primer was in part based on UC Berkeley's view of cloud computing [2]. We concluded this introductory block by looking at the implications of cloud computing on data management in terms of providing cloud data services. The presentation of the corresponding challenges was based on the 2008 Claremont report on database research [1]. Finally, we introduced some ideas that have become household terms in cloud data management such as the CAP "theorem", eventual consistency, and BASE.

2.2 NoSQL Data Management

We structured the block on NoSQL data management into two parts, following David DeWitt's classification of the area into NoSQL OLTP and NoSQL Data Warehousing [5]. Our presentation of NoSQL OLTP data stores was based on Rick Cattell's survey [3] that distinguishes the key-value, document, and column-family data models. For our course, we also included object, graph, and array data models. For each data model, we selected a representative data store that was presented in the lecture. The chosen systems were Amazon Dynamo (key-value), MongoDB (document), BigTable (column family), Neo4j (graph), SciDB (array). To represent object databases, we invited Leon Guzenda, Chief Technology Officer at Objectivity, to talk about Objectivity/DB and InifiniteGraph.

In the context of NoSQL data warehousing, we introduced the Google File System (GFS) and MapReduce, and related them to their open-source counterparts, i.e., the Hadoop File System (HDFS) and Hadoop. We presented Pig/Pig Latin and Hive as declarative ways of specifying data processing tasks that build on Hadoop. For each of these novel data processing paradigms, we compared how it relates to traditional query processing. In particular, we talked about the challenges of executing iterative tasks or relational queries with joins. While we believe it important that students understand the motivation and advantages of new technologies, we are also convinced that they need to know about limitations in order to make informed and realistic decisions about their use in their professional life.

2.3 Scalable SQL Databases

To understand the trade-offs and techniques to horizontally scale traditional relational databases, we looked at two concrete systems. The first system introduced was VoltDB, which we used to emphasize design decisions that set it apart from conventional RDBMS servers. For example, VoltDB manages all data in main memory, maintains replicas for fault tolerance, and avoids user interactions in transactions, i.e., transactions must be registered in advance and can therefore be optimized off-line and scheduled serially.

For the second platform, we invited Michael Rys, Principal Lead Program Manager at Microsoft SQL Server, to present Microsoft SQL Azure, which supports horizontal scalability by sharding data over databases in a SQL Azure Federation, i.e., a cluster of SQL Server instances. Using this infrastructure, SQL-based MapReduce tasks can be defined and executed. Michael's talk concluded with a roadmap of how Microsoft plans to add further support for NoSQL paradigms to their data platform.

2.4 User Perspective

Towards the end of the course, we scheduled a third guest lecture given by Adam Lowry, a Portland State University graduate and Co-Founder of Urban Airship, a local startup that provides a content-based messaging platform for mobile applications. Adam's talk introduced students to the perspective of users of cloud data-management systems. In his presentation, Adam retraced the trials and tribulations of his company that lead them from problems with MongoDB to trouble with Cassandra, and experimentation with HBase. Ultimately, they moved off these platforms completely and their current infrastructure is based on PostgreSQL.

3. ASSIGNMENTS

Our course was accompanied with a series of assignments that students carried out in class or at home. The goal of these assignments was threefold. First, in-class discussion assignments gave us an opportunity to react to developments in cloud data management as they were happening, such as the release of Google Cloud SQL or VMware vFabric SQLFire. Second, homework reading assignments provided students with more details or alternative approaches. Finally, a course project exposed groups of students to practical experience with a specific cloud data-management system.

3.1 Readings

We decided to "front-load" our course in terms of reading assignments clustered primarily in the first five weeks. The first three assigned papers gave more background on topics discussed during the lectures of the corresponding week. We used this approach for the introduction of NoSQL data management by asking students to read Rick Cattell's survey [3] as well as to underpin our presentation of Amazon's Dynamo and Google's BigTable.

For each assignment, students were given a set of tasks which varied for undergrad and grad students to cater for different course requirements. A typical task would probe a student's understanding of the paper by asking him or her to contrast and compare the presented solutions with other approaches. Additionally, grad students were given more open tasks that challenged them to be creative and visionary.

3.2 Project

The setting for the course project was a system that manages and processes social network data. We selected this application scenario based on its appeal to the students, its relevance due to companies such as Facebook that are innovators in the domain of cloud data management, and its versatility, which enables a variety of use cases that range from operating a social-networking site (OLTP) to social-network analysis (OLAP). This assignment was challenging, but again we believe it is important to use a scenario that can also demonstrate some of the limitations of these new technologies.

3.2.1 Modeling

Students formed five teams with five to six students each. Each group worked with a different cloud data-management system. To span a range of models, we selected Voldemort, CouchDB, SimpleDB, Cassandra, and OrientDB. Students first familiarized themselves with their system and compiled a detailed system profile to characterize and compare it with other systems. In order to help them cover the same points, we provided a template for this task. Each group presented the resulting profile in class in order to give all students a chance to learn about the particulars of each system.

After this "warm-up" task, we specified a conceptual graph data model that teams had to express in the logical data models of their systems. We also provided a set of three example queries that the students' designs needed to support. The three queries—friends of a friend, identifying bridges, and transitive closure—represent a variety of use cases with very different complexities. Since none of the systems chosen by the students supports declarative relational queries, any data model design is closely coupled to the queries that need to be supported. The goal of this modeling exercise was making students realize that a design that works well for one type of query might hamper support for other types of queries, or even render them impossible.

3.2.2 Implementation

Students were then asked to implement their design and optionally deploy the application in the cloud. Due to the wide range of systems used, providing support for a common deployment platform was impractical. Rather, we encouraged students to investigate deployment options for their specific system. As a result, some of them used a trial version of Amazon's EC2, while the CouchDB team deployed to the Iris Couch hosting service and the SimpleDB team worked with a trial version managed by Amazon. Once again, we asked students to present their design and implementation in class following an outline we provided.

3.2.3 Report and Essay

The final project task was to write a three-page report and essay based on an outline we distributed. In the report part, students were asked to summarize their roles in the project group as well as to discuss important decisions and choices made throughout the projects. The second part was an essay on cloud-scale data-management systems in today's world of computing. Students needed to demonstrate that they can position these systems within the IT landscape, and that they are aware of their advantages and disadvantages. Finally, they were asked to conclude with their own opinions of the topic. We decided to make this task an individual assignment to give students a chance to voice their personal views and to give us the possibility of evaluating the students individually.

4. EXPERIENCES

From an instructor's perspective, our experience with this course was quite positive. Students were highly motivated and quickly caught on to the issues we intended to convey, which they demonstrated in homework assignments and discussions in class. Designing the curriculum as well as creating the lectures and assignments from scratch was challenging, but we are satisfied that our efforts have paid off.

The course was also evaluated by the department as part of the routine course evaluation and generally received favorable to good reviews. Students were also encouraged to submit comments to help us understand better what the did or did not like. From these comments, we understand that the students appreciated the current and practical nature of the course. They also liked the many different ways of learning, i.e., readings, projects, class discussions, and guest lectures. Or, as one student put it: *"Please keep this course or something like it. It was great to learn about alternative data stores."* Some students criticized the course's workload as too heavy for a non-core course or felt that some details remained vague because the discussed technologies are still very new. In summary, the feedback was a mix of positive comments as well as valid and constructive criticism.

5. OUTLOOK

We conclude by considering how future editions of this course might be adapted and how it could be enriched for universities that have semesters instead of terms. We expect that some blocks, such as the cloud-computing primer, will eventually be embedded in prerequisite courses, making room for more topics related to data management.

There are several options in terms of additional content to compensate for this refactoring or to extend the term course to a semester course. First, lectures on the database support available in commercial cloud platforms such as Windows Azure, Google's AppEngine, or VMware vFabric could be added to the course. Second, the course could go into more details on NoSQL data warehousing by introducing additional data-processing approaches such as Google's Pregel paradigm. Ideally, there would also be an exercise in this area to give students practical experience with some of these tools.

There is a trade-off in terms of the systems introduced between breadth, i.e., range and variety of systems, and depth, i.e., details, documentation, and support provided. We prioritized breadth in an effort to expose students as much as possible to the real world, where there are many systems to chose from and there are not a lot of training materials. With an average age of 28.1 years, Portland State students are more mature and experienced than the average university student, as many of them have worked or currently work. It helped to have such people on each team, who were used to installing and configuring software as well as working with bleeding-edge tools.

6. RESOURCES

The course web site[1] contains resources such as the course schedule, a comprehensive reading list, and all assignments. Fellow instructors are free to reuse any of these resources, provided that the source is acknowledged. A PowerPoint deck with more than 300 slides is also available upon request.

7. ACKNOWLEDGMENTS

The course described in this report would not have been possible without the active and committed participation of the students as well as the help and support of the administrative staff at Portland State University. We would also like to thank Keke Chen at Wright State University for sharing the slides of his cloud computing course with us. Daniel Abadi's DBMS Musings Blog[2] has been a constant source of inspiration, interesting views, and timely commentary on current events. Michael Grossniklaus' work at Portland State University is funded by the Swiss National Science Foundation (SNSF) grant number PA00P2_131452.

8. REFERENCES

[1] R. Agrawal *et al.* The Claremont Report on Database Research. *Commun. ACM*, 52:56–65, 2009.

[2] M. Armbrust *et al.* Above the Clouds: A Berkeley View of Cloud Computing. Technical Report UCB/EECS-2009-28, University of California at Berkeley, February 2009.

[3] R. Cattell. Scalable SQL and NoSQL Data Stores. *SIGMOD Record*, 39(4):12–27, 2010.

[4] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *Proc. Symp. on Opearting Systems Design & Implementation (OSDI)*, pages 137–149, 2004.

[5] D. J. DeWitt and R. Nehme. Big Data – What's the Big Deal? `http://pages.cs.wisc.edu/~dewitt/includes/passtalks/passtalks.html`, 2011.

[1] `http://datalab.cs.pdx.edu/education/clouddbms/`
[2] `http://dbmsmusings.blogspot.com/`