Data Challenges at Yahoo!

Ricardo Baeza-Yates & Raghu Ramakrishnan
Yahoo! Research

ABSTRACT

In this short paper we describe the data that Yahoo! handles, the current trends in Web applications, and the many challenges that this poses for Yahoo! Research. These challenges have led to the development of new data systems and novel data mining techniques.

1. INTRODUCTION

Yahoo! is a universe of several dozens of distinct properties offering capabilities ranging from email, chat, online news, shopping, and classifieds; to Web search and social media sites, such as Flickr and Yahoo! Answers. These Web sites serve more than half a billion unique users per month and generate several billion dollars of revenue through advertising and other services. Due to this ecosystem, Yahoo! generates over a dozen terabytes per day of usage data of various forms. Just this type of data is the equivalent of the entire text content of the Library of Congress.

Storing and managing this ocean of data poses several important challenges. Many of these challenges deal with data management and stretch currently available technology, motivating the development of novel data systems. These new systems are designed in the context of four major interconnected trends that we believe will shape the future of online interactions [11]:

The emergence of structure. The highly heterogeneous data types we manage have distinct and interesting types of structure that can be useful for many applications. We believe that capturing and exploiting such structure is a key to next-generation Web applications, including search and advertising. While exploitation of structure is a natural topic of discussion, capturing structure is also nontrivial: in many cases structure is either user-provided (and sometimes designed to mislead) or automatically extracted (and potentially error-prone). This could also benefit from additional, integrated structure, inferred from the relations across datasets.

Design and dynamics of social systems. Online communities are a fundamental and increasingly important part of the web. A new science that brings to bear the mathematics of social networks, an economic theory of online interactions, and user experience design is required to further our understanding of how communities form and evolve, how we can facilitate their interactions, and how we can learn from shared community activities to enhance Web applications [12]. We view Yahoo! as a natural focal point for this evolution.

The Web as a delivery channel. The Web has become a powerful and ubiquitous means of delivering a range of end-user and collaborative applications to hundreds of millions of users. As online application development moves in the direction of "mash-ups" of online APIs (to a wide range of capabilities that may be combined into an application), the requirements on the backend change substantially. This has created a radically different approach to developing and distributing applications, disrupting the traditional software distribution model. In turn, it has challenged us to develop new types of service-oriented software platforms, new kinds of customizable application environments, and forced us to think about massively distributed systems with novel quality of service guarantees, fail-over mechanisms, and the ability to manage massive numbers of application instances.

The Web as wisdom. The Web has become the largest repository of data in the world, and hence of potential source of information and ultimately, knowledge. This data comes as content (text and multimedia in general), structure (links) and usage (navigation and query logs). Relating all this data allows to find the implicit wisdom or knowledge of the people [8] that create, synthesize and consume Web content. This process of relating data is called today Web mining. Extracting new information and new relations allows to improve the user experience, creating a virtuous circuit of continuous user-driven design. One main challenge is to understand and make sense of all this Web data as today we are just scratching its surface, due to the volume involved and the complexity of the problems addressed. In particular the correct use of semantic understanding is crucial to the next generation of Web search systems [3].

Next we describe the different data that Yahoo! manages, some examples of our data management systems, and the potential behind large-scale data mining.

2. DATA DIVERSITY

Yahoo! deals with enormous amounts of data, and the types of data we deal with include:

Static Text. As a web search engine, Yahoo! naturally deals with an enormous corpus of web pages, crawled and indexed regularly.

Dynamic Text. Yahoos social properties such as Answers, Flickr, del.icio.us, and Groups deal with large amounts of user-generated content; this is typically free-text, but also includes tags and ratings,

EDBT'08, March 25–30, 2008, Nantes, France.

and is organized with metadata such as the posters id, location, posting time, and context.

Structured Data. Many Yahoo! verticals (e.g., Autos, Local, Personals, Shopping) rely upon structured listings databases.

Streams. Many Yahoo! properties (e.g., Finance, News) rely upon content feeds, and of course, there are thousands of RSS feeds that are subscribed to by millions of users. Increasingly, social networking enhancements are leading to many new feeds that collate different kinds of user activities and events across the Yahoo! network. Finally, attentional metadata such as click-streams from a variety of user activities produce enormous data streams on a continuous basis.

Multimedia. Images and videos constitute an important and growing type of data. Videos, of course, represent a distinct kind of streaming data with its own set of characteristics and delivery requirements.

Mail. It is worth calling out the data involved in email, both because of its volumes (billions of messages per day), and the specialized nature of the data management systems that are used to handle the volumes involved in a leading web mail service such as Yahoo! Mail.

3. DATA MANAGEMENT

The diversity of the types of data we deal with is matched by the variety of data management tasks we face. While there are many data management tasks at Yahoo! that are adequately handled by commercial database management systems, two broad classes of data management scenarios that call for novel solutions are worth calling out:

Web application serving: Yahoos portal services include a wide range of web applications, e.g., Answers, del.icio.us, flickr, Groups, and portals, e.g., Yahoo! Finance, Local, and News. These applications typically deal with a large amount of data, including user-data, dynamic user-generated content, structured data, text data, and feeds. Storing and managing the data for a web application, providing a scalable and fault-tolerant data management service, is a recurring and complex challenge. Typical applications use a combination of data management solutions, ranging from light-weight object stores (e.g., the YDHT get-put interface for storing and fetching user-data such as authentication information and recent activity), to relational databases such as MySQL and Oracle, to a variety of file systems, including NetApp filers, to text indexing engines, e.g., Vespa.

Web-scale offline analysis: Analyzing the huge volumes of data that Yahoo! collects is a challenge faced by a number of groups within Yahoo! Some of the analyzes are readily cast as SQL queries; others involve more general data transformations and aggregations that are easier to express in a more programming language style idiom. Examples of the latter include complex statistical models to understand user preferences and content popularity by segment, and calculation of metrics characterizing the web graph (with pages as nodes and links as edges). The common denominator is that the scale of data pushes the boundaries of conventional database systems, and often makes them impractical or, at the least, too expensive.

In the next section, we provide an overview of some data platforms at Yahoo!, and discuss two active research projects that aim to develop scalable systems to address the two classes of scenarios outlined above, web serving and web-scale offline analysis.

3.1 Data Platforms

The brief overview in this section of data management platforms in use at Yahoo! is representative, rather than exhaustive, list.

To begin with, relational DBMSs, e.g., Oracle, are used quite widely, both for internal use in reporting and tracking applications, and as part of the data management infrastructure for externally facing web applications. Unfortunately, their limitations in the latter class of applications have necessitated the development of additional data management platforms.

An early example of such a platform is UDB, a large-scale replicated, distributed hash table service similar to Amazons Dynamo [6], although it is only available for the use of Yahoo! applications. It is essentially a simple get-put interface that virtualizes disk, and manages over fifteen replicas worldwide and handles peak traffic in excess of 350K requests/sec. A typical use is to look up user information when authenticating a user who logs into Yahoo!; this application, called UPES (User Profile and Events Store) implements additional functionality such as record expiration and access-control lists that are missing in UDB.

UDB is an example of a data serving platform. The Sherpa suite of data platforms, described later in this section, extends this class of systems within Yahoo!

Turning to data analysis platforms, Yahoo! is currently the main contributor to the open-source Hadoop project, which has produced HDFS, an open-source version of Googles GFS [7], together with an open-source version of Map-Reduce [5], and a SQL-like query interface called Pig (which we discuss below). There are several-thousand node Hadoop clusters in operation at Yahoo!, facilitating analysis of multi-petabyte datasets.

A very different approach is taken in the Everest OLAP engine. This is a parallel SQL query evaluation engine based on vertical storage, e.g., as in Vertica [10], designed for business intelligence applications. Everest uses a multi-tier fault tolerant architecture, and achieves high query performance by intelligently using vertical data storage to take advantage of clustering inherent in important query workloads.

Yahoo! also has several other specialized storage systems. The Vespa Data Store (VDS) stores partitioned, replicated data by assigning data blocks to servers according to a scalable hash function. It is primarily a document and metadata store.

Yahoo! Mail, which serves over 250 million accounts with unlimited storage, uses a sophisticated custom architecture in which different kinds of messages, different parts of messages, and messages over different periods of time are all carefully organized across different kinds of storage systems.

Another important data system is related to Web search. Crawling, storing, indexing, and searching tens of billions of pages that are constantly changing, poses several additional challenges. For example, indexing hundreds of terabytes implies sophisticated algorithms that runs in large clusters of computers. Query processing on the other hand, implies high dependability of servers and data as well as handling a large query throughput by using caching, distributing the index, and replicating part of it.

In the rest of this section, we briefly describe two ongoing Yahoo! research projects that have produced systems already in wide use within the company. For more details, see [4].

Sherpa and PNUTS. The PNUTS project is building a massively scalable, hosted data management service to provide back-end support for Yahoo!s web workloads. It is part of an integrated suite of data services called Sherpa, including a scalable message delivery service (YMDB) and globally distributed files organized by hashing (YDHT) or sorting (YDOT). Multiple applications concurrently

connect to the PNUTS service to store and query data. This shared service model addresses one of the main data problems faced by Yahoo! today: applications often have to set up, maintain and scale their own data platforms, a significant drain on business resources and an impediment to development of new features.

Four guiding principles have shaped the design of PNUTS. First, the system provides high performance at large scale by using asynchrony, weak consistency and loose coupling. Second, the system uses automated replication and failure recovery to ensure high availability. Third, the system is designed to be easy to use, operate and scale. Ease of use means that the external abstractions hide much of the complexity of the underlying distributed, replicated system. Ease of operation implies extensive self-management and self-tuning. Ease of scaling means that adding capacity is as easy as plugging in new machines and turning them on. Fourth, the system provides multiple rich access methods, including multiple types of primary tables and secondary indexing. PNUTS is both a research project and a key piece of Yahoo!s next generation platform architecture. The system is being designed and built as a collaboration between Yahoo! Research and Yahoo!s Platform Engineering group, and some initial components of the system have already entered production use.

PIG. For queries that perform wholesale analysis over data such as web crawls and search query logs, the PIG system, built on HDFS and Map-Reduce as part of Hadoop, leverages intra-query parallelism. The higher the degree of parallelism, the faster the response time for individual queries, which is critical for continuous queries and for ad-hoc R&D queries.

The basic idea is that if we have a data set of size $|D|$ to analyze, by distributing the load across n nodes can potentially yield linear speed up. Grouping or joining may require repartitioning the data, which only doubles the amount of data each node needs to handle. In practice, parallel query processing techniques that exhibit near-ideal scale-up when $n = 10$ or perhaps $n = 100$, do not continue to do so when $n = 1000$ or more. The PIG project at Yahoo! is studying new architectures and algorithms aimed at good scale-up for $n = 1000$ or even $n = 10,000$, so that we can answer ad-hoc queries over multi-terabyte data sets in minutes. We expect this capability to be a key enabler of R&D activity and business intelligence applications going forward.

4. DATA MINING

Strategic Data Solutions (SDS) is the main department behind data mining at Yahoo!. SDS has a daunting task: combing through the dozen or more terabytes of data that Yahoo users generate daily by clicking links, extracting out the relevant bits, compressing and storing them.

Of course, all that information would be useless without a way to make sense of them. According to Usama Fayyad, Chief Data Officer of Yahoo!, there are three main challenges [9]:

Reliability and Scalability. The first and largest challenge is the ability to capture from thousands of servers all of this data, in a reliable way, process and summarize it, to feed the many applications as well as data warehouses, data marts, dashboards, and scorecards across Yahoo!. You cannot fall behind, because you can never catch up if you do. Because this data stream is always growing you cannot just plan for the existing data load, but always be building ahead of the game.

Measurements. A second challenge is defining metrics that are central to the business and understandable by different units. Figuring out how to process the data and present the results is not easy, especially in the Internet space where things change fast and on an ongoing basis. This also includes keeping up with new pages and new products being launched almost on a daily basis—this is an environment that is very far from static.

Adaptability. The final challenge is related to the management of data mining models. We have to generate thousands of predictive and classification data mining models, updating them daily, and then using them to produce predictions in real time. A huge challenge is to have an adapting process to make sure that all these data mining models are updated, reading the correct information, and their outputs validated. Many companies find it challenging to run a handful of models; we have to run and maintain thousands. This is a scale that is unfamiliar to most data mining practitioners and it requires systematic and product-like thinking—not just analysis-oriented thinking.

Data mining can be directly used for improving the Web design of each Web site. However, it can also be used to find new useful functionalities. For example, what you see today on Yahoo! Mail is a visible result of data mining. Analyzing patterns in usage data, we observed a strong correlation between people reading email and news in the same session. When we shared this with the Yahoo! Mail product team, their first instinct was to test this effect: this was done by building a "news module that highlights news headlines and showing this to a test group of consumers as part of the main Mail front page. For a product like Mail, the critical business pain is to take new "light users, and turn them into "heavy users. If you do that, you reduce churn dramatically. Indeed, we showed from this test that churn in the weakest group was reduced by 40%. This led to the immediate development and launch of the News Module embedded in Yahoo! Mail's front page. Today, hundreds of millions of consumers see and use this product.

Another data mining application is advanced targeting techniques. For example, discovering who is in the market for a given product with high reliability, based on only anonymous browse data, is another big area of success for predictive data mining technology. This ability allows us to create very high-value audiences by understanding what they have in mind and biasing the advertising such that is more relevant to that audience: a win-win for consumers and advertisers.

However, we can go further and bootstrap data mining into Web search or generate pseudo-semantic resources that can later be used by other applications. This is one of the goals of Yahoo! Research. For example, consider that queries are implicit tags of clicked answers. That is, we can envision that queries in search engines or tags in Web 2.0 sites, represent the language of the Web or *web-slang*. Using graph mining techniques accross datasets we can find similar queries that have no common words and then infer semantic relations, which could be then organized in a user-driven taxonomy [2].

One problem of data mining in general is that many times is an ad-hoc process, because depends in what you are looking for and the intended application. To be able to explore data before building an ad-hoc tool, we are developing a data mining system, called WIM (Web Information Mining) [1]. WIM, has a data warehouse component and will have a visualization component. However, the most important component is a high level algebra that allows fast prototyping of Web mining applications, by relating Web content, links and usage. The algebra includes many types of operators, including sql-like constructs, link analysis, set manipulation, etc.

5. REFERENCES

[1] Ricardo A. Baeza-Yates, Álvaro R. Pereira Jr., Nivio Ziviani. WIM: An Information Mining Model for the Web. In *LA-WEB*, 233-241, 2005.

[2] Ricardo Baeza-Yates, Alessandro Tiberi. Extracting Semantic Relations from Query Logs. In *ACM KDD 2007*, San Jose, California, USA, August 2007, 76–85.

[3] Ricardo Baeza-Yates, Peter Mika, and Hugo Zaragoza. Search, Web 2.0, and the Semantic Web. IEEE Intelligent Systems 23, Jan/Feb 2008.

[4] Community Systems Group at Yahoo! Research. Systems, Communities, Community Systems, on the Web. SIGMOD Record, September, 2007.

[5] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters, Proceedings of OSDI, December, 2004.

[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 *Proceedings of SOSP*, October, 2007.

[7] S. Ghemawat, H. Gobioff and S.-T. Leung. The Google File System. In *Proceedings of SOSP*, 2003.

[8] J. Surowiecki. *The Wisdom of Crowds*. Random House, 2004.

[9] Usama Fayyad. Interview in SIGKDD by Gregory Piatetsky-Shapiro. `http://www.sigkdd.org/explorations/-issues/7-2-2005-12/fayyad.html`, 2005

[10] `http://www.vertica.com`

[11] Yahoo! Research Team. Content, Metadata and Behavioral Information: Directions for Yahoo! Research. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 2006.

[12] D. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, 2003.