Content, Metadata, and Behavioral Information: Directions for Yahoo! Research

The Yahoo! Research Team

Abstract

In mid-2005, Yahoo! Inc. began an ambitious program to create a world-class industrial research lab focusing on how to deliver services over the web to a range of stakeholders, including advertisers, site owners, content publishers, and users. The resulting organization, Yahoo! Research, has embarked on a number of research directions. In this document, we report on these directions, with a particular focus on the data engineering problems we see as critical to our mission.

1 Introduction

Yahoo! Research (Y!R) sits at the nexus of more than half a billion unique users per month, several terabytes per day of data of various forms, several billion dollars of advertising revenue, and over one hundred distinct properties offering capabilities ranging from email, online news, and Web search to classifieds, personals, and horoscopes. Our goal is to shape the future of the internet, and we are focusing on a number of key research directions that we consider to be foundational.

Within Y!R, there are teams focusing on five research areas: *search and information retrieval, machine learning and data mining, microeconomics, community systems*, and *media experience and design*. Most projects draw from several of these five areas collaboratively. Common themes have emerged in these projects, many of which center around a combination of massive datasets and the Web as the delivery channel. In this overview, we highlight some key research directions rather than cataloging activities in each focus area, in order to reflect the synergy across areas at Y!R. In subsequent sections, we will cover work taking place in *platforms, advertising* and *search.*

Before describing the work itself, we will briefly describe the environment in which we see Yahoo! operating in the future. Figure 1 illustrates the situation. One may view Yahoo! as a match-maker that aggregates user attention by providing a wide range of high-quality content and community. We collaborate with content providers and site owners to accomplish this, and we then sell differentiated streams of user attention to advertisers, whose dollars flow to Yahoo!, as well as our partners, in order to fund the next round of improvements to the content and user-experience. We view all four stakeholders as central to our research initiatives, and we will give a more detailed perspective on how each research direction allows them to engage more effectively.

Within this environment, we see three overarching trends that we believe will shape the future of online interactions:

The emergence of structure. The highly heterogeneous data types we see range from text and html documents to query logs, user profiles, vertical content, listings, many forms of advertisements, user interactions, images,

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

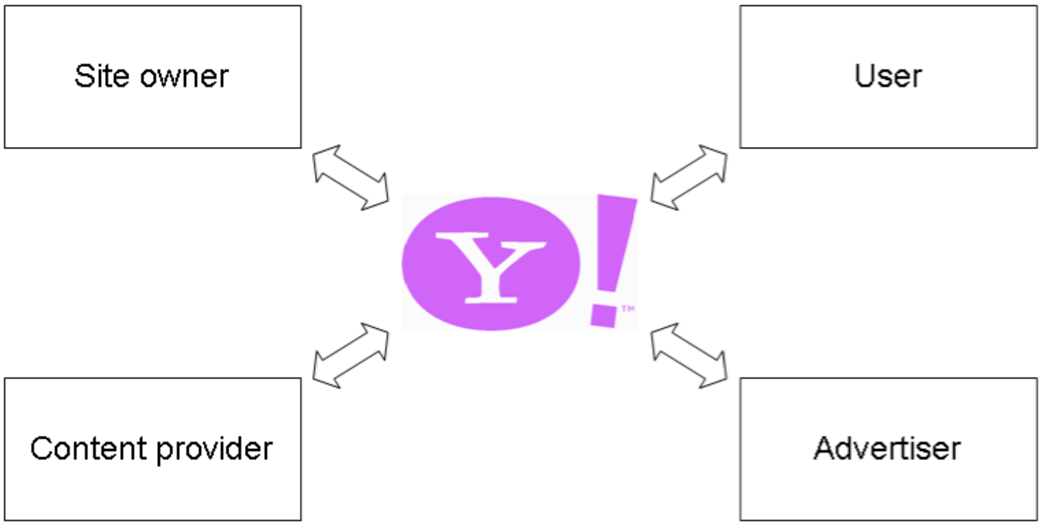

Figure 1: Key players in the Yahoo! ecosystem.

and video. In all these cases, there are distinct and interesting types of structure that can be useful both within and across datasets. 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, we observe that capture of structure is also nontrivial: in many cases structure is either user-provided (and sometimes designed to mislead) or automatically extracted (and potentially error-prone).

The use of many different types of large-scale data could benefit from additional, integrated structure. First, existing content repositories often do not expose structure explicitly (e.g., individuals' home pages often contain contact information, but embedded in free text), and even when they do, rarely conform to uniform metadata standards (e.g. *price* on one shopping site might correspond to *cost* on another), and are not readily interoperable. Structure extraction technology ranging from entity extraction to site wrappers is currently applied only to a limited degree; it is likely this will change. Second, we have massive assets of user behavioral information that could significantly improve the user experience. But this information is not yet fully exploited due to data scale, the imperative to safeguard privacy, and the need to marry increasingly sophisticated user modeling to application logic. Finally, we see a growing stream of user-contributed content becoming a key resource. In many cases, it conforms to a large and growing set of schemas meant to incorporate specifics of either data type (question/answer pairs, new articles, blog posts, Web pages), or domain data (auto repair, medical information, etc). As a variant of user-contributed content, we see user-contributed metadata in the form of reviews, tags, bookmarks, ratings, or even clicks.

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 [8]. Increasingly, online applications in diverse domains must incorporate community tools simply as table stakes, and the richness of the baseline community offerings is growing rapidly. We routinely see bulletin boards with presence and real-time chat, tagging and folksonomy data arrangement, rating and reputation systems, and other such capabilities, on a wide range of sites. Further, as Web application development methodologies mature, it becomes easier for developers to "drop in" such capabilities. Today, it is simple to grab modules providing rich community semantics based on a local MySQL instance. But already the seeds are visible for hosted application providers that make such capabilities available, with potential advantages such as single authentication, integrated payment systems, and shared personalization. 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 (e.g., email) and collaborative (e.g., IM, Y! Answers) 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. Furthermore, the landscape shifts very quickly, making it difficult to settle on business models and interaction paradigms. For example, advertisers are only now beginning to understand per-impression advertising (in which an advertiser is charged when a user loads a page containing a certain ad), and already, AJAX and related technologies are calling into question the entire notion of pageviews, as different parts of a page may update asynchronously in response to user activity or exogenous events.

With these themes in mind, we proceed as follows. Section 2 describes key directions in platform design and development. Then Sections 3 and 4 describe advertising and search respectively, which are key consumers of platform technologies; we conclude in Section 5.

2 Platforms

Data is central to Yahoo!s business. Some common workload patterns are:

- **Nugget storage/retrieval.** User-facing properties constantly store and retrieve data nuggets. In some cases all that is required is `get` and `put` based on a key, e.g. *store/retrieve Bob's stock portfolio*. Increasingly, more complex retrieval tasks are required, e.g. *get Bob's friends' favorite restaurants in cities on Bob's travel itinerary*.

- **Canned large-scale processing.** Many properties additionally rely on large-scale preprocessing of massive data sets. The obvious example is search, which performs link analysis over the many-billion page Web graph, to generate features used in ranking. However, many other Yahoo! properties also rely on large-scale data preprocessing, e.g. classification, clustering, entity extraction or wrapper induction over large corpora of webpages, news or advertisements. Typically, the same processing is run periodically, with every new version of the data set. Some cases require very short periods, in the limit becoming continuous queries over streams.

- **Ad-hoc large-scale processing.** Yahoo! analysts are constantly poring over our data sets, looking for patterns to inform the design of new products (and tune existing ones). Data sets of interest include Web crawls, click streams, advertisement entries, and various derived data. These "R&D queries" tend to be ad-hoc, and make heavy use of user-defined transformations, aggregation, and in some cases joins.

While these workloads are reminiscent of ones that have been studied extensively in the past, the sheer magnitude of the data creates new challenges. Extreme parallelism is the name of the game. We have two research initiatives on this front, one aimed at storage/retrieval workloads (Section 2.1) and one aimed at large-scale processing workloads (Section 2.2).

2.1 High-throughput storage/retrieval in an enormous database

We are investigating challenges underlying a very parallel storage/retrieval service that can support indexes for social-search and geo-search, and handle the types of workloads that arise in delivering a range of advertising, structured search, and community-oriented applications.

As we stated earlier, the Web has become a channel for delivering a range of end-user and collaborative applications to many users. Where users and organizations once had no option but to buy and install a stack of software ranging from database systems to middleware to specific applications, and to provision and maintain complex hardware and software installations, now they have the choice of going to a Web site such as Yahoo! or SalesForce.com and simply creating an instance of the application they want (e.g, an email account, a new Yahoo! group, or collaborating online in a social networking site such as Flickr or Y! Answers.)

The growing popularity of the application-as-service model makes it inevitable that more and more applications will become available as online services. This is a disruptive change to the way applications, and in particular data-driven applications and database management systems, have been traditionally designed. Now, we are challenged to develop systems that must be incredibly inexpensive on a per-application instance basis (the revenue from a Y! group cannot pay for dedicated hardware!), be easy to manage (a few operations people must oversee hundreds of thousands of groups), and very robust (since failures potentially affect thousands of application instances). One way to see this challenge is that we must build systems that allow us to be cost-effective database administrators (not to mention application developers and maintainers!) to the world, rather than vendors of DBMS software. On the other hand, since a given installation is intended to support many instances of a given application template, we have extensive knowledge of the workload, which is likely to be comprised of large numbers of requests (updates or queries) drawn from a relatively small set of request types. Similarly, there is often considerable latitude in the kinds of inconsistency that can be tolerated on specific requests.

Building systems to provide online applications as a service is one of the most intriguing challenges faced by the database community in many years. It requires us to leverage the lessons we have learned from designing parallel [3] and distributed database systems [9, 5], and to develop a new class of massively parallel, self-tuning, robust systems. It is likely that we will have to rethink almost ever aspect of database systems — What kinds of data? What mix of relational and text-based ranking queries? What models of concurrency and consistency? — to achieve the performance criteria required. A number of internal storage solutions at Yahoo! address these issues; see also [1].

2.2 Analytical processing over enormous data sets

For queries that perform wholesale analysis over data such as web crawls and search query logs, we focus on 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.

In principle, if we have a data set of size $|D|$ to analyze, and we divide the data and processing among n nodes, then each node need only handle around $|D|/n$ data. Grouping or joining may require repartitioning the data, which only doubles the amount of data each node needs to handle. In general, if data is to be repartitioned k times, we expect $(k + 1) \cdot |D|/n$ units of work per node. Hence for a given query, doubling the number of nodes should halve the response time.

Unfortunately it is not possible to achieve this ideal scale-up behavior in practice. Moreover, we are finding that 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. We are 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 [2, 7, 4, 6]. We expect this capability to be a key enabler of R&D activity and business intelligence applications going forward.

As our platform becomes used to provide analytic services to a large number of internal R&D "customers," a second major challenge will emerge: intelligent physical design to optimize the overall workload. In the parallel data management space, physical design includes data partitioning, as well as the usual degrees of freedom with respect to the choice of indexes and materialized views. We expect this problem to be very difficult to solve, based on the experience in other distributed computing environments such as grid computing, but it is also very important. If we do not do a good job, then our "customers" will not accept physical data independence, and will spend valuable cycles tinkering with the physical design by hand.

3 Advertising

Web advertising spans Web technology, sociology, law, and economics. It has already surpassed some traditional mass media like broadcast radio and it is the economic engine that drives Web development. It has become a fundamental part of the Web eco-system and touches the way content is created, shared, and disseminated—all the way from static html pages to more dynamic content such as blogs and podcasts, to social media such as discussion boards and tags on shared photographs. This revolution promises to fundamentally change both the media and the advertising businesses over the next few years, altering a $300 billion economic landscape.

As in classic advertising, in terms of goals, Web advertising can be split into *brand advertising*, whose goal is to create a distinct and favorable image for the advertiser's product, and *direct-marketing advertising*, which involves a "direct response": buy, subscribe, vote, donate, etc., now or soon.

In terms of delivery, there are two major types:

1. *Search advertising* refers to the ads displayed alongside the "organic results on the pages of search engines. This type of advertising is mostly direct marketing and supports a variety of retailers from large to small, including micro-retailers that cover specialized niche markets.

2. *Content advertising* refers to ads displayed alongside some publisher produced content, akin to traditional ads displayed in newspapers. It includes both brand advertising and direct marketing. Today, almost all non-transactional Web sites rely on revenue from content advertising. This type of advertising supports sites that range from individual bloggers and small community pages, to the web sites of major newspapers. There would have been a lot less to read on the Web without this model!

Web advertising is a big business, estimated in 2005 at $12.5B spent in the US alone (Internet Advertising Board—www.iab.com). But this is still less than 10% of the total US advertising market. Worldwide, internet advertising is estimated at $18B out of a $300 total. Thus, even at the estimated 13% annual growth, there is still plenty of room to grow, hence an enormous commercial interest.

At Yahoo! Research we are exploring designs for next generation advertising platforms for contextual and search ads. From an ad-platform standpoint, both search and content advertising can be viewed as a matching problem: a stream of queries or pages is matched in real time to a supply of ads. A common way of measuring the performance of an ad-platform is based on the clicks on the placed ads. To increase the number of clicks, the ads placed must be relevant to the user's query or the page and their general interests.

There are several data engineering challenges in the design and implementation of such systems.

The first challenge is the volume of data and transactions. Modern search engines deal with tens of billions of pages from hundreds of millions of publishers, and billions of ads from tens of millions of advertisers. Second, the number of transactions is huge: billions of searches and billions of page views per day. Third, there is only a very short processing time available: when a user requests a page or types her query, the expectation is that the page, including the ads, will be shown in real time, allowing for at most a few tens of milliseconds to select the best ads.

To achieve such performance, ad-platforms usually have two components: a *batch processing component* that does the data collection, processing, and analysis, and a *serving component* that serves the ads in real time. Although both of these are related to the problems solved by today's data management systems, in both cases existing systems have been found inadequate for solving the problem and today's ad-platforms require breaking new ground.

The batch processing component of an ad-system processes collections of multiple TB of data. Usually the data is not shared and a typical processing cycle lasts from a few minutes to a few hours over a large cluster of hundreds, even thousands of commodity machines. Here we are concerned only about recovering from failures during the distributed computation, and most of the time data is produced once and read one or more times. Commercial database systems deployed on the same scale would be prohibitively expensive. The reason for this is that database systems are designed for sharing data among multiple users in the presence of updates and have overly complex distribution protocols to scale and perform efficiently at this scale. The backbone of Web data batch processing components (see Section 2.2) is therefore being built using simpler distributed computation models and distributed file systems running over commodity hardware. Several challenges lie ahead to make these systems more usable and easier to maintain. The first challenge is to define a processing framework and interfaces such that large scale data analysis tasks (e.g., graph traversal and aggregation) and machine learning tasks (e.g., classification and clustering) are easy to express. So far there are two reported attempts to define such query languages for data processing in this environment [6, 7]. However, there has been no reported progress on mapping these languages to a calculus and algebra model that will lend itself to optimization. Conversely, to make the task easier, new machine learning and data analysis algorithms for large scale data processing are needed. In the data storage layer, the challenge is to co-locate data on the same nodes where the processing is performed.

The serving component (see Section 2.1) of an advertising platform must have high throughput and low latency. To achieve this, in most cases the serving component operates over a read-only copy of the data replaced occasionally by the batch component. The ads are usually pre-processed and matched to an incoming query or page. The serving component has to implement business logic that, based on a variety of features of the query/page and the ads, estimates the top few ads that have the maximum expected revenue within the constraints of the marketplace design and business rules associated to that particular advertising opportunity. The first challenge here is developing features and corresponding extraction algorithms appropriate for real-time processing. As the response time is limited, today's architectures rely on serving from a large cluster of machines that hold most of the searched data in-memory. One of the salient points in the design of the ad server is an objective function that captures the necessary trade-off between efficient processing and quality results, both in terms of relevance and revenue.

In summary, today's search and content advertising platforms are massive data processing systems that apply complicated data analysis and machine learning to select the best advertisements for a given query or page. The sheer scale of the data and the real-time requirements make this a very challenging task. Today's implementations have grown quickly, and often in an ad-hoc manner, to deal with a $15B fast growing market. There is a need for improvement in almost every aspect of these systems as they adapt to even larger amounts of data, traffic, and new business models in Web advertising.

Looking beyond, in the context of auctions for internet advertisement, microeconomics and data analysis go hand in hand. Using the tools of microeconomics one can make theoretical predictions regarding how advertisers would respond to changes in an auction mechanism (such as changing reserve price or allowing advertisers to submit separate bids for ads targeted to different demographics). With careful data analysis one can estimate how such changes would impact Yahoo! and its users and advertisers. Doing this kind of data analysis is possible only with a high-performance advertisement platform.

4 Search

The Search focus area at Yahoo! is broadly concerned with research that enables users to satisfy an information need. The scoping is intentionally broader than Web search alone. Yahoo!'s content portfolio ranges well beyond simple Web pages, and the types of objects that must be retrieved are broadly heterogeneous, and in some cases extremely complex. For example, a search across a selection of publicly-accessible bulletin boards could return an appropriate message, a thread consisting of hundreds of posters and thousands of posts, or a board with thousands of members. In the near future, it may well return a much richer structure containing multiple overlapping embedded bulletin boards, multimedia content, questions and answers, news articles, perhaps an online marketplace, and so forth. Effective ranking over such complex objects is well beyond the state of the art.

Search technology is in flux. It is common for employees even at technically sophisticated organizations to have access to enterprise search tools that are markedly inferior to what is available on the Web, despite the much higher average quality of enterprise content. We are pursuing four key research directions, as discussed below.

4.1 User-generated metadata analysis

The first direction is classically responsible for the success of Web search. It involves the incorporation of social information: there are key cues (such as incoming anchortext) that allow internet search engines to employ the aggregate perspective of an enormous number of users in deciding which results to return. And we are currently at the center of a rapid shift in the nature of this social input. Historically, search engines have drawn primary leverage from hyperlinks and anchortext. In other words, most of the input has come from a relatively small number of elite Web users who are capable of authoring and serving html content. Today, however, the number of distinct users generating useful metadata is growing rapidly due to three factors. First, the emergence of simple Web authoring tools such as hosted blogging software makes it possible for authorship to migrate from the elite to a much larger base of online users motivated to express themselves. Second, the introduction of new models for explicit creation of metadata versus content, such as tagging and bookmarking (e.g., through `del.icio.us`), the creation of rich profiles (e.g., `myspace.com`), or even the creation and publishing of multimedia content (e.g., `youtube.com`) lowers the barrier from authorship to lighter-weight interactions like commenting on somebody else's content. And finally, there are situations in which content consumption itself is a generator of useful metadata. For example, when a search engine shows a list of results, the result clicked by a user is potentially helpful information. In the extreme, companies such as Google allow users optionally to share their entire browsing behavior with the search engine. The privacy implications of such sharing have not yet been fully explored, but if the social contract between users and online tools expands to include such activities then another order of magnitude of data scale becomes available to help us understand the quality of content.

4.2 Aggregate analysis

The second key direction in search is the aggregation of data and metadata across multiple dimensions, at multiple levels of granularity. By aggregation, we mean something more than simple rollups using straightforward algebraic functions; instead, we mean sophisticated analysis making use of all information available for a natural grouping of content objects. For example, we might consider all pages on a website, or all content related to medical matters, in order to extract common topics and classify content according to which of the common topics are being discussed. We will speak to the key forms of aggregation in one more level of detail.

First, there is aggregation at the level of *content sources*, for instance a website, community, or blog, which may be viewed as a cohesive generator of potentially useful content over time. Understanding the reputation, topical focus, authorship, affiliation, quality, and reliability of each content source becomes a key advantage in aggregating content from these sources. The aggregation may happen at multiple levels, such as a website and a key subdirectory or subsite of the website. The type of information to be aggregated is not limited to information extracted from the content itself. For instance, it is clearly useful to know if there are coherent groups of users who are the typical exploiters of a content source.

Second, in addition to aggregation at the level of content sources, there are also cross-source forms of aggregation, such as topical aggregation (as in our earlier example of analyzing all content around medical matters), target aggregation (for instance, analyzing the content which draws 14-16 year old male viewers), entity aggregation (all content regarding George Bush, or the Samsung HLS6187 high-definition TV), and so forth.

4.3 Cross-content analysis

The third trend is analysis to extract insights that are accessible only in the combination of multiple giant databases of disparate schemas. This form of analysis arises in two settings. We begin with the first and simpler setting, which is related to federated search or distributed information retrieval.

Queries to a Web search engine are primarily answered by providing a link to a relevant Web page, but increasingly, other sources are being federated in. For example, queries for local restaurants, or musical artists, or movies, or products already receive special handling, as do queries that may be syntactically identified as belonging to a special class, such as airline flights or package tracking identifiers. These tricks may be performed with minimal query processing overhead. Additionally, however, parallel queries to other corpora such as collections of images or video are being performed in Web search to determine whether other classes of content might be appropriate. Yahoo! already federates responses from its Answers corpus, and Google's Coop allows sophisticated users to generate plugins directing certain queries in parallel towards alternate user-specified corpora. Of particular interest to the data management community is the trend towards including structured records in results, as for instance in Google Base or Yahoo! shortcuts to shopping data.

Generally, the move towards retrieving results from numerous disparate corpora is well on its way. However, as yet there is no truly successful work on providing the user with a single cohesive set of results. The best of breed approach is simply formatting the result page so that it is clear which results come from which corpus, and employing only the simplest techniques to merge modules from each relevant corpus onto a result page.

Finally, there is a second setting in which cross-content analysis arises; this latter setting moves beyond the scope of finding results from multiple corpora, namely, the problem of simultaneous batch mining of multiple datasets in order to extract insights or results that are unavailable from any collection singly. For instance, the combination of Web content, site-aggregated metadata, click data, and publicly available user profile data might give a perspective on potential query responses that is far beyond the current model.

4.4 Community analysis

Increasingly, there is now widespread acceptance that meeting a user's information need on the Web is more than a matter of human-computer interaction: it is also a matter of human-human interaction. As an example, consider the remarkable success of "knowledge search," as provided by Naver in Korea and many other related offerings, both in Southeast Asia and increasingly elsewhere in the world (for instance, Yahoo! Answers is a global offering of this form). A user enters a question, which is routed to other online users who might be able to provide an answer. Typically multiple answers are received in well under a minute. Clearly in this case traditional approaches to indexing have little to contribute, and whole new questions around answerer competencies become relevant. Further, there is a completely new question regarding the fusion of online knowledge (either in traditional documents, or in other forms such as answers to prior questions) with human knowledge that resides in wetware but might be accessible at significant latency through the paradigm of asking.

Community analysis throws open a wide array of questions in microeconomics, including incentives and markets that emerge as a result of online human-human interactions. Microeconomic models can be used to derive effective reputation feedback mechanisms that are resistant to gaming. Experimenting with intelligently designed reputation feedback systems and carefully analyzing the data generated by such experiments can help to improve such mechanism and to identify ideas that work in practice as well as predicted by the theory.

5 Conclusions

In this paper, we have outlined three key trends in the future of online interactive media: heterogeneous structured content; online social systems; and web-delivered applications. Based on these trends, we have described a number of heavily data-centric research directions that Yahoo! Research will pursue, in the areas of platforms, advertising, and search.

References

[1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. Gruber. Bigtable: A distributed storage system for structured data. In *Proc. 7th Symposium on Operating System Design and Implementation*, pages 205–218, 2006.

[2] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *Proc. 6th Symposium on Operating System Design and Implementation*, pages 137–150, 2004.

[3] D. J. DeWitt, S. Ghandeharizadeh, D. A. Schneider, A. Bricker, H.-I. Hsaio, and R. Rasmussen. The Gamma database machine project. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):44–62, 1990.

[4] M. Isard, M. Budiu, Y. Yu, A. Birrell, and D. Fetterly. Dryad: Distributed data-parallel programs from sequential building blocks. Technical Report MSR-TR-2006-140, Microsoft Research, October 2006.

[5] J. B. R. Jr., P. A. Bernstein, S. Fox, N. Goodman, M. Hammer, T. A. Landers, C. Reeve, D. W. Shipman, and E. Wong. Introduction to a system for distributed databases (SDD-1). *ACM Transactions on Database Systems*, 5(1):1–17, 1980.

[6] C. Olston, B. Reed, R. Kumar, D. Meredith, U. Srivastava, and A. Tomkins. Querying enormous data with Pig, 2006. Manuscript.

[7] R. Pike, S. Dorward, R. Griesemer, and S. Quinlan. Interpreting the data: Parallel data analysis with Sawzall. *Scientific Programming Journal*, 13(4):277–298, 2005.

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

[9] R. Williams, D. Daniels, L. Haas, G. Lopis, B. Lindsay, P. Ng, R. Obermarck, P. Selinger, A. Walker, P. Wilms, and R. Yost. R*: An overview of the architecture. In P. Scheurman, editor, *Improving Database Usability and Responsiveness*, pages 1–27. Academic Press, New York, 1982.