Data Management Projects at Google

Wilson Hsieh
Google Inc.
Mountain View, CA
wilsonh@google.com

Jayant Madhavan
Google Inc.
Mountain View, CA
jayant@google.com

Rob Pike
Goolge Inc.
Mountain View, CA
r@google.com

1. BIGTABLE: A HIGHLY SCALABLE SYSTEM FOR DISTRIBUTED STRUCTURED STORAGE

Wilson Hsieh

Bigtable is a distributed storage system developed at Google that is designed to scale to very large sizes: petabytes of data across thousands of commodity servers. Many projects at Google store data in Bigtable, including web indexing, Google Print, Google Earth, Orkut, My Search History, Google Video, Google Analytics, and Google Finance. These applications place very different demands on Bigtable, both in terms of data size (from URLs to web pages to satellite imagery) and latency requirements (from backend bulk processing to serving data to users). Nonetheless, Bigtable has successfully met all of these applications' storage needs, because of its simple data model and scalability.

In this talk I will describe Bigtable's data model; give an overview of its implementation and how it fits into Google's infrastructure; and provide some measurements of its performance and how well it scales.

2. GOOGLE BASE: WHEN STRUCTURED DATA MEETS THE INTERNET

Jayant Madhavan

Google Base, launched in late 2005, is a new offering from Google that enables users to *put stuff into Google*. Anyone with a valid Google account can post items and have those items show up as search results to subsequent user queries. Base thus allows content producers to pro-actively add their content to Google rather than relying on crawlers to automatically find it.

Base allows users to *structure* their uploaded content. Each item is annotated by identifying an item type and a set of attributes (with values) that describe the item. For example, a posting for a used 2004 VW Jetta, will typically have the vehicle item type and have attributes such as make = Volkswagen, model = Jetta, year = 2004, and so on. In addition, there can be a url that points to an external web page describing the car and also a link to a picture of the vehicle.

Likewise, Base enables users to exploit item annotations while querying. User queries are sets of keywords and in response the search result is a ranked list of items relevant to that query. In addition, the result page typically includes a list of attributes that are most relevant to the input query. For each of these attributes, there is a select box with values of the attributes that are most relevant to the search results. For example, the query "Jetta 2004 San Francisco" results in attribute value lists for Price, Color, Condition, and Model, attributes that are relevant to cars. A user at this point can use the attributes and their values to refine and re-rank the search results. In the example, the user can pick a price range from the values displayed for Price or a color from the values of Color to refine the search results. Any structure information made available by users is thus used to enhance the users search experience.

The ability to let users post items has been successfully employed in the past in the context of market-place and community portals such as Ebay and Craigslist. What sets Base apart is that there no restrictions on how people choose to structure their items. Though for popular item types Base has a set of recommended attributes, these are merely suggestions and are not required. Each item can have any attribute and any corresponding value and there is no restriction either on the type of each item. This has led to Google Base having structure on a varied content covering from popular areas like vehicles, real-estate, people profiles, and personals, to more esoteric ones like wines, genealogy, and ancient coins.

Google Base is thus a very, very large, self-describing, semi-structured, and heterogeneous database. Each item in Base comes with its own schema and data values. Items of the same type can share attributes and have the same values for those attributes. Such a free-flowing semi-structure database presents many challenges stemming from heterogeneity to many opportunities stemming from innovative ways to exploit the schema in the content.

In this talk, we will give an overview of the Google Base, especially aspects of structured data processing. Google Base is alive and well and can be queried at http://base.google.com.

SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.

3. INTERPRETING THE DATA: PARALLEL ANALYSIS WITH SAWZALL

Rob Pike

Very large data sets often have a flat but regular structure and span multiple disks and machines. Examples include telephone call records, network logs, and web document repositories. These large data sets are not amenable to study using traditional database techniques, if only because they can be too large to fit in a single relational database. On the other hand, many of the analyses done on them can be expressed using simple, easily distributed computations: filtering, aggregation, extraction of statistics, and so on.

We present a system for automating such analyses. A filtering phase, in which a query is expressed using a new procedural programming language, emits data to an aggregation phase. Both phases are distributed over hundreds or even thousands of computers. The results are then collated and saved to a file. The design—including the separation into two phases, the form of the programming language, and the properties of the aggregators—exploits the parallelism inherent in having data and computation distributed across many machines.

This is joint work with Sean Dorward, Robert Griesemer, and Sean Quinlan.

http://labs.google.com/papers/sawzall.html