ORACLE
NOSQL DATABASE

NoSQL Database for Online Display Advertising

Introduction

Browsing content on the web today is largely funded through paid advertising. More specifically, publishers have monetized their content by giving advertisers the opportunity to display ads on their sites. Targeting ads to users who have exhibited interest in a particular area (for example, sports cars enthusiasts) greatly increases the value of the publisher's site. To enable such highly targeted ads, publishers must have a highly scalable, highly available, and very low latency store to perform lookups (or possibly updates) when a web page loads.

How it works

Figure 1 depicts a simplified flow of an "impression" from the user's browser to the server-side systems responsible for deciding whether or not to serve an ad, and if so, which ad will actually get served into the browser. An impression is considered to be a user that is browsing a web page. The entire decision process must complete and return to the browser within fifty milliseconds as this logic is all executed at web page load time. In general, publishers will not tolerate high page load times due to the ad serving process as one of the key components of user affinity to a site is the performance of web page loading into the browser.

When a web page loads, it will contain an "ad tag," which is a simple snippet of JavaScript that will perform an HTTP request back to the ad serving system. On the server side, some key operations must be performed within an extremely small latency window:

- Have we encountered this user before, and if so, how recently and at what frequency have we encountered this user ? Furthermore, what behavioral segments have been associated with this user?
- If we have not encountered this user, save the user's cookie along with any behavioral segments this user may be associated with.
- Update the timestamp that is associated with this user encounter.

The following key points should be noted regarding these operations:

- Scale – The scale of incoming requests is large (internet scale) as is the accumulated data. Medium-scale publishers may encounter up to 500 million impressions a day. Large-scale publishers may encounter several billion impressions a day.
- Low latency – As this is only a small portion of the logic to be executed during the ad serving process, the upper threshold for this part of the process is typically twenty milliseconds.
- Availability – Publishers require five nines of availability for their monetization solutions.

Published April 2012

Audience Segment Management Requirements

As stated earlier, the behavioral segments associated with users (hereafter referred to as audience segments) must be quickly stored for future retrieval. Typically a publisher will use something similar to the following mechanism for the audience association operation:

> Targeted content tagged with pixel beacon – Publishers that create highly targeted content, for example, a sports site that contains a page for NBA basketball player statistics may choose to place a 1x1 pixel beacon on the page. This pixel beacon is an undetectable JavaScript call to a server that will pass the segments "sports enthusiast, basketball lover" back to a server. As the user browses more and more targeted content, the server builds up more and more associations between the cookie and the audience segments.

In the "sports enthusiast, basketball lover" example above, let's presume that this user subsequently browses the news section of the publisher's site. Upon loading the daily news page, an ad call is encountered; an HTTP call is made to the ad server. The ad server looks up the user's cookie and finds that this user is a "sports enthusiast, basketball lover. " The ad server will then search for advertising campaigns that have been booked against this site. Let's say the following three ads are found:

- Mortgage company – A mortgage company has purchased an ad campaign on the news site for $5 per thousand views.
- Online retailer – An online retailer has purchased an ad campaign on the news site for $3.50 per thousand views.
- Basketball shoe manufacturer – A basketball manufacture has purchased an ad campaign on the news site for $10 per thousand views, if and only if those viewers are known basketball enthusiasts.

To maximize the revenue to the publisher, the ad server will almost certainly choose to display the ad from the basketball shoe manufacturer (barring other issues like frequency capping and pacing). Once the ad has been served, the ad server must update the budget for the campaign as well as get this data into the distributed map/reduce cluster for offline analysis.

The Oracle NoSQL database provides a perfect complement to the RDBMS and map reduce cluster in this technology stack. We see the requirements of these three critical components working hand in hand to provide an end-to-end solution;

- Oracle NoSQL Database – Provides critical linear scaling and extremely low latency for those operations closest to the user's browser.
- Oracle RDBMS – Provides critical transaction support and ad hoc financial reporting capabilities.
- Map/Reduce cluster – Provides linear scaling for extremely large scale offline data analysis and model building.

Published April 2012

Other data management and processing

Complementing the Oracle NoSQL architecture are two other data processing components that are critical to the overall solution. As depicted in Figure 1, we see the relational database management system such as the Oracle Database serving up real-time, ad hoc reporting. This critical solution component enables the publisher to perform the following business operations;

- Campaign booking – Advertisers will interact with a classic web application to create campaigns as well as schedule these campaigns to start running.
- Monitoring the current ad campaign performance. Are campaigns being served appropriately?
- Monitoring revenue to date, quarter of quarter, for targeted ads, for untargeted ads.
- Understanding site performance. Are there certain sites that are not attracting targeted visitors? How many campaigns this month have been booked on each site?

Downstream data aggregation is performed by a distributed map/reduce cluster due to the scaling factor of the accumulated data. The size of this data set will typically be in the range of one to five terabytes daily, making a clustered map/reduce architecture a perfect choice for aggregation. In many cases, map/reduce aggregates are then stored back into the RDBMS to provide the publisher with ad hoc reporting functionality. This aggregated data provides the answers to the following business questions:

- What is the audience composition of my sites?
- How many unique visitors do I encounter by week or by month?
- How many impressions in a particular segment (e.g. sports enthusiast) are forecasted to arrive in the next week, month, and quarter and can I commit these impressions to an advertiser?

In summary, the distributed Key-Value store, the batch processing map-reduce system, and the relational database management system work together in order to provide important business value for massive-scale, high throughput, online display advertising system.

Published April 2012

System components

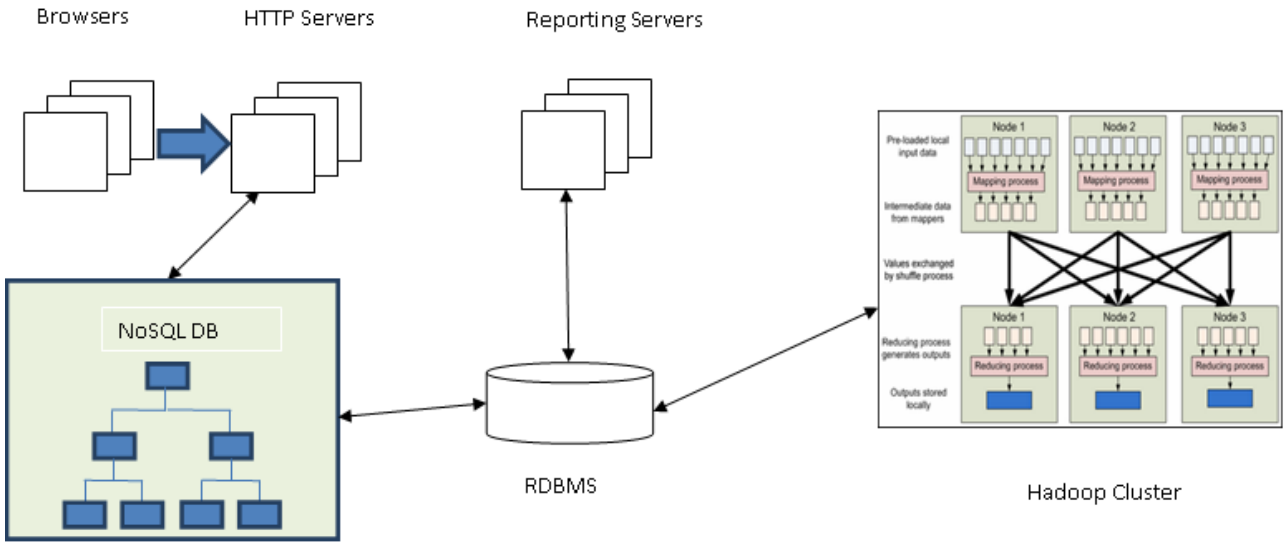

Figure 1: Architecture for Online Display Advertising

Published April 2012