Web Analytics: The Way of Data Analysis

Sudeep Srivastava
EMPI B School, CSKM Educational Complex, Satbari Chattarpur, New Delhi – 110074, India.
Sudeep.srivastava@empi.ac.in and mca2005sudeepsri@gmail.com

ABSTRACT

Web Analytics has become a critical component of many business decisions. In this paper, I describe the importance and intricacies of summarization for analytics and report generation on web log data. I speciûcally elaborate on how summarization is exposed in Organization and discuss analytics search design trade-offs.

***Key Words*- summarization, analysis, analytics, data, and organization, design, critical, summery, web site, web, log**

I. INTRODUCTION

Modern websites contain a wealth of content to provide easy and efficient access to information about an enterprise. These websites range from simple static html to very sophisticated dynamic content. While hosting such content comes with its own set of challenges, such as resource provisioning and traffic anomaly detection, an even larger challenge is to identify business insights based on web access patterns. The most useful source of such insight is the web access log. Mining these logs can provide information on what happened, what to anticipate and how well things are working.

Many enterprises rely on web analytics for business intelligence, i.e., to evaluate and optimize their business decisions. Mining logs can help answer questions such as what search terms are effective in

directing trafûc to our website and how different are our visitor demographics now compared to last quarter. Such information can help inform content layout and search engine ad word campaigns.

Web data, although immensely informative, can often contain millions of events per second based on trafûc volume and verbosity of logging. Much of web analytics tries to extract useful patterns and statistics periodically to generate regular reports. Thus, lthough the data itself changes frequently, the report-generating searches seldom change.

With high volumes of data, it can take hours to generate reports across data spanning large time windows. If daily reports take hours to execute, very little time is left to investigate anomalies and make changes in time for the next roll out. I propose using data summarization to efficiently search large quantities of log data. Data summarization involves running a search at regular intervals to extract information from the raw data. The results of this search are stored in a summary index, against which subsequent searches and reports can be run. Since the summaries of the data are smaller, these searches execute much faster than those run against the raw data. Furthermore, if many different reports have common search elements, the common sub searches can be summarized to make multiple reports run efficiently. Even for the same report, it is more efficient to run the report against varying time granularities. For example, a daily summary of the data would signiûcantly speed up a weekly report on the data as well as a monthly report on the data. As a result, the computational cost of running a search over large volumes of data is amortized over time by running the search periodically on much smaller quantities of data.

In this paper, I examine data summarization applied to the speciûc problem of log-based web analytics. I store and perform analytics on web access logs.

The remainder of this paper resumes by comparing various approaches to web analytics and details what information web logs contain, also I explain in depth how data summarization is implemented in organization. I also discuss various challenges involved when searching summarized data instead of raw data.

II. WEB ANALYTICS BACKGROUND

There are many well-known commercial products for performing web analytics. Rather than delve into detailed feature lists for these product I thought it best to focus on guiding principles for what data is used, how much of it is stored and implications at retrieval time.

The ûrst and foremost consideration is what data to use for web analytics. Traditionally there have been two schools of thought on what web data is used:

a. JavaScript tags per page

b. Web access logs

The largest difference between these two data collection methods is the degree of invasiveness of the approaches.

While this approach allows ûne-grained instrumentation of web pages, the downside is that instrumen tation can be invasive and cause signiûcant slowdowns in page load times. Furthermore, if an organization has strict cyclical website releases, any analysis that requires additional/new instrumentation must wait until the next release cycle. Analytics that solely relies on javascript beaconing also suffers from insufûcient information to perform historical trend analysis, especially if the web site, or even pages, have evolved over time. The second school of thought solely relies on web server logs for an alytics. Google Urchin [4] is an example of a tool that relies on web server log analysis. All pages are treated equally, and no instrumentation is required for each page addition or modiûcation. While a signiûcant advantage of this approach is that it is easier to perform analytics as far back as the logs are available, the challenge is in handling changes in log format over time. Furthermore, with dynamically generated web pages, various key-value pair url parameters must be understood and mined although their churn rate is high. The ideal web analytics approach is to overlay log data with custom javascript beacons to allow both ûne grained instrumentation as well as historical analysis of web access patterns. Additionally, any web analytics sys tem must adapt and account for changes over time and not require re-indexing historical data to account for new schema.

The second consideration for web analytics is how much and what granularity of information is stored. There are three options when considering what to store:

1. All the raw data
2. Samples of the data
3. Analysis of the data

Ideally, one could store and search all the raw data across all time. However, many web analytics products have hard bounds on how much data can be stored at a time, or require a preprocessing step to impose a speciûc schema to the data. The advantage of sampling the data is that you can reduce the quantity of data stored. In fact, Google Analytics imposes data sampling when the traffic volume exceeds a certain thresh old. The major downside of data sampling is that spikes and anomalies are often overlooked and difficult to detect, especially with a coarse sampling granularity. The third option, data summarization, addresses this issue by maintaining summary statistics for various characteristics of the data rather than the raw data itself, and is used by commercial tools such as Omniture. Summarizing average traffic volume on a daily basis provides a signiûcant space saving compared to maintaining all the raw data if the only goal is to calculate traffic volume. That said, although problem detection is easier with summarization than sampling, problem diagnosis is still difficult without the raw unsummarized data. Furthermore, if the summaries are over a coarse enough time granularity, depending on the metrics maintained (e.g. average but not max), spikes may be smoothed out. Preserving the original

data and allowing drilldowns from the summaries as and when necessary can only address this shortcoming.

The third and often most impactful consideration is how easy it is to retrieve the data and search or run reports against it. There is a wide range of possibilities for data retrieval that span from limited predeter mined metrics or reports (e.g. Google Analytics) all the way to needle-in-a-haystack type ûne grained plain text searches. With business needs evolving over time, it is difûcult to anticipate what metrics to track a priori and many times, the need to overlay multiple datasets with web access data greatly inûuences the decision. Since time is often the only axis with which multiple datasets can be interleaved, it is important to preserve temporal patterns in as ûne granularity as possible.

III. WHAT CAN WEB LOGS TELL ME?

Web access logs contain a wealth of information about trafûc served by web servers. There are a variety of web servers, such as Apache [2], IIS [5] and Nginx [6]. Each of these web servers supports a variety of formats for web logs generated. For example, Apache produces both the access combined and access common formats, each with a variety of customizable ûelds. Although the range of possible web log formats is wide, there are several required ûelds captured by all these formats. Two categories of information can be extracted from the logs:

- **Visitor demographics:** Fields such as clientip tell you the visitor's ip address, which can be used to determine geolocation of visitors. Fields such as user agent can indicate the browser or platform used by site visitors.
- **Visitor activity and site usage patterns:** Fields such as uri path show what pages on your site are visited. URL parameters can be used to determine what content, if any, was downloaded. Additional analytics on uri ûelds can show information on how many pages people visit, in what order etc.

  At a cursory level, this information helps the web operations team better provision and manage resources to adapt for load and popularity. However, a few transformations to the data can lead to business-level insights such as marketing strategy, product positioning and revenue channels. There are two transformations I identiûed as crucial to improving business insights:

- **Sessionize the data**: Use clientip, user agent and/or cookie as well as any temporal thresholds necessary to deûne what entails a usage session. Metrics calculated on sessions make it easier to characterize user behavior and therefore expose content based on common actions that lead to a conversion.
- **Coalesce external data sources**: Often, web logs alone do not tell you conclusive information, but

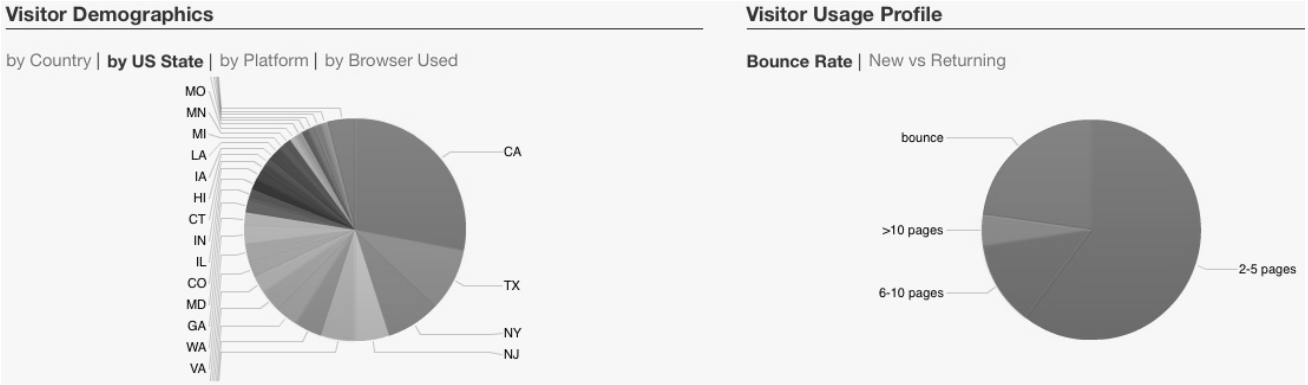

Fig. 1 Example of dashboards built using Org. searches against web log data

when interleaved with another data source, such as a revenue database, can produce more concrete metrics. For instance, web logs can show what search terms users searched for that caused search engines to refer them to your website. While a count of visits by search term produces a popularity metric, it is more meaningful when you multiply this metric with the amount spent on ad words for the corresponding search phrases and keywords. Correlating the popularity data with a marketing campaign cost sheet will allow us to evaluate the return on investment of an ad campaign.

IV. DATA SUMMARIZATION IN ORGANIZATION

Data summarization is crucial to sift through large volume of historical data in a timely fashion. Most approaches to Web Analytics, and Business Intelligence in general, involve acting upon already summarized data.

V. SUMMERY DATA LAYOUT

Since Organization has full control of how summary data is rendered as text, I naturally chose a format that is the easiest and most efficient for the system to process. Although Organization can interpret most common log formats automatically.

VI. SUMMARIZATION CHALLENGES/ TRADE-OFFS

Although the common case of summarizing data with sufficient statistics over a time period is fairly straightforward, there are exceptional cases where the summarization leads to an incomplete or incorrect view of the data. Below, I have four such issues in the context of web analytics.

a. The cardinality curse

b. Border Patrol

c. Caveats of statistics

d. Resurrecting pre-summarized data

VII. SUMMARIZATION EFFECTIVENESS

In this section I provide examples of composing orgenization searches against raw web logs as well as summarized versions of the log data. I compare the performance of searches towards summarized and un-summarized data and explain trade-offs between the approaches.

1. **User session summarization search:** A user session can be deûned using the search

   [get user sessions]

   source = my access combined log status = 200

   | transaction clientip maxpause = 1h

   ‘get user sessions‘ | eval user type = case(eventcount = 1, “bounce”, eventcount <= 5, “2

   “ 5 pages”, “ eventcount <= 10, “6 “ 10 pages”, eventcount > 10, “ > 10 pages”) | stats count by user type

2. **Web traffic by URI and status:** In the next example, I examine the advantage of constraining a summary to cater to a speciûc search rather than an exhaustive set. I populate our hourly summary index for web trafûc as follows:

   eventtype = web “ traf f ic “ external | stats count as “hits”,

   min(time) as earliest hit,max(time) as latest hit by uri, status index = summary hourly | stats sum(hits) as hits,

   min(earliest hit) as earliest hit,

   max(latest hit) as latest hit by uri, status

   index = summary hourly | eval status = toS tring(f loor(status/100)) + “xx” | stats sum(hits) as “totalcount” by uri, status

VIII. CONCLUSION

series	kb
audittrail	3230.477770
scheduler	2472.734764
splunkd	20566.588819
splund_access	81.030276

Table 1: Table rendering of summary data

summary granularity	time (hr:min:sec)
unsummarized	01:38:40
hourly summary	00:08:11
daily summary	00:07:47

Table 2: Comparison of search performance for calculating pages visited per session against unsummarized data, hourly summaries, and daily summaries

summary granularity	time (hr:min:sec)
unsummarized	01:36:46
hourly summary	00:16:00
daily summary	00:15:49

Table 3: Comparison of search performance for calculating top landing pages against unsummarized data, hourly summaries, and daily summaries

In this paper, I presented an approach to web analytics that relies on data summarization. I demonstrated the expressiveness of the Company search language for creating summary indexes and efûciently reporting on web log data. I strongly believe that Company immensely facilitates web-related business and operational insights.

While I demonstrated feasibility and performance of analytics with multi-level summarization, I have yet to assess what user interfaces are required to simplify summarization. It appears desirable to perform summarization with limited user interference, perhaps only to specify summarization granularity. However, I must systematically understand such trade-offs based on the volume of web log data per and the resource requirements for generating the summaries for each environment.

The uses for Company are much broader than web analytics. Web logs are just one example of business-critical time series data. There are many other equally valuable datasets that are commonly mined for insights. Some examples include call detail records and CRM data. Company already allows users to seamlessly index and search these semi-structured and unstructured time series data and temporally overlay heterogeneous datasets.

One signiûcant area of future work includes exploring data summaries as direct input for machine learning algorithms. Currently, the summaries I generate are convenient for human readable reports. As Company moves more towards a data preprocessor for machine learning algorithms, it is equally important to evaluate alternate summary data layout and retrieval mechanisms for machine consumers. I believe this is a promising area of future work that would beneût data mining and analytics.

REFERENCES

[1] Apache Hive Project. http://wiki.apache.org/hadoop/Hive

[2] Apache HTTP Server Project. http://httpd.apache.org

[3] Google Analytics. http://www.google.com/analytics

[4] Google Urchin. http://www.google.com/urchin

[5] IIS. http://www.iis.net

[6] NGiNX. http://wiki.nginx.org

[7] BI T I NCKA L., GANAPATHI A.. , SORKI N S. and ZHANG S. ,"*Optimizing data analysis with a semi-structured time series database. In Proceedings of the 2010 workshop on Managing systems via log analysis and machine learning techniques*", (Vancou- ver, Canada, 2010), SLAML'10, USENIX Association, pp. 7–7.

[8] BODIK P., FRIEDAN G. , BIEWALD L. , LEVINE H. , PATEL , K. , TOLLE G. , HUI J . , FOX O. , JORDAN M. I . , and PATTERSON D., "*Combining visualization and statistical analysis to improve operator conûdence and efûciency for failure detection and localization",* In Proceedings of the 2nd IEEE International Conference on Autonomic Computing (ICAC 05(2005), IEEE Computer Society, pp. 89–100.

[9] HELLERSTEIN J . M., AND STONE BRAKE R M. Readings in Database Systems: Fourth Edition. The MIT Press, 2005