A Survey on Web Data Extraction And Backend Services

Sashikant S Singh[1], Raj Pratap[2], Rajeshwari S [3]
Dept of Information Science
New Horizon College of Engineering, Karnataka, Bangalore, India

Abstract- *Website speed matters in this article, we'll talk about back-end optimization. We'll see ways to speed up your website, showing you which optimizations have the most impact on load times. This means that we'll start by addressing common issues like unoptimized queries that slow down a website's performance and that are easy to identify and fix. We present survey of HTML aware web scrapping techniques and various techniques for a system of distributed workload management to achieve service differentiation and overload protection in such large scale deployment.*

Keywords- Web Scrapping, Back-End Optimization, Html Aware, Workload Management

I. INTRODUCTION

With the rapid expansion of cloud offerings, more applications are hosted in clouds for the benefit of scalability and cost saving. Thus, managing workload with satisfactory quality of service for large scale server deployment becomes a fundamental task to reduce management and operation cost. Large scale server farm usually consists of multiple tiers of servers, e.g., HTTP servers in the front tier, proxy servers in the middle tier, and application servers (usually with database servers) in the back-end tier. There is lots of work in the field of web data extraction. There is number of techniques proposed in literature for web data scraping by number of researcher.

II. LITERATURE REVIEW

In this survey we are grouping tools which are depend on HTML structure and DOM tree structure to extract data HTML-aware. Website performance is tightly linked to the user experience. We all live in an experience economy, where it is only the experience that differentiates an organization to sell and create a recall for their products and services. With end-user patience level dipping constantly every year, it is imperative for businesses to take proactive steps to speed-optimize their website to deliver superior user experience instantaneously and make it work faster on all devices. Although there are various ways and techniques available that can help you speed up your website performance,

A. Memory Overhead:

The memory overhead is the amount of bytes that need to be stored in memory. Those bytes are related to the hashes that point to the disk location, where the actual content is stored. A hash of the URL along with some metadata are stored in memory, and the object itself on the persistent storage. The index entry in Squid is 80B. An optimized indexing methodology [12] would not change the overhead considerably. For the RE system, the chunks are stored on the persistent storage. For every chunk the representing hash value is stored in memory. Let us assume that all chunks are very small, e.g., L = 8B. A hash value of size N = 4B would produce 4.2 billion unique entries for 8B chunks, viz. 24TB of data on the HDD. The 4B hashes are stored in a doubly linked LRU list, for which a 2x3B virtual memory pointers (forward and backward connection pointers) would suffice for an 134GB hash memory space. The chunk size, which is also stored in memory, is 2B; the log generation output is 1B and the disk number 1B (in case a disk array is implemented). The total overhead per chunk is therefore 14B.

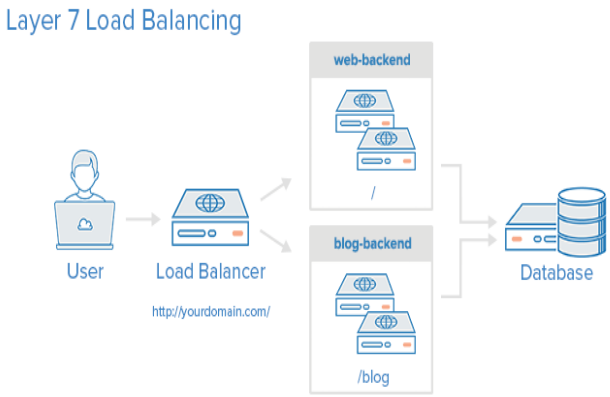

FIG 1.

B. Resource Monitor and Data Collection:

One common practice in resource monitoring is to periodically sample the usage statistics of the bottleneck

resources, e.g., CPU capacity, memory, bandwidth, etc. We adopt two levels of control cycles to obtain reliable resource usage information. There is an internal cycle of finer granularity to collect several samples of resource usage status within one external control cycle. When monitoring the resource usage of server component externally, there may be some delay introduced by data caching inside the monitored components. Collecting multiple samples allows us to filter out outlier and conduct aggregations, thus obtaining more accurate results.

C. Resource Manager and Control Cycle Synchronization:

At each regular control cycle, the rate calculator returns the maximum request admission rate for the local server node.

For Server i, we assume there are J proxies dispatching workload to it, and each of the proxy receives μij out of the maximum rate ri returned by rate calculator on Server i. Thus, PJ j=1 μij = ri. Consequently, for proxy j, if it manages workloads for applications on I servers, it will receive I rate quotas from these servers, such that the total rate it can send back is PI i=1 μij = μj. With the above allocation and aggregation done at the sub-controller and the proxy tier respectively, we have to deal with time synchronization issue among them. Otherwise, the allocation and aggregation
may happen at random moment on different servers, causing inaccuracy in rate calculation and management. An internal NTP service is used to roughly synchronize the clock of sub-controllers and proxy dispatchers. Resource
manager at sub-controller sets its control cycle apart from the control cycle at the proxy tier by a few seconds. Thus the
control cycle of sub-controller can be roughly synchronized to be a few seconds behind the control cycle in the proxy tier. This is to ensure that the resource manager in sub-controller can receive most recent information from the proxies. For each proxy dispatcher, as discussed later, it will adjust the rate whenever it receives the updated quota. This is because the proxies care more about the resource sharing among different service classes instead of the absolute rate being shared. The adaptiveness of our overall framework provides the flexibility that strict clock synchronization among all components is not required.

D. DEPTA:

DEPTA (Data Extraction based on Partial Tree Alignment) [22]: DEPTA founds repeated substring by comparing only adjacent substrings with starting tags having the same parent in the HTML tag tree. In DEPTA single page containing lists of data records is used to extract data. DEPTA consist of following four components Building HTML tag tree, Mining data region, Identifying data records, and Data item extractor. In first step DOM tree is constructed by finding four boundaries of rectangle of each HTML tag. Data region mining step finds the data region by comparing tag strings. Similar nodes are labeled as data region. Generalized node is used to denote each similar node. Neighbor generalized nodes form a data region. Gaps between data records are used to eliminate false node combinations. From data region data records are identified from generalized nodes. Data items are extracted based on partial tree alignment technique. Two steps are performed in data extraction, first is production of one rooted tag tree for each data record; Sub trees of all data record are arranged into a single tree, and second is Partial tree alignment: Tag trees of data records in each data region are aligned using partial alignment. Number of tag trees of multiple data records is needed to be aligned in order to extract data.

E. W4F:

W4F (Wysiwyg Web Wrapper Factory) is a Java toolkit to generate Web wrappers [5]. Wrapper development process in W4F consists of three independent steps: retrieval, extraction and mapping step. In the retrieval phase, document to-be processed is retrieved and given as input to parser that constructs a parse tree. In the extraction phase, extraction rules are applied on the parse tree to extract information. Mapping phase is used to map extracted data to NSL structures.

IV. CONCLUSION

In this paper, we present survey on web scrappimg, back-end optimisation Most of the techniques first clean the fetched web page by removing bad and ill formatted tags. Then these tools construct parsing tree of page, and this tree is used in different way to extract data from that page. The backend, or the server part of your website, stays invisible to the end-user, but it matters a whole lot when it comes to your website’s speed. Some techniques discussed above can be implemented even by non-techy people, though some of them require specialists with a deep technical background.

REFERENCES

[1] Ioannis Papapanagiotou, Robert D. Callaway, and Michael Devetsikiotis,"Chunk and Object Level Deduplication for Web Optimization: A Hybrid Approach", IEEE ICC 2012 - Communication QoS, Reliability and Modeling Symposium, pp 1-9.

[2] Vinayak B. Kadam and Ganesh K. Pakle,"A Survey on HTML Structure Aware and Tree Based Web Data

Scraping Technique",Vinayak B. Kadam et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014,pp 1655-1658.

[3] Shengzhi Zhang, Haishan Wu, Wenjie Wang, Bo Yang, Peng Liu, Athanasios V. Vasilakos,"Distributed Workload and Response Time Management for Web Applications",Network and Service Management (CNSM), 2017 7th International Conference,pp 1-9.

[4] Yixin Diao, Xiaolei Hu, Asser Tantawi, and Haishan Wu. An adaptive feedback controller for sip server memory overload protection. In Proceedings of the 6th international conference on Autonomic computing, pages 23–32, 2009.

[5] A. Finamore, M. Mellia, M. Munafo, R. Torres, and S. Rao, “Youtube everywhere: Impact of device and infrastructure synergies on user experience,” Purdue Research Report, 2011.

[6] Mohammed Kayed and Chia-Hui Chang, “FiVaTech: Page-Level Web Data Extraction from Template Pages,” IEEE transactions on knowledge and data engineering, vol. 22, no. 2 , pp. 249-263, February 2010.