요 약 웹 응용 프로그램의 급격한 증가와 함께 웹 트래픽이 증가하고 있다. 웹에 대한 요청과 그 응답에 대한 기록인 웹 로그 또한 폭발적으로 증가하고 있다. 웹 로그로부터 기록된 정보를 추출하기 위해서는 매우 큰 용량의 데이터를 효과적으로 추출하고 다양한 방식으로 다룰 수 있는 시스템이 필요하다. 본 논문에서는 백본망 로그 기반 대화형 웹 분석 시스템인 CERES를 소개한다. 기존의 웹 분석 시스템들과 달리, CERES는 하나의 웹 서버에 대한 분석이 아닌 백본망에서 생성되는 모든 웹 로그의 분석을 목적으로 한다. CERES는 하둡 분산 파일 시스템 (HDFS)을 저장소로 하는 서버 클러스터에 배포되며, 대용량의 로그에 기반한 분석을 분산 처리를 통해 지원한다. CERES는 백본망에서 생성된 웹 로그 데이터를 관계형 데이터로 변환하고, 사용자는 변환된 관계형 데이터에 대해 SQL을 이용하여 질의를 요청할 수 있다. 내부적으로 CERES는 웹 로그의 통계적 분석에 대해 SQL을 이용하여 처리하기 위해 데이터 큐브를 활용한다. 또한, CERES는 다양한 통계적 분석을 지원하기 위해 대화형 SQL 인터페이스를 포함한 세 가지 형태의 웹 인터페이스를 제공하며 사용자는 이를 통해 쉽게 질의를 요청할 수 있고 그 결과를 시각적으로 확인할 수 있다.

키워드: 웹 로그, 대화형 분석 시스템, 데이터 큐브, 하둡

Abstract The amount of web traffic has increased as a result of the rapid growth of the use of web-based applications. In order to obtain valuable information from web logs, we need to develop systems that can support interactive, flexible, and efficient ways to analyze and handle large amounts of data. In this paper, we present CERES, a log-based, interactive web analytics system for backbone networks. Since CERES focuses on analyzing web log records generated from backbone networks, it is possible to perform a web analysis from the perspective of a network. CERES is designed for deployment in a server cluster using the Hadoop Distributed File System (HDFS) as the underlying storage. We transform and store web log records from backbone networks into relations and then
allow users to use a SQL-like language to analyze web log records in a flexible and interactive manner. In particular, we use the data cube technique to enable the efficient statistical analysis of web log. The system provides users a web-based, multi-modal user interface.

**Keywords:** web log, interactive analysis system, data cube, Hadoop

1. Introduction

With the explosive increase of web applications, web traffics also have increased rapidly over the past few decades. Web service providers gain values by analyzing web log records in various perspectives to improve their services. The key requirements of web log analysis are scalability and efficiency due to the huge volume of web log.

Discovering valuable information from web log generally entails the exploratory analysis in a consecutive way. Thus, it is important for an analysis system to allow users perform interactive and flexible analysis rather than to provide fixed analysis functionalities.

In this paper, we present CERES, a log-based, interactive web analytics system for backbone networks. In CERES, web log records are Collected, ETL-ed into Relations, and Explored via a SQL-like language. More specifically, web log records collected from backbone networks are transformed into relations, and then stored in the Hadoop Distributed File System (HDFS) [1]. The stored data are summarized by using data cube techniques [2]. Users can interact with CERES with a web-based, multi-modal interface which provides three analysis modes including interactive, explorative analysis via a SQL-like language. User queries are processed in parallel by a distributed query engine.

We point out the major differences between the existing systems [3-5] and CERES. First, to the best of our knowledge, most existing web log analysis systems are focused on the web log analysis for a single web server. So, they can cause the scalability problem when applied to the backbone-scale web log analysis. Second, they only provide the analytic results of the requests to a single web server, whereas CERES can provide the analytics results from every web request/response passing through the networks. Third, the existing systems provide only a set of fixed functionalities, thereby limiting interactive and exploratory analysis. Lastly, unlike the existing systems, CERES can respond to the user queries that incorporate a large time range since it is designed to support the analysis on long-term collected data.

The features of CERES are summarized as follows:
- CERES executes user queries in parallel over a Hadoop cluster using a distributed query engine.
- CERES supports a SQL-like language for interactive analysis.
- CERES provides web-based interface that users can submit queries in an easy and convenient way.
- CERES utilizes materialized data cube for efficient query processing.
- CERES effectively maintains disk storage via data cube aging techniques.

The rest of the paper is organized as follows: Section 2 describes a brief overview of the web log analysis. In Section 3, CERES is presented along with its features. Section 4 presents the interfaces of our system and section 5 presents the performance of our system. Our conclusion is presented in Section 6.

2. Web Log Analysis

2.1 Web Log Data Format

Web log records are information about HTTP requests and responses. We gather web log records from a web caching/proxy system [6], called Jaguar™ [7] that is attached to a high-speed backbone network.
Fig. 1 shows an example of web log records. Each line is written in the native squid format [8] followed by five additional fields. The additional fields are:

- **User agent header**: the user agent header of an HTTP request
- **Server IP address**: the IP address of a destined web server
- **HTTP Referer**: the web page that directs to the resource being requested
- **Smartfilter category code/name**: the name and the corresponding code of categories to which the web server belongs

### 2.2 Web Log Analysis Types

In web log analysis, there are two representative patterns of analysis.

**Interactive, Exploratory (IE) Analysis**: IE analysis is used to find values by exploring data in an interactive manner. To analyze data in various perspectives under various conditions, queries of this type generally require a series of consecutive steps. Usually, the results of a query can be used as an input of other queries. For example, analysts can submit a query “Find out the time when the traffic was highest yesterday” followed by “Find out the most visited streaming site at that time.”

**Periodic, Monitoring (PM) Analysis**: When analysts find a useful result from an IE query, they might want the query to be executed repeatedly to monitor changes of web traffic patterns. For example, to monitor the daily traffic, analysts can execute the following query everyday: “Find out the amount of total traffic generated today.”

CERES supports both of the above two analysis patterns efficiently. For IE analysis, we provide a SQL-like, powerful analysis language. For PM analysis, we provide a tool for registering/executing analytical queries with controlled time specifications. In addition, CERES supports various, graphically enhanced chart interfaces which improve the user friendliness and analysis quality of the system.

### 3. The CERES System

CERES receives web log record sets from **Jaguar** periodically (i.e., every 30 minutes) and transforms them into relations. During the transformation, incorrect or corrupted data that cannot satisfy the integrity constraints are removed. Then, the relational web log data are stored in HDFS. As data are stored, CERES registers them as **base tables** partitioned by specific time interval.

When a new base table is registered, the system computes a data cube from that base table. To overcome the storage shortage problem, we periodically compress the data cubes by ‘aging’ daily data cubes into weekly data cubes and weekly data cubes into monthly data cubes. Old base tables are dropped according to the system settings.

CERES allows users to submit queries via a web based multi-modal user interface. Users can submit queries by using a graphical interface as well as the command-line, shell interface using a SQL-like language. Also, the interface supports registering PM queries. The registered PM queries are executed automatically and periodically. CERES analyzes submitted queries and tries to utilize the materialized data cubes to gain performance benefit.

The overall architecture of CERES is illustrated in Fig. 2. The system consists of four components: **ETL, Cube Management, Query Processing and Multi-Modal User Interface**. CERES uses HDFS as the underlying storage. We use Apache Tajo [9] for a distributed query engine. In the below, we present each component in detail along with its features.

---

**ETL** – The ETL (Extract-Transform-Load) component periodically receives a set of web log record as a compressed file from **Jaguar**. It decompresses and parses the web log records. If the ETL component finds any corrupted data during parsing, it removes them to prevent the system from returning inaccu-
rate analysis results. Then, it transforms them into relations by converting the value in each field to the corresponding attribute’s data type. The schema of the relational web log used in CERES is shown in Table I. After storing the transformed data in HDFS, the ETL component registers them as base tables partitioned by a specific time interval. The time interval is set as ‘one day’ by default.

**Cube Management** - The Cube Management component materializes a partial data cube from a base table with pre-defined combinations of attributes. We refer all the attributes used for grouping in cube generation as base cube attributes. The partial data cube is generated by applying aggregate functions (i.e., MIN, MAX, SUM and COUNT) to ‘elapsed_time’ and ‘transferred_bytes’. Note that ‘elapsed_time’ attribute and ‘transferred_bytes’ attribute are numerical attributes that are referred to as measure attributes. They can provide further information when they are aggregated. We call the attributes added by applying aggregate functions as aggregated measure attributes.

Fig. 3 shows an example of a data cube table. The ‘sum_et’ attribute and the ‘sum_tb’ attribute contain values aggregated by applying the SUM function to attributes ‘elapsed_time’ and ‘transferred_bytes’ respectively. The ‘cnt’ attribute contains a value aggregated by COUNT function.

Since computing full data cube requires a lot of processing time and some attribute combinations (i.e., cuboids) are not necessary in web log analysis, we use only a subset of attribute combinations.

Because of the continuously collected web log from a backbone network, the base tables and materialized data cubes can overwhelm the HDFS space. The Cube Management component incorporates two processes to overcome the HDFS space shortage problem:

- **Base table removal**: The component removes the old base tables from the system.
- **Data cube aging**: The component performs merging fine-granule data cubes into coarse-granule ones. For example, seven ‘daily’ data cubes are merged into a single ‘weekly’ data cube. Likewise, weekly cubes are compressed into monthly ones. Note that the size of the aged data cube is always smaller than that of input data cubes. This is because the tuples having the same values in base cube attributes are summarized into a single tuple.

These two processes are triggered 1) when a base table or a data cube is older than a user-specified time value (i.e., age) and 2) when the remaining HDFS space is less than a user-specified threshold. Although old base tables are removed, their aggregates are maintained in the corresponding data cubes. Therefore, the system can still answer the users’ analytical queries (in a limited extent) in spite of the absence of base tables.

**Query Processing** - The Query Processing component receives queries from users and executes them on a Hadoop cluster. In our implementation, we use Apache Tajo as a distributed query engine, which is one of the most popularly used SQL-on-Hadoop systems.

When executing a query, it analyzes the query and checks whether some results or intermediate results that are already materialized into data cubes can be exploited for the query processing. For example, the query “Find out the average daily traffic of February 2015” can be answered from the four weekly data cubes of February 2015, not from the base tables.

The results of IE queries are stored in HDFS and registered as tables for a certain period so that they can be used in subsequent queries. The results of PM queries are also stored and registered as tables to be provided to users immediately. The result table

---

**Table 1 Web Log Schema**

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Data Type</th>
<th>Attribute Name</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>date_time</td>
<td>Datetime</td>
<td>direct</td>
<td>String</td>
</tr>
<tr>
<td>elapsed_time</td>
<td>Long</td>
<td>host_name</td>
<td>String</td>
</tr>
<tr>
<td>transferred_bytes</td>
<td>Long</td>
<td>content_type</td>
<td>String</td>
</tr>
<tr>
<td>cache_result</td>
<td>String</td>
<td>smart_filter</td>
<td>String</td>
</tr>
<tr>
<td>http_code</td>
<td>String</td>
<td>server_ip</td>
<td>IPv4</td>
</tr>
<tr>
<td>client_ip</td>
<td>IPv4</td>
<td>referrer</td>
<td>String</td>
</tr>
<tr>
<td>method</td>
<td>String</td>
<td>device</td>
<td>String</td>
</tr>
<tr>
<td>request_addr</td>
<td>String</td>
<td>browser</td>
<td>String</td>
</tr>
</tbody>
</table>

---

Fig. 3 An example of the data cube table

<table>
<thead>
<tr>
<th>host name</th>
<th>browser</th>
<th>sum_et</th>
<th>sum_tb</th>
<th>cnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>google.com</td>
<td>Chrome</td>
<td>5204803</td>
<td>8113363246</td>
<td>57902</td>
</tr>
<tr>
<td>google.com</td>
<td>Mozilla</td>
<td>3675425</td>
<td>227764546</td>
<td>2309</td>
</tr>
</tbody>
</table>

...
of a PM query is retained until the next query execution.

CERES supports User-Defined Functions (UDFs) to extend the ability of web log analysis. For example, currently we use a UDF named GeoIP that converts an IP address into a country code [10], and a set of time conversion UDFs that transform the granularity of the given time values.

**Multi-Modal User Interface** - The Multi-Modal User Interface component allows the users to interact with the system. Users can submit queries and the query results are displayed in various types of chart.

It consists of three modes, **IE–Graphical Mode**, **IE–Shell Mode** and **PM Mode**. **IE–Graphical Mode** allows users to submit queries using a graphical interface. By assigning attributes and functions to x-axis and y-axis, users can easily construct queries. **IE–Shell Mode** provides an interface that users can submit queries using a SQL-like language. **PM Mode** supports the functionality of registering PM queries that are executed periodically and displaying the execution results.

### 4. Interface Specification

Users can access CERES via web browsers without any additional program installation. The user interface consists of three modes, **IE–Graphical Mode**, **IE–Shell Mode** and **PM Mode**. In common with all three modes, query results are shown as a table with rows and columns. The results can also be represented in a line chart, a bar chart (see Fig. 4), a pie chart and a map chart with axes attributes defined by users. Specifically, a map chart shows the results of a query on a world map with different colored countries. If a country is painted darker, it means that it has a larger value than the countries that are painted lighter. Fig. 5 shows an example of map chart representation. Users can switch the type of the chart easily. We describe the further features of each mode below.

**IE–Graphical Mode** - **IE–Graphical Mode** allows users to construct queries with a few clicks via a graphical interface. Users can decide which attributes are going to be assigned to which axis by clicking attributes listed in the panel. In x-axis, the grouping attributes can be assigned. In y-axis, the aggregate functions can be assigned with the measure attributes as an input.

Users can set the range of input data for analysis with the date picker. The representation type of the results can also be set by clicking the chart icons in the panel. An example of constructing a query in **IE–Graphical Mode** is shown Fig. 6.

**IE–Shell Mode** - In **IE–Shell Mode**, users can submit web analytics queries using a SQL-like language. Fig. 7 shows an example of using our SQL-like query language via the **IE–Shell Mode**.
PM Mode - In PM Mode, users can manipulate (i.e., create, delete, and update) PM queries, specifying query descriptions, execution periods and chart types for the result display. Fig. 8 shows the registering interface.

5. Experiments

In this section, we present the performance of our system. We used TPC-H benchmark [11] data set generated with scale factor 1000. We selected the queries that are best suitable in our analysis scenario. The selected queries from TPC-H benchmark are 1, 2, 3, 6, 10, 12, and 14.

For experiments, we used an in-house cluster that consisted of one master and 32 workers. Each worker is equipped with a 16GB memory, 4TB disk, Intel i5 quad core CPU; the workers are connected to each other via 1 Gbps LAN. We used 2.2.0 version of Hadoop, and 0.8.0 version of Tajo.

In Fig. 9, we show the response time for each query. The response times of every query are shown below an hour. Since the query 1 and the query 6 exploit the cube table, the results are shown below few seconds.

6. Conclusion

This paper presents CERES and shows how the system is structured. The characteristics of web log attributes are well reflected in CERES and many consideration for web log analysis lets CERES to be efficient in both execution time and disk usage. CERES can also execute repetitive queries by just registering it in PM Mode. CERES provides an interface that users can execute the query graphically. Furthermore, with CERES, queries that have consecutive steps could be processed interactively.

References


Ilhyun Suh received his B.S. degree in Computer Science from Korea University, Seoul, Korea, in 2014, and he is currently the M.S. course of IT Convergence at Korea University. His research interests include distributed/parallel processing of large-scale data and analytical processing.

Yon Dohn Chung received his B.S. degree in Computer Science from Korea University, Seoul, Korea, in 1994. He received his M.S. and Ph.D. degrees in Computer Science from KAIST, Daejeon, Korea, in 1996 and 2000, respectively. He was an Assistant Professor in the Department of Computer Engineering at Dongguk University, Seoul, Korea, from 2003 to 2006. He joined the faculty of the Department of Computer Science and Engineering at Korea University, Seoul, Korea, in 2006, where currently he is a Professor. His research interests include distributed/parallel processing of large-scale data, spatial databases and data privacy.