

# Ten Years of SkyServer I: Tracking Web and SQL e-Science Usage

**M. Jordan Raddick, Ani R. Thakar, and Alexander S. Szalay** | Johns Hopkins University  
**Rafael D.C. Santos** | National Institute of Space Research (INPE), Brazil

SkyServer is the primary catalog data portal of the Sloan Digital Sky Survey that makes multiple terabytes of astronomy data available to the world. Here, the process is described of collecting and analyzing the complete record of more than 10 years of Web hits and SQL queries to SkyServer.

The age of Big Data is rapidly changing nearly every aspect of doing science. Access to terabyte- and petabyte-scale online datasets is now routine in fields ranging from astronomy to genomics. Although the radical, new e-science methods enabled by these large datasets have been extensively outlined elsewhere,<sup>1</sup> there has been little research into exactly how the research community and the public are using such methods.<sup>2</sup>

What datasets and query and analysis tools are researchers using? What tools do they use to query, analyze, and visualize those data? How do their e-science methods change as they develop new skills? Understanding how e-science resources are actually being used is critical to maximizing scientific productivity and educational impact.

One approach to answering these questions is to look at case studies of successful e-science projects. This article describes one such case: the online catalog archive of the Sloan Digital Sky Survey (SDSS), an astronomy project that has worked since 1998 to construct the most detailed digital map of the universe in history.

The primary online portal to the SDSS catalog data is the SkyServer, which has been online since June 2001. While the term SkyServer is sometimes used somewhat narrowly to refer to the SkyServer website (<http://cas.sdss.org> or <http://skyserver.sdss.org>), SkyServer is often used as shorthand for the SDSS Catalog Archive Server (CAS) system<sup>3</sup> and all the associated data services that serve the SDSS catalog data online. We use SkyServer in the larger sense here.

For more than 10 years, we've been recording all Web hits to several of the SkyServer's data access interfaces, as well as all SQL queries submitted by users to those sites. The data in our logs offer a fascinating picture of how people are using the resources provided by the SDSS data archive.

The SkyServer log dataset described in this article—containing the complete, 10-year record of each and every Web hit and SQL query submitted to the public online portal of the multi-terabyte catalog archive that ushered in the e-science era in astronomy—represents a unique and invaluable resource to the scientific community and others seeking to understand how this new paradigm is changing the way scientific research is conducted. In particular, the SQL logs are a unique and powerful dataset: we aren't aware of any other e-science projects that have logged their SQL queries in this way. From these data, it's possible, for example, to determine the size and breadth of the user community, the speed and manner with which users adopt new tools in which to do science, the efficacy and usability of various tools available to users, and how members of the public and educational community benefit from this online data set. In particular, this is an unparalleled source of information for understanding the sociology of the new e-science paradigm.

Furthermore, the scrubbed and normalized log database that we've built for this analysis is publicly available for download or for querying using one of our online tools to anyone who wishes to conduct their own analysis.

This article is the first of two parts. This work builds upon the analysis previously done

on the SDSS log data for a five-year period ending in 2006.<sup>4</sup>

We strongly encourage other e-science data providers to save their weblogs for future analysis. Frequently, system administrators just rotate the logs, recording new hits in a new file, and eventually delete the old logs to save space—destroying what could be a treasure trove of information about how users engage with their data. Our hope is that our work will inspire other projects to save their weblog data for future analysis, helping the entire field of e-science make a stronger impact on scientific research and science education.

### Web Analytics and Log Preprocessing

Our research questions focus on user behavior on websites, and our data are the retrospective records of user activity. The ideal methods for answering these questions are those used in the field of Web analytics. This section introduces some concepts used in the academic study of Web analytics. For a review of other Web analytics research using SDSS weblogs as a data source, see the “Prior Work” sidebar.

Web analytics is commonly understood as the collection and analysis of data related to how users interact with websites, with the goal of understanding how users behave on those sites and optimizing the sites to best meet their needs.<sup>5</sup> Most Web analytics applications are used for market research, using data from e-commerce sites to improve sales. Other applications include personalization of content delivery to users, improvement of site performance, and enhancement of site navigation and usability.<sup>5</sup> Within the academic community, research questions investigated through Web analytics include creating behavioral models of users, developing adaptive websites that change based on user interactions, and other applications.<sup>6</sup>

The most common data sources used for Web analytics research consist activity logs collected from web servers, usually as text files. A seemingly-endless variety of tools are available for analyzing weblogs.<sup>6</sup> Commercial applications include Accrue, NetTracker, Elytics Analysis Suite, E.piphany E.6, SPSS Predictive Web Analytics, WebSide Story Hit-Box, IBM SurfAid Analytics suite, WebTrends Log Analyzer (series), Quest Funnel Web Analyzer, and SAS WebHound. A number of free Web analytics packages are also available, such as Webalizer ([www.webalizer.org](http://www.webalizer.org)), Google Analytics ([www.google.com/analytics](http://www.google.com/analytics)), and AWStats ([awstats.sourceforge.net](http://awstats.sourceforge.net)).

But regardless of what tools are chosen to visualize Web analytics data, the underlying raw log

## Prior Work

A few Web analytics studies have already been done using Sloan Digital Sky Survey (SDSS) Web and SQL logs as a data source.

The direct antecedent of our work was a Microsoft Technical Report analyzing SkyServer weblogs from 2001 to 2006;<sup>1</sup> the main article text expands that work using weblog data collected through 2011. Another phase of our current work looked at the evolution of security threats to SkyServer between 2001 and 2011.<sup>2</sup> Our current work represents a major advance over this previously published result for three reasons. First, we have further optimized our dataset (which is twice as large as that used in the prior work) to make queries even more efficient. Second, as described in the main article, we conduct additional analyses of the Web domains from which our Web traffic comes. Third, and most importantly, we make our dataset available to other researchers through the CasJobs interface,<sup>3</sup> using the methods of e-science.

A number of other studies have also used the logs of the SQL queries submitted to the SDSS sites to gain insight into patterns of e-science usage. One study, using queries submitted by users who had created log-ins in CasJobs, studied SQL provenance and user workflow patterns.<sup>4</sup> The study found that although most users don't create workflows, most queries are run by a small percentage of power users. Other studies using the Web and SQL logs have involved building visualization tools,<sup>5</sup> predictive analysis of user paths and sessions,<sup>6</sup> and term frequency in SQL queries.<sup>7</sup>

### References

1. V. Singh et al., *SkyServer Traffic Report—The First Five Years*, tech. report MSR-TR-2006-190, Microsoft Research; <http://research.microsoft.com/apps/pubs/default.aspx?id=64520>.
2. R.D. Santos et al., “Analysis of Web-Related Threats in Ten Years of Logs from a Scientific Portal,” *Proc. SPIE, Defense, Security, and Sensing*, vol. 8408, 2012, p. 84080H.
3. A.R. Thakar and N. Li, “CasJobs and MyDB—A Batch Query Workbench,” *Computing in Science & Eng.*, special issue on SDSS science archive, vol. 10, no. 1, 2008, pp. 18–29.
4. N. Li, *Scalable Database Query Processing*, PhD dissertation, Dept. of Computer Science, Johns Hopkins Univ., 2012.
5. J. Zhang et al., “SDSS Log Viewer: Visual Exploratory Analysis of Large-Volume SQL Log Data,” *Proc. SPIE*, vol. 8294, 2012, article no. 82940D.
6. B. Bhattarai, M. Wong, and R. Singh, “Discovering User Information Goals with Semantic Website Media Modeling,” *Advances in Multimedia Modeling*, Springer, 2006, pp. 364–375.
7. T. Malik, R. Burns, and A. Chaudhary, “Bypass Caching: Making Scientific Databases Good Network Citizens,” *Proc. 21st Int'l Conf. Data Eng.*, 2005, pp. 364–375; doi:10.1117/12.907097.

data are usually quite similar. Each row in a weblog records information about a single hit—that is, a request to the server for a single file by a single client. The data recorded in the weblog depend on the

webserver configuration, but weblog entries typically include information such as the

- IP address of the computer that generated the hit (the client);
- date and time when the hit occurred (the time stamp);
- file that was requested, usually expressed as the relative path from the website root;
- HTTP method used to get the file (defined by the W3C and listed at [www.w3.org/Protocols/rfc2616/rfc2616-sec9.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html));
- result of the hit (success, error, and so on); and
- number of bytes of data sent back to the client.

Our logs save all this information, plus the record of SQL queries submitted.

#### Identifying User Actions: Page Views

A single user action might result in one hit or many. For example, if a user visits a page with several images, their first visit to that page will generate a hit for that page, and additional hits for each image; but on that user's later visits, the images might be cached so the visit will generate only one hit.

Given this uncertainty, a key challenge of Web analytics is determining which hits correspond to volitional user actions, and which hits are ancillary to such actions. In Web usage mining, such user actions are usually described in terms of page views—user requests for pages. Page views are usually not flagged in the logs, so they must be identified post hoc through hit properties, such as the page type. The specific definition of page views is different for different studies.

#### Recognizing Sequential Actions: IPs and Sessions

If a website had only a single user, then the time-ordered sequence of page views in the weblog would provide an orderly complete history of how that user interacted with the site. In reality, though, many users might be accessing the site at once, with their hits (and page views) overlapping. Thus, reconstructing a sequence of user interactions with a website from site logs requires a way to identify which users are which.

Some sites solve this problem by requiring users to log in, or by using server-side analytics scripts embedded in pages, but users might resist. For studies of traditional websites browsed by anonymous users, the most common way to identify users is by their client IP address. Once a user has

been identified by their IP address, a session can be defined as the ordered sequence of hits from that IP before some defined period of inactivity, often 30 minutes. If additional hits from that IP are seen in the weblog after that gap, a new session is declared. As described later in this article, this is in fact the definition of sessions that we adopt for our analysis. It was also the definition used in the five-year study.<sup>4</sup>

#### Context: SDSS Data

Now that we've covered some basics, here we describe the details of the e-science data project that we examine in this case study: the SDSS SkyServer.

#### SDSS Data

The SDSS and its science archive were described in a special issue of *Computing in Science and Engineering* in 2008.<sup>3,7-9</sup> More information about the survey is available from its website ([www.sdss.org](http://www.sdss.org)).

SDSS data begin with observations at the Sloan Foundation 2.5-meter telescope in New Mexico. Its raw images and spectra are stored in a domain-standard file format called Flexible Image Transport System (FITS).

Next, software pipelines process the FITS files, calibrating and extracting measurements for hundreds of millions of stars and galaxies. The pipelines also estimate physical parameters for those stars and galaxies. These measurements and estimates are expressed as numerical values, referred to as science parameters. Examples of science parameters include magnitudes (brightness) and redshifts (distances) for galaxies.

SDSS data are released in discrete batches called data releases, approximately once a year. Each data release contains both raw images/spectra and science parameters. Data releases are cumulative, so each release includes the area covered by all prior data releases; however, because data were sometimes reprocessed, the same object might have different values for its science parameters in different releases.

Table 1 shows the history of SDSS data releases, demonstrating the growth of available data in terms of data size, sky coverage area, and number of sky objects measured. The sky coverage area is given in square degrees—for comparison, the area of the full moon is about 0.2 square degrees, and the area covered by the constellation Orion is about 600 square degrees.

Within each data release, several databases of science parameters were available. The best version

**Table 1. Data sizes of Sloan Digital Sky Survey (SDSS) data releases.**

Data release	Date	Catalog data size (Tbyte)	Sky coverage (square degrees)	Number of unique object images	Number of unique object spectra
DR7	10/31/2008	4.2	14,555	357,175,411	1,053,144
DR6	6/28/2007	2.5	9,583	287,417,920	884,054
DR5	6/29/2006	2.2	8,000	218,218,198	738,449
DR4	6/29/2005	1.8	6,670	180,180,971	608,801
DR3	9/27/2004	1.3	5,282	142,705,734	478,243
DR2	4/15/2004	0.8	3,324	88,281,651	330,215
DR1	6/15/2003	0.5	2,099	52,530,681	163,901
Early data release	6/6/2001	0.2	462	13,791,763	52,896

of the data was the latest, greatest reprocessing through the SDSS pipelines that was officially sanctioned by the collaboration as the best reduction of the data with which to do science. The best version of each data release is referred to as BestDR $n$ , where  $n$  was the number of the release, for example, BestDR2, BestDR5, and BestDR7. Most data releases also included additional databases, but BestDR $n$  was the easiest to use, and we expected it to be the most popular.

As a matter of principle, we never take data releases offline, even when new ones are published. The principle of scientific reproducibility requires that it should be possible for any study to be repeated by any outside researcher at any point in the future; this can be possible only if the data used in the original study are still available in exactly the same form. Thus, in our research, we want to assess usage of these prior releases as well as the final one, data release 7.

### SDSS Science Parameters: CAS and SkyServer

As noted previously, each data release includes two types of data: raw images/spectra and science parameters. SDSS science parameters are stored in the CAS database system. The CAS refers to the database system, independent of what interfaces users use to access those data.

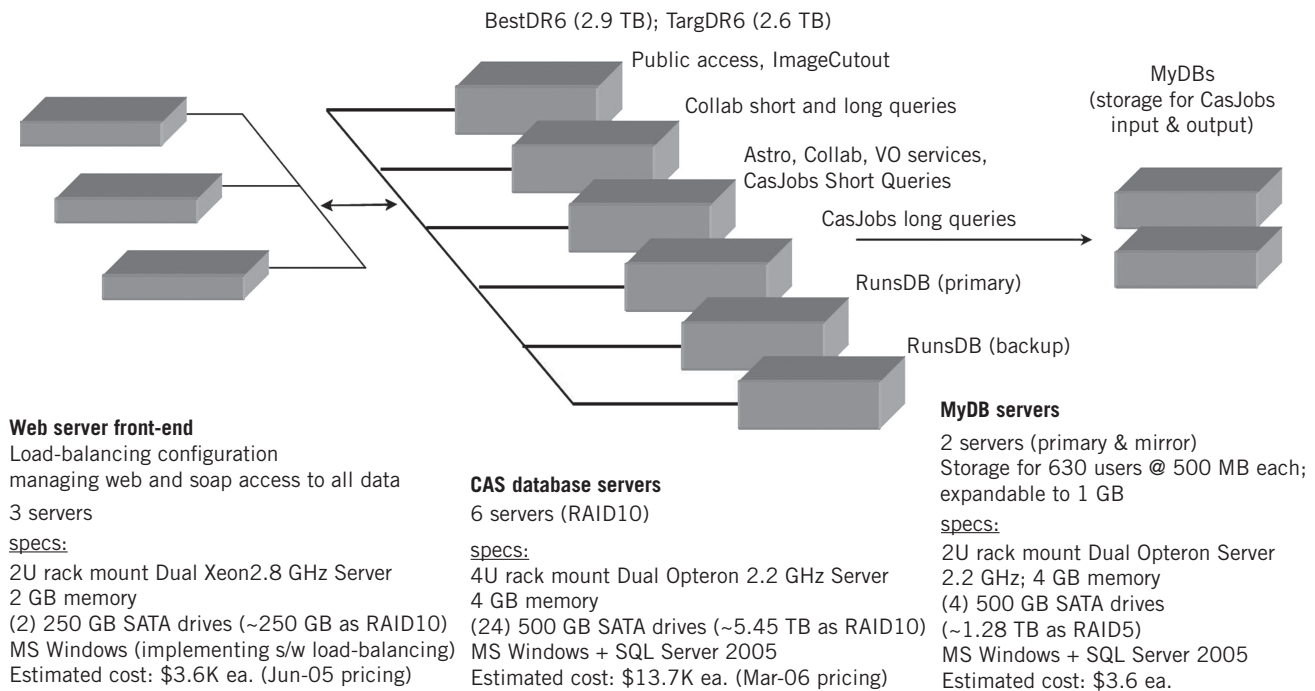
The CAS is a collection of SQL Server databases, each storing a particular release of SDSS data.<sup>8</sup> The CAS is hosted at Fermi National Accelerator Laboratory (Fermilab), with a mirror at Johns Hopkins University (JHU). Figure 1 shows the typical CAS hardware configuration.

Users connect to the SDSS catalog data in the CAS through a variety of interfaces, which are organized into several websites tailored for different audiences. It should be emphasized, however, that all interfaces provide access to exactly the same data.

All the following websites are hosted at either Fermilab or in the Physics and Astronomy Department at JHU:

- *SkyServer*. This is a public website offering browser-based access to the SDSS data, with visual interfaces, SQL tutorials, and lesson plans for learning and teaching science (see <http://cas.sdss.org>).
- *Astronomer portal to SkyServer*. We offer a separate version of SkyServer for professional astronomers with additional data access tools, hosted on separate servers with more generous query limits.
- *Collaboration portal to SkyServer*. Similarly, we offer a password-protected version of SkyServer for members of the SDSS collaboration, with even more generous query limits.
- *CasJobs*. This site is an asynchronous job interface to the CAS,<sup>8</sup> in which users log-in and are given their own personal database (MyDB), into which they can select SDSS data, upload their own data, and cross-match with a variety of datasets (batch jobs for the CAS; see <http://casjobs.sdss.org/casjobs>).

CasJobs has become the default interface to SDSS data for the research community, while SkyServer has become the default interface to SDSS



**Figure 1.** The SkyServer hardware configuration at Fermi National Accelerator Laboratory as deployed in late 2006, hosting data release 6 (DR6). Users interact with the Web servers (left side of diagram), which communicate with the database servers that host the data (center). User-generated data subsets and analyses are stored on individual database “MyDB” servers accessed through CasJobs (right). The figure gives the detailed configurations for each set of servers. The layout depicted also shows how multiple data releases are hosted by our sever farm—older releases are supported indefinitely for continuity and reproducibility of references in published papers.

data for the larger world. This article describes the weblog and SQL log data for the CAS.

**Data: Logs of Web Hits and SQL Queries**

As part of standard hosting practice at both Fermilab and JHU, all Web hits to all sites have been continuously logged since the sites were launched. All the sites listed in the previous section an opt-out privacy policy (available at <http://skyserver.sdss.org/log/en/traffic/privacy.asp>), but so far, no one has opted out. We have also logged all SQL queries submitted to the CAS since early 2002. Our logs of SQL queries are a unique resource—we’re not aware of any other big data providers that have recorded all queries submitted to their databases.

**Harvesting the Weblogs**

A collector running at JHU harvests both types of logs from both institutions every hour on the hour from across the Internet. The harvested logs are stored in a publicly accessible Microsoft SQL Server database, along with a summary of daily, monthly, and yearly activity. For each hit, we record the time of the hit, IP address of the client, agent the client used (browser, script, and

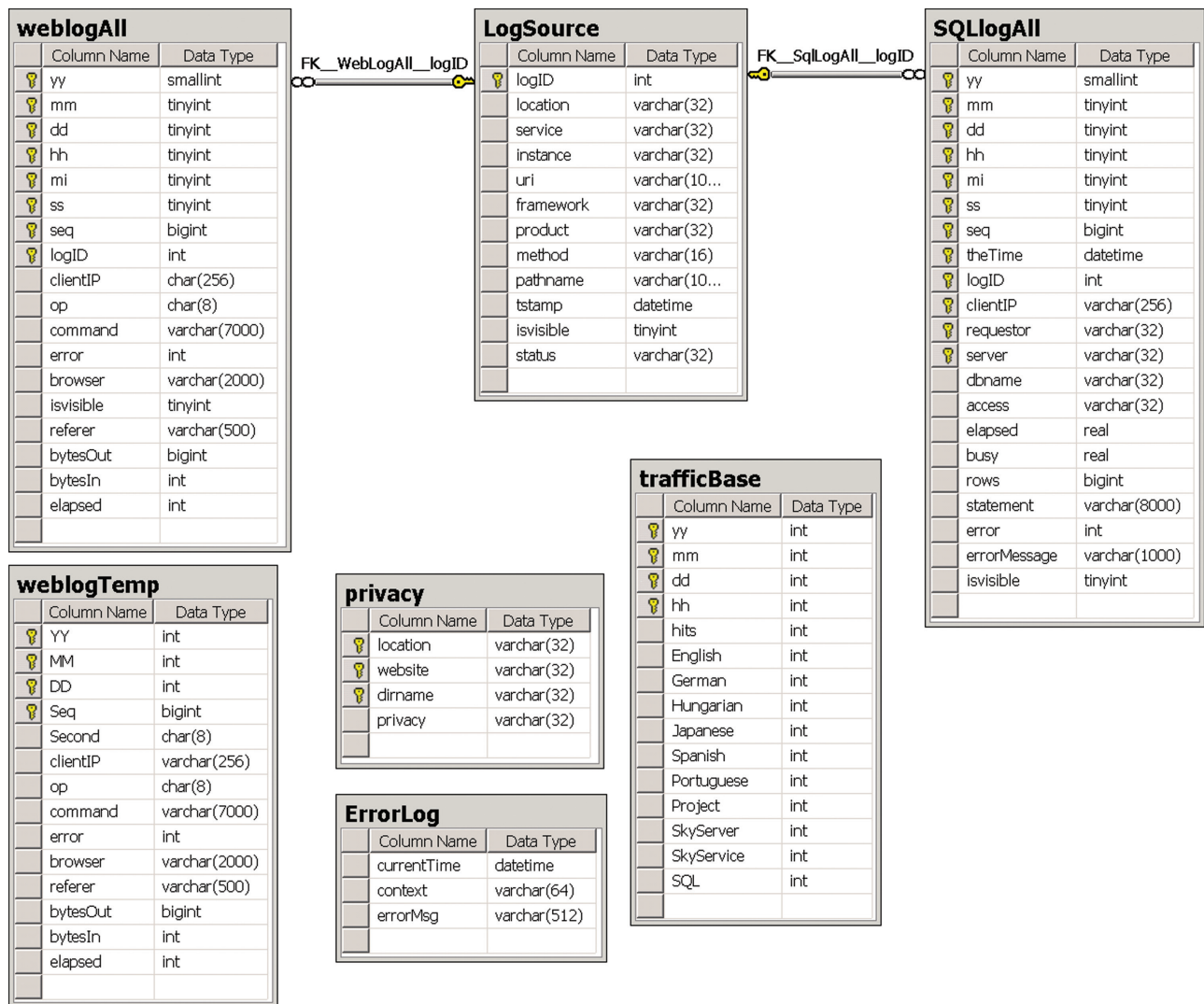
so on), operation and command that was executed, and result.

Figure 2 shows the schema (data model) of the database that holds both the Web and SQL logs. The weblogs monitor 97 servers, and have thus far recorded more than one billion Web hits; as of September 2013, this raw weblog database takes up 900 Gbytes of space. The SQL query logs add an additional 200 Gbytes. Even a simple query to count the rows of the raw weblog data takes 15 minutes; thus, the raw logs are far too large to be thoroughly mined and analyzed in a reasonable time. For this study, we normalized the logs to save space and speed up analysis, using the steps described later.

**Cleaning and Normalizing the Weblogs**

For this article, we chose to study only those hits that occurred during the period in which the data from the original SDSS project were publicly available, but data from its successor project SDSS-III weren’t yet available. Thus, our dataset spans the time period from June 2001 to January 2011.

Within this time period, there are mistakes and anomalies in the log data that need to be corrected



**Figure 2.** The schema of the database that hosts the harvested raw weblogs of all the SDSS websites, which are collected from multiple local and remote sources by our logging software. Records of Web hits are stored in the WeblogAll table; records of submitted SQL queries are stored in the SQLlogAll table. Each of those tables has a foreign key joining to the LogSource table, which contains information about each source of usage data that we log.

or flagged. Each log has time gaps, although they're small and infrequent. Some logs have records with incorrect or missing values due to bugs in our configuration or logging software. We explored several anomalies in the logging data, but in the end, we only flagged out one item from our analysis: a misconfigured user download script that generated more than 7 million hits from a single IP in less than a day, most of which returned errors.

**Ancillary data structures.** The recording of hits and queries in the Web and SQL logs results in much duplication that can be removed when the database is

normalized for analysis. We selected distinct values of IP address, agent, operation, and command into separate tables, and then replaced each occurrence in the main weblog table with a unique identifier that served as a foreign key to the tables of distinct values.

Using this process of substitution, we created the following ancillary data structures:

- *HTTP methods*—all distinct values of the HTTP method (for example, GET, POST, and so on) recorded by the weblogs, including both those specified by the W3C (see [www.w3.org/Protocols/rfc2616/rfc2616-sec9.html](http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html))

and other values that were most likely entered as security attacks;<sup>10</sup>

- *IP addresses*—all distinct client IP addresses from which hits originated;
- *agent strings*—all distinct values of the user agent string supplied by the client;
- *command stems*—starting with all distinct commands that had been run on the webserver (for example, a request for a given page with a series of parameters), we parsed out the command stem, everything before the “?” symbol;
- *command parameters*—starting with all distinct commands that had been run on the webserver (for example, a request for a given page with a series of parameters), we parsed out the command parameters, everything after the “?” symbol;
- *error messages*—for the SQL logs, all distinct errors that the servers returned; and
- *SQL statements*—for the SQL logs, all distinct SQL commands that the servers executed, whether or not those queries returned results.

With all these ancillary data structures, we can quickly sum the number of hits to SkyServer from a given domain or browser, and/or to a given page. This means that when research questions require only overall, time-independent counts, even complex queries can run in seconds.

This normalization process shrank the space required from 1.1 Tbytes to less than 120 Gbytes. The same count query that took more than 15 minutes on the raw weblogs takes only a little more than a minute on the normalized weblogs.

**User agents and domains.** Client IP addresses and agent strings offer much useful information about Web usage, but additional understanding can be gained by further processing to identify commonalities in these data. We did this by parsing the IP addresses and agent strings into second-level ancillary data structures.

For each agent string value, we try to recognize the specific Web agent that produced it by identifying the string in an online list of agents and the agent strings they present to browsers (see [www.user-agents.org](http://www.user-agents.org)). Once we’ve identified the agent associated with each string in our logs, we then classify the agent into one of five classes: anonymous (no agent string reported), bot (for example, search engine crawler), administrative service (for example, local performance monitor), user-created program (for example, data downloader),

or Web browser. In most cases, the categorization is obvious, but sometimes it required some judgment between conflicting online sources. For programs, we also identify the programming language (for example, Python, Java, and so on). For Web browsers, we also identify the browser and version. We stored this agent information in a second-level ancillary data table.

For each IP address, we resolved its domain name, identifying which organizations own which blocks of IPs. By resolving the addresses, we can analyze hits at the level of institutions and categories of institutions (for example, research universities, teaching colleges, commercial ISPs, and so on) rather than just individual IPs.

To resolve domain names from IPs, we first selected valid IP addresses, defined as addresses with four numeric blocks divided by periods in which each block is numbered from 0 to 255; 99.97 percent of page views in our weblog data came from valid IP addresses. We parsed these IP addresses (which were previously stored as strings) into four integers.

Once we had numerical IP addresses, our next step was to resolve their domains using Whois, a local tool installed in most Unix/Linux/Berkeley Software Distribution systems. Whois looks up the records of domain registrations in online databases maintained by one or more regional Network Information Centers (NICs). By design, NIC servers limit the number of queries they will process from a single source within a particular time interval, so a brute-force Whois lookup of each of the nearly four million IPs in our logs would be impractical.

To reduce the number of queries required to the NIC servers, we wrote a Perl script (available on request) that used a simple technique: storing known IP ranges (that is, domain net ranges we had already resolved) in a local database. Information returned by each NIC includes the ranges of continuous IPs owned by each ISP, meaning that queries to any IP within those ranges would yield the same response from the NIC servers.

When our script considered a new IP address, it first checked the local database. If the IP being queried is already in that local database, it’s not sent to the NIC servers; otherwise we execute the query and store the new results in the local database. We searched five NICs using our script to get some information on each IP range: domain owner, IP range, country, and address.

The script can be executed several times. We consider the output to be complete—that we got

all possible information from a set of input IPs—when the script informs us that it has received a response back from at least one NIC server for every IP in the input data, even if many of the responses returned no match found.

After our script completed, we wrote a similar Perl script to search the American Registry for Internet Numbers (ARIN), which is responsible for assigning IP addresses in the US and Canada. We resolved domains using ARIN last because information from other NICs is often copied into ARIN's records, but the information from the other NICs is generally more reliable. Thus, when our ARIN script returned a net range that we already had in our local database, we retained the existing record.

Using these scripts, we were able to resolve domains for 93 percent of page views in our logs. The scripts resolved these addresses into 37,597 unique domains, which we stored in a second-level ancillary data table.

However, the 37,597 domains that our scripts resolved don't simply correspond to 37,597 organizations with SkyServer page views. When an organization owns two nonconsecutive IP address ranges, the organization will appear twice in our table of resolved domains. Even more troubling are the cases where Whois returns two or more different results for the name, but examination of the output clearly shows that they are the same organization. For example, two of the top 50 domains by traffic that we found resolved as "AT&T Internet Services" and "AT&T Services, Inc.," the same parent organization.

Clearly, some level of human judgment is required to interpret the Whois results. Identifying the parent organizations of each domain resolved by our scripts would require detailed research on every one of the 37,597 output domains. Fortunately, the unevenness of Web traffic by domain means that we can account for most traffic by examining only a few domain names: in our case, 95 percent of all page views to our resolved domains by examining just 2,296 results. We looked at each of these 2,296 domain organizations manually, supplemented by some online research, and made judgments about what records represented truly different parent organizations.

This classification process led to 1,047 different domain organizations. We then assigned a category to each domain organization: research institute, university, college, community/technical college, K-12 institution, ISP, business (non-ISP), and government organization.

Last, we loaded all these organizations into the same SQL database as a third-level ancillary data table. There's naturally some uncertainty in this analysis; our classifications are available on request.

**Page views and sessions.** As described previously, a single user action frequently results in many hits, so in Web analytics studies it's important to define a page view as a metric of what actions a user has consciously taken. Our definition of a page view is the same one that we used in our prior five-year study of SDSS weblogs.<sup>4</sup> A hit has all of the following characteristics:

- responds to a GET, HEAD, PUT, or POST HTTP request;
- returns a result (status 200–299);
- comes from a Web browser or a program that the user has written to download data (for example, a Java app or Python script); and
- has a command requesting a file type that delivers data or information of interest to the end user: .asp(x), .htm(l), .asmx, .csv, .fits, .pdf, .swf, .(e)ps, .doc(x), .xls(x), .ppt(x), or a directory default page.

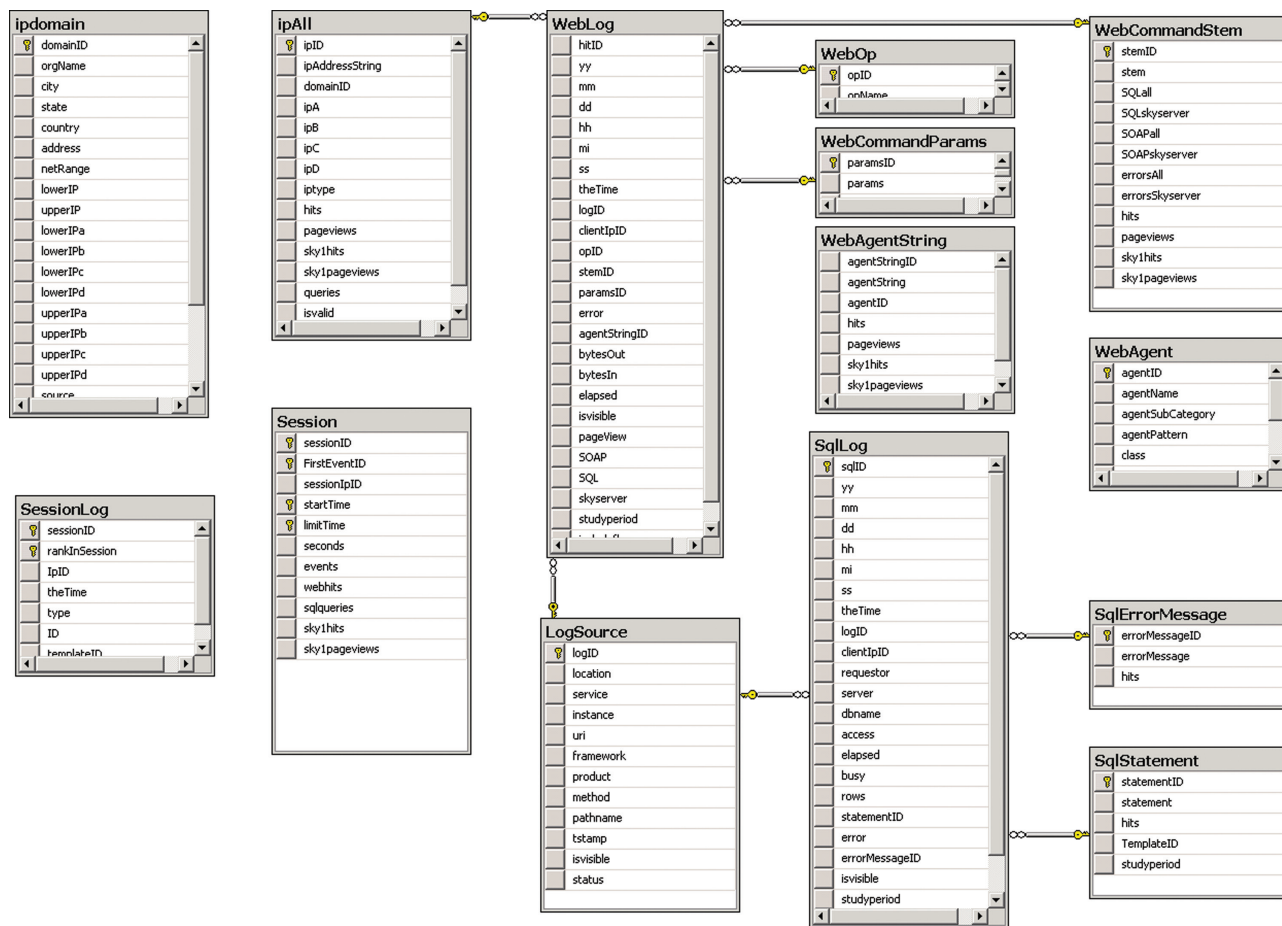
Page views are indicated with a Boolean flag variable called `pageview` in the normalized weblog table.

As described previously, Web analytics studies frequently define a session as the ordered sequence of hits from a single IP address such that the think time between each hit is reasonably short. Our prior research<sup>4</sup> chose this cutoff time to be 30 minutes; in our new 10-year dataset, 98.7 percent of gaps fall under this 30 minute cutoff, giving us confidence that we aren't missing many genuine user sessions with this definition. Thus, we chose to use the same definition of sessions in our current work.

Identifying sessions in this way allows us to create two additional ancillary data structures:

- *Session logs.* The combined weblog and SQL log data is sorted first by client IP address and then by time. The logs include a session ID; when the time gap between successive hits from the same client IP increases above 30 minutes, a new session is declared.
- *Sessions.* Information about each session is identified in the session logs, including the client IP address, the start and end time, and the number of Web hits and SQL queries included in that session.





**Figure 3.** The schema of the database that hosts the normalized weblog and SQL logs harvested from all the SDSS websites. At the heart of the schema are the WebLog and SqlLog tables, which contain normalized versions of the WeblogAll and SQLlogAll tables shown in Figure 2. The WebLog and SqlLog tables connect by foreign keys to a variety of ancillary data tables, providing more information about a single Web hit or SQL query. By using the ancillary tables shown here, it is possible to derive the statistics reported in this article in an efficient way, as well as to reconstruct all the information about each raw weblog and sqlLog record (from Figure 2) if needed.

Querying these tables provides us with the paths that users follow through our sites, allowing us to see where users go to look for information.

**Data Availability**

The cleanup and normalization effort benefited greatly from the work done for the five-year weblog study,<sup>4</sup> but still took several months of effort. Figure 3 shows the database diagram for the resulting normalized database.

The data described in this article are available through CasJobs (see <http://skyserver.sdss3.org/cas-jobs>) as a query context called SdssWeblogs. See the help resources on that site for more information. We have also made the data available for download at <ftp://dss001.pha.jhu.edu/SdssWeblogs/> in two forms: a compressed multifile SQL Server backup,

which can be restored as a SQL Server database with all the original schema; and a compressed comma-separated values dump of the tables in the analysis database, as Figure 3 shows. We encourage other researchers to make use of our data.

The outcome of this research is a normalized, efficient database of Web hits and SQL queries to a large scientific database. Queries that would take hours or days to complete in the raw weblogs complete in seconds in this normalized weblog database. Our weblog database can be a powerful tool for studying how scientists and the public use these data and educational resources.

Many other studies are possible. We invite collaboration with other researchers, and we also strongly

encourage other e-science data providers to save their weblog data for future analyses. These logs contain detailed portraits of how the big data revolution is playing out in real time, in the research community and beyond. We look forward to the new understandings that this knowledge can bring us. ■

### Acknowledgments

Our program of research into the SDSS weblogs was initiated by our colleague Jim Gray of Microsoft Research, who was lost at sea in January 2007. Jim made many direct contributions to this project—and in addition, none of this research would ever have been possible without Jim's pioneering contributions to data-intensive science. Indeed, it was at Jim's insistence that we set up—at the very outset—a system to log and harvest every single Web hit and SQL query that was ever submitted to SkyServer. We're grateful for the opportunity to have known Jim as a respected colleague and dear friend.

We're also grateful to our former colleague Vamsi Vakiti for his help in setting up the analysis database for normalization, and for formulating SQL queries to the database.

This material is based upon work supported by the Alfred P. Sloan Foundation. Funding for the SDSS and SDSS-II was provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the US Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. The SDSS was managed by the Astrophysical Research Consortium for the Participating Institutions.

### References

1. N.W. Jankowski, "Exploring e-Science: An Introduction," *J. Computer-Mediated Comm.*, vol. 12, no. 2, 2007, pp. 549–562.
2. H.W. Park, "Mapping the e-Science Landscape in South Korea Using the Webometrics Method," *J. Computer-Mediated Comm.*, vol. 15, no. 2, 2010, pp. 211–229.
3. A.R. Thakar et al., "The Catalog Archive Server Database Management System," *Computing in Science & Eng.*, special issue on SDSS science archive, vol. 10, no. 1, 2008, pp. 30–37.
4. V. Singh et al., *SkyServer Traffic Report—The First Five Years*, tech. report MSR-TR-2006-190, Microsoft Research 2006; <http://research.microsoft.com/apps/pubs/default.aspx?id=64520>.
5. B.J. Jansen, *Understanding User-Web Interactions via Web Analytics: Synthesis Lectures on Information Concepts, Retrieval, and Services*, Morgan & Claypool, 2009, p. 102.
6. F.M. Facca and P.L. Lanzi, "Mining Interesting Knowledge from Weblogs: A Survey," *Data & Knowledge Eng.*, vol. 53, no. 3, 2005, pp. 225–241.
7. *Computing in Science & Eng.*, special issue on SDSS science archive, vol. 10, no. 1, 2008; [www.computer.org/csdl/mags/cs/2008/01/index.html](http://www.computer.org/csdl/mags/cs/2008/01/index.html).
8. A.R. Thakar and N. Li, "CasJobs and MyDB—A Batch Query Workbench," *Computing in Science & Eng.*, special issue on SDSS science archive, vol. 10, no. 1, 2008, pp. 18–29.
9. A.R. Thakar, "The Sloan Digital Sky Survey—Drinking From the Fire Hose," *Computing in Science & Eng.*, special issue on SDSS science archive, vol. 10, no. 1, 2008, pp. 9–12.
10. R.D. Santos et al., "Analysis of Web-Related Threats in Ten Years of Logs from a Scientific Portal," *Proc. SPIE*, vol. 8408, 2012, article no. 84080H; doi:10.1117/12.919545.

---

**M. Jordan Raddick** is a science education developer at Johns Hopkins University. He's the science evangelist for the Sloan Digital Sky Survey, focused on making SDSS data available and useful to the scientific community and the world. His research interests include looking at how citizen scientists learn science by doing science. Raddick has an MS in science writing and education from the Johns Hopkins University. Contact him at [raddick@jhu.edu](mailto:raddick@jhu.edu).

---

**Ani R. Thakar** is a principal research scientist at Johns Hopkins University. His research interests include data-intensive science and interacting galaxies. Thakar has a PhD in astronomy from the Ohio State University. Contact him at [thakar@jhu.edu](mailto:thakar@jhu.edu).


---

**Alexander S. Szalay** is the Alumni Centennial Professor of Physics and Astronomy at Johns Hopkins University. His research interests include cosmology, the large-scale structure of the universe, data mining, and science with large databases. Szalay has a PhD in astrophysics from Eötvös University, Hungary. Contact him at [szalay@jhu.edu](mailto:szalay@jhu.edu).

---

**Rafael D.C. Santos** is a senior technologist at the National Institute of Space Research (INPE), Brazil. His research interests include data mining and distributed systems. Santos has a PhD in computer science from the Kyushu Institute of Technology in Japan. Contact him at [rafael.santos@inpe.br](mailto:rafael.santos@inpe.br).

---

 Selected articles and columns from IEEE Computer Society publications are also available for free at <http://ComputingNow.computer.org>.