SDSS Dataset and SkyServer Workloads

1 Overview

Understanding the SDSS dataset composition and typical usage patterns is important for identifying strategies to optimize the performance of the AstroPortal and to effectively test the performance of the "stacking" service. This document is a first attempt at trying to describe both the dataset and observed access patterns.

The following sections cover information regarding the SDSS dataset and the workloads we plan to use to test the AstroPortal with. These workloads include random workloads, random workloads with varying data locality, SkyServer workload traces, and realistic random workloads.

1.1 Dataset

About a year ago, I had extracted some 320 million object coordinates from the SkyServer DR4 database. After performing a lookup for every one of the 320M objects, I got a histogram of each file (of the 1.3M files) and the number of objects it contains (ordered by the number of objects). The histogram of the object distribution in the SDSS DR4 dataset is shown below in Figure 1.

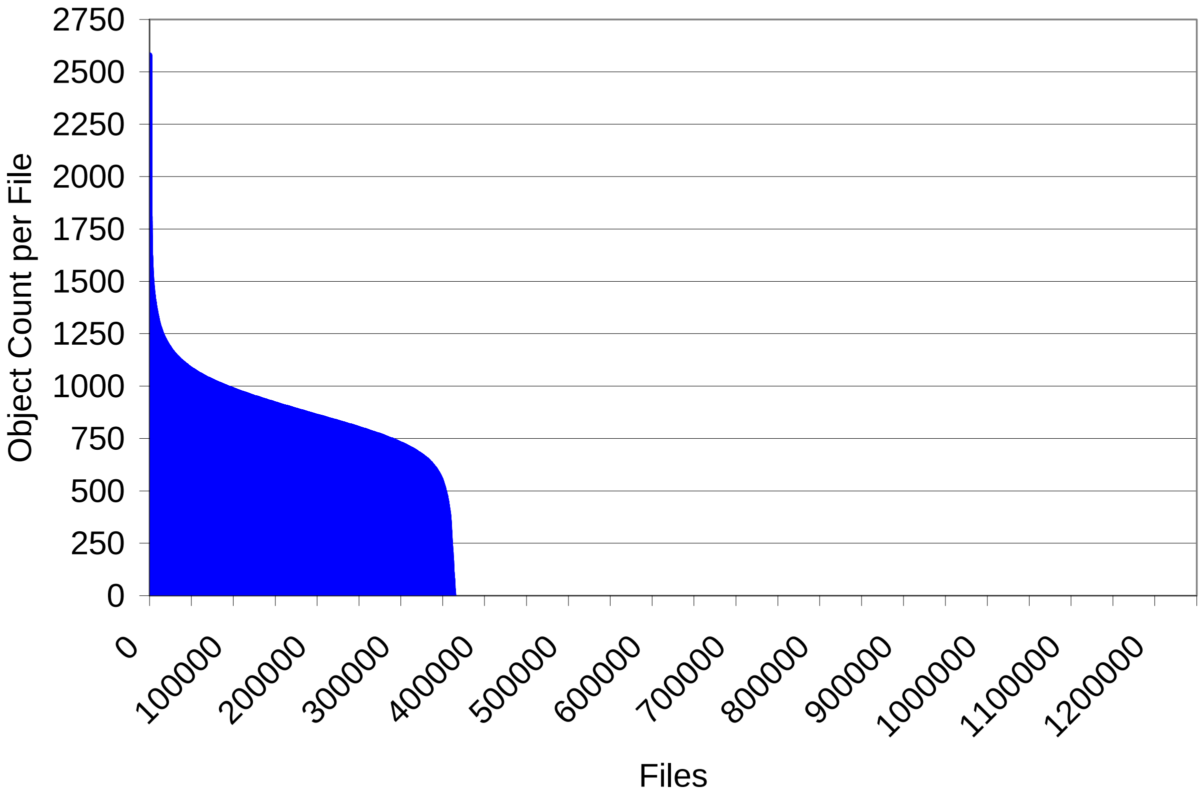


Figure 1: Object distribution in the SDSS DR4 dataset files

From what I see in Figure 1, of the 1.3 million files in DR4, only 350K some files have any objects in them. What is also interesting is that the objects are not uniformly distributed over the data files, but that there is a normal-like distribution to the number of files with low/medium/high number of objects per file.

Figure 2 graphically shows the spatial distribution of the files in the SDSS DR4 dataset.

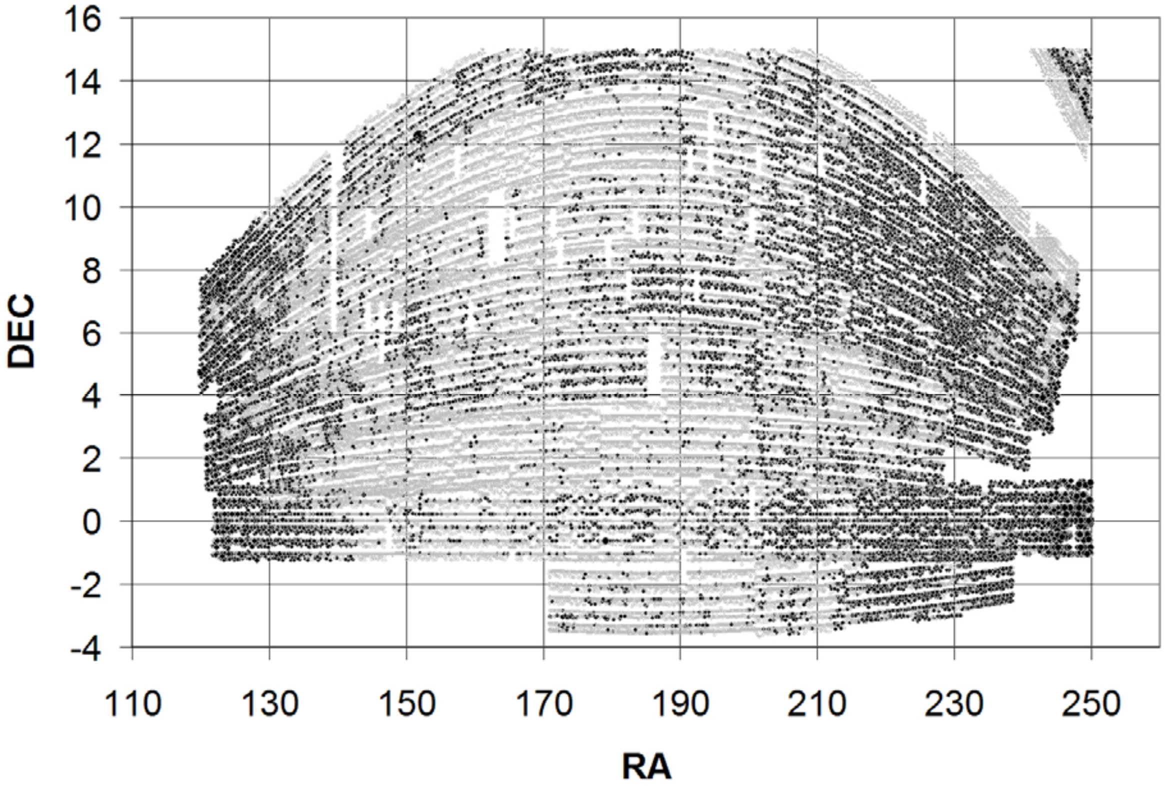


Figure 2: Spatial distribution (according to the RA and DEC coordinates) of the files in the SDSS DR4 dataset; each dot represents a file that could contain a varying number of objects; the more objects there are in a file, the darker and bigger the dot is

In essence, I believe we could use the data shown in Figure 1 and Figure 2 to augment and improve the various caching strategies. In principle, any caching strategy could be enhanced by simply factoring in the weight (# of objects) of each entry, and hence ensuring that entries with many objects will survive longer in the cache before being evicted. Since the object distribution does not change over the life of a dataset, the use of this information should be straight forward and easy to implement.

1.2 Workloads

Choosing the right workloads to quantify the performance of the proposed systems is crucial to understand the real world benefits when the AstroPortal will be in production mode used by the Astronomy community. We have defined several workloads that should test a wide range of performance metrics, and allow us to identify among the various optimizations which ones will offer the best performance improvements.

One important characteristic of the workloads we will define is the data locality. We have 320 million objects that are distributed over 1.2 million files. When a "stacking" query is made, we define the data locality metric for this query to be a percentage according to Equation 1, where numbers close to 100% imply high data locality, and numbers close to 0% imply low data locality:

Equation 1: Data Locality Metric

$$\left(1-\frac{unique\_files\_accessed}{objects\_accessed}\right)*100$$

The motivation behind this data locality metric definition is that we perform data management on the file level, and the fact that access patterns of various objects that allow the same files to be read should produce better performance. The data locality metric can apply to both 1) a single stacking operation of multiple objects, and 2) to multiple stacking operations performed both sequentially and in parallel as the same computational/storage resources could be used to perform the set of stacking operations.

The stacking sizes are also very important to investigate since it will directly impact the communication cost as there will be more control and data information that will need to flow from one component to another. We intend to investigate stacking sizes ranging from 1 to 32768 as that will cover typical real world stacking operations.

1.2.1 Random Workload

The random workload represents a random sampling (uniformly distributed) of objects chosen from the collection of 320 million objects from the SDSS DR4 dataset. We will investigate both stackings of the same size, as well as stackings of varying sizes. Once the workload is defined, the same workload will be used for all experiments in order to allow direct comparisons between the performance of the AstroPortal (as well as the related systems) and the various optimizations included. With the large number of objects (as well as the large dataset), a uniform sampling of objects will likely produce workloads with very little (if any) data locality.

1.2.2 Random Workload with Varying Data Locality

Building upon the random workload defined above, we can further refine the workload by varying the data locality from 0% to 100%. We will define 5 different workloads with data locality set to 0%, 25%, 50%, 75%, and 100%. At one extreme, we have a sweep like access pattern that has each accessed object in a unique file; this should allow us to see any overhead (and potential slowdown) incurred by the data management system as the data caching will likely get very low hit rates. The other extreme, we have full data locality, in which all objects accessed are contained in the same file, allowing the caching strategies to get maximal cache hit rates. The other data locality investigated (25%, 50%, and 75%) gives us an overview of how well various caching strategies fare to varying data locality in the workloads, and at what point they might become ineffective.

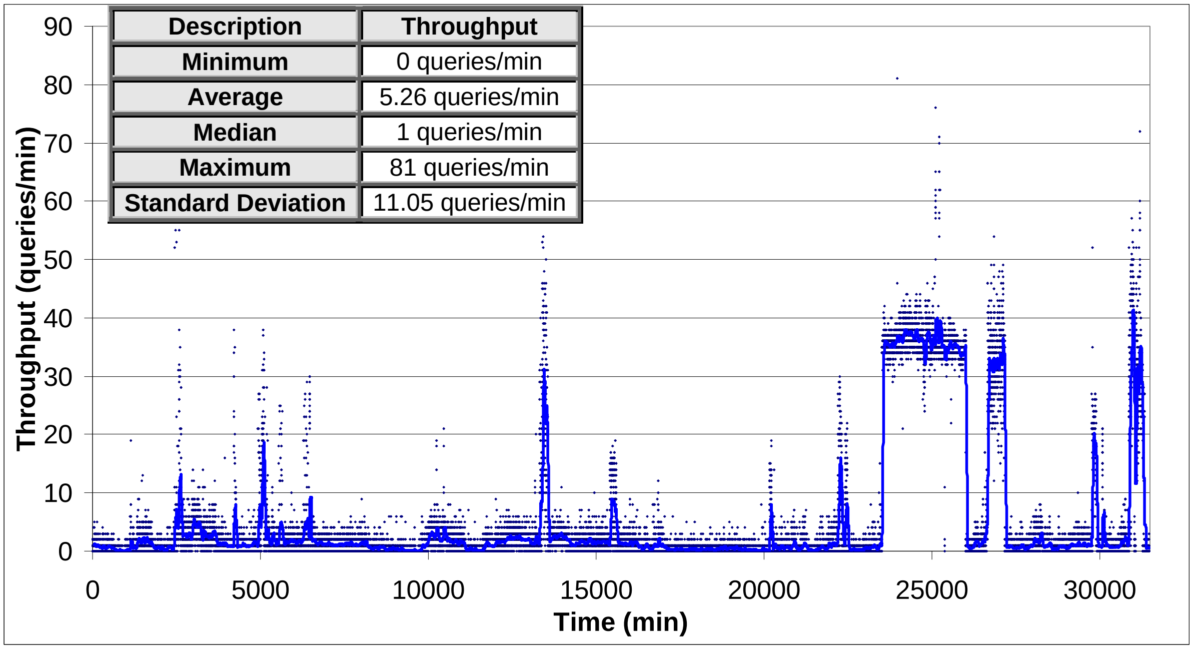
1.2.3 SkyServer Workload Traces

The SkyServer [2] is the most widely used system by the Astronomy community for searching through the SDSS catalogs to locate needed information. All the SQL queries against the SkyServer are logged with the following information: date & time of query, client IP address, requestor, server, database name, access, elapsed time to complete query, the number of rows retrieved, the SQL query, and any error message that might have been produced. The logs can be queried at http://skyserver.sdss.org/dr1/en/tools/search/sql.asp with a sample query of the form “select top 100 * from sdssad2.weblog.dbo.sqllog”. Although the queries retrieved are not from a “stacking” application, all the queries are from the SDSS dataset, and they are likely to be coming from the same set of users that will use the stacking application as well. Until we have the Astronomy community using the AstroPortal so we can gather our own stacking traces, the best we can do is to capture some workload traces of other applications using the same dataset.

The trace we plan to use covers the period from October 29th, 2006 through November 21st, 2006. Since a “stacking” application normally needs at least 2 objects to stack, we omitted any queries that returned either 0 or 1 object. We were left with a trace that was produced by 900 distinct users (based on their IP addresses) containing 165K queries returning 1190M objects.

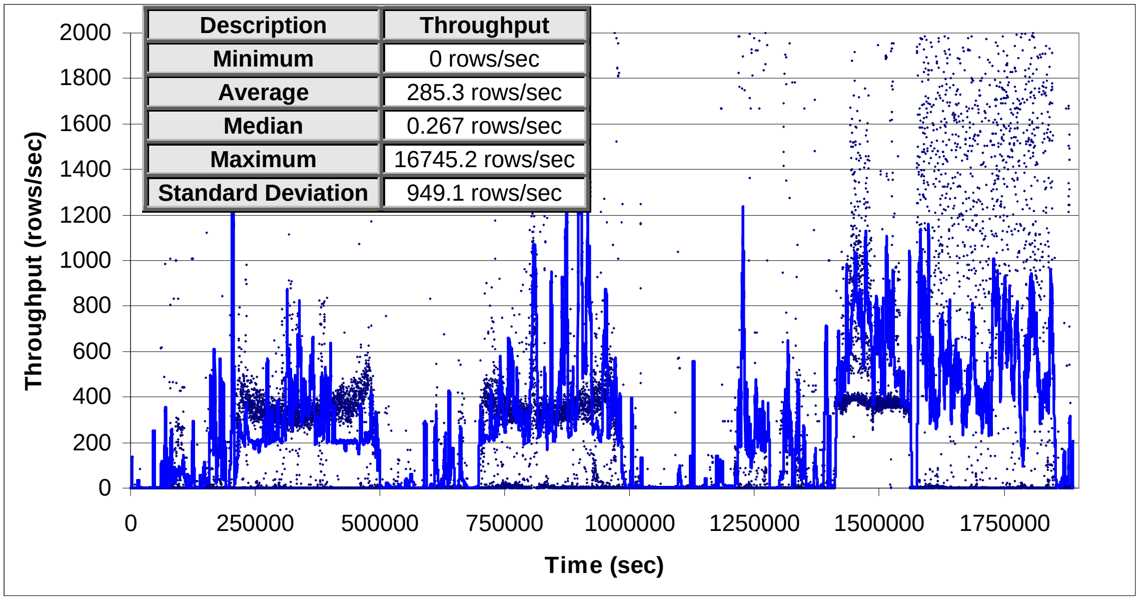
1.2.3.1 Raw Logs

Figure 3, Figure 4, and Figure 5 shows a pictorial representation of the raw logs obtained from the SkyServer between the dates of 10/29/06 and 11/21/06. In Figure 3 and Figure 4, we have plotted the throughput in both number of queries per minute and the number of rows retrieved per second. Both of these metrics gives us an overview of the kinds of sustained throughput that the AstroPortal will have to maintain to keep up with this particular workload. For example, the AstroPortal would need to be able to do on average about 5 queries per minute, translating to an average of about 285 stackings per second.



Description	Throughput
Minimum	0 queries/min
Average	5.26 queries/min
Median	1 queries/min
Maximum	81 queries/min
Standard Deviation	11.05 queries/min

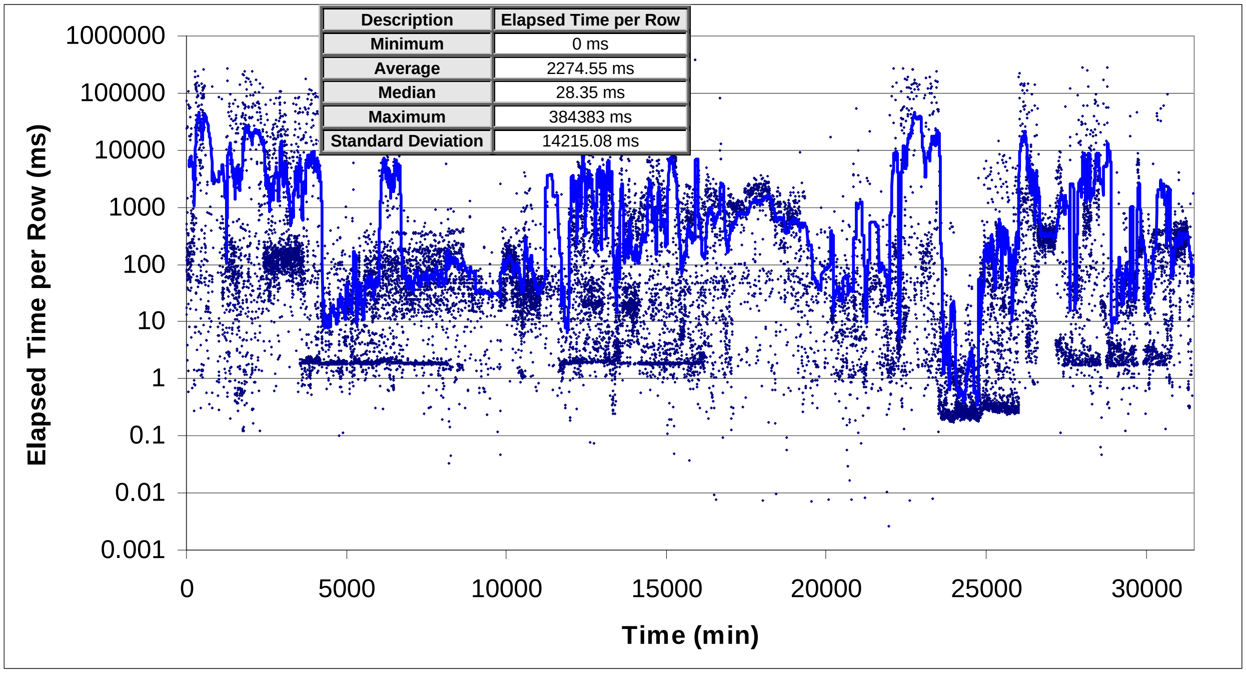
Figure 3: SkyServer logs (10/29/06 – 11/21/06) throughput in terms of SQL queries per minute



Description	Throughput
Minimum	0 rows/sec
Average	285.3 rows/sec
Median	0.267 rows/sec
Maximum	16745.2 rows/sec
Standard Deviation	949.1 rows/sec

Figure 4: SkyServer logs (10/29/06 – 11/21/06) throughput in terms of rows returned per second by the performed SQL queries

Figure 5 shows the time it took the SkyServer to complete the queries, normalized by the number of rows retrieved. Notice that the graph has the y-axis in logarithmic scale (unlike the previous two graphs). Figure 5 shows quite a wide varying time per retrieved row, which shows the wide variance in performance based on the complexity of the query as well as the load on the SkyServer as multiple users concurrently run queries in parallel.



Description	Elapsed Time per Row
Minimum	0 ms
Average	2274.55 ms
Median	28.35 ms
Maximum	384383 ms
Standard Deviation	14215.08 ms

Figure 5: SkyServer logs (10/29/06 – 11/21/06) elapsed time per retrieved row in milliseconds

1.2.3.2 Summary of Logs

The following set of figures (Figure 6, Figure 7, Figure 8, and Figure 9) attempt to summarize the raw logs obtained from the SkyServer between the dates of 10/29/06 and 11/21/06. We have created histograms of various metrics computed from the raw logs:

- Inter arrival time of queries
- Query size
- Time per retrieved row
- Time per query

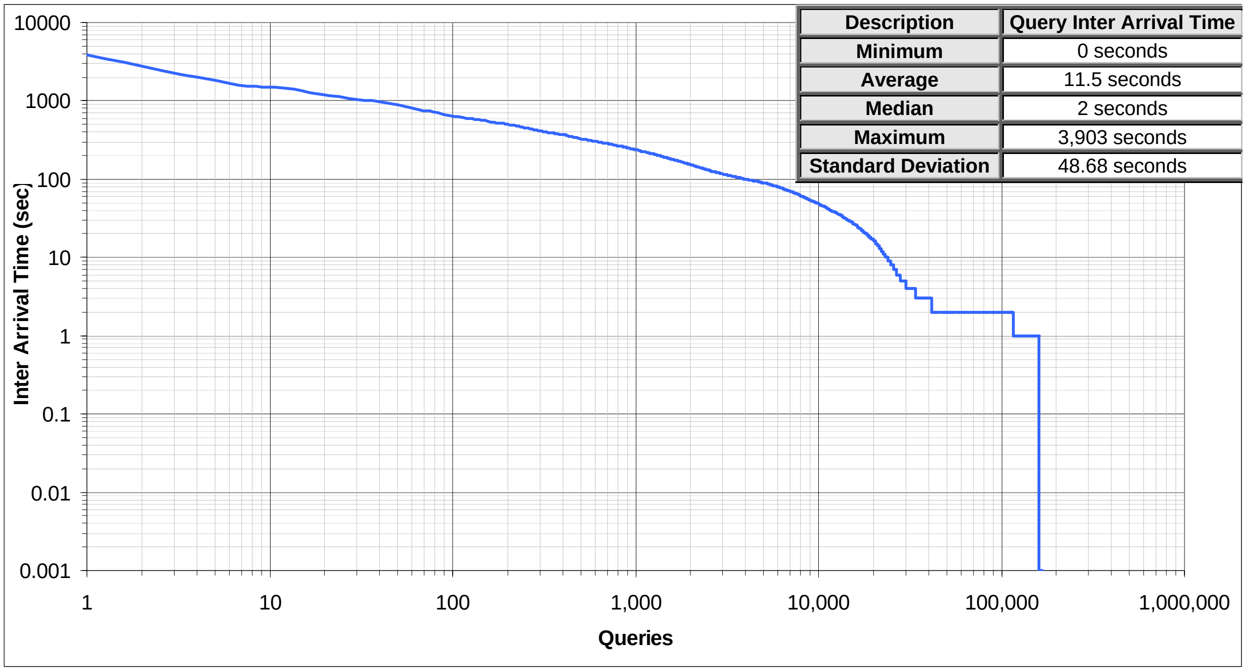
There is another set of figures (Figure 10, Figure 11, Figure 12) which attempt to summarize the same logs in relation to the clients rather than the queries. Characterizing the clients can allow us to understand the workload distribution among the clients.

One thing that we have not investigated yet is the data locality found in the query results, which we plan to do. The data locality information is important as the performance of the various optimizations we have made to the AstroPortal will be heavily influenced based on how low/high the data locality is in typical real world workloads.

Notice that all the graphs have both the x-axis and y-axis in logarithmic scale. Some of these graphs have tendencies to show Zipf distributions; Zipf curves follow a straight line when plotted on a double-logarithmic diagram. A simple description of data that follow a Zipf distribution is that they have:

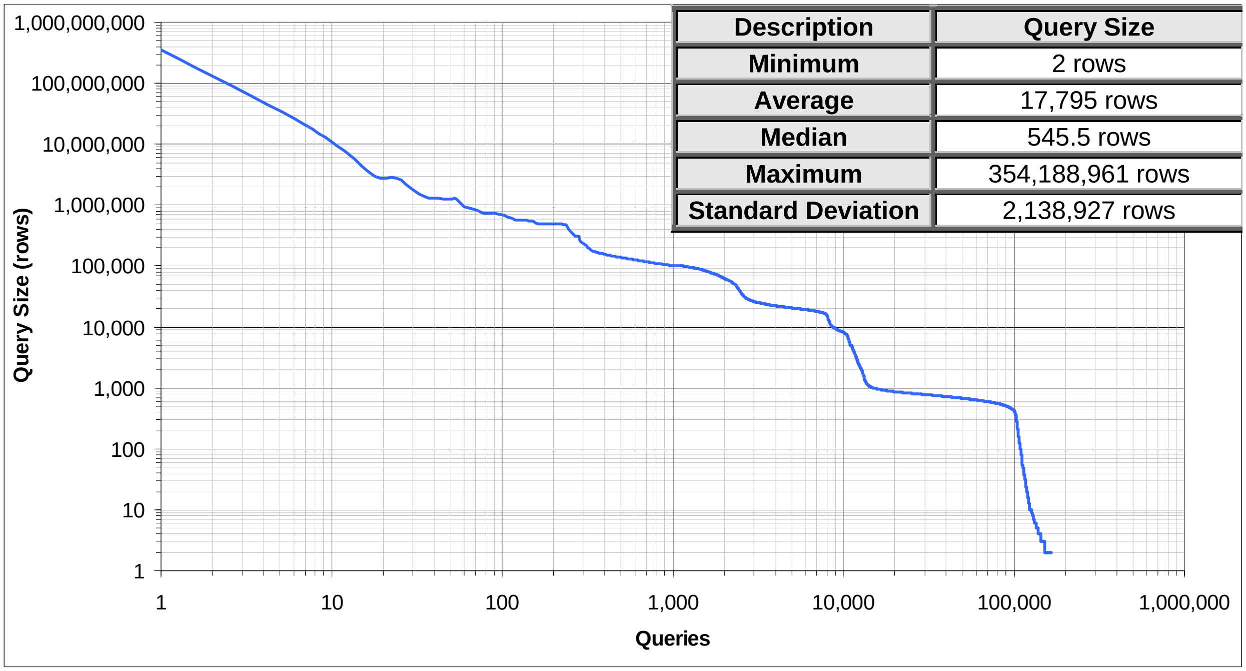
- a few elements that score very high (the left tail in the diagrams)
- a medium number of elements with middle-of-the-road scores (the middle part of the diagrams)
- a huge number of elements that score very low (the right tail in the diagrams)

This property is desirable to identify in the captured logs as it will help generate realistic random workloads based on the distribution of the various metrics that makeup a workload: query size, inter arrival time, time per query, time per row, and the distribution of the work among the various clients.



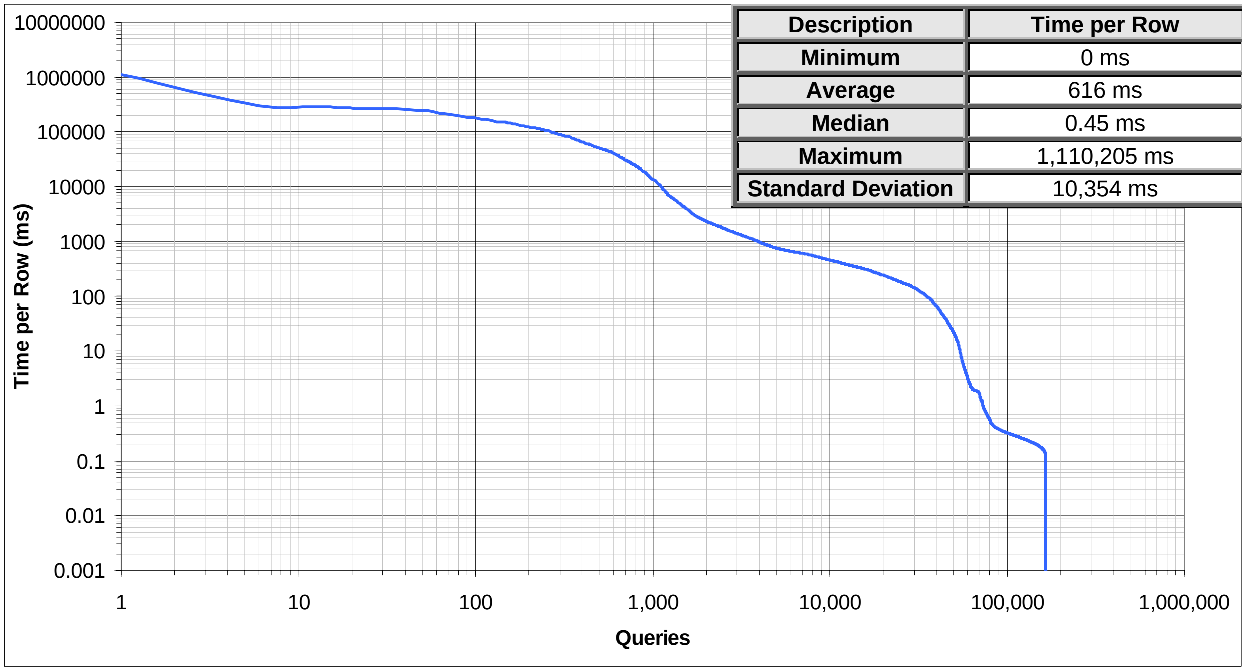
Description	Query Inter Arrival Time
Minimum	0 seconds
Average	11.5 seconds
Median	2 seconds
Maximum	3,903 seconds
Standard Deviation	48.68 seconds

Figure 6: SkyServer logs (10/29/06 – 11/21/06): inter arrival time of queries histogram in seconds



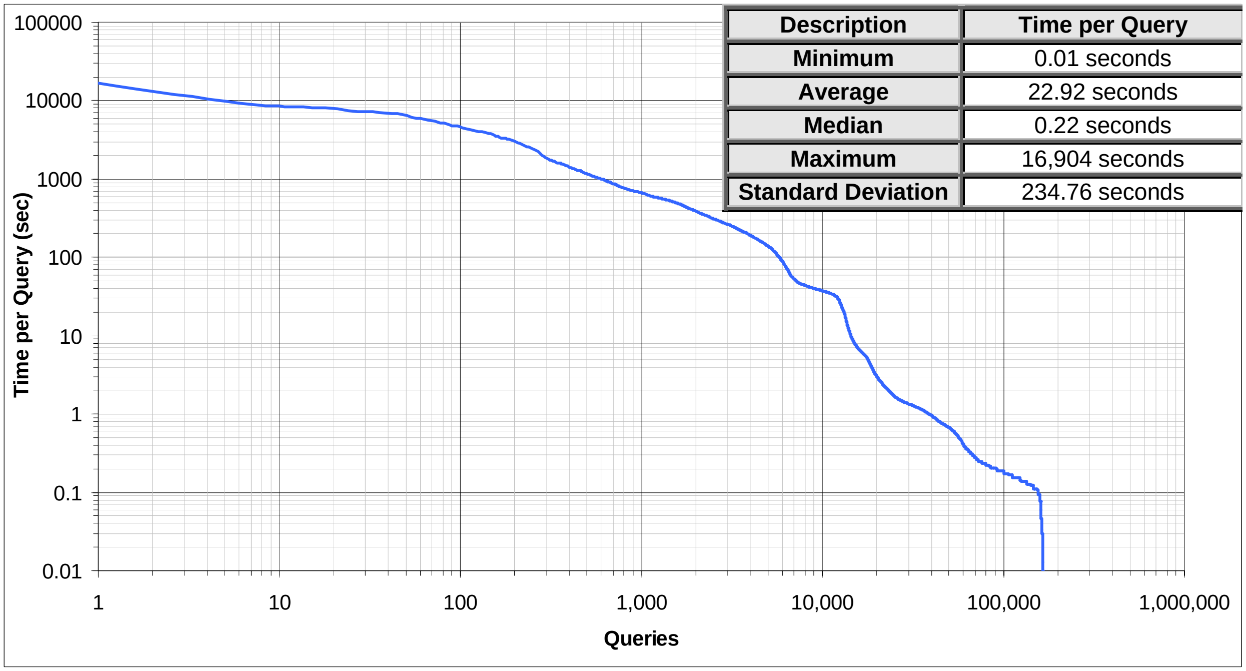
Description	Query Size
Minimum	2 rows
Average	17,795 rows
Median	545.5 rows
Maximum	354,188,961 rows
Standard Deviation	2,138,927 rows

Figure 7: SkyServer logs (10/29/06 – 11/21/06): average query size histogram in seconds



Description	Time per Row
Minimum	0 ms
Average	616 ms
Median	0.45 ms
Maximum	1,110,205 ms
Standard Deviation	10,354 ms

Figure 8: SkyServer logs (10/29/06 – 11/21/06): time per retrieved row histogram in milliseconds

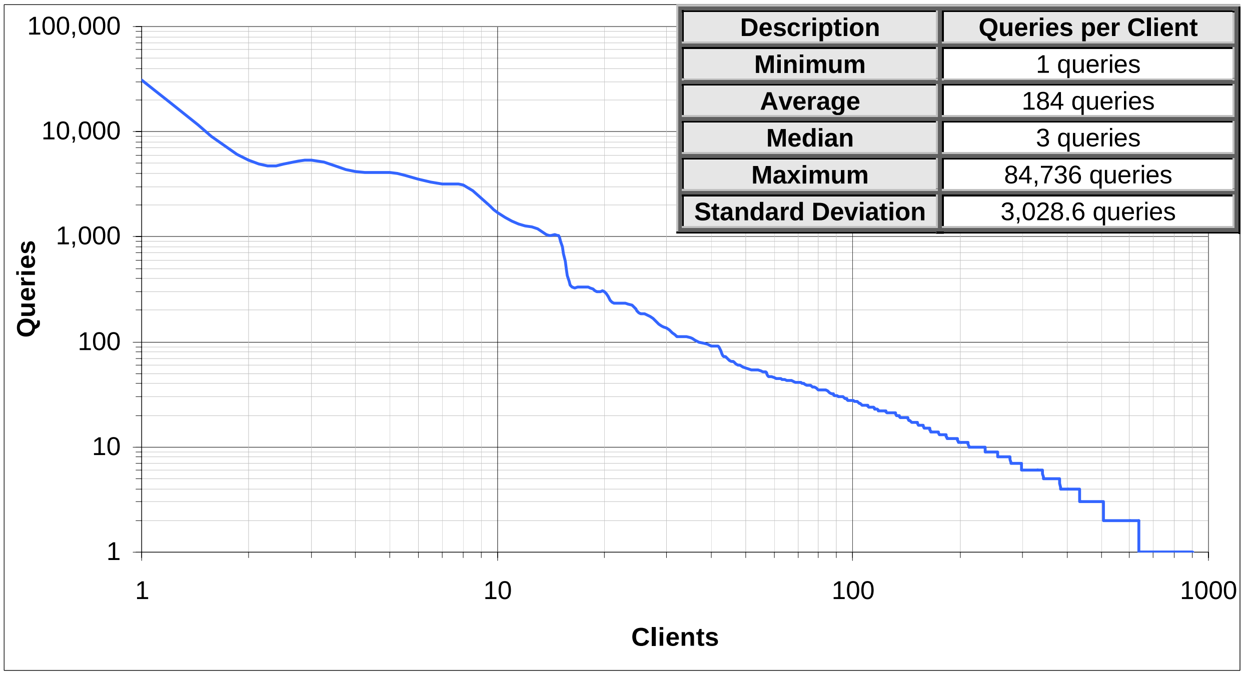


Description	Time per Query
Minimum	0.01 seconds
Average	22.92 seconds
Median	0.22 seconds
Maximum	16,904 seconds
Standard Deviation	234.76 seconds

Figure 9: SkyServer logs (10/29/06 – 11/21/06): time per query histogram in seconds

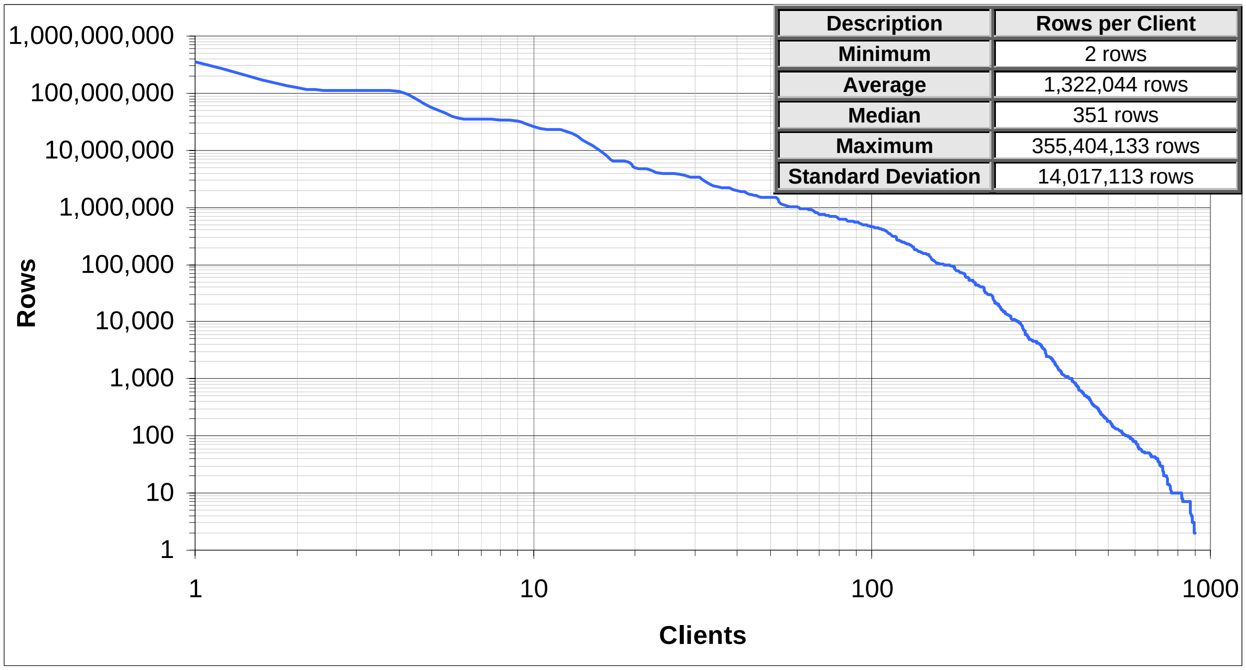
Another set of figures (Figure 10, Figure 11, Figure 12) which attempt to summarize the same logs in relation to the clients rather than the queries are:

- Number of queries per client
- Average rows retrieved per client
- Average query size per client



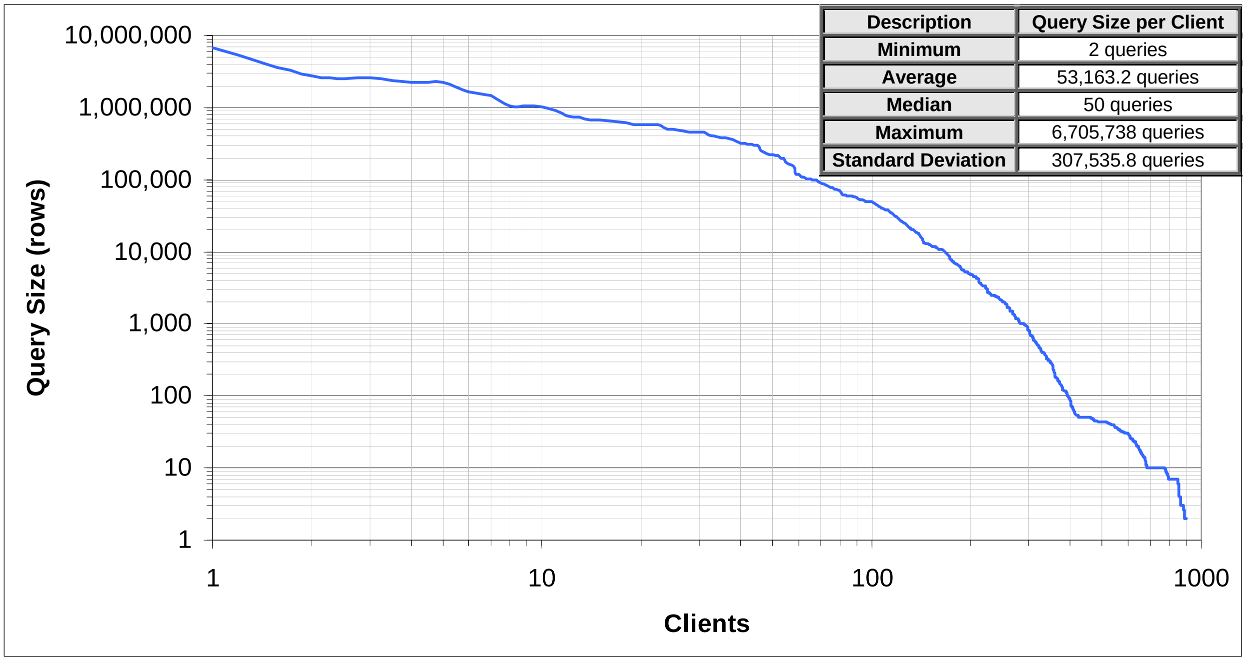
Description	Queries per Client
Minimum	1 queries
Average	184 queries
Median	3 queries
Maximum	84,736 queries
Standard Deviation	3,028.6 queries

Figure 10: SkyServer logs (10/29/06 – 11/21/06): number of queries per client histogram distribution



Description	Rows per Client
Minimum	2 rows
Average	1,322,044 rows
Median	351 rows
Maximum	355,404,133 rows
Standard Deviation	14,017,113 rows

Figure 11: SkyServer logs (10/29/06 – 11/21/06): number of rows retrieved per client histogram



Description	Query Size per Client
Minimum	2 queries
Average	53,163.2 queries
Median	50 queries
Maximum	6,705,738 queries
Standard Deviation	307,535.8 queries

Figure 12: SkyServer logs (10/29/06 – 11/21/06): average query size per client histogram

1.2.4 Realistic Random Workload

We expect that the summary of the SkyServer workload traces to give us enough to be able to build realistic (but randomized) workloads for the AstroPortal using the SDSS dataset.

2 Bibliography

[1] Tanu Malik, Alex Szalay, Tamas Budavari and Ani Thakar, SkyQuery: A Web Service Approach to Federate Databases, The First Biennial Conference on Innovative Database Systems Research (CIDR), 2003.

[2] Alex Szalay, Jim Gray, Ani Thakar, Peter Kuntz, Tanu Malik, Jordan Raddick, Chris Stoughton and Jan Vandenberg, The SDSS SkyServer - Public Access to the Sloan Digital Sky Server Data, ACM SIGMOD, 2002. http://www.skyserver.org/default.aspx