CMPT 815 Assignment 1: Trace Characterization and Modeling

Wenguang Wang

February 29, 2000

1 Workload Characterization

The trace examined in this assignment is a one-day trace gathered from a busy corporate World Wide Web server. It has been preprocessed so that only GET requests with status 200 (successfully completed) are contained. A summary of this trace is shown in Table 1.

Duration	Total Requests	Total Bytes Transferred
17.5 hours	4,067,179	23GB

Table 1: Trace Characteristics

Each line of the trace represents one request for an object (document) stored on the server. It is comprised of six fields as listed in Table 2.

Filed Name	Description
Inter-arrival Time	the time elapsed since the previous request, in microseconds
Service Time	the service time for the request, in microseconds
Object ID	a unique id for every server document referenced in the trace
Client ID	a unique id for each distinct web client presented in the trace
Transfer Size	the transfer size for an object, in bytes.
Object Type	the type of the object requested, including HTML, image, audio, video, *etc*

Table 2: Trace Entry Structure

1.1 Document Types and Sizes

In the trace used, all documents can be classified (based on URL extensions) into the following categories: HTML, Image, Audio, Video, Formatted, Compressed, Programs, Directory, and Others. Table 3 presents the statistics of these document types.

Item	HTML	Image	Audio	Video	Formatted	Compressed	Programs	Directory	Others
% of Requests	10.29	85.83	0.001	0.007	0.09	0.0009	0.004	2.74	1.04
% of bytes	25.99	63.28	0.22	0.14	3.36	0.12	0.91	5.08	0.90
Mean File Size	14,277	4,168	1,044,939	471,421	218,041	794,742	120,7817	10,488	4,866
Median File Size	12,147	888	856,064	77,824	53,248	650,687	503,808	11,456	223
COV of File Size	0.88	1.97	0.88	1.70	2.11	0.83	1.17	0.41	15.95

Table 3: Document Types and Sizes

Table 3 shows that HTML and image documents account for over 95% of the total requests. The image type is the most requested document type (85%), followed by a much smaller percentage of HTML files

(10%). These two types are also responsible for most of the bytes transferred (89%). The image documents rank first (63%), followed by HTML files (26%).

On the contrary, there are relatively infrequent requests to audio, video, compressed, programs, and formatted files (they all together only account for 0.1% of the total requests). However, their percentage of bytes transferred is much larger (totally about 4.7% of the total bytes transferred). This is because of the relatively large sizes of these types of documents compared to the other document types (as shown in table 3, the much larger mean and median file sizes of audio, video, compressed, programs, and formatted types).

These characteristics are consistent with [1] [2].

1.2 Inter-arrival Time

Figure 1 shows the Probability Density Function of inter-arrival time. The y-axis is in log-scale. Because the inter-arrival time in the trace is in microseconds (which is too small to be coped with), all arrivals within milliseconds $[a-0.5ms, a+0.5ms)$ are counted and assumed to occur at millisecond a. Then, the Probability Density Function is computed based on this reduced data set (which is in the unit of milliseconds).

The resulted PDF is very close to a straight line with negative slope, which indicates that it follows an *Exponential* distribution. This is because the PDF of Exponential distribution is $f(x) = \frac{1}{a}e^{-\frac{x}{a}}$. Its equation in log-log scale is $logf(x) = -\frac{loge}{a}x + (-loga)$, which is a straight line with negative slope. To verify this hypothesis, the statistics of inter-arrival time are computed and listed in Table 4. The coefficient of variance (C.O.V.) is 1.18, which is close to the C.O.V of the Exponential distribution (which is 1).

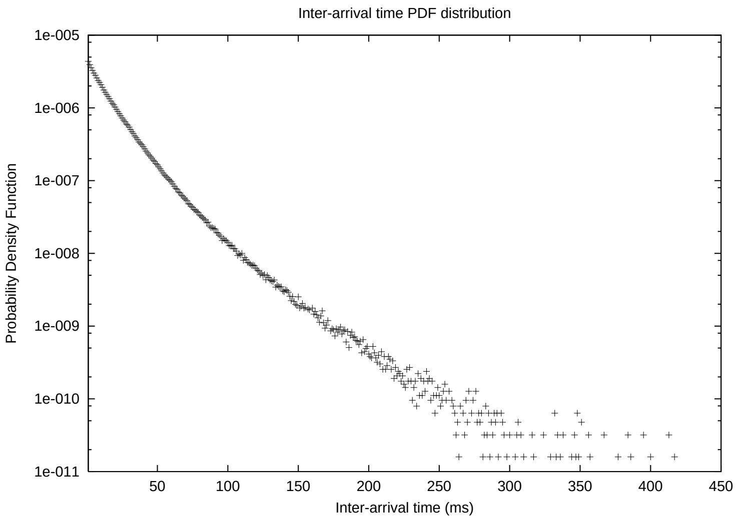


Figure 1: Probability Density Function of Inter-arrival Time

Mean	Stdv	C.O.V.
15471	18300	1.18

Table 4: Characteristics of Interval-arrival Time

The exponentially distributed inter-arrival time means that there is little dependence between the arrivals of client requests. The reason for this is many clients send requests to the Web server independently.

1.3 Transfer Size Distribution

Figure 2 presents three kinds of transfer size distributions (from left to right): the distribution of the reference count of distinct transfer sizes, the distribution of distinct transfer sizes, and the distribution of the bytes transferred, respectively. The x-axis of the figure draws the sorted distinct transfer sizes in a logarithmic scale. The y-axis depicts the Cumulative Density Functions.

It can be easily seen from the distribution of distinct transfer sizes that most of the files transferred are in the range from 1,000 to 100,000 bytes, with some small ones (less than 1000 bytes) and a few large files (more than 100,000 bytes). Table 5 shows that the mean distinct transfer size is much larger than the median value (about 5 times). This skewed mean transfer size indicates that the distribution of distinct transfer sizes is *heavy-tailed*, which means that the very large "elephants" (i.e., outliers) in the tail of the distribution, although relatively few in number, are large enough to contribute significantly to the overall distinct transfer sizes observed. Figure 2 also shows that 80% of the references are for documents less than 10,000 bytes in size. However, the transfers of documents in this category account for only 27% of the total bytes transferred. On the other hand, the transfers over 25,000 bytes account for a significant 30% of the total bytes transferred. Moreover, Table 5 shows that both the mean file size referenced and the mean file size transferred are larger than their corresponding media values. This indicates that the distribution of transfer sizes by reference and the distribution of the bytes transferred are "heavy-tailed" as well.

The transfer size characteristics are consistent with [1] [5].

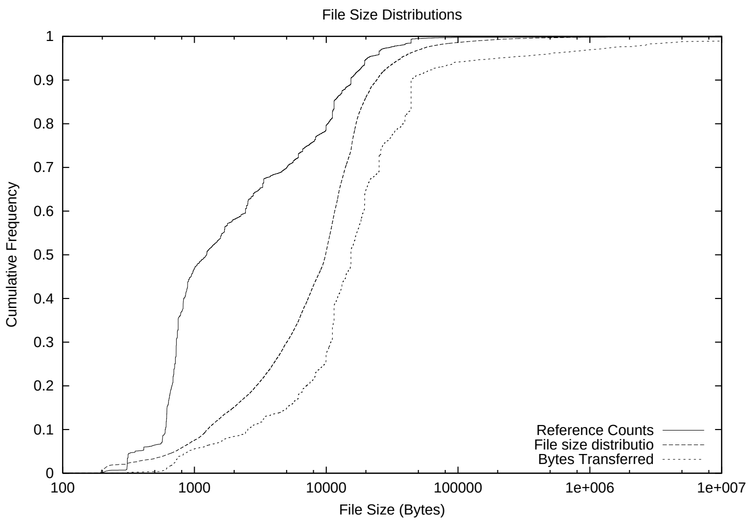


Figure 2: Transfer Size Distribution

Table 5: Transfer Size Statistics

	Distinct Transfer Sizes	Referenced Sizes	Transferred Sizes
Mean (Bytes)	43,701	5,654	975,648
Median (Bytes)	9,877	1,236	15,387
C.O.V	3.4	4.20	16.4

1.4 Document Popularity

This subsection focuses on the file referencing patterns. Clearly, not all Web documents are referenced equally. Some are extremely "hot" and popular documents, accessed frequently by many clients from many sites. Other documents are accessed rarely, if at all. Figure 3 illustrates the referencing behavior of the documents. It is drawn by sorting the list of distinct files into decreasing order based on how many times they were accessed, and then plotting the Cumulative Frequency of Requests versus the fraction of the total files referenced.

Figure 3 illustrates a non-uniform pattern of file referencing behaviors: 10% of the distinct documents are responsible for 94% of all requests received by the server. This characteristic is consistent with [1].

1.5 Client Distribution

In this trace, each client ID represents a unique IP address. If this IP address corresponds to a PC, it may represent a single user. If it is a UNIX machine instead, the requests sent out from it may be an aggregation

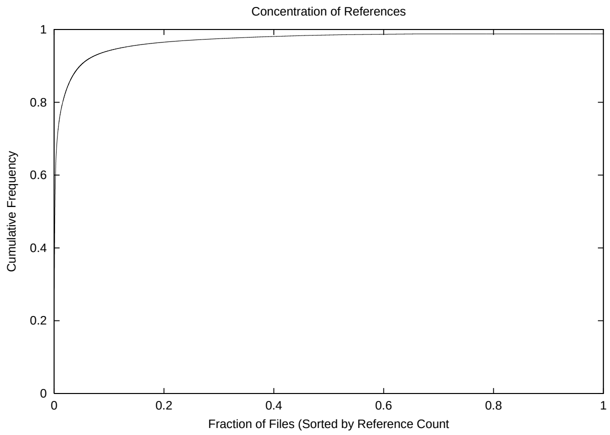


Figure 3: Concentration of Document References

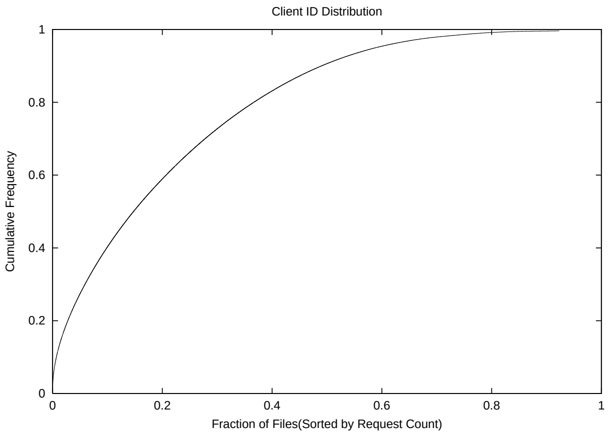


Figure 4: Concentration of Client References

of all users having logged into this machine. If it represents a proxy, it may hide the behaviors of many users in one LAN. Figure 4 shows the client reference distribution, which is drawn by sorting the list of distinct client IDs into decreasing order based on the number of requests they send out, and then plotting the Cumulative Frequency of requests versus the fraction of the total number of requests. Similar to the document popularity distribution, the distribution of client references is non-uniform: 15% clients generate 50% requests, and 49% clients generates 90% requests. This indicates that there are a few "active" clients who refer the documents in this Web server quite often. On the other hand, nearly half of the clients is "inactive" because their requests only account for about 10% of the total visits to the Web server.

1.6 Service Time Distribution

The service time is impacted by many factors, such as file size (a bigger file causes longer service time), Web server workload (a heavier server workload causes longer service time), cache (a cache miss causes longer service time), location of clients (a remote request needs longer service time), and network workload (a heavier network workload causes longer service time). Figure 5 shows the Probability Density Function of service time. The x-axis and the y-axis are both in log scales. It is easily seen from the figure that the service time spans a wide range and is very irregular.

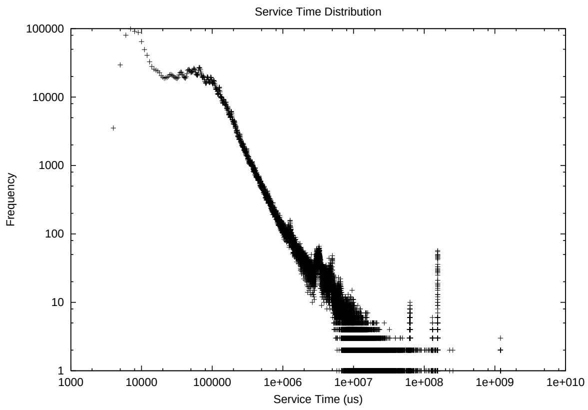


Figure 5: Service Time Distribution

2 Workload Modeling

Because the service time is very irregular, it is not modeled in this assignment. The other five fields of this trace are modeled. The modeling is based on the trace characteristics discussed in §1. The synthetic workload contains 500,000 entries, which is about $\frac{1}{8}$ of the original trace. The transfer size and the document popularity are combined together, while the other three fields are generated separately. The modeling techniques and results are presented in this section.

2.1 General Modeling Techniques

The random generator used for synthetic trace generation is:

$$x_{n+1} = 7^5 x_n \; mod \; (2^{31} - 1)$$

When more than one random streams are needed, precomputed seeds (x_0, $x_{50,000}$, $x_{100,000} \ldots$) are used. Because only 50,000 requests are generated, all random streams are non-overlapped.

The *uniform* distribution is used in the modeling. The Cumulative Density Function (CDF) of the *uniform* distribution is $F(x) = \frac{x}{X}$, where $0 \leq x \leq X$. Therefore, the CDF is a straight line in the linear-linear scale. By *Inverse Transformation* [4], the *uniform* distribution can be generated by a random variable directly.

2.2 Modeling Transfer Size Distribution

It has been analyzed in §1.3 that the distinct transfer size distribution is *heavy-tailed*. The simplest example of a *heavy-tailed* distribution is the doubly-exponential *Pareto* distribution. Its probability density function is:

$$P(x) = \alpha k^{\alpha} x^{-\alpha-1}, \alpha, k > 0, x \geq k$$

The cumulative distribution function for the Pareto distribution is:

$$F(x) = P[X \leq x] = 1 - (k/x)^{\alpha} \tag{1}$$

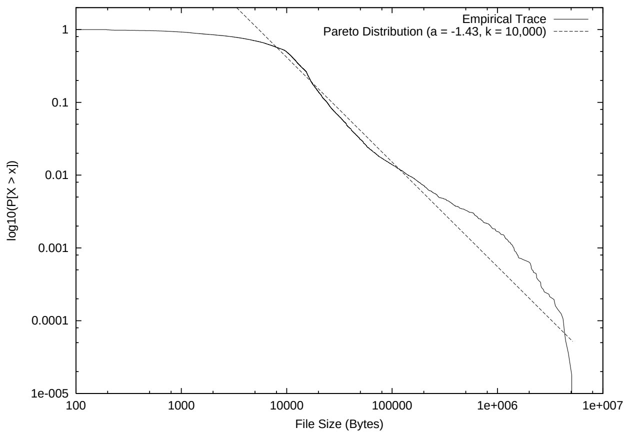
The α parameter is known as the *tail index*, and k defines the beginning of the "tail" of the distribution (i.e. it represents the smallest possible value of the random variable in the heavy-tailed distribution). To estimate the tail index α for the tail part of the distinct transfer size distribution, the approach outlined in [5] is followed. First, a log-log complementary distribution (LLCD) is plotted for the distinct transfer sizes (shown in Figure 6). The x-axis is the transfer sizes in logarithmic scale, and the y-axis is $\log \bar{F}(x) = \log(1 - F(x))$. From Equation 1 we have:

$$\frac{d \log \bar{F}(x)}{d \log x} = -\alpha, x > k$$

Therefore, the tail index α can be estimated by determining the slope of the LLCD plot for values of x greater than k, using least-squares linear regression [4]. From Figure 6, α is estimated to be 1.43 ($k = 10,000$). The coefficient of determination (R^2) is 0.98, indicating a very strong fit. Then, the *Inverse Transformation* is used to generate the Pareto distribution. Suppose u is the random number generated, the tail of the file size is generated by $\frac{1}{u^{\frac{1}{\alpha}}}$.

Pareto distribution models the "heavy-tailed" part of the distinct transfer size distribution (for file sizes larger than 10,000 bytes). For transfer sizes less than 10,000 bytes, Figure 7 shows that the CDF is close to a straight line in the linear-linear scale. Therefore, a *uniform* distribution is employed.

Figure 8 presents the synthetic distinct transfer size distribution versus the empirical one. The x-axis is in logarithmic scale, and the y-axis is the Cumulative Density Function. Figure 8 shows that the shape of the synthetic distribution is close to the empirical distribution.



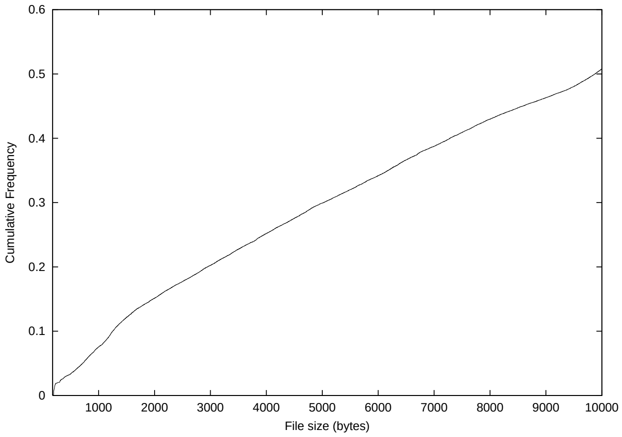


Figure 6: Log-Log Complementary Distribution (LLCD) Plot of Transfer Sizes

Figure 7: Transfer Size Distribution (Size$\leq$10000)

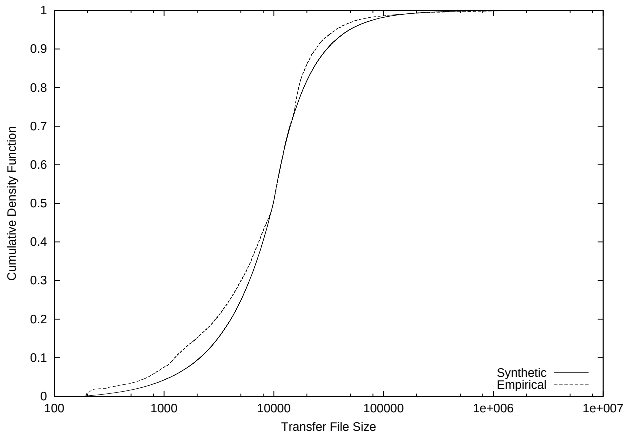


Figure 8: Distinct Transfer Size Distribution (Synthetic vs. Empirical)

2.3 Modeling Document Popularity

Many researchers have noted that Web server document references follow the *Zipf* distribution. *Zipf* distribution states that if items are ranked (r) according to their popularity (P) measured by the frequency of references for individual items, then the popularity of an item is given by:

$$P \sim \frac{k}{r^{\beta}}, k, \beta > 0 \tag{2}$$

where k is the reference count of the most referenced document.

To determine the exponent β, the method in [6] is used. First, all the files are ranked, with the most referenced document being assigned a rank of one, followed by the next most referenced document with a rank of two, and so on. Note that all documents that have been referenced an identical number of times are assigned the same rank. For example, if three different files are referenced 100 times each and are ranked 1000, 1001, and 1002 in the original set, they are assigned one rank number equal to the average of the three ranks. Then, the frequency versus rank plot are drawn (see Figure 9), with both x-axis and y-axis in logarithmic scales. Visual inspection suggests that the distribution follows Zipf distribution for document ranks larger than 15, because this part is close to a straight line. A least-squares linear regression is used to calculate the slope (which is a negative number) of the plot, whose absolute value is equal to the exponent

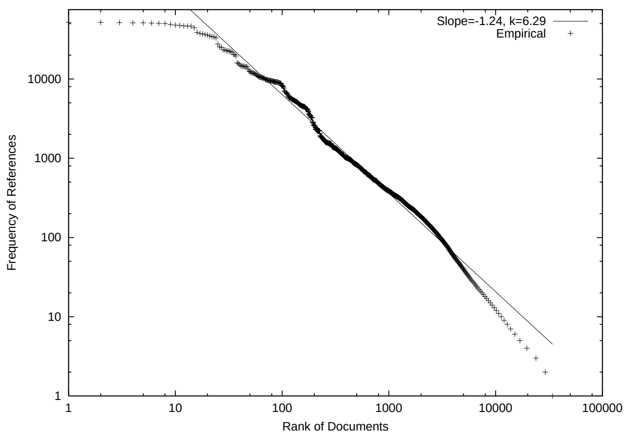


Figure 9: Document Popularity Distribution

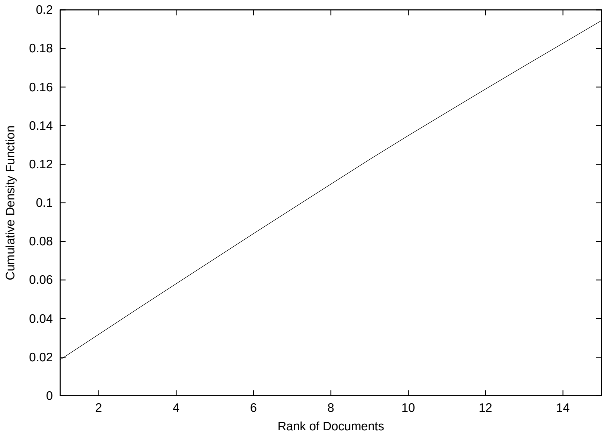


Figure 10: Document Popularity Distribution (Rank 1-15)

β of the Zipf distribution. The estimated β is 1.24, with the coefficient of determination (R^2) equal to 0.99, indicating a very good fit. This theoretical Zipf distribution is depicted in Figure 9 as well. Figure 10 shows that the CDF of document popularity for files ranking from 1 to 15 is very close to a straight line. Therefore, a *uniform* distribution is employed to model the popularity of the first 15 documents. Because the Equation 2 is not a continuous function, the CDF cannot be computed directly from it. To generate the Zipf distribution, the following steps are executed:

1. For each distinct document generate the reference count from Equation 2.
2. Compute the total number of references generated.
3. Compute the reference fraction of each document based on the total number of references. This is the PDF of the document popularity.
4. Compute the CDF from the PDF

Now we have a CDF of the document popularity. The *Inverse Transformation* is used to generate the document popularity. From the above procedure, we can see that the k in Equation 2 does not affect the final result. Only the β dominates the shape of the document popularity distribution.

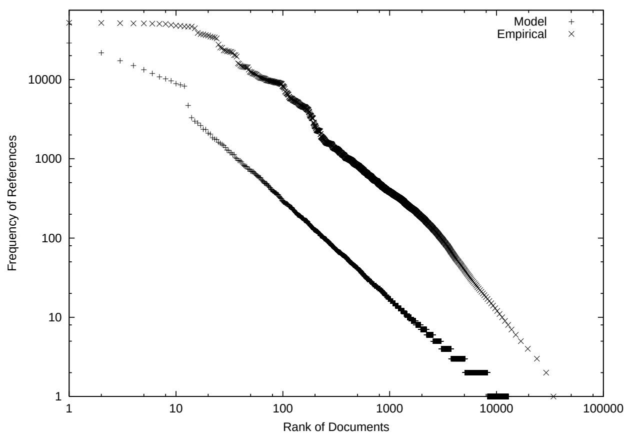


Figure 11: Document Popularity Distribution (Synthetic vs. Empirical)

Figure 11 draws the synthetic document popularity distribution versus the empirical distribution. The shapes of these two distributions are quite similar. However, because the number of synthetic trace is only about $\frac{1}{8}$ of that of the empirical requests, the reference count of each synthetic document is much smaller than that of the empirical document.

2.4 Combining File Size and File Id

In §2.2 and §2.3, we discussed the modeling of file size distribution and the file id distribution, respectively. However, these two are not enough to generate the synthetic trace. The file size distribution is a static property of the documents on the Web server. The document popularity shows the reference pattern of the different files. If the file size and file id are generated separately, one file id may have many different file sizes, which is unrealistic. Therefore, these two fields must be considered together. In Figure 2, we can see that the shape of the file reference count distribution and that of the transfer size distribution is not the same. Further computation shows that the coefficient of correlation between the reference count and the file size is only -0.007. Although the nearly zero value of this parameter does not mean that there is no correlation between these two fields [4], I assume that there is no correlation between them in this assignment. Therefore, the size is assigned randomly to the object id. The algorithm in Figure 12 is used to generate the object id and size. Then reference frequency rank of the object is used as the object id. Figure 13 shows the three size distributions for the synthetic trace. The tranfer size distribution matches the empirical trace well. However, the other two distributions are not consistent with the empirical trace. Therefore, assigning file sizes randomly to documents cannot model the empirical trace well.

```
init table A;
while (not finish)
{
  generate an object id;
  if (this object id already in table A)
  {
    use this size;
  }
  else
  {
    generate a new size for this object;
    insert (object id, size) into table A;
  }
  Output this object id and size;
}
```

Figure 12: Generate Size for Each Object

2.5 Modeling Document Types

All document types present in the trace are modeled. Based on the fraction of each type in the empirical trace, the CDF for the file type is shown in Figure 14.

The types and their percentages in the total requests are listed in Table 6. At first glance the result of the *Compressed* type has very high relative error. A closer examination, however, reveals that this high error is associated with requests that occurred very infrequently. Therefore, this relative big error only has a very small effect to the overall trace. All other types are generated very accurately.

2.6 Modeling Inter-arrival Time

It has been discussed in §1.2 that the inter-arrival time can be considered to follow an Exponential distribution. The Exponential distribution is defined by parameter a, which is equal to the mean and the standard

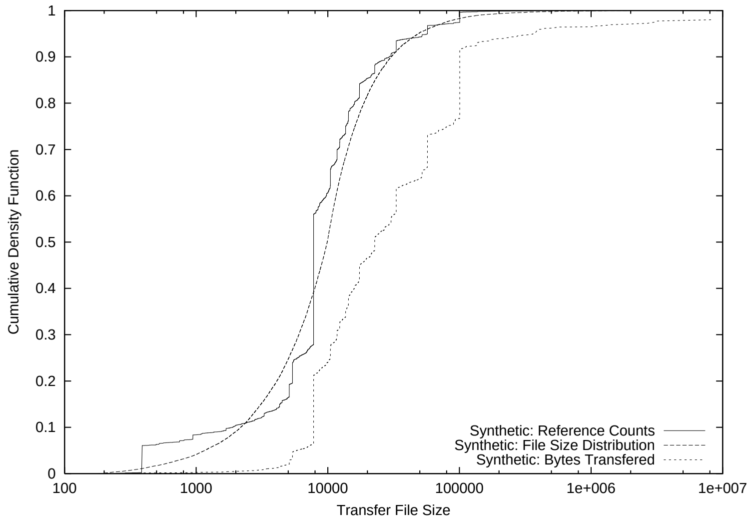


Figure 13: Transfer Size Distributions

$$F(x) = \begin{cases} 48 & \text{(HTML)} & \text{if } 0 & \leq & x & < & 0.1029 \\ 49 & \text{(IMAGE)} & \text{if } 0.1029 & \leq & x & < & 0.9612 \\ 50 & \text{(AUDIO)} & \text{if } 0.9612 & \leq & x & < & 0.96121205 \\ 51 & \text{(VIDEO)} & \text{if } 0.96121205 & \leq & x & < & 0.96128668 \\ 53 & \text{(FORMATTED)} & \text{if } 0.96121205 & \leq & x & < & 0.962158537 \\ 54 & \text{(COMPRESSED)} & \text{if } 0.962158537 & \leq & x & < & 0.9621673892 \\ 55 & \text{(PROGRAMS)} & \text{if } 0.9621673892 & \leq & x & < & 0.962209924 \\ 56 & \text{(DIRECTORY)} & \text{if } 0.962209924 & \leq & x & < & 0.9895888526 \\ 57 & \text{(OTHER\_TYPES)} & \text{if } 0.9895888526 & \leq & x & \leq & 1 \end{cases}$$

Figure 14: CDF of The File Type Distribution

deviation of the data set. In the trace considered, the mean of inter-arrival time is not exactly the same as the standard deviation (see Table 4). In this assignment, the mean value is selected as parameter a to construct an Exponential Distribution. The Exponential Distribution is generated by *Inverse Transformation*. Suppose the u is the random number generated, then $-a \ln u$ is used to generate the inter-arrival time.

The Cumulative Density Functions of the synthetic inter-arrival time versus the empirical inter-arrival time are depicted in Figure 15. As discussed in §1.2, this CDF is computed in the unit of milliseconds. The x-axis (which is the inter-arrival time) then is plotted in logarithmic scale. Figure 15 shows that the synthetic *Exponential* distribution is a good model of the empirical inter-arrival time data set.

2.7 Modeling Client Distribution

Figure 16 shows the Cumulative Density Function of the client reference distribution. It is drawn by ranking the clients based on the number of requests they send. The x-axis and the y-axis are both in logarithmic scales. It is clear that this CDF is close to a straight line. Therefore, it can modeled as follows:

Assume that this line is defined as:

$$\log F(x) = k \log x + b, \text{ where } k \text{ is the slope and } b \text{ is the Y-intercept}$$

Therefore, the Cumulative Density Function (CDF) can be described as:

Type	Fraction of Requests (in percent) Empirical Trace	Synthetic Trace	Relative Error (in percent)
HTML	10.29	10.28	-0.097
Image	85.83	85.80	-0.035
Audio	0.001	0.001	0
Video	0.007	0.007	0
Formatted	0.09	0.09	0
Compressed	0.0009	0.0004	-55.55
Programs	0.004	0.004	0
Directory	2.74	2.77	1.09
Others	1.04	1.04	0

Table 6: Modeling Document Types

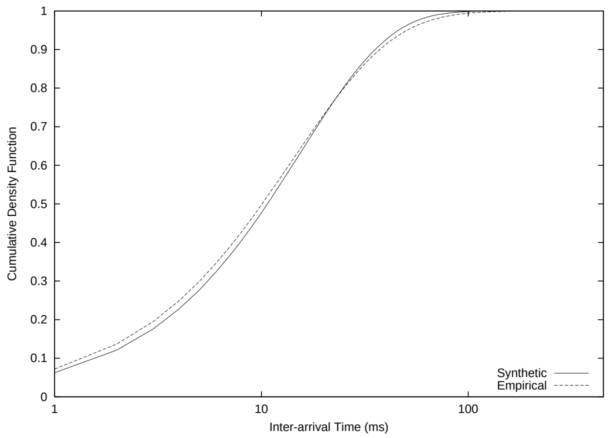


Figure 15: Inter-arrival Time Distribution (Synthetic vs. Empirical)

$$F(x) = b'x^k, \text{ where } b' = 10^b$$

And the Probability Density Function (PDF) is:

$$f(x) = b''x^{k-1}, \text{ where } b'' = 10^b k$$

The least-squares linear regression technique is used to calculate the values of k and b. The results are listed in Table 7.

Figure 17 depicts the Probability Density Functions of the synthetic and the empirical client reference distributions. It can be easily seen from the figure that the synthetic distribution matches well with the empirical distribution.

3 Evaluation of Trace Modeling

Several approaches can be used to validate synthetic models: the *White Box* approach and *Black Box* approach. The *White Box* approach is to compare the empirical trace and the synthetic trace generated by the model in a "side by side" fashion. If the two traces match on key characteristics they could be said to be equivalent. The *Black Box* is to present each of the traces to the computing system and compare the resulting system performance metrics. If they match, the traces can be said to be equivalent.

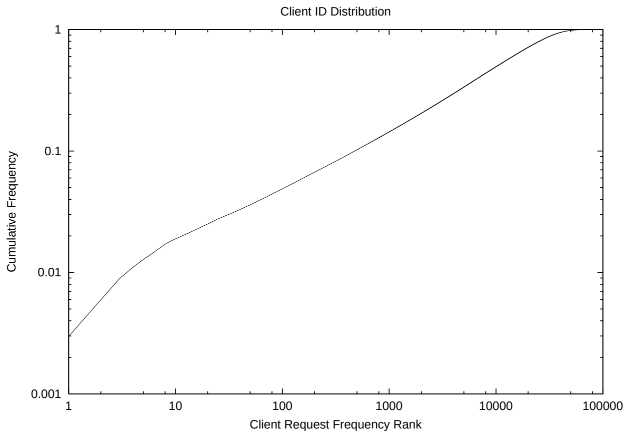


Figure 16: Cumulative Density Function of Client Distribution

Slope(k)	Y-Intercept(b)	R^2
0.47	-2.20	0.98

Table 7: The Estimation of k and b

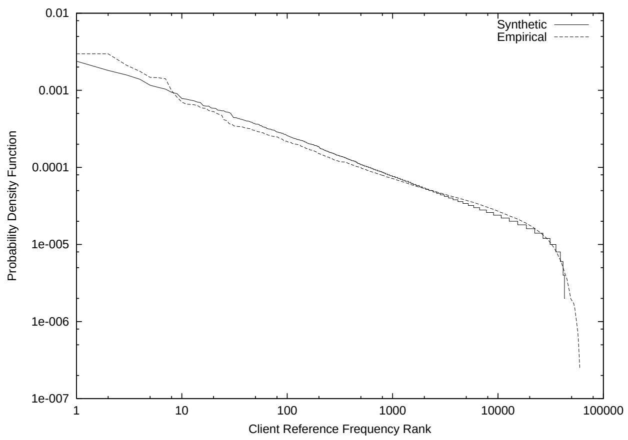


Figure 17: Probability Density Function of Client Reference Distribution (Synthetic vs. Empirical)

3.1 The ***White Box*** approach

In §2, all five fields of the empirical trace are modeled and compared with those of the empirical trace. Therefore, the *White Box* approach is already done. Although the tests in §2 shows that the distributions of the synthetic trace is very similar to that of the empirical trace, there are still some verifications that can be done before we can say that the two traces are equivalent.

1. The locality of the file references should be considered. The distribution of the file references only considers the popularity of each document. However, the locality considers the sequence of the file references. Modeling this property is essential if the synthetic trace will be used in a cache simulator, because different localities may generate very different results in the simulator, even the distributions of the references are the same.
2. The characteristics of different file types (HTML, IMAGE, $\cdots$) may be different. Therefore, both the distributions of size and popularity of each document type should be compared.
3. Although the correlation coefficient between the size and the reference frequency is nearly zero (-0.007), the file reference counts of the synthetic trace does not match the empirical trace well. Maybe there is some kind of correlation that cannot be detected by the *Coefficient of Correlation*.
4. Different clients may generate requests with different characteristics. Therefore, there may be corre-

lation between the client and the reference frequency of documents. This correlation may need to be considered for generating a realistic trace.

3.2 The *Black Box* approach

The *Black Box* approach is not performed in this assignment. For doing this, the synthetic trace and the empirical trace should be inputed to a computing system, say, a cache simulator. Then the outputs are compared. Because the locality of the empirical trace is not considered in the synthetic trace, good match between the synthetic trace and the empirical trace is not expected if they are tested by a cache simulator.

References

[1] Martin Arlitt and Carey L. Williamson. Web server workload characterization: The search for invariants. In *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer SYstems*, volume 24,1 of *ACM SIGMETRICS Performance Evaluation Review*, pages 126–137, New York, May 23–26 1996. ACM Press.

[2] Martin F. Arlitt and Carey L. Williamson. Internet Web servers: workload characterization and performance implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, October 1997.

[3] Roberta R. Bodnarchuk and Richard B. Bunt. A synthetic workload model for a distributed system file server. In *ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, volume 19, San Diego, California, USA, May 1991. ACM.

[4] Raj Jain. *The art of computer systems performance analysis.* John Wiley, New York, 1991.

[5] Anirban Mahanti. Web proxy workload characterisation and modelling, 1999.

[6] C. Roadknight, I. Marshall, and D. Vearer. File popularity characterization. In *proceedings of the 2nd workshop on internet server performance (WISP 99)*, Atlanta, Georgia, USA, May 1999. Available at URL: http://www.cc.gatech.edu/fac/Ellen.Zegura/wisp99/papers/roadknight.ps.