Report

Adam Raphael, Hayden Lummis, Stefan Chen

Initial Input:
Activity Engine and Log files:
Analysis Engine:

Initial Input:

Design decisions for our initial input was based around using structs for events and statistics. They all had their own fields
Event having

- Event name
- CDE
- min
- max
- units
- weight

Stat having:

- Event name
- mean
- standard deviation

We captured these elements using getline functions using ':' and '\n' for delimiters of respective fields and storing them in the correct variable within a struct. These events and stats made up an individual array each (one for stats, one for events) so we could use them as disjoint entities.

Inconsistencies include when the mean is 0 and we have a higher standard deviation which is not possible as there is a min value of 0. Another inconsistency is that the mean must be in the ranges min to max. We checked for these consistencies when we read in events and stats and then compared them to find potential inconsistencies.

Activity Engine and Log files:

1. Steps to generate events approximately consistent with the particular distribution:
    a. Using Gaussian distribution random function to generate daily totals for each event following the given mean and standard deviation we read from files.
    b. To each event, we assign the total of a day to values of instances and values are generated by normal random function specifying the rest quota of day total as a range.
    The generation of values differ between C,D,E events.
    For C events, we generate double values with two two decimal places.
    For D events, we generate value 1 for each event.
    For E events, we generate integer values within range.
    c. In the meanwhile, we generate a time with a value randomly. Then we sort the events of a day ascendantly to shuffle the occurrence order of instances of events.
    d. Finally, we write these records into a log file.

2.
    a. The name of the log file is "complete_logs.txt".

b. The format of log file:

#DAY 1
user:stefan|time:00:05:09|event_name:Emails sent|type:D|value:1|units:time|
user:stefan|time:00:06:51|event_name:Time online|type:C|value:64.22|units:minutes|
user:stefan|time:00:51:43|event_name:Logins|type:D|value:1|units:time|
…....

c. Justification:

The format we use here presents the name of a attribute to achieve human readability. With the delimiter "|" and ":" we can read and split it easily in the analysis engine. What's more, we use "#DAY" to separate events of each day for both human readability and code readability.

Analysis Engine:

The analysis engine can be broken down into three main parts

- Initially read in the overall log file
- Puts all data into useable storage
- Develops statistics based on that stored information using external functions to find mean, std deviation, ect

The daily event totals are stored in the "daily_totals.txt"
The totals are broken down into sections marked by #DAY 1 TOTALS, followed by the day's statistics.
Moreover, the statistics of "live data" are stored in the "current_stats.txt".

#DAY 1 TOTALS
name:Logins|total:1|amount:1|
name:Time online|total:124.89|amount:5|
name:Emails sent|total:36|amount:36|
name:Download volume|total:164596|amount:21|
name:Money made|total:0|amount:0|

Anomalies can occur when reading the file, if a continuous event is in progress, and another instance of that event was to occur during that time period, this would create an anomaly.