Wetpaint Software Engineer Assignment

Revision 1

1. Background

Your assignment is to build a tool for determining the most popular funnel for a website. This tool will consist of a log parser and a web interface for displaying the results. You will have a chance to show off your skills in several areas:

- Back-end, batch processing of large-scale datasets, performant algorithms
- Front-end UI's for displaying and navigating the results of your analysis

Please read the assignment end-to-end before beginning to implement it; there will likely be tradeoffs that you'd like to make for the back-end processing piece based on your approach to the front-end.

2. Part One: Back-end Assignment

We have a tab-delimited file of Date/Time -> User Id -> Url where entries have been ordered by Date/Time (i.e. Web Log).

There will only be limited number of possible different page URLs.

The goal is to determine most popular ordered sequence of consecutive 3 pages visited by the same user within a session.

For example, if the file contains:

1/1/2012 5:21:30 AM	Sally	C
1/1/2012 8:14:58 AM	David	F
1/1/2012 11:59:57 AM	Sally	E
1/1/2012 6:28:52 PM	Sally	A
1/1/2012 9:52:28 PM	Peter	F
1/2/2012 1:04:21 AM	Peter	D
1/2/2012 1:25:28 AM	Peter	A
1/2/2012 10:27:04 AM	David	E
1/2/2012 10:48:35 AM	David	A
1/2/2012 11:53:58 AM	David	C
1/2/2012 10:52:10 PM	Sally	C

Figure 1: Example log file

The winning combination for the example in Figure 1 would be ***E->A->C*** since both Sally and David had it in their visit sequence.

Requirements:

- Programming language is up to you. Feel free to use Java, C/C++, Ruby, PHP,.NET. The main thing is to exercise File I/O to actually be able to read the file and process it efficiently.
- Account for file not being found, being corrupted, etc.
- Memory footprint is a concern.
- Input file can stay open for a long time, but it may be large.
- Don't forget to include tests**.**
- Session detection: if the site has not been visited for 20 minutes by the same user, it is a new session for that user.

Question to answer in the comments:

- Estimate the runtime of this application given that incoming file has N records in it.

3. Part Two: Front-End Assignment

The script that you write in part 1 needs to generate one or more JSON files. These files will be used by the second part of your assignment, a simple web interface for rendering the results of your script. The structure of these files is up to you to design.

Figure 1 contains a mockup of the report webpage:

Figure 2: user interface mockup

Key elements of this user interface:

1. Filter bar at the top that contains:
    - Dropdown with a sorted list of dates that appeared in the log files. Most recent dates on top. There is no “all dates” option.
    - Text box that allows the user to filter the list of paths. If any of the page names matches the string in the textbox, it is to be included. An empty text box means that the filter is to be ignored.
2. A table that includes a sorted list of paths that satisfy the conditions specified in the filter bar.
    - The maximum number of items that are to be shown for any date is 100.
    - The list is to be sorted by the number of occurrences of a given path.

User entry in the filter bar should result in a refresh of the data in the table below.

Other requirements:

- Your web interface portion is to use only browser *(*client-side) technologies – JavaScript, HTML, and CSS. No server-side languages (PHP, ASP.NET, Java, Ruby) are allowed for this part of the assignment.
- Performance of the user interface is important.

3. What Matters

Here are the criteria that we'll evaluate your submission on:

- Functionality: your application should work with many different versions of the input log files.
- Adherence to engineering best practices. The quality of your submission should be on par with what you would have written for a production system.
- Performance. Generating JSON files should be fast. Navigating around your reports application should be quick and user-friendly.
- Avoid "reinventing the wheel" - if there are off-the-shelf components that do the job, we expect you to use them.
- Craftsmanship and cleanliness of your code. Formatting, comments where appropriate, variable names – everything matters.

Just like in real life, if requirements are unclear, feel free to follow up with questions. Good luck!