April 16, 2019

Adaptive Rendering of News Feeds based on Device Statistics

Anonymous

Adaptive Rendering of News Feeds based on Device Statistics

Abstract

The present disclosure describes a client-server system for adaptively rendering news feeds on a social networking platform. The rendering is based on (a) server ranked content feed and (b) device metrics. Core services of a client device read the device metrics (radio signal strength, device temperature, CPU usage, available battery percentage, etc.,) and forward them to a feed render priority selector. The feed render priority selector computes a device ranked priority of different format of news feeds and returns the computed value to a feed render engine. Meanwhile, a front end of the client device sends a query to a social networking server. The query is generated as a result of activities of a user such as watching videos, liking posts, joining groups etc. The social networking server responds to the front end with a generate ranked feeds (referred to as a server ranked feed) based on a news feeds ranking algorithm. The render engine renders a final news feeds based on both the server ranked feed and the device ranked priority.

Problem

Currently, ranking of news feeds on social networking platforms is based on relevance and importance of the news feeds to the user in a reverse chronological order. The ranking does not consider device metrics such as CPU usage, available battery percentage, device temperature, signal strength, etc., for generating the news feeds. Using the metrics along with the user's affinity to the news feeds would provide the user a diversified and better experience.

The present disclosure implements a system that considers server ranked news feeds and the device metrics for generating the news feeds to be displayed on the user's device.

System and working

The present disclosure describes a client-server system for adaptive rendering of a news feeds on a social networking platform based on device (client-side) metrics.

The system, as shown in Figure 1, comprises of:

1. A social networking server
2. A client device

The client device has the following components:

- Core services
- Feed render priority selector
- Feed render engine
- Front end – includes a Graphical User Interface (GUI)

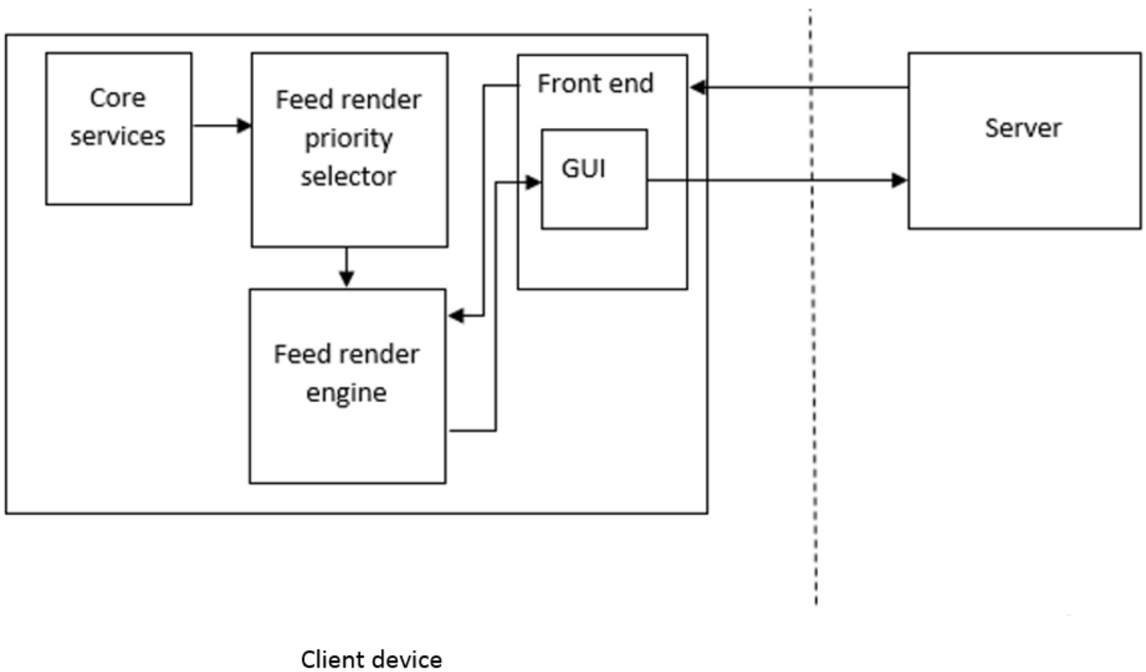

Figure 1: The client-server system for adaptive rendering based on the device metrics

A user of the social networking platform logs in to his/her account using the GUI of the client device. A news feeds generation algorithm at the social networking server generates a content feed and a highlights feed (i.e. the server ranked content feed). The content feed includes stories recently posted by friends or entities connected to the user. The entities may include pages liked, groups joined on the social networking platform by the user. The highlights feed is a subset of the content feed that displays the stories of relevance and importance to the user. The highlights feed is determined based on an activity of the user on the social networking platform. For example, if the user watches a video relating to some sports, the highlights feed displays the videos related to the sports in the highlights feed. The user may manually apply filter(s) to the type of content that he/she wishes to be presented in the highlights feed.

For example, the user may wish to render specific type of the content (say, the videos) from selected friends. These activities are perceived as queries (i.e. feed story queries) by the social networking server initiated from the GUI of the client device. The social networking server sends the highlights feed to the front end of the client device in response to the queries. The front end forwards the highlights feed to the feed render engine. The core services forward the device metrics to the feed render priority selector. The core services are fundamental system services that all applications on the client device use to determine the device metrics (radio signal strength, device temperature, CPU usage, available battery percentage, etc.). The device metrics may also include sensor data received from one or more sensors in the client device. The feed render priority selector determines status of the device metrics, maintains a threshold for each device metrics and calculates values of rendering coefficients (alpha, beta and gamma) for determining a device ranked priority as shown in Figure 2.

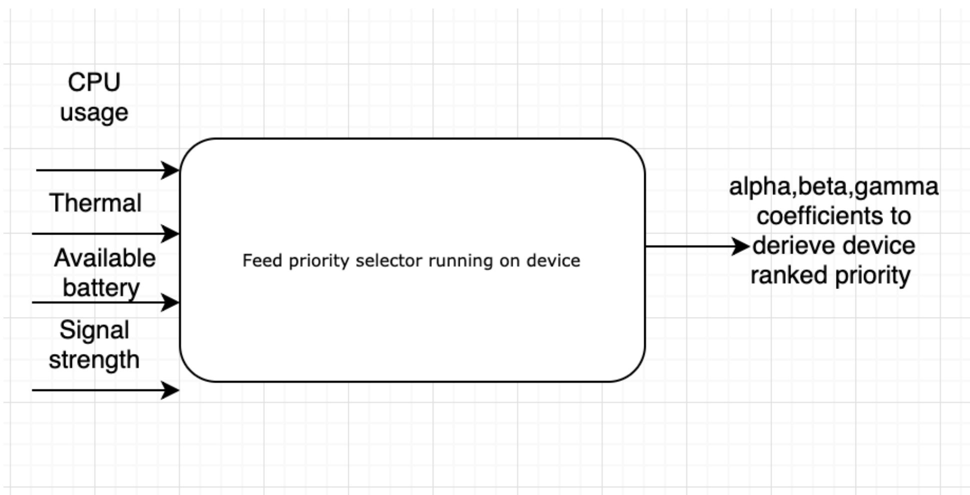

Figure 2: Calculations of the rendering coefficients by the feed render priority selector.

The device ranked priority prioritizes text-based stories, stories with images, and stories with videos. The thresholds are of two types: A lower threshold and a higher threshold. The higher thresholds are defined for the CPU usage and the device temperature. The lower thresholds are defined for the available battery and the signal strength. If the CPU usage and the device temperature are higher than their respective higher thresholds and the available battery percentage and the signal strength are lower than their respective lower thresholds, the rendering coefficients are assigned following values:

alpha = 1, beta = 0, gamma =0

If the CPU usage and the device temperature are lower than their respective higher thresholds and the available battery percentage and the signal strength are higher than their respective lower thresholds, rendering coefficients are assigned following values:

alpha = 1, beta=1, and gamma = 1

For any other status of the device metrics, alpha = 0 or 1 and the values for the beta and gamma varies between 0 and 1 depending on the device metrics.

If the client device is not connected to the social networking server, then alpha = 0, beta = 0, and gamma = 0. This might occur due to loss of internet connectivity.

The feed render priority selector computes the device ranked priority using equation:

Device ranked priority = (alpha*text-based stories) + (beta*stories with image) + (gamma*stories with the videos),

and forwards it to the feed render engine as shown in Figure 3 below.

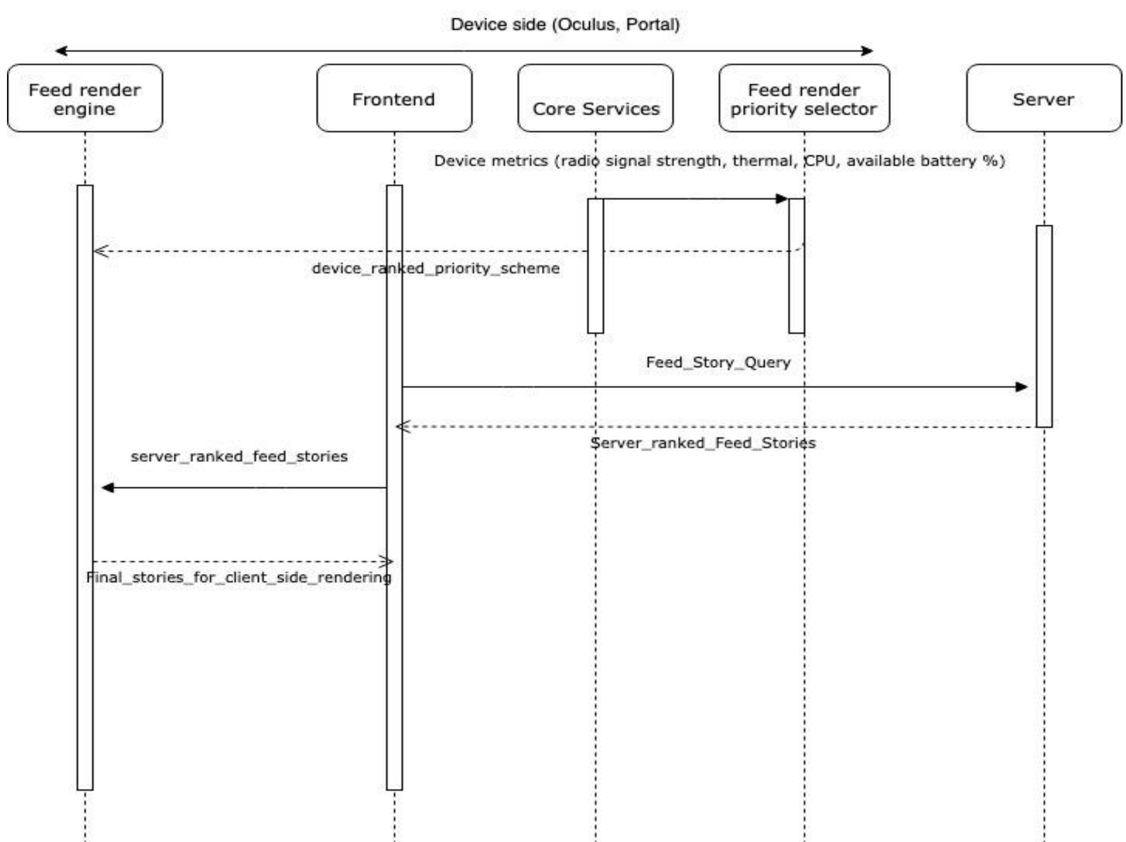

Figure 3: Process flow for adaptive rendering based on the device metrics.

The feed render engine re-orders the highlights feed based on the device ranked priority. For example, the text-based stories will appear in preference to the stories with the image and the videos in the highlights feed after reordering if the alpha has higher value than the beta and the gamma. Finally, the feed render engine renders the news feeds (shown as “Final_stories_for_client_side_rendering” at Frontend in the Figure 3), which is based on both the server ranked content feed and the device ranked priority.

Additional Embodiments

In one embodiment, the user’s experience with a GUI object may be enhanced by dynamically changing an outlook of the GUI object. The GUI objects are virtually implemented physical objects that are constructed of metal or glass. The GUI objects are either embedded in an image format or a video format on the GUI of the client device. In the video format, a location of the GUI object may vary in consecutive frames of the video. The outlook of the GUI object is changed according to the device metrics. Changing the outlook of the GUI object means changing the way it appears to the user on the client device. The user’s experience with the GUI object can be made better if the GUI object interacts with the environment of the client device in a manner like how the physical object would interact. Interaction with the environment comprises interaction with light, motion experienced by the client device, etc. For example, a brushed metal device reflects light in a manner characteristic to that material. Similarly, light interaction can be simulated on a brushed metal GUI object. The brushed metal GUI object is the virtually implemented brushed metal device. The sensors such as a camera or optical proximity sensors on the client device reads a direction and intensity of a light source. The light source is representative of natural light of the environment falling on the client device. The direction and intensity are utilized to simulate the light source shining on a metal surface. Motion data of the client device is obtained using an accelerometer and a gyroscope embedded in the client device. If the sensors indicate no or minimal motion, then it might be unnecessary to update the GUI objects that are rendered based on changes in the motion data. In this case, the GUI object is displayed in a reduced detail mode. The frame is not advanced ahead for the GUI objects in the video format (i.e. a current frame of the video is paused). Thus, the GUI objects in the video format gets converted into the image format. In this case, values of the rendering coefficients are: beta = 1, gamma = 0. In the reduced detail mode, the GUI object may be presented in lower quality. Also, if the sensors indicate a large amount of motion, the user may not be able to perceive a high level of detail of the GUI object. In this case too, the GUI object is displayed in the reduced detail mode and the current frame of the video is paused (if the GUI object in the video format).

If the motion data is within a predetermined range, then the outlook of the GUI object need to be rendered in real time for the client device.

In another embodiment, if battery life of the client device is low or processor usage is very high (as determined by the core services of the client device), then again, the GUI object(s) can be rendered in the reduced detail mode and the current frame for the video of the GUI object is paused (if the GUI object is in the video format). The values of the rendering coefficients for the GUI object in the video format becomes: beta = 1, gamma =0. In this scenario, the light interaction can be simulated with a fixed light source and directional changes are calculated based on orientation of the client device as sensed by the accelerometer or the gyroscope. In some critical cases, a frame rate of the GUI object may also be reduced for reducing an activity of the client device. This help is reducing the processor usage and enhancing the battery life by reducing activities of the client device.

In another embodiment, the device metrics may include Wi-Fi connectivity status of the client device. If the client device is found to be connected to a Wi-Fi network and the value of the gamma is higher than the alpha and beta for including the videos in the news feeds, then the feed render engine may automatically play (auto-play) the videos for the user. However, the user may turn off the auto-play of the videos using settings of the client device.

In yet another embodiment, information provided to the client device by the social networking server is adapted based on display characteristics of the client device. A software installed on the client device provides a user agent string to the social networking server. The user agent string includes a device model, an operating system (OS) and version of the OS, a browser type and a version of the browser in the client device. The social networking server searches for the device model in a look-up table server. The look-up table server determines and sends display characteristics to the social networking server. Examples of the display characteristics include supported multimedia format, supported JavaScript, whether the device model supports MMS messaging, etc. The social networking server selects a page profile from a set of stored page profiles. The page profile represents a combination of an HTML version and a style sheet version. The page profile is selected so that the display characteristics are closely matched with profile characteristics of the page profile. A page with the said page profile is rendered for the client device.

Conclusion

Device characteristics or device metrics vary with time and with the device. Since type of device and frequency of usage vary with users, it is pertinent that displayed content is modified basis device metrics of the user. Adaptive rendering based on behavior and/or profile of the user has been referred to in previous arts. In the present disclosure, adaptive rendering for a resource-constrained device has been taken into consideration. Rendering the content that can dynamically adapt with specifications or the metrics of the device can be very useful to the user. There can be multiple metrics that carry potential to be leveraged for adaptive rendering.