Cameron Thorp

Module 6: Memory and Storage Management

CS230 Operating Platforms

Memory and Storage Management Analysis

For memory management, it would be ideal to utilize an organized file system to access the images via microservices for efficient access by users. Within memory, the images could be randomly determined prior to starting the game, and loaded individually by the microservice prior to rendering the image. This would ensure that all users within a single game have a shared experience independent of their relative connection speeds. The utilization of microservices also allows for increased efficiency as well as easier management for cross-platform game instances. Because the middleware of the microservices is responsible for interfacing with the client’s operating platform, the server doesn’t have to change protocols or methods for transferring the image binaries.

The storage component would be primarily based on the server-side. Because Draw It or Lose It is a web based game, it would be inefficient and detrimental to the user experience to attempt and store the high-resolution images on each client-side machine. 200 8MB images is over 1.5GB of overall data, most of which will not be accessed in a single game of Draw It or Lose It. Additionally, this data shouldn’t exist on the user’s machine once the instance of the game is over, therefore keeping the image binaries on the server-side is the optimal solution, only transferring the data for the randomized images which are pre-determined when the game instance is created.

An effective combination of memory and storage management is key for resource expenditure as well as for user experience. It couldn’t be guaranteed that the images would load at the same rate over different remote connections unless they are local, but it would be very costly to load all of the images from the server to storage prior to a game beginning. This is why server-side storage is ideal for maintaining a file system with the images, which can then be accessed by a microservice and transferred to users independently based on their local operating platforms. The end state of this transfer being that users have reliable and consistent access to the data required to participate in a game; without needing to download all of the source files, without needing to share a universal operating platform, and without needing a high-speed network connection.

Sources:

Arsov, K., “Microservices vs. SOA - Is There Any Difference At All?”, Medium, 10 Nov. 2017, https://medium.com/microtica/microservices-vs-soa-is-there-any-difference-at-all-2a1e3b66e1be

Wagner, G., Gal, A., Wimmer, C., Eich, B., Franz, M., “Compartmental Memory Management in a Modern Web Browser”, ACM Digital Library, 04 Jun. 2011, https://dl.acm.org/doi/10.1145/1993478.1993496