Nicholas Kreuziger

CS-230

August 8th, 2022

**Storage & Memory for Draw It or Lose It**

The game “Draw It or Lose It” consists of four rounds of play lasting one minute each. Drawings are rendered at a steady rate and are fully complete at the 30-second mark. The game “Draw It or Lose It” will cycle through 200 high-definition image files at 8 megabytes in size a piece. In total that is 1600 megabytes of image data that will have to be stored and accessed via memory at a framerate that doesn’t interrupt user experience. At minimum, 24 FPS is required for distinct frames to appear as one moving image and anything higher than this will increase display quality and viewer experience (HP, 2018). Per the client, “The application will render images from a large library of stock drawings as clues.”, with the demands of the gameplay in mind we must also multiply those demands by the number of active games running on the platform. If we were to relate this to short- and long-term memory, the short-term memory is the memory and the long-term memory is storage. A comprehensive plan for the storage and memory will set “Draw it or Lose it” up for success.

**What considerations and specific approaches would it take to ensure that memory is effectively managed in the software application, Draw It or Lose It?**

The success of the game will depend upon the game performing smoothly for the end user. Accurately predicting the Memory needs of the application will be the first exercise to achieve smooth performance. Random Access Memory (RAM) is a repository of data loaded for quick access by the CPU, the tradeoff for that performance is expense (GeeksforGeeks, 2020). Ensuring that the application has adequate RAM to provide the 24 FPS minimum will create a better user experience. The requirements for Memory can be passed onto our own server or the Client. An additional consideration is that virtualization would allow a task in memory to be stored in storage to be retrieved later at the expense of processing time (*Memory & Storage: Different Tasks, Different Technologies - CHM Revolution*, n.d.). Questions that need to be answered are as follows.

* Will the bulk of image processing be done on the client’s computer or the server?
* What is the impact to user experience if virtualization is heavily utilized?
* What is the memory requirement per game session to ensure stable user experience?
* What is our maximum capacity of active games before the client experience suffers?
* Based upon current user engagement, what memory capacity will be needed at launch?
* What will be the anticipated cost and time of scaling the solution upwards when demand for game sessions increases?

Answering these questions will involve determining the throughput of loading the images in-game, gauging the performance impacts of virtualization, assessing the memory requirements per active game session, and determining the cost per each live game session on hardware resources. Analyzing our memory needs at launch based upon current user engagement is also recommended to ensure a smooth launch. Additionally, it would be prudent to predict the costs to scale these memory needs as user engagement grows.

**What considerations and specific approaches would you take to determine how much storage is needed and how to manage storage for your client’s application, Draw It or Lose It?**

Accounting for the long-term memory of our application is equally important. If a user’s information is not properly stored, they will likely abandon interaction with our product. The storage will be used to store the images the game loads in addition to the information of game sessions, teams, and users. Questions that need to be answered are as follows.

* Will the game images be stored on the client’s computer or the server?
* Will the game, team and user information be stored on the client’s computer or the server?
* What are the performance impacts of the storage location?
* How much storage is required for each user, team, and game session?
* How many users, teams and game sessions can be stored at maximum?
* How much storage do we anticipate needing at launch?
* What is the cost and time of increasing storage as demand increases?

Answering these questions will involve determining the cost of in-house storage hardware vs server storage, the storage size of images, game session, team, and individual user information. Analysis of the current number of users should also be considered for storage needs at launch. Calculating storage costs per user can help determine the cost to scale our storage as application engagement increases.

**What are the differences in how memory and storage are used in terms of the game application functionality?**

Game Application functionality will be affected by memory and storage differently. Memory will affect the speed that the data is retrieved. Storage will impact the amount of user data that can be stored and how fast it can be served to the memory. Additionally, the utilization of these two resources and how heavily either/or is utilized in the execution of the game can greatly affect the cost of implementation and the end user’s experience. A plan to balance the storage and memory needs of our application will aid us in future expansion.

Citations

GeeksforGeeks. (2020, March 25). *Difference between Memory and Storage*. Retrieved August 3, 2022, from <https://www.geeksforgeeks.org/difference-between-memory-and-storage/>

HP. (2018, November 20). *What is Frame Rate and Why is it Important to PC Gaming?* Retrieved August 3, 2022, from <https://www.hp.com/us-en/shop/tech-takes/what-is-frame-rate>

*Memory & Storage: Different Tasks, Different Technologies - CHM Revolution*. (n.d.). Computer History Museum. Retrieved August 3, 2022, from <https://www.computerhistory.org/revolution/memory-storage/8/249>