**6-1 Assignment: Memory and Storage Management**

Cristiano Miranda

Southern New Hampshire University

CS-230: Operating Platforms

Professor Sarkar

12 October 2025

**6-1 Assignment: Memory and Storage Management**

            The Gaming Room's application, Draw It or Lose It, will utilize a stock library of images that will need to be displayed to multiple users. Additionally, since multiple games will be running simultaneously, the server will need to handle sending different images to multiple users concurrently. The stock image library contains 200 high-definition images, each around 8 megabytes, making the total storage requirement for the images approximately 1.6 gigabytes: 200 x 8MB = 1600MB x 0.001GB = 1.6 GB. While this size requirement is not of major concern, as most hardware utilizes terabytes to measure its maximum storage capabilities, which is 1000 gigabytes, an issue is presented in the way the application is being developed. While web applications can store data on a user's system using local storage or other methods, this storage is limited to string types, which requires additional processing power to convert image files into a storable string. These storage types are often not reliable enough to properly store images in the long run, requiring images to be downloaded every time the user connects to the web application and to be held in the user's system memory rather than storage. Even though most devices have a large storage capacity, memory is still relatively limited, especially in mobile devices, where some have a total of 4 GB of RAM. Additionally, requiring users to request 1.6GB of photos from the server when accessing the web application and waiting for it to complete before joining a game would create a poor user experience. Furthermore, if multiple users attempt to download the extensive image collection simultaneously, it can easily overload the server.

            For this application, the user memory requirement should be relatively small to avoid limiting the number of compatible devices that can access the web application and to ensure a good user experience. Thus, requiring a device to have an additional 1.6 GB of memory is problematic. Instead, having the server deliver a clue at the start of each round will address this potential problem by reducing the need for the user to store all 1.6 GB of photos in their main memory upon accessing the application. By limiting clients to handling a single clue per round, users will only need to request and store an 8 MB file instead of downloading 1.6 GB of files at once. This solution eliminates the need for users to wait for files to download initially, allowing them to instantly join a game upon creating their profiles, resulting in a better user experience. By developing the web application to remove the previous clue from memory when a new clue is requested from the server, the user will never have to worry about accumulating excess memory dedicated to holding old, unused clues. One of the main benefits of creating this as a web application is that it avoids the need to consider potential differences between operating systems. Although they may use different web browsers, these browsers handle memory in virtually the same way. Thus, instead of designing specific logic for certain operating systems to handle memory management, the same process can be used on almost every operating system. However, for the server, memory will be crucial because multiple games could run concurrently with a large number of users, and each game might require different images. The server requires careful consideration for memory, as running multiple concurrent games necessitates sufficient physical memory to prevent the web application from slowing down. Even though most server operating systems can utilize virtual memory to allow the web application to load data beyond the capacity of physical memory by temporarily moving currently unused data to storage, this comes at the cost of additional resource usage and slower operation. As Lenovo states on its website, one of the significant differences between storage and memory is that "memory has faster access times than storage does…" (2021). Even though more games are allowed to run simultaneously, these games would greatly slow down. The server would have to constantly swap games between storage and memory, leading to a poor user experience, as games would be put on hold or paused. Thus, a solution to this problem is to limit the number of active games so that this variable can be controlled and the required memory can be planned for. Additionally, if the client desires to scale the number of games, increasing the server's physical memory should be a feasible option when determining the server components. In the end, the client-server pattern allows most of the processing to be done on the server side. However, considerations still need to be made for users, and the memory requirements of user systems should be minimized to ensure wider compatibility with lower-end devices. Still, the server should be configured with sufficient memory to load a predetermined number of games, limiting the need for virtual memory and avoiding additional delays from swapping data between memory and storage.

            Hosting the web application on a server and allowing users to connect to it removes the need to store any data or information on users' systems. By utilizing the client-server architecture, the web application can use the server to store all webpages and application data without relying on the client's system. While there are instances where specific data would benefit from being stored on a client system, such as setting preferences, web applications have a way to store simple data for convenient access. Thus, storage considerations do not need to be made for the user, as their system will likely not require storing any data. If some data is stored, it will be minimal in size. Regarding the server, since other data, besides the 200 clues, will need to be stored and encrypted for user security, it is crucial to ensure the server has quick access and ample storage. Right away, there are some known requirements, such as being able to hold all 200 clues, which is around 1.6 GB, along with the data for the web application. The data can vary as the application grows, but the initial version serves as a good starting point for determining the storage requirement. However, user data will be the primary source of conflict. While we can account for the amount of data stored for each user, it is more challenging to determine how many users will access the web application and store their information on the server. To avoid any issues of running out of storage, the server should utilize a storage system that can be scaled in the future if storage space begins to run out. One such storage system is a database, which, while primarily benefiting from scalability, also allows quick access to the information housed within it. As Abraham Silberschatz et al. explain, storage systems that utilize direct access are beneficial as they allow for "immediate access to large amounts of information" (Silberschatz et al. 2008). While storage mediums like hard drives or solid-state drives are beneficial for storing items indefinitely, they do require more time to access this information. Considering that multiple users could potentially attempt to access their information stored on the server, it is crucial that the storage medium can retrieve this information relatively quickly and in an organized manner. Overall, while storage considerations do not need to be made for the client systems as they will likely store minimal to no data, the server must handle storing essential files like the web application, the stock image library, and user data while still allowing these files and information to be quickly accessed and served to multiple users concurrently.

            In this scenario, memory will be used to store all information that needs to be accessed quickly and in multiple instances. Game data of the currently running games, including clues in use and information for various teams and users, along with the application itself, would benefit from being held in memory. By selecting the number of games that can be running concurrently, it will be possible to determine the amount of physical memory that should be added to the server to ensure that enough physical memory is available so that all games are running at the same speeds to avoid any delay caused by swapping between virtual memory and physical memory to create the best user experience possible. Still, storage is necessary because specific data, such as user data, must be stored indefinitely and accessed only in certain situations. Additionally, storage will be required to store the stock image library, as unused clues can be housed in storage, while currently used clues can be loaded in memory for faster access. Storage is beneficial for holding data in the long term and ensuring that data generated by users and the game is not lost. Memory is volatile, and any information within it is lost in the event of a server shutdown or reset. Overall, the client's computer does not need to be considered for storage and memory requirements, as most of the web application processing and data storage will be performed on the server. In the case of the server, though, it will be beneficial to have an appropriate amount of physical memory to allow the predetermined number of games to run concurrently to ensure that all users, regardless of what game they are involved with, have the same experience and are not delayed due to the wait caused by swapped data from memory to storage. Storage is another essential factor that we can plan for initially, but some uncertainties cannot be accounted for. However, by using a database, we can scale the storage in the future if available storage begins to run out.

References

*Silberschatz, A., Galvin, P. B., & Gagne, G. (2008). Operating System Concepts, 8th Edition.*

*John Wiley & Sons.* [*https://learning.oreilly.com/library/view/operating-system-concepts/9780470128725/*](https://learning.oreilly.com/library/view/operating-system-concepts/9780470128725/)

*Computer Memory Vs Storage: Computer Memory | Memory vs Storage | Lenovo US. (2021).*

*Lenovo.com.* [*https://www.lenovo.com/us/en/glossary/memory-vs-storage/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOoqRHL\_nmm6lRIZynxuu\_E9p3d8VGK2KVAYYNtKILep0XWMhbGLI*](https://www.lenovo.com/us/en/glossary/memory-vs-storage/?orgRef=https%253A%252F%252Fwww.google.com%252F&srsltid=AfmBOoqRHL_nmm6lRIZynxuu_E9p3d8VGK2KVAYYNtKILep0XWMhbGLI)

*‌*

‌