Module 6: Memory and Storage Management

Meghan Walsh

The Gaming Room's Draw It or Lose game is a multi-user game application that will require in-depth consideration of how storage and memory are managed in each platform and how management differs.

Given the Draw It or Lose game requires pictures that will need to be rendered at a steady and rapid rate, developers will need to consider how memory is effectively managed in the software application in order to render the pictures in a minimally memory-intensive way. Additionally, running lists of players, teams, and games in current play must be constantly updated, and thus the memory requirements of this constantly-updating database must be managed appropriately. For web applications, memory management is crucial for handling these constantly running lists and image rendering across network conditions. In order to minimize memory usage throughout gameplay, we could consider using hardware acceleration, which offloads some of the memory burden from the CPU to other, more specialized hardware components, like GPUs. Given GPUs are built to accelerate graphics rendering, this would be a great way to manage image-rendering-based-memory more efficiently. However, not all devices that access web applications have GPUs. A large area of consideration in the image-rendering process is file operations, which oftentimes is a memory-intensive process (Silberschatz et al 2009). To render each image, we will need to select, load into memory, and read each image file, all of which can be memory intensive (Silberschatz et al 2009). To avoid the memory-intensive process of constantly loading each selected file, we could load images into memory at the start of the game. Given the memory requirements of the game and the hardware that could manage it (like GPUs), I suggest offloading memory management to remote servers, which have the memory requirements and hardware we can use to optimize memory management. Both Windows and Linux have notable servers with options that our developers could choose from, based on what works best for our application (OpenSource.com 2018, Microsoft, n.d.). Windows' 2025 server VMs, for example, boast 240 TB of RAM, 2048 virtual processors, and GPU partitioning, all of which would greatly help with our games image-rendering and gameplay data memory needs (Microsoft, n.d.). Another approach that game developers could consider is using a serverless platform. Serverless platforms allow for a cloud provider to dynamically manage the allocation and provisioning of servers (Bashir 2019) . In the context of our game, serverless platforms can automatically allocate and deallocate the memory we need in order to run the game without lag. Via HTTP requests using REST APIs, we can offload memory requirements to our serverless functions and thus allow for no local resources to be used on any game user's host device (Gupta 2025, Bashir 2019). As a result, the game can run rapidly and effectively on all operating platforms because users' own devices, with varying operating platforms, memory requirements, and hardware, are not responsible for managing the game's memory allocation.

To determine how much storage is needed for the Gaming Room's application, we must consider what data we must store. The game application will have 200 high-definition image files to choose from, each one approximately 8 megabytes in size, thus 1600 megabytes total needed. Additionally, a database of all players, teams, and games ever played must be stored as historical records, which must be factored into how much storage is needed. Given the game will handle multiple game instances at the same time, we will need to account for the high storage needs of many games, each with their own player data, teams, scores, etc., all running simultaneously, and thus pulling from and updating our data in storage. All of the above will need to be considered to determine how much storage is needed. Developers working on this game must have in-depth understanding of considerations and specific approaches it would take to determine how much storage is needed in the context of this game and how to manage storage for the Gaming Room's application. A computer's storage space and capabilities, along with the level of abstraction enacted upon it by the operating system, is variable across the many machines a user could use to access our web application. Directing the web application's storage requirements to remote servers will allow us to create a storage system and environment that fits our storage needs. For example, Windows Server 2025's improved database storage capabilities will allow us to store our historical database of all players, teams, and games. Windows Server 2025 can manage our large library of high-definition image files through features like Resilient File System, a high-performance file system introduced by Microsoft, to allow us to utilize scalable storage (Microsoft, n.d.). Given that we are creating a web application for the Gaming Room, we should take note that web applications are highly network-dependent. Utilizing Windows Server 2025's enhanced networking capabilities, including Network ATC and Network HUD, will allow our game application to improve network performance, traffic management, and efficient data transfer to ensure fast, lag-free delivery of images during gameplay (Microsoft, n.d.).

Across all cloud or server options, we must consider pricing. Cloud service providers, for example, offer many different cloud pricing points depending on our compute, storage, database, analytics, application and deployment requirements (Chapel 2019). For example, AWS has traditionally been viewed as the most cost-effective option, but Azure and GCP have made significant strides in pricing and innovation to close the gap (Chapel 2019). For our needs, choosing a service provider or platform will mean balancing price with the storage and memory capabilities needed for web applications needs.

The memory management aspects of our game's development and the storage management aspects of our game's development have different considerations, approaches, and requirements. While the two are related, there are important differences in how memory and storage are used in terms of the game application functionality. In our game application, we focus on RAM, Random Access Memory, to store temporary data, like which picture is being shown at the current time in a current game, or which teams or players are in a current game. Storage, however, is long-term. Computers can store information on various storage media, such as magnetic disks, magnetic tapes, and optical disks (Silberschatz et al 2009). The operating system's role in storage management is to provide a logical view of information and abstract from the physical storage hardware to define a storage unit called a file (Silberschatz et al 2009). While our image files or gameplay information are loaded into memory during the game's runtime, they must be stored long-term as files on a storage media. The process of storing our image file, from the disk, into game memory involves reading from the file a system call that specifies the file name and where in memory to place its next block (Silberschatz et al 2009).

Citations

Bashir, F. (2019, July 5). *What is Serverless Architecture? What are its Pros and Cons?* freeCodeCamp. Retrieved April 2, 2025, from https://www.freecodecamp.org/news/what-is-serverless-architecture-what-are-its-pros-and-cons/

Chapel, J. (2019, July 1). *Cloud storage cost comparison: AWS vs. Azure vs. google*.

Medium. https://jaychapel.medium.com/cloud-storage-cost-comparison-aws-vs-azure-vs-google-844dfff3d324

Gupta, L. (2025, April 1). *What is REST?: REST API Tutorial*. REST API Tutorial. https://restfulapi.net/

OpenSource.com. (2018, May 31). *What is a linux server and Why does your business need one?*. Linux.com. https://www.linux.com/news/what-linux-server-and-why-does-your-business-need-one-0/

Silberschatz, A., Galvin, P. B., & Gagne, G. (2009). Operating System Concepts, 8th Edition. In *O'Reilly Online Learning*. https://learning.oreilly.com/library/view/operating-system-concepts/9780470128725/

Windows Server Operating System. Microsoft. (n.d.). https://www.microsoft.com/en-us/windows-server#overview