**Memory Management:**

As we know, memory is critical part of every operating system and we shall utilize it very efficiently. We will discuss memory management in different operating system for the scenario described in the document.

**Android:**

Images comes in various sizes. Some are high resolution and some are low resolution. As we know that our images are of 8 megabytes then suppose if the memory does not support 8MB images and we try to show it, it won’t make any difference as memory capacity is less. In this case, to save memory, we can reduce images to low resolution so that they can be easily displayed as well as transferred on the operating system.

To effectively render images there are different things that needs to be taken into account.

The memory allocation that takes place varies with different configurations. If your image does not have alpha, I would recommend that you use the **RGB\_565** config. It will give you better quality than **ARGB\_4444** because it is using the bits of the alpha channel to store more data for colors. If your image has alpha, go for **ARGB\_4444**. In normal circumstances with these configurations, you are not going to see bad quality images. If you need high quality images and you cannot afford to lose pixels, go for **ARGB\_8888**.

For high quality images, Picasso library is used due to its lot of benefits.

It transform images to better fit into layouts and to reduce memory size.

For most cases, it is recommend that you use the [Glide](https://github.com/bumptech/glide) library to fetch, decode, and display bitmaps in your app. Glide abstracts out most of the complexity in handling images rendering efficiently and other tasks related to working with bitmaps and other images on Android

**Windows:**

As we want to render large amount of files constantly, so windows uses a [swap file](https://kb.iu.edu/d/aagb), also called a page file, and is an area on the hard drive used for temporary storage of information. Windows uses the swap file to improve performance. A computer normally uses primary memory, or RAM, to store information used for current operations, but the swap file serves as additional memory available to hold additional data. Therefore, to display images, images will be stored in swapfile.Then in order to show next image, images will be taken from swap file instead of disk so access to the images is speeded up. This was memory will be managed and files will be transferred in windows platform.

Speaking of generally in terms of speeding image loading,many different things are to be taken care of.After extensive research the most important factors to speed up the image rendering are follows.

**1) Compress image before loading:**

Never upload non-compressed, or rescaled, images to your website. Even if the rendered image is only 100 x 100 pixels, if the uploaded file is 1000 x 1000 pixels, the server must send 10 times the amount of data than necessary.

**2) Images dimensions must be of same size**

Suppose you want to render large amount of images, now the image sizes shall be the same to make it consistent and it takes very less time to load and transfer such images. The less open-ended commands within your code, the faster the page loads.

This point refers to all the operating systems including windows, linux, mac,android so there is some **comparison** as well and the importance of consistent data to speed up whole process.

**Storage Management:**

Storage management is another very important aspect when it comes to performance of overall application. Different types of storages are present in different operating system. Let’s discuss them

**Android**

There are many different storage mediums provided by android to store data depending on your app need.Few of them are follows

**App-specific storage:**  Store files that are meant for your app's use only, either in dedicated directories within an internal storage volume or different dedicated directories within external storage

**Shared storage:** Store files that your app intends to share with other apps, including media, documents, and other files.

**Preferences:** Store private, primitive data in key-value pairs

**Databases:** Store structured data in a private database using the Room persistence library

So keeping in view of the images rendering and showing large images one by one **App-specific storage is most suited. Because it is also secure and data is secured and cannot be accessed by other applications so this storage system in which dedicated directory within internal storage is used.**

**Windows:**

**Like android, there are different storage sources in windows operating system as well.Few of them are**

**Folder Redirection:**

**The major advantage of this is that for even a local folder a path is present which is accessed from server and local folder can be assessed from any computer.**

**Offline Files:**

makes network files available to a user, even if the network connection to the server is unavailable or slow. When working online, file access performance is at the speed of the network and server. When working offline, files are retrieved from the Offline Files folder at local access speeds.

Now the advantage is that even if the connection to server is not made,files i:e images will be rendered without much lag and this is what we require.

So in this case offline files storage will be used as it is in our best interest and it would be very helpful for quick rendering of images in which access required is quick.

Same is the case with other operating systems like Mac and linux.

**Compare and Summarize**

So to summarize and compare different operating systems,we can say that each of the operating system have its own mechanism of memory and storage management and differ vastly with each other.With our requirement i have analyzed different memory techniques as well as storage requirements to handle the given scenario.