

Q). Difference between On-heap and Off-heap memory in Spark.

Onheap memory refers to the memory allocated on the Java heap, which is the memory space managed by the JVM for storing Java objects. Onheap memory is allocated by the JVM's garbage collector and is subject to the limitations of the heap size.

Offheap memory, on the other hand, refers to the memory that is allocated outside the Java heap, typically by the operating system. Offheap memory is not managed by the JVM's garbage collector, which means that it does not contribute to the Java heap size limit. However, since offheap memory is not managed by the JVM, it is the responsibility of the developer to allocate and deallocate this memory explicitly.

Peak onheap memory refers to the maximum amount of memory used by a Java application on the heap during its lifetime. This includes all the objects that have been allocated on the heap, including objects that have already been garbage collected.

Peak offheap memory, on the other hand, refers to the maximum amount of memory used by a Java application outside the heap during its lifetime. This includes memory allocated by the application using direct byte buffers or other offheap memory allocation techniques.

In summary, the main difference between peak onheap and offheap memory is the memory space they use. Onheap memory is allocated on the JVM's heap and is managed by the garbage collector, while offheap memory is allocated outside the heap and is not managed by the garbage collector.