Nginx Buffers

Table of contents :

Introduction :

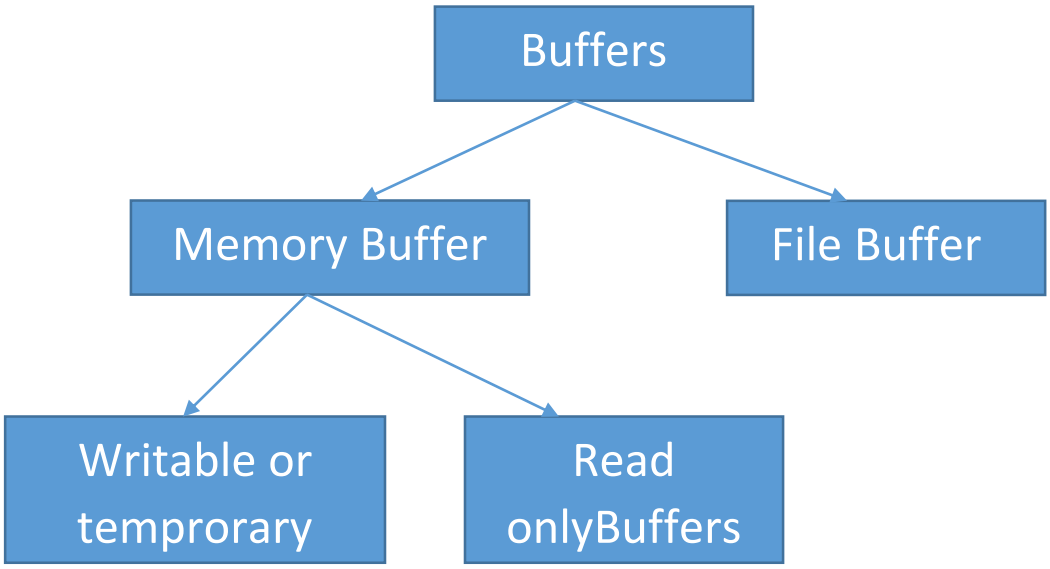
In nginx, Buffers are used to track the progress of reception, transmission and processing of data. They use the concept of windows to keep track of data send, processed and received.

The concept of Buffers is implemented in nginx using these two files :

- ngx_buf.c -> Present in src/core/ directory
- ngx_buf.h -> Present in src/core/ directory

Buffer Structure :

The buffer Structure is declared in ngx_buf.h file. The buffers in nginx can be classified as :



The Buffer data structure is defined as shown :

```
typedef struct ngx_buf_s  ngx_buf_t;
struct ngx_buf_s {
    u_char          *pos;
    u_char          *last;
    off_t           file_pos;
    off_t           file_last;

    u_char          *start;         /* start of buffer */
    u_char          *end;           /* end of buffer */
    ngx_buf_tag_t    tag;
    ngx_file_t      *file;
    ngx_buf_t       *shadow;


    /* the buf's content could be changed */
    unsigned         temporary:1;
```

```
    /*
     * the buf's content is in a memory cache or in a read only memory
     * and must not be changed
     */
    unsigned         memory:1;

    /* the buf's content is mmap()ed and must not be changed */
    unsigned         mmap:1;

    unsigned         recycled:1;
    unsigned         in_file:1;
    unsigned         flush:1;
    unsigned         sync:1;
    unsigned         last_buf:1;
    unsigned         last_in_chain:1;

    unsigned         last_shadow:1;
    unsigned         temp_file:1;

    /* STUB */ int   num;
};
```

The buffer structure contains many fields.To implement different type of buffers we need to look at corresponding fields and ignore others.

Implementation of Memory Buffers :

1. Writable or Temporary Memory Buffers

These are called temporary memory buffers because as data can be written onto them. Thus these buffers can be updated. For writable memory buffers the fields of interest are :

```
struct ngx_buf_s {

        [...]
        u_char          *pos;
        u_char          *last;
        u_char          *start;
        u_char          *end

        unsigned temporary:1;

        [....]
    };
```

The field unsigned temporary:1 indicates that the buffer is a memory writable or temporary buffer.

2. Read only Buffers :

Read only Buffers are those in which the data cannot be updated.The data can be only read.The main advantage of these buffers is that it can be accessed by multiple process at a time.Thus they act as a shared resource.
For Read only memory buffers the fields of interest are :

```
struct ngx_buf_s {

        [...]
        u_char          *pos;
        u_char          *last;
        u_char          *start;
        u_char          *end

        unsigned memory:1;

        [....]
    };
```

The field unsigned temporary:1 indicates that the buffer is a memory read only buffer.

3. File Buffers

To share the data between the files a file buffer is used.This can be accessed by multiple files at a time.
For File buffers the fields of interest are :

```
struct ngx_buf_s {

        [...]
        off_t           file_pos;
          off_t           file_last;
          ngx_file_t      *file;
        unsigned in_file:1;

        [....]
    };
```

The field unsigned in_file:1 indicates that the buffer is a file buffer.

Other Fields :

The other important fields that are common to all buffers are :
pos: It is the pointer to the data to be stored in the buffer
last : pointer to the end of the data
file_pos : When the buffer refers to the data in a file then file_pos points to the start of this data i.e it is the file offset
file_last : When the buffer refers to the data in a file then file_pos points to the end of this data
start : pointer to the start of the buffer
end : pointer to the end of the buffer
file : when buffer is a file buffer then file is a pointer to the actual file object
nmap: When nmap:1 then it means that content of the memory is nmap()ed
temporary: 1 -> Indicates that the buffer is a memory writable buffer
memory:1 -> Indicates that the buffer is a read only memory buffer
file_in:1 -> Indicates that the buffer is a file buffer