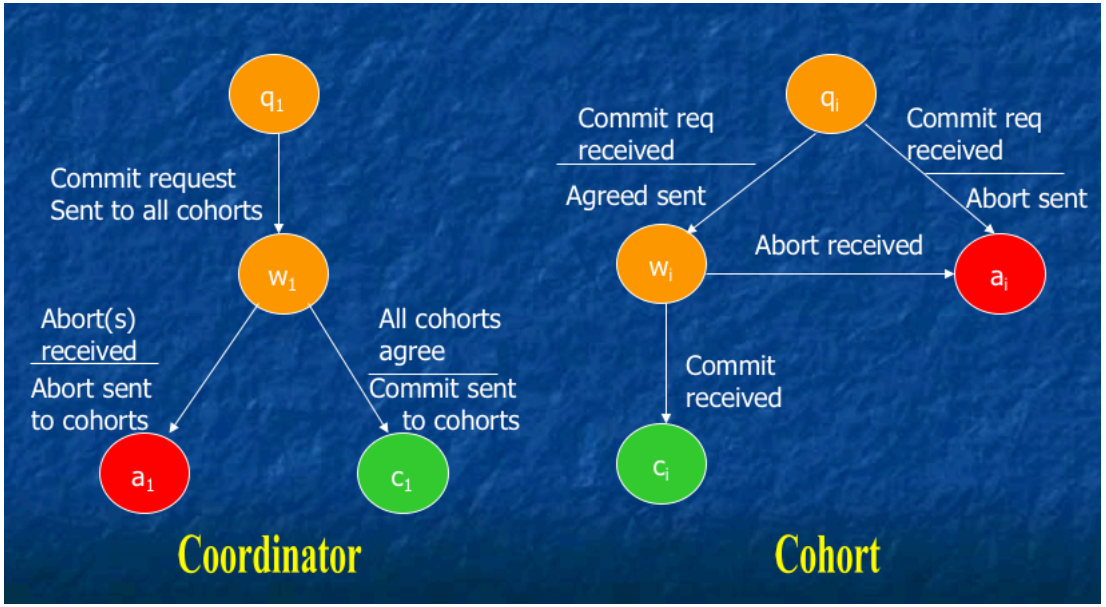
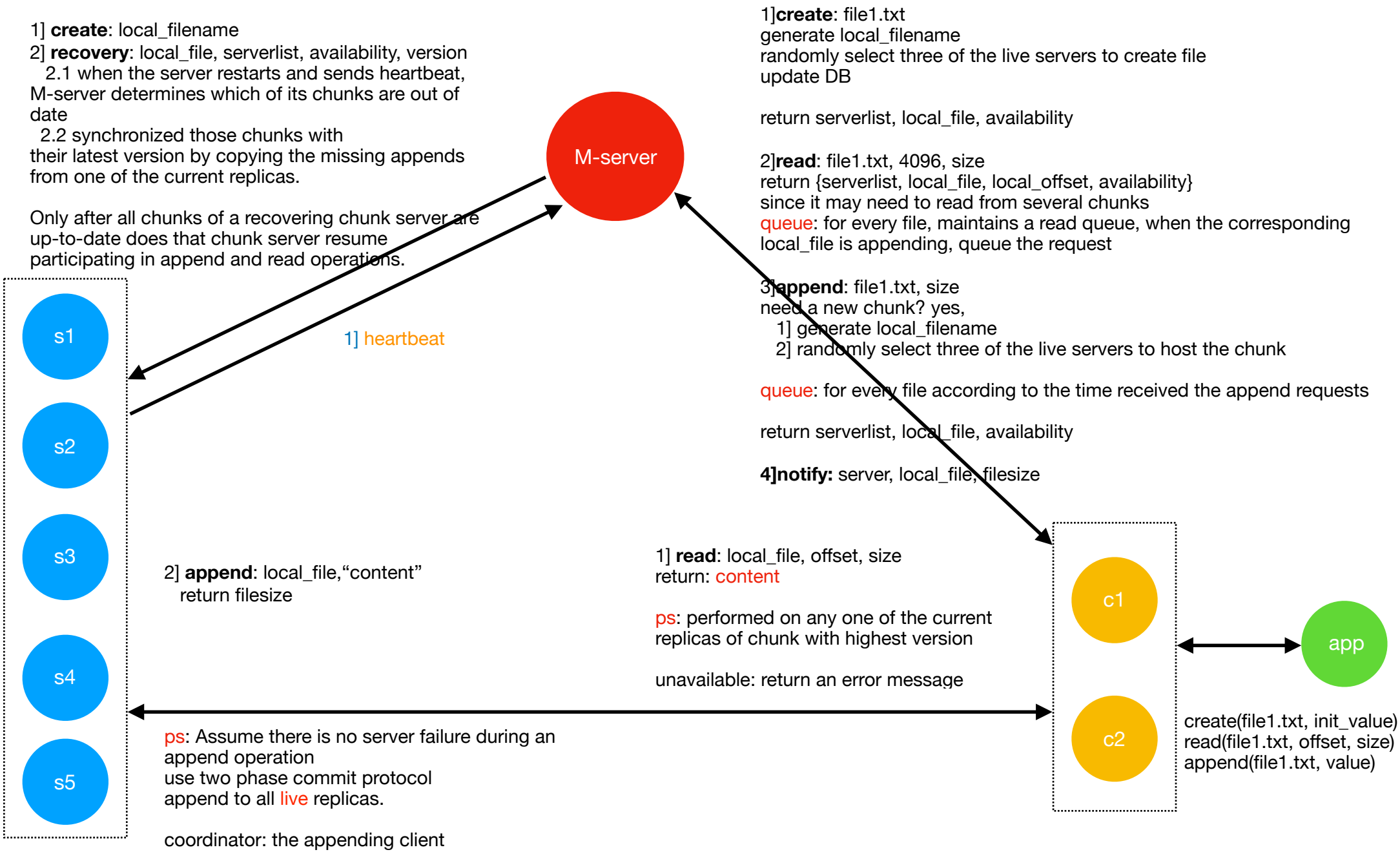
filename	chunk_id	chunk_size	local_file	serverlist	avaiability	version	timestamp
file1.txt	1	8192	ABC	S3,S2,S5	[1 -1 0]	[1 2 2]->[1 3 3]	
file1.txt	2	8192	DEF	S1,S2,S4	[1 1 1]		
file1.txt	3	4096	GHI	S2,S3,S5	[1 1 1]		
file2.txt	1	5	XYZ	S4,S2,S1	[1 1 1]		

primary key(filename, chunk_num)

The maximum amount of data can be appended at a time is 2048 bytes.
8192 - S < appended data size: a new chunk

Test case:
1. normal:
   create and append, then check the chunks and DB
2. read
   2.1 read
   2.2 read more than a chunk maximum size
3. append:
   3.1 one append
   3.2 current append
4. recovery
   turn off s1
   append to file, which has chunks on s1
   recovery s1
   check the result




1] **create**: local_filename
2] **recovery**: local_file, serverlist, availability, version
2.1 when the server restarts and sends heartbeat, M-server determines which of its chunks are out of date
2.2 synchronized those chunks with their latest version by copying the missing appends from one of the current replicas.
Only after all chunks of a recovering chunk server are up-to-date does that chunk server resume participating in append and read operations.
M-server
1] heartbeat
s1
s2
s3
s4
s5
1]**create**: file1.txt
generate local_filename
randomly select three of the live servers to create file
update DB
return serverlist, local_file, availability
2]**read**: file1.txt, 4096, size
return {serverlist, local_file, local_offset, availability}
since it may need to read from several chunks
queue: for every file, maintains a read queue, when the corresponding local_file is appending, queue the request
3]**append**: file1.txt, size
need a new chunk? yes,
1] generate local_filename
2] randomly select three of the live servers to host the chunk
queue: for every file according to the time received the append requests
return serverlist, local_file, availability
4]notify: server, local_file, filesize
2] **append**: local_file,"content"
return filesize
1] **read**: local_file, offset, size
return: content
ps: performed on any one of the current replicas of chunk with highest version
unavailable: return an error message
c1
c2
app
create(file1.txt, init_value)
read(file1.txt, offset, size)
append(file1.txt, value)
ps: Assume there is no server failure during an append operation
use two phase commit protocol
append to all live replicas.
coordinator: the appending client
cohorts: servers