**int** num = 0;

**int** size = Main.*serverList*.size();

Future<Object> future1 = **null**;

Future<Object> future2 = **null**;

Future<Object> future3 = **null**;

**for**(String server: Main.*serverList*){

String[] s = server.split(":");

String address = s[0];

**int** port = Integer.*valueOf*(s[1]);

Callable<Object> qs = **new** QuerySender(address, port,cJSON);

**if**(num%3==0){

future1 = pool.submit(qs);

**if**(num==size-1){

**try** {

*match\_List*.addAll((List<Resource>) future1.get(5,TimeUnit.***SECONDS***));

System.***out***.println(*match\_List*.size());

} **catch** (InterruptedException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (ExecutionException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

} **catch** (TimeoutException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

That is the code of how we add the result comeback from other servers by query command with relay true.

Using 3 threads (implement by Callable because we need to handle to return data from threads), to send query to the servers in server list at the same time when our server received the request of query (relay).

This sentence will block the main thread until the other threads have done.

*match\_List*.addAll((List<Resource>) future1.get(5,TimeUnit.***SECONDS***));

If servers in server list are more than 3, the task of sending request to these servers will be put in a queue. Every time pick the first 3 servers to send the request.

e.g. serverlist: 1. Server1 2. Server2 3. Server3 4. Server4

first time: thread1 send request to server1 && thread2 send request to server2 && thread3 send request to server3 and then waiting for reply.

second time: thread1 send request to server4 and then waiting for reply

