CLUSTER SHELL

There is one server and n nodes in the cluster are clients.

Server Usage : *gcc server.c -o server*
./server <Server Port>

Client Usage : gcc client.c -o client
./client <Server IPaddress> <Server Port Number>

Server accepts path to file containing a space-separated name and IP address mapping in each line in the following fomat -
n1 172.17.72.29
n2 172.17.72.143
n3 172.17.72.82

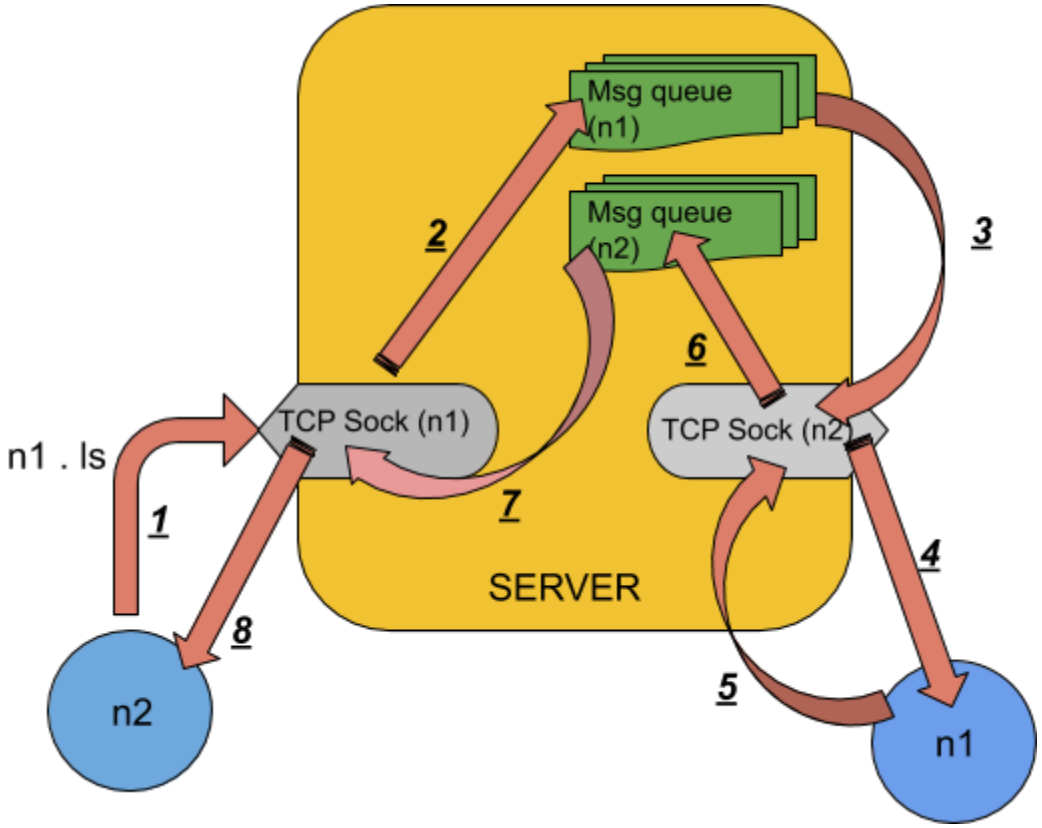
Client uses separate processes, the socket descriptor and standard input. It receives tasks from socket and executes them and sends output back to server to be redirected to proper node. Client also accepts commands to be executed from standard input by itself or else send to server to be executed on some other node.

Client assumes all operators in command separated by space for easy tokenization of arguments for exec calls and avoid confusion between the '.' operator in the problem statement and '.' in filenames (eg: file.txt)

Server creates n child processes for each socket connection with n nodes. Server queues the tasks received for various nodes in a message queue and sends them to the clients.

Each of these server child process checks if there is new message for the corresponding client. This process creates another child process which is responsible for receiving commands from corresponding node intended to be executed in some other nodes.

Hence flow of data between server and its each client is handled by 2 different child processes, one handling the TCP Socket while the other is handling the message queue.


Msg queue
(n1)
Msg queue
(n2)
2
3
6
TCP Sock (n1)
TCP Sock (n2)
n1 . ls
1
7
4
SERVER
8
5
n2
n1