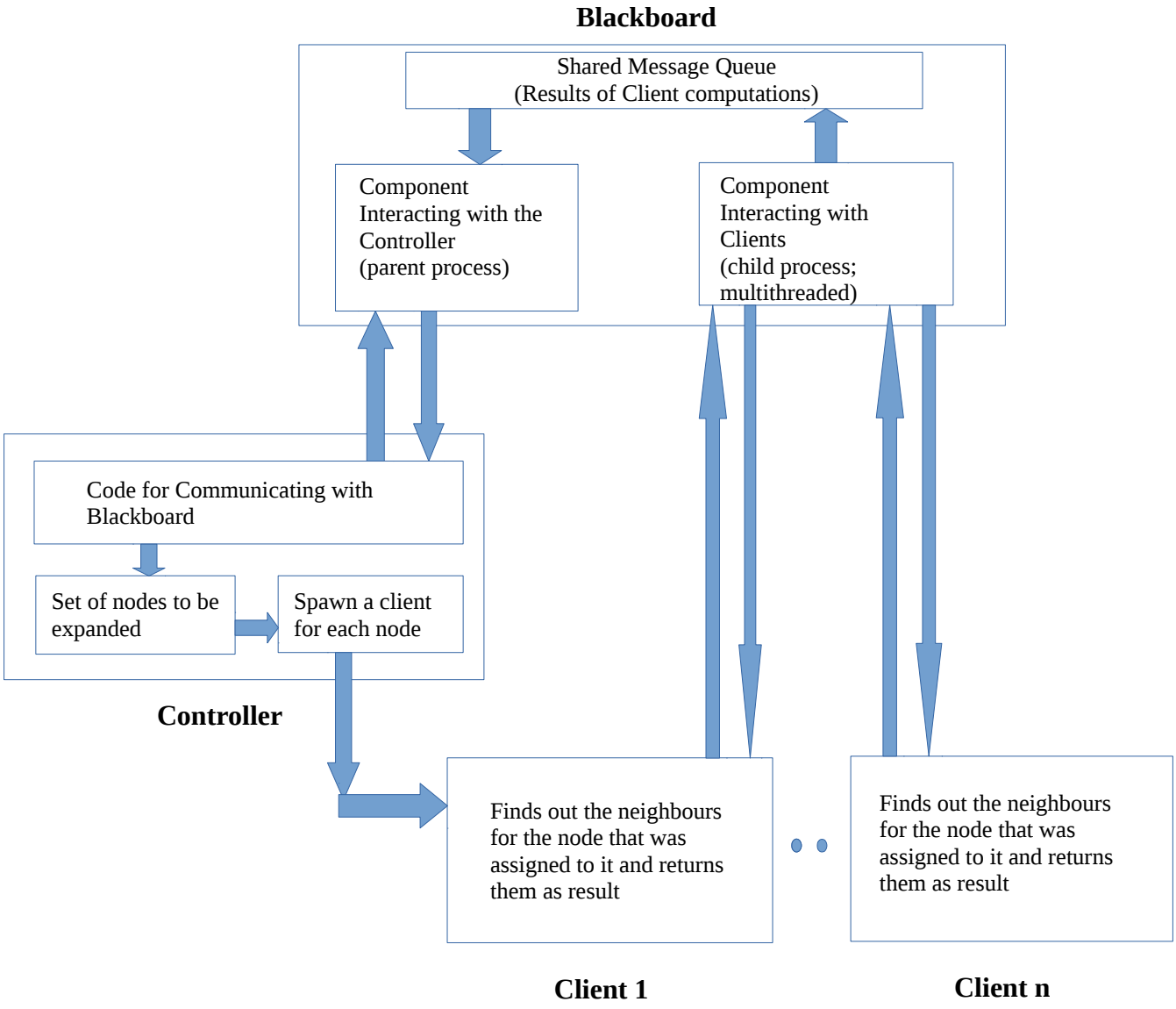
CS718

Implementation Blackboard Architecture

Harshad Chavan 143050028
Aby Sam Ross 143050093

README File

Overall Structure of the implementation:



Contents of the submission:

- */bb/bb.c* :
  C source for the **Blackboard**
- */bb/bb.h* :
  Header file for the Blackboard source
- */bb/run_bb* :
  Executable for Blackboard

- */bb/graph.txt* :
  Contains the adjacency matrix for the graph on which BFS needs to be run. Each row is written on one line of the file and each element in a row is separated with ";" (semi-colon)
  e.g.
  0;1;1;
  1;0;1;
  1;1;0;
- */bb/keys.txt* :
  Contains a ";"(semi-colon) separated list of values(keys) on each node in the graph in a single line
  e.g.
  a;b;c;
- */bb/target.txt* :
  Contains the target key i.e. the key that the BFS will search in the graph.
  e.g.
  c;
- */bb/Makefile* :
  Makefile for compiling the Blackboard source with appropriate flags and building an executable

- */ctrl/ctrl.c* :
  C source for the **Controller**
- */ctrl/run_ctrl.c* :
  Executable for Controller
- */ctrl/BFS_client.c* :
  C source for a **Client** that expands the node assigned to it
- */ctrl/BFS_client* :
  Executable for Client
- */ctrl/trusted_hosts.txt* :
  List of hosts on which the controller can spawn clients.
  Contains one line per host having
  - Hostname(*user@ipaddress*)
  - Password
  - Path to the directory where BFS_client resides on the host

  in a ";"(semi-colon) separed format
- */ctrl/Makefile* :
  Makefile for compiling building executables of client and controller

- /README.pdf :
  Contains instructions to set up and run codes, sample graph and screen shots.

<u>Instructions for setting up and compiling the code:</u>

1. Extract the contents of submission into a directory of your choice. Let that be "*/mydir/*".

2. First we need to compile the Blackboard. Open up a terminal and type:
   $ cd *path-to-mydir/bfs/bb*

3. Compile the Blackboard source using a makefile i.e. type into the terminal:
   $ *make*

4. Open up another terminal and type:
   $ cd *path-to-mydir/bfs/ctrl*

5. Compile the Controller and Client source using a makefile i.e. type into the terminal:
   $ *make*

6. The list of hosts on which the Controller can spawn BFS_client is available in *trusted_hosts.txt* file. We can add/delete any hosts from this list by deleting a line from the file. While adding, make sure that the format is as follows:
   username@ipaddress;password;absolute-path-to-the-directory-where-BFS_client-resides-on-host;

   e.g. *harshad@10.196.29.205;xyz;/home/harshad/bfs;*

7. Repeat steps 1 through 6 on each machine on which you want to deploy any component(Blackboard, Controller or Client) of the system. (Alternative you can only set up blackboard code on then machine where you want to a blackboard only, and so on for machines with Client and Controller.)

Instructions for Running the Code:

Following steps have to pe performed in the same order.

1. Setting up a problem Instance:

   The problem instance for BFS search is specified in files *graph.txt, keys.txt* and *target.txt*. **If you wish to** change the problem instance i.e. graph, node to be serached and values on each graph node, open the corresponding file in an editor an make appropriate changes as required, before proceeding to step 2.

2. Running the Blackboard:

   - Open up a terminal and type:
     $ cd *path-to-mydir/bfs/bb*

   - Run the blackboard by typing in the following command:
     $ ./run_bb <*port_num_1*> <*port_num_2*>

     where <*port_num_1*> is the port Blackboard uses to communicate with the Controller and <*port_num_2*> is the port the Blackboard used to communicate with the Client/s. You need to remember these values and use the same while running the controller.

3. Running the Controller:

   - Open up another terminal and type:
     $ cd *path-to-mydir/bfs/ctrl*

   - Run the controller by typing in the following command:
     $ ./run_ctrl <*ip_address_of_the_blackboard*> <*port_num_1*> <*port_num_2*> <*client_type*>

where <*port_num_1*> and <*port_num_2*> should be the same values used for blackboard's execution. <*client_type*> can have values *"L"* or *"R"*.

- *"L"* tells the Controller to spawn clients locally i.e. on the same machine as the Controller
- *"R"* tells the Controller to spawn clients remotely i.e. on another machine which is one of the hosts specified in the *trusted_hosts.txt* file.

You will be able to see the messages sent during the communication between Blackboard, Controller and Clients. Once the BFS is done and the key is found in the graph, Controller will print the Node ID (and integer identifier) of the node of the graph which contains the target key.

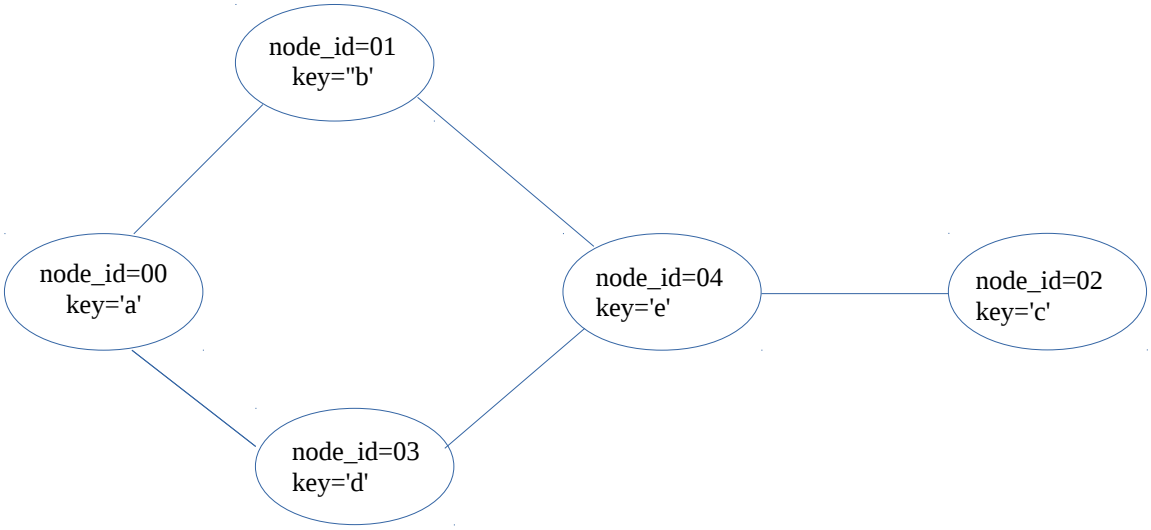
Note 1:
Between successive runs of Blackboard(*run_bb*) , perform following actions:
1. Type the following command in the terminal:
   $ ipcs
2. Note the *msqid* from "Message Queues" section.
3. Type the following command in the terminal:
   $ ipcrm msg <*msqid*>
   e.g.
   $ ipcrm msg *32769*

These actions are required for deleting any active message queues that might be created but not closed by the Blackboard.

Note 2:
The search on the graph will always start from the node with *node_id=00.*

Example Graph:

(example submitted with the assignment)



Adjacency Matrix (*graph.txt*):

0;1;0;1;0
1;0;0;0;1
0;0;0;0;1
1;0;0;0;1
0;1;1;1;0

Keys List (*keys.txt*):
a;b;c;d;e

Target (*target.txt*):
e;

Screenshots:

Blackboard Execution:

```
┌─(~/bb_bfs/bb)──────────────────────────────────────────(diana@aby-Lenovo-Z50-70:pts/3)─┐
└─(20:29:45)──> ./run_bb 6000 6001                                            ──(Mon,Feb01)─┘
bb parent : Target: e
bb parent : keys: abcde
bb parent : 0: 01010
bb parent : 1: 10001
bb parent : 2: 00001
bb parent : 3: 10001
bb parent : 4: 01110


                --------------Welcome to Blackboard--------------

                ------------------------------------------------

                        Host: aby-Lenovo-Z50-70

                        AF:  2

                        Port:  6000

                        IP:  10.196.105.250

                ------------------------------------------------

                ------------------------------------------------

                Client: "10.196.105.250" connected on socket: 7.

                10.196.105.250 IP: 10.196.105.250.

                10.196.105.250 running on: 59367.
```

```
                ------------------------------------------------

bb parent: Controller In
bb parent: ctrl "10.196.105.250" connected on socket  7 says: get

bb parent: result with ctrl: res00
bb parent: ctrl "10.196.105.250" connected on socket  7 says: get

                ------------------------------------------------

                Client: "mars.cse.iitb.ac.in" connected on socket: 7.

                mars.cse.iitb.ac.in IP: 10.105.1.11.

                mars.cse.iitb.ac.in running on: 53725.

                ------------------------------------------------

bb child: Client "mars.cse.iitb.ac.in" connected on socket  7 says: get

bb child: Received get from client: mars.cse.iitb.ac.in  on socket: 7

sending: 01010
sending: 10001
sending: 00001
sending: 10001
sending: 01110
bb child: Received res: res0103 from client: mars.cse.iitb.ac.in  on socket: 7

bb parent: result with ctrl: res0103
bb parent: ctrl "10.196.105.250" connected on socket  7 says: get
```

```
                ------------------------------------------------

                Client: "mars.cse.iitb.ac.in" connected on socket: 8.

                mars.cse.iitb.ac.in IP: 10.105.1.11.

                mars.cse.iitb.ac.in running on: 53727.

                ------------------------------------------------

bb child: Client "mars.cse.iitb.ac.in" connected on socket  8 says: get

bb child: Received get from client: mars.cse.iitb.ac.in  on socket: 8

sending: 01010
sending: 10001
sending: 00001
sending: 10001
sending: 01110
bb child: ERROR: recv(): while reading from connected client socket: 8: bb.c: 407 !!
Connection reset by peer
```

```
                ------------------------------------------------

                Client: "10.196.29.205" connected on socket: 7.

                10.196.29.205 IP: 10.196.29.205.

                10.196.29.205 running on: 59231.

                ------------------------------------------------

bb child: Client "10.196.29.205" connected on socket  7 says: get

bb child: Received get from client: 10.196.29.205  on socket: 7

sending: 01010
sending: 10001
sending: 00001
sending: 10001
sending: 01110
bb child: Received fnd: fnd04 from client: 10.196.29.205  on socket: 7

bb parent: result with ctrl: fnd04
bb parent: ctrl "10.196.105.250" connected on socket  7 says: OK

bb parent: ctrl "10.196.105.250" connected on socket  7 exiting!!: OK
```

Controller and Client Execution:

(client **runs on a different machine** than controller but since controller spawns all the BFS_client, the stdout messages of BFS_clients appear in the same window as the controller)

```
┌──(~/bb_bfs/ctrl)──────────────────────────────────────────────────────────────────(diana@aby-Lenovo-Z50-70:pts/4)─┐
└─(20:30:04)──> ./run_ctrl 10.196.105.250 6000 6001 R                                                 ──(Mon,Feb01)─┘
CTRL : Connected to server
CTRL : Server replied: res00
CTRL : Blackboard says: 00
CTRL : Node ID to be explored: 0
CTRL : Node ID string being passed: 0
CTRL : Server replied: res0103
CTRL : Blackboard says: 0103
CTRL : Node ID to be explored: 1
CTRL : Node ID string being passed: 1
CTRL : Node ID to be explored: 3
CTRL : Node ID string being passed: 3
CLIENT (3) : Connected to Blackboard
CLIENT (3) : Target is : e
CLIENT (3) : Set of keys: abcde
CLIENT (3) : Matrix line 0: 01010
CLIENT (3) : Matrix line 1: 10001
CLIENT (3) : Matrix line 2: 00001
CLIENT (3) : Matrix line 3: 10001
CLIENT (3) : Matrix line 4: 01110
CLIENT (3) : Result sent to res0103
CLIENT (3) : Connected to Blackboard
CLIENT (3) : Target is : e
CLIENT (3) : Set of keys: abcde
CLIENT (3) : Matrix line 0: 01010
CLIENT (3) : Matrix line 1: 10001
CLIENT (3) : Matrix line 2: 00001
CLIENT (3) : Matrix line 3: 10001
CLIENT (3) : Matrix line 4: 01110
CLIENT (3) : Result sent to fnd04
CLIENT (3) : Connected to Blackboard
CLIENT (3) : Target is : e
CLIENT (3) : Set of keys: abcde
```

```
CLIENT (3) : Matrix line 0: 01010
CLIENT (3) : Matrix line 1: 10001
CLIENT (3) : Matrix line 2: 00001
CLIENT (3) : Matrix line 3: 10001
CLIENT (3) : Matrix line 4: 01110
CLIENT (3) : Result sent to fnd04
CTRL : Server replied: fnd04
CTRL : Key was found on node with node_id : 04
CTRL : Problem solved. Controller exits !!
 Waiting for unfinished clients...
┌──(~/bb_bfs/ctrl)──────────────────────────────────────────────────────────────────(diana@aby-Lenovo-Z50-70:pts/4)─┐
└─(20:37:09)──>                                                                                       ──(Mon,Feb01)─┘
```