

## CS454 A3 System Manual

Anton Shevchenko - ashevche - 20549540

Jonathan Donas - jdonas - 20631146

### Design Choices

#### *Marshalling/Unmarshalling Data:*

Function names and IP addresses are sent as 65 byte strings over the network. This is because the assignment specification guarantees that function names will not exceed 64 bytes. IP addresses also do not exceed 64 bytes, in string form. The client sends argTypes to the server one-by-one as integers.

For scalar values, the client and server send the proper number of bytes by using the sizeof() function. A contiguous block of memory, determined by the size of the data type multiplied by the number of entries in the array, is sent for arrays. The server uses argTypes to determine the size of the arguments before receiving them. It then allocates heap memory and receives the arguments into the memory.

From client → server over the network, only values of input arguments are sent. Only values of output arguments are sent from server → client. The client and server read and write these values to the appropriate memory locations when they are received.

#### *Binder Database Structure:*

The binder uses a simple map of (function names → function structure). The function name is a string of the function name concatenated with the string representation of each argType integer. A sample function name might look like "f010001000." It is important to note that arrays are stored with length 1, for any array size. This is so that functions with arguments that differ only by array size (greater than 0) are treated as the same.

The function structure holds a list of available server structures that can service the particular function. It also stores extra information, such as the current server to use, to help with round-robin scheduling. Server structures simply hold an IP address and a port.

#### *Function Overloading:*

Function overloading did not require any special implementation. As mentioned in the previous section, functions are identified by the function name concatenated with the string representation of each argType integer. This makes a string that's unique to the function and its arguments. So, functions with the same name but different arguments are treated as different.

#### *Round-Robin Scheduling:*

The binder has a global variable that keeps track of the most recently used server, for any request. When a new request comes in, the binder looks up all the available servers for that request. Function structures keep track of the current server that should process this request. If the current server is the same as the server that was last used (stored globally), then the binder will use the next available server instead (if there is one). The server trackers are then updated. This ensures that servers are used as equally as possible: and in a round-robin fashion.

#### *Termination Procedure:*

To terminate, a client simply sends an integer to the binder that corresponds to the TERMINATE constant. The binder then iterates through the list of connected servers and sends a TERMINATE message over the binder/server socket. The server will free memory and exit all threads as soon as it receives this message. In case of pre-mature server disconnection, the binder will remove all function mappings for this server.

## Error and Warning Codes

### *rpcInit:*

- environment variables not set: -1
- could not connect to binder: -2

### *rpcCall:*

- environment variables not set: -1
- could not connect to binder: -2
- could not connect to server: -3
- LOC\_FAILURE, no server found: -4
- EXECUTE\_FAILURE, function not found: -5
- EXECUTE\_FAILURE, skeleton returned error: -6

### *rpcCacheCall:*

- environment variables not set: -1
- could not connect to binder: -2
- could not connect to server: -3
- CACHE\_LOC\_FAILURE, no servers found: -4
- could not execute on any servers: -5

### *rpcRegister:*

- could not register: -1
- warning, duplicate function registration: 1

### *rpcExecute:*

- no registered functions: -1

### *rpcTerminate:*

- environment variables not set: -1
- could not connect to binder: -2

## Features Implemented

All features (including the bonus feature of *rpcCacheCall*) were implemented.