**CprE 550 P1 Report**

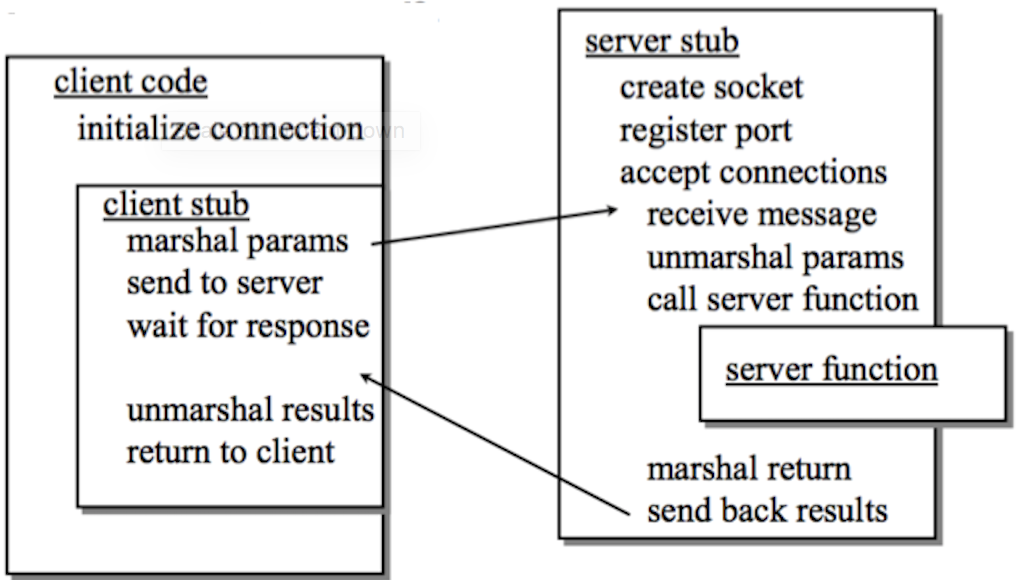
**Objective:**

The objective of the machine problem is to write a network management application that tracks user logins, CPU usage and other statistics on a host and allows querying by a RPC-based network management system. Its output can be used to feed into an analysis component for deciding on corrective actions in self-managing distributed systems.

**Preparation:**

The main component in this project is called ‘stub’. ‘stub’ make remote procedures appear local through stub functions.

Here is the structure of the design:



The Sun RPC compiler is called rpcgen. As input, it takes a list of remote procedures (interfaces) defined in an interface definition language (IDL)

Here are the brief steps for run RPC:

1. Define remote procedure interface in interface definition language (IDL)
2. Use RPC compiler called rpcgen to compiler the IDL.
3. Write server code and client code. Both will call the header file from rpcgen.
4. Use gcc compiler to compiler the server and client code.
5. Run server code first and then run client code.

For this project, the IDL has already been defined for us.

**Detailed Implementation:**

* Client Side Code:

For the client.c, we don’t need to write anything associated with reading system info. All we need to do is to print the returned pointer value.

* Server Side Code:

1. Read Memory: We use the Linux header file <sys/sysinfo.h> to fetch the memory info. From the sysinfo, we can get free ram and total ram. After that, we only need to do a floating-point division to get the percentage.
2. Process: Same as Read Memory. The info is called loads[0].
3. CPU usage: We read the file “/proc/stat” to get the process. The stat contains the following info:
   1. user: normal processes executing in user mode
   2. nice: niced processes executing in user mode
   3. system: processes executing in kernel mode
   4. idle: twiddling thumbs
   5. iowait: waiting for I/O to complete
   6. irq: servicing interrupts
   7. softirq: servicing softirqs

We need to set a time interval to find the changes of the running process and total idle process in order to get the CPU usage.