Homework 3 (due 11/13/2009 at 11:05am)

via email to ada@cc or hardcopy in class (preferred)

1. In the URPC lecture we discussed ways in which data can be efficiently moved across address space boundaries. The URPC implementation uses request queues in shared memory segments. Discuss how would you design the request data structure in order to make the data movement most efficient. Your design will likely need to incorporate different options so you can efficiently handle data of different sizes.

2. In the remote procedure call (RPC) paper, in several places, the implementation makes use of the fact that calls are synchronous and a client does not make the next call before the result for the pending call is returned. In many applications, an asynchronous RPC is useful which allows a process to make a call and then collect the results at a later time. Thus, the process is able to execute in parallel with the execution of the call at the server node. Since the process may issue multiple asynchronous calls before the first one returns, it is no longer possible that the runtime deals with at most one call between a client-server pair. (Multithreading can also result in multiple outstanding calls even when calls are synchronous).

Discuss how the design and implementation presented in the paper will have to be changed to accommodate asynchronous RPCs. In particular, discuss the data structures maintained at client and server side and describe the changes that will be necessary.