Java RMI Notes

How it works:

1. Develop the parameters for a request in the Client application
2. Pass the request (parameter) to the Server using RMI
3. The server separates the request, into its component parts, and schedules the request on each queue
4. A thread within each queue picks up the request and invokes the application class passing it the input message (optional) from the parameter of the original request.
5. The server collects the (optional) return objects from the application classes and presents this response to the requesting client for synchronous requests or (optionally) further schedules the agent for asynchronous requests.

Remote Invocation without using Java RMI

1. I would say you can get by with 1) using java.lang.reflect.\* for creating proxies (i have some code for this, will post tonight)  
   sockets & threading for binding
2. I know that java 1.5 has new classes for managing thread pools, which might be good for resource management
3. Serialization for marshalling. i have an old post that shows you how to implement serialization for streams which might be helpful, filed under RMI.
4. Thread synchronization for concurrent stuff. You will need to have a multithreaded server listening.