Lars Sorenson
CS 422 – Lab 2 Bonus Problem Writeup

To implement IPC with the least overhead in the case of a pool of workers being delegated tasks would probably be something along the lines of a blocking queue. Having the parent process queue requests and the children pulling from the queue as they are available, and mutexing or using a semaphore to make sure no other child is reading the same request, would be easier and offer the least resistance as compared to some other solutions. For example, having the parent attempt to pipe directly to each child would require overhead of maintaining a pipe to each process, checking each process through a series of pipe reads and writes to determine if it is available, blocking and interrupts to prevent waiting too long, etc. would all accumulate to a lot of overhead versus having a queue in shared memory, with a counter to indicate if it has items in it and a flag, mutex, semaphore etc that would block a reader if someone was already reading or if the parent was writing to prevent race conditions. The queue would allow the children to read from it as they are free to work, cutting down on the work the parent has to do, and the queue will maintain itself, also lending to further efficiency.