

## Project #3 – Erlang

### Overview

The purpose of this project is to gain additional exposure to message passing in Erlang. In the interest of time, this assignment is broken into two parts. The first is a “toy” message passing exercise. The second is a reimplementaion of part of project #2 (the problem domain and decomposition should be familiar).

### Part 1

This exercise requires matching processes of different types with each other. There are two types of processes, Oxygen and Hydrogen. An Oxygen process must be matched with two Hydrogen processes (assume that each process has a unique identity; the process id can serve this purpose or you can assign an integer to each).

Your program should take a single argument,  $N$ , and spawn  $N$  Oxygen process and  $2 * N$  Hydrogen processes. These processes then communicate with a “server” process that waits for messages from two Hydrogen processes and one Oxygen process, in any order, and tells each the identities of the matched processes. Each Hydrogen and Oxygen process should print its respective matches and then exit.

### Part 2

Write an Erlang program to implement the Jacobi iteration problem described in project #2. The program must take as arguments the size,  $N$ , of the grid ( $N * N$ ), the number of processes that will work on this grid, and the number of iterations to compute. You may, again, assume that the processes will evenly split the grid. In particular, your implementation need only partition the space into rows.

Measure the execution time of different configurations and report these as part of your submission. You are encouraged to compare these against the MPI implementation, but you will need to rerun those experiments on falcon to do so (since falcon is the only machine that currently supports the Erlang installation).

### Handin

Submit, using `handin`, your source code and paper to `458.hw3` on vagon.