Parallelism Day – Exercises

Avi Kulkarni

July 2, 2022

The exercises do not require the speedups of parallelism. Their purpose is entirely didactic.

For security reasons, Do not `DistributedManager` *on multi-user machines.*

All files referenced can be found in the directory `cookbook/exercises/`

1. Using `SetNthreads`, determine the N on your machine which minimizes the execution time of

```
SetNthreads(N);
X := Random(MatrixRing(GF(5), 10000));
time P := X*X;
```

2. Write a program with a manager process that launches 10 workers. Each worker should print

   "Hello world, I am doing job N."

   where N is the value received from the manager. You should only need one shell, and control should return to the user after the work is finished.

3. Run "dyad_institute.m" on your machine. Rescue the resulting orphans.

4. Write a function which takes in a sequence of multivariate polynomials and returns their squares. This function should invoke parallelism, and should setup/tear-down all necessary resources to do this.

5. Using task groups, write a script which takes in a list of curves

   $$[* C_1, \ldots, C_n *]$$

   over the rationals and returns the L-polynomials over the primes $11, 13, 17$. You may find the function `LPolynomial(C, p)` is of some help.

   Input is provided in "curves.m". You should not worry about handling errors.

6. Find a real quadratic field whose class group contains an element of order 79.

7. A list of polynomial ideals is provided in the file "ideals.m". Using task groups, race two workers using the `Al` parameter switch of magma's `Groebner` method:

```
Groebner(I : Al:="Default"); // Worker 1
Groebner(I : Al:="Direct");  // Worker 2
```

   In each example, report which method terminated first, and terminate the loser to save time.