Implementation and Performance Analysis of one of the following algorithms

CPAR exercise 2

To be develop by groups of 2 elements, which aims to implement and evaluate one of the following algorithms.

1. LU factorization

The LU factorization is a modified form of Gaussian elimination algorithm and is used to solve systems of linear equation, of the form *Ax=b*.

1. Implement a sequential version.

2. Implement a block oriented sequential version.

3. Implement a shared memory version using OpenMP and OpenCL/CUDA (if available).

4. Performance analysis of the implementations.

The time complexity of the algorithm for a matrix of size (n,n) is $\Theta(2/3n^3)$.

Data range to consider (n): from 1000 to 6000, with a step of 1000.

2. The Sieve of Erastosthenes

The Sieve of Eratosthenes is a simple algorithm to find the prime numbers up to a given number ***n***.

Consider the following implementations:

(i) sequential, on a single CPU-core;
(ii) parallel, on a shared memory system, using OpenMP;
(iii) parallel, on a distributed memory system using only MPI and MPI with the shared memory version.

The following steps describe the algorithm:

```
1. Create list of unmarked natural numbers 2, 3, …, n
2. k ← 2
3. Repeat
        (a) Mark all multiples of k between k² and n
        (b) k ← smallest unmarked number > k
   until k² > n
4. The unmarked numbers are primes
```

The time complexity of the algorithm is $\Theta(n \ln \ln n)$.

Data range to consider (n): from 2^{25} to 2^{32}.

3. Distributed matrix multiplication algorithm (SUMMA)

The Scalable Universal Matrix Multiplication Algorithm is a straightforward, highly efficient, scalable implementation of common matrix multiplication operations. Implement SUMMA using MPI and compare it to a shared memory algorithm (OpenMP, OpenCL or CUDA).

NOTE

Performance analysis consists in analyzing single processor performance and, speedup, efficiency and scalability from 1 to P processors for the parallel versions. A discussion on the obtained results is also expected.

Computing Platforms

Two computing platforms are available in the Lab:

PLATFORM 1:
One multicore processor.

PLATFORM 2:
One or more nodes, each with a multicore Processor.

To be delivered up to: 5/05/2017

Parameters for Report Evaluation (Maximum of 10 pages):

- Problem description;
- Sequential solutions and performance measures;
- Parallel algorithms and their characterization;
- Time measures of the parallel programs;
- Performance evaluation and scalability analysis;
- Writing and results analysis.