INTRO. TO PARALLEL & DIS. PRO.

My home ► My courses ► Managed Courses ► Semester 1702 ► 1702-COL380 ► 2 April - 8 April ►
Lab 3: Learn MPI

Lab 3: Learn MPI

The assignment is due on 15 April.

In this assignment you will do 2D sorting using multiple processes. In short, there is a 2D grid of <key, value> pairs. The key is a floating point number and the values are strings of specified length. The goal is to sort them so that each row is sorted from left to right and each column is sorted from top to bottom. The algorithm you should follow is:

```
Repeat until done:
   Sort each row
   Sort each column.
```

The data will be provided in multiple little-endian binary files, one column per file. Each column file begins with a 4-byte integer (call it n) indicating the length of all strings in that file. (All strings in a column are the same length.). That 4-byte integer header is followed by a sequence of <key, value> pairs placed contiguously. Key is a 4 byte float and the value is a sequence of n bytes. The number of columns has to be determined by the reader. This means the number of elements in different columns may be different. In such case, rows having fewer elements must still be sorted left to right, without holes.

The number of columns will be specified on the command line as follows (to be included along with mpirun)

```
sort2d <count> <basefile>
```

<count> is an integer indicating the number of columns, which are respectively in files named <basefile>1, <basefile>2 .. <basefile><count>, one column per file.

The output must be in a single file called <basefile>0 in row major order (with no header or other extra information). The code will be compiled using openmpi 1.10 and tested by employing 2-48 processes and checking the result for correctness. More than 5 test cases will be used. The maximum number of columns can be up to 10,000 and the maximum number of rows can be up to 10 million. If results are correct you get 75% of the marks. Incorrect results yield 0. The remaining 25% of the marks will be based on the speed on the largest test case. The fastest 10% will get the full 25%. The remaining will be scaled appropriately.

To get the 75%, the results must be correct on the first run on an unannounced test case, without any code modification needed after the submission. Do not make any assumptions about the test. Disable any debugging printfs in your code before submission.

Your submission should be a single zip file called assignment3.zip, which must include all your source code and a makefile, which produces the executable sort2d. The following steps should be able to run your code (notice, all steps are in the same directory):

1. unzip assignment2.zip
2. make
3. mpirun sort2d <count> <basefile>

(Makefile should use mpicc/mpiCC of openmpi 1.10.0)

See below for an example set of input column files (zipped).

cold.zip

Submission status

Submission status	No attempt
Grading status	Not graded
Due date	Sunday, 15 April 2018, 11:55 PM
Time remaining	9 days 23 hours
Last modified	Friday, 6 April 2018, 12:17 AM
Submission comments	▸ Comments (0)

Add submission

Make changes to your submission

NAVIGATION

My home

- Site home
  - Site pages
  - My profile
  - Current course
    - 1702-COL380
      - Participants
      - General
      - 1 January - 7 January
      - 8 January - 14 January
      - 15 January - 21 January