Your Special Challenge

Develop a parallel sorting application with C and MPI

Sorting is a fundamental problem in computer science. Unfortunately, for most serial sorting algorithms, it is challenging to implement versions that benefit from the potential speedup of parallel computers with distributed memory.

Parallel Mergesort [1][2][3][4] is one example of a sorting algorithm that is useful for parallel systems.

The individual steps of your task

1. **Investigate** how to implement with the programming language C and with MPI an application, that does the Mergesort in parallel. Find out which parts of the sorting process can be parallelized. Literature provides many helpful sources, covering sorting algorithms and how to implement it in C with MPI [5][6][7].

2. **Deploy** a MPI environment for testing and development on physical hardware, inside virtual machines [8][9] or inside a public cloud infrastructure service like EC2 [10][11][12].

3. **Develop** your application with C and MPI.

4. **Test** your application on your own MPI cluster and test it on our 128 node cluster with the 512 CPU cores. Test your application...

   - with sufficient large problem sizes (this means you prior need to generate some data sets of several MB in size) and
   - with different numbers of cores (1, 2, 4, 8, 16...512) and

   calculate the speedup. Your outcomes may prove some of the laws and limitations we already discussed during class in slide set 1.

5. **Create** a presentation (max. 30 Minutes) with maximum 15 slides and additionally a live demonstration. Give your presentation during class or during an exercise session.

Some final words to motivate you

In real life, one or two weeks is a typical amount of time to do such a task in parallel to your daily tasks. In this course you have several weeks and you are not alone.

Last but not least, questions of the exam will cover this special task!

References

[1] "Parallel Mergesort." http://www.mcs.anl.gov/~itf/dbpp/text/node127.html.

[2] "One implementation of the mergesort in parallel using MPI." https://github.com/racorretjer/Parallel-Merge-Sort-with-MPI.

[3] C. Siebert and J. L. Träaff, *Efficient MPI Implementation of a Parallel, Stable Merge Algorithm.* Springer, 2012. https://apps.fz-juelich.de/jsc-pubsystem/aigaion/attachments/siebert_merging.pdf-02e21dab116de083c8e60d6da16ebe4c.pdf.

[4] "A Specimen of Parallel Programming: Parallel Merge Sort Implementation." http://penguin.ewu.edu/~trolfe/ParallelMerge/ParallelMerge.html.

[5] T. Bräunl, *Parallele Programmierung: Eine Einführung.* Vieweg, 1993.

[6] D. Knuth, *The Art of Computer Programming – Volume 3 – Sorting and Searching.* Addison-Wesley, 2nd edition ed., 1998.

[7] "Parallel Algorithms – sorting." http://www.dcc.fc.up.pt/~fds/aulas/PPD/1112/sorting.pdf.

[8] "MPI Tutorials 01." https://www.youtube.com/watch?v=2rpWEZY0aPo.

[9] "Create test MPI cluster using VirtualBox images." http://www.zamaudio.com/?p=837.

[10] "Launching an Amazon EC2 MPI Cluster." http://mpitutorial.com/tutorials/launching-an-amazon-ec2-mpi-cluster/.

[11] "Using MPICH in Amazon EC2." https://wiki.mpich.org/mpich/index.php/Using_MPICH_in_Amazon_EC2.

[12] "MPI on EC2." https://www.youtube.com/watch?v=049CVTGHO1k.