Parallel Computing

Benson Muite

benson.muite@ut.ee
http://kodu.ut.ee/~benson
https://courses.cs.ut.ee/2016/paralleel/fall/Main/HomePage

24 October 2016
Given a data set, decompose it into similar items

Parallel computing useful for large data sets

Many possible algorithms

Will look at K-means algorithm

Presentation follows F. Nielsen

Introduction to HPC with MPI for Data Science
Consider grouping points in N-dimensional space
As an example, consider dimensions of passenger road vehicles
May wish to split them into cars, vans, buses, trains
What dimensions would be most useful?
Having chosen the dimensions, need an algorithm

Have already decided on 4 categories

Typically do not know anything more about the data

For simplicity, assume there is at least one representative in each category
a) Pick 4 cluster centroids randomly
   i) Calculate distance of each point to a centroid
   ii) Put each point in a cluster based on centroid it is closest to
   iii) Calculate centroids of each cluster
   iv) Repeat i-iv until sum of distances from centroids stops decreasing
K means

- Typically use square of euclidean distance
- Can use other distances, depending on application, some may be better than others
- Method converges, because at each iteration “energy” or sum of squares of euclidean distances always decreases, but remains positive (fixed point theorem)

Example at
http://shiny.rstudio.com/gallery/kmeans-example.html
Parallelization

- Can parallelize calculating distances from centroids, no communication
- Can parallelize calculating centroids, reduction and broadcast communications needed
- Reduction and broadcast also needed to check for convergence
- Can use parallel IO
- Should weak and strong scale quite well
- Examples at
  [http://rbigdata.github.io/documentation/pmclust/01-pmclust_pkmeans.html](http://rbigdata.github.io/documentation/pmclust/01-pmclust_pkmeans.html) and
  [https://github.com/RBigData/pmclust/blob/master/demo/ex_kmeans.r](https://github.com/RBigData/pmclust/blob/master/demo/ex_kmeans.r)
- Can later compare speed to own code
Heterogeneous architectures for high performance with low energy consumption
Many different kinds of hardware
Many programming models
Accelerators

- Graphics Processing Units (GPU)
- Field Programmable Gate Arrays (FPGA)
- Xeon Phi
- Massively Parallel Processor Array (MPPA)
- Other specialized processing units, for example for encryption and signal processing
Nvidia GPUs

- http://www.nvidia.com

- 2 Tflop double precision performance
- Programming APIs CUDA, CUDA Fortran, OpenCL, OpenACC
- For compute and graphics
AMD Firepro GPUs

- [http://www.amd.com](http://www.amd.com)

- 2 Tflop double precision performance
- Programming APIs OpenCL, OpenACC, HCC and HSAIL
- For compute and graphics
Intel Xeon Phi

- http://www.intel.com

- 1 Tflop double precision performance
- Programming APIs OpenCL (old versions), OpenMP, MPI, CILK, Fortran, C
- Latest versions can be self hosted
- For compute and graphics

http://www.parallella.org/
https://en.wikipedia.org/wiki/Adapteva

- Programming APIs OpenCL, C, pthreads
- Embedded applications
- Energy efficient computing 50 single precision Gflops/Watt
- Latest version has 1024 cores

https://www.parallella.org/blog/
Nvidia Tegra K1 and X1

- http://www.nvidia.com/
- https://en.wikipedia.org/wiki/Tegra#Tegra_K1

- 0.19 Tflops double precision
- Programming APIs CUDA, OpenCL
AMD APU

- http://www.amd.com
- https://en.wikipedia.org/wiki/AMD_Accelerated_Processing_Unit

- 0.700 Tflops single precision
- Programming APIs OpenCL, OpenACC, Fortran, C
- For compute and graphics
Intel HD graphics

http://www.intel.com

- Programming OpenCL
- For compute and graphics
- **Massively Parallel Processor Array** (Kalray, Pezy)
- **Field Programmable Gate Array** (Xilinx, Altera)
Pattern Matching Example

- Opencl parallelization of Naive, Knuth-Morris-Pratt and Boyer-Moore-Horspool pattern matching algorithms
- Code at https://github.com/JaakTree/pattern_matching/tree/test

- Possible project based on work by Handre Elias
  http://kodu.ut.ee/~handre/

- Possible projects related to machine learning
Balras G. “Multicore and GPU programming an Integrated Approach” Morgan Kauffman 2015

Nielsen, F. “Introduction to HPC with MPI for Data Science” Springer 2016

PBD R https://rbigdata.github.io/


