High Performance Computing: Sheet 2

Nils Döring Michael Mardaus Julian F. Rost

13. November 2013

Question 1

Question 2

a)

The sequential fraction of a program is the part which can not be parallelized, such as initiation or termination of the program, communication between parts of the program and time used by synchronizing the different parts of the program running parallel. This sequential fraction can not be parallelized and does not speed up with the number of processors.

Within Amdahl's Law the sequential fraction is fixed and we can not infinetly improve performance.

b)

Gustavsons Law looks at growing sizes of problems. Therefor the fraction of sequential code shrinks while the problem-size is growing. Therefor the speedup can be much greater than assumed by Amdahls Law.

Question 3

a)

Another example of superlinear speedup:
If the algorithm is randomized and seraches a specific element in a datastructure, lets say a tree. The more processors that start to search at different enrtypoints of the tree, the faster they will succeed.

b)

An apparent alternative would be to identify aggregate tasks with elements of `bin_counts`*, so an aggregate task would consist of all increments of* `bin_counts[b]` *and consequently all calls to* `find_bin` *that return* `b`*.*

In this idea we would split up the incrementation by 1 of some variable over multiple cores, whilst the much harder task to find the correct bin for each data piece (that needs to be incremented) remains the bottleneck of the function.