Alexander Schnapp
Max Menges
introhpc02

Intro HPC: Blatt 7

24.11.1014

7.1 Reading: LogP: Towards a Realistic Model of Parallel Computation

The author is intruducing a model, that discribe the bottlenegs of a distributed memory multiprocessor in which processors communicate by point-to- point messages. Therefore it is based on a few of parameters like latency (L), overhead (o) , bandwidth (g) of communication and the number of processes (P) and the assumption of a finite capacity. Furthermore it is generall for different types ofcommunication protocolls or applications.

The author is finding, that for some applications some parameters can get negligible, which makes a simplification of the model possible. He testeted the model on different workloads like the FFT and the LU-Decomposition, to show how the use of the model can lead to effecient parallel applications in practice. In the paper he is also comparing this model to the widely used PRAM and the BSP model, which do not accurately reflect the performance characteristics of suchy systems, in his opinion.

This model is promissing a couple of advantages to other often used models like including assynchrounus algorithms. It is very generall, so it might give a good overview of a system but not very detailed.

7.2 Reading:Roofline - An Insightful Visual Performance Model for Multicore Architectures

In this paper the author introduces a visual computational model for multicore Architecutures called 'roofline' model. It relates the operational intensity (mean operations per byte of DRAM traffic) with an upper bound for performance of a kernel (Attainable GFlops/s). For that 'roofline' is uses the minimum of Peak Floating-Point Performance (constant) and the Peak memory Bandwith multiplied with the operational intensity (line with postiv slope), so wether the problem is compute-bound or memory-bound.

In that way it tells the user what optimizations should he implement and in what order, by pointing out the limiting factor of perfomance. Using micro benchmarks he is testing 4 different cournels to apply this model.

The Roofline model seams to be a very clear and easy to aply way for characterizing multicore architures for to find a first starting point for optimizations.

7.3 n-Body Problem – Partitioning/Communication Design

In our n-Body code firstly the arrays of the accelation and positions and the mass are allocated dynamicly. Then we have the functions 'accelartions' and 'pos_ update' that

update first the accelartation and afterwards the positions of all the particles in every step.

Since the masses are fix it will be the best to broadcast it once at the beginning of the calculation. For the communication to update the positions will have an ring communication. So everyboday sends the new postion to the neighbour till its broadcastet completely.

To hide the latecies we use an unblocking send so the proceses can go on computing while sending the results of the particle before.

To avoid collectiv calls the sum over all other particles in the force calculation starts at i+1 so firstly we make sure no pair is countet twice and secondly the firstly needed particle position is diffrent for every process. In this case you have to consider newtons laws by counting force and anti-force.