文本, 信件

描述已自动生成­­­­­­文本

描述已自动生成

synchron: at first and after calculation ensures elements are used in each thread. M[row\*width+p\*tile+tx], N[(p\*tile+ty)\*width + col]. P\*tile+ty < width load, else 0

Monte Carlo Method: given large dataset and draw identical samples to approx distri. Advantage: Error reduced to 1/sqrtn. Work with parallelism. Find estimation p time faster or reduce error by sqrtp

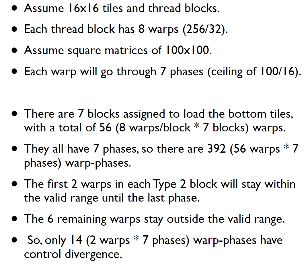
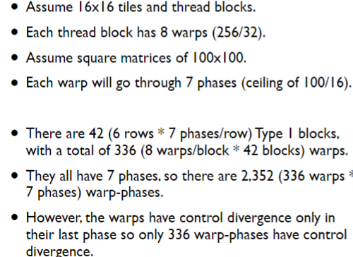
Random Number Generation: Usually generate x and x\_i+1 = (axi+c) mod m

cuRAND: Timing: clock\_t start & stop, in device returns the value of a per-multiprocessor counter that is incremented every clock cycle. Speedup Factor: p = number of pro S(p) = time using one / time using multiple.

Time component: inherently sequential cop, parallel comp, communication. Efficiency: S(p)/ p

Amdahl’s law: just ignores commu time to calculate maximum speed up and set f = portion of sequential time S(p) = p/1+(p-1)f, f = o(n)/o(n)+q(n) (constant problem size)

Gustafson-Barsis’s : predict scaled speedup. S(p) = p+s(1-p), s = o(n)/o(n)+q(n)\p(time constant) “singh” mem const



Histogramming: Interleaving Partition of Input: coalescing and better memory accessing threads runs at same set Thread Coordination: \_\_syncthreads & atomics operations: modifying a value back to memory without the interference of any other threads, can be used in shared and global (expensive) Hierarchical Atomics: per-thread atomic to shared sum, per-block add to total.

文本

描述已自动生成

N-body problem: predicting motion of N objects.

Parallel verlet: making “neighbor list” with rij < rl, only compute force cutoff radius. Use CUDPP scan sorted value in master list get neighbor.

CudaHostalloc: page locked buffers higher bandwidth. Copy mem: cudaMemcpyAsync(stream) overlap memory transfer with kernel execution; Multi GPUs cudaSetDevice(), can assign streams for different gpu PCIe P2P direct mem transfer within gpus cudaMemcpyPeer(). Left-right: no contention for links, no 2 commu in same direction.

Pariwise: better or same 2-gpu base and have contention for 8 or more gpus. (MPI) network used transfer different node pinned mem. Gpu to cpu cpu exchange (MPI\_Sendrecv) cpu exchange to gpu

OpenCL: portability, basic host program initialize device, build program, create buffers, set arguments, enqueue kernel, read back result.

Supervised learning: label training data to make predictions. Unsupervised learning: no label separating into clusters.

Reinforcement learning: reward with submodule and replicate actions. Recognition: segmentation, lighting deformation, viewpoint

Multiclass classification: logistic regression output vectors; replicated feature approach: feature detector; CNN: take tiny patch and do neural network, squeeze dimension to classifier.

OpenACC: minimal barrier but can accelerate. #pragma acc parallel loop collapse(2) copyin, out data into gpu. Device\_type num\_gangs(200)