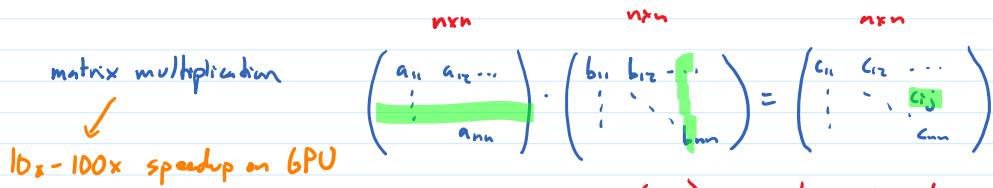
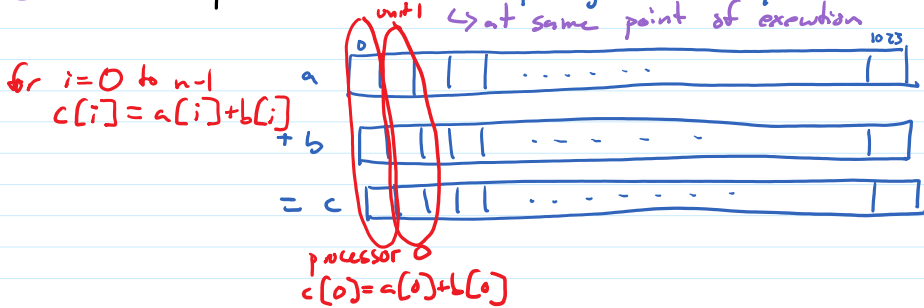


# GPU Programming

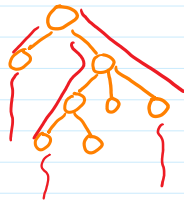
task parallel : different tasks (functions) operating simultaneously on data  
 ↳ each with independent point of execution

GPU → data parallel : same task operating simultaneously on different partition of data



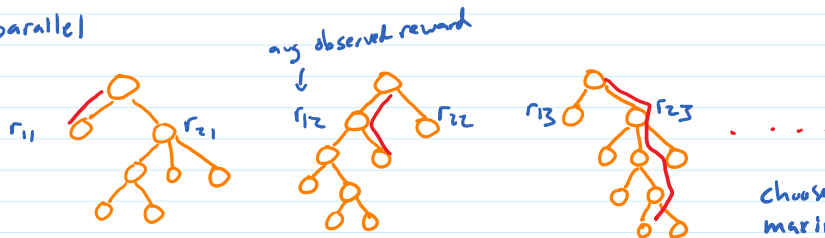
processor  $(i,j)$  computes  $c_{ij}$  by  
 $tot = 0$   
 for  $k=1$  to  $n$   
 $tot += a_{ik} \cdot b_{kj}$   
 $c_{ij} = tot$

Parallel MCTS: tree parallel



requires locks on tree  
 (but for a small percentage of time  
 if playouts take longer than tree updates)

root parallel

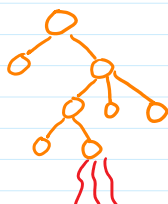


1 tree / thread

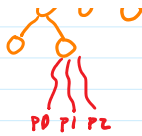
choose child  $i$   
 maximize  $\frac{r_{i1} \cdot n_{i1} + \dots + r_{ik} \cdot n_{ik}}{n_{i1} + \dots + n_{ik}}$

leaf parallel

parallel playouts (data parallel?)



1 tree traversal  
 1 playout from new node / thread



1 tree traversal  
1 ployout from new node / thread

GPU : multiple streaming multiprocessors (SM)  
1-8 (5 for \$750 laptop)

- cache
- + shared memory
- + registers
- + many load/store units
- + many scalar processors (cores, or SPs) 8-192 (128 for \$750 laptop)

OpenCL

NVIDIA CUDA : threads divided into blocks (~512 threads) each mapped to 1 SM

within blocks into warps (32 threads) each thread in warp executes in lockstep

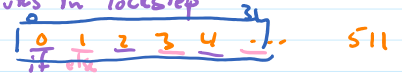
```

if (C)
  A;
else
  B;
  
```

divergence :

```

id
if (x is even)
  e++
else
  odd++
  
```



memory : on GPU [ registers shared (on SM), constant, global ]

fast but small (compiler handles alloc)

large but slow

host system (main system)

matrix multiplication can have 10-100x speedup

leaf parallel MCTS is far from optimal workload

divergence from lengths of playouts

```

0:444440444440 0:444440444440 0:444440444440
↑ ↑ ↑
0:044440444440 0:444440044440 0:440440444440
0:054440444440 0:444440144440 0:440544044440
⋮ ⋮ ⋮
0:055540444440 0:4444401555440 0:440555144440
1:055540444440 1:4444401555440 0:440555144440
⋮ ⋮ ⋮
  
```

← divergence from different sets of legal moves

← divergence from different lengths of loops

← divergence from capture / free move

given starting pos p, distance k, compute index of resulting

$r \leftarrow p+k$   
if  $p \rightsquigarrow r$  gets over opp store  
 $r \leftarrow r+1$   
 $r \leftarrow r \% \text{size}$