

# Pattern Réduction

Nombre opération :  $N-1$

Nombre d'étapes :  $\log(N)$

Utilisation de dynamique parallélisme ? Permet de lancer moins de threads ? Evite les threads qui ne font rien ? Problème de synchronisation ?

```
__shared__ float partialSum[2*BLOCK_SIZE];
unsigned int t = threadIdx.x;
unsigned int start = 2*blockIdx.x*blockDim.x;
partialSum[t] = input[start + t];
partialSum[blockDim+t] = input[start + blockDim.x+t];

for (unsigned int stride = 1;
     stride <= blockDim.x; stride *= 2)
{
    __syncthreads();
    if (t % stride == 0)
        partialSum[2*t] += partialSum[2*t+stride];
}
```

Avec stride :

```
for (unsigned int stride = blockDim.x;
     stride > 0; stride /= 2)
{
    __syncthreads();
    if (t < stride)
        partialSum[t] += partialSum[t+stride];
}
```

Divergence : sur les 5 derniers steps. Dans le cas précédent, seuls les 5 premiers n'avaient pas de divergence