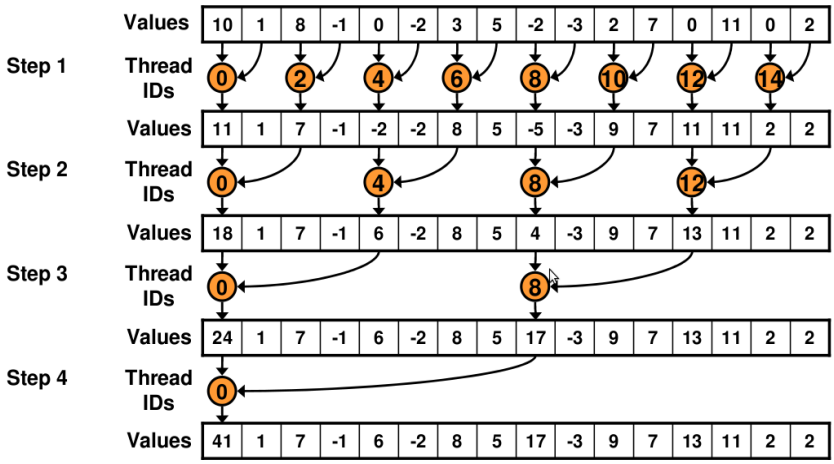
Computación de Altas Prestaciones
Computación Paralela

Ejercicio Tema 3: GPU

1.- Se quiere implementar en CUDA una función de reducción para sumar N números, de modo que se realice la reducción completa, en diferentes pasos, tal y como se muestra a continuación:



La figura es una representación para N=16, pero se debe considerar que se calcula la suma para N=8192, indicando la distribución más adecuada de threads, en una arquitectura que permite 512 threads por bloque y hasta 8 bloques por multiprocesador (Streaming Multiprocesor-SM). Ademas la arquitectura dispone de 16 SM y cada uno de ellos es capaz de ejecutar simultáneamente 24 Warps. Cada warp son 32 threads.

a) Indique, una planificación adecuada de bloques y threads para realizar esta reducción. ¿Cuantos warps se van a utilizar? Indique si se consigue optimizar la utilización de warps por SM ¿Cuantos warp por SM se están planificando?

b) Se pretende disminuir coste de hardware, optimizando la planificación de warps por SM, para conseguir una GPUs con menor coste al disminuir el número de SM. ¿Qué cambios realizaría para conseguirlo? ¿ Cuál es el número de SM mínimo necesario para esta planificación?

c) Para le esquema de reducción propuesto, la parte del código de interés donde se asume el vector ya cargado es la siguiente:

```
__shared__ float partialSum[]

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

Explique el funcionamiento de la reducción propuesta y su comportamiento en términos de eficiencia.

d) Proponga a un esquema alternativo de reducción que sea más eficiente detallando la razón de la mejora en eficiencia.