

Proyecto 1: Introducción a la computación paralela

Cecilia Hernández

April 7, 2022

Fecha inicio: 7 de Abril 2022. Fecha término: 7 de Mayo 2020.

El proyecto tiene como objetivo familiarizarse con el diseño, implementación y análisis de algoritmos paralelos. A continuación se entrega un resumen del modelo PRAM, y se presenta el ejemplo de suma de prefijos, el cual se asocia al problema mas general denominado scan. En el contexto de paralelismo, scan, es un patrón conocido y utilizado en la resolución de otros problemas paralelos.

1. PRAM: Parallel Random Access Machine.

Modelo idealizado de arquitectura de memoria compartida, donde no se consideran costos de acceso a memoria, ni sincronización, ni aspectos técnicos de implementaciones de las plataformas HW y SW donde se ejecutan. Normalmente se usa este modelo para considerar el máximo paralelismo, sin importar aspectos técnicos. Normalmente se usan como cotas inferiores.

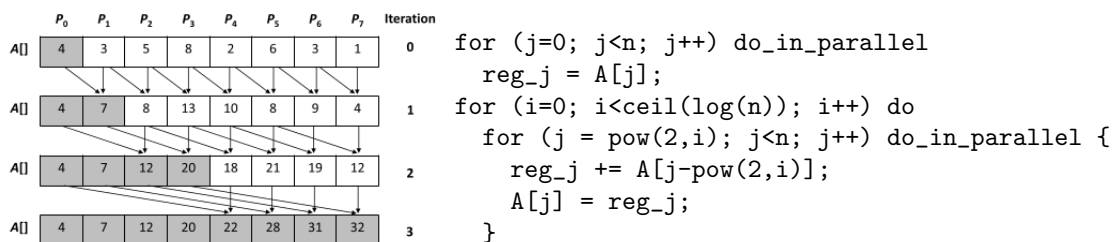
El modelo PRAM considera P procesadores iguales que se coordinan en los siguientes pasos sincronizados: leer, computar, escribir. Las variantes del modelo PRAM incluyen CRCW, EREW, CREW.

2. Suma de prefijos o también conocido como scan.

```
for (i=1; i<n; i++) A[i] = A[i] + A[i-1];
```

Asumiendo que podemos usar $p = n$ procesadores, podríamos querer copiar un valor a cada procesador

Un algoritmo posible:



Cada procesador primero copia un elemento a un registro local. Luego en paralelo, en un primer paso, cada procesador lee de memoria compartida el elemento vecino, computa la suma de lo que tiene con lo que acaba de leer, y escribe el resultado en su registro local. En el siguiente paso, cada procesador lee de memoria compartida el dato que esta a distancia 2^i de él, computa suma, y escribe en registro local, y luego continua con los pasos siguientes hasta llegar a $\log(n)$. Osea el número de pasos del algoritmo es $\log(n)$, y en cada uno de estos pasos los p procesadores se ejecutan en paralelo. En realidad en el primer paso se usan $n - 1$ procesadores, en el siguiente

paso es $n - 2$, luego $n - 4$, luego $n - 8$, y así sucesivamente. Este algoritmo es EREW. La complejidad de tiempo de ejecución es $O(\log(n))$. El costo es $p \log(n) = n \log(n)$

Analicemos esto. Todos los procesadores realizan una suma en paralelo. Y bueno tenemos n procesadores haciendo esto, lo cual parece muy poco trabajo por procesador. Sobre todo este nivel de paralelismo parece sobredimensionado para un laptop con pocas hebras posibles de ejecutar en paralelo. Sin embargo, como sabemos el servidor chome.inf.udec.cl tiene un nivel de paralelismo mucho mayor, dado que tiene 80 hebras.

El proyecto a desarrollar consiste en modificar este algoritmo de manera de disminuir el número de procesadores y ver de qué manera esto afecta al costo y la aceleración alcanzada por el algoritmo modificado en su computador personal y en el servidor chome.

3. En su desarrollo considere los siguientes puntos:

- (a) Proporcionar el algoritmo en pseudocódigo y una figura que muestre su funcionamiento.
- (b) Realizar el análisis correspondiente siguiendo el modelo PRAM y proporcionar el tiempo de ejecución, aceleración y costo asociado.
- (c) Implementar el algoritmo modificado y usando hebras de c++ o bien usando OpenMP.
- (d) Hacer el análisis teórico y experimental proporcionando gráficos que muestren cómo varía el tiempo de ejecución, costo, y aceleración para distintos escenarios donde se varíen los tamaños del arreglo de entrada y el número de hebras.
- (e) Construya un script en python o shell para automatizar la ejecución de los distintos escenarios.

4. Evaluación. La evaluación considera los siguientes componentes:

- (a) [20%] Descripción y análisis de su algoritmo.
- (b) [40%] Implementación y la calidad de sus códigos.
- (c) [30%] Análisis teórico y experimental.
- (d) [10%] Claridad de su informe.