UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA

Report Progetto

SciddicaT

227805 Spagnolo Domenico

Sommario

1. Abstract

Obiettivo principale di questo report è lo sviluppo di diverse versioni parallelizzate di SciddicaT con la libreria CUDA.

2. Introduction

SciddicaT è un modello ad uso scientifico per la simulazione di flussi di fluidi non inerziali. Si basa sul *Cellular Automata Computational Paradigm* e sul *Minimization Algorithm of the Differences* (per il calcolo dei flussi tra celle adiacenti). Nonostante la sua semplicità, il modello è in grado di simulare frane non inerziali su superfici topografiche reali, come quella del Tessina in Italia avvenuta nel 1982, la cui configurazione iniziale viene anch'essa fornita insieme al codice sorgente dell'applicativo.

SciddicaT è definito come:

SciddicaT = <R, X, S, P, σ>

R: è il dominio computazionale bidimensionale, suddiviso in celle di uguale dimensione;

X: è l'intorno di von Neumann (un pattern geometrico che identifica la cella centrale e un insieme di quattro celle poste nelle direzioni nord, ovest, est e sud, adiacenti a quella centrale). Le celle appartenenti all'intorno vengono etichettate con i seguenti valori: 0, 1, 2, 3, 4 mentre i flussi dalla cella centrale verso le quattro adiacenti sono identificate attraverso i seguenti indici: 0, 1, 2, 3.
Il diagramma sottostante mostra gli indici e le etichette di una cella e dei suoi dintorni:

```
Cell format: |flow_index:cell_label:(row_coord,col_coord)|
   cell_label in {0,1,2,3,4}: cell labels
   flow_index in   {0,1,2,3}: outgoing flow indices
       (row_index,col_index): relative cells coordinates


                |0:1:(-1, 0)|
  |1:2:( 0,-1)|_:0:( 0, 0)|2:3:( 0, 1)|
                |3:4:( 1, 0)|
```

S: è l'insieme degli stati della cella. È suddiviso nei seguenti sottostati:

- S_z: è l'insieme dei valori che rappresentano l'altitudine topografica (es. quota s.l.m.);
- S_h: è l'insieme dei valori che rappresentano lo spessore del fluido;
- S_0^4: sono gli insiemi dei valori che rappresentano i deflussi dalla cella centrale alle quattro celle vicine adiacenti.

P = $\{p_\epsilon, p_r\}$: è l'insieme dei parametri che regolano la dinamica del modello. In particolare, p_ϵ specifica lo spessore minimo al di sotto del quale il fluido non può defluire dalla cellula per effetto dell'aderenza, mentre p_r è il parametro del tasso di rilassamento, un fattore di smorzamento del deflusso.

$\boldsymbol{\sigma}$: $S^5 \rightarrow S$ è la funzione di transizione cellulare deterministica. È composto da tre processi elementari, elencati di seguito nell'ordine in cui vengono applicati:

- $\sigma_0: S_0^4 \rightarrow S_0^4$, imposta i deflussi dalla cella centrale alle celle adiacenti a zero.

- σ_1: $(S_z\ x\ S_h)^5\ x\ p_\in\ x\ p_r \rightarrow S_0^4$, calcola i flussi in uscita dalla cella centrale alle quattro celle adiacenti (nord, sud, ovest, est) applicando l'algoritmo di minimizzazione delle differenze
- σ_2: $S_h\ x\ (S_0^4)^5 \rightarrow S_h$, determina il valore dello spessore dei detriti all'interno della cella considerando lo scambio di massa nelle vicinanze della cella.

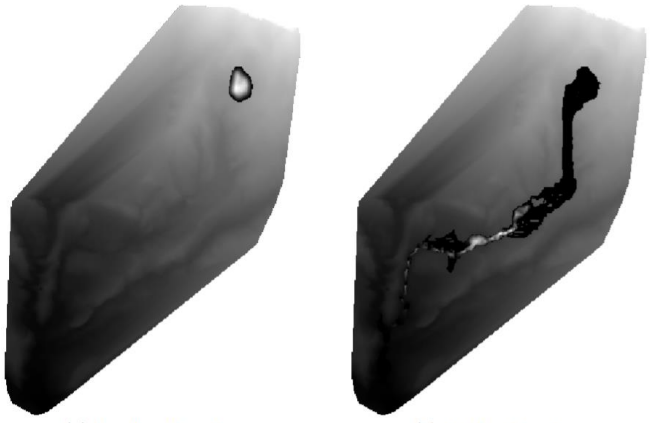
Oltre ai valori da assegnare ai parametri, che sono definiti nel codice sorgente (con $p_\in$ = 0.001 e p_r = 0.5), l'input al modello è rappresentato da tre file di testo in formato Ascii. Il primo (l'intestazione) definisce le dimensioni del dominio (numero di righe e colonne), le coordinate geografiche (della cella in basso a sinistra della griglia), la dimensione della cella e un valore no-data (usato per etichettare le celle con mancanza di informazioni). In realtà, vengono prese in considerazione solo le dimensioni del dominio e i valori no-data. I restanti file (mappe a griglia) rappresentano le informazioni riguardanti l'altitudine topografica e lo spessore del flusso per ogni dominio cellula.

La configurazione iniziale del sistema è definita come segue:

- Una mappa topografica viene letta da un file per inizializzare il sottostato S_z.
- Viene letta una mappa dello spessore di massa da un file per inizializzare il sottostato S_h;
- I livelli del sottostato di deflusso sono inizializzati a zero ovunque.

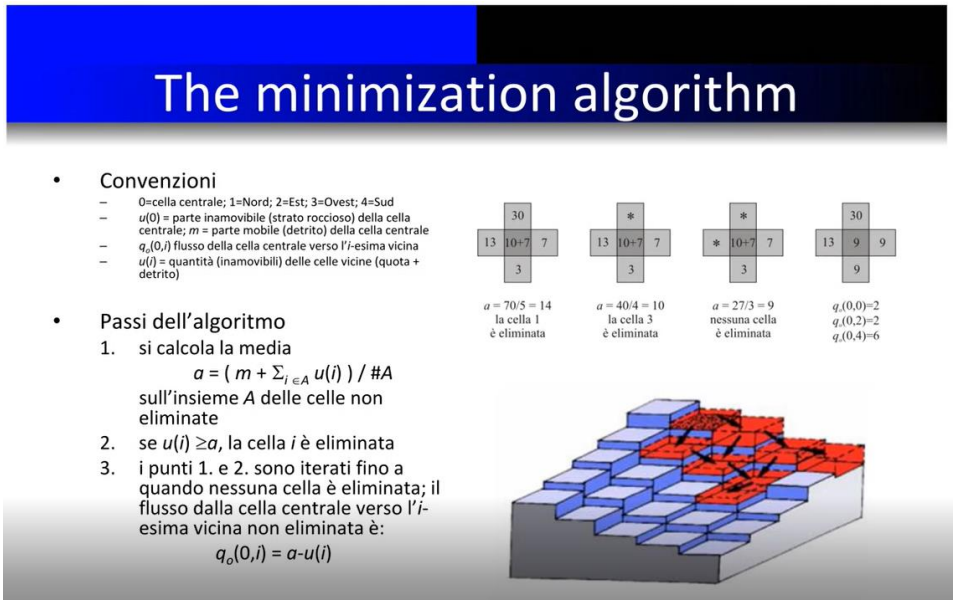
Ad ogni iterazione, sono calcolati tre sottopassi di base, corrispondenti ai processi della funzione di transizione sopra descritti.

- Il kernel *sciddicaTResetFlows* corrisponde al processo elementare σ_0;
- Il kernel *sciddicaTFlowsComputation* corrisponde al processo elementare σ_1;
- Il kernel *sciddicaTWidthUpdate* corrisponde al processo elementare σ_2.



(A) Initial configuration (B) Final configuration

FIGURE 2: Initial and final configuration (after 4000 computational steps) of the system.



3. Parallel Implementations

- Straightforward parallelization
  - *Monolithic*: il kernel non ha modo di gestire il caso in cui il numero di elementi è maggiore del nemero di thread disponibili.
  - *Grid-Stride*: risolve il problema del kernel monolithic con i cicli. In particolare, vengono eseguiti più cicli sui dati. Il passo di un ciclo è *blockDim.x*gridDim.x* che è il numero totale di thread nella griglia. Quindi, se ci sono 1280 thread nella griglia, il thread-0 calcolerà gli elementi 0, 1280, 2560, ecc. Questo è il motivo per cui si chiama Grid-Stride. Usando un ciclo con stride (passo) uguale alla dimensione della griglia, ci assicuriamo la massima coalescenza della memoria, proprio come nella versione monolitica.

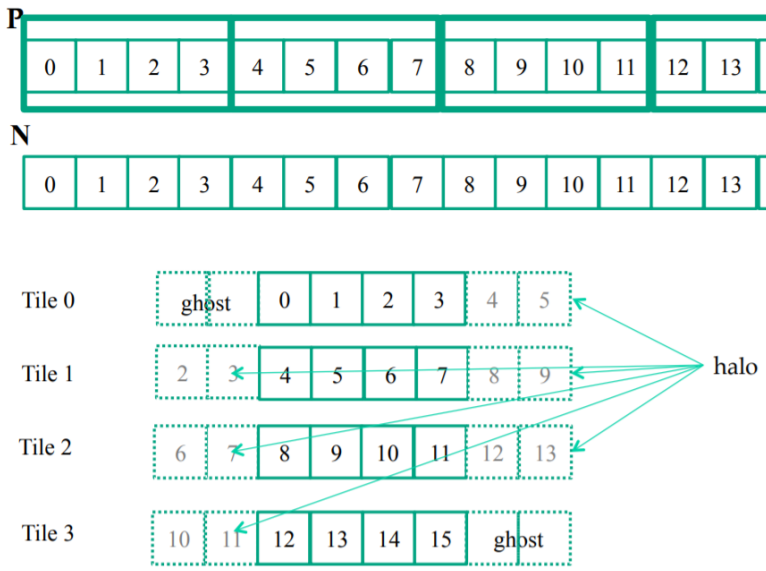
Vantaggi:

- ✓ **Scalabilità e riutilizzo dei thread**. Utilizzando un loop, è possibile supportare qualsiasi dimensione del problema anche se supera la dimensione massima della griglia supportata dal dispositivo CUDA. Inoltre, è possibile limitare il numero di blocchi utilizzati per ottimizzare le prestazioni.
- ✓ **Debugging**. Usando un loop invece di un kernel monolitico, è possibile passare all'elaborazione seriale avviando un blocco con un thread.
- ✓ **Portabilità e leggibilità**. Il codice del Grid-Stride loop è più simile al codice del ciclo sequenziale originale che al codice del kernel monolitico, rendendolo più chiaro per gli altri utenti. In effetti, è possibile scrivere facilmente una versione del kernel che si compila e viene eseguita come kernel CUDA parallelo sulla GPU o come loop sequenziale sulla CPU.

- Tiled parallelization with Halo Cells

  Viene fatto uso della tecnica di Tiling, che consiste nel copiare pezzi (o meglio "piastrelle" quadrate) della matrice principale (memoria dichiarata **__global__**) su mini-matrici condivise (memoria dichiarata **__shared__**). In generale, la tecnica Tiling permette di ridurre gli accessi alla memoria globale (più lenta rispetto alla memoria condivisa) e di velocizzare le operazioni. Tutti i valori della matrice che "escono fuori" dalla "piastrella" (tile) vengono poste a 0 e sono chiamate celle Halo. Solo i thread che hanno entrambi gli indici più piccoli della larghezza della tile sono elaborati.

  Di seguito, un esempio di Tiled 1D Convolution con Halo cells:



- Tiled parallelization without Halo Cells

  Situazione simile alla precedente, ma questa volta non vengono considerate le celle Halo.

- MPI Parallelization

  Con la funzione *updateHalo* vengono aggiornate le celle Halo. Nello specifico si sfruttano le funzioni base della libreria MPI, mpi_Isend e mpi_Irecv, per consentire la comunicazione NON bloccante tra thread e quindi l'aggiornamento.

  Le funzioni *compute_process* ricevono l'input dal processo server data, computano l'algoritmo ed eventualmente inviano indietro i chunk aggiornati al processo server data.

  Prima di tutto, viene allocata sia sull'host che sulla GPU la memoria necessaria per memorizzare il chunk, incluse le halo. Ricevuto il chunk dal server data viene copiato nella GPU. In seguito viene allocata anche la memoria per i dati di output.

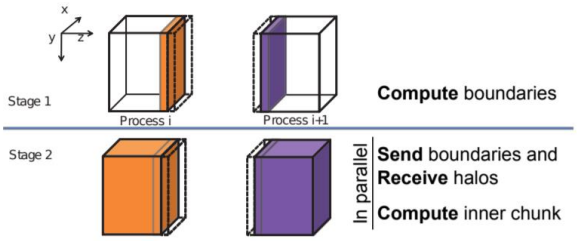
  Le attività di calcolo sono suddivise in due fasi:

  Fase 1.

  - Tutti i processi di calcolo aggiornano le loro celle di confine (sia a sinistra che a destra).

Fase 2.

- Copia delle celle di confine aggiornate nell'host.
- Invio delle celle di confine ai processi adiacenti/ricezione delle celle di confine da processi adiacenti e memorizzazione nelle halo corrispondenti.
- Calcolo della parte interna del blocco.



4. Computational Performance

In questa sezione gli algoritmi sono analizzati dal punto di vista delle performance computazionali.

- Monolithic

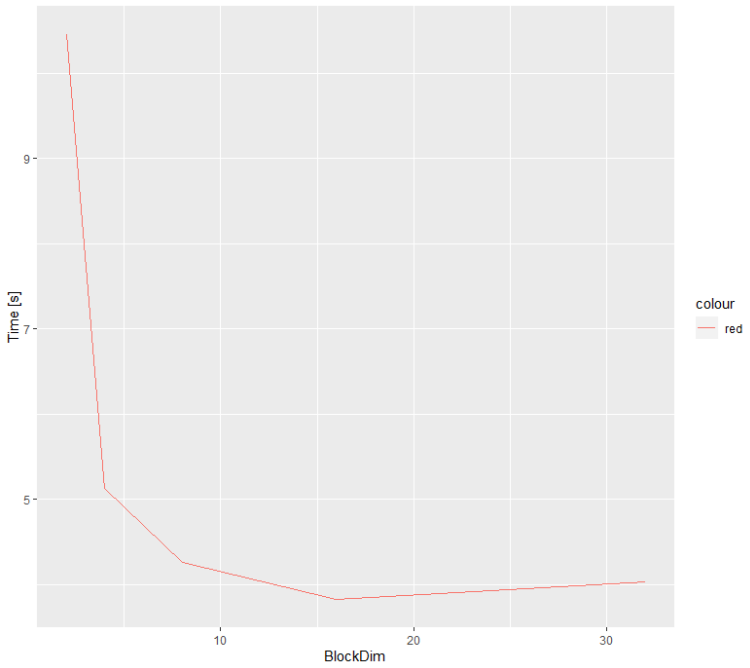
	dimBlock(2,2,1) dimGrid(305,248,1)	dimBlock(4,4,1) dimGrid(153,124,1)	dimBlock(8,8,1) dimGrid(76,62,1)	dimBlock(16,16,1) dimGrid(38,31,1)	dimBlock(32,32,1) dimGrid(19,16,1)
Time [s]	10.45	5.13	3.65	3.94	4.27

SpeedUp

Tempo di esecuzione seriale = 67.5 s

67.5/10.45 = 6.46
67.5/5.13 = 13.16
67.5/3.65 = 18.49
67.5/3.94 = 17.13
67.5/4.27 = 15.81

Plot



Profiling

dimBlock(8,8,1)
dimGrid(76,62,1)

```
==524863== NVPROF is profiling process 524863, command: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.650000 [s]
Releasing memory...
==524863== Profiling application: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==524863== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   41.40%  1.07206s      4000  268.02us  262.34us  342.79us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   34.20%  885.76ms      4000  221.44us  211.68us  236.64us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   24.40%  631.82ms      4000  157.95us  156.13us  456.93us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  15.296us         1  15.296us  15.296us  15.296us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   69.56%  2.70726s     12001  225.59us  9.5310us  466.67us  cudaDeviceSynchronize
                   23.74%  923.95ms     12001  76.989us  70.429us  2.1647ms  cudaLaunchKernel
                    6.62%  257.81ms         5  51.562ms  21.299us  256.36ms  cudaMallocManaged
                    0.04%  1.7270ms         5  345.41us  28.002us  866.07us  cudaFree
                    0.02%  763.23us       202  3.7780us     211ns  172.53us  cuDeviceGetAttribute
                    0.01%  371.67us         2  185.83us  180.90us  190.77us  cuDeviceTotalMem
                    0.00%  76.311us         2  38.155us  33.142us  43.169us  cuDeviceGetName
                    0.00%  27.505us         2  13.752us  4.0790us  23.426us  cuDeviceGetPCIBusId
                    0.00%  1.8580us         4     464ns     249ns     887ns  cuDeviceGet
                    0.00%  1.6900us         3     563ns     276ns     941ns  cuDeviceGetCount
                    0.00%     830ns         2     415ns     348ns     482ns  cuDeviceGetUuid

==524863== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.745600ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.818560ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

dimBlock(16,16,1)
dimGrid(38,31,1)

```
==524933== NVPROF is profiling process 524933, command: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.935000 [s]
Releasing memory...
==524933== Profiling application: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==524933== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   45.67%  1.31917s      4000  329.79us  322.08us  445.92us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   33.08%  955.52ms      4000  238.88us  224.58us  255.20us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   21.26%  614.08ms      4000  153.52us  151.58us  435.07us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  17.344us         1  17.344us  17.344us  17.344us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   71.86%  3.00003s     12001  249.98us  9.5010us  455.17us  cudaDeviceSynchronize
                   21.89%  914.05ms     12001  76.164us  68.770us  2.1510ms  cudaLaunchKernel
                    6.17%  257.69ms         5  51.538ms  21.115us  256.22ms  cudaMallocManaged
                    0.04%  1.7524ms         5  350.48us  26.559us  870.47us  cudaFree
                    0.02%  736.37us       202  3.6450us     207ns  168.39us  cuDeviceGetAttribute
                    0.01%  367.41us         2  183.71us  179.04us  188.37us  cuDeviceTotalMem
                    0.00%  75.566us         2  37.783us  33.434us  42.132us  cuDeviceGetName
                    0.00%  21.835us         2  10.917us  1.7720us  20.063us  cuDeviceGetPCIBusId
                    0.00%  1.6870us         3     562ns     285ns     908ns  cuDeviceGetCount
                    0.00%  1.5730us         4     393ns     238ns     833ns  cuDeviceGet
                    0.00%     942ns         2     471ns     334ns     608ns  cuDeviceGetUuid

==524933== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.742080ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.810048ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

dimBlock(32,32,1)
dimGrid(19,16,1)

```
==525065== NVPROF is profiling process 525065, command: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 4.273000 [s]
Releasing memory...
==525065== Profiling application: ./es1 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==525065== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   55.33%  1.92298s      4000  480.74us  475.68us  925.22us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   27.26%  947.27ms      4000  236.82us  229.47us  276.55us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   17.41%  605.18ms      4000  151.29us  149.41us  171.81us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  27.392us         1  27.392us  27.392us  27.392us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   79.40%  3.58478s     12001  298.71us  9.2570us  934.74us  cudaDeviceSynchronize
                   14.81%  668.47ms     12001  55.700us  36.120us  2.0790ms  cudaLaunchKernel
                    5.73%  258.75ms         5  51.750ms  32.884us  257.36ms  cudaMallocManaged
                    0.04%  1.7332ms         5  346.63us  27.585us  885.97us  cudaFree
                    0.02%  747.39us       202  3.6990us     210ns  165.15us  cuDeviceGetAttribute
                    0.01%  378.17us         2  189.09us  181.99us  196.18us  cuDeviceTotalMem
                    0.00%  107.81us         2  53.906us  33.392us  74.420us  cuDeviceGetName
                    0.00%  21.366us         2  10.683us  3.1680us  18.198us  cuDeviceGetPCIBusId
                    0.00%  2.1010us         4     525ns     261ns     809ns  cuDeviceGet
                    0.00%  1.7680us         3     589ns     234ns     977ns  cuDeviceGetCount
                    0.00%  1.2210us         2     610ns     439ns     782ns  cuDeviceGetUuid

==525065== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.742464ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809856ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

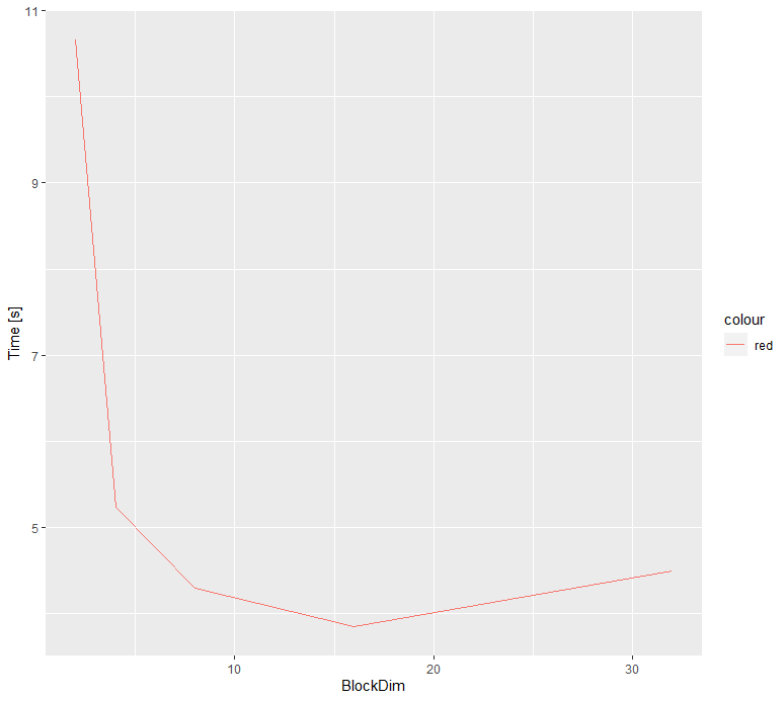
- Grid-Stride

	dimBlock(2,2,1) dimGrid(305,248,1)	dimBlock(4,4,1) dimGrid(153,124,1)	dimBlock(8,8,1) dimGrid(76,62,1)	dimBlock(16,16,1) dimGrid(38,31,1)	dimBlock(32,32,1) dimGrid(19,16,1)
Time [s]	10.66	5.24	3.25	3.53	4.18

SpeedUp

67.5/10.66 = 6.46
67.5/5.24 = 13.16
67.5/3.25 = 20.77
67.5/3.53 = 19.12
67.5/4.18 = 16.15

Plot



Profiling

dimBlock(8,8,1)
dimGrid(76,62,1)

```
==503701== NVPROF is profiling process 503701, command: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.254000 [s]
Releasing memory...
==503701== Profiling application: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==503701== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   42.50%  1.10444s      4000  276.11us  270.34us  303.33us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   33.13%  860.88ms      4000  215.22us  202.21us  236.07us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   24.37%  633.42ms      4000  158.35us  156.55us  467.27us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  16.384us         1  16.384us  16.384us  16.384us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   77.33%  2.70638s     12001  225.51us  9.2570us  476.66us  cudaDeviceSynchronize
                   15.18%  531.08ms     12001  44.253us  42.009us  2.0822ms  cudaLaunchKernel
                    7.41%  259.35ms         5  51.869ms  20.734us  257.98ms  cudaMallocManaged
                    0.05%  1.6945ms         5  338.89us  27.329us  859.41us  cudaFree
                    0.02%  737.37us       202  3.6500us     211ns  169.12us  cuDeviceGetAttribute
                    0.01%  367.09us         2  183.54us  178.64us  188.45us  cuDeviceTotalMem
                    0.00%  74.343us         2  37.171us  32.289us  42.054us  cuDeviceGetName
                    0.00%  19.376us         2  9.6880us  1.9870us  17.389us  cuDeviceGetPCIBusId
                    0.00%  2.0640us         4     516ns     250ns     833ns  cuDeviceGet
                    0.00%  1.4920us         3     497ns     298ns     737ns  cuDeviceGetCount
                    0.00%  1.0440us         2     522ns     301ns     743ns  cuDeviceGetUuid

==503701== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.745504ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.810016ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

dimBlock(16,16,1)
dimGrid(38,31,1)

```
==525183== NVPROF is profiling process 525183, command: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.529000 [s]
Releasing memory...
==525183== Profiling application: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==525183== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   46.71%  1.34420s      4000  336.05us  326.85us  776.77us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   31.79%  914.73ms      4000  228.68us  213.09us  246.88us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   21.50%  618.56ms      4000  154.64us  152.83us  200.93us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  18.240us         1  18.240us  18.240us  18.240us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   79.08%  2.98400s     12001  248.65us  9.4110us  785.42us  cudaDeviceSynchronize
                   14.02%  529.11ms     12001  44.088us  41.960us  2.0761ms  cudaLaunchKernel
                    6.82%  257.34ms         5  51.468ms  24.147us  255.98ms  cudaMallocManaged
                    0.05%  1.7095ms         5  341.91us  27.663us  862.45us  cudaFree
                    0.02%  739.26us       202  3.6590us     210ns  168.35us  cuDeviceGetAttribute
                    0.01%  369.24us         2  184.62us  180.41us  188.83us  cuDeviceTotalMem
                    0.00%  75.771us         2  37.885us  33.025us  42.746us  cuDeviceGetName
                    0.00%  20.436us         2  10.218us  1.8860us  18.550us  cuDeviceGetPCIBusId
                    0.00%  1.5830us         3     527ns     247ns     803ns  cuDeviceGetCount
                    0.00%  1.5100us         4     377ns     250ns     758ns  cuDeviceGet
                    0.00%  1.0710us         2     535ns     346ns     725ns  cuDeviceGetUuid

==525183== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.748576ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809376ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

dimBlock(32,32,1)
dimGrid(19,16,1)

```
==525232== NVPROF is profiling process 525232, command: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 4.177000 [s]
Releasing memory...
==525232== Profiling application: ./es2 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==525232== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   55.53%  1.95350s      4000  488.37us  484.03us  739.75us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   27.28%  959.80ms      4000  239.95us  232.29us  408.19us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   17.19%  604.67ms      4000  151.17us  149.66us  168.22us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  27.776us         1  27.776us  27.776us  27.776us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   81.98%  3.62669s     12001  302.20us  9.3980us  749.58us  cudaDeviceSynchronize
                   12.08%  534.21ms     12001  44.513us  42.523us  2.0745ms  cudaLaunchKernel
                    5.88%  259.95ms         5  51.989ms  18.150us  258.58ms  cudaMallocManaged
                    0.04%  1.7119ms         5  342.39us  26.977us  862.61us  cudaFree
                    0.02%  745.03us       202  3.6880us     213ns  171.61us  cuDeviceGetAttribute
                    0.01%  370.60us         2  185.30us  180.81us  189.79us  cuDeviceTotalMem
                    0.00%  76.763us         2  38.381us  33.097us  43.666us  cuDeviceGetName
                    0.00%  24.805us         2  12.402us  4.0380us  20.767us  cuDeviceGetPCIBusId
                    0.00%  14.008us         4  3.5020us     252ns  12.911us  cuDeviceGet
                    0.00%  1.7400us         3     580ns     267ns     953ns  cuDeviceGetCount
                    0.00%     851ns         2     425ns     349ns     502ns  cuDeviceGetUuid

==525232== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.742912ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809376ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

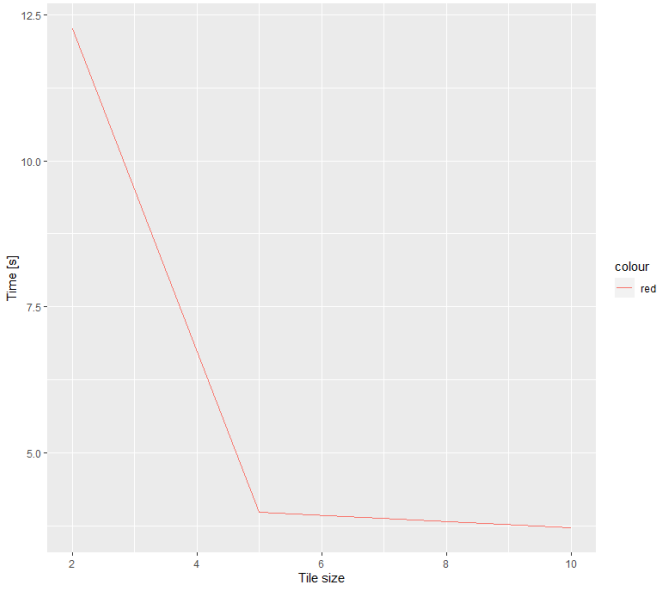
- Tiles without halo

	tile_size = 2	tile_size = 5	tile_size = 10
Time [s]	12.26	3.99	3.79

SpeedUp

67.5/12.26 = 5.51
67.5/3.99 = 16.92
67.5/3.79 = 17.81

Plot



Profiling

Tiles_size = 2

```
==523547== NVPROF is profiling process 523547, command: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 12.257000 [s]
Releasing memory...
==523547== Profiling application: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==523547== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   68.53%  7.64714s      4000  1.9118ms  1.8797ms  2.4290ms  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   26.53%  2.96102s      4000  740.26us  731.97us  1.0653ms  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                    4.93%  550.66ms      4000  137.67us  136.16us  151.07us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  153.34us         1  153.34us  153.34us  153.34us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   90.13%  11.2626s     12001  938.47us  10.370us  2.4385ms  cudaDeviceSynchronize
                    7.79%  973.73ms     12001  81.137us  60.417us  2.1585ms  cudaLaunchKernel
                    2.05%  256.59ms         5  51.317ms  23.323us  255.14ms  cudaMallocManaged
                    0.01%  1.7262ms         5  345.24us  27.396us  869.04us  cudaFree
                    0.01%  738.04us       202  3.6530us     213ns  172.06us  cuDeviceGetAttribute
                    0.00%  365.20us         2  182.60us  177.90us  187.30us  cuDeviceTotalMem
                    0.00%  74.571us         2  37.285us  32.884us  41.687us  cuDeviceGetName
                    0.00%  20.069us         2  10.034us  1.6840us  18.385us  cuDeviceGetPCIBusId
                    0.00%  9.9820us         4  2.4950us     238ns  8.9020us  cuDeviceGet
                    0.00%  1.5610us         3     520ns     277ns     827ns  cuDeviceGetCount
                    0.00%     905ns         2     452ns     316ns     589ns  cuDeviceGetUuid

==523547== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.746976ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809888ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

Tiles_size = 5

```
==523702== NVPROF is profiling process 523702, command: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.990000 [s]
Releasing memory...
==523702== Profiling application: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==523702== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   51.99%  1.52172s      4000  380.43us  374.53us  815.04us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   27.86%  815.38ms      4000  203.84us  193.22us  400.80us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   20.15%  589.60ms      4000  147.40us  146.05us  176.06us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  28.544us         1  28.544us  28.544us  28.544us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   71.65%  3.03151s     12001  252.60us  9.1580us  824.82us  cudaDeviceSynchronize
                   22.17%  937.90ms     12001  78.151us  67.915us  2.1568ms  cudaLaunchKernel
                    6.12%  258.83ms         5  51.766ms  22.925us  257.38ms  cudaMallocManaged
                    0.04%  1.7235ms         5  344.70us  26.800us  876.69us  cudaFree
                    0.02%  744.72us       202  3.6860us     210ns  173.25us  cuDeviceGetAttribute
                    0.01%  368.71us         2  184.36us  180.09us  188.62us  cuDeviceTotalMem
                    0.00%  76.173us         2  38.086us  33.001us  43.172us  cuDeviceGetName
                    0.00%  21.753us         2  10.876us  3.3320us  18.421us  cuDeviceGetPCIBusId
                    0.00%  1.6600us         3     553ns     295ns     878ns  cuDeviceGetCount
                    0.00%  1.6150us         4     403ns     243ns     857ns  cuDeviceGet
                    0.00%     924ns         2     462ns     319ns     605ns  cuDeviceGetUuid

==523702== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.743744ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.810208ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

Tiles_size = 10

```
==523850== NVPROF is profiling process 523850, command: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 3.794000 [s]
Releasing memory...
==523850== Profiling application: ./es4 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==523850== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   51.95%  1.42785s      4000  356.96us  347.90us  787.04us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   25.39%  697.92ms      4000  174.48us  169.06us  216.74us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   22.65%  622.50ms      4000  155.62us  154.05us  461.19us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  18.432us         1  18.432us  18.432us  18.432us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   70.94%  2.86119s     12001  238.41us  9.6150us  796.31us  cudaDeviceSynchronize
                   22.60%  911.43ms     12001  75.946us  60.071us  2.1519ms  cudaLaunchKernel
                    6.39%  257.55ms         5  51.510ms  18.902us  256.10ms  cudaMallocManaged
                    0.04%  1.7205ms         5  344.11us  27.693us  872.27us  cudaFree
                    0.02%  736.11us       202  3.6440us     211ns  171.01us  cuDeviceGetAttribute
                    0.01%  365.64us         2  182.82us  179.40us  186.24us  cuDeviceTotalMem
                    0.00%  75.737us         2  37.868us  33.052us  42.685us  cuDeviceGetName
                    0.00%  22.517us         2  11.258us  3.7170us  18.800us  cuDeviceGetPCIBusId
                    0.00%  1.5630us         3     521ns     276ns     833ns  cuDeviceGetCount
                    0.00%  1.4720us         4     368ns     241ns     731ns  cuDeviceGet
                    0.00%     941ns         2     470ns     325ns     616ns  cuDeviceGetUuid

==523850== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.744096ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809792ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

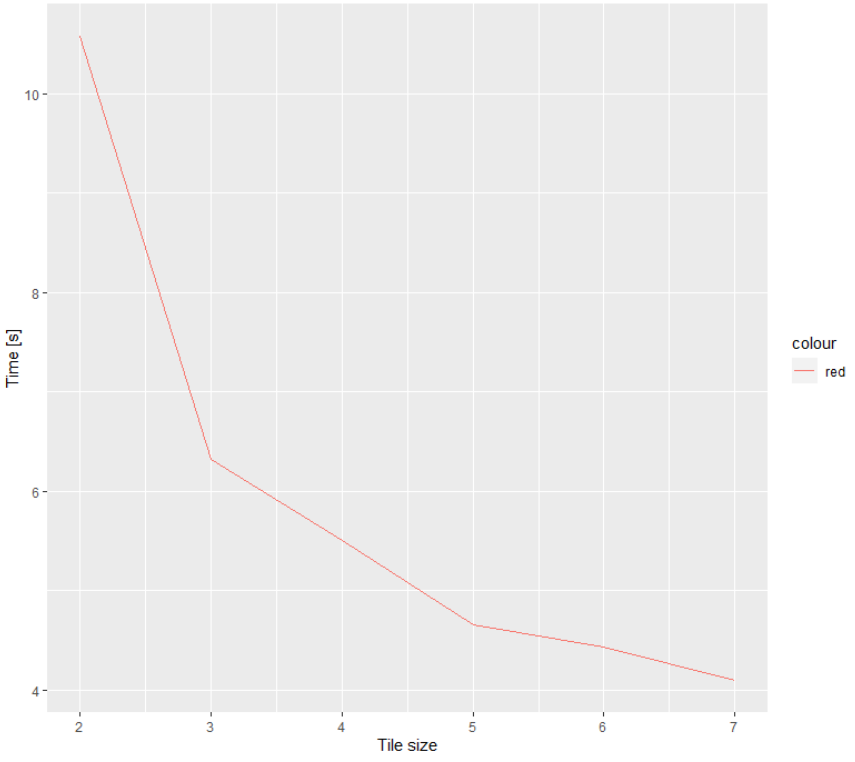
- Tiles with halo

	tile_size = 7 (block_width=9)	tile_size = 6 (block_width=8)	tile_size = 5 (block_width=7)	tile_size = 4 (block_width=6)	tile_size = 3 (block_width=5)	tile_size = 2 (block_width=4)
Time [s]	4.10	4.44	4.66	5.51	6.32	10.58

SpeedUp

67.5/4.10 = 16.46
67.5/4.44 = 15.20
67.5/4.66 = 14.48
67.5/5.51 = 12.25
67.5/6.32 = 10.68
67.5/10.58 = 6.38

Plot



Profiling

mask_size = 3
tile_size = 7
block_width = 9

```
==516877== NVPROF is profiling process 516877, command: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 4.092000 [s]
Releasing memory...
==516877== Profiling application: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==516877== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   55.92%  1.91988s      4000  479.97us  472.13us  979.14us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   25.75%  884.10ms      4000  221.03us  212.51us  237.47us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   18.33%  629.36ms      4000  157.34us  155.07us  427.71us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  21.248us         1  21.248us  21.248us  21.248us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   81.69%  3.54044s     12001  295.01us  9.4290us  988.54us  cudaDeviceSynchronize
                   12.36%  535.70ms     12001  44.638us  42.929us  2.0752ms  cudaLaunchKernel
                    5.89%  255.14ms         5  51.029ms  19.023us  253.79ms  cudaMallocManaged
                    0.04%  1.7154ms         5  343.08us  28.270us  867.07us  cudaFree
                    0.02%  738.47us       202  3.6550us     216ns  169.93us  cuDeviceGetAttribute
                    0.01%  363.98us         2  181.99us  179.04us  184.95us  cuDeviceTotalMem
                    0.00%  85.975us         2  42.987us  32.875us  53.100us  cuDeviceGetName
                    0.00%  20.972us         2  10.486us  3.7840us  17.188us  cuDeviceGetPCIBusId
                    0.00%  1.9210us         3     640ns     255ns     914ns  cuDeviceGetCount
                    0.00%  1.7690us         4     442ns     253ns     866ns  cuDeviceGet
                    0.00%     854ns         2     427ns     346ns     508ns  cuDeviceGetUuid

==516877== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.745792ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.810016ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

mask_size = 3
tile_size = 6
block_width = 8

```
==516616== NVPROF is profiling process 516616, command: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 4.437000 [s]
Releasing memory...
==516616== Profiling application: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==516616== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   55.85%  1.88732s      4000  471.83us  467.01us  970.66us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   25.43%  859.58ms      4000  214.90us  208.90us  232.16us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   18.72%  632.63ms      4000  158.16us  156.67us  180.64us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  20.800us         1  20.800us  20.800us  20.800us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   74.74%  3.49677s     12001  291.37us  9.3080us  979.91us  cudaDeviceSynchronize
                   19.68%  920.83ms     12001  76.729us  58.613us  2.1517ms  cudaLaunchKernel
                    5.52%  258.27ms         5  51.655ms  19.464us  256.81ms  cudaMallocManaged
                    0.04%  1.7022ms         5  340.44us  27.952us  859.63us  cudaFree
                    0.02%  743.30us       202  3.6790us     214ns  172.84us  cuDeviceGetAttribute
                    0.01%  368.38us         2  184.19us  179.97us  188.41us  cuDeviceTotalMem
                    0.00%  77.762us         2  38.881us  32.803us  44.959us  cuDeviceGetName
                    0.00%  19.690us         2  9.8450us  3.6900us  16.000us  cuDeviceGetPCIBusId
                    0.00%  1.5790us         3     526ns     289ns     782ns  cuDeviceGetCount
                    0.00%  1.5580us         4     389ns     246ns     800ns  cuDeviceGet
                    0.00%     912ns         2     456ns     316ns     596ns  cuDeviceGetUuid

==516616== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.744224ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.810240ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

mask_size = 3
tile_size = 5
block_width = 7

```
==516384== NVPROF is profiling process 516384, command: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
Elapsed time: 4.664000 [s]
Releasing memory...
==516384== Profiling application: ./es3 ../data/tessina_header.txt ../data/tessina_dem.txt ../data/tessina_source.txt ./tessina_output_serial 4000
==516384== Profiling result:
            Type  Time(%)      Time     Calls       Avg       Min       Max  Name
 GPU activities:   60.51%  2.42051s      4000  605.13us  596.54us  695.78us  sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
                   24.73%  989.44ms      4000  247.36us  235.65us  403.62us  sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
                   14.76%  590.40ms      4000  147.60us  145.57us  427.81us  sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
                    0.00%  28.000us         1  28.000us  28.000us  28.000us  sciddicaTSimulationInit_Kernel(int, int, double*, double*, int, int, int, int)
      API calls:   83.67%  4.10732s     12001  342.25us  9.6750us  704.94us  cudaDeviceSynchronize
                   11.00%  540.19ms     12001  45.011us  42.625us  2.0628ms  cudaLaunchKernel
                    5.26%  258.23ms         5  51.645ms  17.534us  256.86ms  cudaMallocManaged
                    0.04%  1.7261ms         5  345.23us  26.911us  868.03us  cudaFree
                    0.02%  745.06us       202  3.6880us     210ns  173.99us  cuDeviceGetAttribute
                    0.01%  364.97us         2  182.48us  179.36us  185.60us  cuDeviceTotalMem
                    0.00%  75.289us         2  37.644us  32.601us  42.688us  cuDeviceGetName
                    0.00%  24.501us         2  12.250us  3.9450us  20.556us  cuDeviceGetPCIBusId
                    0.00%  9.7060us         4  2.4260us     241ns  8.9470us  cuDeviceGet
                    0.00%  1.4430us         3     481ns     262ns     730ns  cuDeviceGetCount
                    0.00%     778ns         2     389ns     312ns     466ns  cuDeviceGetUuid

==516384== Unified Memory profiling result:
Device "GeForce GTX 980 (0)"
   Count  Avg Size  Min Size  Max Size  Total Size  Total Time  Name
       5  947.20KB  8.0000KB  1.5586MB  4.625000MB  1.743360ms  Host To Device
      70  102.23KB  4.0000KB  568.00KB  6.988281MB  2.809472ms  Device To Host
Total CPU Page faults: 35
8ed78fa13180c12b4d8aeec7ce6a362a  ./tessina_output_serial
```

5. Roofline Assessment

Il modello Roofline è un modello di performance visuale che consente in maniera intuitiva di stimare le performance di un dato kernel computazionale o di una applicazione che esegue su architetture di calcolo di tipo multi-core mostrando graficamente le limitazioni inerenti all'hardware usato e le potenziali ottimizzazioni da poter applicare, nonché la priorità con cui esse necessitano di essere applicate.

Parametri modello:

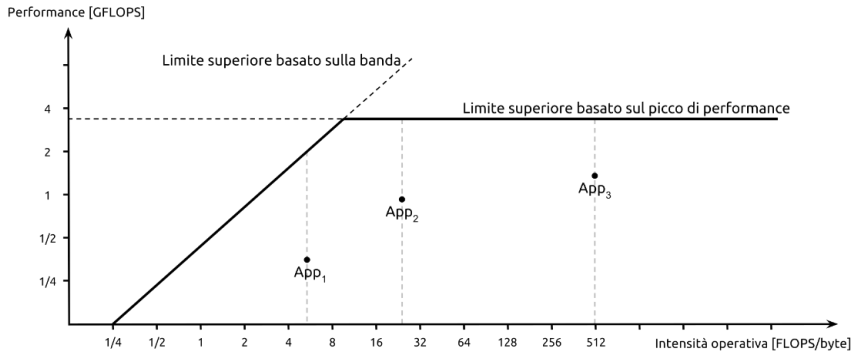
- Un parametro tenuto in considerazione dal modello è l'intensità delle comunicazioni, ciò che viene valutato qui è la **larghezza di banda della memoria**, misurata in Gigabyte per secondo (miliardi di byte trasferiti al secondo). Il motivo per cui viene valutato questo aspetto è dovuto al fatto che la memoria ha una grande incidenza sul tempo di risoluzione di un algoritmo poiché le unità aritmetiche devono operare su dati scritti in memoria. Per cui si ricorre a meccanismi di caching per rendere la fase di lettura e scrittura dei dati da e per la memoria più rapidi. Inoltre, anche le ottimizzazioni dal punto di vista software possono aiutare ad evitare dei cache miss quindi a minimizzare le comunicazioni inutili da e verso le memorie.
- Altro aspetto è la **locazione della memoria**: in caso di memoria distribuite più esse saranno numerose, maggiore è la probabilità di un aumento delle comunicazioni; per questo è necessario applicare le tecniche per favorire la località dei dati.

Il modello Roofline consiste in un grafico/tabella che considera le caratteristiche descritte precedentemente, e il fulcro di questo modello è quello di mettere in relazione i GFLOPs con i GByte/s con una quantità chiamata intensità aritmetica (in letteratura anche definita col nome di Computational Intensity). L'intensità aritmetica è la stima del numero di richieste che un software effettua alla memoria, o meglio è il numero di operazioni in virgola mobile per byte di memoria a cui è richiesto l'accesso alla memoria.

Arithemtic Intensity, Algorithm dependent — Bandwidth, HW dependent

$$FLOPS = \frac{\#FLOP}{\text{time } (Sec)} = \frac{\#FLOP}{Byte} \times \frac{Byte}{Sec}$$

$$FLOPS = AI \times BW$$



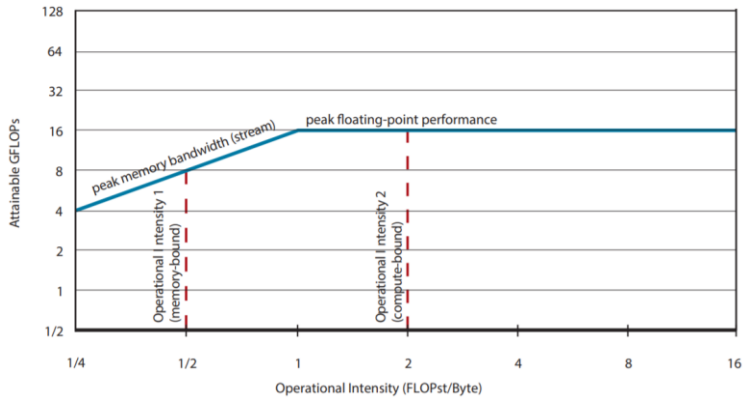
Se un kernel è **memory-bound**, ovvero quando il tempo affinché ci sia un output dal software è pregiudicato principalmente dalla velocità e dalla capacità della memoria, si ha:

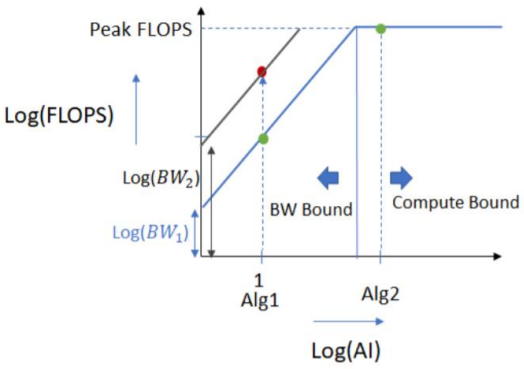
$$AI < \frac{P_{peak}}{B_{peak}}$$

dove P_{peak} è il valore che indica la performance di picco, mentre B_{peak} è il valore che indica il picco del badwidth di memoria.

Mentre un kernel è **compute-bound** si ha che:

$$AI \geq \frac{P_{peak}}{B_{peak}}$$





Specifiche JPDM2

Theoretical performance: 4612 GFLOP/s (device specs)

Theoretical Global Memory bandwidth: 224.32 GB/s (device specs)

Calcolo Intensità aritmetica e applicazione del modello

Per calcolare l'Intensità aritmetica (AI) è possibile usare la seguente formula:

AI = FP / (TR + TW) * tempo kernel (dal precedente passo)

FP = floating point operations

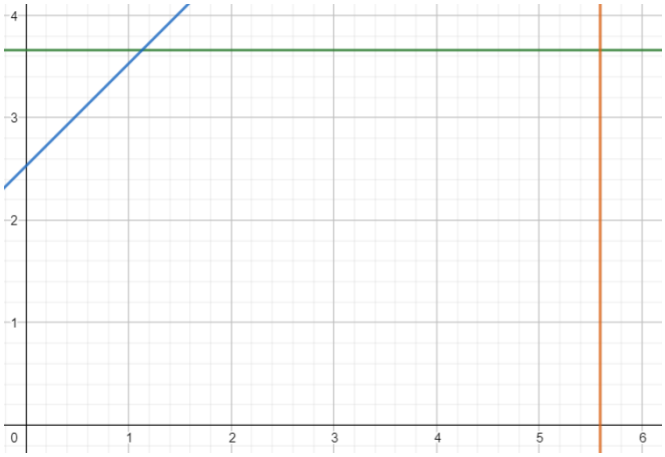
TR/TW= dram read/write throughput

- Monolithic

```
==529589== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)           0           0           0
       1665                        dram_read_throughput                Device Memory Read Throughput  33.193MB/s  295.56MB/s  77.445MB/s
       1665                       dram_write_throughput               Device Memory Write Throughput  55.935GB/s  57.721GB/s  56.718GB/s
       1665                     dram_read_transactions                Device Memory Read Transactions         187        1660         435
       1665                    dram_write_transactions               Device Memory Write Transactions      323059      331039      326675
```

```
==529722== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
       1498                            flop_count_dp   Floating Point Operations(Double Precision)    11458065    11479930    11472791
       1498                        dram_read_throughput                Device Memory Read Throughput  13.592GB/s  14.075GB/s  13.824GB/s
       1498                       dram_write_throughput               Device Memory Write Throughput  24.264GB/s  25.450GB/s  24.840GB/s
       1498                     dram_read_transactions                Device Memory Read Transactions      164764      168921      166744
       1498                    dram_write_transactions               Device Memory Write Transactions      294087      304326      299616
```

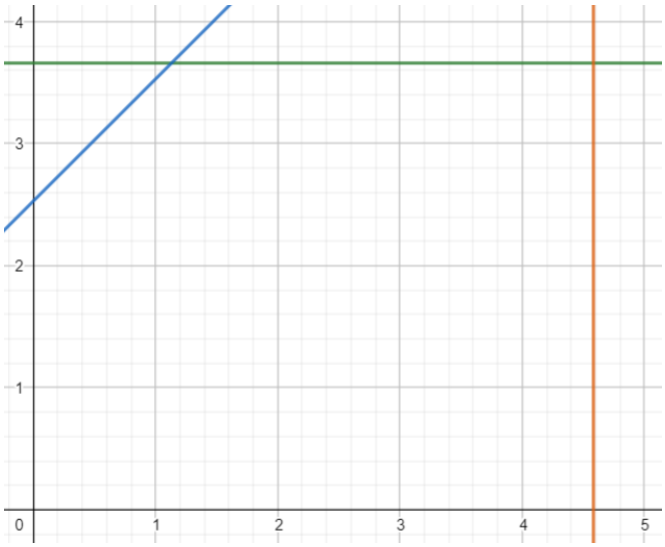
AI = 11.472.791/(13,82 + 24,84) * 1,32 = 391.725 [FLOP/byte]



compute-bound

```
==529890== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)     2402816     2402816     2402816
       1665                        dram_read_throughput                Device Memory Read Throughput  46.881GB/s  52.040GB/s  49.422GB/s
       1665                       dram_write_throughput               Device Memory Write Throughput  9.9767GB/s  11.154GB/s  10.591GB/s
       1665                     dram_read_transactions              Device Memory Read Transactions      386623      412453      399941
       1665                    dram_write_transactions             Device Memory Write Transactions       80345       89197       85708
```

955,52 ms = 0,95552 s
AI = 2.402.816/(49,42 + 10,59) * 0.956 = 38.278 [FLOP/byte]



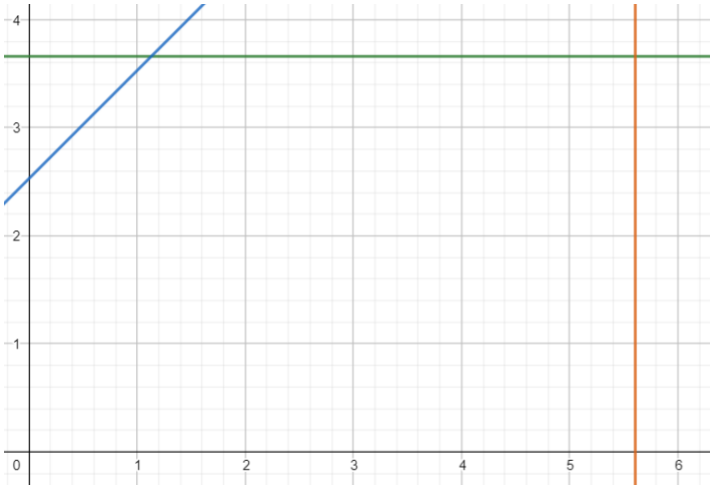
compute-bound

- Grid-Stride

```
==529394== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)           0           0           0
       1665                        dram_read_throughput                Device Memory Read Throughput  33.676MB/s  303.64MB/s  77.837MB/s
       1665                       dram_write_throughput               Device Memory Write Throughput  49.915GB/s  52.732GB/s  52.071GB/s
       1665                     dram_read_transactions              Device Memory Read Transactions         191        1719         440
       1665                    dram_write_transactions             Device Memory Write Transactions      288413      304456      301978
```

```
==530181== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
       1498                            flop_count_dp   Floating Point Operations(Double Precision)    11458065    11479930    11472791
       1498                        dram_read_throughput                Device Memory Read Throughput  13.390GB/s  13.860GB/s  13.624GB/s
       1498                       dram_write_throughput               Device Memory Write Throughput  23.876GB/s  24.893GB/s  24.401GB/s
       1498                     dram_read_transactions              Device Memory Read Transactions      166169      171066      168493
       1498                    dram_write_transactions             Device Memory Write Transactions      296060      306219      301790
```

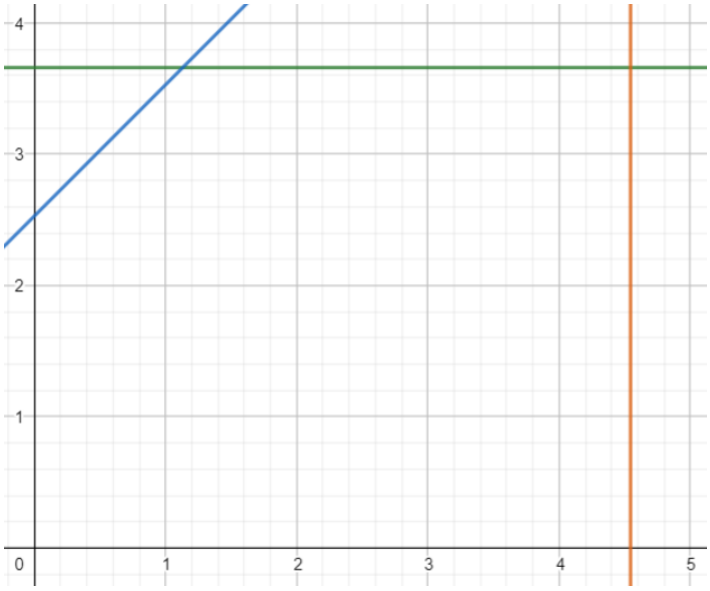
AI = 11.472.791/(13,62 + 24,40) * 1,34 = 404.354 [FLOP/byte]



compute-bound

```
==530262== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
       1498                            flop_count_dp   Floating Point Operations(Double Precision)     2402816     2402816     2402816
       1498                        dram_read_throughput                Device Memory Read Throughput  49.309GB/s  54.519GB/s  51.803GB/s
       1498                       dram_write_throughput               Device Memory Write Throughput  10.421GB/s  11.619GB/s  11.032GB/s
       1498                     dram_read_transactions                Device Memory Read Transactions      390361      415721      401415
       1498                    dram_write_transactions               Device Memory Write Transactions       80884       88486       85482
```

914,73 ms = 0,91473 s
AI = 2.402.816/(51,80 + 11,03) * 0,915 = 34.992 [FLOP/byte]



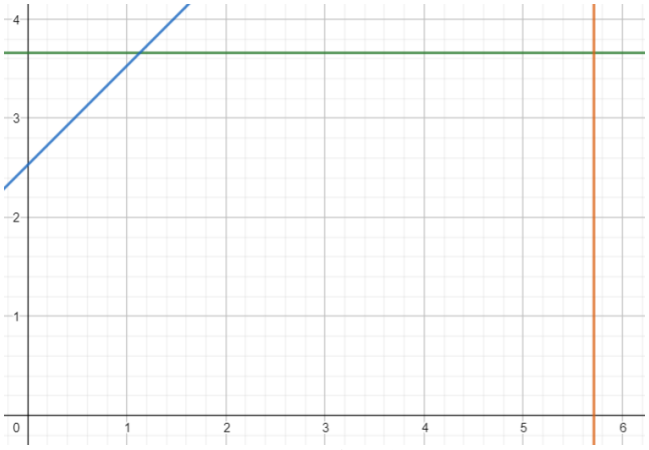
compute-bound

- Tiles without halo

```
==530414== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)           0           0           0
       1665                        dram_read_throughput                Device Memory Read Throughput  32.568MB/s  291.27MB/s  76.009MB/s
       1665                       dram_write_throughput               Device Memory Write Throughput  56.703GB/s  58.123GB/s  57.358GB/s
       1665                     dram_read_transactions                Device Memory Read Transactions         185        1652         431
       1665                    dram_write_transactions               Device Memory Write Transactions      329769      338072      333482
```

```
==530443== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
       1498                            flop_count_dp   Floating Point Operations(Double Precision)    13193885    13219010    13210859
       1498                     dram_read_throughput                 Device Memory Read Throughput  12.501GB/s  12.870GB/s  12.661GB/s
       1498                    dram_write_throughput                Device Memory Write Throughput  23.251GB/s  24.443GB/s  23.859GB/s
       1498                   dram_read_transactions               Device Memory Read Transactions      165033      168542      166615
       1498                  dram_write_transactions              Device Memory Write Transactions      307811      319697      313976
```

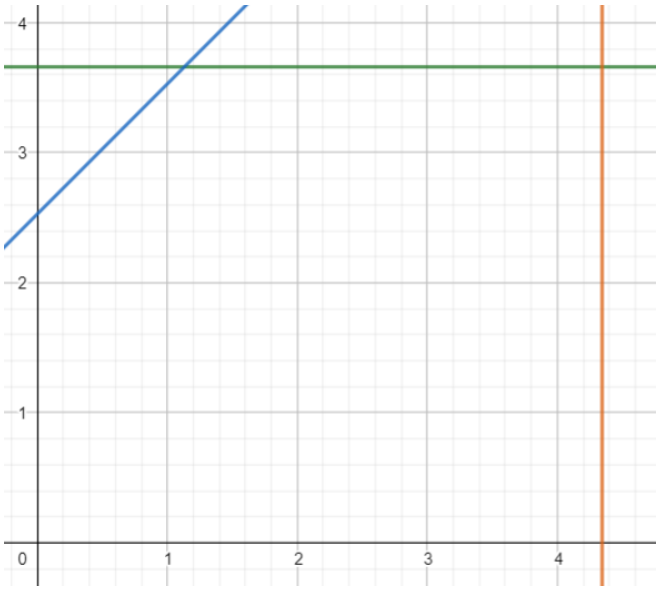
AI = 13.210.859/(12,66 + 23,86) * 1,43 = 517.293 [FLOP/byte]



compute-bound

```
==530468== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)     2402816     2402816     2402816
       1665                     dram_read_throughput                 Device Memory Read Throughput  61.593GB/s  63.865GB/s  62.681GB/s
       1665                    dram_write_throughput                Device Memory Write Throughput  12.559GB/s  13.649GB/s  13.143GB/s
       1665                   dram_read_transactions               Device Memory Read Transactions      385907      394579      390325
       1665                  dram_write_transactions              Device Memory Write Transactions       78009       85194       81846
```

697,92 ms = 0,69792 s
AI = 2.402.816/(62,68 + 13,14) * 0,698 = 22.120 [FLOP/byte]



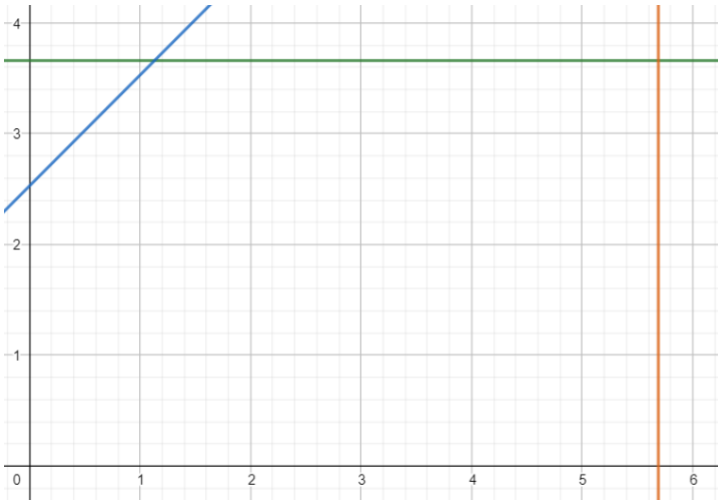
compute-bound

- Tiles with halo

```
==530579== Metric result:
Invocations                               Metric Name                         Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTResetFlows_Kernel(int, int, double, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)           0           0           0
       1665                     dram_read_throughput                 Device Memory Read Throughput  33.402MB/s  292.37MB/s  75.623MB/s
       1665                    dram_write_throughput                Device Memory Write Throughput  58.486GB/s  61.715GB/s  60.637GB/s
       1665                   dram_read_transactions               Device Memory Read Transactions         191        1674         432
       1665                  dram_write_transactions              Device Memory Write Transactions      341835      360819      355064
```

```
==530607== Metric result:
Invocations                               Metric Name                                Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTFlowsComputation_Kernel(int, int, double, int*, int*, double*, double*, double*, double, double, int, int, int, int)
       1498                            flop_count_dp   Floating Point Operations(Double Precision)    13338623    13363748    13355597
       1498                        dram_read_throughput                 Device Memory Read Throughput  20.192GB/s  20.591GB/s  20.389GB/s
       1498                       dram_write_throughput                Device Memory Write Throughput  31.791GB/s  32.305GB/s  32.027GB/s
       1498                     dram_read_transactions                 Device Memory Read Transactions      353200      359500      356436
       1498                    dram_write_transactions                Device Memory Write Transactions      554741      563943      559910
```

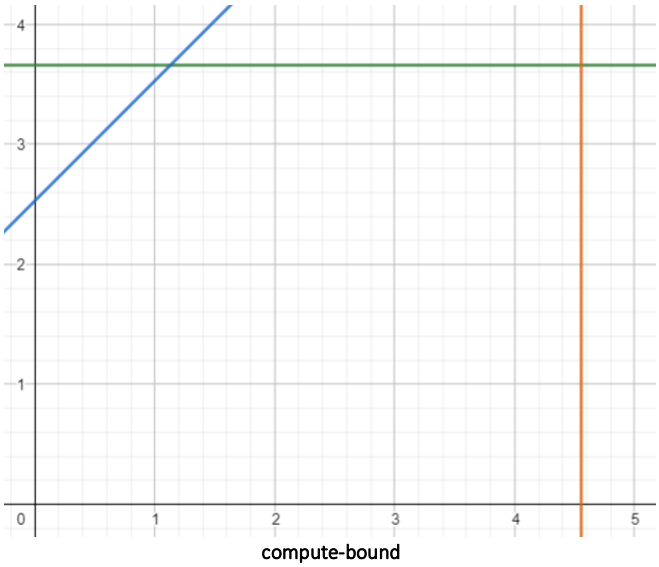
AI = 13.355.597/(20,39 + 32,03) * 1,92 = 489.179 [FLOP/byte]



compute-bound

```
==530639== Metric result:
Invocations                               Metric Name                                Metric Description         Min         Max         Avg
Device "GeForce GTX 980 (0)"
    Kernel: sciddicaTWidthUpdate_Kernel(int, int, double, int*, int*, double*, double*, double*, int, int, int, int)
       1665                            flop_count_dp   Floating Point Operations(Double Precision)     2394008     2394008     2394008
       1665                        dram_read_throughput                 Device Memory Read Throughput  47.288GB/s  48.941GB/s  48.048GB/s
       1665                       dram_write_throughput                Device Memory Write Throughput  10.535GB/s  11.295GB/s  10.924GB/s
       1665                     dram_read_transactions                 Device Memory Read Transactions      374761      378932      376720
       1665                    dram_write_transactions                Device Memory Write Transactions       82595       88408       85650
```

884,10 ms = 0,8841 s
AI = 2.394.008/(48,05 + 10,92) * 0,884 = 35.888 [FLOP/byte]



compute-bound

6. Conclusions

In questo report sono stati analizzati differenti algoritmi di parallelizzazione di SciddicaT in CUDA. Per raggiungere tale obiettivo ci si è serviti di: strumenti di profilazione di programmi CUDA (come *nvprof*),

modelli di performance visuali (come il Roofline Model) e varie misure di performance. La profilazione ha prodotto come output le rispettive statistiche (tempi di esecuzione CPU/GPU, tempi massimi, minimi, ecc.) degli algoritmi e con le opportune opzioni del comando nvprof ha fornito alcune metriche utili alla costruzione del modello Roofline mentre i plot hanno permesso l'analisi anche dal punto di vista grafico. Il Roofline ha fornito informazioni riguardanti la natura dei kernel, e quindi l'appartenenza alla famiglia dei "compute-bound" o dei "memory-bound".