Calcolo parallelo su GPU/Constraint Propagation

Francesco Nascimben

6 dicembre 2019

Indice

1 Stato dell'arte

2 Introduzione

- Descrizione generale del tema (calcolo sequenziale vs calcolo parallelo): potenzialità (migliori prestazioni in certi ambiti) e criticità (non applicabilità a tutti gli ambiti, necessità di utilizzare linguaggi e pattern ad-hoc, accesso condiviso alle risorse) del calcolo parallelo.

3 L'architettura GPU Nvidia CUDA

3.1 Architettura fisica di una GPU Nvidia (SM, memorie)

3.2 Esecuzione di un programma parallelo su GPU (struttura di un kernel, paradigma a warp)

4 Un ambito applicativo: il Constraint Solving

4.1 Definizione di CSP/COP

4.2 Applicazione del calcolo parallelo nel Constraint Solving (Constraint Propagation)

5 Analisi di prestazioni su CPU e GPU

5.1 Somma di un array (con e senza cache)

Somma su CPU e GPU (un solo thread), con e senza cache, tempi misurati.

- Osservazione: le prestazioni su GPU sono peggiori di quanto ci si aspetterebbe basandosi solo sulla differenza nelle frequenze di processori e memorie.

5.2 Somma parallela di molteplici array (misura dello speedup)

Somma di n array, sequenziale su CPU e parallela su GPU (un thread per ogni array). Misure dei tempi e dei rapporti tra tempi CPU e tempi GPU (i.e. speedup).
Descrizione dell'andamento dello speedup (migliori prestazioni registrate al variare del numero di thread, prestazioni con un numero fisso di blocchi, prestazioni con un numero fisso di thread...).

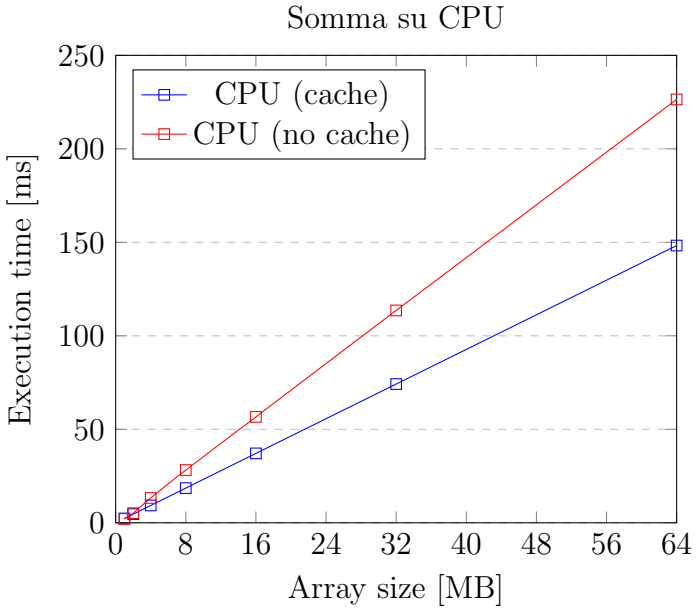
5.3 Simulazione della constraint propagation utilizzando un array di interi

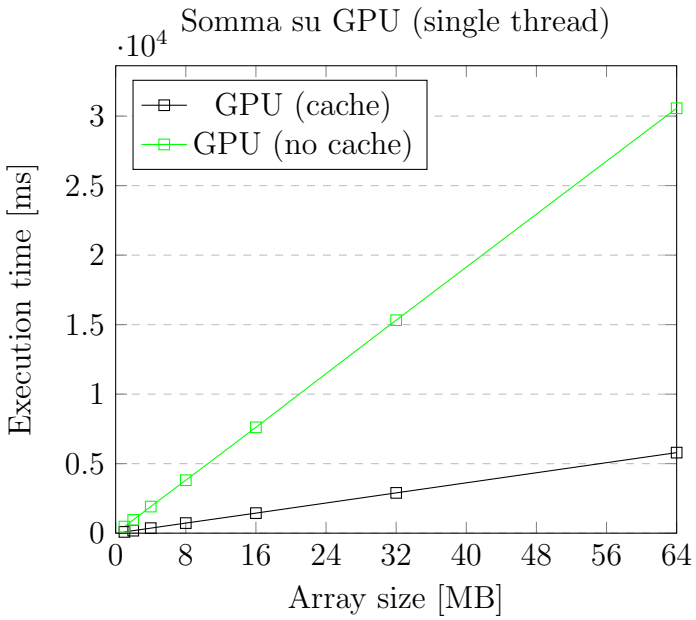
Simulazione di constraint propagation utizzando un array di interi (propagazione di un vincolo = somma delle variabili legate da esso).
Misura dei tempi di propagazione di vincoli binari, ternari, random (4-128 variabili) su CPU e GPU (parallelo), con dimensioni dell'array e numero di vincoli variabili.

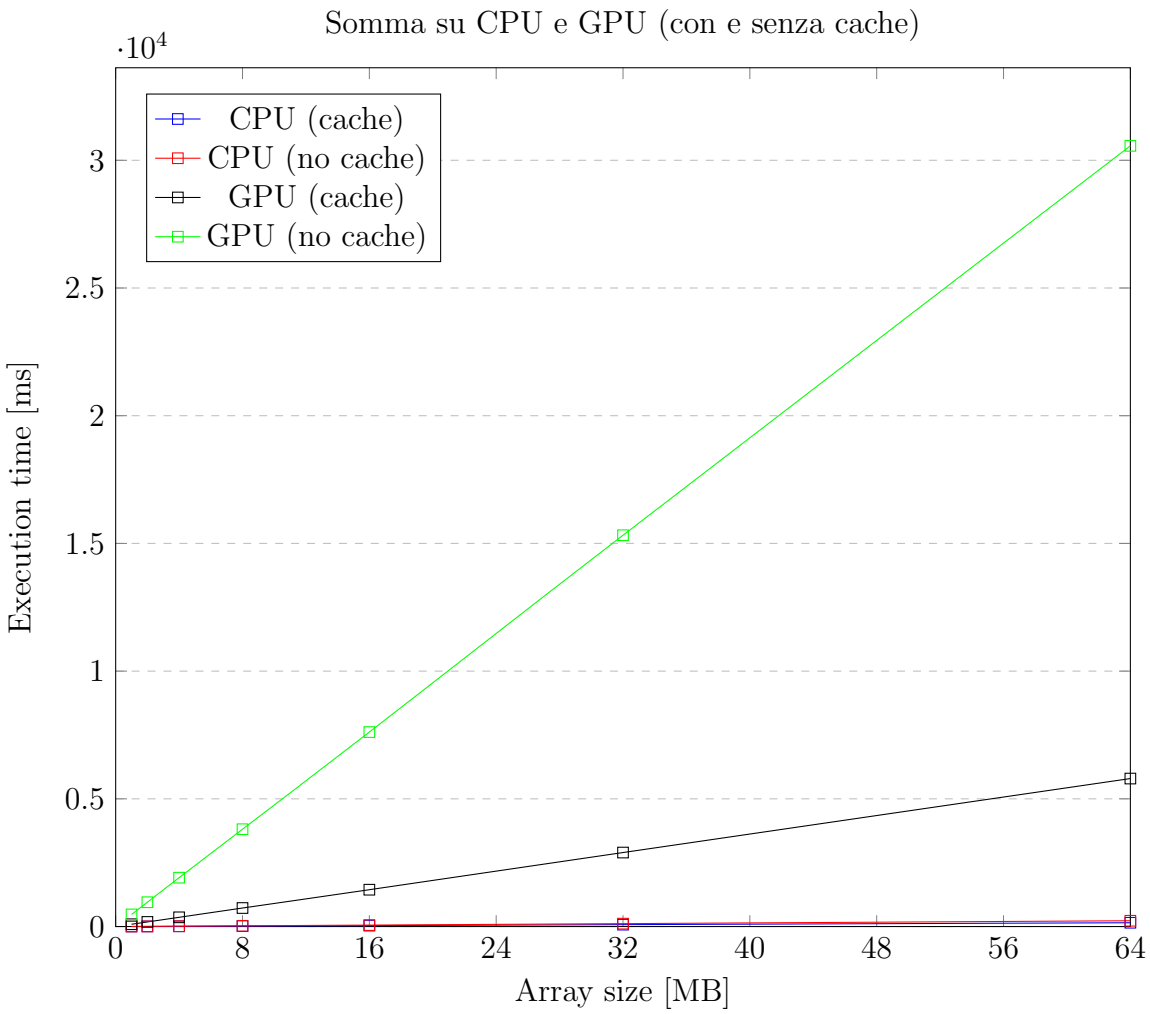
6 Risultati

6.1 Confronto prestazioni CPU e GPU (single thread, con e senza cache)

Somma sequenziale di un array su CPU e GPU (con e senza cache).
Dati tratti da *arraySumCPU_GPU_Results.txt*





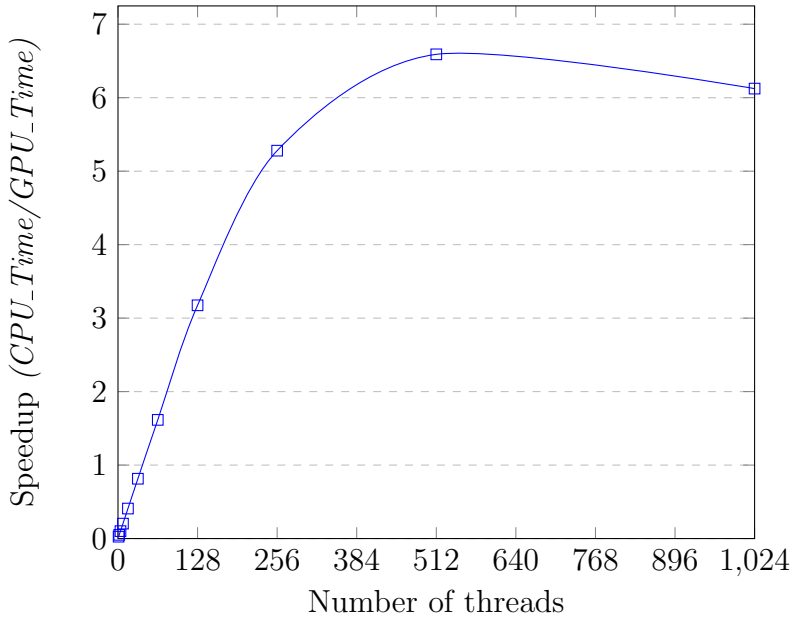

Somma su CPU e GPU (con e senza cache)
$\cdot 10^4$
CPU (cache)
CPU (no cache)
GPU (cache)
GPU (no cache)
Execution time [ms]
Array size [MB]
0
0.5
1
1.5
2
2.5
3
0
8
16
24
32
40
48
56
64

6.2 Confronto prestazioni su CPU e GPU (multithread)

Confronto tra i tempi di esecuzione della somma di n array di interi di dimensione 1 MB su CPU (somma eseguita sequenzialmente) e su GPU (somma eseguita in parallelo, un array per ogni thread).

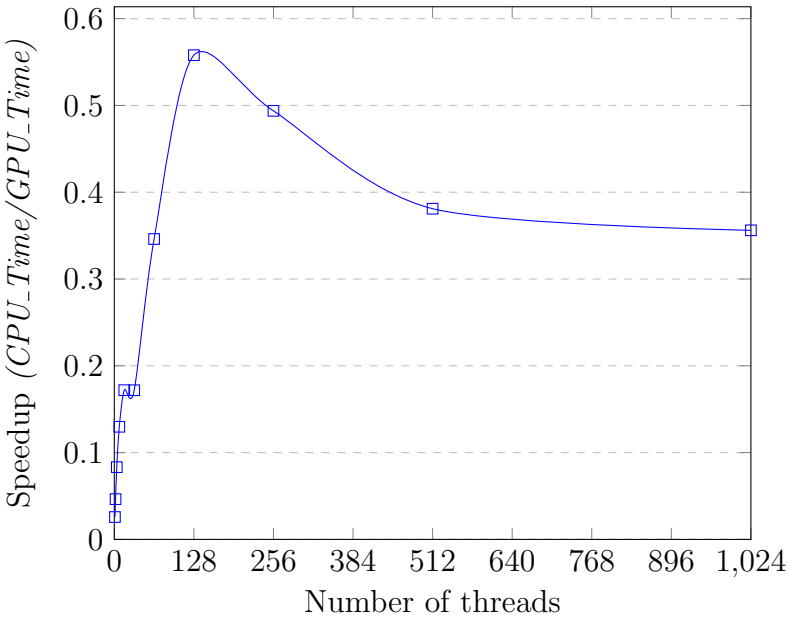
- Sono stati calcolati i tempi di esecuzione e gli speedup per ogni possibile configurazione dei kernel CUDA con al massimo 512 blocchi e 1024 thread totali.
- Dati tratti da *rank_Configuration_Results.txt*.

Rapporto tra i tempi di esecuzione su CPU e su GPU al variare del numero di thread.

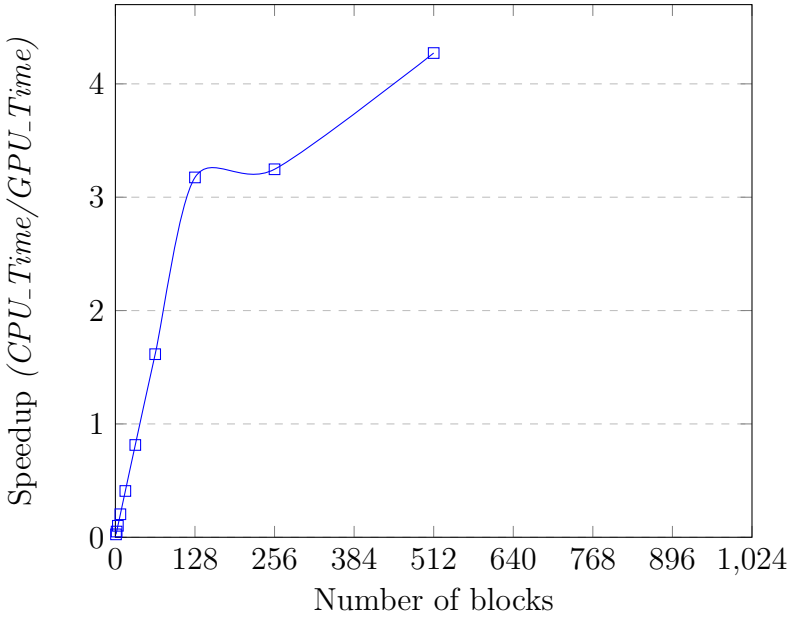


- Nel grafico sono mostrati i migliori speedup osservati in configurazioni con [1,2,4,...,1024] thread totali.

Andamento dello speedup (1 blocco, numero di thread variabile)



Andamento dello speedup (numero di blocchi variabile, 1 thread per blocco)

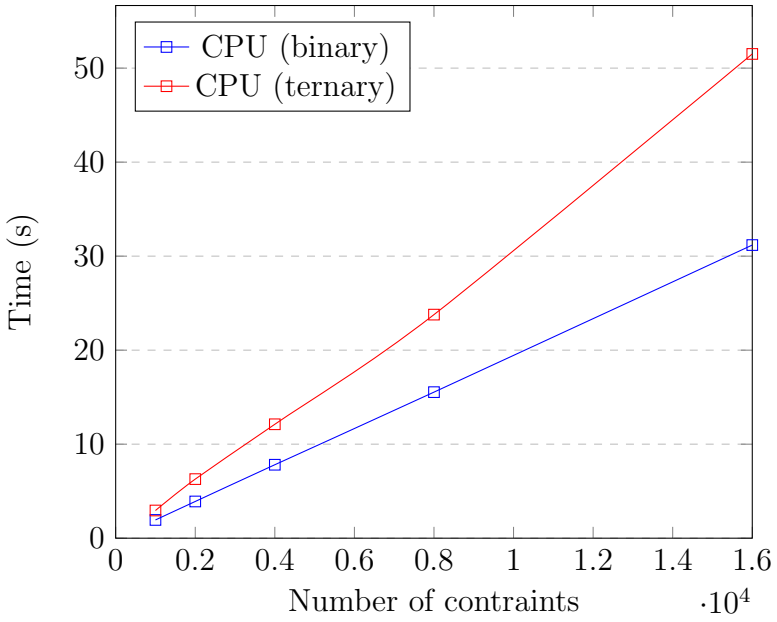


- I due grafici precedenti mostrano la necessità di saturare i multiprocessori (aumentando il numero di blocchi) per sfruttare adeguatamente il parallelismo della GPU(aumentare eccessivamente il numero di thread per blocco porta ad un peggioramento delle prestazioni).

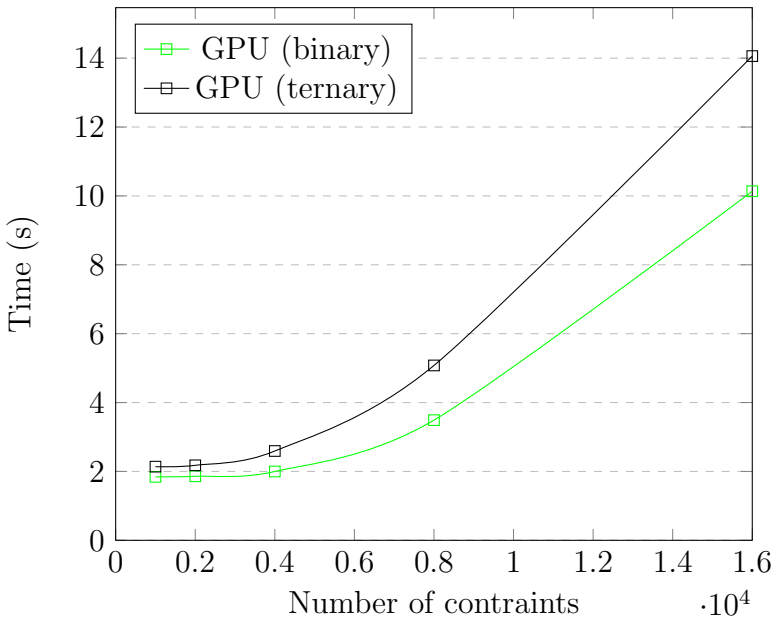
6.3 Constraint Propagation: confronto tra CPU e GPU

Simulazione della constraint propagation usando un array di interi. Andamento dei tempi su CPU (sequenziale) e GPU (parallelo, un constraint per ogni thread) cpn vincoli binari, ternari e di dimensione casuale tra 4 e 128. Tempi misurati effettuando 500K ripetizioni.

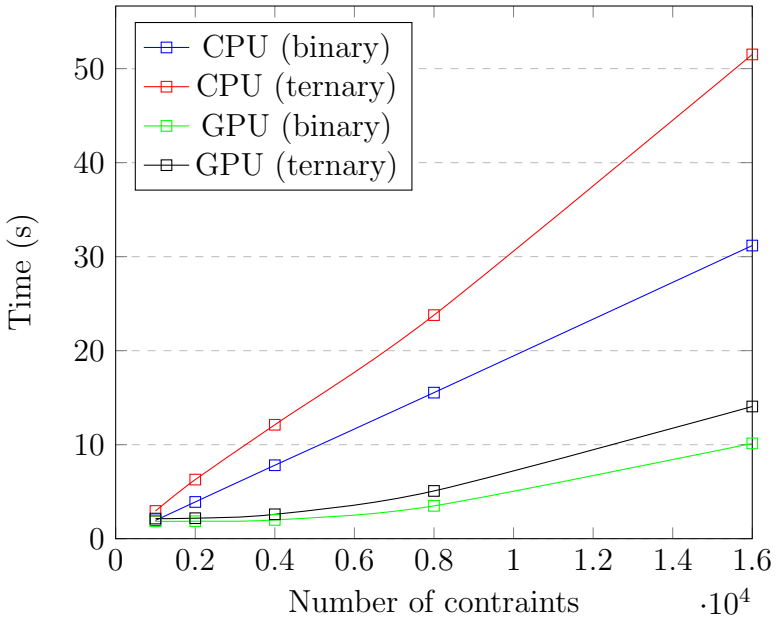
Andamento CPU con vincoli binari e ternari



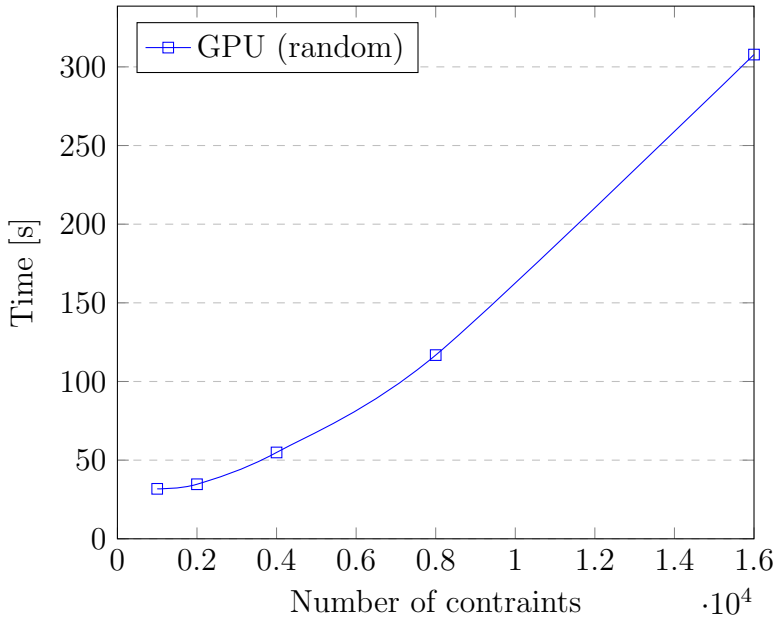
Andamento GPU con vincoli binari e ternari



Andamento CPU e GPU con vincoli binari e ternari



Andamento GPU con vincoli di dimensione casuale tra 4 e 128



- Osservazione: sotto una certa soglia la crescita dei tempi è sublineare rispetto al numero di vincoli.

Andamento CPU e GPU con vincoli di dimensione casuale tra 4 e 128