

# Calcolo Parallelo LT

## Simulazione prova scritta – 2 Dicembre 2019 – 2h

PARTE 1 – RISPOSTA SINGOLA - Ogni domanda ha una sola risposta VERA.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
  - Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
  - Una risposta lasciata in bianco viene valutata 0
1. (3, -.5) I linguaggi OpenMP e MPI
    - a) Non possono coesistere all'interno dello stesso sistema, perché non implementano lo stesso modello di memoria
    - b) Implementano lo stesso modello multi-thread
    - c) Possono coesistere solo all'interno dello stesso processo
    - d) Implementano lo stesso modello a memoria condivisa
  2. (3, -.5) Lo speedup ideale per un programma parallelo che gira su p processori vale:
    - a) p
    - b) 1
    - c) kp
    - d) 1/p
  3. (3, -.5) Un modello fork-join consiste in
    - a) Un modello dove è proibita la sincronizzazione "diretta" fra due thread, utilizzando ad esempio semafori o lock
    - b) Un modello dove un thread ne crea altri, ed è responsabile della sincronizzazione degli altri thread uno ad uno
    - c) Un modello estremamente flessibile, dove la sincronizzazione avviene solo in punti ben definiti, detti lock
    - d) Nessuna delle precedenti

## PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE -

Ogni domanda può avere da una a quattro risposte CORRETTE.

- Ogni risposta esatta viene calcolata: +1
  - Ogni risposta errata viene calcolata: -0.5
  - Una risposta lasciata in bianco viene calcolata: 0
4. Relativamente al GPU computing, segna le affermazioni esatte:
    - a) la GPU è tipicamente una architettura MIMD
    - b) la GPU è tipicamente una architettura SIMD
    - c) il modello di memoria della GPU è UMA
    - d) il modello di memoria della GPU è NUMA
  5. Che tipi di parallelismo esistono?
    - a) a livello di architettura
    - b) a livello di codice
    - c) a livello di sistema
    - d) a livello di istruzione
  6. Nei linguaggi che supportano un paradigma shared memory
    - a) E' possibile la condivisione di dati fra thread paralleli attraverso puntatori condivisi
    - b) E' possibile la condivisione di dati fra thread paralleli solo attraverso file
    - c) Ogni thread può, potenzialmente, accedere ad ogni area di memoria del processo senza limiti di sicurezza
    - d) Non è possibile creare aree di memoria "riservate" a un singolo thread
  7. Le API di programmazione OpenMP
    - a) Hanno in comune con i PThreads i costrutti di sincronizzazione fra thread
    - b) Hanno in comune con CUDA la possibilità di creare barriere "task-based", ovvero basate sul lavoro e non sui thread
    - c) Hanno in comune con PThreads la gestione esplicita della creazione dei thread, seppure questi ultimi non abbiano il concetto di "team di thread"
    - d) Hanno in comune con PThreads la gestione esplicita della creazione e distruzione di "team di thread"
  8. Un programmatore parallelo in un sistema shared memory
    - a) Deve prestare attenzione a non sovradimensionare il numero dei thread, per non sovraccaricare il sistema operativo inutilmente
    - b) Deve conoscere bene l'algoritmo che va a parallelizzare, perché lo speedup che si otterrà dipenderà, fra gli altri fattori, anche dal parallelismo intrinseco dell'algoritmo stesso
    - c) Può dare per scontato la correttezza del programma, quando si appresta a scrivere codice
    - d) Può generalmente evitare di concentrarsi sulla concorrenza nell'accesso in scrittura alle variabili condivise, in quanto questa funzione è tipicamente svolta dal sistema operativo

### PARTE 3 – DOMANDE APERTE

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
  - Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore
  - Una risposta lasciata in bianco viene calcolata: 0
  - L'eventuale sforamento del limite di righe o parole (laddove imposto), porterà a una decurtazione di un punto per ogni riga
  - SI RICORDA CHE L'UNICO FOGLIO DA CONSEGNARE E' IN CALCE AL COMPITO. QUESTO FOGLIO, PUO' SERVIRE ESCLUSIVAMENTE COME "BRUTTA COPIA". EVENTUALI RISPOSTE SCRITTE IN QUESTO FOGLIO NON VERRANNO PRESE IN CONSIDERAZIONE
9. (5 pt) Si descrivano brevemente, in non più di 8 righe, le differenze e le similitudini fra un modello di architettura a memoria *uniform memory access (UMA)* e quello *non uniform memory access (NUMA)*. Eventuali disegni o schematici non verranno tenuti conto nel calcolo delle righe

**10. (6 pt)** Si descrivano i principali costrutti di sincronizzazione di uno qualsiasi dei linguaggi visti nel corso, eventualmente con esempi.

# Calcolo Parallelo LT

## Simulazione prova scritta – 2 Dicembre 2019 – 2h

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche una copia di questa pagina per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (6 pt):

# Calcolo Parallelo LT

## Simulazione prova scritta – 2 Dicembre 2019 – 2h

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_

Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Usate questa copia per calcolare il voto da sole/soli durante la correzione.

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					3	-0.5
2					3	-0.5
3					3	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (5 pt):

Risposta alla domanda 10 (6 pt):