

High Performance Computing

Simulazione di prova scritta – 4 giugno 2019 – 1h30

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. (2, -.5) Il modello di programmazione OpenMP
 - a) Si può definire a tutti gli effetti un linguaggio per la programmazione parallela
 - b) Espone al programmatore la gerarchia di memoria interna del sistema target
 - c) Non si può implementare su una macchina a memoria distribuita
 - d) Richiede, per la sua implementazione, supporto a livello di compilatore e di sistema di runtime
 2. (2, -.5) Lo speedup ideale per un programma parallelo che ha il 20% di istruzioni sequenziali e che gira su un numero p di processori molto alto è:
 - a) 1
 - b) 5
 - c) $1/p$
 - d) p
 3. (2, -.5) Un modello fork-join consiste in
 - a) Un modello dove un thread ne crea altri, ed è l'unico responsabile per la sincronizzazione globale tra tutti i threads
 - b) Un modello dove ogni thread ha visibilità delle aree di memoria accessibili agli altri threads
 - c) Un modello dove un thread ne crea altri (fork), e l'unica forma di sincronizzazione avviene implicitamente al termine del lavoro parallelo (join)
 - 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 SIMD
 - b) la GPU non è tipicamente una architettura MIMD
 - c) il modello di memoria della GPU è tipicamente NUMA
 - d) la gerarchia di memoria della GPU è visibile al programming model (e.g., CUDA)
5. La metodologia di sviluppo HLS per FGPA
- a) può sfruttare lo stesso livello di astrazione di OpenMP per indicare al compilatore le opportunità di identificare parallelismo nel sistema
 - b) non espone al programmatore molti dettagli di come funziona l'hardware
 - c) espone al programmatore molti dettagli di come funziona l'hardware
 - d) richiede una approfondita conoscenza di come funziona l'hardware per ottenere performance analoghe ad un design HDL
6. La forma Static Single Assignment (SSA)
- a) E' implementata da tutti i compilatori
 - b) E' implementata in GCC sopra la rappresentazione intermedia GIMPLE
 - c) Si riesce a implementare senza overhead di memoria
 - d) Semplifica l'analisi e la trasformazione della rappresentazione intermedia di un compilatore
7. Le API di programmazione CUDA
- a) Richiedono il partizionamento esplicito del lavoro tramite identificazione del thread corrente all'interno di un blocco e di una griglia
 - b) Hanno in comune con PThreads la gestione esplicita della creazione e distruzione di "team di threads"
 - c) Non contemplano primitive di sincronizzazione
 - d) Gestiscono in maniera trasparente la località dei dati
8. Un programmatore di un sistema shared memory
- a) Può dare per scontata la correttezza del programma parallelo se modifica un programma C sequenziale preesistente, perché la parallelizzazione non può alterarne la semantica
 - 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 generalmente evitare di concentrarsi sulla concorrenza nell'accesso in scrittura alle variabili condivise, in quanto questa funzione è tipicamente svolta dal sistema operativo
 - d) Deve prestare attenzione a non sovradimensionare il numero dei thread, per non sovraccaricare il sistema operativo inutilmente

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. **(8 pt)** Si descrivano brevemente (una decina di righe) i blocchi principali che costituiscono l'hardware di una FPGA.

10. (8 pt) Si descriva brevemente (una decina di righe) la struttura di un compilatore moderno, spiegando le responsabilità principali dei vari blocchi costituenti.

High Performance Computing

Simulazione di prova scritta – 4 giugno 2019

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					2	-0.5
2					2	-0.5
3					2	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (8 pt):

Risposta alla domanda 10 (8 pt):

High Performance Computing

Simulazione di prova scritta – 4 giugno 2019

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					2	-0.5
2					2	-0.5
3					2	-0.5
4						
5						
6						
7						
8						

Risposta alla domanda 9 (8 pt):

Risposta alla domanda 10 (8 pt):