

Progettazione di sistemi distribuiti

Valutazione delle prestazioni: cenni

Performance



Cosa vuol dire che un sistema è più veloce di un altro?

- **Tempo di risposta (tempo di esecuzione):** differenza tra T_c l'istante in cui un task viene completato, e T_s l'istante in cui un task viene iniziato
- **Performance** è l'inverso del tempo di risposta
- **Throughput:** numero di task al secondo che un sistema è capace di eseguire
 - Un utente del sistema è interessato al tempo di risposta
 - Il gestore del sistema è interessato al throughput
- Il tempo di risposta ed il throughput sono legati
 - un basso tempo di risposta implica un alto throughput
 - se il sistema lavora con un carico di lavoro maggiore o uguale al throughput, il tempo di risposta può aumentare indefinitamente

Make the common case fast!



- **Principio:** quando si fanno scelte di progetto, si favorisca il caso più frequente rispetto a quello meno frequente
- **Vantaggi**
 - si migliorano le prestazioni
 - si spendono le risorse in modo più efficiente
 - il caso più frequente è spesso quello più semplice e può essere eseguito più velocemente di quello meno frequente
- **Esempio.** Nella progettazione di un circuito per la somma di due numeri ottimizzare il caso "assenza di overflow" rispetto a quello "presenza di overflow"

Make the common case fast!



- L'applicazione di questo principio richiede di rispondere a due domande:
 - qual è il caso più frequente?
 - di quanto le prestazioni migliorano rendendo questo caso più veloce?
- La **legge di Amdhal** permette di quantificare questo principio



- **Legge di Amdhal**

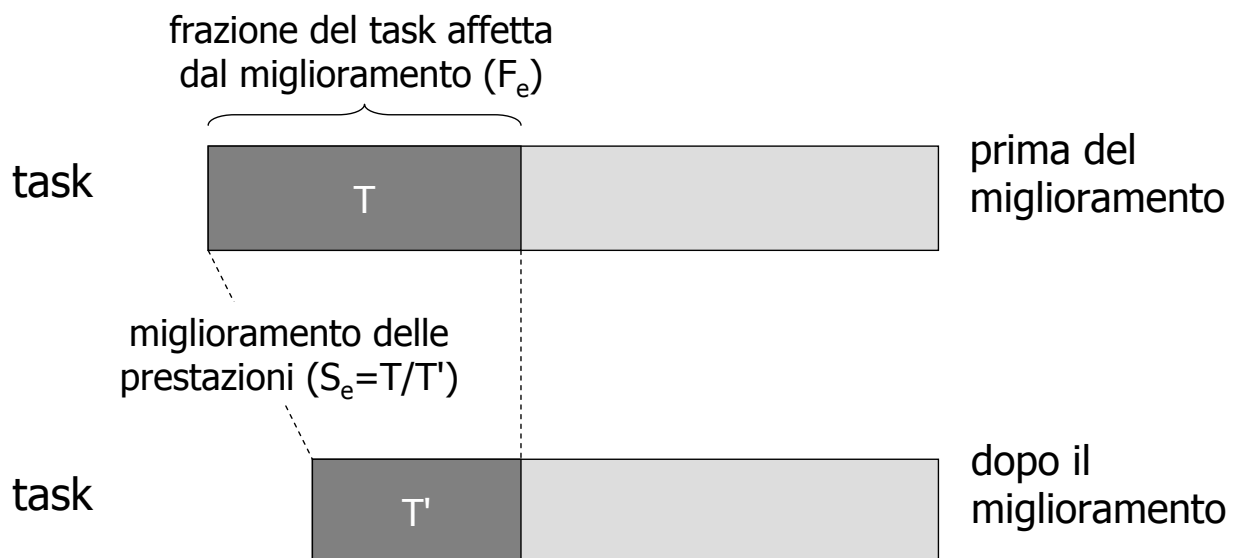
i miglioramenti delle prestazioni che possono essere ottenuti usando un miglioramento sono limitati dalla frazione di tempo in cui tale miglioramento può essere applicato

(*) un modo di esecuzione più veloce

- La legge di Amdhal definisce lo **speedup**

$$Speedup = \frac{\text{performance for entire task using enhancement}}{\text{performance for entire task without using enhancement}}$$

$$Speedup = \frac{\text{Execution time for entire task without using enhancement}}{\text{Execution time for entire task using enhancement}}$$





- **Sistema originale** o **sistema vecchio**: il sistema **prima** del miglioramento
- **Sistema nuovo**: il sistema **dopo** il miglioramento
- Sia **F_e (Fraction enhanced)** la frazione del tempo di esecuzione del sistema originale che può essere convertito per avvantaggiarsi del miglioramento
- Sia **S_e (Speedup enhanced)** l'incremento delle prestazioni dovuto al miglioramento, cioè quanto più velocemente il task sarebbe eseguito se il miglioramento potesse essere esteso a tutto il task



T_e^{old}, T_e^{new} tempo di esecuzione del task prima e dopo, rispettivamente il miglioramento

$$T_e^{new} = T_e^{old} \times \left((1 - F_e) + \frac{F_e}{S_e} \right)$$

$$\text{Speedup totale} = S_o = \frac{T_e^{old}}{T_e^{new}} = \frac{1}{(1 - F_e) + \frac{F_e}{S_e}}$$



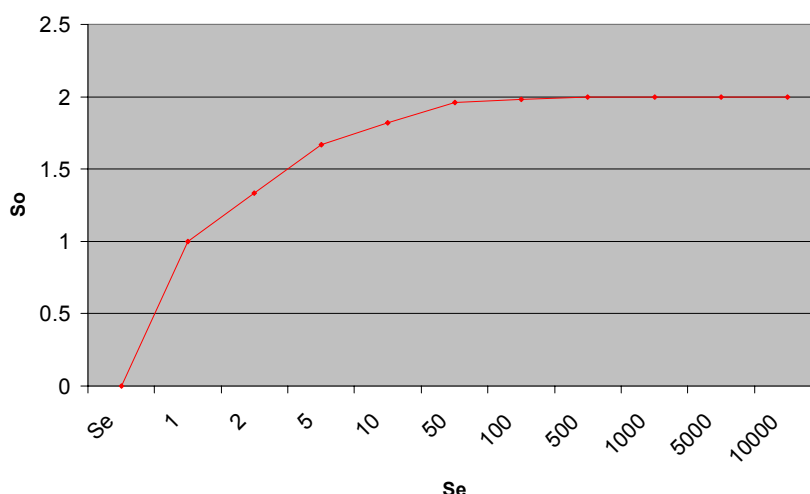
ESEMPIO

- Si consideri un miglioramento che gira 10 volte più veloce della macchina originale ma che può essere usato solo nel 40% delle volte. Determinare lo speedup globale S_o
 - $S_e = 10$
 - $F_e = 0.4$
 - $S_o = 1/(0.6 + 0.4/10) = 1/0.64 = 1.56$

Legge di Amdhal: corollario



S_o in funzione di S_e a parità di F_e



$$\frac{1}{(1 - F_e)}$$

- A parità di frazione F_e , il guadagno incrementale di prestazioni dovuto a miglioramenti addizionali è decrescente
- Un miglioramento che può essere utilizzato solo per una frazione F_e del task non può dare uno speedup maggiore di $1/(1 - F_e)$

Legge di Amdhal: esercizio



L'implementazione della radice quadrata in floating-point (FPSQR) ha prestazioni molto variabili

Supponendo che FPSQR sia responsabile del 20% di un benchmark si confrontino le due alternative di progetto

Alternativa FPSQR. Aggiungere un FPSQR hardware che incrementa le prestazioni di 10 volte

Alternativa FP. Rendere le istruzioni FP più veloci. Le istruzioni FP sono responsabili del 50% del tempo di esecuzione del benchmark. Il team di progetto ritiene di poter rendere le istruzioni FP 2 volte più veloci con lo stesso sforzo necessario per FPSQR hardware

Legge di Amdhal: esercizio



Per confrontare le due alternative di progetto si confrontano i relativi speedup

$$Speedup_{FPSQR} = \frac{1}{(1 - 0.2) + \frac{0.2}{10}} = \frac{1}{0.82} = 1.22$$

$$Speedup_{FP} = \frac{1}{(1 - 0.5) + \frac{0.5}{2}} = \frac{1}{0.75} = 1.33$$

Migliorare le prestazioni delle istruzioni FP è globalmente meglio a causa della loro maggiore frequenza