

## Progetto di Tecnologie e Architetture per la Gestione dei Dati – A.A. 2022/2023

Lo scopo del progetto è quello di modellare le prestazioni di un DBMS sulla base del vostro computer e del benchmark TPC-H. Redigere una relazione che comprenda i seguenti punti (ad ogni passo, identificare chiaramente il collo di bottiglia del sistema):

1. Analisi delle prestazioni della macchina reale (tempo di esecuzione delle query, tempo di servizio e utilizzazione dei centri) con numero massimo di query concorrenti pari a 1 ed il container docker configurato per esporre soltanto una CPU.
2. Confronto delle prestazioni osservate sulla macchina reale rispetto a quelle predette da un modello simulativo.
3. Utilizzo del modello per predire le prestazioni (tempo di esecuzione delle query, utilizzazione dei centri, throughput) con un numero massimo di query concorrenti pari a 2 e dello stesso tipo, e verificare e discutere eventuali scostamenti rispetto alle prestazioni ottenute eseguendo il benchmark con questa configurazione sulla macchina
4. Quali sono le prestazioni predette dal modello nel caso in cui vengano utilizzate 2 CPU ed un numero massimo di query concorrenti che varia tra 1 e 5? Rispecchiano le prestazioni effettive riscontrate eseguendo il benchmark con questi parametri?
  - a. Dimensionare il sistema in modo tale che il tempo di risposta medio sia inferiore ai 30 secondi per un numero di query concorrenti massimo non superiore a 20, assicurandosi che l'utilizzazione sia nell'intervallo 60%-70%. Il sottosistema di storage deve essere implementato con un RAID 10.
5. Dimensionamento di un sistema che risponda ai seguenti requisiti:
  - a. Il sistema è soggetto ad un workload costituito da un mix di 5 query: *A, B, C, D, E*
  - b. Le query *A, B, D, E* esibiscono tassi di arrivo esponenziali  $\lambda_a = \lambda_b = \lambda_d = \lambda_e = 2 \text{ req/sec}$ ; la query *C* invece è di tipo *bursty*, caratterizzata da due intervalli:
    - Intervallo A, si verifica con probabilità 0.9 ed ha una durata campionata da una distribuzione esponenziale con media 30 secondi. Durante questo intervallo, il tasso di arrivo è di tipo esponenziale con parametro  $\lambda_c = 2 \text{ req/sec}$ .
    - Intervallo B, si verifica con probabilità 0.1 ed ha una durata campionata da una distribuzione esponenziale con media 3 secondi. Durante questo intervallo, il tasso di arrivo è di tipo esponenziale con parametro  $\lambda_{c,bursty} = 20 \text{ req/sec}$ .
  - c. I tempi di servizio di CPU e disco sono estrapolati da quelli identificati al punto 1. Si assuma in particolare per le query di tipo A e di tipo B un tempo di servizio per CPU e disco pari a quello della tipologia di query più CPU-intensive; si assuma per le query D ed E un tempo di servizio per CPU e disco pari a quello della tipologia di query più disk-intensive; infine, si assuma per le query di tipo C un tempo di servizio di CPU e disco pari a quello della query con tempi di servizio più bilanciati tra CPU e disco.
  - d. Il tempo di risposta di ciascuna query non deve superare in media il tempo di risposta della query più lunga a sistema scarico.
  - e. L'utilizzazione di ogni centro del sistema deve essere compresa tra 60% e 70%
  - f. Il sottosistema di storage deve essere implementato con un RAID 10

Note:

1. Si consiglia di utilizzare un fattore di scala per il benchmark pari a 5, laddove il computer utilizzato per il progetto sia dotato di un quantitativo di RAM maggiore o uguale a 8 GB. Se si

dispone di una macchina con un quantitativo di memoria inferiore, contattare i docenti per concordare il fattore di scala da utilizzare.

2. Configurare postgres con i seguenti parametri:
  - a. `max_worker_processes = 0`
  - b. `max_parallel_workers = 0`
  - c. `track_io_timing = on`