

Simone Albero 552166, Matteo Cardilli 550208

Tecnologie e architetture per la gestione dei dati

21 giugno 2023

Analisi delle prestazioni e dimensionamento di un DBMS

Il progetto si propone di sviluppare un modello formale per valutare le prestazioni di un Database (DBMS) utilizzando il benchmark TPC-H. Sarà impiegato il software "Java Modelling Tools" per dimensionare i parametri di una simulazione sulla base dei risultati generati dal benchmark. Le misurazioni verranno eseguite su due computer; tuttavia, essendo i risultati ottenuti concordi in entrambi i contesti, si è scelto, al fine di migliorare la leggibilità della seguente stesura, di fare riferimento al dispositivo con le seguenti specifiche:

- Computer: MacBook Pro
- Sistema operativo: Ventura 13.4
- Chip: Apple M1 Pro
- Memoria RAM: 16GB

Nel contesto delle misurazioni, il sistema Docker sarà utilizzato come intermediario al fine di garantire l'integrità e l'uniformità dei risultati. Tuttavia, è importante considerare che le prestazioni del container potrebbero essere compromesse a causa del fatto che l'immagine utilizzata non dispone di una versione specifica per l'architettura ARM e, di conseguenza, è soggetta ad emulazione.

PASSO I:

In questa fase, si procederà all'analisi delle prestazioni della macchina reale, includendo il tempo di esecuzione delle query, il tempo di servizio e l'utilizzo dei centri. Il numero massimo di query concorrenti sarà impostato su 1, e il container Docker sarà configurato per utilizzare solo una CPU. Saranno identificati i parametri rilevanti per la simulazione, tenendo conto delle specifiche del computer misurate tramite il benchmark. A tale scopo, si è scelto di dimensionare il benchmark su scala 5, caricando i dati nel DBMS PostgreSQL e eseguendo solo le query che non comportano modifiche al contenuto del database. La decisione di non apportare modifiche al contenuto del DBMS è stata presa al fine di accelerare l'esecuzione di eventuali test futuri.

Di seguito sono riportati i dati osservati tramite l'estensione "pg stat statements" dopo l'esecuzione delle 22 query del benchmark. Si tenga presente che i risultati ottenuti rappresentano un'approssimazione della realtà, in quanto sono osservati a livello del DBMS, ovvero a un livello relativamente alto. Oltre alle query del benchmark, vengono eseguite altre query di gestione con un "total execution time" molto basso e di poca rilevanza per l'analisi, per questo motivo le abbiamo omesse.

Pg_stat_statements								
queryid	total_exec_time	D_CPU	D_Disk	CPU/IO	mean_exec_time	blk_read_time	calls	
-7225261453483776278	41546.132099999995	5057,02434516667	1867,33100483333	2,70815636439242	6924.35535	11203.986029	6	
2626751255145609859	117332.146347	17148,0234253333	2407,33429916667	7,12324143400828	19555.357724499998	14444.005794999997	6	
6918813626601998185	55675.016234	6887,77593033333	2391,393442	2,88023535122387	9279.169372333334	14348.360652	6	
-1086153583272167779	141890.81535699998	21439,6079596667	2208,8612665	9,7061813183217	23648.469226166668	13253.167598999999	6	
-8623419497723448577	145.542293	14,9973115	9,25973733333334	1,61962601746946	24.257048833333336	55.55842400000001	6	
2013159534999621399	39957.580563	4972,23655566667	1687,36020483333	2,94675466531926	6659.5967605000005	10124.161229000001	6	
-782868727798575624	21738.097425	2091,004557	1532,0116805	1,36487507478896	3623.0162375	9192.070083000002	6	
1415028499643213595	89118.30325000001	10765,6184573333	4087,43208433333	2,63383421062743	14853.050541666667	24524.592506	6	
7938628011141446549	17512.461801999998	2778,53085216667	140,2127815	19,8165304363973	2918.743633666667	841.276689	6	
6811819842153739267	13395.220046	836,629388333333	1395,907286	0,599344524327766	2232.5366743333334	8375.443716	6	
3170858136560744772	52832.717482	7069,14174	1736,31117366667	4,07135647527492	8805.452913666664	10417.867042	6	
450838253780030881	32445.150264	1349,55266783333	4057,97237616667	0,332568224406735	5407.525044	24347.834257	6	
-38097028399993979407	30046.956722	2840,85910533333	2166,967015	1,31098400929436	5007.826120333333	13001.802090000001	6	
-241439892892349202	27051.266516000003	485,833802166666	4022,71061716667	0,12077274464969	4508.544419333333	24136.263703	6	
7299199919828511469	51857.293022	4612,67311466667	4030,20905566667	1,14452452737682	8642.882170333332	24181.254334	6	
-597569764262573984	5976.902797000001	577,092318	419,058148166667	1,37711752062265	996.1504661666668	2514.348889	6	
1819064965949763676	22189.78326	1276,291079	2422,006131	0,526956171854546	3698.2972099999997	14532.036786	6	
427808448988726885	64815.590363	3984,52293666667	6818,07545716667	0,584405813883803	10802.598393833334	40908.452743	6	
-7828248469682243801	7228.443004	258,962627166667	945,7778735	0,27380914104951	1204.7405006666665	5674.667241	6	
-7363898342359107977	36601.077017	3793,7514335	2306,42806933333	1,64486006910095	6100.179502833333	13838.568415999998	6	
-699892568854201504	34492.478557999995	5283,31241933333	465,434007	11,3513674116497	5748.746426333333	2792.6040420000004	6	
-2004558998418799363	25821.136258000002	1893,49274533333	2410,02996433333	0,785671868547541	4303.522709666668	14460.179786	6	

Si noti che il service demand del disco relativo a una specifica query è ottenibile mediante la divisione del tempo totale di lettura per il numero totale di chiamate. Il service demand della CPU relativo a una specifica query, invece, può essere ottenuto sottraendo il service demand del disco dal tempo medio di esecuzione. È importante sottolineare che tale calcolo è possibile in quanto il numero massimo di query concorrenti è impostato a 1.

Per dimensionare i parametri della simulazione, si è scelto di utilizzare tre rappresentanti tra le 22 differenti query. Utilizzando questi tre rappresentanti, possiamo valutare le prestazioni del sistema in scenari diversi e ottenere un'idea del comportamento del sistema sotto diverse condizioni di utilizzo di CPU e disco. In particolare, definiamo i seguenti rappresentanti:

- Query CPU intensive: Una query caratterizzata da un notevole utilizzo di CPU e ridotto utilizzo di disco.
- Query Disk intensive: Una query caratterizzata da un notevole utilizzo di disco e ridotto utilizzo di CPU.
- Query Bilanciata: Una query caratterizzata da un utilizzo equilibrato tra CPU e disco.

Di seguito sono riportati i parametri indicativi di queste tre tipologie di query:

CPU intensive:	D_CPU	D_Disk
19,8165304363973	2,91874363366667	0,1402127815
Query N: 16.sql		
Disk intensive:	D_CPU	D_Disk
0,12077274464969	0,485833802166666	4,02271061716667
Query N: 11.sql		
Balanced	D_CPU	D_Disk
1,14452452737682	4,61267311466667	4,03020905566667

PASSO II:

In questa fase verranno confrontate le prestazioni osservate sulla macchina reale (Prestazioni reali) con quelle predette da un modello simulativo (Prestazioni simulate). Per realizzare ciò, si prevede di sviluppare formalmente un modello simulativo semplificato che sarà fondato su due centri di elaborazione (CPU) e un disco (Disco). Inoltre, sulla base dei rappresentanti precedentemente individuati, sarà necessario definire due tipi di classi, ovvero "CPU intensive" (Intensivo per la CPU) e "Disk intensive" (Intensivo per il disco), opportunamente dimensionate.

Sulla base di quanto esposto, nel rispetto del vincolo sul massimo numero di query concorrenti, è stato effettuato un processo di simulazione del sistema. Durante tale processo, il sistema è stato sottoposto in modo alternato a un singolo job appartenente a ciascuna delle due classi precedentemente definite. Di seguito sono riportati i risultati ottenuti dalla simulazione:

CPU Intensive	CPU	Disk	Disk Intensive	CPU	Disk
Job	0.9547	0.0482	Job	0.1082	0.8905
Utilizzazione	0.9547	0.0482	Utilizzazione	0.1082	0.8905
Throughput	0.3433	0.3409	Throughput	0.2243	0.2221

Si osservi che ogni centro è occupato o inattivo, poiché il sistema è sottoposto a un solo job alla volta. Pertanto, nel caso in cui il job sia di tipo "CPU intensive", si osserva un'utilizzazione della CPU vicina al 100% e un'utilizzazione del Disco vicina allo 0%. La stessa situazione si verifica anche nel caso "Disk intensive", ma in modo opposto. Il dato più interessante risulta essere il throughput, il quale, poiché i due centri sono soggetti allo stesso numero di visite, risulta essere lo stesso per entrambi.

Al fine di confrontare i risultati ottenuti con il sistema reale, è stata effettuata una serie di cinque esecuzioni per ciascuno dei due diversi tipi di query, al fine di determinare il throughput effettivo. I risultati ottenuti sono riportati di seguito:

	CPU Intensive		Disk Intensive
Throughput_test_total:	14,36	Throughput_test_total:	7,56
Throughput:	0,348189415041783	Throughput:	0,661375661375661

PASSO III:

In questa fase, verrà condotta un'analisi simile a quella della fase precedente, con l'eccezione che sarà impostato un limite massimo di 2 query concorrenti dello stesso tipo. Durante il processo di simulazione, il sistema è stato sottoposto alternativamente a due job appartenenti a ciascuna delle due classi precedentemente definite. Di seguito sono riportati i risultati ottenuti dalla simulazione:

CPU Intensive	CPU	Disk
Job	1.9479	0.0527
Utilizzazione	0.9979	0.0503
Throughput	0.3580	0.3585

Disk Intensive	CPU	Disk
Job	0.1302	1.8706
Utilizzazione	0.1175	0.9870
Throughput	0.2434	0.2451

Si osservi che, anche in questo caso, il numero di job si avvicina a 2 nel centro con l'utilizzo più elevato. Tuttavia, per quanto riguarda il throughput, non si notano significativi aumenti nonostante la presenza contemporanea di due job. Questo fenomeno può essere giustificato dal fatto che, in entrambi i casi, il centro più gravato dalla query tende a raggiungere un utilizzo prossimo al 100%, agendo come collo di bottiglia per il sistema.

Al fine di riscontrare quanto detto nel caso reale, sono state lanciate due stream concorrenti di cinque esecuzioni per ciascuno dei due diversi tipi di query, al fine di determinare il throughput effettivo. I risultati ottenuti sono riportati di seguito:

	CPU Intensive
Throughput_test_total:	28,74
Throughput:	0,34794711203897

	Disk Intensive
Throughput_test_total:	9,87
Throughput:	1,01317122593718

Si noti che, a differenza del caso "CPU intensive" in cui i dati reali corrispondono a quelli ottenuti durante la simulazione, nel caso "Disk intensive" il throughput aumenta notevolmente. La giustificazione di questo particolare fenomeno è assimilabile ai meccanismi di parallelizzazione tramite molteplici canali implementati da alcuni dischi moderni. Tali meccanismi non possono essere controllati e/o limitati, portando ad un miglioramento delle prestazioni all'aumentare del numero di query concorrenti.

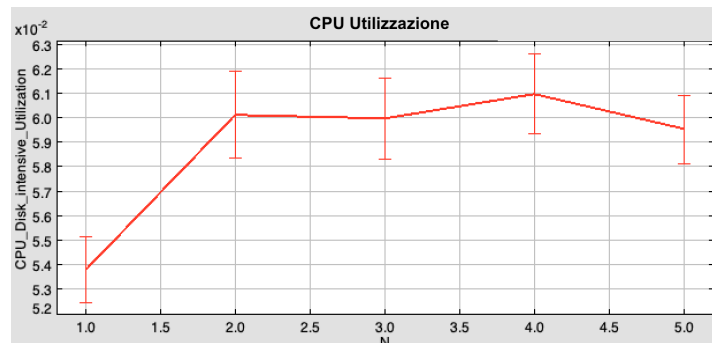
PASSO IV:

In questa fase, sarà condotta un'analisi su una variante del modello precedentemente osservato, che prevede l'utilizzo di 2 CPU. In particolare, si condurrà un'analisi di tipo "what if" con un numero massimo di query concorrenti variabile da 1 a 5. Di seguito sono riportati i risultati ottenuti dalla simulazione:

CPU Intensive	N = 1	N = 2	N = 3	N = 4	N = 5
Job CPU	0.9519	1.8981	2.8865	3.8824	4.8853
Utilizzazione CPU	0.4760	0.9491	0.9943	0.9993	1.000
Throughput CPU	0.3458	0.6801	0.7238	0.7374	0.7173

Disk Intensive	N = 1	N = 2	N = 3	N = 4	N = 5
Job Disco	0.8956	1.8810	2.8805	3.8799	4.8785
Utilizzazione Disco	0.8956	0.9942	0.9996	1.000	1.000
Throughput Disco	0.2232	0.2488	0.2494	0.2498	0.2483

Si osserva che, nel caso di query "CPU intensive", l'aggiunta di una CPU aggiuntiva consente l'esecuzione simultanea dei due job, portando a un raddoppio del throughput iniziale. Tuttavia, nei casi in cui il numero di job supera 2, non si riscontrano aumenti significativi nel throughput a causa dell'utilizzo delle CPU che si avvicina al 100%. Al contrario, nel caso di query "Disk intensive", l'aggiunta di una CPU aggiuntiva non ha portato miglioramenti al sistema, poiché il disco presenta un utilizzo prossimo al 100% fin dall'inizio, diventando il collo di bottiglia del sistema. In questo caso, infatti, come mostrato nella figura seguente, l'utilizzo della CPU non solo rimane basso, ma diminuisce all'aumentare del numero di job a causa della saturazione del disco.



Al fine di verificare quanto affermato nel contesto reale, sono state eseguite le query del benchmark con un numero di stream pari a 1, 2 e 5, al fine di determinare i throughput effettivi. Di seguito sono riportati i risultati ottenuti:

CPU Intensive	N = 1	N = 2	N = 5
Throughput_test_total:	13,95	14,26	35,72
Throughput:	0,3584229390681	0,701262272089762	0,699888017917133

Disk Intensive	N = 1	N = 2	N = 5
Throughput_test_total:	8,63	9,16	11,91
Throughput:	0,579374275782155	1,09170305676856	2,09907640638119

Dopo aver esaminato il comportamento del sistema sia nel caso reale che nel caso simulato, tenendo conto delle considerazioni precedenti, si è proceduto al dimensionamento del sistema con l'obiettivo di garantire un tempo di risposta medio inferiore a 30 secondi per un numero massimo di query concorrenti non superiore a 20, assicurando che l'utilizzo delle risorse sia compreso nell'intervallo del 60%-70%. Inoltre, si è stabilito che il sottosistema di storage debba essere implementato con un RAID 10.

Nella prima fase di questo processo, si è lavorato per garantire un adeguato livello di utilizzazione della CPU, concentrandosi esclusivamente sui dati relativi alle query "CPU intensive". Dai dati precedentemente analizzati, è emerso che un job satura quasi completamente una CPU, avvicinandosi al 100% di utilizzo. Pertanto, al fine di garantire un utilizzo compreso tra il 60% e il 70%, è necessario inserire un numero di CPU leggermente inferiore al doppio del numero dei job da gestire.

Nella fase successiva, si è proceduto in modo analogo ma opposto, concentrandosi esclusivamente sui dati relativi alle query "Disk intensive", al fine di determinare correttamente il numero di dischi necessari. Dai dati precedentemente analizzati, si è osservato che con un job, l'utilizzo del disco tende a essere intorno al 90%. Pertanto, anche in questo caso, il numero totale di dischi sarà approssimativamente il doppio del numero dei job.

Dopo aver verificato queste osservazioni, si è proceduto a tentativi tramite simulazione al fine di raggiungere un dimensionamento adeguato, ottenendo i seguenti parametri:

Dimensionamento	CPU	Disco
Numero di Job	20	20
Numero di serventi	27	28
Utilizzazione	0.6347	0.6392

La fase conclusiva di questo processo ha comportato la sostituzione, nella simulazione, del centro "Disco multiservente" con un sistema RAID 10. Poiché, secondo i vincoli di progetto, il carico è esclusivamente di lettura, è possibile dimensionare l'unità RAID considerando un livello di parallelizzazione su tutti i dischi che la compongono. Sulla base di queste considerazioni, si prevede di utilizzare un numero di dischi nell'unità pari al numero di dischi precedentemente stimato per il centro multiservente. Ognuno di questi sarà caratterizzato dai service demand effettivamente osservati nella realtà, divisi per il numero totale di dischi.

Tuttavia, si osserva che, ridimensionando l'unità in questo modo, l'utilizzo ottenuto durante la fase di simulazione tende a raggiungere valori troppo elevati, rendendo necessario aumentare il numero di dischi per garantire un utilizzo accettabile.

PASSO V:

In questa fase, sarà condotta un'analisi su vari tipi di query, supponendo che il sistema sia effettivamente destinato alla produzione. Questa fase del progetto si differenzia dalle altre in quanto lo studio non viene più condotto su un sistema a ciclo chiuso, ma su un sistema a ciclo aperto. Nel contesto del sistema a ciclo chiuso, si fa riferimento a una rete di code in cui il sistema non comunica con l'esterno e non presenta tassi di arrivo esponenziali. In questo tipo di sistema, i flussi di lavoro all'interno della rete di code sono completamente controllati e non dipendono da input o output esterni. Un sistema a ciclo aperto, d'altra parte, è caratterizzato da una rete di code in cui il sistema comunica con l'esterno attraverso ingressi e uscite ed è caratterizzato da tassi di arrivo esponenziali.

Si noti che l'obiettivo di questa fase non è quello di studiare singolarmente le varie query, ma piuttosto di analizzare un mix di 5 query (A, B, C, D, E) e valutarne gli impatti sui centri di calcolo. Nel dettaglio, le query A, B, D, E presentano tassi di arrivo esponenziali con un valore comune: $\lambda_a = \lambda_b = \lambda_d = \lambda_e = 2$ richieste al secondo.

Per quanto riguarda la query C invece, questa è di tipo bursty, il che significa che il suo carico di lavoro varia in modo differenziato su intervalli specifici, causando un improvviso aumento del traffico o delle richieste in un breve lasso di tempo. Il primo intervallo si verifica con una probabilità del 0,9 ed ha una durata di 30 secondi. Durante questo intervallo, il tasso di arrivo delle richieste segue una distribuzione esponenziale con un parametro $\lambda_c = 2$ richieste al secondo. L'altro intervallo della query C si verifica con una probabilità del 0,1 e ha una durata di 3 secondi. Durante questo intervallo, il tasso di arrivo delle richieste è esponenziale e pari a $\lambda_{c,bursty} = 20$ richieste al secondo.

I tempi di servizio delle suddette query invece, sono stati configurati sulla base dei tempi di servizio della CPU e del Disco estrapolati al passo 1. Per le query di tipo A e B, si assume un tempo di servizio per la CPU e il disco pari a quello della tipologia di query più intensiva in termini di utilizzo della CPU. Per le query D ed E, si assume un tempo di servizio per la CPU e il disco pari a quello della tipologia di query più intensiva in termini di utilizzo del disco. Infine, per la query di tipo C, si assume un tempo di servizio della CPU e del disco pari a quello della query con tempi di servizio più bilanciati tra la CPU e il disco, ovvero con un rapporto tra i due valori più prossimo a uno. I requisiti di prestazione stabiliscono che il tempo di risposta medio di ogni query non deve superare il tempo di risposta della query con il tempo di esecuzione più lungo nel sistema senza carico. Inoltre, l'utilizzazione di ogni centro di servizio deve essere mantenuta tra il 60% e il 70%. Infine, è prevista l'implementazione di un sottosistema di storage utilizzando la configurazione RAID 10.

Appurate le specifiche sopra menzionate, si è proceduto con la modellazione del sistema considerando cinque diverse classi, una per ciascuna query specifica. Ogni classe è stata modellata accuratamente in conformità ai requisiti precedentemente definiti. Considerando il tipo di analisi richiesta, si è focalizzati principalmente sull'osservazione delle risposte del sistema in termini di utilizzazione e tempi di risposta. A tal fine, sono stati definiti gli indici di performance rilevanti per tale contesto e si è proceduto con la simulazione del sistema.

Al fine di soddisfare i vincoli di progetto, i server dei due centri (CPU e Disco) sono stati ridimensionati in modo adeguato. Si è iniziato con un numero elevato di server e si sono effettuati decrementi opportuni in base a quanto osservato durante la simulazione. Di seguito sono riportati i parametri di dimensionamento del sistema ottenuti come risultato di questa analisi:

Dimensionamento	CPU	Disco
Numero di server	35	38
Utilizzazione	0.6858	0.6702

	A	B	C	D	E	Max
Response Time	3.1880	3.1900	9.0867	4.5952	4.6511	19.5553