

Analisi comparata di MySQL, PostgreSQL e SQLite

30 luglio 2007

Master in Tecnologia del Software Libero e Open Source

Corso di Sistemi Informativi

Anno Accademico 2005/2006

Autori: Federico Ceratto <federico.ceratto@gmail.com>, Roberto A. Foglietta
<roberto.foglietta@gmail.com>, Pietro Marrone <pietro.marrone@gmail.com>

Sommario

Un' analisi comparativa delle funzioni e delle performance di tre dei più diffusi RDBMS Open Source. Viene verificata la conformità con gli standard SQL e la presenza di features rilevanti quali ridondanza e scalabilità. Sono stati svolti dei benchmark su differenti sistemi, i cui risultati vengono comparati rivelando punti di forza e debolezza dei diversi RDBMS. In base agli esiti, viene indicato lo scenario di utilizzo più adatto per ognuno dei tre sistemi.

Indice

1	Introduzione	3
2	Comparazione delle caratteristiche	4
3	Analisi delle prestazioni	7
3.1	Lo strumento di misura	7
3.1.1	Il processo di misura	7
3.1.2	La scelta dello strumento di misura	8
3.1.3	Sistemi utilizzati nella conduzione dei benchmark	9
3.2	Comparazione dei risultati	10
4	Conclusioni	13
4.1	MySQL	13
4.2	PostgreSQL	13
4.3	SQLite	14

1 Introduzione

Negli ultimi anni il software Open Source ha iniziato ad aggredire i segmenti di fascia bassa dei sistemi di gestione di basi di dati (RDBMS), offrendo alcuni prodotti di qualità paragonabile alle più diffuse soluzioni closed. I sistemi più evoluti si preparano a lambire anche il segmento di fascia alta.

A favore delle soluzioni aperte si conta l'accessibilità alla documentazione e ai codici sorgenti, che permette di analizzare la qualità del software e comprenderne il funzionamento in modo non altrimenti possibile. Un'ulteriore risorsa è rappresentata dalla comunità che ruota attorno a ogni prodotto, all'interno della quale è possibile confrontarsi con amministratori e sviluppatori anche molto preparati.

D'altro canto la possibilità di effettuare scelte consapevoli porta con sé del lavoro aggiuntivo riguardante l'attività di ricerca di informazioni e di selezione del candidato più adatto, e necessita quindi di competenze specifiche. Inoltre è opportuno considerare in alcuni casi l'assenza di un vendor che si assuma le responsabilità sul corretto funzionamento del software e che si occupi di fornire assistenza tecnica.

Lo scopo di questa analisi è di fornire una visione panoramica sui prodotti più conosciuti e maturi. Al fine di facilitare l'approfondimento sono stati inseriti frequenti richiami alla bibliografia, basata il più possibile su fonti di documentazione ufficiali.

Gli RDBMS che sono stati presi in considerazione manifestano caratteristiche differenti e le loro funzioni sono fortemente eterogenee. E' necessario valutare diversi aspetti fra i quali le funzionalità che hanno in comune, le relative prestazioni, effettuare un'analisi sulla compatibilità con gli standard SQL e infine identificare e proporre gli scenari di utilizzo più adatti per ogni RDBMS.

Si è quindi preferito basare l'analisi su più prospettive, fornendo in primis una comparazione delle differenze e delle peculiarità, per poi proseguire con un'analisi delle prestazioni e infine identificare e proporre alcuni scenari di utilizzo più adatti per ogni RDBMS.

Su quest'ultimo punto occorre fare una precisazione cautelativa: la scelta dell'RDBMS più adatto non è un compito semplice. La nascita di discussioni su quale sistema sia preferibile è un evento ricorrente all'interno delle mailing list dedicate agli amministratori di database e non sembra essere vicino il momento in cui sarà possibile dare una risposta univoca.

2 Comparazione delle caratteristiche

La tabella comparativa rende evidenti le profonde differenze e le peculiarità dei sistemi:

Caratteristiche \ RDBMS	MySQL	PostgreSQL	SQLite
Dimensione massima del DB	64TB - 386ExaB	Illimitata [2]	minore di 100GB
Access Control	Si, molto granulare [17]	Si [18]	No
Data Types	11 [3]	32 [4]	5 [5]
Custom Data Types	No	Si [6]	No
Supporto Large Objects	Si	Si	No
Supporto dati spaziali/geometrici	Si [8]	Si [7]	No
Indexes per tabella	64 contenenti massimo 16 colonne	N/A contenenti massimo 15 colonne	Teoricamente illimitati
Supporto Unicode	Si	Si	Si
Implementazione dei Triggers	SQL	pgSQL, Tcl, Perl, Python	SQL, limitata
Stored Procedures	SQL	pgSQL, Tcl, Perl, Python	No
Inheritance	No	Si	No
Sequences	No, ma esiste AUTO_INCREMENT	Si	No, ma esiste AUTO_INCREMENT
AUTO_INCREMENT	Si	No, ma esistono le Sequences	Si
Cursors	Si	Si	No
Views	Si	Si	Si
Controllo di tipo sui campi	Si	Si	No
Supporto ALTER TABLE	molto completo [14]	parziale [15]	molto limitato [16]
ALTER TABLE DROP FIELD	Si	Si	No: è necessario creare una nuova tabella e cancellare la precedente
AUTOVACUUM	Si	No	Si
Tabelle con transazioni	Si	Si	Si
Tabelle senza transazioni	Si [11]	No	No
Complessità dell'installazione	Media	Media/Alta	Bassa
Accesso parallelo	Si	Si	limitato: possibile attraverso processi paralleli
Architettura Client-Server	Si	Si	No
Clustering	Si [1]	No	No
Replicazione Master-Slave	Si	Si	No

Hot Backup	Si	Si	Parziale: necessita write lock
Funzioni EXPLAIN / ANALYZE	EXPLAIN [12]	ANALYZE [13]	No
Locking granularity	Tabella/Row	Tabella/Row con MVCC	Database
Multithreading / parallelismo su CPU	Si	Si	Parziale
ODBC e JDBC	Si [20]	Si [21]	No
Piattaforme: Linux, BSD, Windows, MacOSX	Si	Si	Si
Licenza	Molteplici, tra cui GPL	Simile alla BSD	Public Domain
Modello di sviluppo	Centralizzato	Community	Un solo autore
Supporto e assistenza vendor	Si	No	Si, ma limitata
Numero di voci su Google ¹	159 Mln	31.4 Mln	8 Mln

Si delineano alcuni insiemi di caratteristiche principali che caratterizzano diversamente ogni RDBMS:

Le dimensioni di scala di utilizzo

Parametri quali la dimensione massima del Database e il supporto di funzioni di clustering sono indicativi per un utilizzo a livello Enterprise. PostgreSQL è indubbiamente il sistema più evoluto, seguito da MySQL. SQLite si profila immediatamente come inadatto per questi ambiti.

Le funzionalità SQL

Negli scenari dove si richiede a un RDBMS di gestire della Business Logic di complessità elevata, è importante disporre di un'insieme ampio di istruzioni, siano esse native SQL o anche estensioni proprietarie. La valutazione coinvolge il numero di tipi di dati, ivi compreso il supporto di dati spaziali e geometrici, il supporto di Triggers, Stored Procedures e simili. I risultati confermano quelli del paragrafo precedente.

La sicurezza

L'unica voce che mostri differenze significative relativamente alla sicurezza è l'Access Control. E' MySQL in questa fattispecie a vantare le funzioni più evolute, con una granularità molto elevata sui permessi di accesso a tabelle, campi e record.

¹Il numero di voci rilevate dal motore di ricerca fornisce una misura approssimativa della dimensione della base di utenza

La complessità di installazione e manutenzione

La regola empirica, secondo cui i sistemi più sofisticati sono anche i più complessi come gestione, sembra verificarsi. SQLite, nella sua semplicità, permette un'installazione molto veloce ed è quasi privo di esigenze di manutenzione, contrariamente a PostgreSQL e MySQL.

Il supporto e la comunità

Oltre alle caratteristiche prettamente tecniche, non è possibile prescindere da aspetti quali: la possibilità di ottenere supporto dal parte del fornitore, la dimensione della comunità degli utenti (misurata empiricamente attraverso Google), la licenza e il modello di sviluppo. MySQL risulta decisamente il sistema più diffuso.

3 Analisi delle prestazioni

Le prestazioni offerte da un RDBMS sono un elemento imprescindibile nella valutazione della qualità generale del prodotto. Esistono applicazioni di scala molto ridotta nelle quali la frequenza delle operazioni che coinvolgono il database è tale da non richiedere una porzione rilevante delle risorse computazionali di cui si dispone.

Tuttavia, al crescere del carico di lavoro richiesto all'RDBMS le risorse necessarie aumentano di conseguenza, non sempre in modo proporzionale.

E' è pratica comune dedicare uno o più server in maniera esclusiva alla gestione dei Database, arrivando anche a diverse decine in casi più rari. All'interno di strutture di grandi dimensioni diviene evidente come le prestazioni degli RDBMS influiscano in maniera diretta sui costi di esercizio e sulla qualità del servizio offerto.

3.1 Lo strumento di misura

Il tipo di strumento utilizzato per svolgere un'analisi di prestazioni, in particolare per gli RDBMS, è intrinsecamente un elemento critico relativamente all'attendibilità dei risultati che si vogliono ottenere.

Date le circostanze, la possibilità di reperire uno strumento complesso che fornisca all'utente un unico parametro di qualità immediatamente fruibile è una speranza assolutamente ottimistica. E' invece necessario rivolgersi ad una suite di strumenti deputati alla misurazione di una molteplicità di parametri. La suite stessa produce una serie di valori riconducibili ad aspetti macroscopici. E' compito dell'utente arrivare ad un giudizio complessivo attribuendo il peso opportuno ai valori di cui dispone.

3.1.1 Il processo di misura

Tipicamente un benchmark viene svolto in fasi successive:

1. **Inserimento dei parametri generali di test**

Viene richiesto all'utente di stabilire un numero limitato di parametri di importanza macroscopica: si pensi ad esempio il quantitativo di memoria RAM utilizzabile da parte del RDBMS sottoposto al test.

2. **Verifica delle condizioni operative**

Prima di procedere all'analisi delle prestazioni vera e propria viene svolta una verifica sullo stato del sistema e dell'RDBMS. Lo scopo è quello di evitare l'esecuzione di test che si rivelerebbero privi di significato o, peggio, ingannevoli. Perciò lo strumento di test ha il compito di verificare innanzitutto la possibilità di contattare il demone in analisi, la presenza di un'utenza abilitata al benchmark la quale sia fornita degli opportuni diritti di accesso, l'effettiva disponibilità della memoria RAM necessaria per il test, etc.

3. **Esecuzione sequenziale dei test**

Qualora la fase di verifica avesse avuto esito positivo il tool procede sottoponendo il sistema ad una serie di test. In questa fase il sistema dovrà essere utilizzato esclusivamente dello strumento di benchmark e qualsiasi altra attività svolta dall'utente dovrà essere

considerata un'interferenza che potrebbe invalidare i risultati. Gli stessi test vengono normalmente somministrati in maniera sequenziale piuttosto che parallela in modo da evitare che si creino delle influenze reciproche che produrrebbero risultati piuttosto difficili da analizzare e confrontare. Tuttavia, data la complessità dei comportamenti manifestati da un RDBMS, non è del tutto ragionevole aspettarsi che i risultati di test eseguiti singolarmente siano rappresentativi delle performance che si avrebbero nell'utilizzo reale. Pertanto vengono eseguiti anche test di contenuto eterogeneo che sottopongono il sistema in analisi a schemi di utilizzo progettati per simulare una situazione tipica.²

4. Analisi dei risultati e generazione di un report

L'ultima fase consiste nell'eseguire un'analisi automatica dei risultati forniti dai test precedenti e produrre un numero ridotto di variabili significative. Non deve sorprendere che la fase di test produca molte decine di misure. Naturalmente i valori ottenuti vengono forniti nella loro interezza, tuttavia è preferibile generare un sommario che permetta all'utente un notevole risparmio di tempo nella valutazione dei risultati. Per poter estrarre degli indici generali di prestazione dall'insieme delle misure ottenute è necessario fare delle medie pesate e altre operazioni che introducono inevitabilmente delle componenti soggettive nella valutazione. Tipicamente sono gli autori del tool stesso a stabilire i valori con cui verranno pesate le misure, in base all'importanza relativa che attribuiscono a ogni singolo test. Una scelta ragionevole di tali valori porta ad avere un valido sistema di benchmark dove per valido si intende che i risultati forniscano un'immagine ragionevolmente vicina alle prestazioni ottenute nel caso reale quando questo non sia un caso particolare o affetto da patologie nella scelta dei dati o delle istruzioni SQL.

La relativa complessità dei tool di benchmark rende evidente che in un'analisi delle prestazioni il risultato non dipende unicamente dal sistema preso in esame ma anche dallo strumento di misura e in particolare proprio la fase di analisi dei risultati è la causa principale della "soggettività" che si riscontra. D'altronde non sarebbe ragionevole pretendere livelli di precisione particolarmente elevati da metodi di test che sono stati sviluppati a prescindere dalle condizioni di utilizzo reale degli RDBMS. La misurazione delle prestazioni realmente fornite da sistemi in produzione caso non si tratta di benchmark di tipo generico generici ma uno degli strumenti utilizzati dall'amministratore durante il lavoro di ottimizzazione delle basi dati e del codice SQL che viene eseguito.

3.1.2 La scelta dello strumento di misura

In base alle precedenti considerazioni, risulta evidente che la scelta dello strumento da utilizzare debba essere compiuta con particolare attenzione in tal senso sono state valutate le seguenti alternative:

- Sviluppare uno strumento "fatto in casa" consistente in relativamente poche righe di codice il quale sia in grado di svolgere una serie ridotta di misure di velocità circa le chiamate SQL più comuni;
- Utilizzare strumenti Open Source già esistenti:

²Si vedano ad esempio i test Wisconsin e ATIS svolti da sql-bench, basati su dati verosimili

- La suite sql-bench presente all'interno del pacchetto di sorgenti di MySQL versione 5.0;
- MySQL SuperSmack [24] una suite indipendente orientata all'uso del RDBMS come base dati per siti web;
- Utilizzare strumenti non Open Source.

La scelta si è orientata verso sql-bench in quanto soddisfacente dei requisiti fondamentali:

- Prodotto maturo e quindi ragionevolmente privo di banali difetti strutturali e/o implementativi;
- Utilizzato e mantenuto da un'ampia base di utenti: si tratta dello strumento di benchmarking ufficiale di MySQL;
- Compatibilità: esistono vari tool in grado di interfacciarsi ed analizzare solamente un RDBMS. La loro funzione può essere utile durante l'ottimizzazione di un sistema, ma nel caso in questione sono stati necessariamente scartati. Sql-bench è in grado, nativamente, di utilizzare vari RDBMS tra cui PostgreSQL, MySQL anche se non è previsto il supporto per SQLite;
- Open Source: grazie alla possibilità di apportare modifiche al codice sorgente, il supporto per SQLite è stato aggiunto ad hoc ³;
- Gratuito.

3.1.3 Sistemi utilizzati nella conduzione dei benchmark

La suite di benchmark è stata utilizzata su sistemi differenti sia dal come hardware, sia come sistema operativo e anche nella sottoversione degli RDBMS presi in considerazione. La tabella seguente ne descrive le caratteristiche principali:

	Sist. operativo	Versione	Kernel	Processore
1	Debian Linux	Unstable	2.6.18	AMD Athlon XP 3200+ 2.0GHz
2	Ubuntu Linux	6.06 LTS	2.6.15	INTEL Pentium 4 2.8 GHz
3	Windows XP Pro.	5.1.2600 SP2	-	AMD Athlon 64bit 3500+ 2.2Ghz

	RAM	File System	MySQL	PostgreSQL	SQLite
1	1GB	ReiserFS	5.0.32	8.1.6	3.3.8
2	512Mb	Ext3	5.0.22	8.1.4	3.2.8
3	1GB	NTFS	5.0.19	-	3.3.9

³Una copia delle modifiche effettuate al tool verrà inviata agli sviluppatori originali

3.2 Comparazione dei risultati

Il grafico che segue mostra una comparazione fra i tempi relativi a una sequenza di istruzioni di tipo CREATE TABLE da cui si possono estrapolare alcune considerazioni:

1. MySQL dimostra di essere l'RDBMS più veloce di tutti;
2. le prestazioni di MySQL sono quelle meno influenzate dal sistema su cui gira;
3. nessuna delle macchine presenti nel testbed mostra di essere sistematicamente la più veloce;
4. fra la migliore e la peggiore prestazione si ha un fattore 6x nella velocità.

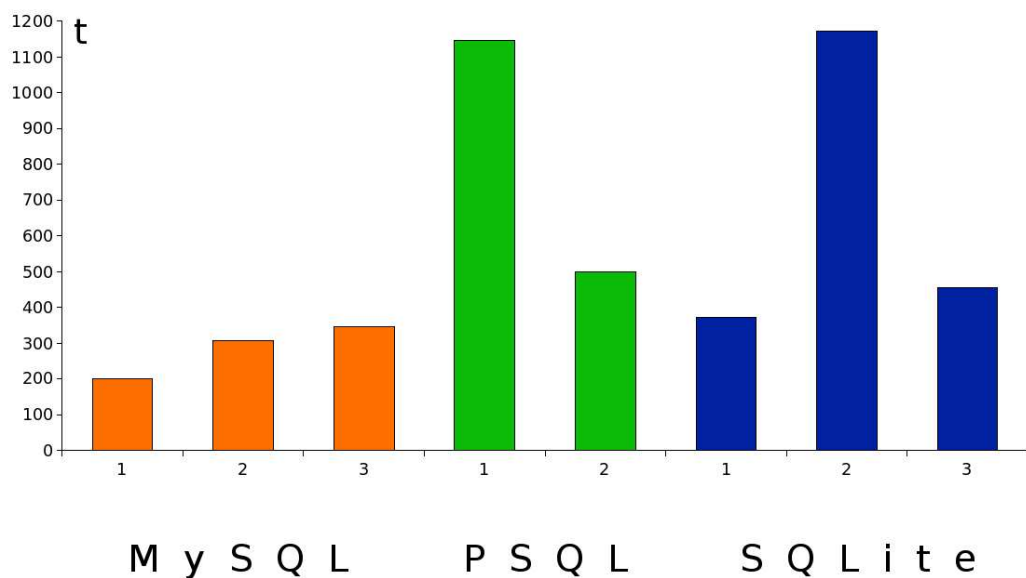


Figura 1:

E' molto interessante notare come le prestazioni di MySQL non oscillino molto al variare del sistema. Questo suggerisce che MySQL rispetto agli altri RDBMS ha molto meno bisogno di essere ottimizzato rispetto al sistema ospitante, quindi dal punto di vista delle prestazioni è un RDBMS più facile da installare e configurare.

Riguardo alla terza considerazione si possono anche invertire i termini del confronto: non più paragonando fra loro gli RDBMS ma utilizzando unicamente MySQL per fare il benchmark di diversi sistemi HW/SO. Da questa nuova prospettiva diviene evidente come le prestazioni di MySQL rispettino l'andamento della potenza di calcolo delle CPU (Linux@3.2 vs Linux@2.8) a parità di sistema operativo, mentre fra Microsoft Windows e Linux risulta essere più performante quest'ultimo (Linux@3.2 vs Windows@3.5).

Premettendo che l'ottimizzazione di RDBMS rispetto al sistema ospitante non è un compito banale e quindi introdurrebbe un elemento fortemente arbitrario⁴ si può invece considerare la

⁴L'elemento soggettivo è nell'incertezza insita in questa domanda: "sono davvero riuscito a ottimizzare l'integrazione RDBMS e sistema ospite al massimo?" mentre l'ottimizzazione minima è semplice da ottenere perché è quella si ha con l'installazione out-of-the-box.

proporzione fra i diversi minimi, rispetto al RDBMS, quindi il rapporto 6x passa a 2.5x cioè in definitiva il rapporto di prestazioni fra diversi RDBMS appropriatamente configurati rispetto al sistema ospitante⁵ dovrebbe non essere maggiore di 2.5x.

Ovviamente la CREATE TABLE non è stata scelta a caso altrimenti queste considerazioni, in particolare quella al punto 1, non potrebbero essere sostenute infatti prendendo in considerazione una funzione piuttosto il giudizio precedente potrebbe essere tranquillamente ribaltato.

Il grafico che segue mostra la comparazione fra i vari RDBMS questa volta su diverse funzioni SQL:

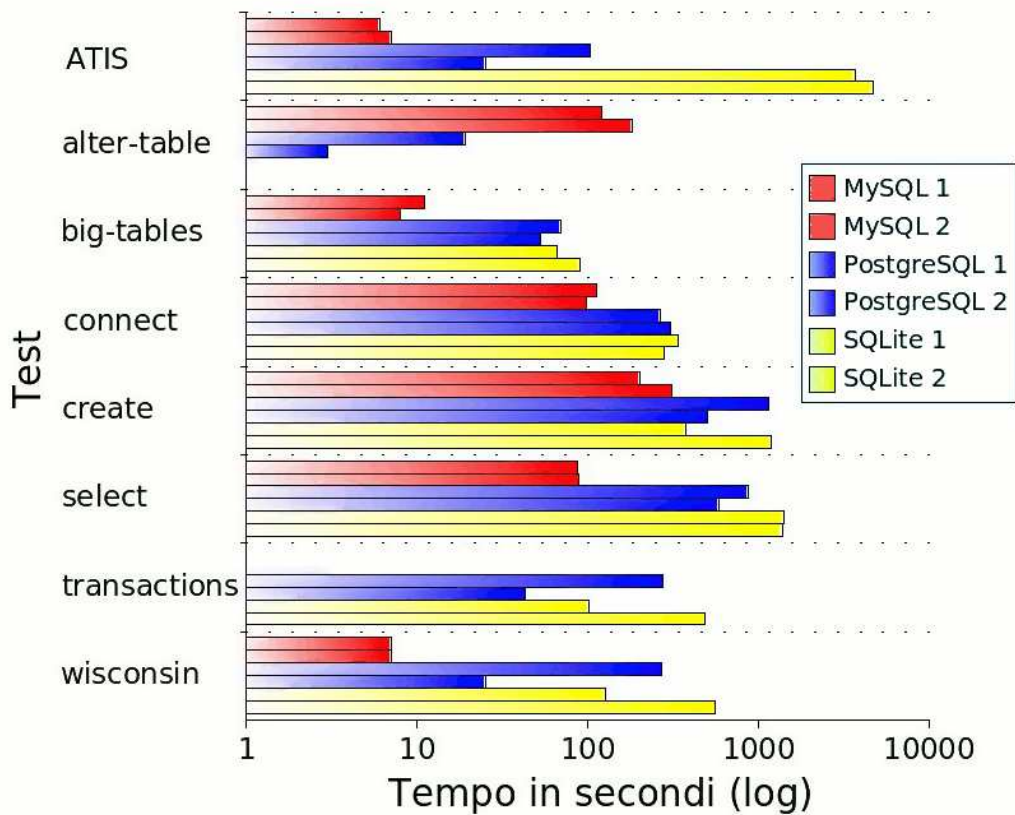


Figura 2:

Dal grafico sopra si nota come l'istruzione ALTER-TABLE manifesti prestazioni meno buone per MySQL rispetto agli RDBMS presi in esame. A parte quest'eccezione tutte le altre misure ne confermano le prestazioni.

⁵Si può infatti ottimizzare un RDBMS sotto diversi aspetti quali ad esempio: rispetto all'integrazione con il sistema ospitante che include eventualmente la scelta del sistema operativo più adatto e della configurazione hardware più adatta, rispetto alla base date che include eventualmente la dimensione delle tabelle, data types e poi rispetto al codice SQL che vi accede dove gli INDEX e le VIEW sono una via di mezzo fra base dati e metodi di accesso.

A questo punto si rende necessario verificare se tale primato rimanga costante anche al variare di certi parametri - in particolare uno fondamentale: la dimensione delle tabelle misurata in numero di righe:

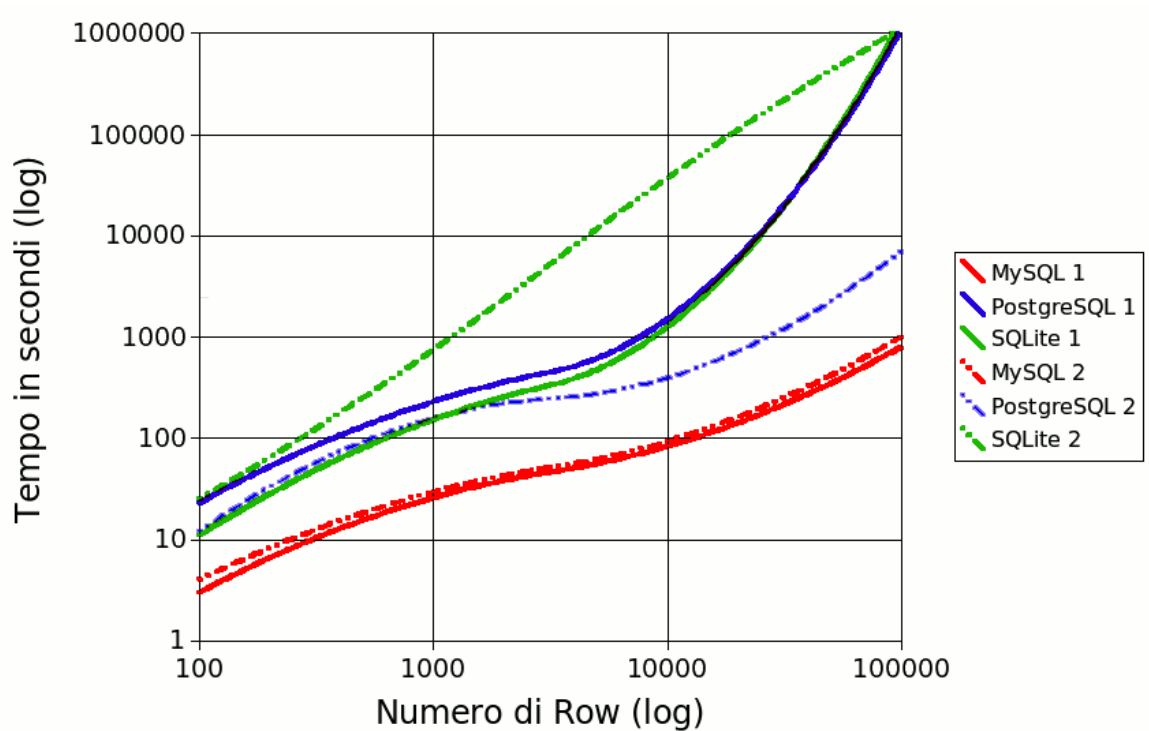


Figura 3:

Il grafico sopra mostra l'andamento dei tempi di esecuzione in funzione del numero di righe della tabella su cui si esegue l'istruzione INSERT. Entrambi gli assi sono di tipo logaritmico per agevolare la lettura.

Questo aspetto viene denominato capacità di scalare (scalabilità) rispetto alla dimensione della base dati e aggiunge un notevole grado di complessità alla valutazione degli RDBMS in scenari di utilizzo nei quali le prestazioni siano un obiettivo primario.

MySQL mostra anche in questo contesto di avere prestazioni buone e molto costanti rispetto alle piattaforme hardware.

4 Conclusioni

Un aspetto indubbiamente positivo messo in evidenza dall'analisi svolta è la complementarità che esiste tra i tre diversi RDBMS. Infatti non si può affermare che nessuno dei sistemi superi un altro sotto ogni punto di vista. Questo implica che ognuno abbia i propri punti di forza, così come delle debolezze.

Questa eterogeneità a prima vista potrebbe essere interpretata come risultato di uno sviluppo parziale dei prodotti presi in esame, tuttavia la loro maturità è riconosciuta generalmente. La causa delle suddette differenze si identifica quindi nelle scelte compiute dagli sviluppatori: questi RDBMS sono stati pensati come strumenti diversi per soddisfare esigenze diverse.

Non ci si trova nella situazione di dover scegliere quale sistema preferire e quali tralasciare; esistono invece diversi scenari di utilizzo dove ognuno dei sistemi rappresenta la scelta più appropriata.

4.1 MySQL

Come si è evidenziato durante l'analisi delle prestazioni, MySQL è stato progettato e sviluppato tenendo in considerazione la performance come parametro fondamentale. La struttura interna dell'RDBMS sottolinea questo proposito: sono stati sviluppati differenti motori di archiviazione, il cui compito è di gestire i file che contengono, in formato opportuno, le tabelle dei database. [19]

I principali motori (MyISAM, InnoDB, Archive, Merge, Memory) realizzano compromessi diversi tra funzionalità e prestazioni. La loro presenza permette all'amministratore di utilizzare il sistema più appropriato in base alle proprie esigenze.

Inoltre in MySQL è presente il supporto nativo per il clustering, le repliche master-slave e parallele.

Dal punto di vista della compatibilità con lo standard SQL questo RDBMS si può considerare adatto alla maggior parte degli utilizzi comuni.

Sono i server Web e quelli di posta elettronica che tipicamente non necessitano di funzioni avanzate e possono trarre il massimo beneficio dalla rapidità con cui MySQL risponde, anche in assenza di server dedicati esclusivamente al servizio di database. Non a caso MySQL è il sistema più utilizzato all'interno delle web farm.

Un'architettura dei server opportuna può incrementare notevolmente l'efficienza generale e, al tempo stesso, permettere una gestione semplificata e con un buon livello di sicurezza. La soluzione più diffusa prevede la separazione dei ruoli tra tre cluster: uno dedicato al servizio di frontend, sia esso Web o posta elettronica, uno dedicato alla gestione dei database e infine uno per lo storage.

Laddove vengono offerti servizi al pubblico la frequenza con cui si subiscono attacchi può essere estremamente elevata. La granularità elevata con cui MySQL gestisce i permessi è sicuramente un punto di vantaggio sul fronte della sicurezza.

4.2 PostgreSQL

PostgreSQL è l'RDBMS più ricco dal punto di vista delle funzionalità e del supporto dello standard SQL. E' in grado di fornire un buon supporto ai trigger e alle stored procedures e di

gestire moli di dati molto elevate. Inoltre ha lo spettro dei data types più ampio, nel quale include anche la valuta e supporta la definizione di nuovi tipi di dati.

L'utilizzo più appropriato è all'interno di realtà complesse, dove si verificano condizioni quali:

- Accesso ai server database da parte di una molteplicità di client, spesso realizzati in periodi temporali e con linguaggi differenti. In tale contesto si richiede all'RDBMS di garantire la correttezza dei dati attraverso varie funzioni di verifica, piuttosto che demandare questo compito ai client.
- La base dati ha dimensione elevata e i dati in essa contenuti hanno valore economico rilevante. Si tratta di una condizione frequente nelle grandi aziende dove si preferisce l'affidabilità rispetto all'efficienza computazionale. In questo contesto PostgreSQL compete con RDBMS come il noto prodotto sviluppato da Oracle piuttosto che con MySQL.
- Vengono utilizzate frequentemente le Stored Procedures. La possibilità di scrivere le Stored Procedures in quattro linguaggi (pgSQL, Tcl, Perl, Python) è una peculiarità di PostgreSQL. Le S.P. vengono utilizzate frequentemente per implementare della business logic all'interno dei db-server in maniera da centralizzare il trattamento dei dati.

Le prestazioni meno brillanti rispetto a quelle di MySQL diventano comprensibili alla luce di questi aspetti.

4.3 SQLite

Lo scenario di utilizzo di SQLite si dimostra conforme a quanto dichiarato[38] dall'autore sul sito ufficiale del prodotto ma può essere esteso comprendendo anche i seguenti utilizzi, compatibilmente con le funzionalità relativamente limitate di cui dispone:

- A bordo di sistemi dotati di limitate capacità di calcolo, memoria RAM e di memoria per il salvataggio permanente. Un caso tipico sono i sistemi embedded dove le modeste richieste⁶ di SQLite lo distinguono dagli altri RDBMS decisamente più esigenti.
- Può essere inserito, sotto forma di libreria, all'interno di applicazioni stand-alone. Ad esempio in tutti quegli applicativi che non necessitano di un database centralizzato e condiviso. In questo caso la compattezza di SQLite permette un'installazione semplice dell'applicativo in quanto non richiede la configurazione di database esterni. Trattandosi di un unico file, il backup e la rimozione del database gestito da SQLite è altrettanto semplice.
- Come si è potuto constatare dai risultati dei test esistono alcune condizioni di utilizzo in cui SQLite è il sistema più performante fra tutti. Un applicativo che richiede frequenti operazioni SELECT su un numero di campi e di tabelle limitati, quale ad esempio un dizionario, un forum web-based, un piccolo sistema di logging, etc. etc. può avvantaggiarsi dell'uso di SQLite anche sotto l'aspetto dell'efficienza.

⁶Basti pensare che SQLite ha un footprint di 250 KByte

- E' molto frequente l'utilizzo all'interno dei software di librerie di astrazione di database [37], che permettono all'applicativo di interfacciarsi a una molteplicità di RDBMS diversi in maniera trasparente rispetto al funzionamento dell'applicativo stesso. Questo vantaggio in termini di flessibilità è purtroppo compensato da una limitazione sui comandi SQL che possono essere utilizzati, che sono pari al sottoinsieme delle funzioni supportate da tutti gli RDBMS. Durante le fasi di sviluppo e test di un software di questo tipo può essere conveniente l'utilizzo di SQLite per giungere velocemente ad un prototipo funzionante, anche nella previsione di passare ad un RDBMS più completo successivamente.

Diritti d'autore

Licenza

Questo testo è sottoposto alla licenza Creative Commons Attribuzione 2.5 Italia. [39]

Tutti i marchi registrati citati appartengono ai rispettivi proprietari.

La licenza in sintesi

Chiunque è libero di riprodurre, distribuire, comunicare al pubblico, esporre in pubblico, rappresentare, eseguire, recitare e modificare quest'opera alle seguenti condizioni:

- Deve attribuire la paternità dell'opera nei modi indicati dall'autore o da chi gli ha dato l'opera in licenza.
- Ogni volta che usa o distribuisce quest'opera, deve farlo secondo i termini di questa licenza, che va comunicata con chiarezza.
- In ogni caso, è possibile concordare col titolare dei diritti d'autore utilizzi di quest'opera non consentiti da questa licenza. [40]

Riferimenti bibliografici

- [1] MySQL Clustering <http://dev.mysql.com/doc/refman/5.1/en/mysql-cluster-basics.html>
- [2] Introduzione a PostgreSQL <http://www.postgresql.org/about>
- [3] MySQL Data Types <http://dev.mysql.com/doc/refman/5.1/en/data-types.html>
- [4] PostgreSQL Data Types <http://www.postgresql.org/docs/8.2/interactive/datatype.html>
- [5] SQLite Data Types <http://www.sqlite.org/datatype3.html>
- [6] PostgreSQL CREATE TYPE <http://www.postgresql.org/docs/8.2/interactive/sql-createtype.html>
- [7] PostgreSQL Geometric Types <http://www.postgresql.org/docs/8.2/interactive/datatype-geometric.html> e Geometric Functions <http://www.postgresql.org/docs/8.2/interactive/functions-geometry.html>
- [8] MySQL Spatial Extensions <http://dev.mysql.com/doc/refman/5.1/en/spatial-extensions.html>
- [9] SQLite CREATE INDEX <http://www.sqlite.org/lang-createindex.html>
- [10] PostgreSQL Inheritance <http://www.postgresql.org/docs/8.2/interactive/ddl-inherit.html>
- [11] MySQL Storage Engines <http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html>
- [12] MySQL EXPLAIN <http://dev.mysql.com/doc/refman/5.1/en/explain.html>
- [13] PostgreSQL ANALYZE <http://www.postgresql.org/docs/8.2/interactive/sql-analyze.html>
- [14] MySQL ALTER TABLE <http://dev.mysql.com/doc/refman/5.1/en/alter-table.html>
- [15] PostgreSQL ALTER TABLE <http://www.postgresql.org/docs/8.2/interactive/sql-altertable.html>
- [16] SQLite ALTER TABLE <http://www.sqlite.org/lang-altertable.html>
- [17] MySQL Access Privilege System <http://dev.mysql.com/doc/refman/5.1/en/privilege-system.html>
- [18] PostgreSQL Roles and Privileges <http://www.postgresql.org/docs/8.2/interactive/user-manag.html>
- [19] MySQL Storage Engines <http://dev.mysql.com/doc/refman/5.1/en/storage-engines.html>
- [20] Mysql ODBC <http://dev.mysql.com/downloads/connector/odbc/3.51.html> e MySQL JDBC <http://www.mysql.com/products/connector/j/>
- [21] PostgreSQL ODBC <http://pgfoundry.org/projects/psqlodbc/> e JDBC <http://jdbc.postgresql.org/>
- [22] Documentazione sulla suite sql-bench <http://dev.mysql.com/doc/refman/5.0/en/mysql-benchmarks.html>
- [23] Altri benchmarks <http://dev.mysql.com/doc/refman/5.0/en/custom-benchmarks.html>
- [24] SuperSmack <http://vegan.net/tony/supersmack/> benchmarking your whole application under the worst possible load
- [25] osdb <http://osdb.sourceforge.net/index.php?page=home> the Open Source Database Benchmark (richiede pacchetti di sviluppo o installazione dei sorgenti dei vari database da testare)
- [26] Using MySQL to benchmark OS performance <http://software.newsforge.com/software/04/12/27/1238216.shtml?tid=72&tid=29> Pagina informativa sui vari programmi di benchmark per DB
- [27] <http://corpo.mandriva.com/xwiki/bin/download/Main/Technology/mysql-performance-whitepaper.pdf> Analisi dettagliata delle performances di MySQL documento di benchmark di MySQL
- [28] sysbench.sourceforge.net Un tool di analisi delle performance di sistema che possono influenzare maggiormente il funzionamento di un RDBMS: allocazione e trasferimento in memoria, I/O da file, efficienza dello scheduler. Sviluppato prevalentemente in base alle esigenze di MySQL.
- [29] <http://benchw.sourceforge.net/> Un semplice tool di benchmarking presente su Sourceforge. Supporta MySQL, PostgreSQL e altri 6 RDBMS

- [30] TPC <http://www.tpc.org> il TPC definisce uno standard per il benchmark di RDBMS
- [31] OSDL http://www.osdl.org/lab_activities/kernel_testing/osdl_database_test_suite OSDL ha sviluppato l' "OSDL Database Test Suite", una collezione di strumenti di test.
- [32] <http://www.sqlite.org/cvstrac/wiki> Documentazione su SQLite
- [33] <http://www.db-gonzales.de/html/download.html> Programma di benchmark per SQLite per windows
- [34] <http://www.initd.org/tracker/pysqlite/wiki> SQLite per Python
- [35] <http://www.debianadmin.com/top-84-mysql-performance-tips.html> 84 suggerimenti su come ottimizzare le prestazioni di MySQL
- [36] <http://www.sqlite.org/speed.html> Comparazione delle prestazioni di SQLite con altri RDBMS presente sul sito ufficiale
- [37] Due esempi di librerie di astrazione: DBI - librerie di astrazione DB per Perl <http://dbi.perl.org/ADOdb> - librerie di astrazione di DB per Php e Python <http://adodb.sourceforge.net/>
- [38] <http://www.sqlite.org/whentouse.html> Consigli sui tipi di utilizzo appropriati per SQLite
- [39] <http://creativecommons.org/licenses/by/2.5/it/legalcode> Testo integrale della licenza Creative Commons adottata
- [40] <http://creativecommons.org/licenses/by/2.5/it/> Sintesi della licenza Creative Commons