Università degli studi di Padova

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di

Tecnologie Open Source

Analisi del progetto

Docente
Luigi Bellio

Studente
Luca Dei Zotti

Anno Accademico 2008/2009

Prefazione

Dall'introduzione del termine *Open Source* il mondo ha fatto numerosi passi avanti. Dalle cantine, dalle camere e dalle scrivanie di pochi smanettoni, il codice aperto è passato a diventare un'innovativa idea di businness, nella cui forma più pura ognuno ha potenzialmente dei vantaggi.
Migliaia di occhi che ispezionano i programmi alla ricerca di falle ed ottimizzazioni possibili, programmi limpidi, comunità che permettono a chi lavora di svolgere bene il proprio compito e a chi usufruisce dei prodotti finali di ottenere il massimo dal proprio software.
Per questo, a ventiquattro anni di distanza dall'emblematica fondazione della *Free Software Foundation*, studiare lo sviluppo e l'evoluzione di software open source di successo può aiutare a comprendere quanto importante e funzionale sia l'idea che ne costituisce il nucleo.
Ci si soffermerà in particolare sull'analisi della storia di MySQL, come simbolo del successo mondiale di un prodotto che ha rivoluzionato le credenze sull'open source, e del significato che le sue ultime evoluzioni finanziarie hanno per il mondo del software aperto.

Indice

1 Introduzione **1**

2 Visione **3**
2.1 Gli Obiettivi 3
2.2 MySakila 3

3 Mercato **5**

4 Storia **7**
4.1 Nascita 7
4.2 Lo Sviluppo Iniziale 7
4.3 Il Primo Cataclisma 8
4.3.1 L'acquisizione da parte di Sun 8
4.3.2 Le reazioni 9
4.3.2.1 L'amarezza di Forbes 9
4.3.2.2 O'Reilly: una visione più ampia 10
4.4 Il Secondo Cataclisma 10
4.4.1 L'acquisizione di Sun da parte di Oracle 10
4.4.1.1 La guerra di Monty 11
4.4.2 Considerazioni 11

5 Licenza **13**
5.1 Il modello Aladdin 13
5.2 Dual Licensing 13
5.2.1 Osservazioni sul modello di Dual Licensing 15

6 Business Model **17**
6.1 Quid pro Quo 17
6.2 Critiche al modello di business di MySQL 18

7 Development Process **19**
7.1 Dalla foresta al vaso: il modello di trunk 19
7.2 Milestones 20

8 La comunità **23**
8.1 Il sito 23
8.2 Mailing List e Newsletter 23

8.3 Il forum 24
8.4 Documentazione 24
8.5 Articoli e Blog 24
8.6 Meetups e Gilde 24
8.7 IRC 25

9 Strumenti di sviluppo **27**
9.1 MySQL DevZone 27
9.2 La Forgia 27
9.3 Bazaar 27
9.4 Bug Reporting 28
9.5 Revisione del Codice 28

Capitolo 1

Introduzione

MySQL, la cui pronuncia corretta $mai, \varepsilon skju :' \varepsilon l$ è spesso sostituita dagli addetti del settore con la più intuitiva *mai siquel*, è un Database Management System relazionale sviluppato da TcX ataconsult, ad oggi MySQL AB, attualmente molto diffuso nei più svariati campi di applicazione. Statistiche risalenti al 2008 [3] indicano per questo software almeno 11 milioni di utilizzatori, rendendo di fatto MySQL una delle punte di diamante del mondo Open Source, tanto da attirare le attenzioni di uno dei *big* della Silicon Valley: *Sun Microsystems*, la cui acquisizione da un miliardo di euro sarà approfonditamente trattata in 4. Ma gli esempi di utilizzo di MySQL spaziano oltre, dai piccoli siti web alle applicazioni interne di un altro dei grandi del mondo informatico odierno, Google [5], la quale fra l'altro ha apportato ad alcune release di MySQL alcune modifiche, rilasciate sotto il nome di Google MySQL Tools [7]. Proprio nel web development MySQL ha potuto affondare maggiormente le radici, diventando la base per piattaforme di sviluppo come LAMP, utilizzata fra gli altri dalla famosa enciclopedia online Wikipedia. Il DBMS è scritto in C/C++ ed utilizza tools come automake e autoconf, il che apre la strada alla compatibilità verso numerosi sistemi. Attualmente, infatti, MySQL è completamente supportato da un'ampia varietà di sistemi operativi, un cui parziale elenco è AIX, AmigaOS, BSDi, Unix, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell NetWare, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, SGI Irix, Tru64 e la serie di sistemi Windows successivi a Windows 95. Altresì è possibile interfacciarsi con svariati linguaggi tramite appositi driver, tra cui C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, Tcl nonché con le piattaforme Mono e .Net.

L'evoluzione di MySQL, che sarà analizzata approfonditamente in seguito, è piuttosto interessante dal punto di vista del mondo Open Source, giacchè ' la sua licenza ed i suoi modelli di sviluppo sono piuttosto controversi, e sono stati spesso oggetto di ambiguità e contestazioni talvolta anche dal lato dei lavoratori interni stessi. Proprio da queste diatribe più o meno interne sono nati i vari fork del progetto, fra i quali vale la pena di citare *Drizzle*, basato su un'architettura a microkernel, *Proven Scaling*, nato per mantenere la Community Edition di MySQL, gradualmente lasciata decadere dalla società a favore della versione Enterprise, e *Maria*, creato per supportare l'omonimo storage engine.

Capitolo 2

Visione

2.1 Gli Obiettivi

L'intenzione iniziale degli sviluppatori MySQL era quella di utilizzare il sistema di database mSQL per collegare tabelle tramite le routines ISAM (*Indexed Sequential Access Method*). Tuttavia, i primi test svelarono che il motore mSQL non risultava sufficientemente veloce, nè tantomeno flessibile. I contatti con l'autore di mSQL rivelarono che egli stava già lavorando alla seconda versione del motore, e che di conseguenza aveva già approntato una propria infrastruttura di indicizzazione. MySQL nasce quindi come nuova interfaccia al database, ma con un'interfaccia API praticamente identica a mSQL, progettata per permettere il porting agevole di codice di terze parti scritto per mSQL su MySQL.

Lo sviluppo di MySQL non si è ovviamente fermato all'interfacciamento con mSQL, divenendo ben presto un sistema DBMS completo, seppur fortemente orientato oggi al web dinamico, e in rapida crescita grazie alle idee che verranno analizzate in seguito, fino a raggiungere un livello di distribuzione tale da potersi mettere in competizione con big del settore quali Microsoft SQL Server e Oracle.

2.2 MySakila

Non è chiara quale sia l'origine del nome *MySQL*. Da una parte si sa che un gran numero delle librerie utilizzate nell'azienda del principale fondatore, Michael Widenius, avevano come prefisso my. Allo stesso modo, la figlia di Widenius stesso si chiama My.

L'origine del nome del delfino che troneggia nel logo di MySQL, *Sakila*, è invece più chiara. Esso fu scelto dai fondatori dell'attuale MySQL AB grazie ad un apposito concorso. Il propositore del nome fu Ambrose Twebaze, sviluppatore Open Source africano, che lo scelse dal linguaggio Swaili, il SiSwati.

Capitolo 3

Mercato

É indubbio il fatto che MySQL rappresenti la più grande realtà del mondo dei database Open Source. Uno studio del 2008, scaricabile a pagamento da Forrester [23], una compagnia indipendente di ricerca in campo finanziario, lo colloca in testa alle classifiche di utilizzo sia gratuito che a pagamento nel campo dei prodotti Open Source, nonché' prima fra le compagnie con i più alti indici di crescita del settore.

Ma MySQL non è una grande solamente nel mondo aperto. Uno studio precedente [8], condotto da Evans Data Corporation (EDC) e datato 2007, mostra la crescita dell'utilizzo di MySQL negli anni precedenti, e fornisce credibili previsioni sull'andamento del progetto in relazione al crescente utilizzo dei software open source, nonché dell'aumento del numero degli sviluppatori.

In figura 3.1 possiamo notare i risultati di un sondaggio risalente al 2004, che indica MySQL terzo fra tutti i propri concorrenti, con un terzo degli utilizzi.

Molto del merito per la diffusione di MySQL può essere attribuito anche al successo della piattaforma LAMP, di cui rappresenta la M.

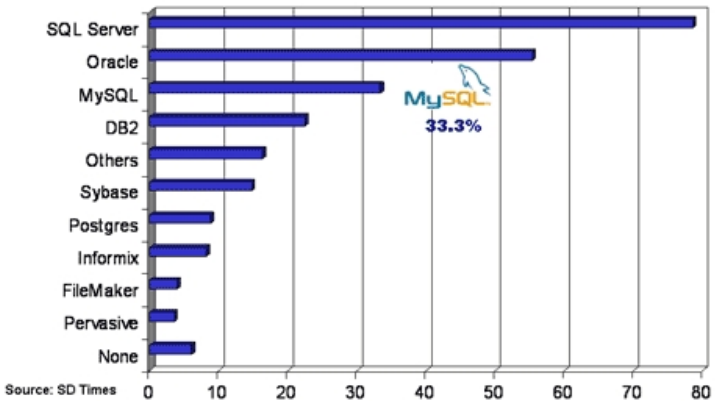

Figura 3.1: Statistica sugli utilizzi dei principali DBMS

Capitolo 4

Storia

4.1 Nascita

La sezione 2 ha descritto MySQL come una reazione ai difetti e alle mancanze di mSQL.
Tuttavia, relegarlo a tale origine sarebbe quantomeno limitativo.
Michael Monty Widenius, il suo principale inventore, lavorava come (unico) sviluppatore nella compagnia svedese TcX, e la sua esperienza pratica in basi di dati risale almeno al 1979. Risale a quello stesso anno infatti UNIREG, un tool da egli approntato per la gestione di basi di dati, sviluppato negli anni fino a raggiungere ottimi risultati anche nella gestione di grandi moli di dati e riscritto in vari linguaggi (in C sotto UNIX circa dal 1986).
La svolta avvenne quando, nel 1994, TcX iniziò a sviluppare applicazioni basate su web, rendendosi conto che UNIREG, a tale scopo, creava un eccessivo overhead per la generazione di contenuto dinamico. Fu così che TcX si interessò a mSQL, il quale si trovava al tempo alla propria versione 1.x, ancora senza supporto per l'indicizzazione.
Widenius contattò David Huges, autore di mSQL, per proporgli la connessione del proprio prodotto con le strutture ISAM di UNIREG. Huges, dal canto suo, era già al lavoro per la versione 2 del proprio pupillo, ed aveva già approntato la propria struttura di indicizzazione, tanto che TcX decise di creare la propria versione di un database server adatto ai propri bisogni.

4.2 Lo Sviluppo Iniziale

MySQL non fu un prodotto reinventato da zero. TcX ebbe l'idea vincente di costruire per UNIREG una serie di API inizialmente estremamente simili a quelle di mSQL: in questo modo da una parte era possibile sfruttare strumenti di terze parti creati per mSQL, e dall'altra gli utenti mSQL che avessero voluto migrare al crescente sistema TcX avrebbero dovuto apportare solo poche, banali modifiche al proprio codice. Va comunque fatto notare, a scanso di equivoci, che il codice di TcX era completamente originale.

MySQL 1.0 vede la luce nel Maggio 1995, e vede immediatamente le pressioni di David Axmark e della sua Detron HB, un partner di TcX, riguardo il rilascio

del server MySQL su Internet. Tali pressioni erano mirate a testare il modello di business già collaudato da L Peter Deutsch, fondatore di Aladdin Enterprises e creatore di Ghostscript. La sua visione verrà affrontata nel capitolo 5. Quello che al momento è rilevante sapere è che tale modello di licenza rese MySQL più libero ed accessibile di mSQL.

Lo sviluppo di MySQL è stato, dal 1995 in poi, piuttosto lineare. Si sono susseguite, a distanza di non troppo tempo, numerose versioni di rilascio, dalla 3.23 del 1999, che possiamo definire la prima versione di produzione, alla 5.1 del 2005. MySQL si è affermato, come visto nella sezione 3, come concorrente competitivo di sistemi blasonati e validi, divenendo al contempo un pilastro del mondo Open Source che sembrava inamovibile.
Ma, come recita un vecchio adagio, *Vuoi far ridere Dio? Confessagli i tuoi piani...*

4.3 Il Primo Cataclisma

4.3.1 L'acquisizione da parte di Sun

The Network is the Computer

Visione di Sun

Il 16 Gennaio 2008, dopo mesi di indiscrezioni e voci, Sun Microsystems acquista MySQL AB per la cifra di 1 miliardo di dollari, stimando di fatto il mercato dei database 15 miliardi di dollari [12]. E gettando nel panico la comunità Open Source.

L'intenzione di Sun può essere compresa usando le parole dell'amministratore delegato di Sun stessa, Jonathan Schwartz:

> *Today's acquisition reaffirms Sun's position at the center of the global Web economy. Supporting our overall growth plan, acquiring MySQL amplifies our investments in the technologies demanded by those driving extreme growth and efficiency, from Internet media titans to the world's largest traditional enterprises. MySQL's employees and culture, along with its near ubiquity across the Web, make it an ideal fit with Sun's open approach to network innovation. And most importantly, this announcement boosts our investments into the communities at the heart of innovation on the Internet and of enterprises that rely on technology as a competitive weapon.*

Marten Mickos, CEO di MySQL, dal canto suo rincara la dose:

> *The combination of MySQL and Sun represents an enormous opportunity for users and organizations of all sizes seeking innovation, growth and choice. Sun's culture and business model complements MySQL's own by sharing the same ideals that we have had since our foundation – software freedom, online innovation and community and*

> *partner participation. We are tremendously excited to work with Sun and the millions of members of the MySQL open source ecosystem to continue to deliver the best database for powering the modern Web economy.*

Evidente quindi si delinea la mossa di Sun, un'espansione del proprio target di utenza e servizio, l'inserimento in un nuovo, promettente mercato sotto un'almeno apparente facciata di appoggio al mondo Open Source.

4.3.2 Le reazioni

Se la prima reazione del mondo Open Source è stata, se non altro, di aperta diffidenza nei confronti del colosso Sun, una serie di ragionamenti a freddo hanno potuto calmare le acque.
Gli interessi di Sun nel settore LAMP erano infatti troppo ampi per potersi permettere la compromissione dei rapporti con la comunità Open Source, e sebbene fosse ovvia la lieve spinta verso l'adozione di soluzioni tecniche di Sun (primi fra tutti il linguaggio Java e il file system ZFS), essa non avrebbe mai potuto a breve termine tramutarsi in una costrizione. La soluzione di Sun sembra quindi una buona mediazione fra la qualità delle soluzioni Open Source e la sempiterna ricerca delle istituzioni di partner commerciali consolidati.

Tuttavia, questa mossa non è stata indolore.

4.3.2.1 L'amarezza di Forbes

Il 31 Gennaio 2008, la rivista finanziaria Forbes pubblica un articolo [10] dai toni piuttosto amari sull'acquisizione di MySQL AB da parte di Sun, dal titolo *Cash Me Out*, seguito da un sottotitolo autoesplicativo: *The end of open source as counterculture.*.

In tale articolo, Daniel Lyons esprime le proprie tristi constatazioni su quanto il movimento Open Source si sia spostato, dalle cantine di *Hackers radicali ai margini dell'industria informatica, disinteressati ai soldi* descritte nell'articolo *Peace, Love and Software*, alle affollate strade del mainstream. Emblematiche le sue parole

> *These deals have nothing to do with peace, love and software, and everything to do with money. The open source guys realize they can't build a decent business unless they hook up with the old closed source guys.*

Allo stesso modo si indica quanto il mondo Open Source sia in realtà utilizzato dalle grandi società del mondo del business per minacciare i propri concorrenti, come fa IBM con i propri finanziamenti verso il mondo Linux per intimorire Microsoft, e Microsoft stessa, stringendo accordi con Novell per poter sottrarre fette di mercato a RedHat.

In sostanza, la visione negativistica di Forbes smitizza la libertà , la condivisione, l'idealismo dell'Open Source, sottolineandone lo stretto rapporto con l'arido mondo della finanza e della monetizzazione delle idee, ed allo stesso modo

critica, con un sarcasmo che merita sicuramente la lettura, la mossa finanziaria di Sun, già amputata di parte dei proventi di molti suoi prodotti di punta dopo averli resi open source (come il succitato Java, il sistema operativo Solaris o i progetti dei microprocessori Sparc).

4.3.2.2 O'Reilly: una visione più ampia

In risposta all'articolo di Forbes, il 20 Febbraio 2008 O'Reilly pubblica nel proprio blog un articolo in cui esprime una visione più ottimistica dell'acquisizione [20].

In particolare, egli si basa su un proprio precedente articolo [21] per spiegare come la comunità Open Source non sia stata sepolta dal grigio mondo del mercato, ma si sia in realtà spostata su un altro livello, allo stesso modo in cui la leadership del mercato è passata da IBM a Microsoft quando la rivoluzione del PC ha portato il core business dall'hardware al software. Sottolinea inoltre quanto il modello di business Open Source si volga al software come prodotto, e come ogni rivoluzione informatica si possa tramutare in un'opportunità per i vecchi del settore, portando come esempi proprio l'hardware come commodity che Dell e HP hanno sfruttato a loro tempo.

Una visione, in sostanza, che vede il mondo del codice aperto solamente spostato di pochi metri, sempre pronto a rispondere agli attacchi alla propria libertà con il polimorfico spirito di flessibilità e adattamento che caratterizza i suoi prodotti.

4.4 Il Secondo Cataclisma

4.4.1 L'acquisizione di Sun da parte di Oracle

> Io penso che nessuno possa essere il padrone di un progetto Open Source; i progetti sono definiti dal project leader e dagli sviluppatori che ci lavorano. Se una compagnia perde la fiducia di queste persone, allora essi possono andarsene, fare un fork del progetto e svilupparlo secondo le loro intenzioni
>
> M.M. Widenius

Un secondo cataclisma sconvolge l'ancora inquieto mondo dell'Open Source il 20 aprile 2009, quando Oracle annuncia l'acquisizione di Sun Microsystems [11]: una manovra da 7,4 miliardi di dollari.
Questa mossa, se possibile, si rivela ancora più problematica, da punti di vista tanto etici quanto pratici, di quella precedente di Sun: Oracle era infatti uno dei più serrati concorrenti di MySQL nel campo delle basi di dati.

Sono legittimi quindi i dubbi della comunità : MySQL sopravviverà o si tratta dell'ennesima manovra di acquisizione a buco nero, dove un prodotto viene comprato e lasciato a spegnersi lentamente?

Per il giornalista tecnologico John Dvorak, la faccenda è semplice: si tratta di una sorta di complotto [6] atto ad uccidere MySQL in favore del prodotto Oracle, come si evince dalle direttissime parole *I'm close to being convinced that Oracle wanted to buy MySQL to kill the product, but knew it couldn't pull off the stunt itself. So it sent in a stooge to do the job.*
Non tutti tuttavia concordano con la visione sensazionalistica ed apocalittica, probabilmente influenzata dalla deformazione professionale, del giornalista.

Le prime dichiarazioni di Oracle-Sun d'altra parte non sono utili a riguardo. Oracle spiega l'interesse verso le tecnologie Java e Solaris, ma nell'annuncio ufficiale MySQL non viene neppure nominato.
Eppure, la rete pullula di indiscrezioni, notizie e previsioni. Nonché di qualche azione più eclatante.

4.4.1.1 La guerra di Monty

Se a qualcuno la mossa di Oracle non è stata proprio gradita, questi è certamente Widenius stesso.
Egli, dopo aver già lasciato Sun a causa di diatribe con il management ed aver fondato il Monty Program AB [15] per lavorare al nuovo motore transazionale di MySQL, Maria, risponde con un seccato ma lucido comunicato tramite il proprio blog [14] a chi gli chiede impressioni e opinioni, ma lascia altresì un messaggio ai quartieri alti di Oracle:

> *Mr. Ellison, you are undoubtably a master tactician. However, thinking two moves ahead in the open source world is not good enough. You need strategy. Long term, meaningful, viable strategy. You need to think years ahead, not just to the next fiscal quarter.*

Contemporaneamente, eccolo formare l'Open Database Alliance [16], consorzio con lo scopo di fornire un punto di riferimento per lo sviluppo di MySQL e di MariaDB, ed incoraggiare un ambiente di sviluppo aperto supportato da una comunità , assicurando che la qualità del codice di MySQL rimanga alta ed invitando, fra un riferimento ad Oracle e l'altro, tutte le aziende interessate all'utilizzo di tali strumenti ad unirvisi.

4.4.2 Considerazioni

L'acquisto di Sun Microsystems da parte di Oracle è sicuramente stato un colpo piuttosto duro per i puristi della filosofia Open Source. Va infatti ricordato che Sun è responsabile non solo dello sviluppo del neoacquisito MySQL, ma anche di progetti come OpenOffice, potenziale concorrente della famosa suite di casa Microsoft, VirtualBox, software di virtualizzazione per ora ancora suddito del leader del settore, VMware; per non parlare dei già citati Solaris, Java e ZFS.
Se gli ultimi tre sono stati apertamente mirati da Oracle nell'acquisizione, i primi due possono preoccupare, perchè non strettamente legati al core business dell'azienda.

La licenza GPL viene in soccorso della comunità , impedendo che il codice di questi prodotti venga chiuso, ed è plausibile pensare che nessuno di un management anche solo vagamente lungimirante possa decidere di inimicarsi un'intera comunità di sviluppatori. Uccidere un progetto Open Source supportato da una comunità così vasta è d'altronde molto difficile, proprio grazie alla succitata licenza GPL, che ne permetterebbe immediati fork come quelli che già MySQL ha visto nascere.
Va comunque fatto notare che l'improvvisa mancanza di una parte così consistente dei finanziamenti si tradurrebbe in una battuta d'arresto rilevante per ognuno dei progetti coinvolti, ma non per forza letale, come hanno dimostrato già vari progetti mutilati di fondi governativi o aziendali [9].

Un'insieme infinito di combinazioni si apre ora davanti al mondo informatico.

Capitolo 5

Licenza

5.1 Il modello Aladdin

Come già accennato nella sezione 4.2, la prima versione di MySQL venne rilasciata secondo il modello della Aladdin Enterprises di L Peter Deutsch. Vale la pena di riportare un commento di Deutsch riguardo la Aladdin Free Public License:

> *This License is not an Open Source license: among other things, it places restrictions on distribution of the Program, specifically including sale of the Program. While Aladdin Enterprises respects and supports the philosophy of the Open Source Definition, and shares the desire of the GNU project to keep licensed software freely redistributable in both source and object form, we feel that Open Source licenses unfairly prevent developers of useful software from being compensated proportionately when others profit financially from their work. This License attempts to ensure that those who receive, redistribute, and contribute to the licensed Program according to the Open Source and Free Software philosophies have the right to do so, while retaining for the developer(s) of the Program the power to make those who use the Program to enhance the value of commercial products pay for the privilege of doing so.*

La licenza di Ghostscript, infatti, era di tipo detto *scheduled*: prevedeva infatti che le nuove versioni non fossero GPL, mentre quelle vecchie lo diventassero dopo un certo tempo, in genere un anno. Inoltre, la licenza Aladdin, differentemente da quelle supportate dalla Free Software Foundation, non prevedeva la reciprocità della licenza nel prodotto distribuito.

5.2 Dual Licensing

MySQL viene spesso utilizzato come esempio quando si tratta di descrivere il modello di Dual Licensing, poichè il prodotto è distribuito sia secondo la licenza GPL (a partire dal 2000) che sotto una licenza commerciale separata.
Questo avviene poiché' MySQL è spesso incorporato in progetti più vasti, che a

seconda dell'interpretazione data alla licenza GPL possono di conseguenza divenir soggetti al vincolo di reciprocità della licenza. Per alcuni potenziali utilizzatori il rilascio del sorgente del loro prodotto derivato è tuttavia una condizione inaccettabile. Tale problema è risolto appunto rilasciando MySQL con una seconda licenza commerciale che, dietro pagamento, permette di non esser obbligati a mantenere il codice aperto.

Marten Mickos, CEO di MySQL fino al Febbraio del 2008, descrive la doppia licenza con queste parole:

> *Our paying customers get what they pay for: a commercially supported product with a level of assurance from the vendor and without any typical open source requirement that linked software must be open sourced as well [...] Dual licensing allows companies to build viable long-term businesses while at the same time accommodating the needs of the open source/free software community.*

Le sue parole saranno meglio analizzate nel capitolo 6.

Per rientrare nei termini della FOSS (Free and Open Source Software) License Exception, il prodotto derivato deve essere distribuito secondo una delle seguenti licenze approvate da MySQL:

- Release Early Certified Software
- Academic Free License v2.0
- Apache Software License v1.0/1.1/2.0
- Apple Public Source License v2.0
- Artistic license From Perl v5.8.0
- BSD license del 22 Luglio 1999
- Common Development and Distribution License (CDDL) v1.0
- Common Public License v1.0
- Eclipse Public License v1.0
- GNU Library or Lesser General Public License (LGPL) v2.0/2.1/3.0
- Jabber Open Source License v1.0
- MIT License
- Mozilla Public License (MPL) v1.0/1.1
- Open Software License v2.0
- OpenSSL license (SSLeay) 2003 (1998)
- PHP License v3.0/3.01

- π Python license (CNRI Python License)
- π Python Software Foundation License v2.1.1
- π Sleepycat License, 1999
- π University of Illinois/NCSA Open Source License
- π W3C License, 2001
- π X11 License, 2001
- π Zlib/libpng License
- π Zope Public License v2.0

Da questo modello di licenza si evince quindi che il produttore del derivato può rientrare nella licenza FOSS, purché rispetti i termini della licenza GPL. In caso contrario, si rientrerà nella licenziazione commerciale.

5.2.1 Osservazioni sul modello di Dual Licensing

Il problema principale del modello di doppio licenziamento è che contributi di terze parti alla versione GPL di MySQL devono essere a loro volta licenziati sotto GPL. I detentori del copyright possono autorizzare il doppio licenziamento, ma non sono obbligati a farlo. Per ovviare a questo problema si può agire in diverse maniere, dal richiedere che l'autore ceda all'azienda il copyright del contributo, o direttamente dal discriminare la collaborazione tramite una licenza apposita che permetta la libertà voluta all'azienda.
Dal canto suo, MySQL-Sun ha agito esattamente in quest'ultimo modo, vincolando i contributi a una Contributor License Agreement (CLA) [13], scelta che tuttavia ha attirato le critiche di una parte della comunità , come si vedrà nella sezione 6.2.

Capitolo 6

Business Model

6.1 Quid pro Quo

Il modello di business adottato da MySQL si fonda sul cosiddetto *Quid pro Quo*: qualcosa in cambio di qualcosa.
Il già citato Mickos spiega che gli utilizzatori commerciali del software traggono benefici dagli utilizzatori Open Source grazie all'intrinseca affidabilità garantita da un *rigorous battle testing*, per utilizzare le sue parole. Allo stesso tempo, il mondo Open Source trae vantaggio da quello commerciale, poiché i proventi delle vendite permettono all'azienda di continuare ad investire nei miglioramenti del prodotto.

Il contributo dell'azienda al mondo libero tuttavia non si ferma al database server. Nel 2003 MySQL ha acquisito da Ericsson NDB Cluster, uno strumento per la gestione di tabelle su cluster, rilasciandolo sotto GPL, mentre l'anno dopo è stata la volta di Eventum, un *trouble ticket management system*, senza contare l'impressionante mole di documentazione e di tools grafici a supporto dei propri prodotti. Più direttamente, MySQL contribuisce tramite donazioni alla Free Software Foundation, ed è attivista nella lotta contro i brevetti software.

La FOSS stessa, proclama Kaj Arnö [1], è un grande contributo alla comunità , poichè permette agli sviluppatori una metodologia di licenziamento flessibile.

Ovviamente, MySQL non è un ente di beneficenza, e deve quindi far rientrare ogni costo. L'azienda riceve la propria retribuzione da numerose attività :

- π Certificazione e formazione
- π Consulenza e supporto tecnico (performance, migrazioni, progettazione...)
- π Modifiche personalizzate, esterne allo sviluppo previsto
- π Diritti di copyright su nomi e loghi
- π Licenze commerciali di MySQL, per chi vuol mantenere il proprio codice chiuso o per chi desidera affidarsi alla versione Enterprise del sistema.

Da qualche tempo, inoltre, MySQL fornisce un servizio differenziato, offrendo agli utenti della versione Enterprise alcuni add-on a pagamento.
Cosa che, come vedremo, non è passata inosservata...

6.2 Critiche al modello di business di MySQL

Il pluricitato Marten Mickos scrisse a suo tempo una lista di 13 modelli di sviluppo open source, purtroppo ora non più disponibile nel post originale. Senza citarli tutti, rimandando a tale scopo la lettura di [2], si utilizzerà la stessa nomenclatura e numerazione per sottolineare che la principale critica che si fa a MySQL è di essere passata dal modello *3. Software is free but if you embed it in closed source, you better pay a fee* (Il software è libero, ma se vuoi includerlo in un progetto a codice chiuso devi pagare un tributo) al modello *6. Software is free but some enterprise features are not* (Il software è libero, ma alcune caratteristiche non lo sono).

Il terrore della comunità , inizialmente acuito dall'acquisizione di Oracle e per ora smentito (si veda 7) da poco, che si possa giungere ai modelli *7. Software is free but we built a closed-source product around it* (Il software è libero ma abbiamo costruito un prodotto chiuso attorno ad esso) o *9. Software is free but we sell everything else on the planet, including closed source software* (Il software è libero, ma noi vendiamo qualunque altra cosa, incluso codice chiuso).

Il problema di MySQL non è quindi tanto un discorso di funzionamento del modello, quanto di immagine pubblica.
Mickos difende le proprie posizioni con queste parole:

> *If the world were perfect, we would only produce GPL code and we would have a great business that can fund the software development. But we have found that the world is not perfect. We have been experimenting with a variety of business models around FOSS (dual licensing, support only, simple subscriptions, different binaries for community and enterprise, non-open source features) to find the best one. And we will continue to experiment until we are satisfied. We need to find a model that allows us to produce a ton of great code under GPL while having the financial strength to do all this*

Altre voci critiche al modello di business di MySQL asseriscono come, in realtà , questo non sia veramente un modello open source: la colonna portante di questa affermazione è il modello di sviluppo vincolato alla Contributor License Agreement (CLA), che richiede all'autore di rinunciare a diritti fondamentali sul codice, spesso per vedere le proprie patches stagnare per anni e non venir incluse nelle releases successive. Un'altra critica si basa sul fatto che, solitamente, un prodotto open source viene sviluppato da una rete decentralizzata di pari, esattamente il contrario di quanto avviene per MySQL.
Secondo queste critiche, più o meno fondate a seconda di quanta importanza si dia al fatto che MySQL sia, a tutti gli effetti, un'azienda che deve aver controllo sui propri prodotti, il prodotto dovrebbe essere definito *Free Software*, ma non *Open Source*.

Storie di seconda mano [22] mettono altresì in bocca ad addetti del reparto vendite, fra l'altro totalmente esterni a quello che MySQL rappresenta, le poco lusinghiere parole *You have to give us something we can sell. This isn't working.* Un insieme di voci, in definitiva, che tacciano la compagnia di un'ipocrisia inaccettabile per chi si è sempre mostrato aperto al mondo del software open.

Capitolo 7

Development Process

7.1 Dalla foresta al vaso: il modello di trunk

Per molti anni, MySQL ha mantenuto un modello di rilascio piuttosto tradizionale, che vedeva la definizione di una serie di nuove caratteristiche ed il rilascio di una nuova versione al momento della loro implementazione.
Con gran sollievo della comunità , Oracle-Sun ha recentemente annunciato [19] un nuovo modello di sviluppo, più dinamico e moderno, con una maggior frequenza di rilasci, confermando per ora in tal modo il proprio interesse nel non lasciar morire il progetto. In questo modo, mentre per chi è solamente interessato a mantenere il proprio server stabile non ci saranno grossi cambiamenti, gli utenti più smaliziati avranno a disposizione una maggior scelta di versioni da testare.

Un riassunto veloce del nuovo modello di sviluppo può essere dato dai seguenti punti:

- π l'albero ufficiale dei sorgenti (in gergo *trunk*) sarà sempre almeno in uno stato di beta
- π le milestones non inizieranno più in alpha, ma almeno in beta, come merge tra il trunk e l'albero di sviluppo (*stage tree*)
- π il rilascio di una Release Candidate (RC) dopo il raggiungimento di una milestone avverrà con frequenza tra i 3 e i 6 mesi
- π la finestra di integrazione tra una milestone e la successiva consente l'inserimento di nuove funzionalità dall'albero di sviluppo;
- π i rilasci Generally Available (GA) avverranno con frequenza tra 12 e i 18 mesi;
- π il supporto non coprirà più di due release.

Il concetto del *trunk* deriva dalla passata esperienza di MySQL, che ha visto lo sviluppo contemporaneo di più versioni del software, e di conseguenza di più alberi dei sorgenti. Questo ha portato ovviamente numerosi problemi in fase di bug fixing, di merging del codice per creare una GA e di gestione in generale. Così , il nuovo modello prevede di passare ad un singolo trunk, aggiornato più spesso e in grado di raggiungere uno stato stabile ad intervalli regolari.

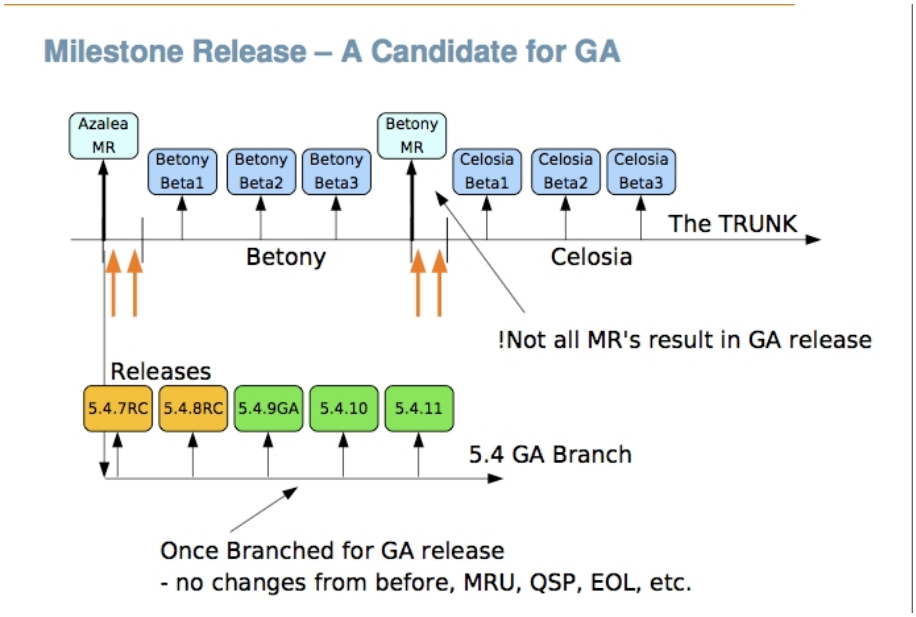

Figura 7.1: Milestones per le nuove versioni di MySQL server

7.2 Milestones

Una milestone è formata da tre fasi: nella prima il trunk è aperto all'accettazione di nuove modifiche, per un tempo che viene detto Integration Window. Alla fine di questo periodo il trunk rimane aperto solo per il bugfixing, fino a quando il codice non raggiunge una maturità adeguata per essere definito una Release Candidate.
Quando il trunk raggiunge una qualità sufficiente e le features annunciate sono state implementate, si dice che la milestone è raggiunta. Se nel periodo di testing (stabilizzazione) vengono riscontrati bugs, e non si riesce a stabilizzare la RC entro le tempistiche previste, la funzionalità viene rimossa dalla RC.

Nel periodo che va dal raggiungimento di una milestone (*Milestone Release*) alla definizione di nuove funzionalità da aggiungere, si deve prendere la decisione se tramutare o meno la MR in una GA.
In figura 7.2 si può vedere un confronto fra la vecchia modalità di sviluppo e la nuova.

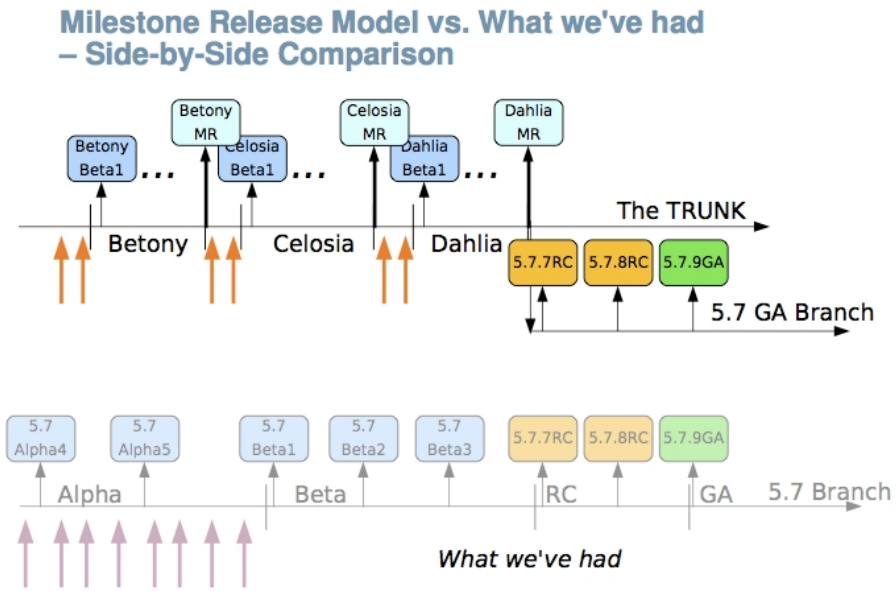

Figura 7.2: Comparazione fra vecchio e nuovo Development Cycle

Capitolo 8

La comunità

Come ogni progetto Open Source di un certo successo, a MySQL non può mancare una community cresciutagli attorno nel tempo e svilupattasi con esso. Gli strumenti a disposizione di utilizzatori e sviluppatori sono veramente considerevoli, ed andremo ad analizzarne i principali.

8.1 Il sito

Al giorno d'oggi, per qualsiasi azienda il biglietto da visita a livello mondiale è rappresentato dal proprio sito internet. A maggior ragione questo vale per le aziende informatiche, ed in modo ancor più accentuato per qualunque progetto abbia a che fare con un ambiente collaborativo.

Per MySQL, la cui comunità è stimato esser formata da circa 6 milioni di individui, il sito rappresenta il punto focale attorno al quale ruotano tutti gli aspetti della comunità . La mole di informazioni che contiene è abnorme, e di conseguenza orientarvisi attraverso può non essere semplice. Tuttavia non si tratta tanto di una deficienza strutturale, poichè ' il sito è ben fatto, suddiviso in categorie intuitive e costantemente aggiornato, quanto piuttosto di una conseguenza logica della completezza delle informazioni.

Dal sito si può accedere ai principali mezzi a disposizione della comunità , ed interagire appieno con il mondo di MySQL.

8.2 Mailing List e Newsletter

Uno dei primi strumenti di informazione collaborativa, la mailing list rappresenta uno strumento vitale per la comunicazione rapida all'interno di un progetto. MySQL ha all'attivo un ampio numero di mailing lists divise per linguaggio e soggetto. La lingua principale è l'inglese, ma non vengono risparmiati il brasiliano, il serbo o il giapponese.
Oltre a quella dedicata alla community ci sono liste riservate a MySQL Server, ai connettori di MySQL con i pišvariati linguaggi, al già citato Eventum e ai vari MySQL User Groups sparsi in giro per il mondo, ognuna delle quali ulteriormente suddivisa nelle sue aree di interesse.

Altresì è possibile iscriversi alla newsletter, in modo da tenersi aggiornati sulle principali evoluzioni del progetto.

8.3 Il forum

Sempre dal sito di MySQL è possibile accedere al forum, comprendente circa 100000 topic attivi e comprensivo di un servizio RSS per gestire gli aggiornamenti. Le tematiche del forum coprono svariate sezioni e lingue, più di quelle usate nelle mailing list (in queste ultime, ad esempio, non è presente l'italiano, compreso invece nel forum). Anche in questo caso, ai fini dell'ordine e della precisione, si ha una ben precisa e minuziosa divisione degli argomenti trattati. Inutile precisare, inoltre, che sebbene quello del sito sia il forum ufficiale, la rete pullula di forum o sezioni di essi dedicate a MySQL.

8.4 Documentazione

Un altro punto forte di MySQL è la documentazione. Dall'apposita sezione è possibile accedere ad una quantità esorbitante di informazioni riguardo ogni prodotto MySQL in svariate lingue e formati, dall'HTML al PDF. É interessante notare che tale documentazione non è distribuita secondo licenza GPL, nonostante rappresenti uno dei doni fatti dall'azienda alla comunità .

8.5 Articoli e Blog

Una sezione del sito piuttosto movimentata è anche quella che presenta gli articoli degli sviluppatori e degli utenti. Questa si suddivide nella sezione Articoli, comprendenti interviste e news di sviluppo fornite ufficialmente dal management MySQL, e il servizio Planet. Quest'ultimo è un interessante servizio che funge da aggregatore di post di blog di sviluppatori ed utenti MySQL, ma anche di impiegati e membri dei team ufficiali: chiunque può richiedere di essere aggiunto ai feed e partecipare così all'informazione riguardante MySQL, anche qui in svariate lingue.

8.6 Meetups e Gilde

Come ogni comunità che si rispetti, anche quella degli sviluppatori e degli utenti MySQL talvolta esce dalle polverose cantine, sfrega gli occhi contro il sole e si raduna nei MySQL Meetup dislocati in tutto il mondo. In questo caso, il sito si appoggia ad un servizio, *meetup.com*, abbinato a Google Maps per trovare l'incontro fra utenti più vicino a casa propria. Può essere significativo notare che in Italia non viene organizzato alcun meeting, cosa che accade invece regolarmente anche in posti più isolati, come le isole Hawaii.

Il sito di MySQL fornisce inoltre una pagina aggiornata annualmente dedicata alle cosiddette *gilde*, ovvero insiemi di persone, divise in campi di interesse, che si

sono distinte per l'impegno verso il progetto, da conferenzieri a giornalisti, esperti o sviluppatori, che vengono fra l'altro indirettamente suggerite come ospiti per l'appunto nei MeetUp.

8.7 IRC

MySQL dispone anche di due discretamente frequentati canali IRC: *#mysql*, utilizzato per comunicazioni veloci riguardanti aspetti variegati del mondo MySQL, dall'installazione a tematiche su linguaggi ed interfacce, e *#mysql-dev*, destinato agli sviluppatori.

Capitolo 9

Strumenti di sviluppo

Ovviamente, la comunità di MySQL utilizza gli strumenti elencati nel capitolo 8 sia a scopo di utilizzo del software che di aggiornamento e sviluppo, ma esistono alcuni altri tool destinati prettamente al secondo scopo, per chi vuole sviluppare tools per o con MySQL.

9.1 MySQL DevZone

La zona sviluppatori è un'area del sito principalmente dedicata allo sviluppo *con* MySQL, tramite i più noti linguaggi di programmazione. Essa presenta guide, riferimenti ed esempi molto completi per lavorare in PHP, C#, Python, e via dicendo.

9.2 La Forgia

Per chi invece vuole entrare a far parte del progetto MySQL, esiste MySQL Forge: questo è a tutti gli effetti un repository di strumenti, roadmaps e wiki creato da e per la comunità degli sviluppatori MySQL. Al suo interno [18] sono disponibili le informazioni utili a chi voglia contribuire attivamente al progetto, prima fra tutte la guideline interna. In questa vengono spiegati esaustivamente i dettagli di basso livello del funzionamento di MySQL, dalle linee guida della redattura del codice, alla struttura delle classi e del sistema transazionale, le modalità con cui si supportano i plug in, fino ad arrivare al significato dei vari messaggi d'errore o alle modalità di testing.

9.3 Bazaar

Bazaar [4] è il revision control system attualmente utilizzato da MySQL, dopo anni di utilizzo di *BitKeeper*. Si tratta di un vcs distribuito ed open source supportato da Canonical (fra le altre cose supporter del crescente sistema Ubuntu). Bazaar si appoggia a sua volta a Launchpad, un servizio web sempre creato e gestito da Canonical per l'hosting di progetti, comprendente svariati tools utili agli sviluppatori. Fra le caratteristiche principali, la visione online del codice,

un sistema di bug reporting, di gestione ed acquisizione di specifiche (*blueprints*) e conoscenze di base dei progetti, nonchè un sito dedicato alla localizzazione linguistica dei progetti.

Anche in questo caso, la Forgia contiene numerosi manuali e riferimenti per comprendere il funzionamento di MySQL tramite Bazaar, e poter così entrare a far parte attivamente della comunità .

9.4 Bug Reporting

Nonostante il supporto al bug reporting fornito da Launchpad, MySQL dispone di un proprio sistema atto allo scopo, basato su quello usato dal progetto PHP. Le statistiche visualizzabili dallo strumento, d'altronde, dimostrano quanto sia meglio avere il controllo - ed assumersi il carico - sul sistema di bug reporting: a Giugno 2009 il numero di bug segnalati, comprendenti quelli risolti e pendenti, supera le 38000 unità .

9.5 Revisione del Codice

Il processo di revisione del codice spedito dalla comunità di sviluppo è pressoché il medesimo, sia che si tratti di una patch che di un bug fixing: la differenza sostanziale sta nel numero di altri sviluppatori che lo revisioneranno, maggiore nel primo caso.

Le linee guida per l'accettazione di una patch sono numerose e variegate [17], ma comunque tutte legate ovviamente ad ottenere la massima pulizia ed efficienza dal codice prodotto. Fra le più importanti rientrano quelle sulla limitata estensione delle modifiche apportate da una patch, l'aderenza alle linee guida sulla scrittura del codice e alle precedenti specifiche di protocollo, la sicurezza ed i casi di testing, che devono essere annessi o comunque sviluppati, anche con l'aiuto di personale di MySQL.

Importante ricordare che l'accettazione e la pubblicazione delle patch sono soggette alle limitazioni già discusse nella sezione [**?**].

Bibliografia

[1] Kaj Arnö. Quid pro quo. URL http://blogs.mysql.com/kaj/2005/09/27/mysqls-quid-pro-quo/.

[2] Matthew Aslett and Marten Mickos. Mixed bizness - how to make money from open source. URL http://www.businessreviewonline.com/os/archives/2006/11/mixed_bizness_-.html

[3] Charles Babcock. Sun locks up mysql, looks to future web development. URL http://www.informationweek.com/news/software/open_source/showArticle.jhtml?a

[4] Canonical. Bazaar-vcs. URL http://bazaar-vcs.org/.

[5] Thomas Claburn. Google releases improved mysql code. URL http://www.informationweek.com/news/internet/showArticle.jhtml?articleID=199

[6] John Dvorak. The sun-mysql deal stinks. URL http://www.marketwatch.com/story/the-sun-mysql-deal-stinks?dist=hplatest.

[7] Google et al. Google - mysql tools. URL http://code.google.com/p/google-mysql-tools/.

[8] Janet Hendrickson and Daly. Mysql gains 25survey shows. URL http://www.evansdata.com/press/viewRelease.php?pressID=5.

[9] Eugenia Loli-Queru. Darpa pulls openbsd founds. URL http://www.osnews.com/story/3307.

[10] Daniel Lyons. Cash me out. URL http://www.forbes.com/forbes/2008/0225/060.html.

[11] Sun Microsystems. Oracle to buy sun, . URL http://www.sun.com/aboutsun/pr/2009-04/sunflash.20090420.1.xml.

[12] Sun Microsystems. Sun microsystems announces agreement to acquire mysql, developer of the world's most popular open source database, . URL http://www.sun.com/aboutsun/pr/2008-01/sunflash.20080116.1.xml.

[13] Sun Microsystems. Sun microsystems, inc. contributor agreement, . URL http://forge.mysql.com/contribute/cla.php.

[14] M.M.Widenius. To be (free) or not to be (free), . URL http://monty-says.blogspot.com/2009/04/to-be-free-or-not-to-be-free.html.

[15] M.M.Widenius. Monty program ab, . URL http://askmonty.org/wiki/index.php/Main_Page.

[16] M.M.Widenius. Open database alliance, . URL http://opendatabasealliance.com/.

[17] MySQL. Code review process, . URL http://forge.mysql.com/wiki/Code_Review_Process.

[18] MySQL. Contributing code, . URL http://forge.mysql.com/wiki/Contributing_Code.

[19] MySQL. Mysql development cycle, . URL http://forge.mysql.com/wiki/Development_Cycle.

[20] Tim O'Reilly. Take the money and run? i think not, . URL http://radar.oreilly.com/archives/2008/02/forbes-on-mysql-sun-open-source.html.

[21] Tim O'Reilly. Open source paradigm shift, . URL http://www.oreillynet.com/pub/a/oreilly/tim/articles/paradigmshift_0504.html.

[22] Xaprb. Does mysql really have an open-source business model? URL http://www.xaprb.com/blog/2008/12/23/does-mysql-really-have-an-open-source-busine

[23] Noel Yuhanna, Mike Gilpin, and Catherine Salzinger. Market update: Open source databases. URL http://www.forrester.com/Research/Document/0,7211,46061,00.html.

Elenco delle figure

3.1 Statistica sugli utilizzi dei principali DBMS 5

7.1 Milestones per le nuove versioni di MySQL server 20
7.2 Comparazione fra vecchio e nuovo Development Cycle 21