Alex Petrov

Database Internals: A Deep-Dive Into How Distributed Data Systems Work

Sebastopol, O'Reilly, 2019, 200 p.

L'obiettivo di questo manuale è fornire gli strumenti necessari per scegliere tra varie tecnologie e metodi di sviluppo e gestione di banche dati distribuite e accessibili in rete e far parte produttivamente e responsabilmente dei team che se ne occupano in tutte le fasi progettuali.

Il libro si rivolge pertanto non solo a ingegneri del software ma anche a quanti, con vari ruoli e gradi di *expertise* tecnica, lavorano alla creazione, gestione, accesso e distribuzione di collezioni di dati.

L'importanza di comprendere professionalmente il trattamento automatico dell'informazione è da molto tempo alla base dell'offerta di numerosi programmi di formazione universitaria, inclusi gli insegnamenti di informatica documentale, gestione automatica di archivi e biblioteche e biblioteche digitali nell'ordinamento universitario italiano, indirizzo archivistico e librario dai primi anni Ottanta. Nell'ultimo decennio, le applicazioni di intelligenza artificiale e *machine learning* sono entrate nella sfera di interessi dei corsi di laurea in scienze dell'informazione un po' ovunque, soprattutto nei paesi anglosassoni e in Cina (tra le scuole più famose che hanno intrapreso questa direzione citiamo la School of Information, UC Berkeley e la School of Information and Communication at Rudtgers, The State University of New Jersey). Pertanto l'argomento non potrebbe essere più attuale e il libro in un certo senso davvero necessario.

Tuttavia, l'adozione di *Database Internals* come libro di testo o manuale per studenti sia di scienza dei dati che di scienze dell'informazione sarebbe forse una scelta parziale, in quanto il libro manca, ed è questo al contempo il suo pregio maggiore e il suo solo limite, di qualsivoglia riferimento a questioni di interpretazione, epistemologia, *policy* e politiche culturali, scelte editoriali e apparati critici, tassonomie, creazione artistica o finanziamenti privati, accesso, conservazione o cen-

sura di collezioni di dati e database e via dicendo.
L'esperienza pratica dimostra che la progettazione e implementazione di database sono sempre influenzate da contesti organizzativi, economici e socio-culturali, che si riflettono sul livello delle operazioni sui dati, dal modo di scrivere e manipolare il codice agli aggiornamenti. Ma la prospettiva dell'autore – che è di fatto quella diffusa nell'intera industria informatica e delle telecomunicazioni, dalle società di ricerca del personale alla stampa specializzata – rimane circoscritta a questioni di esclusiva performance e risoluzione di problemi tecnici anche se ha la dichiarata ambizione di trattare aspetti che determinano il funzionamento delle infrastrutture di cui i database fanno parte.
Una volta compresa e accettata la fondamentale asetticità di questa dimensione tecnica e la sua conseguente visione parziale di che cosa costituisca l'infrastruttura delle collezioni di dati, il lettore motivato e interessato, anche privo di familiarità con la materia, troverà il testo molto utile.
Convenzioni tipografiche ed esempi pratici favoriscono vari livelli di lettura studio o consultazione. Riferimenti bibliografici per approfondimenti mirati si trovano alla fine di ogni capitolo e puntano a tecniche e algoritmi particolari.
L'autore riesce a rendere chiari concetti astratti e complessi, spesso offuscati dal gergo informatico, in un campo di conoscenze teoriche e applicazioni pratiche tumultuoso, documentato (e mai davvero del tutto) tramite conferenze, comunità di pratiche e reti di relazioni informali tra professionisti e cultori della materia, più che attraverso corsi di formazione, insegnamenti universitari o libri di testo.
Le due parti del volume sono complementari ma autonome e trattano delle due caratteristiche fondamentali dei sistemi di gestione di database: la prima riguarda lo *storage* o *database engine*, cioè il modo in cui i sistemi software memorizzano i dati; la seconda parte concerne il modo in cui i database distribuiscono e scambiano dati, usando macchine e algoritmi fisicamente distinti ma logicamente parte dello stesso sistema.
Chi ha familiarità con l'argomento (soprattutto dal punto di vista delle tecnologie di *retrieval*, *indexing*, interfacce utente e altre caratteristiche che determinano aspetti semantici e commerciali come l'inserimento della pubblicità nei contenuti dei database), noterà certamente la clamorosa assenza di un terzo aspetto dei sistemi di database, cioè i linguaggi di gestione delle interrogazioni (*query languages*). Questa scelta si deve al fatto che ogni sistema ha il suo *query handler* proprietario e non esiste un linguaggio comune a tutti i sistemi considerati: quindi tanto vale concentrarsi, spiega l'autore, su ciò che si può davvero considerare conoscenza condivisa e di pubblico dominio su come i dati vengono immagazzinati e distribuiti, sia pure da sistemi differenti e spesso in concorrenza tra loro.
Nell'introdurre la prima parte, l'autore ammette che sarebbe comodo se chiunque potesse usare database come *black boxes*, senza bisogno di dare un'occhiata al codice (in fondo non è così che usiamo moltissimi prodotti elettronici o elettrodomestici?), ma la pratica dimostra che prima o poi qualcosa di problematico, un *bug software*, un sovraccarico di richieste di accesso ai dati o un semplice sbalzo di corrente, fa sì che sia necessario dare una occhiata alle caratteristiche interne: sapere dove e come intervenire in caso di black out, perdita di dati e altri disastri riduce rischi e aumenta le probabilità di un veloce ripristino delle operazioni. Perciò, uno degli strumenti più importanti è il *benchmarking* necessario per comparare gli aspetti che determinano le performance dei vari software disponibili sul mercato, in modo da dimensionare il proprio sistema nel modo più consono alla singola realtà operativa, alle ambizioni progettuali e alle prospettive di crescita.
A questo fine, tornano utili i concetti che spiegano, nei primi tre capitoli del volume, l'architettura e i componenti fondamentali di un database system, come *in memory* e *disk based storage*, gli aspetti di durabilità, volatilità, sicurezza dei dati, i metodi per memorizzare dati per righe o per colonne che rispondono a requisiti differenti di velocità di accesso ma anche di complessità e profondità di analisi, gli indici, *clustered e non clustered*, primari e secondari, la struttura e le tecniche di manipolazione (*buffering*, mutabilità/immutabilità e ordinamento). Queste nozioni, sia nel settore della ricerca che nelle applicazioni industriali e commerciali, richiedono di essere comprese anche per poter valutare e indirizzare gli sviluppi più recenti (con le strutture dei dati probabilistiche e gli algoritmi per il *machine learning*).
I successivi sei capitoli della prima parte immergono il lettore nelle distinzioni relative a *binary search tre-*

es and *B-trees (*termine ombrello per una famiglia di strutture di dati), tecniche di organizzazione di dati in forma binaria (*file system design*), le cosiddette proprietà ACID (*atomicity*, *consistency*, *isolation*, *durability*) che permettono di comprendere le transazioni di lettura, scrittura e recovery dei dati, i meccanismi di *buffing/caching*, e gli algoritmi di *auditing* e controllo, argomento introdotto dalla citazione di un breve motto di Pat Helland: "I contabili non usano gomme per cancellare, altrimenti finiscono in galera".

Petrov conclude la prima parte incoraggiando il lettore a considerare che il progresso in materia di database è incrementale. Parole appropriate per introdurre la seconda parte, sui *Distributed systems*, in cui l'opera non fa economia di suggerimenti per convincerci, se mai ve ne fosse bisogno, che il presente e il futuro di questi sistemi dipende dagli algoritmi distribuiti: questi assicurano l'interconnessione di molte macchine in rete che lavorano come una singola logica risorsa. Il modello – chiamato *horizontal scaling* – è adottato ormai in pratica in ogni settore industriale e il lettore non farà fatica a considerare quanti esempi ricorrano nell'esperienza quotidiana di chiunque: dagli OPAC delle biblioteche che lo hanno sperimentato sin dagli anni Ottanta ai sistemi di posta elettronica, dalle applicazioni di telemedicina alle riunioni virtuali via Zoom, qualsiasi database che funzioni come una applicazione client/server è infatti un sistema distribuito. Eppure del funzionamento dell'*horizontal scaling* si parla e si sa davvero poco e in sostanza l'intero argomento rimane una faccenda per pochi iniziati, spesso poco documentato in termini di requisiti tecnici, di valutazioni o *risk assessment*.

La principale avvertenza che molto onestamente il libro fornisce introducendo questa giovane disciplina è che il rischio di fallimenti è elevatissimo e il *know how* a disposizione ancora piuttosto immaturo a dispetto della straordinaria domanda commerciale: un sistema distribuito, ci dice Petrov citando Leslie Lamport (noto autore di algoritmi per il *distributed computing*), è quello in cui il malfunzionamento di un computer dall'altra parte del globo e di cui ignoravi completamente l'esistenza può all'improvviso rendere il tuo sistema totalmente inutile.

Ciò nonostante, l'*horizontal scaling* è ormai stato adottato come modello di crescita dal settore informatico e delle telecomunicazioni a livello mondiale: non c'è nuova azienda, prodotto, processo, servizio o tecnologia alcuna che si possa considerare ormai sostenibile senza ricorso alla tecnologia cloud e a qualche applicazione di database distribuiti. Ma nulla nell'ambito dei sistemi distribuiti è completamente affidabile o immune dal rischio di errori a catena.

Gli algoritmi esistenti sono non solo sempre molto più soggettivi o parziali di quanto ci si possa immaginare, ma anche costantemente rivisti, aggiornati e sostituiti da versioni più recenti, dove la migliore resa commerciale di una certa creazione intellettuale compete con la necessità di allineamento tecnico. Si parla così di coordinamento, cooperazione, disseminazione, consenso come se questi fossero processi agili, flessibili, sempre opportuni e controllati e non già esito di molteplici operazioni (con nulle o minime interazioni o controlli umani) di scambio di dati concatenati, che non sempre riescono dove, come dice la famosa legge di Murphy, se qualcosa può andare male di sicuro lo farà, senza possibilità di ricorso al "buon senso comune".

Se tutto va bene, gli algoritmi concorrono per il funzionamento dello stesso sistema logico sincronizzando le operazioni gestite da macchine differenti. Ma molto spesso switches, cavi, settaggi di routers e fusi orari differenti determinano disconnessioni inaspettate, malfunzionamenti a catena, valanghe di errori che si traducono nella indisponibilità delle risorse chiave che si erano date per scontate.

Dunque la prima missione di ogni sistema distribuito consiste, banalmente, nella condivisione di informazioni affidabili e in tempo reale sullo stato dei singoli nodi (macchine o processi) del sistema.

A questa, un poco drammatica ma realistica, panoramica introduttiva sull'interoperabilità per le architetture cloud e i database distribuiti, concentrata su terminologia e concetti di base (link, processi e partizioni), seguono sette dettagliati capitoli che spiegano i rimedi che si possono adottare per il continuo monitoraggio, la prevenzione e l'individuazione rapida dei più comuni malfunzionamenti ed errori.

La prima e più importante soluzione alle problematiche e agli errori dei sistemi distribuiti consiste nella disponibilità di una gamma ormai piuttosto articolata di algoritmi di pubblico dominio tra cui scegliere per la identificazione di anomalie e malfunzionamenti, ognuno con vantaggi e svantaggi in termini di semplicità, accuratezza o precisione.

Il libro non menziona specificatamente sistemi di categorizzazione automatica e *machine learning* che recentemente sono divenuti di grande successo commerciale per la gestione automatica dei "tickets" o richieste di assistenza e intervento, ma l'autore spiega pazientemente le convenzioni standard usate per notificare lo stato di sistemi e ricorda che non esiste una semantica veramente affidabile a livello di protocolli di rete (malgrado gli errori del calcolo distribuito siano un'area di studi e sperimentazioni internazionali dove molto è noto agli addetti ai lavori sin dall'inizio degli anni Novanta).

Un capitolo è quindi interamente dedicato agli algoritmi di *leadership* e *leader election* che rivestono una particolare importanza: la maggior parte dei database distribuiti in rete necessitano di un punto di riferimento centrale (leader) per trattare dati in modo consistente, lasciando molte altre decisioni alle versioni locali (l'esempio più banale che viene in mente a proposito è l'architettura delle piattaforme dei social media).

Altrettanto articolata è la presentazione dei modelli (lineari, sequenziali o causali) e altri concetti relativi alla consistenza dei dati: queste tecniche nel loro insieme permettono a un sistema di continuare a funzionare e gestire o aumentare la disponibilità di accesso ai dati anche quando qualche processo fallisce (o diviene "frozen") tramite la sincronizzazione di copie multiple. Il coordinamento supplementare richiesto in modo da garantire il livello di consistenza desiderato suggerisce di scegliere una o l'altra tecnica a seconda dei casi, e ancora una volta sembra che la qualità e affidabilità delle applicazioni sia frutto di esperienza, arte e creatività più che una scienza esatta.

Uno dei dilemmi dei database distribuiti più dibattuti da ingegneri programmatori e sviluppatori in rete è il *trade off* tra consistenza e disponibilità, soprattutto con riferimento a *big data*, ed è oggetto di un altro capitolo, a cui segue la trattazione delle diverse modalità e degli algoritmi, primi fra tutti *consensus algorithms* e *atomic commit algorithms*, utili per implementare transazioni distribuite.

La tipologia più nota di algoritmi Consensus (Paxos) è materia di un capitolo dedicato che esamina le differenze tra le più diffuse varianti.

Il testo di questa seconda parte del volume è più impegnativo per chi non sia in qualche modo un addetto ai lavori ma abbonda di metafore che aiutano a comprendere nozioni complesse, riducendo sofisticate operazioni matematiche a semplici esercizi logici come nel caso del noto problema dei due generali che devono coordinare un attacco militare in modo asincrono, senza mai incontrarsi e avere certezza reciproca di ricezione delle comunicazioni.

Nella breve conclusione l'autore raccomanda di guardare ai due principali aspetti di ogni *database system*, performance e scalabilità, dalla prospettiva del design dei vari sottosistemi e delle loro interazioni, in modo il più possibile olistico, una concessione in direzione di altri approcci interdisciplinari al disegno e alla implementazione dei database certamente opportuna.

Infatti, per chi guarda all'evoluzione della tecnologia da un punto di vista umanistico o socio-tecnico la lettura, consultazione o studio di questa opera produce un certo sgomento in quanto ci si rende conto che il marketing delle soluzioni per i database e per i *big data* in generale continua ad alimentare aspettative di completezza, di certezza e di risoluzione dei problemi in modi che in realtà non raggiungono mai del tutto l'obiettivo di totale affidabilità e scalabilità e rimangono esposti al rischio di compromessi, variazioni, divergenze, adattamenti e sorprese inaspettate, tanto di malfunzionamenti quanto di efficienza operativa.

La creazione e distribuzione in rete di database è un mondo in costante tumulto: a volte il contributo di un informatico o ingegnere del software geniale fa fare al settore giganteschi passi in avanti, con nuove idee per l'architettura dei dati o per la applicazione condivisa di un algoritmo, come il caso dell'algoritmo di Paxos di Leslie Lamport. Tuttavia, nel complesso, la scienza dei dati, a ben guardare come questi sono immagazzinati, trattati e distribuiti, sembra ancora ancorata a concetti dell'origine del calcolo automatico, lontana dal potersi considerare un mondo di conoscenze standard o scientifiche o semplicemente robuste e mature al punto da garantire investimenti che richiedono non già decadi ma anche solo pochi anni per ammortamenti.

Al contrario, il libro documenta come la scienza dei dati sia ancora fondamentalmente un mondo di prodotti e processi artigianali, resi più affidabili o più efficienti grazie alla creatività intellettuale e alla automazione dei controlli, alle trovate geniali così come all'opportunismo pratico.

La bibliografia finale, con oltre 200

riferimenti recenti, soprattutto a relazioni e presentazioni a convegni tenute da esperti, e il generoso indice analitico risulteranno utili tanto al ricercatore quanto al professionista che vogliano rivedere scelte tecniche, pratiche gestionali e organizzative o disegnare nuove ambiziose architetture cloud ma sempre... con i piedi per terra.
L'opera è disponibile anche tramite il sistema di e-learning dell'editore e ha un sito web interessante nel quale l'autore raccoglie commenti, apprezzamenti, feedback dei lettori.

BRUNELLA LONGO
Online Data Assessment
icm2re@gmail.com

DOI: 10.3302/0392-8586-202108-060-1