

IT-STUDIES

RELATIONELE DATAMANAGEMENTSYSTEMEN ZIJN AL DECENNIALANG HET STANDAARDGEREEDSCHAP OM DATA TE BEHEREN. MAAR DATAWETENSCHAPPERS NEGEREN DEZE SYSTEMEN GROTENDEELS, OMDAT HET TRAAG EN OMSLACHTIG IS OM ZE TE COMBINEREN MET DE TOOLS DIE ZIJ ZELF GEBRUIKEN. MARK RAASVELDT WIL DIE KLOOF OVERBRUGGEN. HIJ STELT NIEUWE TECHNIEKEN VOOR DIE DEZE INTEGRATIE GEBRUIKSVRIENDELIJKER EN EFFICIËNTER MAKEN.

door Mark Raasveldt beeld Shutterstock

Zo krijg je datawetenschappers aan de relationele databases

DATABASE MAKKELIJK EN SNEL KOPPELEN AAN ANALYTISCHE TOOLS

Van kleine webshops tot techgiganten: allemaal gebruiken ze relationele databasesystemen als belangrijkste methoden voor dataopslag en databeheer. Sinds de uitvinding van dit soort systemen, in 1970, zijn ze uitgegroeid tot een wereldwijd gebruikte tool. En dat is niet voor niets: al decennialang weten database-onderzoekers enorme vooruitgang te boeken met het verbeteren van deze systemen. Zo zijn database-engines uitgegroeid tot bijzonder krachtige tools die efficiënte analytische zoekverwerking mogelijk maken.

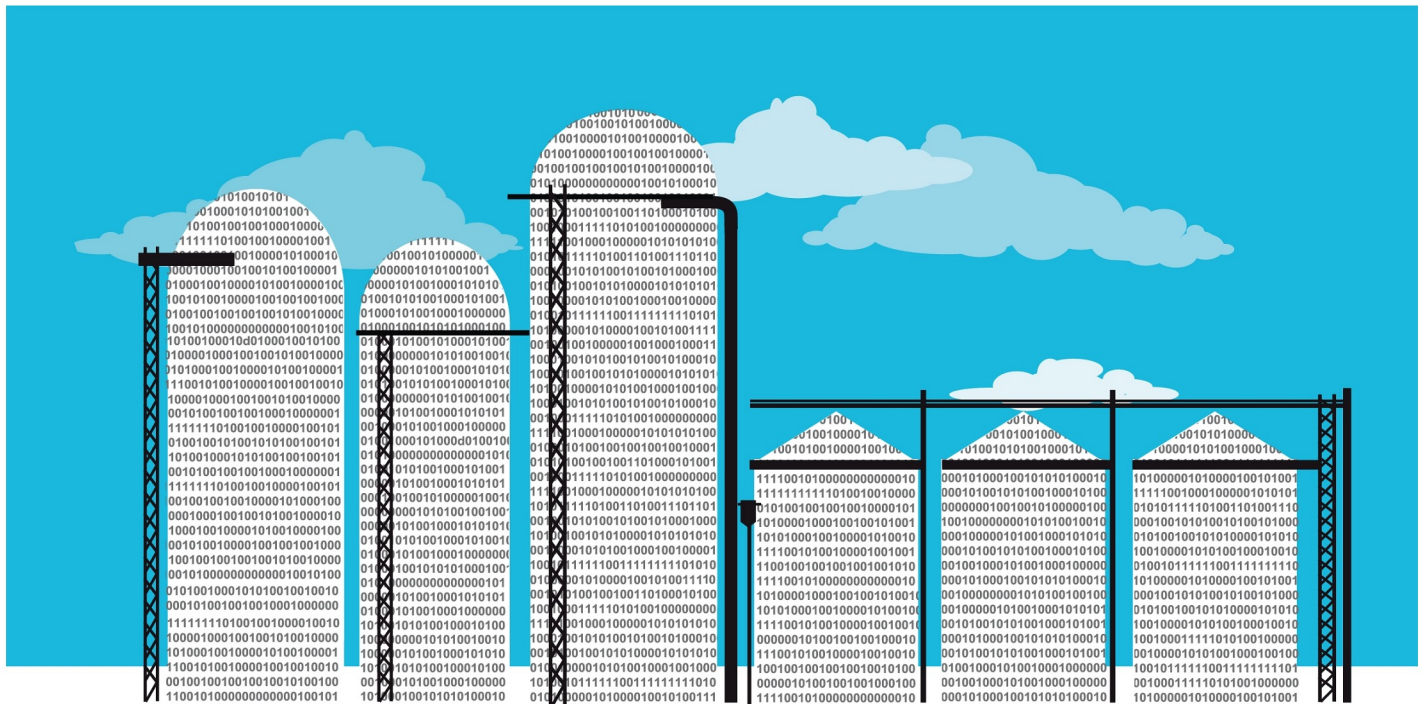
Maar hoe krachtig ze ook zijn, datawetenschappers hebben ze nooit omarmd. De oorzaak daarvan ligt in de moeilijke koppeling van deze tools met databases. De huidige methoden om databases met analytische tools te koppelen, zijn traag en omslachtig, waardoor het gebruik van een databasesysteem meer hindert dan helpt. Hadley Wickham, een invloedrij-

ke ontwikkelaar rondom de programmeertaal R, dacht er ook zo over. Veelzeggend is de mening die hij in 2018 gaf in een artikel over het database-pakket dplyr: 'If your data fits in memory, there is no advantage to putting it in a database: it will only be slower and more frustrating.'

EEN WAAIER AAN ALTERNATIEVEN

Exit datamanagementsystemen dus? Nee, zo eenvoudig ligt het niet. Ook al gaat die integratie en koppeling moeilijk, datawetenschappers hebben wel degelijk behoefte aan manieren om data te beheren en analyseren. In plaats van bestaande databasesystemen te gebrui-

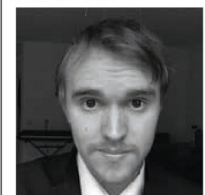
DUCKDB DRAAIT DIRECT IN DE CLIENT-APPLICATIE



ken, ontwikkelen datawetenschappers eigen oplossingen. Het resultaat is een enorme waaier aan alternatieven voor databaseer. Deze systemen voeren vergelijkbare taken uit als de klassieke databasemanagementsystemen, maar lopen tegen veel problemen aan die decennia geleden al door database-onderzoekers zijn opgelost. Een voorbeeld hiervan is de populaire Pandas-extensie voor Python. Pandas verricht klassieke databaseoperaties, zoals aggregaties of joins. Het is makkelijk te installeren en te gebruiken, met volledige integratie binnen Python. Hierdoor is het heel populair onder datawetenschappers en staat het centraal in de datasciencewereld. Voor R zijn de meest prominente alternatieven voor databasystemen R DataFrames en dplyr. Net als Pandas worden deze gebruikt voor databaseoperaties en als in-memory opslagmethode. Deze extensies lijken op het eerste gezicht erg op een databasemanagementsysteem, maar er ontbreken essentiële onderdelen, waardoor de gebruiker onverwacht in de problemen kan raken. Een essentieel verschil is dat deze extensies de data alleen in het werkgeheugen opslaan. Als de gebruiker de data wil

bewaren, moet hij dat zelf regelen. Dat gebeurt vaak door de data op te slaan in meerdere flat files – bijvoorbeeld CSV, HDF5 of Parquet – die de gebruiker zelf moet beheren. Naarmate datasets groter worden, kan dit erg veel werk worden en bovendien is het erg foutgevoelig. Een ander groot nadeel hiervan is dat de gehele dataset in het werkgeheugen van de computer moet passen. Met grote datasets is dit vaak niet het geval. Gebruikers moeten dan bijvoorbeeld zelf de data in verschillende segmenten opdelen (partitioning) en vervolgens de data zelf in en uit het geheugen laden. Een databasemanagementsysteem doet dit volledig automatisch en vaak efficiënter dan de gebruiker het zelf zou doen. Verder ontbreekt er in deze extensies een query optimizer. Het is een cruciaal component van een databasemanagementsysteem dat naar het gehele queryplan kijkt om – ongeacht hoe de gebruiker zijn vraag heeft geformuleerd – de meest efficiënte manier van aanpak te vinden. In deze extensies wordt het queryplan exact uitgevoerd zoals de gebruiker het heeft geformuleerd. Daardoor is de gebruiker zelf verantwoordelijk om het optimale queryplan te vinden.

AUTEUR



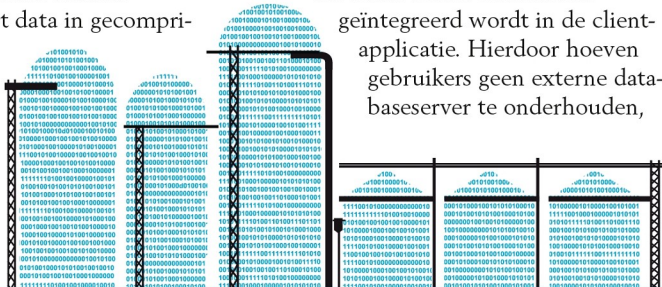
MARK RAASVELDT is verbonden aan het Centrum Wiskunde & Informatica, waar hij werkt aan de ontwikkeling van DuckDB. Eerder dit jaar promoveerde hij aan de Universiteit Leiden op zijn proefschrift *Integrating Analytics with Relational Databases*. In 2018 won Raasveldt de SIGMOD student research competition voor zijn werk aan embedded databases.

IT-STUDIES

KILOBYTES AAN DATA

Het moet mogelijk zijn om deze kloof te overbruggen en databasemanagementsystemen gebruiksvriendelijker en efficiënter te maken voor datawetenschappers. Mijn onderzoek richt zich in de eerste plaats op het verbeteren van de standaardmethode voor database-clientintegratie: client-serververbindingen. Hier draait de database als een externe applicatie (de 'server'), en verbindt de client zich met deze server door middel van een netwerkverbinding. Data worden dan over deze netwerkverbinding heen en weer gestuurd tussen de database en het externe programma. De moeilijkheid schuilt in de protocollen die hiervoor worden gebruikt: de standaard databaseclient-serverprotocollen zijn decennia geleden ontworpen om kilobytes aan data te versturen. Deze protocollen zijn totaal ongeschikt voor de grote hoeveelheden data die vandaag de dag worden gegenereerd. Een nieuw client-serverprotocol dat tijdens mijn onderzoek werd ontwikkeld, kan de overdracht van grote hoeveelheden data vele malen sneller maken.

Dit protocol stuurt data in gecomprimeerde vectors, waardoor meer data zich laten verpakken in kleinere pakketten. Daarnaast



GRATIS EN OPEN SOURCE

Op dit moment is DuckDB beschikbaar als pre-release software, gratis en open source. Het wordt ontwikkeld door de onderzoeksgroep Database Architectures van het Centrum Wiskunde & Informatica. Om DuckDB verder te ontwikkelen, werkt de groep nu aan gecomprimeerde opslag, data-encryptie en verdere snelheidsverbeteringen. Daarbij werkt de groep samen met datawetenschappers – het onderzoeksveld dat DuckDB uiteindelijk kan gebruiken als degelijke tool in plaats van hun eigen databaseheerfabricaten. Nu al kunnen datawetenschappers de pre-release versie gebruiken; een definitieve ziet naar verwachting over een paar jaar het licht. En dan staat niets een uiteindelijke omarming in de weg. Meer informatie over DuckDB is te vinden op www.duckdb.org.

Nieuw client-serverprotocol stuurt data in gecomprimeerde vectors

kun je deze pakketten sneller in- en uitpakken.

Zelfs met dit protocol kost het nog steeds erg veel tijd om data over een netwerkverbinding te sturen. Daarbij kost het opstarten en het onderhouden van een aparte databaseserver veel werk, en dit is niet altijd noodzakelijk.

Het uitgangspunt van client-server-architectuur is namelijk dat meerdere mensen met dezelfde data kunnen samenwerken. Maar datawetenschappers werken vaak alleen en hebben die extra complexiteit dus niet nodig.

DUCKDB

Speciaal voor datawetenschappers in deze situatie is een nieuw systeem ontworpen, genaamd DuckDB.

Het systeem is uniek omdat het niet als client-server draait maar

geïntegreerd wordt in de client-applicatie. Hierdoor hoeven gebruikers geen externe databaseserver te onderhouden,

wat het gebruik van het systeem immens versimpelt. Omdat DuckDB direct in de client-applicatie draait, hoeven data niet over een netwerkverbinding te worden verstuurd. Hierdoor laten data zich gemakkelijk en goedkoop heen en weer schuiven tussen de databaseserver en de client – en dat maakt het systeem veel efficiënter in gebruik.

Deze integratie maakt het ook mogelijk voor DuckDB om kosteloos data te lezen en naar client-native data structures te schrijven. Zo kan DuckDB bijvoorbeeld direct een Pandas DataFrame lezen en hier een SQL-query over draaien, en het resultaat weer meteen in een Pandas DataFrame opslaan. Dit versimpelt het gebruik van het systeem in combinatie met bijvoorbeeld machine learning of classification libraries die van deze datastructuren gebruikmaken.

DuckDB biedt verder wel alle voordelen van een databasesysteem en heeft volledige SQL-ondersteuning. Het systeem beschikt over uitvoerige optimalisaties, parallelisme en ondersteuning voor datasets die groter zijn dan het werkgeheugen. Verder ondersteunt het volledig de zogeheten ACID-eigenschappen. Dit voorkomt datacorruptie en zorgt ervoor dat data nooit verloren gaan.

DuckDB biedt naast SQL ook een aparte interface – de relational API – die het gebruik van het systeem gemakkelijk maakt. Met deze interface kunnen queries stapsgewijs worden opgebouwd, kunnen tussenresultaten makkelijk worden bekeken en kunnen queries makkelijk worden gecombineerd zonder gebruik te maken van string-concatenatie. Dit voorkomt meteen ook een bekende exploit genaamd SQL-injectie, een soort cyberaanval waarbij een hacker een SQL-codefragment gebruikt om een database te manipuleren.