]

Performantievergelijking van database-systemen : Mariadb en SQLServer

Casus onderzoeksproces

Jonas Moens[1], Levi Goessens[2], Mauritz Cooreman[3],

Samenvatting
Ons onderzoek is een van de vele onderzoeken voor het meten van de performantie tussen verschillende DataBase Management Systemen of kort DBMS. Ons doel hiervan is het informeren van organisaties en instanties voor het kiezen van een gepast systeem zodat die ook het meest efficiënt kunnen gebruikt worden.Veel van deze onderzoeken zijn toch niet verschenen, nochtans zijn ze broodnodig.En van de onderzoeken die al reeds gepubliceerd werden hebben sommigen ronduit foute methoden gebruikt of bezitten ze een luchtje van nonchalance. Ons onderzoek werd door beperkte middellen en faciliteiten gedwongen om enkel de uitvoeringssnelheid van verschillende soorten query's te meten in een dataset van ongeveer 1000000 rijen. Deze werden onderworpen aan een Mariadb database systeem dat plaatsvond op een virtuele machine en een SQLserver database systeem dat geplaatst werd op een Windows hostsysteem.In dit document vind u een literatuurstudie als ook onze experimentele data met hypothese en uitkomsten gepaard met enkele staafdiagrammen en boxplotten. Verder werd dit onderzoek statistich getoetst met een z-score en kunden we een korte conclusie naar voren schuiven.Sommige queries konden niet gebruikt worden of waren redundant. Terwijl we bij de simpelste queries een verschil in snelheid ten voordele zagen voor het mariadb systeem, zagen we echter dat de trend voor het grootste deel toch ten voordeel van het SQLserver bleek te zijn.Dus onze conclusie is dat sql server toch voor het grootse deel de beste performantie blijkt te bezitten. Dit moet natuurlijk met een korrel zout genomen worden sinds dat onze infrastructuur niet optimaal was en dat we toch een omgeving opzetten die nadelig zou kunnen zijn voor Mariadb. Verder onderzoek kan leiden tot wat nu echt de oorzaak is van die toch iets mindere performantie van mariadb. Daarom raden we ten strengste ook aan om ons onderzoek verder te zetten of zelfs te weerleggen voor ieder die zich geroepen voelt!

Sleutelwoorden
Database-beheer. Relationele databases — Performantie

Contact: [1] Jonas.Moens@student.hogent.be; [2] Levi.Goessens@student.hogent.be; [3] Mauritz.Cooreman@student.hogent.be; [4] voornaam.naam@student.hogent.be

Inhoudsopgave

1. Inleiding

Elk bedrijf heeft nood aan een DBMS (=DataBase Management System), maar de markt hiervoor is zo uitgebreid dat zelfs experts niet weten welke nu de beste is, als die er al is. Na vele onderzoeken en studies zoals door **Bassil2012** en **Cloudhary2014** blijkt eerder dat er geen algemene winnaar is maar dat men eerder specifieke DBMS'en heeft in specifieke voorvallen die uitblinken en die efficiënter werken op bepaalde vlakken of services. Daarom is er nood aan het onderzoeken welk van de huidige systemen het meest efficiënt is voor de omstandigheden van een bedrijf of publieke instantie. Om die efficiëntie te meten wordt er meestal onderscheid gemaakt tussen verschillende elementen, waardoor een DMBS als efficiënt werd ervaren. Namelijk door uitbreidbaarheid, betrouwbaarheid en performante. De onderzoeken van **Bassil2012**, **Gyoeroedi2015**, **AlshafieGafaarMhmoudMohmmed2017** focussen vooral op dat laatste aspect, namelijk de performantie. Ons onderzoek zal zoals de vooraf vermelde onderzoeken zich ook focussen op deze performantie en meer bepaald de uitvoeringssnelheid van bepaalde query's. We kunnen immers ons onderzoek niet baseren op andere elementen door een gebrek aan middelen en faciliteiten. Toch hebben we geprobeerd een identieke omgeving op te zetten voor de vergelijking van de 2 DBSM'en.

Het artikel werd als volgent onderverdeeld: in sectie 2 wordt ons literatuuronderzoek naar het onderwerp grondig besproken, Sectie 3 geeft kort onze hypothese weer, Sectie 4 beschrijft onze gevolgde methodologie, Sectie 5 beschrijft de resultaten die ondervonden zijn bij het uitvoeren van het experiment, Sectie 6 beantwoordt de vraag of ons onderzoek wel statistisch significante waarden heeft gebruikt en tot slot sluit het artikel zich af met een conclusie in Sectie 7.

2. Literatuuronderzoek

Het verschil tussen het artikel van Bassil en het 'Rise Of NewSQL is dat in het artikel van Bassil duidelijk wordt beschreven hoe de verkregen resultaten werden gevonden. Bij het artikel over NewSql worden alleen de bevindingen beschreven. Het gebruikte vakjargaon wordt telkens goed uitgelegd en er worden maar liefst 9 criteria aangehaald die zeer uitgebreid worden omschreven. In het artikel van Bassil worden er een aantal DBMS'en vergeleken, maar wordt er niet echt een enkelvoudige winnaar uit afgeleid doordat ze op meerdere aspecten werden getest. Bij het artikel over NewSQL wordt wel duidelijk welke voordelen NewSQL allemaal te bieden heeft en wat voordien al dan niet al mogelijk was. Beide artikels zijn interessant en geven een brede inkijk in de verschillen tussen DBMS'en en de verschillen qua SQL.

NewSSQL is, in tegenstelling tot de standaard relationele DMBS'en wel in staat om big data te verwerken. Waar de prestaties van steeds groter wordende data bij RDBMS'en nog steeds snel gaat doet NewSQL dit nog sneller, dit met minimale performance overhead.

Wanneer iets performant werkt, wil dat zeggen dat het zowel efficiënt als effectief werkt. In vele gevallen is dit niet makkelijk te achterhalen, of zelfs onmogelijk. Daarom is het vanzelfsprekend dat de testen die worden uitgevoerd duidelijk moeten zijn en vaak moeten herhaald worden. Op die manier bekom je een resultaat waaruit je met zekerheid een conclusie kan trekken. Hierbij is belangrijk dat de correcte zaken onderzocht worden en dat datgene wat onderzocht wordt voldoende gespecifieerd moet zijn om zo een eenduidig antwoord te krijgen op de vragen die eraan vooraf gingen.

3. Hypothese

Na het doornemen van alle achtergrondinformatie tijdens het litereatuuronderzoek en het lezen van het artikel van **Bassil2012** kwamen wij tot conclusie om een soortgelijke hypothese te formuleren. Deze hypothese is namelijk dat we verwachten dat SQL-server een betere performantie heeft dan MariaDB. Tot deze conclusie kwamen we omdat MariaDB een iets minder bekende Database systeem is, in plaats van de fel-begeerde SQL-server dat van een hoge populariteit geniet. Ook is MariaDB open Source terwijl SQL-server ontwikkeld is door de Tech-gigant Microsoft.

4. Methodologie

Voor het genereren van de grote hoeveelheid data hebben wij gebruikt gemaakt van de SQL Data Generator van Redgate . Met behulp van de trial versie is het mogelijk om één miljoen rijen te genereren voor elke tabel in de database. De software maakt het mogelijk om vanuit SQLServer de tabellen op te vullen. Om de data over te zetten naar MariaDB hebben we gebruikt gemaakt van query's die per tabel alle rijen selecteert. Dit resultaat hebben we dan gekopieerd naar een Excel bestand en opgeslaan als een CSV bestand. Elk CSV bestand kreeg de naam van de tabel waarvan hij de gegevens bevatte, dit zorgde ervoor dat het privision script van de virtuele machine, met MariaDB, ze kon inlezen. Toen beide databanken de data ter beschikking hadden, zijn we begonnen met het testen van de query's van Bassil (2012). Tijdens het testen van de query's van Bassil (2012) merkten we dat er enkele fouten in stonden. We hebben sommige query's dan ook lichtelijk aangepast zodat ze wel werkten. Al onze query's en gegenereerde data is ook te vinden op onze GitHub.

Specificaties testcomputer:

- Merk en type: HP Pavilion Notebook P4A68EA#UUG
- Besturingssysteem: Windows 10 Home versie 1709
- Moederbord: HP 8096 (U3E1)
- Processor: I7-5500U CPU @ 2.40GHz 2.39GHz
- RAM: 12GB Dual-Channel DDR3 @797MHz
- Opslag: 931 GB Hitachi HGST
- System type: x64-based

Specificaties van de virtuele machine:

- RAM: 4GB
- Aantal CPU's: 2

5. Experimenten

Voor de experimenten hebben we alle query's, behalve query 3, 7 en 8, 40 keer uitgevoerd. Tijdens de uitvoering hebben we de uitvoeringstijd bijgehouden en steeds opgeslaan in csv bestanden. De reden waarom we query 3 niet hebben uitgevoerd is omdat deze te veel geheugen in beslag nam voor de virtuele machine en extreem lang duurde op de testcomputer. Query 7 was geen probleem in SQLServer, maar wel in MariaDB. In query 8 stond dan weer een fout die we niet konden vinden, daarom hebben we ook deze laten vallen.

In tabel 1 kan u de gemiddelde uitvoeringstijd van de query's in beide databanken terugvinden. Om deze waarden te verduidelijken hebben wij deze ook in een staafdiagram gegoten. Dit staafdiagram vindt u terug in afbeelding 1. Wat u kan zien is dat SQLServer een snellere uitvoeringstijd heeft voor alle query's behalve query 1 en 11. Hiervoor hebben wij echter geen verklaring want we hadden verwacht dat SQLServer overal sneller zou zijn. Query 1 en 11 zijn ook gewone SELECT query's met als enige verschil dat 11 ook gebruik maakt van een join. In tabel 1 ziet u ook dat er geen uitvoeringstijd is voor query 7 in MariaDB, dit komt omdat deze na meerdere uren nog steeds geen resultaat had, dit in tegenstelling tot SQLServer die al een resultaat had na ongeveer 4 seconden.

De waarden voor de standaardafwijking van alle query's in zowel MariaDB als SQLServer vindt u terug in tabel 2. Opnieuw hebben wij deze waarden ook in een staafdiagram gegoten die voorgesteld staat op afbeelding 2. Desondanks de lagere uitvoertijden van MariaDB lijkt de database wel stabieler te werken. Dit kan u zien aan de lagere standaardafwijking van verschillende query's. MariaDB heeft een lagere standaardafwijking bij 5 van de 8 uitgevoerde query's.

In afbeelding 3 en 4 kan u een boxplot zien van query 10 voor respectievelijk MariaDB en SQLServer. Op het eerste zicht zijn dit zeer opmerkelijke boxplotten, maar als we de query nauwer bekijken zien we dat dit een logisch resultaat is. Query 10 is namelijk een DELETE op basis van enkele WHERE voorwaarden. Hierdoor duurt de eerste uitvoering veel langer dan de daarop volgende uitvoeringen waar de rijen die aan de voorwaarden voldoen al verwijdert zijn.

In afbeelding 5 en 6 ziet u een boxplot van query 11, alweer voor respectievelijk MariaDB en SQLServer. Wat u kan zien in de boxplotten bewijst ook de resultaten van de standaardafwijking. Wat een uitschieter is bij MariaDB, is nog steeds binnen het maximum voor SQLServer. Ook de afstand tussen de mediaan en de kwartielen ligt veel hoger bij SQLServer dan bij MariaDB.

6. Statistische Toetsing

Om te bepalen of ons onderzoek wel gebruik maakt van statistisch significante waarden maakten we gebruik van de z-test. Deze wordt meestal gebruikt voor onderzoeken van een middel-grote of zeer uitgebreide steekproef. De reden dat we niet

kozen voor de Student's t-test is, omdat de 3 voorwaarden waarvoor die ervoor zorgen dat een z-test gebruikt moet worden in plaats van de Student's t-test ook wel degelijk allemaal aanwezig waren. namelijk :

- De steekproef moet voldoende groot zijn (n¿30—n=30)
- De variatie van de toetsingsgrootheid moet normaal verdeeld zijn
- We veronderstellen dat de standaardafwijking van de populatie,s, gekend is

Onze steekproef heeft een populatiegrootte n = 40 (omdat onze query's 40 keer uitgevoerd werden), onze variatie is normaal verdeeld en de standaardvergelijking van onze steekproef in beide reeksen is gekend Voor het gebruik van deze test gebruiken we de reeds uitgerekende gemmiddelden van uitvoeringstijden per query van Mariadb en SQLserver. om de z-test van deze waarden te vinden stellen we onze nulhypothese ,dat Mariadb even snel zal zijn als SQLserver, als h0 in. Onze eigen hypothese dat Mariadb trager zal zijn dan SQLserver stellen we in als h1.Het kritieke gebied om H0 te verwerpen ligt dus aan de linkerzijde van de curve en we noemen deze toets dan ook linkszijdig. Als we nu een functie schrijven die de z-score gemakkelijk automatisch berekent in R zoals die van **Lobos2009** krijgen we met deze waarden een z = 0.7184457. Rekening houdend met alpha = 0,05 geldt dat indien $|z| < 1{,}96$ geldt er dat het verschil tussen de 2 reeksen heel significant is en we dus H0 moeten verwerpen .En indien $|z| > 1{,}96$ is er bijna geen significantie aan de steekproef. Bij onze steekproef is z = 0.7184457 wat kleiner is dan 1,96 en we dus met volle overtuiging onze nulhypothese kunnen weggooien.Dit geeft aan dat de verschillen in performantie tussen de 2 DBMS's statistisch significant zijn. Ook verstevigt dit ons vertrouwen dat ons onderzoek en hypothese in de goede richting gaan en bewijst nogmaals dat SQLserver sneller is dan Mariadb.

7. Conclusie

Bij ons onderzoek hebben wij ons alleen gebaseerd op de doorvoersnelheden, andere factoren, zoals het RAM gebruik en in hoeverre de CPU wordt belast hebben wij niet betrokken. We hebben geprobeerd op SQL server en MariaDB zo goed mogelijk met elkaar te vergelijken door de factoren die beïnvloedend kunnen werken zo gelijk mogelijk te houden. We kunnen besluiten dat onze hypothese, dat SQL server query's sneller verwerkt dan MariaDB grotendeels klopt. MariaDB is namelijk minder belastend voor het systeem, maar is daardoor ook minder snel. Voor gecompliceerdere query's is het aan te raden gebruik te maken van SQL Server.

8. Referenties

Györödi, C., Györödi, R., Pecherle, G. , Olah, A.(2015) A Comparative Study: MongoDB vs. MySQL verkregen van https://www.researchgate.net/profile/Cornelia_Gyroedi2/publication/278302676_A_Comparative_Study_MongoDB_vs_MySQL/links/557f

Sushant H.,Cloudhary T., Jain V.(2014) Rise Of NewSql Verkregen van https://www.researchgate.net/profile/Sushant_Choudhary/public

Mohmmed, A. G. M., Osman S.E.F. (2017). Study on SQL vs. NoSQL vs. NewSQL. Verkregen van http://www.jmess.org/wp-content/uploads/2017/07/JMESSP13420354.pdf

Bassil, Y. (2012, februari). A Comparative Study on the Performance of the Top DBMS Systems. Journal of Computer Science and Research, 1(1), 20–31.

Logos, T. (2009). Two sample Z-test. Verkregen van https://www.r-bloggers.com/two-sample-z-test/

9. Afbeeldingen

	Query1	Query2	Query4	Query5	Query6	Query7	Query9	Query10	Query11
MariaDB	3,05	0,64	12,05	16,81	10,80	/	1,46	4,51	14,41
SQLServer	7,49	0,50	7,44	7,58	3,90	3,89	0,64	3,19	16,45

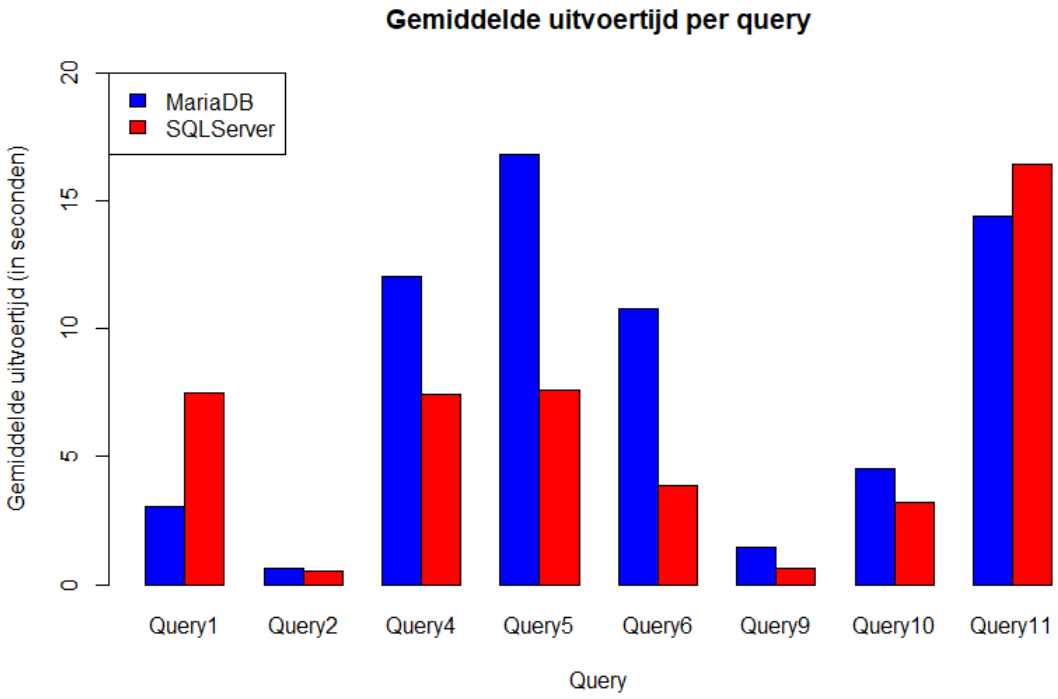
Tabel 1: Gemiddelde Uitvoertijd (in seconden)

Figuur 1

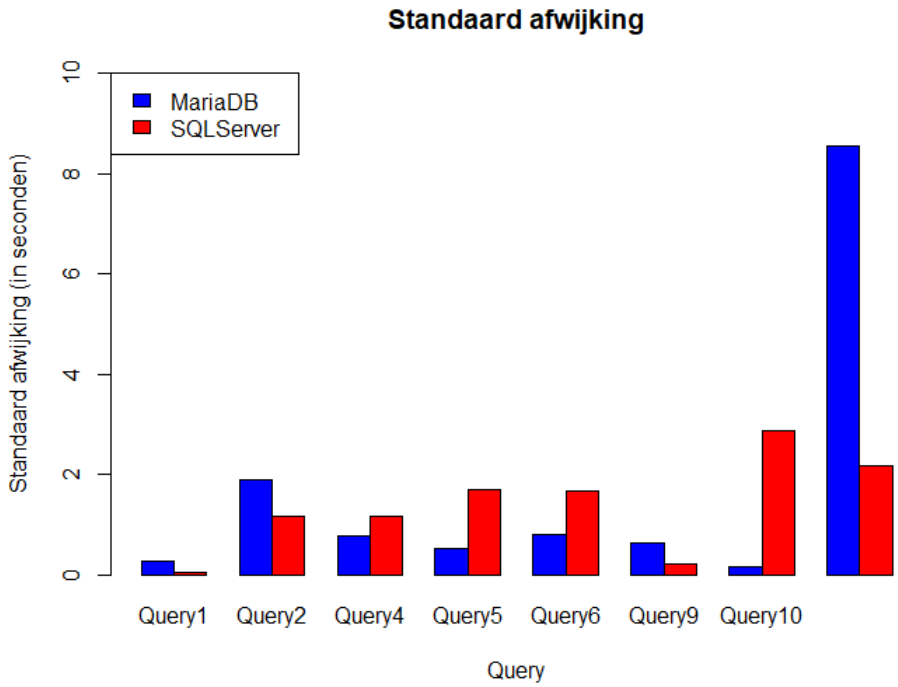
	Query1	Query2	Query4	Query5	Query6	Query9	Query10	Query11
MariaDB	0,269	0,046	1,888	1,154	0,771	1,160	0,524	1,696
SQLServer	0,789	1,666	0,643	0,199	0,158	2,863	8,545	2,163

Tabel 2: Standaardafwijking

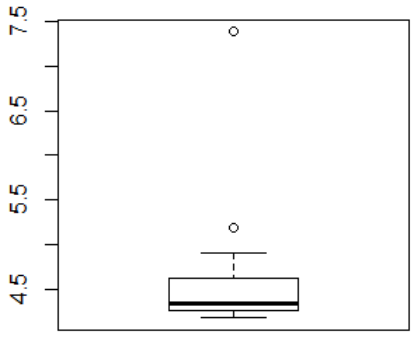
Figuur 2



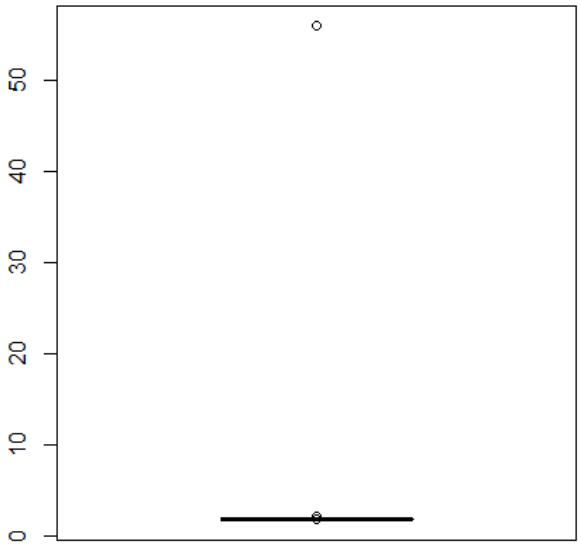
Figuur 3. afbeelding 1



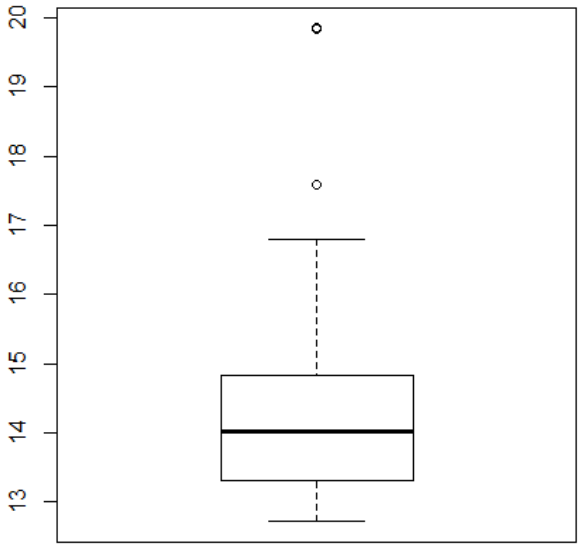
Figuur 4. afbeelding 2



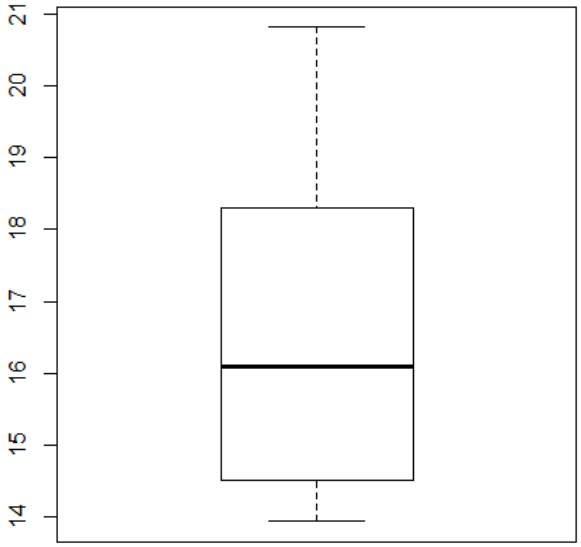
Figuur 5. afbeelding 3 Query 10 Mariadb



Figuur 6. afbeelding 4 Query 10 SQLServer



Figuur 7. afbeelding 5 Query 11 Mariadb



Figuur 8. afbeelding 6 Query 11 SQLServer