Universiteit van Amsterdam

Scheduling in the cloud

Cloud Computing: Het prioriteren van taken met behulp van scheduling

25 mei 2018

Tutor:
Willem Vermeulen

Practicumgroep:
A1 ALGOL

Docent:
Robert van Wijk

Cursus:
Besturingsystemen

Vakcode:
5062BEST6Y

Student:
Mo Diallo
11317329

1 Introductie

Cloud computing is een belangrijk onderdeel geworden van het informatietijdperk. De mogelijkheid om de rekenkracht van hardware te combineren en aan te bieden als dienst wordt steeds aantrekkelijker. Bedrijven realiseren zich dat het bezighouden met hun eigen IT infrastructuur niet voordelig is en het moeilijker maakt om de capaciteit van dien omhoog of omlaag te schalen. Doordat cloud computing hen de mogelijkheid geeft om dynamisch resources af te nemen investeren zij massaal in de groei van de technologie (Tripathy en Ranjan Patra, 2014).
De groei in populariteit van cloud computing providers brengt voor cloud service providers de vraag met zich mee hoe zij optimaal gebruik kunnen maken van de hardware die zij tot hun beschikking hebben. Studies tonen aan dat in een groot aantal datacenters het resource gebruik onderbenut is (Song, Xiao en Chen, 2013). Met cloud computing is het mogelijk die resources te combineren en te distribueren.
Vooral voor de wetenschap is cloud computing aantrekkelijk. Zei beschikken meestal niet tot het kapitaal dat noodzakelijk is voor het onderhouden van hardware. Overigens willen zij meestal alleen voor een korte tijd beschikken tot deze rekenkracht om calculaties en simulaties uit te voeren(Tripathy en Ranjan Patra, 2014). Als de performance kan worden verbeterd met behulp van een beter scheduling algoritme geeft dit hen de mogelijkheid om krachtigere calculaties uit te voeren, overigens zal efficiënter gebruik van de resources leiden tot een kostenbesparing bij cloud service providers, waarvan ook de gebruikers zullen profiteren.
In dit onderzoek zal centraal staan wat voor een scheduling algoritme het beste te pas komt bij cloud computing wanneer dit wordt aangeboden als hardware as a service (HaaS), en hoe de performance van cloud computing hiermee kan worden verbeterd.
Voorgaande onderzoeken(Vouk, 2008) hebben aangetoond dat traditionele scheduling algo-

ritmes (FCFS/RR) niet geschikt zijn voor cloud computing doordat deze algoritmes het distribueren van taken overlaat aan de volgorde van de taken zelf. Om in dit onderzoek verder te gaan op eerdere bevindingen zal eerst gekeken worden naar waaraan een scheduling algoritme voor cloud computers moet voldoen, daarna zal op verschillende algoritmes worden ingegaan en de gepastheid van deze algoritmes bekeken. Daaruit zal voortkomen welk scheduling algoritme het best voldoet aan de eisen en op welke manier deze verbeterd kan worden.

2 Cloud Computing, een introductie

Voordat kan worden gekeken naar welk algoritme gepast is en hoe deze verbeterd kan worden moet bekend zijn hoe cloud computing werkt, en daaruit komt voort welke eisen worden gesteld aan een scheduling algoritme voor cloud computing.

2.1 De werking van cloud computing

Cloud computing is een combinatie van parallelcomputing en gedistribueerde computing (Selvarani en Sadhasivam, 2010). Door middel van de virtualisatie van hardware is het mogelijk de rekenkracht van meerdere systemen te combineren en daarover taken te distribueren. Deze rekenkracht kan dan worden aangeboden als dienst.
Het voordeel van dit model van hardware as a service (HaaS) is dat voor de gebruikers, zolang zij normgaal gebruik maken van deze service, het lijkt alsof zij beschikken tot een oneindig aantal resources. Zij hoeven zich hierdoor geen zorgen te maken over de onderliggende hardware bij het gebruik van het systeem. Dit doordat de cloud service provider de voor hen beschikbare resources zal schalen naar het gebruik(Selvarani en Sadhasivam, 2010). Voor zowel de gebruiker en de cloud service provider is dit kost effectief. De gebruiker betaalt nu immers alleen voor het gebruik, en de cloud service provider heeft de mogelijkheid om ongebruikte resources verder te verdelen over de gebruikers.

2.2 Scheduling binnen cloud computing

Nu bekend is hoe cloud computing werkt kunnen eisen worden opgesteld voor een scheduling algoritme.
Aangezien cloud computing gedistribueerd is, moet een scheduling algoritme de taken verdelen over de beschikbare hardware. Dat is namelijk de kern van cloud computing(Hu e.a., 2010).
Ten tweede is het de bedoeling dat een gebruiker performance ervaart die beter is dan het hebben van eigen hardware of het huren van virtuele hardware. Het doel is dat het mogelijk wordt dat resources die een gebruiker tot zijn beschikking heeft zich aanpassen aan het gebruik. En gezien de resources gedeeld zijn onder gebruikers binnen cloud computing, is het wenselijk dat gebruikers geen hinder ondervinden van het gebruik van anderen (Vouk, 2008).
Als laatste een eis die niet specifiek is aan cloud computing: Het is wenselijk als een scheduling algoritme zo efficiënt mogelijk om gaat met de resources.

3 Scheduling Algoritmes

Wetend aan welke eisen een scheduling algoritme moet voldoen, kan worden gekeken naar de eigenschappen van verschillende scheduling algoritmes.
Uit Goel en Garg (2013) komen de volgende eigenschappen voort, kenmerkend voor scheduling algoritmes:

- Pre-Emptive/Non-Pre-Emptive (staat dit algoritme het toe om lopende taken te pauzeren voor andere taken)
- Priority based (Wordt een prioriteit gegeven aan bepaalde taken)
- Eerlijkheid (Krijgt iedere taak een eerlijke verdeling van de resources)
- Resource Optimalisatie (Wordt het resource gebruik vermindert waar mogelijk)

Tabel 1: Vergelijking scheduling algoritmes

	Pre-Emptive	Prioriteit aan processen	Eerlijkheid	Resource Optimalisatie
Round Robin	Ja	Nee	Ja	Nee
FCFS	Nee	Nee	Nee	Nee
Priority Based	Ja/Nee	Ja	Nee	Ja/Nee

3.0.1 First come first serve: Een vergelijking

Zoals de naam doet denken zal een FCFS algoritme taken scheduling in de volgorde waarin ze arriveren. Dit gebeurt ongeacht de eigenschappen/eisen van de taak. Er wordt geen optimalisatie gedaan om de performance te verbeteren, of belangrijkere taken te schedulen. Dit algoritme heeft als grootste nadeel dat het mogelijk is voor een taak om de resources bezet te houden voor onbepaalde tijd.

3.0.2 Round Robin: Een vergelijking

Een round-robin is een variatie van het FCFS algoritme. Iedere taak krijgt een tijdsblok waarin het taken mag uitvoeren. Echter brengt dit probleem met zich mee dat taken worden uitgevoerd in de volgorde waarin zij aankomen. Een belangrijkere taak zal mogelijk niet voldoende tijd krijgen om te voltooien. Een nadeel van dit algoritme is dat een taak niet de volledige tijd kan krijgen om te voltooien. Hierdoor is er een verspilling van geheugengebruik wanneer een taak wordt gepauzeerd.

3.0.3 Priority based job scheduling: Een vergelijking

Bij priority based job scheduling krijgt iedere taak een prioriteit. Aan de hand daarvan worden taken gescheduled. De manier waarop deze taken een prioriteit worden toegekend is volledig over aan de implementatie zelf. Overigens kan ook de keuze worden gemaakt of taken wel of niet worden onderbroken wanneer een taak met een hogere prioriteit binnenkomt. Het nadeel aan dit algoritme is dat voor ieder process een prioriteit moet worden vastgesteld. Taken met een lagere prioriteit krijgen geen eerlijke verdeling van de resources.

3.1 Meest geschikte algoritme

Op de eerder gegeven algoritmes zijn variaties in implementatie mogelijk. Echter blijft de basis veelal hetzelfde.
Als voor iedere algoritme wordt gekeken naar of het zal voldoen aan de eerder gestelde eisen valt te zien dat zowel round-robin en FCFS niet beschikken over de gewenste eigenschappen. Beide algoritmes maken geen onderscheid tussen belangrijkere processen.
De prioriteit van taken is belangrijk omdat in cloud computing verschillende taken eerder moeten worden uitgevoerd. Dit kunnen taken zijn die essentieel zijn voor de infrastructuur van het systeem, of doordat er voor iedere gebruiker een bepaald aantal minimum resources zijn gereserveerd(Patel en Bhoi, 2014).
Hieruit kan worden geconcludeerd dat een geschikt scheduling algoritme voor cloud computers een variatie is van priority based job scheduling.

3.2 Priority based scheduling: Een implementatie

Een implementatie van priority based job scheduling heeft dus de mogelijkheid om te voldoen aan de eerder gestelde eisen voor cloud scheduling algoritmes.
Een obstakel bij het implementeren is het bepalen van de prioriteit van taken. De scheduler moet de prioriteit van een taak kunnen bepalen zodra deze binnenkomt. Voor systeemtaken is het simpel om hen de hoogste prioriteit te geven, maar voor willekeurige taken moet de scheduler een *estimated-guess* maken aan de hand van de bekende informatie (Tripathy en Ranjan Patra, 2014). De informatie waaruit deze overweging moet worden gemaakt moet komen uit: Het totale aantal beschikbare resources, het totale aantal gereserveerde resources voor de indiener van deze taak en het gebruik van de indiener van deze taak.
Het priority based scheduling algoritme moet dus dynamisch prioriteiten geven aan processen.
Zodra de taken een prioriteit hebben is het triviaal om aan de hand daarvan ze te distribueren over de beschikbare hardware.

3.2.1 Verbeteringen

Deze eerdergenoemde implementatie vormt een basis voor een scheduling algoritme voor cloud computing. Er kan echter nog op worden verbeterd. Dit is mogelijk door hardware te specificeren die bestemd is voor alleen taken met de hoogste prioriteit. Hierdoor kan worden voorkomen dat lagere prioriteit taken volledig worden onderbroken door hogere prioriteit taken.
Overigens zorgt dit ook voor een betere resource optimalisatie. Als een taak eenmaal bezig is, wordt hiervoor geheugen gealloceerd. Als deze taak wordt onderbroken door een taak met een hogere prioriteit, zal dit geheugen nog steeds in gebruik zijn. Dit ondanks dat deze taak nu gepauzeerd is. Als het aantal gestopte taken dus kan worden verminderd, zorgt dit voor een verhoging in de doorvoer. Waardoor de utilisatie van resources optimaal blijft.

4 Discussie

Tot slot, het is nu bekend dat voor cloud computing geschikte scheduling algoritmes variaties zijn van algoritmes die prioriteiten toekennen aan taken, dit gaat hand in hand met eerdere bevindingen die aantoonden dat traditionele scheduling algoritmes die niet in staat zijn om dit te doen ongeschikt zijn. Dit doordat er veel verschillende taken zijn binnen cloud computing en niet ieder van deze taken even belangrijk is. En bij cloud computing is het essentieel dat de taken geprioriteerd worden. Aangezien cloud computing een samenhang is van verschillende systemen waarover taken moeten worden verdeeld.
Uiteindelijk is het mogelijk om de resource optimalisatie te verbeteren door een specifiek gedeelte van de hardware te specificeren voor alleen taken met de hoogst mogelijke prioriteit. Dit verkleint de kans dat taken met een lagere prioriteit gepauzeerd moeten worden, waaruit een verbetering van de doorvoer volgt. Zoals aan het begin van dit onderzoek aangegeven, zal dit ervoor zorgen dat zowel afnemer en aanbieders van de dienst(HaaS) hiervan profiteren zowel doordat het financieel aantrekkelijker zal worden om deze dienst af te nemen en doordat het schalen van performance ongeëvenaard is.
Dit onderzoek is algemeen ingegaan op de eisen van cloud computing systemen wanneer deze worden ingezet voor het aanbieden van HaaS, echter is het vanzelfsprekend dat specifiekere eisen kunnen worden gesteld aan de hand van hoe deze systemen worden ingezet. Een vervolg onderzoek zou kunnen kijken naar de mogelijke manieren waarop cloud computing kan worden ingezet, en in hoeverre dit scheduling algoritme daarvoor geschikt is. Overigens zou een vervolg onderzoek kunnen kijken naar manieren om de energie-efficiëntie te verbeteren. Dit onderzoek heeft dat aspect namelijk achterwege gelaten. De mogelijkheid bestaat dat wanneer energie-efficiëntie van belang is, een andere manier van scheduling beter geschikt is. Echter is de verwachting dat ook dan een variatie van priority based scheduling het meest

geschikt blijft.

Referenties

Vouk, M. A. (2008). "Cloud computing x2014; Issues, research and implementations". In: *ITI 2008 - 30th International Conference on Information Technology Interfaces*, p. 31–40. DOI: 10.1109/ITI.2008.4588381.

Hu, J. e.a. (2010). "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment". In: *2010 3rd International Symposium on Parallel Architectures, Algorithms and Programming*, p. 89–96. DOI: 10.1109/PAAP.2010.65.

Selvarani, S. en G. S. Sadhasivam (2010). "Improved cost-based algorithm for task scheduling in cloud computing". In: *2010 IEEE International Conference on Computational Intelligence and Computing Research*, p. 1–5. DOI: 10.1109/ICCIC.2010.5705847.

Goel, Neetu en R. B. Garg (2013). "A Comparative Study of CPU Scheduling Algorithms". In: *CoRR* abs/1307.4165. arXiv: 1307.4165. URL: http://arxiv.org/abs/1307.4165.

Song, W., Z. Xiao en Q. Chen (2013). "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment". In: *IEEE Transactions on Parallel Distributed Systems* 24, p. 1107–1117. ISSN: 1045-9219. DOI: 10.1109/TPDS.2012.283. URL: doi.ieeecomputersociety.org/10.1109/TPDS.2012.283.

Patel, S. J. en U. R. Bhoi (2014). "Improved Priority Based Job Scheduling Algorithm in Cloud Computing Using Iterative Method". In: *2014 Fourth International Conference on Advances in Computing and Communications*, p. 199–202. DOI: 10.1109/ICACC.2014.55.

Tripathy, Lipsa en Rasmi Ranjan Patra (2014). "Scheduling In Cloud Computing". In: *International Journal on Cloud Computing: Services and Architecture* 4, p. 21–27. DOI: 10.5121/ijccsa.2014.4503.