Performance Guarantees for Web Applications

Jiang, D.

2012

document version
Publisher's PDF, also known as Version of record

Samenvatting

Prestatiegaranties voor webapplicaties

Gebruikers stellen steeds meer eisen aan responsieve webapplicaties. Uit een onderzoek uit 2006 blijkt dat 62% van de internetgebruikers slechts 6 seconden of minder bereid is te wachten tot een enkele pagina te geladen is, voordat ze de website verlaten. Een meer recent onderzoek (2009) gaf aan dat deze prestatieverwachting hoger is geworden, en 83% van de internetgebruikers verwacht een webpagina te laden in 3 seconden of minder. Daarnaast toonde dit onderzoek ook dat 79% van de "online shoppers" die een bezoek aan een slecht presterende website brachten waarschijnlijk niets zouden kopen van deze site. Hieruit blijkt dat prestatiegaranties voor internetapplicaties bedrijfskritisch zijn.

Een belangrijke prestatienorm is de reactietijd van een webapplicatie. De reactietijd kan worden opgesplitst in drie delen: de wachttijd aan de gebruikerskant, de netwerkwachttijd en de wachttijd op de server. Onlangs hebben webapplicaties code die aan de op de machine van de gebruiker uitgevoerd wordt, zoals JavaScript, in gebruik genomen om de functionaliteit van applicaties uit te breiden. De wachttijd aan de gebruikerskant is de tijd die nodig is om de code aan de gebruikerskant uit te voeren. De onderzoeksgemeenschap heeft zich ingespannen om meerdere problemen met de prestaties aan de gebruikerskant op te lossen, zoals het gedrag van JavaScript code tijdens de executie te onderzoeken om de representativiteit van de benchmarkpakketten te verbeteren, het toepassen van bewaking op afstand ten behoeve van een prestatiediagnose van de gebruikerskant, en het doorvoeren van prestatieoptimalisaties voor JavaScript door op "traces" gebaseerde "just-in-time"-compilatie. De browseroorlog tussen de verschillende leveranciers in de ICT-industrie richt zich ook voor een belangrijk deel op prestatieverbeteringen voor JavaScript. De wachttijd aan de gebruikerskant hangt voornamelijk af van twee factoren: de toegepaste code aan de gebruikerskant, en specifieke mechanismen ingebouwd in elke webbrowser. Vanuit het perspectief van aanbieders van internetdiensten zijn deze twee factoren niet door hen te controleren.

De netwerkwachttijd verwijst naar de zendtijd van een reactie op een verzoek van de server naar de gebruiker over een netwerk zoals het Internet. Verschillende technieken, zoals “edge computing”, “caching” van gegevens, en de replicatie van gegevens zijn voorgesteld om deze wachttijden te verminderen. Commerciële producten zoals Akamai CDN en Amazon CloudFront richten zich ook op het waarborgen van de best mogelijke toegangsprestaties. Deze academische en industriële inspanningen tezamen reduceren de netwerkwachttijd van webapplicaties aanzienlijk, en zijn zeer succesvol.

Hoewel het optimaliseren van de wachttijd aan de gebruikerskant en op het netwerk van belang is, kunnen wij prestaties van een webapplicatie niet garanderen als niet ook de wachttijd op de server onder controle is. Eerdere experimenten toonden bijvoorbeeld aan dat de wachttijd op de server verantwoordelijk kan zijn voor bijna 50% van de totale wachttijd op een webapplicatie. Aangezien webapplicaties steeds complexer geworden kunnen we verwachten dat de wachttijd op de server alleen maar zal toenemen. De wachttijd op de server verwijst naar de verblijftijd van een inkomend verzoek op de server, dat wacht op een reactie. Een typische webapplicatie bestaat bijvoorbeeld uit een bedrijfslogicalaag en een gegevenslaag, waarbij de bedrijfslogicalaag kan worden uitgevoerd op een applicatieserver, terwijl de gegevenslaag vaak wordt uitgevoerd op een database-server. De wachttijd op de server omvat dan zowel de tijd benodigd voor het uitvoeren van de applicatiecode op de applicatieserver als de tijd benodigd voor het verkrijgen van gegevens van de database-server.

Het waarborgen van prestaties aan de aanbiederskant van een webapplicatie wordt bemoeilijkt door het feit dat de belasting van webapplicaties op computersystemen sterk fluctueert en zeer onvoorspelbaar is. Deze onvoorspelbaarheid en fluctuaties introduceren twee belangrijke eisen aan het hostingsysteem. Ten eerste moet een webapplicatiearchitectuur in staat zijn om willekeurige niveaus van belasting te kunnen accommoderen. Ten tweede moet het in staat zijn om haar eigen capaciteit aan te passen, teneinde wisselende volumes van webverkeer aan te kunnen.

Aan de ene kant kan men, gezien het feit dat webverkeer onvoorspelbaar is, niet op voorhand voorspellen wat de maximale werkbelasting van een webapplicatie zal zijn. Tegelijkertijd zijn aanbieders van webapplicaties erop gericht om zoveel mogelijk gebruikers aan te trekken voor een efficiëntere bedrijfsvoering. Daarom moet een webapplicatie schaalbaar zijn. Een schaalbare webapplicatie kan willekeurige volumes van verkeer verwerken door IT-middelen toe te voegen, zodat een acceptabel prestatieniveau behouden kan worden. De bouw van een schaalbare webapplicatie is echter niet eenvoudig, aangezien dit een zorgvuldige partitionering van zowel de bedrijfslogica- als datalaag vereist.

Aan de andere kant maken de fluctuaties in de werkdruk op de webapplica-

tie het onmogelijk om een “goede” vaste hostingcapaciteit tegen minimale kosten te bepalen. Kostenbewuste webapplicatieaanbieders, zoals bijvoorbeeld kleine en middelgrote aanbieders, verwachten een kosteneffectieve manier om hun applicaties te hosten. Door het “utility computing”-model toe te passen op de hosting van webapplicaties en het aantal IT-middelen dat webapplicaties gebruiken te variëren naar de daadwerkelijke belasting verwachten applicatieaanbieders de kosten te verminderen.

Utility computing biedt een model om IT-middelen te verpakken als een “bemeterde dienst”. Sinds het jaar 2000 hebben IT-leveranciers zich ingespannen om producten en diensten te onwikkelen die het “utility computing”-model implementeren voor computerclusters en datacenters. Onlangs is cloud computing begonnen met het toepassen van utility computing door IT-middelen aan te bieden op een “pay-as-you-go” manier. In clouds worden IT-middelen zoals rekenkracht, dataopslag en netwerkcapaciteit verhuurd als diensten en afgerekend naar gebruik. Het “utility computing”-model voorziet in het dynamisch toekennen van IT-middelen aan webapplicaties om aan verschillende niveaus van vraag naar deze middelen te voldoen. Een efficiënte dynamische toekenning van middelen wordt echter bemoeilijkt door uitdagingen van zowel de kant van webapplicaties als van de kant van hostingomgevingen. Dat brengt ons tot de centrale onderzoeksvraag van dit proefschrift: hoe kunnen de prestaties op de server van webapplicaties gewaarborgd worden op een kosteneffectieve manier.

Dit proefschrift maakt gebruik van de gemiddelde reactietijd van de server als de prestatienorm voor webapplicaties. Andere prestatienormen, zoals percentielen van de reactietijd, zijn ook bruikbaar om prestatiegaranties mee uit te drukken. Wij geloven dat onze technieken kunnen worden uitgebreid om dergelijke normen ook te ondersteunen. De kwestie van het waarborgen van prestaties van de server kan worden vertaald naar het behouden van een redelijke gemiddelde reactietijd voor webapplicaties bij fluctuerende verkeersvolumes. Een redelijke reactietijd is gedefinieerd als de maximale reactietijd waarin een applicatie een binnenkomende aanvraag moet verwerken. Webapplicatieaanbieders definiëren deze maximale reactietijd doorgaans in hun “Service Level Objectives” (SLO’s).

Naast het kiezen van prestatienormen gebruikt dit proefschrift het aantal gebruikte machines als een maat voor de kosten. Een gebruikte machine kan zowel een toegewezen fysieke machine in een cluster of een virtuele machine in een cloud zijn. Het aantal machines kan verder worden vertaald naar de monetaire kosten, als er een kostprijs per machine is vastgesteld.

Dit proefschrift behandelt voornamelijk twee aspecten van onze onderzoeksinspanningen om onze centrale onderzoeksvraag te beantwoorden: i) de bouw van een schaalbare webapplicatiearchitectuur, en ii) het ontwerpen van dynamische IT-middelentoewijzingssystemen.

De bouw van een schaalbare webapplicatie kan op twee manieren: opschalen en uitschalen. Opschalen betekent dat meer capaciteit, zoals de processorsnelheid en geheugen, aan de individuele applicatiesystemen en databasesystemen wordt toegevoegd. Uitschalen betekent daarentegen dat er meer systemen aan de twee lagen worden toegevoegd. Uitschalen presteert beter dan opschalen als de verhouding tussen prestaties en kosten voor webapplicaties in acht wordt genomen. Opschalen heeft ook een harde grens in de capaciteit van de hardware, terwijl uitschalen het mogelijk maakt om continu IT-middelen toe te voegen. Hierom construeren we in dit proefschrift een schaalbare webapplicatiearchitectuur met behulp van uitschaaltechnieken.

Het toevoegen van meer servers aan de bedrijslogicalaag van een webapplicatie kan de prestaties verbeteren, doordat de werklast voor iedere afzonderlijke server op dat niveau wordt verlicht. Het toewijzen van meer servers aan de gegevenslaag verbetert echter niet altijd de prestaties van die laag voor alle mogelijke niveaus van de belasting. Gedeeltelijke replicatie van gegevens en een zorgvuldige verdeling en plaatsing van gegevens kan leiden tot een verbeterde schaalbaarheid van de gegevenslaag, als er meer IT-middelen worden toegevoegd. De grove granulariteit van de verdeling beperkt echter de mate van schaalbaarheid van de huidige schaalvergrotingstechnieken. In dit proefschrift tonen we de potentiële schaalbaarheid van webapplicaties als gevolg van een fijnere granulariteit van de gegevensverdeling.

Hoewel een schaalbare webapplicatiearchitectuur veelbelovende mechanismen voor het waarborgen van de prestaties van webapplicaties biedt, worden webapplicaties nog steeds geconfronteerd met het probleem van fluctuerende verkeersvolumes. Het toewijzen van te veel middelen aan webapplicaties op basis van de maximale werkdruk kan leiden tot inefficiënt gebruik van IT-middelen, terwijl bij een toewijzing van te weinig IT-middelen een schending van de SLO wordt geriskeerd. De meest eenvoudige technologie die gebruikt wordt om de prestaties te garanderen voor webapplicaties bij fluctuerende verkeersvolumes is een dynamische IT-middelentoewijzing. Deze technologie bestaat uit het toewijzen van extra IT-middelen aan een webapplicatie wanneer de reactietijd de SLO dreigt te overtreden, en het afnemen van matig gebruikte IT-middelen van een webapplicatie indien dit mogelijk is met behoud van de SLO.

Complexe webapplicaties en heterogene hostingomgevingen vormen echter een probleem voor de huidige dynamische IT-middelentoewijzingstechnieken. Aan de ene kant zijn de huidige webapplicaties niet ontworpen als monolithische applicaties, bestaande uit drie lagen. De webapplicatie die gebruikt wordt om webpagina's van Amazon.com te genereren bestaat bijvoorbeeld uit honderden diensten. Het is in dergelijke applicaties, die bestaan uit meerdere interactieve diensten, moeilijk te achterhalen wat het knelpunt is voor de prestaties. Het is

nog moeilijker om dit probleem op te lossen met behulp van een dynamische en efficiënte toewijzinging van IT-middelen. Aan de andere kant leiden heterogene fysieke machines en virtuele machines in datacentra en clouds tot heterogene prestaties van virtuele hostingsmiddelen. Deze eigenschap beperkt ook de toepasbaarheid van de huidige middeltoewijzingstechnieken die uitgaan van het gebruik van homogene middelen.