



Michael Kok
Verkorten van de tijdlijnen van een performancetesttraject
Voorjaarsevent Testnet: 22 juni 2009

Samenvatting:

Slechte performance is een aanzienlijk risico voor nieuwbouwsystemen. Met het reactief corrigeren van performanceproblemen zijn aanzienlijke kosten gemoeid. De gevolgschade door slechte performance is niet zelden vele malen groter. Voor het reduceren van deze risico's kunnen we een performancetesttraject uitvoeren.

Een van de uitdagingen waar performancetesters net als alle andere testers mee te maken hebben is de verkorting van de tijdlijn.

In een applicatieontwikkeling is een zekere periode opgenomen tussen de oplevering van de applicatie en de datum waarop het systeem operationeel moet worden. De bouw van een applicatie heeft de natuurlijke neiging om uit te lopen en de opdrachtgever houdt graag vast aan de datum van in bedrijf name. Als gevolg daarvan blijft er gaande het project steeds minder tijd over voor het testen en het performance testen. Het hele testtraject komt daarmee onder druk te staan. Performancetesten maakt daarvan weer onderdeel uit en daarvoor blijft dan de minste tijd over. Het is dus interessant om te kijken naar mogelijkheden om het testtraject te verkorten en of eerder te starten. In deze presentatie belichten we de mogelijkheden van performancemodellen om dit tijdtraject te verkorten.

Voor het uitvoeren van performancetesten hebben we twee mogelijke technieken:

- Het traditionele performancetesten
- Performanceanalyse met behulp van een model.

Bij het traditioneel performancetesten sluiten we PC's of servers aan met software (de virtuele gebruikers) die net als menselijke gebruikers transacties uitvoeren op het systeem. Door het aantal virtuele gebruikers op te voeren wordt een toenemende belasting op het te testen systeem gelegd. Dit wordt gedaan totdat het systeem overbelast raakt en hetzij extreem langzaam gaat

functioneren hetzij uitvalt. Daarbij wordt tegelijkertijd het transactievolume bepaald en worden de responstijden gemeten. Op het moment dat het systeem niet meer aan zijn responstijdnorm voldoet weten we dus ook welk transactievolume is bereikt. Daarmee weten we dus of het systeem aan de eisen voldoet. Er zijn test-varianten waarbij bepaald wordt of het systeem bestand is tegen een snelle toename van het transactievolume (stress test) en of het systeem stabiel blijft onder langdurige belasting. Deze vorm van performancetesten vereist een acceptatietestomgeving met capaciteiten die vergelijkbaar zijn met die van de beoogde productieomgeving.

Bij deze techniek wordt de performance van het te testen systeem beoordeeld tijdens de uitvoering van de test. De beoordeling gebeurt op basis van transactievolume en responstijden. Het resultaat is primair dat we een go/nogo beslissing over het in dienst nemen van het systeem kunnen ondersteunen.

Bij de performanceanalyse worden ook transacties in een testomgeving uitgevoerd. Tegelijkertijd worden meetgegevens verzameld over het capaciteitsbeslag van de transacties op de infrastructuur. Dit gebeurt met de hand door een enkele gebruiker of net als bij het traditioneel performancetesten met behulp van een automatisch uitgevoerd script. De performance van het systeem wordt nog niet tijdens deze actie beoordeeld. Samen met andere gegevens over het gebruiksvolume en de capaciteiten van de infrastructuur worden deze gegevens in het model ingevoerd. Hier zitten tevens meetgegevens bij waarmee de uitkomsten van het model worden gecontroleerd. Hiermee worden twee validaties gedaan waardoor de betrouwbaarheid van het model wordt geborgd. Met behulp van het model worden de performance en de capaciteitsbehoefte van het systeem bepaald en beoordeeld. In het model kunnen de responstijden berekend worden. Daarbij kan de configuratie in het model worden aangepast, kunnen de capaciteiten horizontaal en vertikaal geschaald worden en kan het gebruiksvolume worden aangepast. Voor iedere verandering die we in deze inputs van het model invoeren wordt opnieuw de resulterende performance en capaciteitsbehoefte berekend. We zien dus in het model of de responstijden onvoldoende zijn en of dit komt door capaciteitsgebrek en / of andere oorzaken. Vervolgens kunnen we de responstijden opnieuw laten uitrekenen voor een productieconfiguratie met meer capaciteit. Doordat we de configuratie en capaciteiten kunnen aanpassen in het model (schalen) kunnen we de meetgegevens ook verzamelen op een testomgeving waarvan de capaciteiten niet vergelijkbaar zijn met de beoogde productieomgeving. Een testomgeving met beperkte capaciteiten volstaat voor het verzamelen van de benodigde meetgegevens. Als eerste rekenen we dan door wat de performance wordt in de productieomgeving bij het beoogde volume. Daarna kunnen we performance en capaciteit weer verder analyseren, bottlenecks en performanceproblemen in de applicatie identificeren etc.

In veel projecten komt de acceptatietestomgeving met productiecapaciteiten aanzienlijk later tot stand dan de “gewone” testomgevingen met kleine

capaciteiten. Het verzamelen van de meetgegevens in een gewone testomgeving levert veelal een tijdwinst van een maand op.

Er is een tweede aspect dat tijdwinst oplevert namelijk de grotere diepgang die performanceanalyse met behulp van een model biedt. Dit komt onder andere tot uitdrukking in de volgende gevallen:

Meervoudige bottlenecks in de infrastructuur

Bij de traditionele performancetest worden de bottlenecks een voor een bepaald en door schaling van de hardware ook een voor een verholpen. Met behulp van het model worden alle bottlenecks in een keer tegelijk zichtbaar gemaakt en in een keer door schaling van de hardware verholpen. Per bottleneck levert dit een tijdwinst van minimaal een week, maar niet zelden aanzienlijk langer.

Software resources

Er zijn diverse software resources zoals connection pools, sessies en software servers die via parameterinstellingen voldoende ruim moeten worden geconfigureerd omdat ze anders performancedegradaties veroorzaken. Bij de traditionele performancetest moet voor iedere parameterinstelling een aparte test worden gedaan. In contrast daarmee stelt het model ons in staat om de performance voor allerlei parameterinstellingen zonder aanvullende tests / metingen te bepalen.

Volumeveranderingen

Indien het gebruiksvolume van een applicatie drastisch toeneemt kan de impact daarvan met het model zichtbaar worden gemaakt zonder verdere tests.

In de presentatie wordt een voorbeeld van een performanceanalyse getoond.

Conclusies

Met behulp van performanceanalyse kan in een performancetesttraject een aanzienlijke tijdwinst worden geboekt doordat gewoonlijk eerder kan worden gestart.

Doordat een grotere diepgang wordt gehaald bij het analyseren van de performance en capaciteitbehoefte van het systeem kunnen correcties aan applicatie en / of infrastructuur sneller worden uitgevoerd. Gezien de korte tijd die gewoonlijk beschikbaar is voor de uitvoering van het testtraject betekent de tijdwinst een aanzienlijk voordeel dat zich vertaalt in een verdergaande reductie van de risico's van een slechte performance.

Biografie

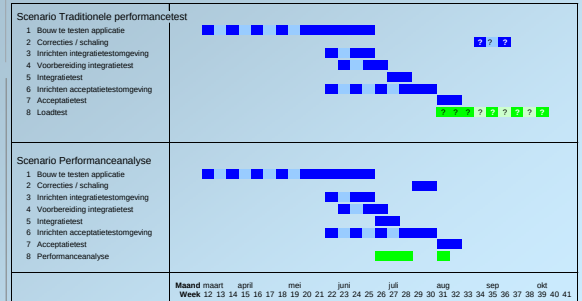
Michael Kok is actief in de praktijk van het performancetesten met behulp van de mBrace-aanpak, een aanpak waarbij een performancemodel centraal staat. Hij is oprichter en CTO van Contentional, een bedrijf gespecialiseerd in proactief performance management. aspecten van het testen, o.a. voor ICSTEST, TestNet, Logica en Capgemini.

De langste dag? Verkort tijdslijnen in een performancetesttraject mbv performanceanalyse

by Michael Kok

TestNet Voorjaarsevenement 22 juni 2009

De langste dag? Tijdslijnen?



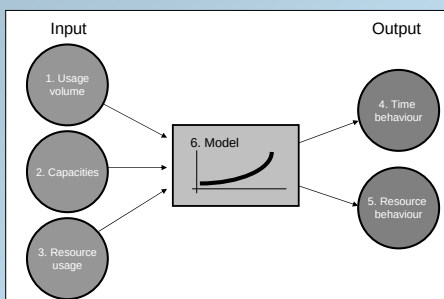
Traditionele Loadtest

- Beproeft systeem tijdens de test
- Rapportage
 - Responstijden
 - Gedeeltelijk bezettingen van hardware resources
 - Voor een situatie

Performanceanalyse

- Verzamelt tijdens de test meetgegevens
- Analyse naderhand mbv model
- Rapporteer:
 - Responstijden
 - Componenten van responstijden
 - Bezettingen van alle hardware en software resources
 - Voor iedere relevante situatie

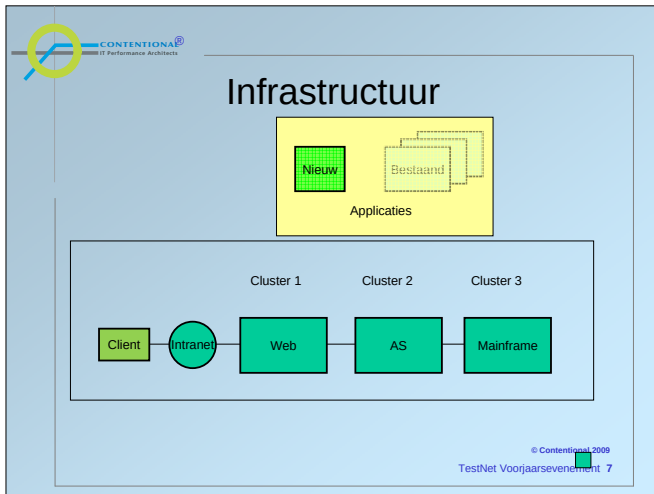
Overview of the model



Data are collected into a meta model resulting in a specific model.
The specific model is maintained as long as the customer is interested.

Voorbeeld performanceanalyse

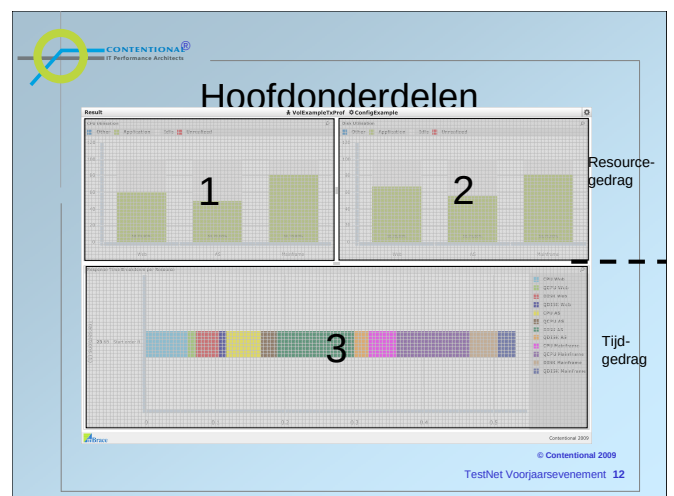
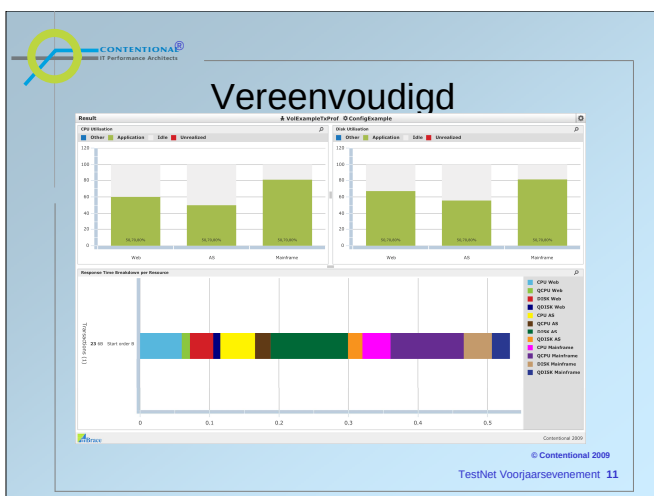
- Voorbeeld van een performanceanalyse uitgevoerd mbv het mBrace model

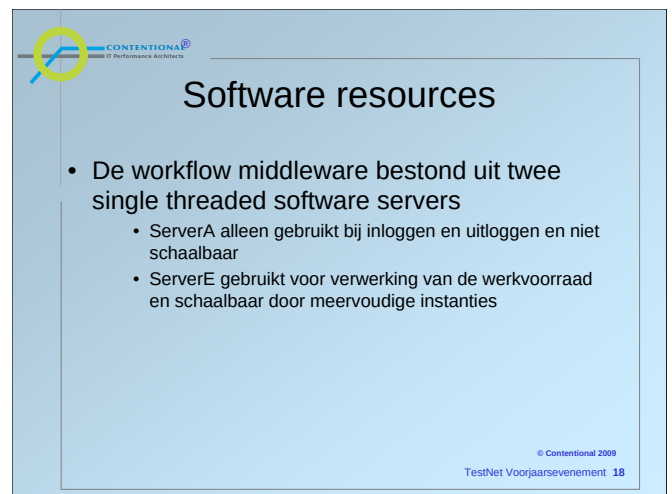
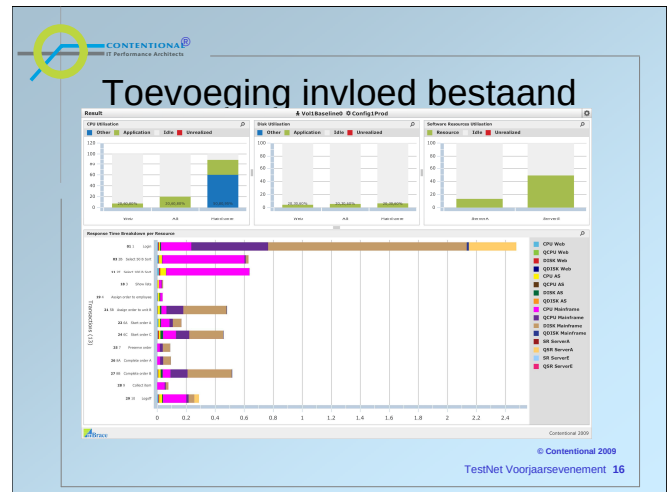
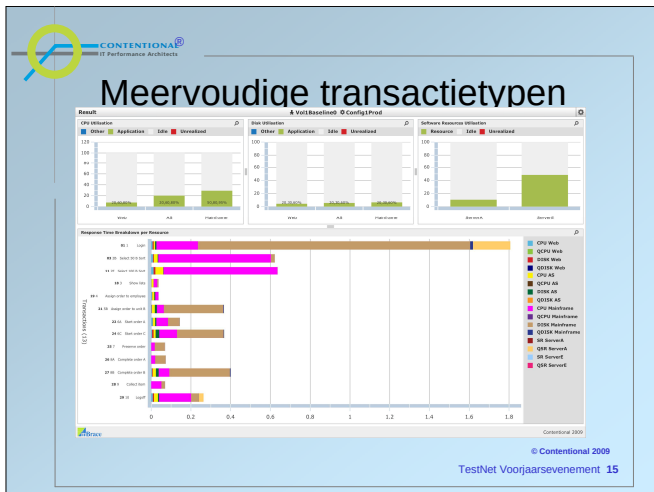
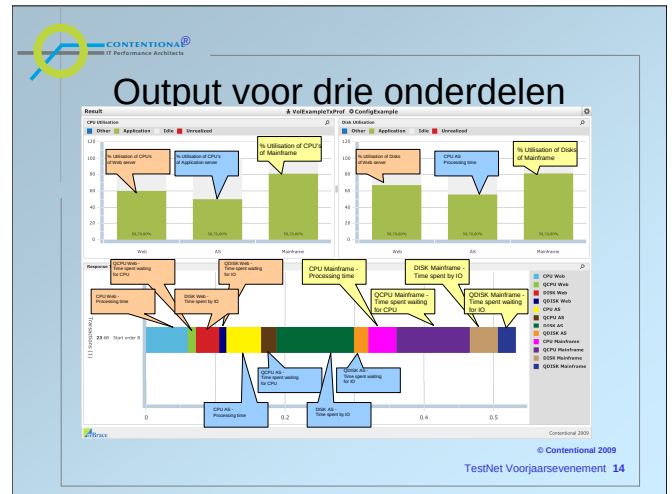
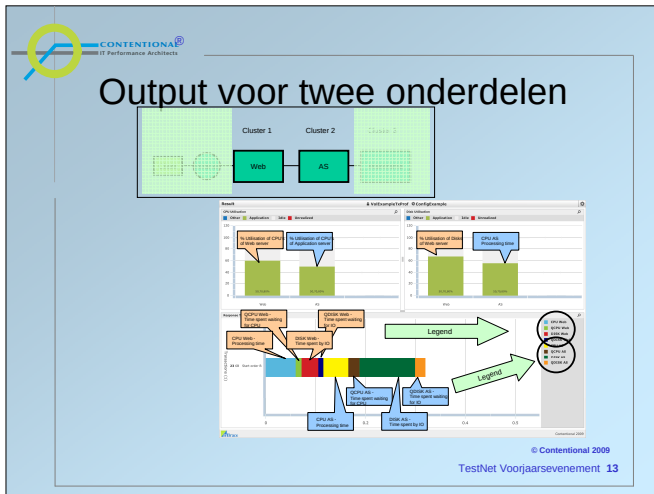


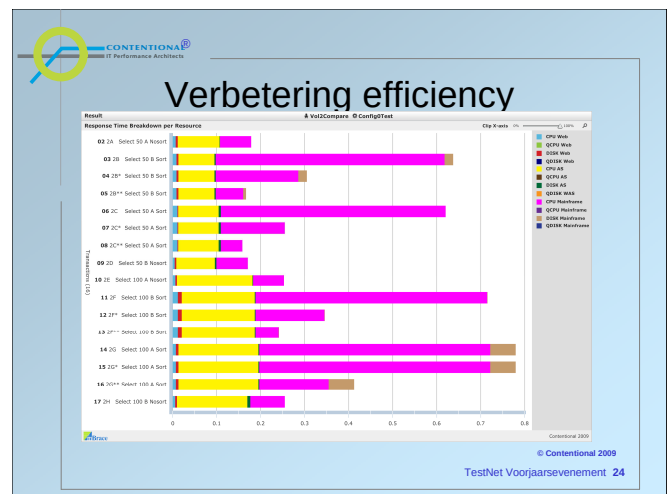
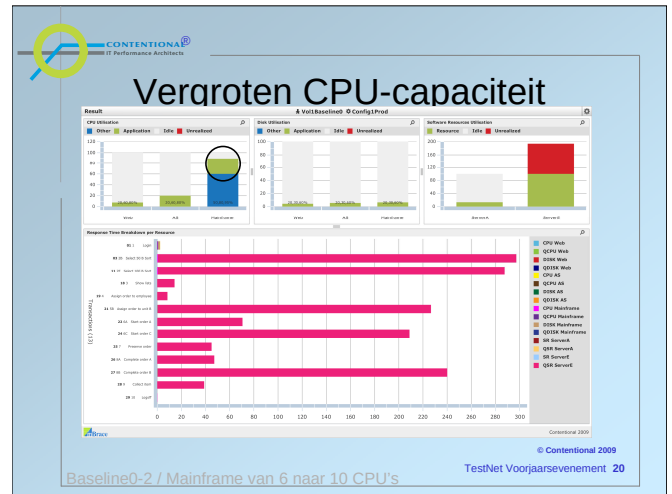
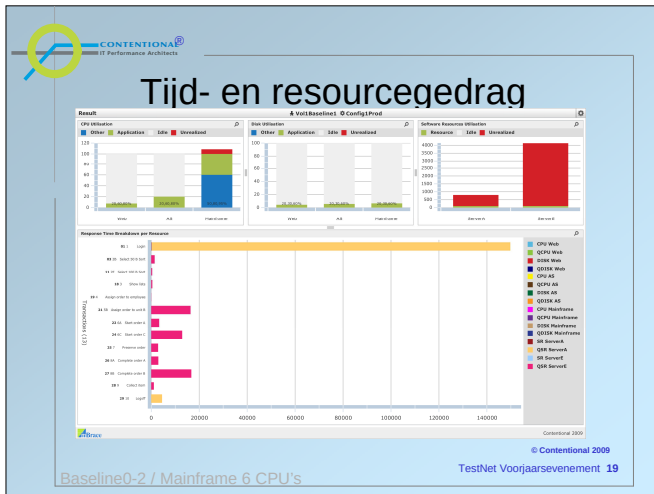
- Scope**
- Op basis van risicoanalyse werden Client en Netwerk buiten de scope gehouden
- © Contentional 2009
TestNet Voorjaarsevenement 8

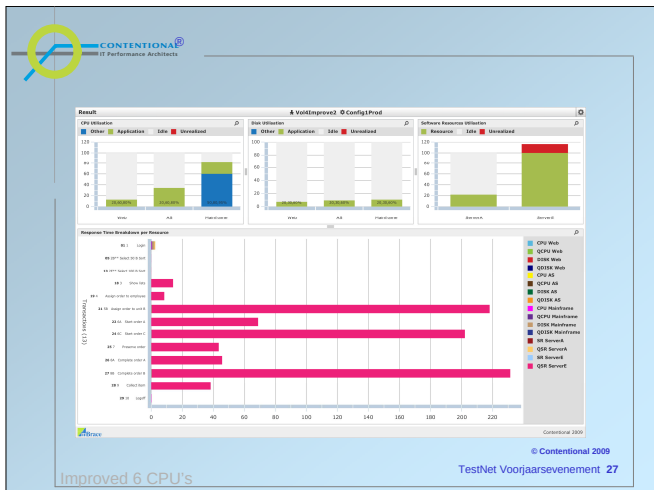
Toelichting grafische output

© Contentional 2009
TestNet Voorjaarsevenement 9









Samenvatting

- Een (1) test / meting uitgevoerd op een kleine testomgeving
- Met het model werd hierop een analyse uitgevoerd met als resultaten:
 - Meer bottlenecks opgelost: hardware en software, in 1 analyse
 - Door transactiegewijze analyse werd een mogelijke efficiencyverbetering ontdekt
 - Deze werd door de applicatieontwikkelaars gerealiseerd
 - Hiermee werd een grote besparing gerealiseerd

© Contentional 2009
TestNet Voorjaarsevenement 28

De langste dag?

- Door testen / meten in de gewone testomgeving kon de start een maand worden vervroegd
- De eerste performanceanalyse werd binnen drie weken uitgevoerd
 - Twee weken voorbereiding testomgeving
 - Een week meten en analyseren
- De correcties aan de applicatie kostten twee weken
- De tweede performanceanalyse kostte 1 week
- De deadline voor in productie name bleef gehandhaafd
- Een traditionele loadtest had pas vijf weken later kunnen starten
- Het stabiliseren van de performance van software resources kost gemiddeld aanzienlijk langer dan via de performanceanalyse

© Contentional 2009
TestNet Voorjaarsevenement 29