

De intrede van J2EE applicaties in grote infrastructuren brengt voor beheerders van deze infrastructuren belangrijke extra aandachtspunten. Juist door de grote flexibiliteit van J2EE blijven eventuele problemen ook sneller verborgen. Problemen die een onverwachte impact kunnen hebben op de totale infrastructuur.

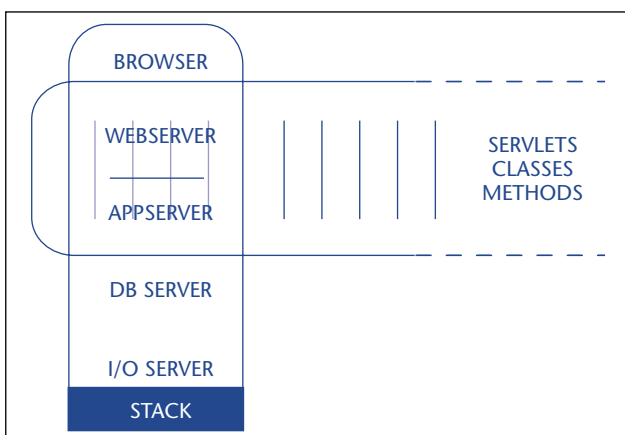


Performance management in J2EE omgevingen

Grote flexibiliteit en verborgen problemen

J2EE is dé development omgeving voor deze internetgebaseerde applicaties. De behoefte om snel enterprise- en dotcom applicaties over het web aan te bieden blijft groeien. Deze groei van het aantal op Internet-technologie gebaseerde applicaties zal volgens analisten voorlopig nog doorgaan. Het J2EE-platform biedt grote flexibiliteit, geeft ongekende mogelijkheden voor gedistribueerde development teams en is eindelijk een echt open development platform. J2EE is dusdanig ontworpen dat 'iedereen' erin kan programmeren. En met de overdraagbaarheid van het ene systeem naar het andere heeft het platform ongekende populariteit verworven.

AANDACHTSPUNTEN BIJ BEHEER Typisch voor J2EE applicaties zijn de grote development teams waarmee gewerkt wordt. Deze samenwerkingsstructuur is



FIGUUR 1. De complexiteit van J2EE-applicaties verdeelt zich over twee assen

een goede voedingsbodem voor verborgen problemen. Debuggers worden natuurlijk gebruikt echter deze checken slechts het semantische en syntaxische deel. Doordat deze development teams 'onderdelen' opleveren, bestaande uit vele servlets, classes en methods, en omdat al deze onderdelen weer invloed hebben op de diverse servers in de infrastructuur ontstaat een situatie waarin de complexiteit zich over twee assen verdeelt.

Enerzijds lijkt programmeren simpel, anderzijds brengt Java meer complexiteit dan andere programmeertalen als C en C++. Programma's in deze talen geven bijvoorbeeld een directe vertaling van broncode naar machinetaal. Bij Java zit daar nog de vertaalslag de zogenaamde 'interpreter' tussen, ook wel Java Virtual Machine genoemd. J2EE applicaties staan niet op zichzelf. Ze zijn onderdeel van de totale infrastructuur en hebben ook effect hierop. En daarin ligt ook precies de keerzijde van de openheid en het gemak van Java. Deze keerzijde komt tot uiting in de beheeromgeving. De voorbeelden zijn talrijk: bij het openen van een webpagina wordt de database niet één keer, maar tientallen keren geraadpleegd door een simpele fout in de broncode. Op die manier komt bijvoorbeeld database performance onder druk te staan van J2EE applicaties.

KRITISCHE MIDDENLAAG Zodra aan de beheerkant duidelijk wordt wat de keerzijde van development in J2EE is, is eigenlijk een groot deel van het gevaar geweken. Als rekening gehouden wordt met de eigenschappen van dit platform die verborgen problemen bevorderen en men de centrale rol die deze Java applicaties spelen serieus onder de loep neemt, is eigenlijk al

de eerste stap gezet. Dan is duidelijk dat het cruciaal is voor de prestaties van de totale infrastructuur om continu inzicht te hebben in de effecten van acties en raadplegingen van de webapplicaties.

Moderne web applicaties gebruiken middenlaag (mid-tier) servers die aan alle voorwaarden voldoen voor wat betreft schaalbaarheid, snelheid en hoge beschikbaarheid. Webservers in deze middenlaag handelen alle aanvragen van gebruikers af en geven direct

De samenwerkingsstructuur is een goede voedingsbodem voor verborgen problemen

respons wanneer alleen statische content wordt gevraagd. Wanneer er dynamische content gevraagd wordt, worden de aanvragen doorgestuurd naar het applicatie-verwerkingsdeel van deze mid-tier servers. De plaats waar ook alle datarequests van de back-end servers of andere remote services terechtkomen.

Dit is de plaats waar tegelijkertijd vele aanvragen verwerkt moeten worden waardoor deze middenlaag een kritisch component wordt in de applicatie infrastructuur.

WEB-BASED APPLICATIES Gedistribueerde web-architecturen zijn ontworpen om te kunnen reageren op grote hoeveelheden aanvragen die ook nog eens zeer gevarieerd zijn: hoge volumes en een brede transactiemix. De onderliggende architectuur moet zich kunnen aanpassen aan deze eisen. Dat geldt ook voor de performance management oplossingen. Performance management van web-based applicaties moet minstens net zo flexibel zijn als deze onderliggende applicatie infrastructuur. Strenge eisen aan performance moeten erin resulteren, dat het verzamelen en analyseren van performance informatie minimale impact mag hebben op het operationele systeem: met weinig belasting realtime informatie verzamelen die nodig is om tactische bewaking uit te kunnen voeren, terwijl tegelijkertijd uitgebreidere informatie gegeven wordt voor meer gedetailleerde analyses en strategische planning. Dit is de eeuwige tweestrijd tussen lage overhead in de productie omgeving en uitgebreide en gedetailleerde informatieverzameling.

STANDAARDCOMPONENTEN Verder is de snelheid waarmee nieuwe applicaties ontwikkeld en in gebruik genomen worden een complicerende factor voor performance management. Het snel kunnen aanpassen van performance management oplossingen aan veranderende applicaties is essentieel geworden. Om zoveel

Advertentie

mogelijk op tijd en kosten te besparen moet dit bij voorkeur geautomatiseerd plaatsvinden.

De toename van aanbod van standaard applicatie componenten door derde partijen is een antwoord van de markt op de steeds grotere druk op tijd en kosten. Ook de invoeging van deze componenten als een belangrijk onderdeel van de totale applicatie moet ondersteund worden door de performance beheer oplossingen. Deze componenten zijn vaak niet ontworpen voor specifieke productie omgevingen. De performance van de componenten moet echter wel bewaakt worden in relatie tot zo'n specifieke omgeving. Performance metingen van deze componenten in een specifieke applicatie zijn een belangrijk onderdeel bij de beslissing of de voordelen van hergebruik in balans zijn met de beperkingen in schaalbaarheid die dit kan opleveren. Het inzetten van dezelfde tools voor zowel performance testing als de operationele performance bewaking biedt de mogelijkheid om echte performance analyses uit te voeren.

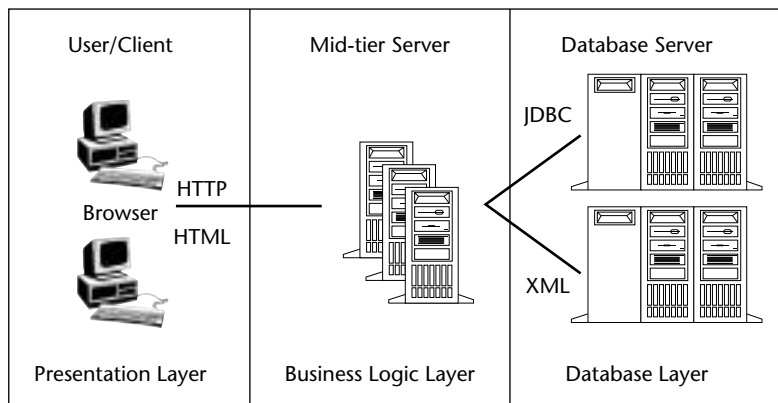
Operationele performance bewaking vereist dezelfde technologie als load testing, maar vereist ook additionele functionaliteit zoals:

- Bewaking van service levels om de juiste performance te bieden tegen minimale investeringen en operationele kosten.
- Instellen van alarms, gebaseerd op J2EE performance.
- Aanpassing aan configuratieveranderingen en tuning op andere systemen, waardoor de bottlenecks van het systeem verschuiven.
- Het genereren van accurate trends en forecast rapportage zodat resources op het meest efficiënte moment ingezet kunnen worden.
- Het bewaken van services op afstand ten opzichte van het vastgelegde service level.

De middenlaag is afhankelijk van het host operating system en de server configuratie, maar vooral ook van de Java applicatie server en Java Virtual Machine. Zoals al eerder genoemd wordt deze performance beïnvloed door de JVM runtime compiler, maar ook door de vele keuzen die er zijn voor applicatie pool, queue en cache management. De snelle voortgang van Java- en applicatieserver technologie resulteert in ontelbare tuning parameters waardoor performance management voor de middenlaag een zeer complexe taak is.

AANDACHTSPUNTEN Performance management oplossingen voor J2EE applicaties moeten aan vele voorwaarden voldoen. Belangrijk zijn in ieder geval de volgende aandachtspunten:

Rapid Application Development en snelle inzetbaarheid van applicaties betekent dat er snelle automatische aanpassing mogelijk moet zijn. Zowel bewaking als ana-



FIGUUR 2. De mid tier server is een kritisch component in de infrastructuur voor webapplicaties

lyse zijn daarin belangrijk. Een snel en goed inzicht in de situatie betekent dat problemen snel gevonden en opgelost kunnen worden. Response tijd en afhandeltijd zijn essentiële performance gegevens. Om snelle pro-

Zodra aan de beheerkant de keerzijde van J2EE-development duidelijk wordt, is eigenlijk het gevaar geweken

bleemdiagnoses te kunnen stellen, moet er gedetailleerde correlatie, drill down en analyse uitgevoerd kunnen worden. Zo moet bijvoorbeeld CPU resource consumptie gemeten vanuit het onderliggende besturingssysteem in relatie gebracht kunnen worden met een specifieke applicatie activiteit.

Correlatie van JVM activiteiten met de applicatie geeft een extra inzicht in de oorzaken van performance teruggang. Dit kan bereikt worden door JVM sampling te combineren met applicatie instrumentatie. Sampling kan CPU hot spots detecteren die op hun beurt weer in relatie gebracht kunnen worden met applicatiecomponenten en aanvragen van gebruikers. CPU hot spots, die ontdekt zijn met sampling, moeten in relatie gebracht kunnen worden met applicatiecomponenten en gebruikersaanvragen op alle niveaus.

Natuurlijk zijn ook historische gegevens, trends en analyse-mogelijkheden en een overzichtelijke user interface belangrijk bij performance management van deze, wat beheer betreft, complexe J2EE omgeving.

Anjo Kolk is werkzaam bij Precise Software Solutions. Voordat hij bij Precise in dienst trad werkte hij meer dan zestien jaar bij Oracle Corporation, waar hij zich vooral bezig hield met rdbsm-performance. Hij is per e-mail te bereiken op akolk@precise.com.