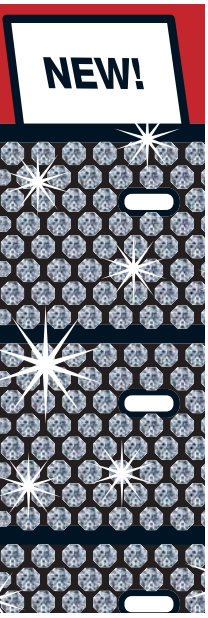


Werkwijze wordt omgedraaid met inzet van query's

Complex Event Processing

Robbert Hoeffnagel

De fameuze real-time enterprise waar goeroes en analisten het de afgelopen tijd zo vaak over hebben, stelt forse eisen aan de onderliggende technische infrastructuur. Hoe gaan we bijvoorbeeld om met een TCP/IP stack die per seconde tienduizenden 'berichten' doorstuurt? De traditionele database kan dat niet real-time verwerken. Met een aanpak die 'Complex Event Processing' wordt genoemd, komen we wellicht wel een eind.

Er zijn van die situaties waar geen sprake is van 'technology push', maar van – zeg maar – 'management pull'. Een mooi voorbeeld daarvan is het fenomeen 'real-time enterprise'. De term is de afgelopen jaren met name door Gartner populair gemaakt. Het idee is even eenvoudig als technisch complex: nu we in een wereld van globalisering leven en het gevaar c.q. de concurrentie steeds vaker uit onverwachte hoek komt, heeft het management van de organisatie grote behoefte aan snelle informatie (zie afbeelding 1). Hoe dichter we wat dat betreft bij 'real-time' informatie kunnen komen, hoe groter de kans dat het management snel op mogelijkheden of bedreigingen kan inspelen.

TCP/IP stack

Bij een real-time enterprise spelen tal van nieuwe ontwikkelingen: Business Intelligence, Service Oriented Architecture, Business Process Management, noem maar op. Al die omgevingen brengen enorme hoeveelheden data voort. Neem alleen al het fenomeen Business Activity Monitoring (BAM). Iedere aanbieder van SOA- en BPM-technologie heeft er de mond van vol, maar vaak blijkt pas in de praktijk wat er daadwerkelijk aan dataverkeer ontstaat als men een beetje complex proces continu moet monitoren.

> De query's zullen in de regel worden opgesteld door gespecialiseerde medewerkers

Analisten die zich met deze materie bezig houden, spreken van 'TCP/IP stacks' die honderdduizenden berichten per seconde doorgeven. In Europa denken we dan al gauw dat die aantallen alleen bij de megabanken op Wall Street voorkomen, maar zelfs als we dat aantal terugbrengen tot tienduizenden berichten per seconde, hebben we het natuurlijk over zeer grote hoeveelheden data die moeten worden verwerkt. Al die berichten zullen bovendien geanalyseerd moeten worden. Dat vereist niet alleen uiterst snelle hardware die de hiervoor ontwikkelde programmatuur ondersteunt, maar we moeten ook nog iets met die data doen. Een traditionele database-omgeving is echter niet in staat om al deze data in real-time vast te leggen en vervolgens weer beschikbaar te stellen aan applicaties die analyses willen toepassen.
In een eerder artikel in Database Magazine ('Balans tussen brute kracht en complexe acties', februari 2007) is al eens aangegeven dat min of meer traditionele oplossingen als in-memory databases of de met name uit de industriële automatisering bekende 'alert engines' ook geen soelaas bieden. De eerste productgroep vereist toch weer dat data eerst worden opgeslagen voordat er van een analyse sprake kan zijn, terwijl alert engines weliswaar zeer snel zijn, maar nauwelijks intelligentie kennen.

Stream processing

Het zijn wellicht toch weer die megabanken die voor een oplossing zorgen. Bij grote banken wordt al geruime tijd gewerkt met een aanpak die 'algoritmic trading' wordt genoemd. Op de aandelenbeurzen met zijn enorme aantal financiële producten kunnen de handelaren zelf niet meer alle koersontwikkelingen in de gaten houden. Bovendien is snelheid hier van cruciaal belang. Een verschil in koers tussen twee beurzen of aandelen dat slechts een paar minuten bestaat, kan al de kans op een miljoenenwinst – of verlies – opleveren.
In deze sector doet daarom momenteel een aanpak opgang die 'Complex Event Processing' heet. Overigens is die naam wellicht wat verwarrend. Er wordt namelijk ook wel gesproken van 'event stream processing' of simpelweg 'stream processing'. Het draait echter allemaal om het razendsnel verwerken van 'events'.

Afbeelding 1. De rol van Complex Event Processing binnen de real-time enterprise.

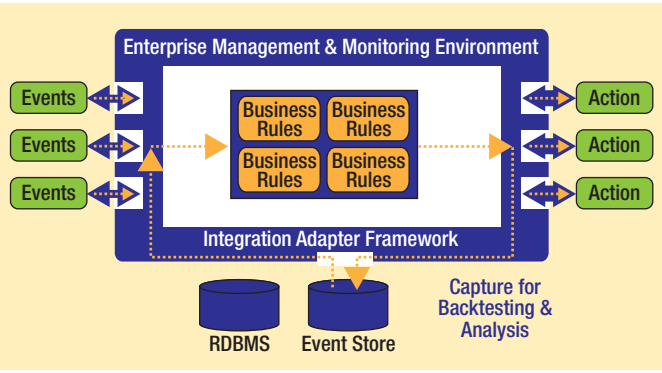
Een event is hierbij een gegeven dat een gebeurtenis representeert die in de echte wereld heeft plaatsgevonden. Een voorbeeld: RFID-tag nummer 121.19.1818 is om 13.09 uur gescand bij poort 10. Of neem een security-voorbeeld: TCP/IP-adres 128.1.32.298 heeft toegang verkregen tot server 5. Beheersystemen brengen continu dit soort meldingen voort. Het zijn allemaal events.

Individuele events zullen in de regel weinig informatie opleveren, tenzij bijvoorbeeld een security rule wordt overtreden. Anders wordt het echter als er zich reeksen van gerelateerde events voordoen: event A volgt op event B, waarna binnen zeven seconden event C optreedt. Tijd speelt hierbij een belangrijke rol: het optreden van een reeks van events binnen een bepaalde periode. Maar het kan ook gaan om een geografische afbakening. Met andere woorden: locaties. Bij Complex Event Processing gaat het er om deze relaties tussen events in real-time te herkennen.

Gekoppelde acties

Bovendien is het de bedoeling dat er aan het optreden van zo'n reeks van events direct een actie kan worden gekoppeld. Bij die megabanken van Wall Street is dat uiteraard een aankoop- of verkooporder, maar in het geval van een supply chain kan het optreden van een bepaalde reeks van events bijvoorbeeld duiden op een knelpunt bij een toeleverancier. Dan dient een alarm te worden afgegeven of moet een opdracht uitgaan, waardoor liefst langs geautomatiseerde weg een verandering in het proces optreedt zodat het probleem wordt opgelost. Daarmee zitten we dus op het spoor van de real-time enterprise. Als uit de Business Activity Monitoring rond een proces blijkt dat er bepaalde events optreden en deze gegevens direct geanalyseerd kunnen worden, kan in principe dus ook direct handelend worden opgetreden.

De vraag is natuurlijk wel hoe Complex Event Processing nu technisch in zijn werk gaat. Er is een omgeving nodig die data ofwel events ontvangt, waarna vervolgens gekeken wordt of zich een relevant patroon voordoet. Dat leidt dan vervolgens tot een specifieke actie (zie afbeelding 2). Klassieke databases en datawarehouses zijn hiervoor niet geschikt, omdat deze ontwikkeld zijn met het oog op het verwerken van statische data. Er is bij het analyseren van dynamische data geen tijd om data eerst vast te leggen, vervolgens indexen bij te werken, om pas daarna die data beschikbaar te stellen aan een analytische applicatie. Daarom wordt bij Complex Event Processing de werkwijze omgedraaid. In plaats van de data vast te leggen en daar vervolgens een analyse op los te laten, wordt nu een query gebouwd waar de data vervolgens doorheen kunnen stromen. Als we algoritmisch handelen in aandelen als voorbeeld nemen, dan zien we dat de banken die hiermee aan de slag zijn gegaan uitgaan van scenario's. Het is dus veelal niet zo dat de CEP-software zelf relaties en patronen moet zien te ontdekken.



Afbeelding 2. Werkwijze van een systeem voor Complex Event Processing.

Er wordt gewerkt met te verwachten patronen. Deze worden omgezet in query's. Een simpel voorbeeld: als het aandeel Shell omhoog gaat, mag verwacht worden dat ook aandelen BP stijgen. Dat biedt een kans en dient bijvoorbeeld tot de actie 'koop aandelen olie' te leiden. De query zal in dat geval dus zoeken naar het patroon waarbij event A (aandeel van Shell stijgt) én event B (BP-aandeel omhoog) zich voordoet, waarna een trigger volgt naar de gewenste actie.

Event streams opslaan

Scenario's zullen in de praktijk uiteraard veel complexer zijn. Neem bijvoorbeeld verschillen in noteringen tussen diverse beurzen. Hetzelfde aandeel kan om wat voor reden dan ook in Frankfurt een paar eurocent goedkoper zijn dan in Londen. Wie dat verschil op tijd ziet, kan daar direct actie op ondernemen. De query's die bij algoritmisch handelen worden gebruikt, kunnen daardoor uiterst complex zijn. Daarbij kan het handig of nuttig zijn om meerdere query's achter elkaar te plaatsen, waarbij het resultaat van de eerste query eventueeel ook als input voor de volgende kan worden gebruikt.

> Het vastleggen van de event streams biedt een aantal mogelijkheden

De data die van de diverse aandelenbeurzen binnenkomen, worden vervolgens door deze query's geleid. Maar wat gebeurt er hierna met deze 'event streams'? Opslaan is in principe niet nodig, zo leggen de deskundigen op dit terrein uit. De data hebben betrekking op events die al achter ons liggen en zullen in veel gevallen dus veel van hun waarde verloren hebben. Vaak zal dit uiteraard anders liggen. Neem maar weer het voorbeeld van algoritmisch handelen. Hier is het uiteraard noodzakelijk om een audit trail op te bouwen. In andere gevallen kan het voldoende zijn om bijvoorbeeld alleen die event streams vast te leggen waarin ook daadwerkelijk relevante patronen zijn gevonden.
Het vastleggen van de event streams biedt een aantal mogelijkheden. Allereerst dus het opbouwen van een audit trail. Wat gebeurde wanneer, hoe heeft de afdeling of de handelaar daarop gereageerd? Om vast te kunnen stellen of daarbij alle regels in acht zijn genomen moeten dus zowel de events als de analyses als de acties die daaruit voortkwamen worden vastgelegd.

Playback

Via dezelfde programmatuur kan vervolgens de klok letterlijk worden teruggedraaid en kunnen gebeurtenissen die zich eerder hebben voorgedaan opnieuw worden afgespeeld. Die 'playback'-functie kan meerdere doelen dienen. De query's zullen in de regel worden opgesteld door gespecialiseerde medewerkers. Zij zullen een grondige kennis moeten hebben van het proces dat wordt gevolgd. Aan de hand van event streams kunnen zij een voortschrijdend inzicht opbouwen. Ook voor nadere analyse van bepaalde situaties of problemen is het belangrijk om op de *real world* gegevens te kunnen terugvallen.
Overigens noemt men de database-technologie die wordt toegepast voor het vastleggen van event streams ook wel 'data stream management'. Het gaat vaak om *in-memory* opslag. Hierin worden zowel de ruwe basisgegevens vastgelegd als de afgeleide data. Met andere woorden: wie een playback doet, ziet allereerst een exacte kopie van de situatie die opnieuw bekeken moet worden. De events worden in dezelfde volgorde gepresenteerd als zij zich daarvoor in real-time voordeden, terwijl ook kan worden gezien welke patronen werden vastgesteld, welke acties een trigger kregen en dergelijke. Daarnaast is het mogelijk om analyses te doen op de vastgelegde data. Wat zou er bijvoorbeeld zijn gebeurd als een bepaalde query net even iets anders was samengesteld?
De business analisten die de query's ontwerpen, zullen in veel gevallen ook in staat zijn deze daadwerkelijk te bouwen. Dit heeft wel wat weg van scripting, waarbij gewerkt wordt met een bibliotheek vol standaard functies die op basis van allerhande conditionele voorwaarden aan elkaar worden gekoppeld.
Naarmate het gebruik van Complex Event Processing toeneemt, kan binnen de organisatie behoefte ontstaan aan eigen of bijvoorbeeld samengestelde functies. Die kunnen of door business analisten zelf of door IT-professionals worden samengesteld.
Bij een bedrijf als Progress – dat voor dit soort toepassingen de Apama software levert – noemt men dit soort extra functies 'smart blocks' die in één keer aan een query kunnen worden gekoppeld.

Output queue

Het is uiteraard van groot belang dat CEP-systemen met een groot aantal bronsystemen uit de voeten kunnen. Denk aan typische 'event sources' als de datastromen die van aandelenbeurzen worden ontvangen, maar bijvoorbeeld ook aan de beheer-software van RFID-readers. Ook e-mail moet als input kunnen worden gebruikt. Er zijn zelfs al voorbeelden van softwareprogramma's voor tekstanalyse die in staat zijn uit bijvoorbeeld persberichten van beursgenoteerde bedrijven koersrelevante informatie te halen en als input aan CEP-systemen door te geven. Ook klassieke databases kunnen als input gelden, evenals ERP-systemen en andere bedrijfsapplicaties. Uiteraard kan ook middleware als IBM MQSeries en dergelijke als bron van data gelden.
De output van dit soort systemen kan op meerdere manieren worden doorgegeven. Het zoeken naar patronen levert een zogeheten 'event output queue' op. Hierin staan triggers die naar andere programma's worden gestuurd. Deze kunnen als directe input voor een geautomatiseerde reactie worden gebruikt, naar kunnen bijvoorbeeld ook een dashboard voeden. Desnoods volgt er een trigger naar e-mail of sms-bericht.

Robbert Hoeffnagel is freelance journalist.