

**Java wordt op heel veel plaatsen gebruikt, maar sommige plaatsen spreken veel meer tot de verbeelding dan andere. Het CERN in Genève is een goed voorbeeld. Met behulp van Java worden pogingen ondernomen om de fundamentele vragen van het leven te beantwoorden, vragen als: waaruit is massa ontstaan en wat gebeurde er in de eerste paar milliseconden na de big bang.**

# Met Java terug naar de Big Bang

## Indrukwekkende Java-toepassingen: CERN

**E**en applicatie die ervoor zorgt dat deeltjes een snelheid van 99,999999% van de lichtsnelheid bereiken, zal in de strijd om de titel snelste Java-applicatie met zeer grote waarschijnlijkheid winnen. Ook verder spreken de Java-toepassingen bij CERN tot de verbeelding. In de fysica zijn fundamentele vragen nog steeds niet beantwoord. Wel hebben theoretische fysici theorieën ontwikkeld, maar deze wachten al vele decennia op machines die in staat zijn de omstandigheden te creëren, waarin de experimentele bevestigingen van de theorieën geleverd kan worden. Zelfs delen van de nu al bijna honderd jaar oude theorieën van Einstein wachten nog op experimentele bevestiging. De LHC van het CERN (large hadron collider, oftewel grote hadron botser) is zo'n machine. De deeltjesversneller is de opvolger van een minder krachtige versneller in dezelfde tunnel, de Large Electron-Positron Collider. Een hadron is overigens een gebonden staat van quarks. Quarks zijn op hun beurt weer subatomaire deeltjes, een proton bestaat bijvoorbeeld uit drie quarks.

Het principe van een deeltjesversneller is vrij eenvoudig: twee stralen met deeltjes worden vanuit een bestaande oudere en

kleinere versneller met een grote beginsnelheid op weg gestuurd en vervolgens over een traject van zevenentwintig kilometer steeds verder versneld met behulp van supergeleidende extreem krachtige magneten. De ene straal gaat rechtsom, de andere linksom. De op elkaar botsende deeltjes zijn gelijksoortig, hetzij protonen, hetzij lood-ionen. Bij hun reis door de LHC bevinden ze zich in omstandigheden die goed vergelijkbaar zijn met die in de ruimte: in een vacuüm bij 1,9 graden Kelvin (-271,25 °C). Supergeleidende magneten spelen daarbij een zeer belangrijke rol. Doordat ze bij zeer lage temperaturen geen weerstand vertonen (en dus geen warmte produceren) kunnen zeer grote hoeveelheden energie toegevoerd worden, waardoor ze extreem sterke magnetische velden kunnen opwekken. De besturing ervan is zeer complex en wordt in Java-applicaties geregeld. Bij de nadering van de deeltjes worden ze door de magneten aangetrokken, terwijl de magneten bij het passeren bliksemsnel van polariteit wisselen, zodat ze de deeltjes weer een duwtje in de rug kunnen geven. Iedere straal bestaat uit bijna 3000 pakketjes van deeltjes, die op hun beurt weer uit honderd miljard deeltjes bestaan. Omdat de deeltjes extreem klein zijn, is de kans

op een botsing echter erg klein. Van 200 miljard deeltjes die elkaar al spookrijdend ontmoeten, zullen er slechts twintig botsen. Er zullen echter dertig miljoen pakketjes van honderd miljard deeltjes per seconden passeren, zodat er nog steeds tot 600 miljoen botsingen per seconde zullen plaatsvinden.

Bij de botsingen vallen bekende deeltjes uiteen in kleinere deeltjes. Er zijn nogal wat deeltjes met de meest vreemde namen, maar er is ook een theorie die hun samenhang verklaart en dat is het zogenaamde standaardmodel (1977). Voor de experimentele bevestiging van die theorie is het nodig het zogenaamde Higgs Boson-deeltje waar te nemen. Dat is één van de doelen van de experimenten in de LHC. Onderzoek daarnaar is tegelijk onderzoek naar de vraag waaruit de wereld, de materie, nu eigenlijk bestaat als naar de vraag hoe de wereld ooit ontstaan is – of naar de vraag of hij ooit ontstaan is, want er zijn ook theorieën die uitgaan van een voortdurende afwisselingen van een imploderend en exploderend heelal. Van veel van die deeltjes die men met behulp van de LHC hoopt te kunnen waarnemen, wordt aangenomen dat die ontstaan zijn in de eerste milliseconden na de Big Bang. Het meest fascinerende van de experimenten is wel dat men niet weet wat ze precies op zullen leveren. Concrete vragen waarop men nu al antwoorden probeert te vinden zijn bijvoorbeeld: waarom is er zoveel materie en zo weinig antimaterie? Ook zijn er hypothesen over het bestaan van andere krachten dan die welke wij al kennen. De zwaartekracht met name, is een vrij zwakke kracht (gemeten aan de massa die eraan te pas komt) en zou wel eens een afspiegeling kunnen zijn van een sterkere kracht die wij tot nu toe nog niet waargenomen hebben. De zwaartekracht is overigens ook niet opgenomen in het standaardmodel. Uiteindelijk draait het bij CERN om het vinden van antwoorden op zeer fundamentele vragen als: waaruit is massa ontstaan, wat is de aard van zwarte materie en zijn er verborgen dimensies van de ruimte?

De functie van het CERN is vooral om experimentele data te leveren. Deze worden door 10.000 fysici over de gehele wereld geïnterpreteerd. In dat licht is ook het Grid van groot belang. (Zie kader). Het levert zowel de data als de rekenkracht aan fysici in de gehele wereld om de botsingen te kunnen analyseren en

## Panne, pech en winterslaap

De LHC wordt gedurende de koude periode van het jaar niet gebruikt. CERN heeft een afspraak met de energieleveranciers dat ze tegen zeer concurrerende prijzen energie kunnen afnemen, onder de voorwaarde dat ze in de periode waarin grote vraag is, niets afnemen. De maanden tussen november en maart is de tijd voor onderhoud, voor het aanpassen en testen van applicaties en zo meer. Dit jaar is dat bijzonder belangrijk, omdat bij de eerst keer dat de deeltjes door de gehele versneller verstuurd werden een incident optrad: door een ondeugdelijke elektrische verbinding tussen twee magneten ontstond een vlamboog en lekten duizenden liters vloeibaar helium weg. Om niet te zeer achter te raken op schema, overweegt men de testfase bij 5 TeV over te slaan, en begin 2009 direct op 7 TeV te gaan draaien.

nieuwe deeltjes te kunnen waarnemen. Het grid is in feite weer een zeer extreme toepassing van een idee dat binnen het CERN geboren werd: het internet. In 1989 schreef CERN-wetenschapper Tim Berners-Lee een voorstel om een gedistribueerd informatiesysteem voor het laboratorium te ontwikkelen. Binnen een paar maanden ontwikkelde hij daaruit de basis van internet zoals wij dat nu nog steeds gebruiken. Berners-Lee loopt nog steeds rond in het CERN. Hij zit echter niet meer in dezelfde kamer waar hij ooit de ideeën voor internet ontwikkelde. De kamer heeft echter een waardige nieuwe huurder: de man aan wie die dit James Gosling jaar de Duke uitreikte: Derek Mathieson. Mathieson bouwde – uiteraard met vele anderen – een groot deel van de Java-applicaties die

## Grid

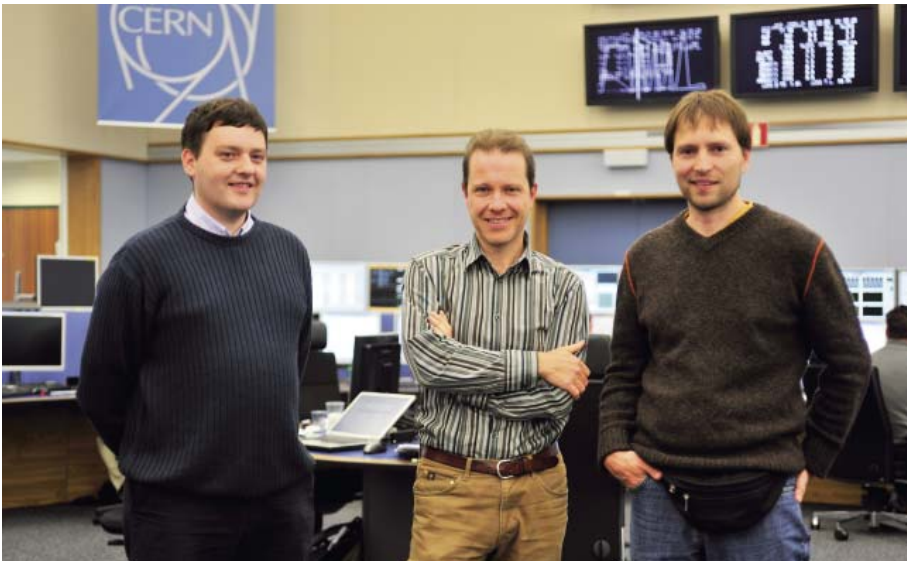
Bijna net zo indrukwekkend als de LHC zelf, is het LHC computing grid. In feite is het een enorme gedistribueerde computer met 100.000 processoren verdeeld over 140 instituten in 33 landen. Meer dan 700 fysici hebben daardoor bijna real time toegang tot de data van de experimenten en kunnen gebruik maken van de enorme rekenkracht om berekeningen uit te voeren met deze data. De datahoeveelheid is enorm: per jaar zal de LHC de komende tien tot vijftien jaar ongeveer 15 petabytes (15 miljoen gigabytes) produceren.

de werking van de LHC mogelijk maken. Daarbij hoort de Technical Infrastructure Monitoring application (is het toeval dat hij TIM heet?) die helemaal uit Java-code bestaat. Het is een Swing-applicatie die JMS gebruikt voor communicatie en Ilog Views voor de rendering. Het wordt gebruikt in de controleruimte om het gehele versnellercomplex en de bijhorende systemen te monitoren en te besturen. De bijbehorende Event viewer die werkt met Java 3D spreekt misschien wel het meest tot de verbeelding: hij maakt de sporen van de deeltjes zichtbaar.

## CERN

*CERN staat voor Conseil Européen pour la Recherche Nucléaire. Al heel snel is de naam veranderd in 'Organisation européenne pour la Recherche nucléaire, maar niemand minder dan Heisenberg verzond dat de afkorting ook kon blijven bestaan wanneer de naam veranderd zou zijn. Om die redenering te begrijpen moet je waarschijnlijk heel goed in kwantumfysica zijn.*

Concreter kun je de experimenten in de deeltjesversneller niet maken. Derek Mathieson ontving op JavaOne in 2008 uit naam van CERN een Duke voor de Java-applicaties binnen CERN. De Schot die nu al jaren met vrouw en kinderen in Genève woont, is er blij mee, alhoewel je soms de indruk krijgt dat hij het leuker vindt voor Java dan voor hemzelf. Samen met zijn collega's Vito Baggolini en Niall Stapley praatte hij met Javamagazine over het bijzondere en het gewone van hun Java-toepassingen. Mathieson begon bij CERN in 1989, nog als student informatica, in de accelerator en beams-afdeling. Na jaren ging hij naar de VS om daar aan voor andere deeltjesversneller Java-applicaties te bouwen. Na zijn terugkomst stortte hij zich op het ERP-systeem van CERN. Ook dat is geen simpele applicatie: het heeft een elektronisch document management systeem, dat volgens Mathieson in wezen alle business processen van de organisatie draait. Er zijn links naar twintigduizend leveranciers en meteen groot deel van hen wordt elektronisch documenten uitgewisseld. De applicatie is oorspronkelijk geschreven in C, daarna werd het C++ en in 1999 is hij herschreven in Java en sindsdien is het een Java-applicatie



Derek Mathieson, Vito Baggiolini en Niall Stapley in het control center van CERN

gebleven. Backend is een Oracle Database, Oracle 10. Er zijn 5000 concurrent users, met een totale user populatie van 11.000, gemiddeld 12 hits/seconde, 60.000 elektronische handtekeningen en 25.000 documenten per maand.

*Een Duke krijg je niet zomaar, het is meestal het einde van een lang proces. Hoe is aan het begin van dat proces de beslissing tot stand gekomen om voor Java te kiezen?*

Mathieson: "Ik was vanaf het begin in Java geïnteresseerd. We hebben de alpha API gebruikt om er applets mee te schrijven voor sommige interactieve elementen. We hadden webpagina's die op basis van CGI-scripts werden geserveerd, maar we wilden encryption validation aan de client-kant hebben, dus dat hebben we in Java gedaan. Er zijn meerdere redenen waarom ik van Java houd. CERN heeft een vrij hoge wisseling van personeel. We hebben veel studenten, mensen die hier één tot maximaal drie jaar werken, een doctoraalscriptie schrijven en dan weer weggaan. Daarom hadden we een applicatieontwikkelomgeving nodig die mensen snel konden leren, een gestructureerd framework waarbinnen ze zich kunnen bewegen en waarmee het gemakkelijk is om correcte code te schrijven. Het probleem met C en C++ is dat het juist gemakkelijk is om er slechte code mee te schrijven. Het is moeilijk om slechte code in Java te schrijven. Dingen als multiple inheritance en pointers zitten niet in Java en dat maakt het een iets cleanere taal wat het gemakkelijker maakt voor ons. Het

tweede punt is dat het overgrote merendeel van de mensen die hier willen komen werken, Java kennen. Het is de meest gebruikte taal op de universiteiten."

Stapley: "Dat was misschien vroeger waar maar tegenwoordig is het iets anders. C++ heeft meer dan 70 keywords. Maar nieuwe ontwikkelingen in Java maken de taal ook gecompliceerder."

Mathieson: "Ja, generics en closures en zo maken het concept van de taal wel gecompliceerder, maar ik zie het als iets waar je gewoon rekening mee moet houden. Je moet er naar kijken en je de vraag stellen of je het nodig hebt. Soms heeft het echt zin, en je krijgt dan code die veel beter te lezen en te onderhouden is en dan is het

geweldig. Er zitten een paar heel goede kanten aan generics."

*Wat doen de Java-applicaties in de accelerator and beams department nu precies?*  
Baggiolini: "We zorgen voor de accelerator en de stralen voor het experiment. De botsing genereert uiteindelijk via de detectors zeer veel data en dat is waar het grid van pas komt."

Mathieson: "Er bestaat een wijdverbreid misverstand over de functie van CERN. Wij verzorgen de faciliteiten aan de fysici voor de experimenten. De kleinste afdeling binnen CERN is de theoretische fysica-afdeling. Het grote aantal fysici hier bij CERN is er om de faciliteiten voor de experimenten te verzorgen."

Stapley: "De A&B afdeling verzorgt niet alleen de stralen voor de LHC, maar ook voor andere experimenten. We beginnen met het genereren van stralen, in feite pakketten van stralen, die op veel verschillende plaatsen in één keer afgeleverd worden. In die zin is het werk wat wij doen vrij complex. Op iedere plaats in de keten gaat het om het afleveren van verschillende soorten stralen, tien in totaal. De afdeling is verantwoordelijk voor het controleren en besturen van de versneller, die in feite bestaat uit afzonderlijke versnellers, en het versneller-complex. Ze hebben allemaal een operator nodig en controls, modellen om ze te besturen. De operators zitten voor schermen waarop Java GUI's te zien zijn die de versneller besturen."

Het controlesysteem werkt met een drie



Voor het congresgebouw van CERN staat een supergeleidende magneet. De magneten zijn de motor achter de LHC.

tier architectuur. Allereerst is er het low level-gedeelte dat dichtbij de echte hardware zit en draait op gespecialiseerde frontend computers. Dan is er een gedeelte wat op een hoger niveau speelt, en dat al geschreven is in Java, de resource tier, die services aanbiedt. Ten slotte zijn er de GUI's die met die services verbonden zijn.

Baggiolini: "In het algemeen zorgen wij voor de service die API's aanbiedt voor andere mensen die vaak geen professionele Java-programmeurs, maar fysici. Ze gebruiken onze API's om GUI's te schrijven, want dit zijn de mensen die uiteindelijk die GUI's in de controlroom zullen gebruiken." (Zie foto control room)

*Dat klinkt wel heel erg agile, om de eindgebruikers zelf te laten programmeren.*

Baggiolini: "Ze hebben in feite een dubbelrol. Hun belangrijkste functie is die van een bestuurder van de machine, ze moeten weten hoe die machine werkt en de deeltjesversneller aansturen. Daarnaast hebben ze een twee job waaraan ze minder dan de helft van hun tijd aan besteden, in de tijd dat de versneller stilgelegd is. Sommigen doen Java, andere dingen."

Baggiolini: "Wij zijn meer verantwoordelijk voor de middle tier service, de architectuur, het uitkiezen van tools, ontwikkelingen van de service en ter beschikking stellen van de API aan de programmeurs en uiteraard het onderhoud. Maar de operators, fysici, kunnen allemaal Java schrijven, al zijn sommigen meer ervaren dan anderen. We zullen hun code reviewen en ze helpen er het beste uit te halen. Er zijn voor hen ook trainingen maar je moet het toch in de praktijk leren. In de ideale situatie – en dat heb ik in mijn projecten geprobeerd te doen - geef je ze hands on training en kijk je naar hun code. Je vertelt ze waarom doe je dit niet doet of dat juist wel, probeert een profiler-tool te gebruiken om te zien of je een memory leak hebt, dat soort dingen."

*Welke frameworks zijn gebruikt? Heeft u eigen oplossingen moeten ontwikkelen of zijn er dingen naar voren gekomen die u echt gemist heeft in het Java-platform?*

Baggiolini: "We gebruiken het Spring framework om onze applicaties te bouwen, maar verder gebruiken we relatief weinig third party libraries. We hebben vrij strikte constraints vanwege de real

time aspecten van de versneller. Zo hebben we eigenlijk nooit de juiste middleware gevonden om heel veel C en C++ data aan te leveren. We hebben daarvoor dus zelf een oplossing moeten schrijven, bovenop Corba. Ik denk dat we een probleem hadden met EJB's, we begonnen met EJB 1.1, en zijn in alle vallen gelopen waarin je kunt lopen. Uiteindelijk hebben we het de rug toegekeerd en hebben ons op Spring gericht. Verder zijn we geen problemen tegengekomen. We hebben een omgeving opgezet, zodat je voorbeeldcode kunt bekijken, code van jongere ontwikkelaars wordt gereviewed zodat men er zeker van kan zijn dat die aan een aantal conventies voldoen, we doen Unit tests en we hebben een release systeem dat heel mooi geïntegreerd is in wat we doen."

Stapley: "Wat ik wel gemist heb, is object versioning. Dat zit niet in Java, zeker als er zoveel dependencies zijn kan dat een probleem zijn. Als er een nieuwe release is van een project, hoe ga je dan om met de andere projecten die daarvan afhankelijk zijn. Net als veel andere organisaties hebben we uiteindelijk ons eigen release-systeem gebouwd."

*CERN schrijft geen methodologie voor. Niettemin gebruiken beide teams Scrum.*

Stapley: "De taal alleen lost niet het merendeel van de problemen op. In dit geval, zijn goede procedures en goede communicatie belangrijker gebleken, dat kan een taal niet vervangen, je hebt een goed team nodig dat elkaar begrijpt."

Baggiolini: "Je moet een gemeenschappelijke manier opbouwen om de dingen te doen en dan kun proberen het gebruik van tools als statische analyse-tools af te dwingen, om de meest voor de hand liggende bugs te vinden. Mensen die informatica gestudeerd hebben, zijn geneigd om unit-tests te doen, anderen minder."

Stapley: "Er zijn nog meer verschillen tussen commerciële en industriële software en de academische wereld van informatici. CERN heeft het voordeel dat het academici aantrekt, daarom moet je meer dan ergens anders de software engineering-kant aanmoedigen. In academia wordt wel samengewerkt bijvoorbeeld, maar veel minder openlijk dan je nodig hebt voor software engineering. In academia is er toch vaak een soort eigenaarschap van het gebied waaraan ze werken. Bij ons blijft code twintig tot dertig jaar bestaan, we gebruiken nog

steeds code uit de jaren tachtig, dus we willen het eigenaarschap van code ontmoedigen."

## Realtime

Stapley: "De versneller doet dingen in hardware, het control system, het hoger deel erna, hoeft niet realtime te zijn. Het realtime gedeelte is PLC-gebaseerd en systeem-gebaseerd. Het is het laagste niveau van de drie lagen. Wij hebben het langzame systeem, wij werken eerder in orde van grootte van seconden dan van milliseconde. Het laagste niveau is realtime, en daar gebruiken we ook realtime operating systems als Lynx OS, Linux en PLC's. Het is goed dat het gesepareerd is."

*Maar moet het systeem op uw niveau dan niet precies zijn?*

Stapley: "Ja en nee, omdat in feite wanneer we de versneller in een andere cyclus brengen, dan kan dat als een transactioneel iets gedaan worden. Dingen worden van tevoren ingesteld en geladen en achteraf getriggerd. Wanneer het eenmaal in die laagste systemen is, dan werkt het op zichzelf. Het safety critical gedeelte wordt in hardware gedaan. Soms zijn er in de hardware ook helemaal geen talen. Op het gebied van veiligheid wil je misschien wel helemaal niet op software vertrouwen, wil je gewoon koper en een aan/uit-knop." Na afloop van het gesprek volgde een bezoek aan de controlroom, het commando-centrum van de LHC. Mathieson en zijn collega's gaven met veel enthousiasme een rondleiding. Met een zeker ontzag spraken ze over de experimenten in de LHC. Hoe prominent Java hier ook in de vorm van GUI's aanwezig was, je merkte dat het hier toch eigenlijk om iets anders ging, om leptonen, bosonen en andere onvoorstelbaar kleine deeltjes en om de grote vragen die daarmee samenhangen. Het bijzondere was misschien wel dat Java hier even onzichtbaar kon worden, juist omdat de applicaties hun gecompliceerde taak zo geruisloos en foutloos klaarden. Zoiets is misschien wel meer dan één Duke waard. «

## Belastinggeld

Nederland is een van de lidstaten van het CERN en doet ook mee aan het project. Het verzorgt 4,5%, oftewel 27,8 miljoen euro, van het jaarlijkse inkomen van CERN.