De e-scientist. Universele wetenschapper of specialist?

Mevrouw de Rector Magnificus,
Mijnheer de Decaan,
Leden van het curatorium van de leerstoel ‘‘Efficient computing for eScience’’,
Bestuursleden van de Stichting Nederlands eScience centrum,
Geachte dames en heren,
Lieve Lucas en Kayleigh,

In de komende drie kwartier wil ik graag wat vertellen over mijn onderzoek. Ik begin meteen met een filmpje.

1. eScience

In 2017 zat ik bij deze Formule 1 race op het Belgische Spa Francorchamps, samen met 265.000 andere fans. Waarom laat ik dit nu zien? Na de race werd bekend dat de oorzaak van het uitvallen van Max Verstappen een softwareprobleem in de Renault motor was. Er was dus niets kapot, het was gewoon een bug.

Dat is interessant. Wat kunnen we hier nu van leren? Het eerste punt is dat software overal is, en van groot belang is voor onze maatschappij, ook al is het

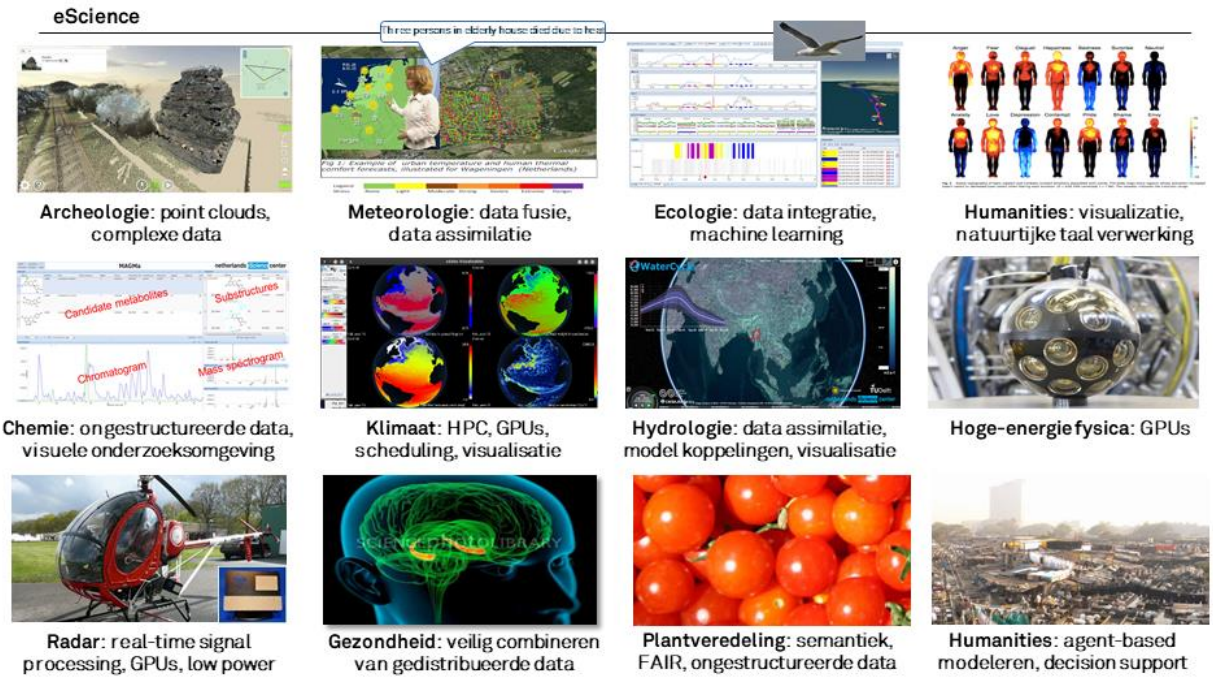
vaak niet zichtbaar. Software zit dus in auto's, maar ook in vliegtuigen, wasmachines, verlichting, televisies, en, waar mijn interesse ligt: in wetenschappelijke instrumenten. Daar wil ik het vandaag over hebben.

Het tweede punt is dat het toch wel opmerkelijk is dat dit mis gaat. De belangen in de Formule 1 zijn groot. In totaal kijken er jaar wereldwijd 600 miljoen kijkers naar de Formule 1. Red Bull investeert ongeveer 350 miljoen euro per jaar in het team. De Renault motor kost rond de 50 miljoen naar het schijnt. Dus, zelfs als de belangen en het budget zo groot zijn, lukt het niet om software te maken die werkt. Kennelijk is het moeilijk om goede software te schrijven. Helaas is er geen enkele reden om aan te nemen dat het met software in de wetenschap beter gesteld is. Sterker nog, er is maar weinig budget, grote tijds- en prestatiedruk, en de software wordt vaak geschreven door wetenschappers zonder een formele training in software engineering. "What could possibly go wrong?"

Een derde observatie is dat het hier gaat om een complex systeem. Er zijn tientallen losse componenten met aparte computers in een auto, die met elkaar communiceren via een netwerk. Dat bemoeilijkt het programmeren van het systeem. Sterker nog, het blijft niet beperkt tot de auto zelf, er wordt in de Formule 1 live telemetrie data naar de pit muur gestuurd voor analyse. Vanaf die plek wordt het via snelle verbindingen naar de fabriek in Engeland gestuurd, waar

het nog verder geanalyseerd wordt, en onder andere wordt gebruikt om live allerlei simulaties en modellen te draaien, waarvan de output weer wordt teruggekoppeld naar de pit muur, waar dan strategische beslissingen worden genomen. We noemen zo'n systeem met componenten op verschillende fysieke plekken die samenwerken een gedistribueerd systeem. Ook in de wetenschap zien we meer en meer gedistribueerde systemen, en die zijn dus nog moeilijker te programmeren dan gecentraliseerde systemen. We hebben dus een probleem.

Mijn onderzoek houdt zich bezig met grootschalige wetenschappelijke software, en in het bijzonder software van grootschalige gedistribueerde systemen en instrumenten.



1.1. Inleiding eScience: ICT in de wetenschap

Wetenschap wordt steeds complexer en heeft meer en complexere ICT nodig. Het toepassen en onderzoeken van innovatieve complexe ICT in andere wetenschappelijke disciplines noemen we eScience. Een echt kenmerk van eScience is dat het onderzoek plaatsvindt in nauwe samenwerkingsverbanden tussen informatici en wetenschappers in de disciplines. Zoals u kunt zien in deze voorbeelden bestrijkt eScience alle wetenschapsgebieden, van de levenswetenschappen tot klimaatonderzoek en hoge-energie fysica. De

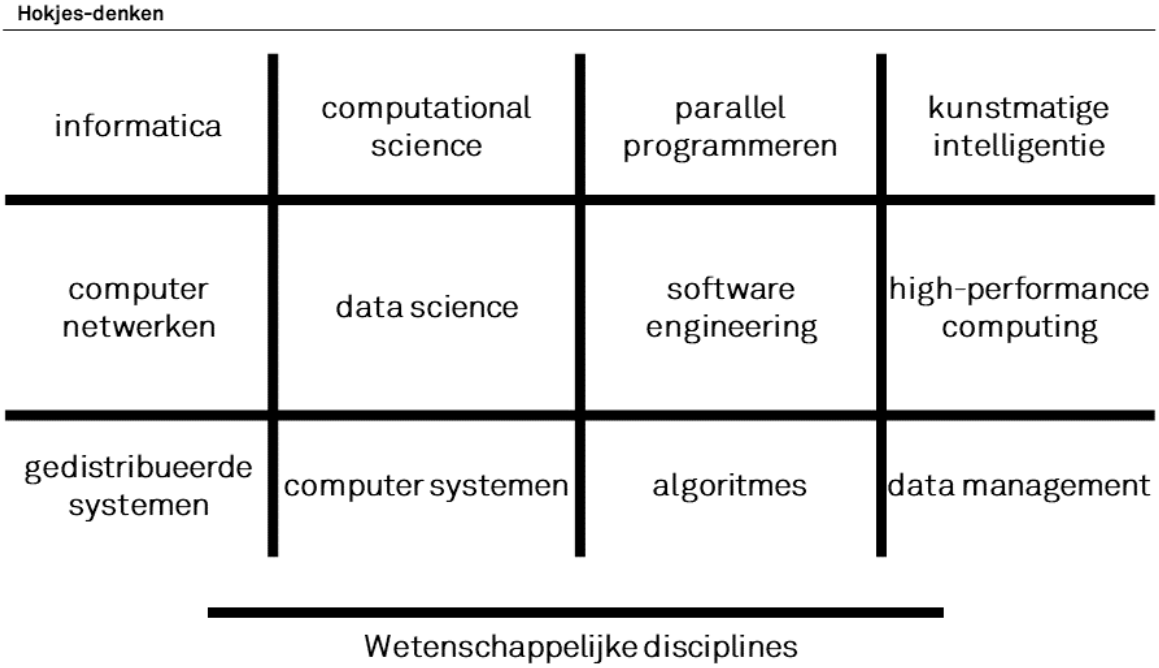
voorbeelden betreffen concrete projecten die wij bij het Nederlands eScience centrum uitvoeren of uitgevoerd hebben.

Ook technisch is eScience enorm breed. Afhankelijk van de wetenschappelijke vraag en het vakgebied kan het gaan om grootschalige simulaties, het koppelen van modellen, het integreren van observaties en modellen, het koppelen van verschillende databronnen, het zoeken in ongestructureerde data, visualisatie, machine learning en nog veel meer.

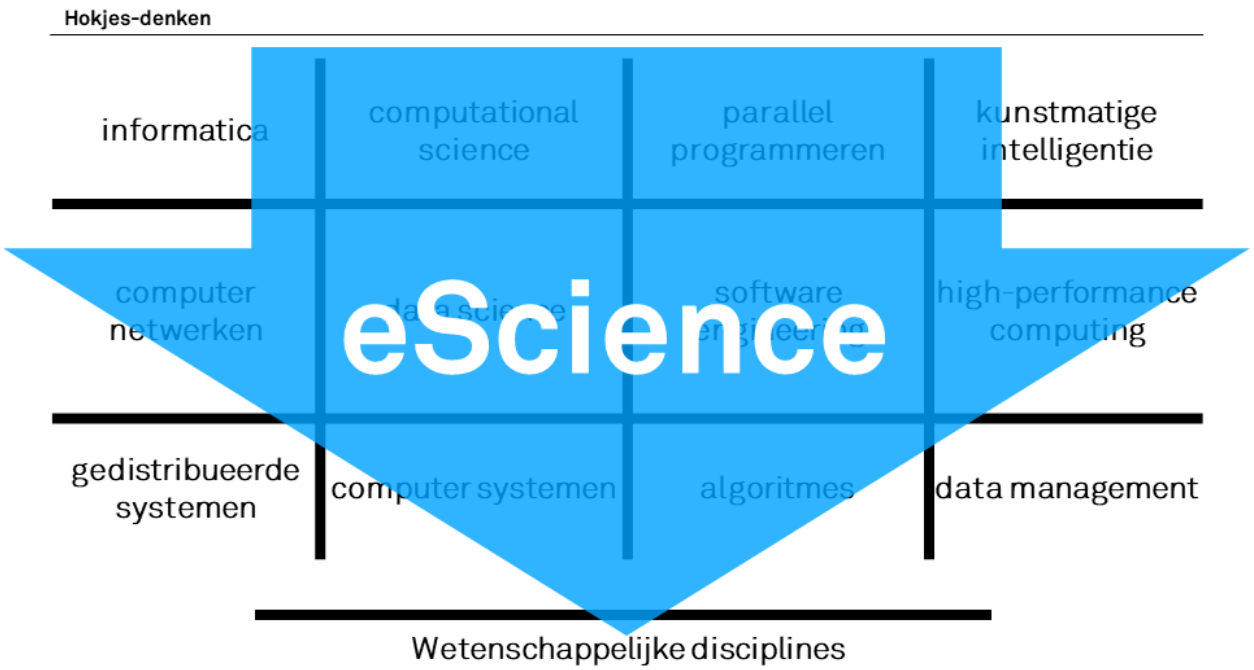
Het is enorm uitdagend om state-of-the-art informaticaonderzoek toe te passen om lastige wetenschappelijke problemen aan te pakken. Juist omdat dit zo moeilijk is levert eScience ook enorm veel inspiratie op voor nieuw en innovatief informaticaonderzoek. Daarnaast is voor mij is een belangrijk aspect van eScience het hebben van impact met dat informaticaonderzoek. Dan gaat het niet alleen om impact in de wetenschap, maar ook maatschappelijke impact, bijvoorbeeld in de gezondheidszorg, veiligheid, of juist in het bedrijfsleven. Als je kijkt naar mijn publicatielijst dan zie je dat ook terug. Naast publicaties in de informatica heb ik ook publicaties in de archeologie, flight-safety, massa spectrometrie, signaalverwerking, en radioastronomie. Ongebruikelijk voor een informaticus, maar voor een e-scientist doodnormaal. Je wilt immers impact maken in de disciplines, dus je publiceert ook in die disciplines.

Tegelijkertijd werkt het ook omgekeerd, en kunnen we als e-scientists met kennis van nieuwe technische mogelijkheden wetenschappers in de andere disciplines juist uitdagen om hun onderzoeksvragen aan te scherpen en nog groter te denken. Deze wisselwerking is heel spannend.

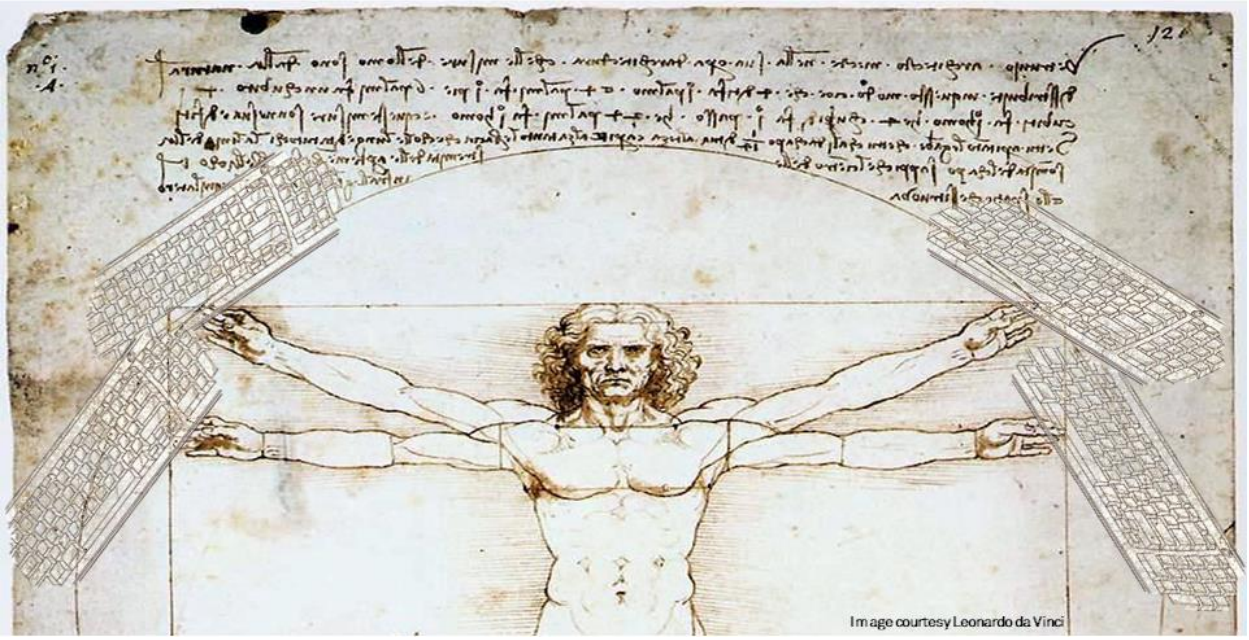
1.2 Hokjes-denken & generalisatie versus specialisatie



Wetenschap kenmerkt zich vaak door verregaande specialisatie. Dat geldt zeker ook de informatica. Dat is ook nodig, omdat we steeds voortbouwen op bestaande kennis. Het risico van deze specialisatie is wel dat we te veel in hokjes gaan denken. Binnen de informatica zijn er veel vakgebieden die redelijk onafhankelijk opereren. Naar mijn mening zijn we wat te ver doorgeschoten in het specialiseren, en er is als gevolg daarvan te weinig kruisbestuiving tussen de hokjes.



Het eScience vakgebied gaat juist over het doorbreken van de hokjes, we doen namelijk wat nodig is om de wetenschappelijke vragen in de disciplines op te lossen. Vrijwel altijd is daar expertise uit meerdere hokjes voor nodig. Niet het informaticaonderzoek, maar de onderzoeksvragen uit de wetenschappelijke disciplines zijn leidend.



Het is dus voor een e-scientist van groot belang om een brede technische en discipline-overstijgende blik te hebben. Een soort Uomo universale, een universele wetenschapper dus. De werkelijkheid is uiteraard genuanceerder dan dit. De praktijk wijst uit dat het goed werkt om eScience teams op te zetten, waarin eScience specialisten en generalisten samen werken met informatici en domein experts. Van nature is eScience dus erg gericht op samenwerking in plaats van competitie.

Toch heb je uiteraard ook de competitie van de wetenschappelijke methode nodig om de kwaliteitsslag te maken: eScience papers zijn uiteraard peer-reviewed, en financiering wordt onder andere door het Nederlands eScience centrum en steeds meer ook door andere financiers via zeer competitieve open calls uitgezet. Nederland is internationaal vrij uniek, met een nationaal eScience centrum dat zelf veel eScience projecten financiert en ook uitvoert. Er is veel meer vraag naar eScience dan we op dit moment aankunnen, dus de competitie is groot.

De laatste jaren is er dan ook een trend waar er veel lokale eScience, data science, en ICT support groepen worden opgezet bij de lokale campussen van de universiteiten en onderzoeksinstellingen. Ook aan de Universiteit van Amsterdam is dat actueel. Het is belangrijk voor de efficiëntie en het succes van deze

zogenaamde ‘‘digital competence centers’’ om kennis te delen en van elkaar te leren. De technologische en discipline-specifieke uitdagingen zijn te groot voor elk individueel initiatief. Ook op deze schaal draait het dus om samenwerking, en niet om competitie. Naast een brede technische en discipline-overstijgende blik is samenwerkingsgerichtheid is dus ook een belangrijke eigenschap van de e-scientist.

De menselijke maat

deVolkskrant

Oktober 2016

Help, de wetenschapper verzuipt!

De hoeveelheid data die de wetenschap produceert, neemt elk jaar met eenderde toe. Hoe voorkomen wetenschappers dat zij verdrinken in de datazee?

Door **Martijn van Calmthout** en **Bard van de Weijer**

Illustratie **Thijs Balder**

Ook bij big data komt het uiteindelijk neer op intermenselijke communicatie

Rob van Nieuwpoort
bijzonder hoogleraar efficient computing

scoop altijd opnieuw metingen laten doen, de meeste sterren veranderen niet zo snel. Hier wordt zo veel data vergaard, dat je dingen moet weggooien, anders stik je in je gegevens. 'Alleen weet je ook nu nooit zeker of je de juiste gegevens weglaat.' Betere instrumenten die meer data opleveren, hoeven hierdoor niet per se tot betere data te leiden, zegt Van Nieuwpoort.

eScience probeert wetenschappers te ondersteunen bij het verkrijgen van goede data, en bij de analyse daarvan. Dat gebeurt onder meer door 'gewone' wetenschappers te koppelen aan computerexperts. Een bioloog kan precies duidelijk maken welke gegevens hij nodig heeft en welke weg kunnen, zegt Van Nieuwpoort. De computerwetenschapper weet hoe dat moet. Resultaat: betere gegevens, is het idee. 'Uiteindelijk komt het dus toch neer op intermenselijke communicatie, ook bij big data', aldus Van Nieuwpoort.

Loopt de wetenschap tegen de grenzen aan? 'Het is een beetje van alle tijden', relativeert sterrenkundige Huub Röttgering. 'Zeker in de sterrenkunde. Zodra er een nieuw apparaat komt, krab je je even op je kop en moet je bedenken hoe je al die nieuwe gegevens gaat bolwerken.'

'Eigenlijk', zegt Röttgering, 'hebben we gewoon een tekort aan data.' Sterrenkundigen willen liever nog veel meer, omdat ze dan verder kunnen kijken, en in meer detail. 'We willen dieper. Als je met een factor honderd kan inzoomen op een object is beter.' De in aanbouw zijnde radiotelescoop Square Kilometre Array kan aan deze zucht naar meer voldoen. Al zal zijn komst betekenen dat de berg ruis nog immenser zal zijn. 'Maar', zegt Röttgering, 'we komen er wel uit.' ●

RUIMTETELESCOOP GAIA

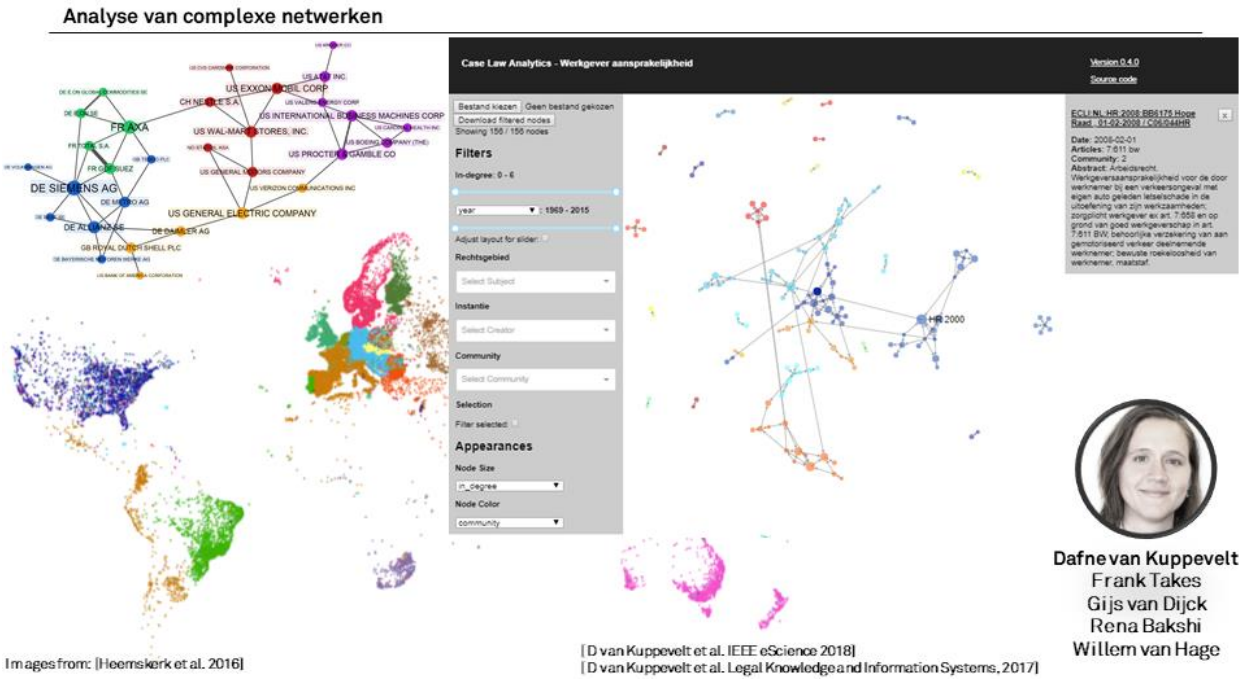
De Europese ruimtetelescoop Gaia, die bijna drie jaar geleden werd gelanceerd om de locatie en de beweging van een miljard sterren in het melkwegstelsel in kaart te brengen, is ook een grootleverancier van wetenschappelijke bulkdata. Hoewel de hoeveelheid ruwe gegevens die de satelliet naar de aarde stuurt nog wel meevalt. 'Die past bij wijze van spreken op mijn laptop', zegt Anthony Brown, de Leidse sterrenkundige die verantwoordelijk is voor de gegevensverwerking van het project. Zo'n 40 gigabyte per dag komt er binnen, zo'n 73 terabyte in vijf jaar.

Pas als de ruwe data worden bewerkt, zwelt de gegevensberg op, tot vermoedelijk een petabyte als de missie is afgerond. Vooral in het ordenen (zodanig in databases rangschikken dat de software ermee uit de voeten kan) en het controleren of alle gegevens juist zijn ontvangen, gaat veel werk zitten, zegt Brown.

Maar de grootste bottleneck is het transport van al die bits. De gegevens komen binnen in Madrid, waarna ze worden verstuurd naar een van de vijf aangesloten academische datacentra in Europa voor verwerking. Het verschepen van al die bytes gaat in de praktijk soms trager dan gedacht, zegt Brown.

Doordat de capaciteit van de datalijnen van de universiteiten beperkt is, en ze ook gebruikt worden door andere onderzoekers, ontstaan weleens opstoppinkjes op de digitale snelweg.

Het wetenschappelijk onderzoek loopt er geen vertraging door op, al had Brown het liefst gezien dat alle dataverwerking in één instituut was gedaan. Dan hadden er geen gegevensmeren heen en weer te hoeven gepompt en konden onderzoekers bij elkaar zitten, wat ook de menselijke communicatie ten goede komt. 'Maar dit is zowel politiek als in de praktijk lastig uitvoerbaar', zegt Brown.



Images from: [Heemskerk et al. 2016]

[D van Kuppevelt et al. IEEE eScience 2018]
[D van Kuppevelt et al. Legal Knowledge and Information Systems, 2017]

1.3. De menselijke maat

Veel van mijn onderzoek draait om grootschaligheid. Van wetenschappelijke instrumenten, van computersystemen, en van software. Het is daarom ook interessant om vanuit het eScience perspectief te kijken naar schaal. Niet vanuit het oogpunt van de traditionele technische definities, maar juist vanuit de domeinwetenschapper geredeneerd. Het gaat dus om de menselijke maat.

Een bekend onderzoeksthema van de laatste jaren is “Big Data”. Veel technische definities van wat Big Data nu eigenlijk is noemen een aantal V’s. De bekendste zijn “Volume”, “Velocity”, en “Variety”. De definitie die ik zelf altijd gebruik is een typische eScience definitie die uitgaat van de wetenschapper. En dat is: “data is Big Data als een onderzoeker of een discipline er niet meer mee weet om te gaan”. Dat is voor elke discipline dus expliciet anders. Voor de een gaat dit om vele petabytes (zoals in de radio astronomie en hoge-energie fysica), maar voor de andere is het een gigabyte. Het hangt er helemaal vanaf wat voor soort data het is, hoe complex en heterogeen het is, en vooral ook wat voor gereedschap beschikbaar is om informatie uit de data te destilleren. Het hangt dus voor een groot deel af van de beschikbare software, en niet van de data zelf. Mijn stelling is dus dat Big data gaat over software, en niet om data.

In 2016 heb ik meegewerkt aan een artikel over Big Data in de wetenschap in de Volkskrant. Dat artikel heet dan ook “help de wetenschapper verzuipt”. Er staat dus niet “help, de data explodeert”. Dit geeft goed het eScience perspectief aan: denk vanuit de wetenschapper en de wetenschappelijke discipline, niet vanuit de technologie.

Analoog aan data kun je ook voor software spreken over schaal. Wat is grootschalige software, of een grootschalig softwareproject? Je kunt dit uitdrukken in technische termen, zoals het aantal regels code, of allerlei metrieken over de complexiteit van software. Maar, tijdens de IEEE eScience conferentie, die wij hier in Amsterdam vorige jaar georganiseerd hebben, hoorde ik een aardige definitie, gegeven door Reed Milewicz. Hei zei “Grootschalige software is software waar er niet langer 1 enkele persoon is die weet hoe alles werkt”. Zodra je deze grens overgaat wordt het nog belangrijker om na te denken over softwarekwaliteit, documentatie, en zaken als versie beheer, en testen.

1.3.1. Voorbeeld: analyse van complexe netwerken (Dafne van Kuppevelt)

Een mooi voorbeeld van de menselijke maat in eScience is het werk van Dafne van Kuppevelt, die promotieonderzoek doet bij het Nederlands eScience centrum. Dafne werkt in verschillende wetenschappelijke disciplines aan de analyse van complexe netwerken. Met professor Gijs van Dijck uit Maastricht werken we aan juridische netwerken van rechtszaken. Het is belangrijk voor het beantwoorden van de juridische onderzoeksvragen om te weten welke rechtszaken welke jurisprudentie citeren, en in welke context. Het mooie is dat Dafne soortgelijke analysetechnieken toepast in haar werk met Frank Takes, dat gaat over de analyse van complexe netwerken tussen de besturen van grote bedrijven. In beide gevallen zijn de netweken in termen van bytes of aantal knopen niet enorm groot, maar wel complex, en het is erg lastig om data uit verschillende bronnen te koppelen, en de interessante verbanden binnen de netwerken te identificeren.

Er zijn veel algoritmen om netwerkanalyse te doen, en recent zijn grootschalige rekenmethodes voor dit soort netwerken ook een actieve onderzoeksrichting. Echter, de stap maken van het resultaat van een algoritme, naar de betekenis hiervan in de discipline is erg lastig, en meestal subjectief. Er is nog echt een interpretatiestap nodig, waar je vaak gebruikt maakt van visualisatie, en interactieve grafische interfaces, en die ziet u hier. Omdat het subjectief is, is dit

interpretatie aspect in de informatica vaak onderbelicht, maar voor eScience juist cruciaal.

1.4. Big Data, inleiding radio astronomie

Dan een heel ander voorbeeld. Een van de meest inspirerende toepassingen van eScience voor mij persoonlijk zit in de radioastronomie. Ik ben vanaf 2007 bezig met eScience in dit vakgebied. In deze discipline gaat het juist wel om het volume van de data. LOFAR, of de “Low Frequency Array”, de telescoop die u hier ziet, genereert ongeveer 2 *terabits* aan data per seconde, dat is dezelfde orde van grootte als het internetverkeer dat wordt afgehandeld door de Amsterdam Internet Exchange. Deze datastroom is zo groot, dat het niet mogelijk is om het op te slaan. Daarom is de dataverwerking een real-time systeem: je moet de datastroom bijhouden, anders verlies je gegevens.

LOFAR is een gedistribueerd sensor netwerk. Het instrument bestaat uit ongeveer 100.000 relatief eenvoudige antennes, die samen een virtuele telescoop vormen. Het centrale punt van de telescoop staat in Exloo in Drenthe.

In totaal zijn er 38 Nederlandse antennevelden, en daarnaast ook 19 internationale velden. Dit is dus met recht een grootschalig gedistribueerd systeem. Dat is belangrijk, omdat een grotere afstand tussen de antennevelden betekent dat je hogere resolutie beelden kunt maken. De antennedata wordt deels al verwerkt in het veld, en daarna naar een supercomputer in Groningen gestuurd, waar de verdere dataverwerking plaatsvindt. Alle verwerking, analyse en het vertalen van de radiosignalen in beelden gebeurt in software. Dit type telescoop wordt daarom ook wel een softwaretelescoop genoemd.

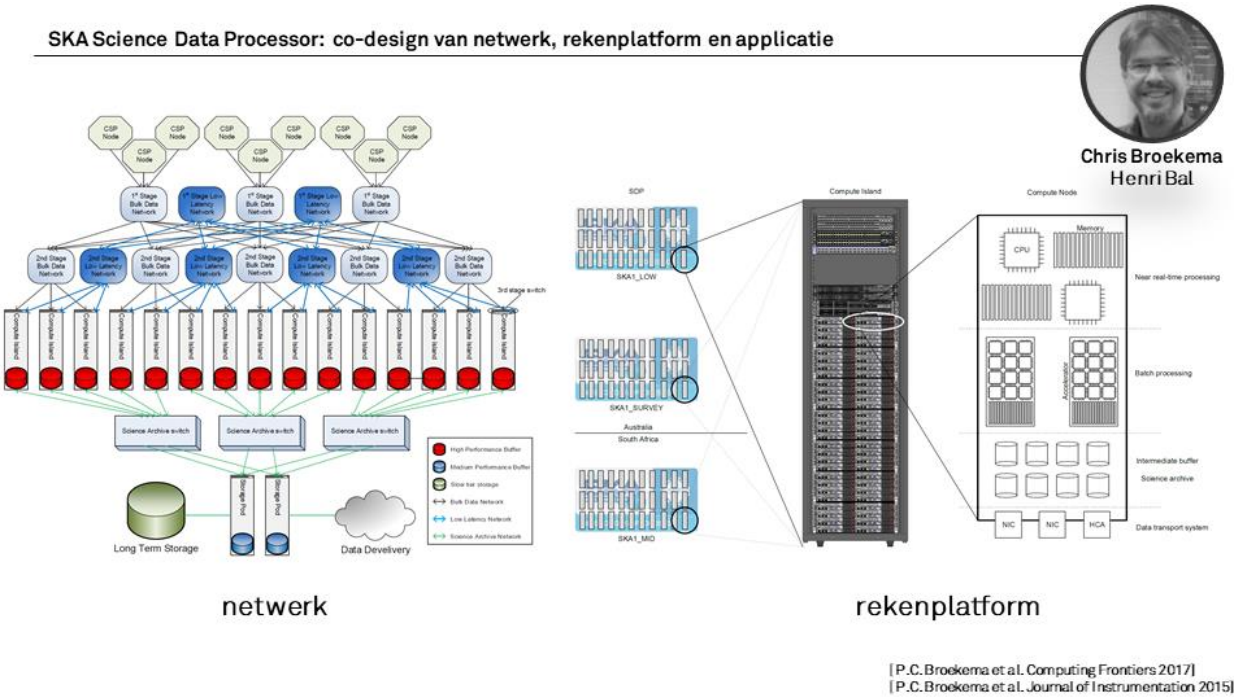
Er zit veel informaticaonderzoek en innovatie in deze telescopen. Zo was de supercomputer in Groningen in eerste instantie een IBM Blue Gene. Ik heb, samen met onder andere John Romein bij ASTRON onderzocht of we de dataverwerking ook konden doen met zogenaamde “accelerators”. In dit geval is er uiteindelijk gekozen voor grafische kaarten, ook wel GPUs genoemd. Deze GPUs zijn eigenlijk ontworpen voor computerspellen die tegenwoordig zeer snel fotorealistische beelden moeten genereren, waar veel rekenkracht voor nodig is. Wij “misbruiken” die grote rekenkracht voor de dataverwerking. Deze oplossing inmiddels geïmplementeerd door ASTRON, en deze is sneller, goedkoper, en energiezuiniger. Dit onderzoek stamt al uit 2007, en inmiddels gebruiken we deze GPUs ook in heel veel andere eScience toepassingen.

Het is enorm spannend om te zien hoeveel innovatie er zit in de instrumentatie zelf. Dit is APERTIF, een recente upgrade van de Westerbork telescoop. Hier wordt een “Focal Plane Array” gebruikt in het brandpunt van de schotel. Je kunt dit een beetje vergelijken met de sensor in een modern fototoestel die meerdere megapixels tegelijk vastlegt. Door deze techniek zijn de gevoeligheid en het blikveld van de telescoop enorm vergroot. Je kunt dus sneller de hemel afzoeken. Een van de astronomen die APERTIF ontworpen heeft en ook gebruikt voor zijn onderzoek naar pulsars en fast radio bursts is Joeri van Leeuwen. Samen met hem hebben we gekeken naar de grootschalige dataverwerking. Het bleek noodzakelijk om de volledige software pipeline compleet opnieuw te ontwerpen om zeer efficiënt op GPUs te draaien. Een voormalig PhD student van mij, Alessio Sclocco, heeft hier een grote bijdrage aan geleverd tijdens en na zijn promotieonderzoek.

LOFAR en APERTIF zijn gebouwd door ASTRON, het Nederlands instituut voor radioastronomie, en ik denk dat we enorm trots kunnen zijn dat wij deze instrumenten van absolute wereldklasse in Nederland kunnen ontwerpen, bouwen en exploiteren.

De volgende generatie telescoop die momenteel ontworpen en gebouwd wordt is de SKA, of “Square Kilometre Array”, een combinatie van schotels en verschillende antennes. Nederland is een belangrijke partner in dit project. SKA is een gedistribueerd sensor netwerk op enorme schaal, want de antennes staan hier zelfs op twee continenten, namelijk in Zuid Afrika en West Australië. De SKA is een echt exascale instrument. Exascale slaat op de grootte van de datastroom. SKA gaat exabytes aan data genereren. In totaal is dat, per seconde, meer data dan het totale, wereldwijde internetverkeer bij elkaar. Exascale betekent ook dat er 10 tot de 18^{e} berekeningen per seconde nodig zijn om deze data te verwerken.

Door de dataexplosie moet een nog groter deel van de dataverwerking in real-time op streaming data gaan werken, terwijl dat vroeger achteraf gebeurde. Met Thijs van den Berg doen we onderzoek naar wat ervoor nodig is om dit te bewerkstelligen. Soms lukt het met verbeteringen van klassieke algoritmiek, soms moeten nieuwe algoritmes ontworpen worden, en soms moeten klassieke algoritmes vervangen we door een benadering, bijvoorbeeld door het gebruik van deep learning. Hier kom ik straks nog op terug.

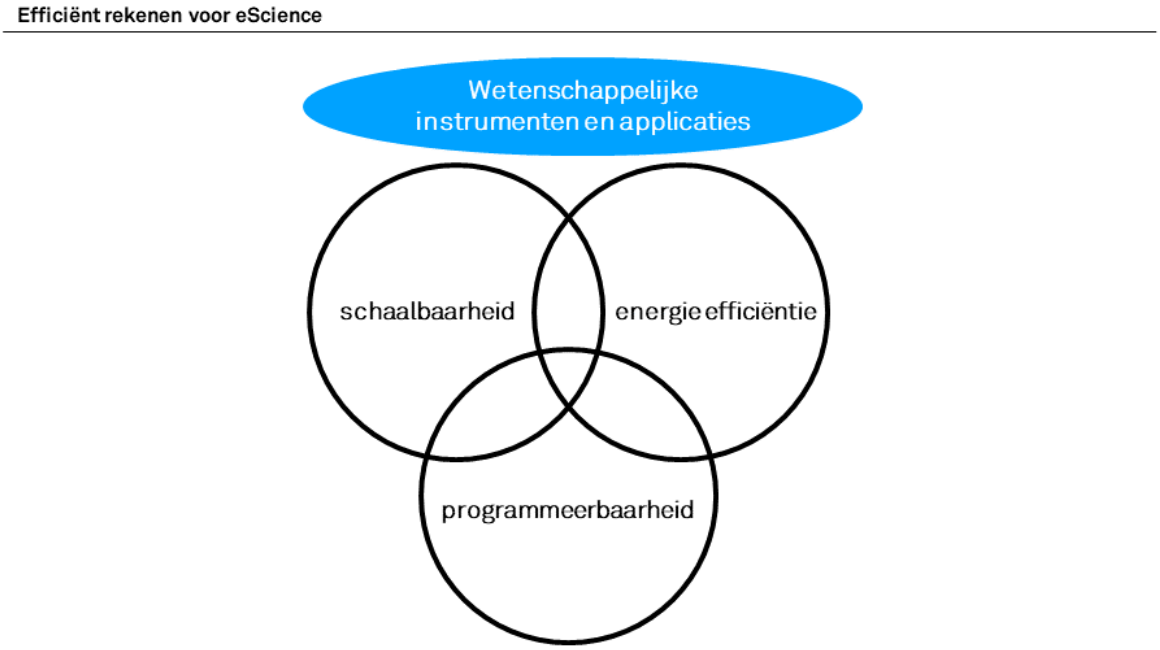


1.5. Ontwerp van SKA

Chris Broekema, een onderzoeker bij ASTRON die ik samen met Henri Bal van de VU begeleid in zijn promotietraject, heeft onderzocht hoe je het platform voor de dataverwerking van de SKA kunt ontwerpen. Het is essentieel om te onderzoeken welke berekeningen je op welke plek doet, met zo min mogelijk data verplaatsing, en zo min mogelijk energieverbruik. Welke hardware platformen selecteer je, en hoe kom je tot die keuze? Met Chris hebben we onderzocht of we moderne programmeerbare netwerken kunnen gebruiken, waar er dus echt intelligentie in het netwerk zelf zit.

Een van de conclusies is dat je rekenen en datatransport niet afzonderlijk moet zien, maar juist het instrument, de netwerken voor het datatransport, en het data-analyse platform inclusief de software stack integraal moet bekijken. Dit noemen we ook wel co-design. Door de co-design methode, en door het platform geheel programmeerbaar te maken in software, nu inclusief het netwerk dus, kunnen we de telescoop efficiënter en flexibeler maken. Co-design blijkt ook in de literatuur inmiddels een belangrijk thema te worden voor onderzoek naar exascale toepassingen.

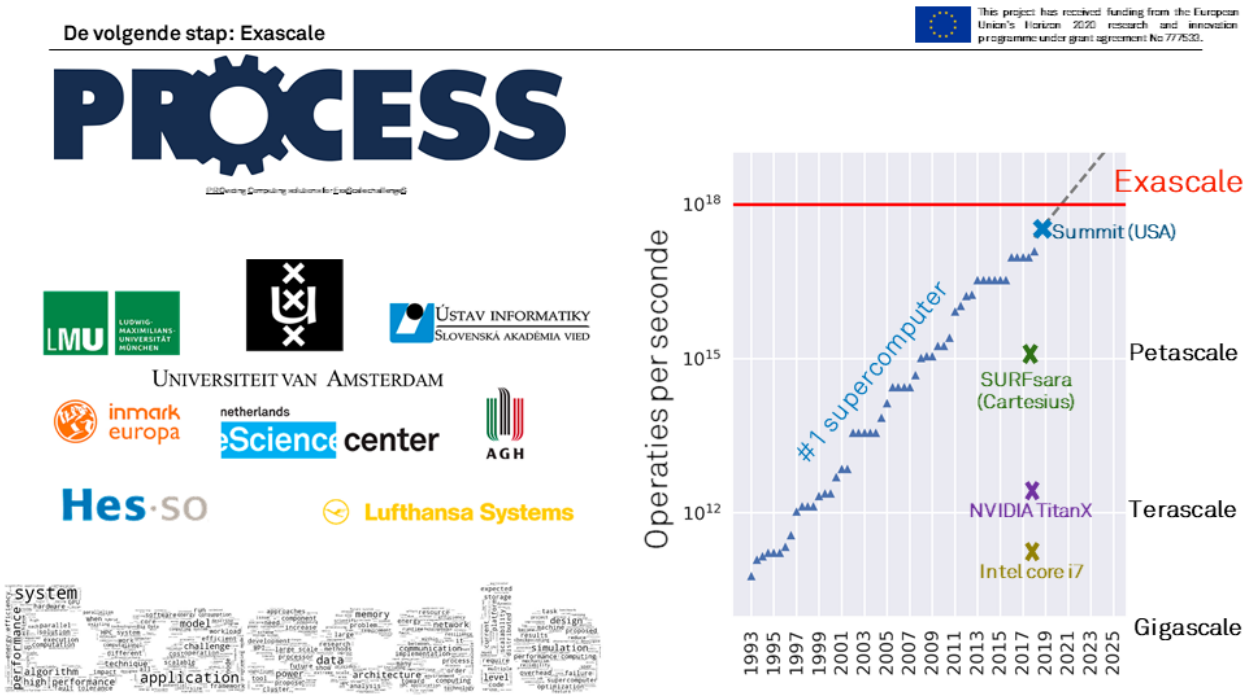
2. Grootschalig efficiënt rekenen



Deze afbeelding laat op een heel hoog niveau de thema's van mijn onderzoek zien. Mijn leerstoel heeft de titel "Efficient Computing for eScience". Wat eScience precies is heb ik in het voorafgaande behandeld. Nu wil ik dieper ingaan op het tweede deel: Efficiënt computing, oftewel efficiënt rekenen. Efficiënt rekenen heeft drie verschillende aspecten. Efficiëntie op grootschalige systemen, de schaalbaarheid dus; energie efficiëntie, en de efficiëntie van het programmeerproces zelf. Op elk van deze punten zal ik nu dieper ingaan.

2.1 Architecturen & hiërarchische systemen

Maar voor we het kunnen hebben over efficiëntie moeten we eerst kijken naar de structuur van de computersystemen we willen gebruiken.

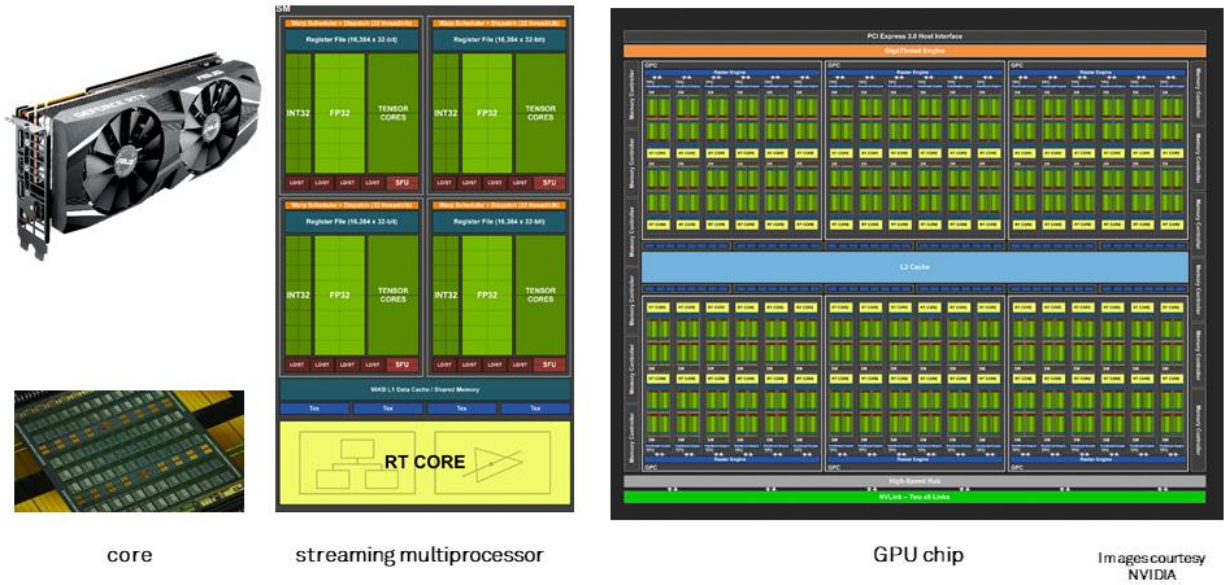


2.1.1 Exascale

Deze grafiek laat de snelheid van computersystemen over de tijd zien. De verticale as is het aantal rekenoperaties per seconde, de horizontale as is de tijd. De blauwe rechte lijn geeft de snelheid van de snelste supercomputer ter wereld op dat moment aan. Let op: de verticale schaal is logaritmisch. Elke verticale stap is een factor 1000 sneller. De groei is dus exponentieel. In 1 plaatje ziet u hier een normale processor (gigascale), een GPU (terascale), en de snelste supercomputers ter wereld (petascale).

Onze nationale supercomputer die bij SURFsara staat, de Cartesius, kan ongeveer 2 peta operaties per seconde leveren (2 peta-ops). Op dit moment staat het snelste systeem ter wereld bij het Oak Ridge National Lab in de Verenigde Staten, en dit systeem levert 200 peta-ops, ongeveer 100x sneller dan de Nederlandse Cartesius. Men verwacht rond 2021 de eerste exascale systemen in Amerika en China. In Europa verwachten we rond 2023 twee exascale systemen te bouwen.

In het Europese project PROCESS, waar zowel het Nederlands eScience centrum als de UvA partner zijn, ontwikkelen wij nieuwe software infrastructuur die het mogelijk maakt om dit soort grootschalige systemen efficiënt te gebruiken.



2.1.2. Hiërarchische systemen

Een mobiele telefoon heeft tegenwoordig al 8 cores of rekenkernen. Moderne high-end PCs hebben er al 16 of meer. Ik noemde eerder al dat wij veel rekenen op grafische kaarten of GPUs. Die GPUs maken gebruik van massief parallelisme, en hebben soms al meer dan 5000 cores, in 1 chip. De structuur van een GPU is niet plat, maar hiërarchisch. Een GPU chip is opgebouwd uit een aantal streaming multiprocessors. Elk daarvan heeft weer een groot aantal cores. Je moet hier als programmeur rekening mee houden: cores binnen streaming multiprocessor kunnen met elkaar communiceren, cores die in verschillende streaming multiprocessors zitten niet. Er is dus een interne hiërarchische structuur die bepalend is voor hoe je zo'n apparaat programmeert.

Systemen worden steeds hiërarchischer en heterogener

Images courtesy Intel, NVIDIA

In een compute node zitten over het algemeen zowel een snelle general-purpose processor, en een aantal GPUs. Er zijn nu twee dingen gebeurd. Het systeem is nu heterogeen geworden, er zijn twee verschillende processorarchitecturen, en er is nog een niveau van parallelisme toegevoegd. Meerdere GPUs kunnen in parallel rekenen.

Systemen worden steeds hiërarchischer en heterogener

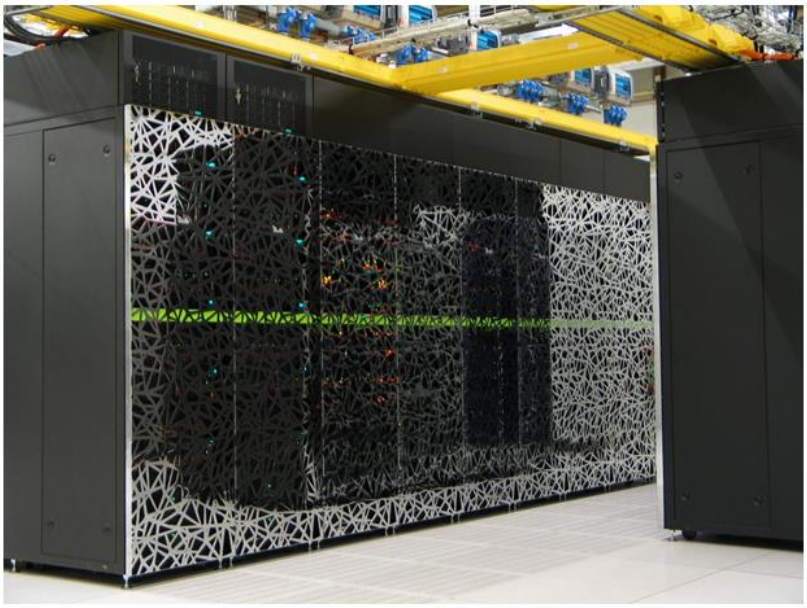


Image courtesy SURFsara

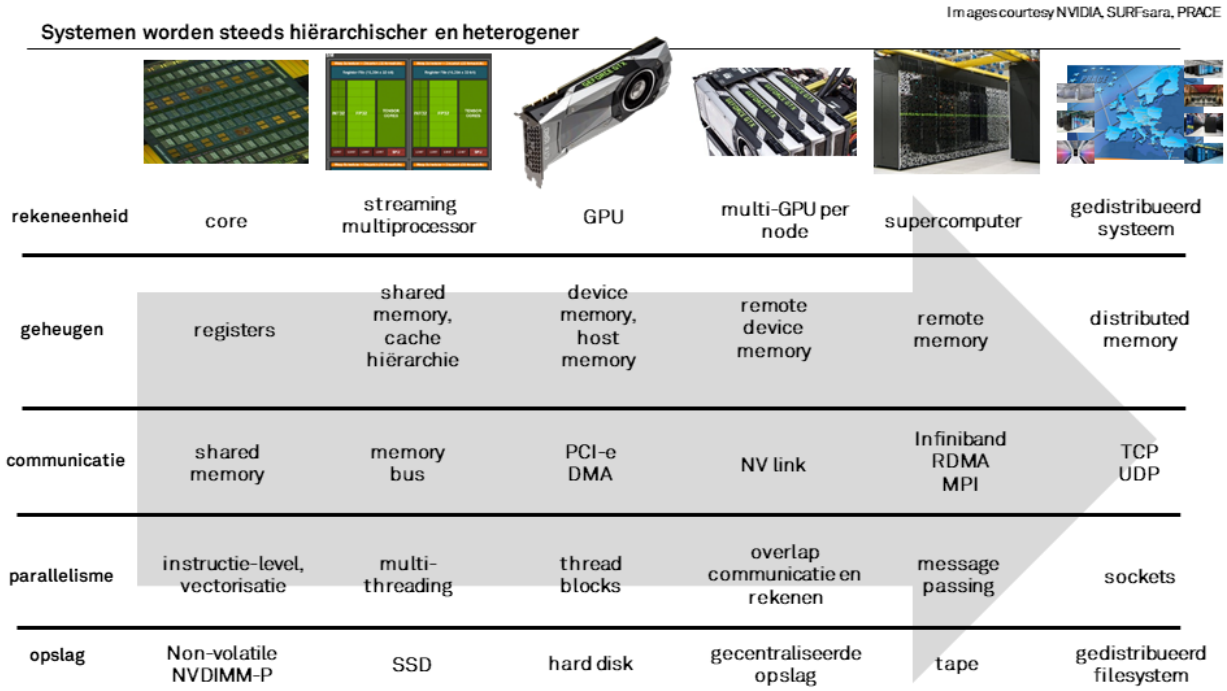
Een supercomputer bestaat uit een groot aantal van deze compute nodes, verbonden door een snel netwerk. Hier ziet u de Cartesius, de nationale supercomputer die bij SURFsara staat, en die op deze manier is opgebouwd.

Systemen worden steeds hiërarchischer en heterogener

Image courtesy PRACE

In sommige gevallen gebruiken de grootschalige wetenschappelijke applicaties waarin ik geïnteresseerd ben zelfs meerdere supercomputers die verbonden zijn

door snelle glasvezelverbindingen tegelijk. Voorbeelden zijn de SKA-telescoop, of gekoppelde simulatiecodes, bijvoorbeeld voor klimaatonderzoek.



De rekeneenheden in exascale systemen zitten dus in een hiërarchische structuur. Hetzelfde punt kun je ook maken voor geheugen, communicatie en parallelisme. Ik verwacht dat dit in de toekomst ook meer en meer zal gaan gelden voor dataopslag. Dat is samengevat in deze tabel. Van links naar rechts wordt het systeem steeds grootschaliger. U hoeft alle technische termen niet te lezen of te begrijpen, het gaat erom dat je enorm veel verschillende technologieën moet kunnen gebruiken en combineren om efficiënt te zijn.

Dus wat gebeurt er nu eigenlijk als we richting exascale systemen gaan?

Ten eerste: systemen worden extreem parallel en hiërarchisch.

Ten tweede: Systemen worden heterogeen, er zijn steeds meer accelerators zoals GPUs voor snel rekenen, speciale processoren voor deep learning, en FPGAs als meer flexibiliteit nodig is.

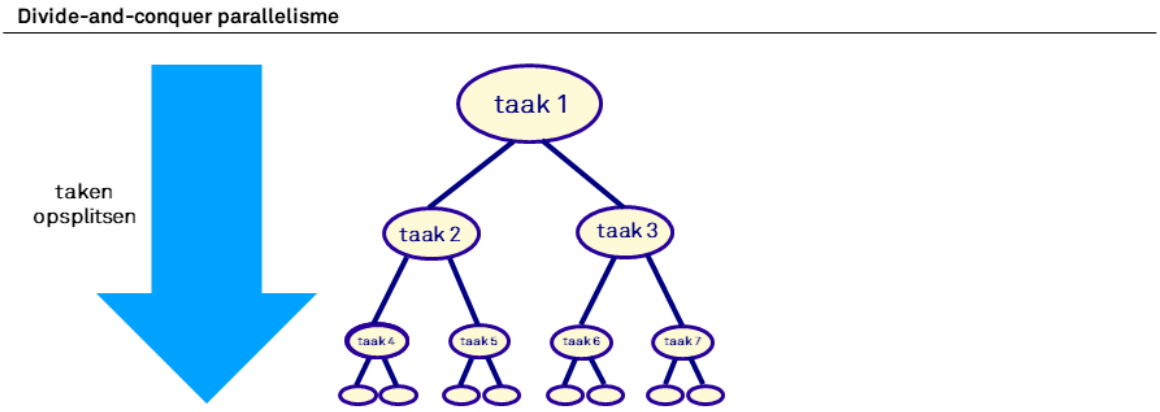
Het derde punt is dat er fouten optreden. Deze systemen zijn enorm complex en grootschalig. Als je kijkt naar de "mean time between failures" van componenten, dan kun je uitrekenen dat er altijd wel iets kapot is in het systeem. Je moet dus inherent rekening houden met fouten.

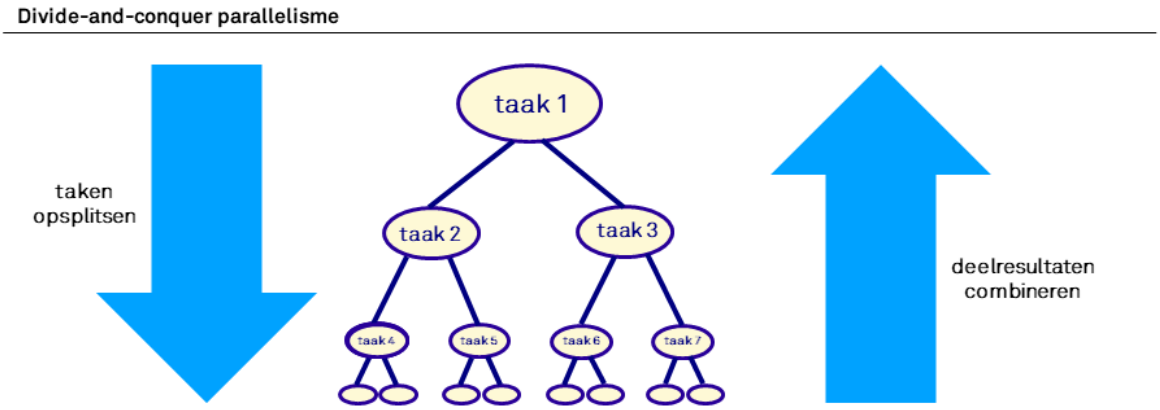
Het laatste punt is dat er veel meer flexibiliteit nodig is, door de heterogeniteit, door het gebruik van clouds en virtualisatie waarbij je de hardware vaak niet voor jezelf hebt, en door de fouten die onverwacht optreden.

Al deze complexiteit bemoeilijkt de programmeerbaarheid enorm.

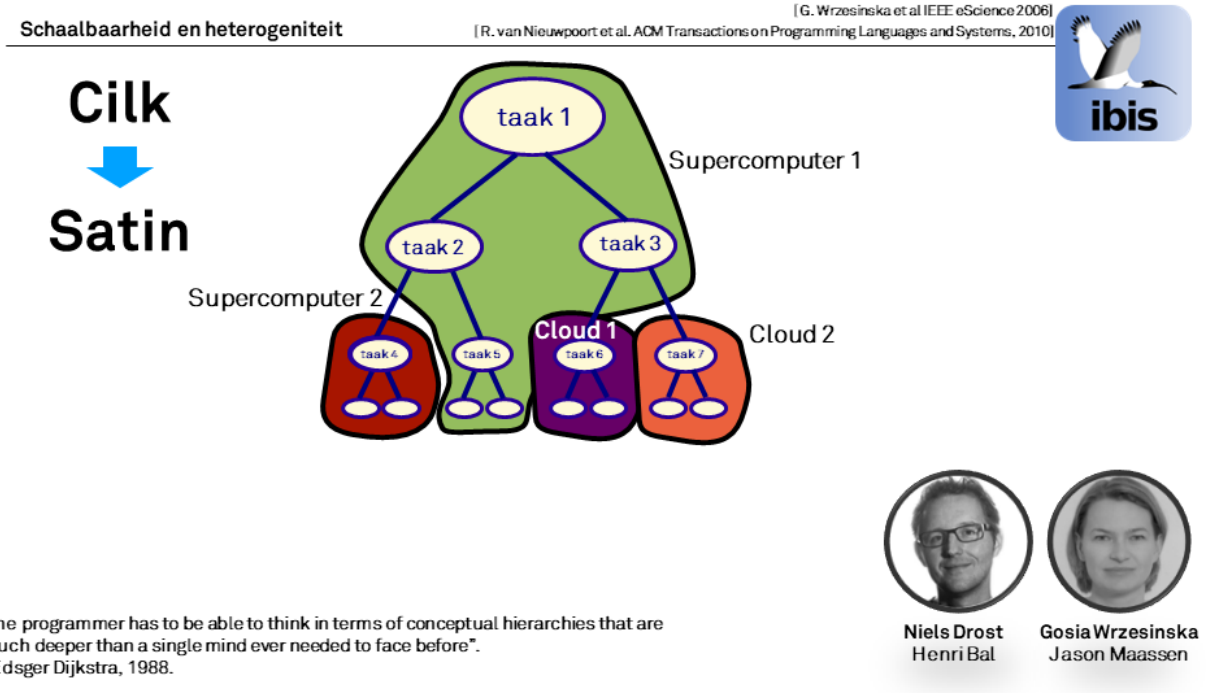
2.2 Parallel rekenen en heterogeniteit

Ik onderzoek programmeermethodes die zelf inherent hiërarchisch zijn. Het is mogelijk om met de hand hiërarchische optimalisaties te introduceren op applicatieniveau, in communicatiebibliotheken zoals MPI, of in hoog niveau programmeermodellen. Het bekendste voorbeeld van een inherent hiërarchisch model is divide-and-conquer.





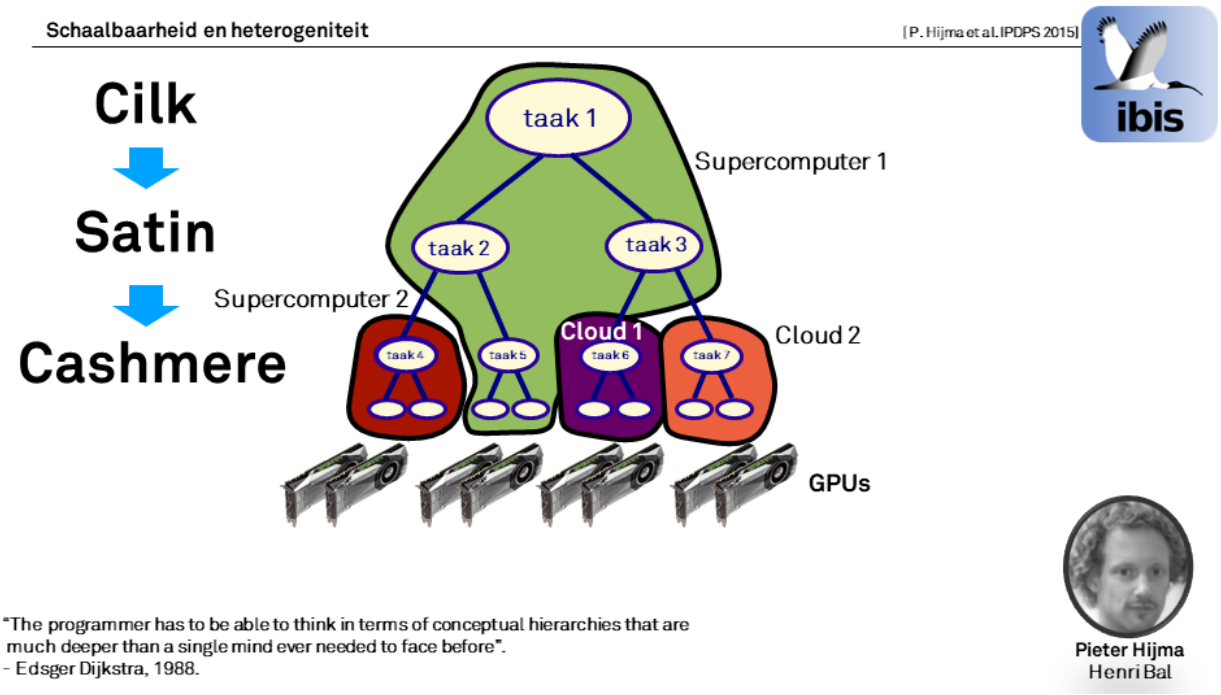
Bij divide-and-conquer splits je een probleem op in een aantal deelproblemen. Vervolgens splits je elk deelprobleem weer verder op. Dit blijf je recursief doen tot je een heel klein probleem overhoudt, dat eenvoudig in korte tijd op 1 processor op te lossen is.



Vervolgens worden alle deelresultaten weer stapsgewijs gecombineerd totdat het totale probleem opgelost is. Dit model volgt een boomstructuur, en is dus

inherent hiërarchisch. Mijn visie is dat zo'n hiërarchisch programmeermodel goed af te beelden is op hiërarchische systemen. Ons onderzoek laat ook zien dat dit inderdaad het geval is.

Met het Cilk project, uitgevoerd bij het Amerikaanse MIT, heeft men aangetoond dat divide-and-conquer toepassingen zeer efficiënt kunnen draaien op kleinschalige shared memory systemen. Tijdens mijn promotieonderzoek heb ik een opvolger van dit Cilk systeem gemaakt: Satin. Satin draait divide-and-conquer toepassingen efficiënt op grootschalige systemen met duizenden cores zelfs wanneer die verspreid zijn over verschillende continenten. Hiervoor moesten we nieuwe slimme algoritmes ontwerpen voor de werkverdeling. Met Gosia Wrzesinska, Jason Maassen en Niels Drost hebben we ervoor gezorgd dat ons model ook foutbestendig is. Zowel de algoritmes voor de werkverdeling als de foutbestendigheid blijken we heel efficiënt te kunnen implementeren door handig gebruik te maken van de hiërarchische structuren van het programmeermodel en het computersysteem.



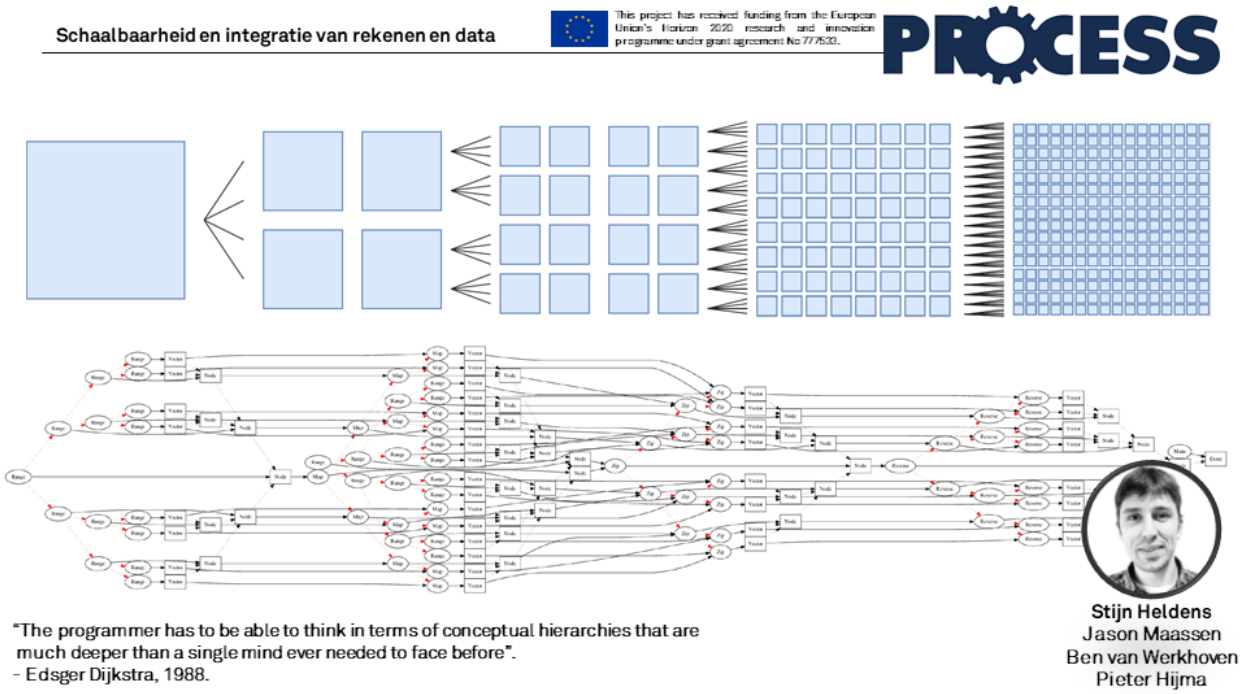
Pieter Hijma heeft Satin vervolgens uitgebreid met een nieuwe programmeermethode om accelerators te programmeren. Het resulterende systeem heet Cashmere. Het eindresultaat is dus dat we grootschalige wetenschappelijke applicaties kunnen draaien op wereldwijde hiërarchische en

heterogene systemen, zelfs als er fouten in het systeem optreden. De programmeur merkt vrijwel niets van al deze complexiteit.

Er doen dus al drie generaties van promovendi onderzoek met dit systeem. Een van de redenen dat we over een lange tijd zo succesvol zijn geweest met dit platform is dat we expliciet goed na hebben gedacht over het ontwerp, alles zeer modulair hebben opgezet, en dat we continu investeren in de softwarekwaliteit.

2.2.1. Nu: compute en data (Stijn)

In parallel programmeren onderzoek werken we vaak met benchmarks en vereenvoudigde applicaties om onze hypotheses en systemen te testen. Dit werkt heel erg goed voor rekenintensieve applicaties. Helaas gaat dit in de praktijk soms verkeerd bij echte wetenschappelijke applicaties, die over het algemeen juist erg data intensief zijn. Data verplaatsen is duur en gebruikt ook nog eens erg veel energie. Je moet dataopslag en transport dus meenemen in het ontwerp van algoritmes en systemen. Het hokjes denken waar ik het in het begin over had, en onze informatica-centrische blik, waar we de applicaties te veel vereenvoudigd hebben om te kunnen redeneren over het systeem, leidt dus soms tot blinde vlekken.

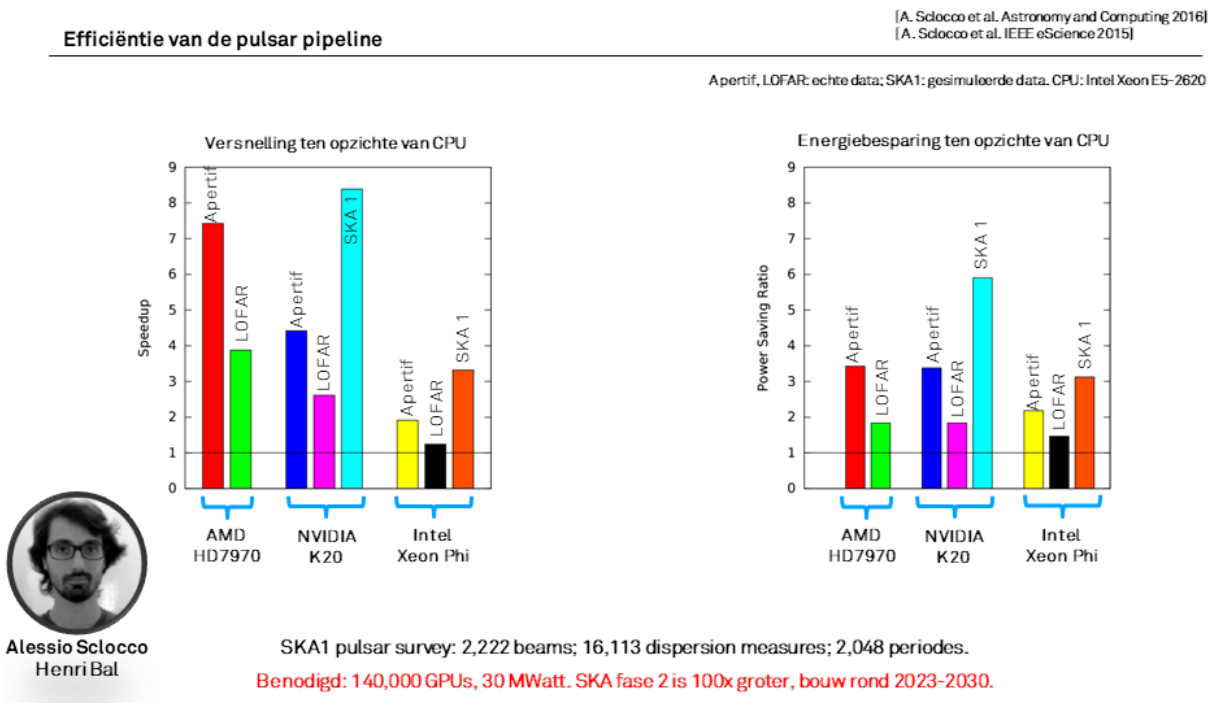


In het Europese project "PROCESS" werken we met Stijn Heldens aan modellen

die wel hiërarchisch zijn, en wel uitgaan van divide-and-conquer, maar die opslag en data lokaliteit wel mee nemen. Het verschil is dat we niet de taken opsplitsen in stukken, maar juist de data. Het systeem kan dan automatische afwegingen maken of het de data verplaatst naar plekken waar rekenkracht beschikbaar is, of juist andersom. In dat geval laat het systeem de data juist staan, en brengt het de berekeningen naar de data. Ons onderzoek moet uitwijzen welke aanpak in welk geval efficiënter is.

2.3 Energie efficiëntie

Het volgende aspect van efficiënt rekenen dat ik onderzoek is de energie efficiëntie. Voor grootschalige instrumenten en simulatiecodes is het energieverbruik inmiddels een belangrijke beperkende factor. De afgelopen jaren heb ik onder andere met Alessio Sclocco en John Romein onderzoek gedaan naar de energie efficiëntie van radioastronomie pipelines. Het gaat er dan om dat je zoveel mogelijk operaties per Joule kunt uitvoeren.



Een van de technieken de we gebruiken is een combinatie van run-time code generatie en auto-tuning. Met run-time code generatie genereren we de code pas vlak voordat we deze gaan uitvoeren. Op dat moment zijn veel parameters bekend. Deze parameters kunnen eigenschappen van het platform zijn, zoals het

aantal streaming multiprocessors en cores. Maar nog interessanter zijn applicatie-specifieke parameters, zoals in ons geval bijvoorbeeld het frequentiebereik van de observatie, of het aantal antennevelden dat mee doet. Als je dat weet, dan weet je precies hoe je moet parallelliseren, en hoe je de data in het geheugen kunt ordenen voor de meest efficiënte toegang. Je kunt dus een enorm goede afbeelding maken van het specifieke probleem dat je wilt oplossen naar de exacte hardware waar je op dat moment op draait.

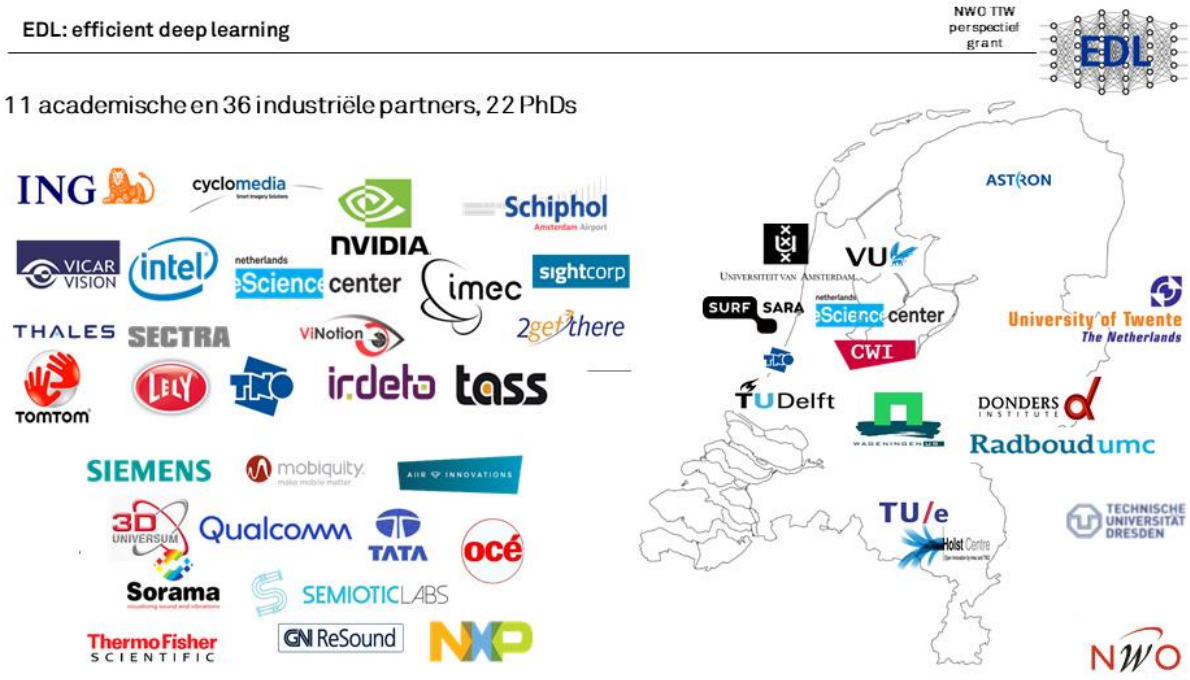
Met auto-tuning genereren we niet 1 versie van de code, maar vele duizenden versies, met andere optimalisatieparameters. Het systeem zoekt dan automatisch de meest efficiënte set van parameters. Een voorbeeldresultaat ziet u hier. We hebben een radioastronomie pipeline ontwikkeld die zoekt naar pulsars: zeer snel roterende neutronensterren. In dit geval vergelijken we 3 verschillende accelerators, en ook 3 verschillende telescopen, hier LOFAR, Apertif en SKA fase 1. We kijken specifiek naar het energieverbruik. Wat we zien is dat telescoopeigenschappen zoals het frequentiebereik een grote impact hebben op de snelheid en ook de energie efficiëntie. We halen op GPUs versnellingen tot een factor 8, terwijl we tegelijkertijd 6x minder energie verbruiken ten opzichte van een traditionele processor. Er zijn voor verschillende hardware platformen, verschillende telescopen en verschillende observaties echt verschillende optimalisaties nodig. Met de hand is dat niet meer te doen.
Het voordeel van onze automatische aanpak met run-time code generatie en auto-tuning is dus een zeer goede portabiliteit en performance, met een laag energieverbruik. Er is echter ook een nadeel. Door de geautomatiseerde zoektocht naar het optimum heb je geen inzicht meer in de reden *waarom* je een bepaalde performance haalt. Dat is op dit moment een van de thema's van ons vervolgonderzoek.

Als we onze benchmark resultaten extrapoleren naar wat er nodig is voor de SKA, dan blijkt dat er in totaal voor fase 1 al 140.000 GPUs nodig zijn, met een energieverbruik van 30 megaWatt. De ambitie voor SKA fase 2 is om het systeem nog met een factor 100 op te schalen. Dan heb je dus 3000 megaWatt nodig. De grootste kolencentrale van Nederland, de Eemshavencentrale, levert ongeveer 1500 MW. Dus, om SKA fase 2 te bouwen met de technologie die we hier getest hebben zijn er 2 van die centrales nodig, alleen maar voor de pulsar pipeline. Dit

geeft wel de enorme schaal en ambitie aan die men nastreeft. Er is overduidelijk nog veel onderzoek nodig om deze droom waar te maken.

2.4. Schaalbaarheid

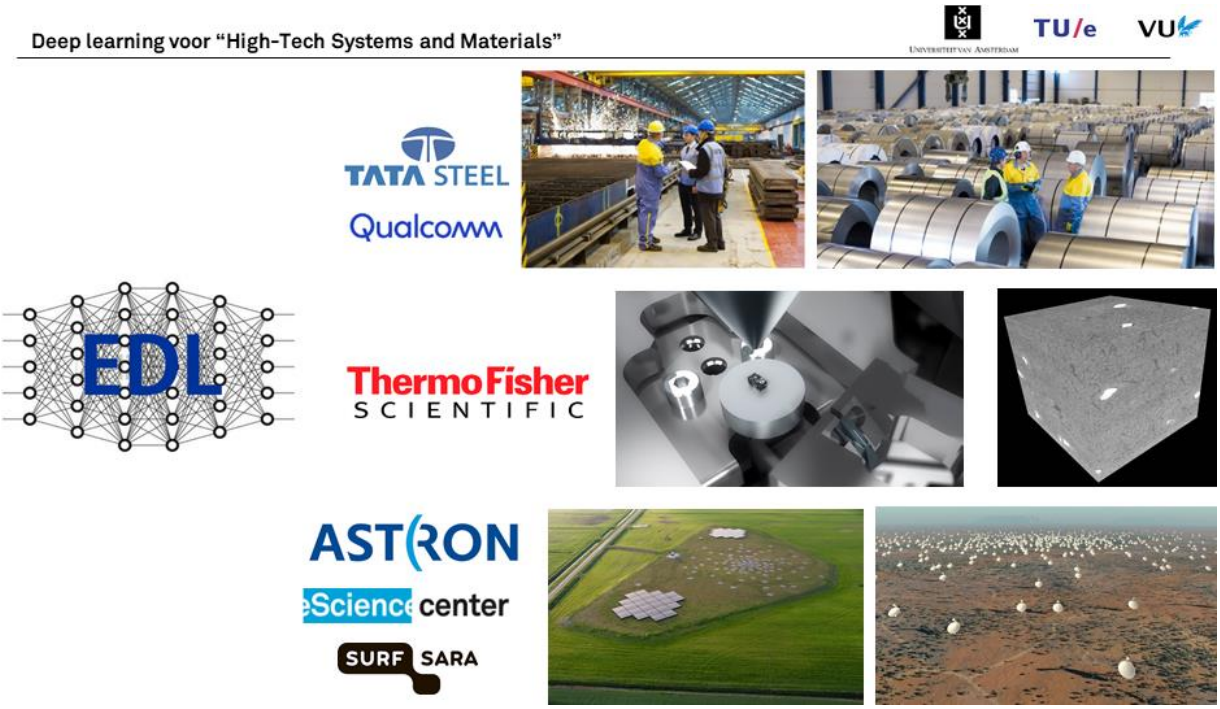
Het laatste aspect van efficiënt rekenen waar ik het over wil hebben is schaalbaarheid. Nu speelt schaalbaarheid al een grote rol in alles wat ik tot nu toe verteld heb. Daarom wil ik nu alleen nog ingaan op een specifiek aspect dat erg actueel is: het opschalen van zelflerende systemen, en in het bijzonder deep learning, een heel actueel onderzoeksthema.



2.4.1. Deep learning op schaal

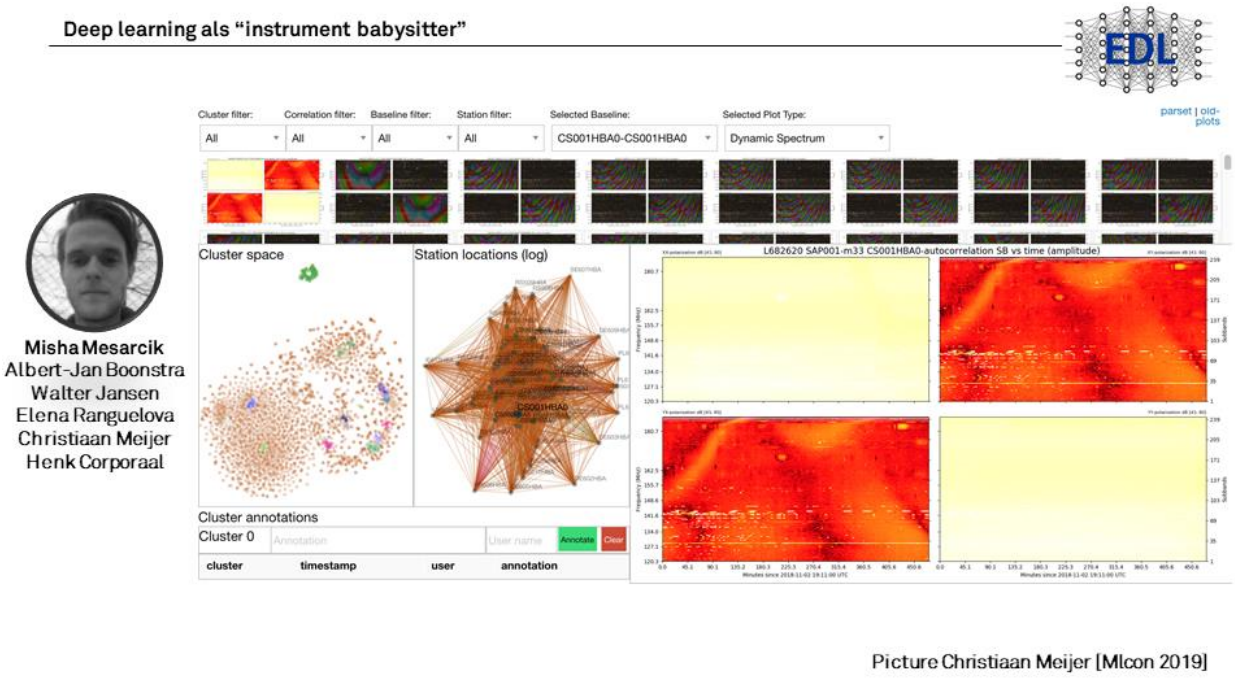
We zijn met dit thema bezig in een groot project, EDL, dat staat voor “Efficient Deep learning”. In dit project zitten naast 11 academische partners ook 36 partners uit de industrie, die spannende uitdagende use cases aanleveren. EDL is een prachtig voorbeeld van het type interdisciplinaire onderzoek dat ik nastreef. Het doorbreekt een aantal hokjes binnen de informatica, het brengt namelijk de Nederlandse experts op het gebied van grootschalig rekenen en de experts op het gebied van kunstmatige intelligentie samen. Het project slaat ook een brug tussen

wetenschap en industrie, met ook expliciet het doel om deep learning toegankelijk te maken voor een veel bredere groep gebruikers, inclusief bijvoorbeeld het MKB. Een mooie kans dus om impact te hebben met eScience.



Ik ben zelf projectleider van een deelproject dat “deep learning voor High-Tech Systems and Materials” heet. Wellicht herkent u dit als een van de topsectoren. We gaan drie use-cases aanpakken die draaien om het opschalen van industriële productieprocessen en instrumenten. Met Tata steel en Qualcomm kijken we naar het opsporen van defecten tijdens het fabricageproces van staal. Met Thermo Fischer onderzoeken we of we de acquisitiesnelheid van elektronenmicroscopen kunnen verhogen, terwijl we tegelijkertijd de beeldkwaliteit verbeteren. Tenslotte kijken we met Astron, het eScience centrum en SURFsara naar het detecteren van instrumentfouten voor LOFAR.

Er zijn een aantal gemeenschappelijke uitdagingen bij deze use cases. We onderzoeken het opschalen van deep learning zodat we kunnen omgaan met de enorme datavolumes die we hebben. De data is in ons geval ook heterogeen: we combineren verschillende soorten data met verschillende eigenschappen uit verschillende bronnen.



Picture Christiaan Meijer [MIcon 2019]

2.4.2. EDL ASTRON use case

Ik wil even inzoomen op de LOFAR use case, waar we deep learning gebruiken als instrument babysitter. Zoals ik al eerder vertelde zijn er bij instrumenten en computersystemen op deze schaal eigenlijk permanent dingen kapot. Ook externe stoorzenders, in dit geval zelfs letterlijk, kunnen de data vervuilen. Onze ambitie is om dit automatisch te detecteren met deep learning. Om dit mogelijk te maken willen we het algoritme de observatie data zelf geven, maar ook domeinkennis in de vorm van meta data over de observatie, informatie over de status van de sensoren, het netwerk, het computerplatform en de software. Door al de informatie met elkaar te combineren en te correleren hopen we automatisch verbanden te kunnen leggen tussen anomalieën in de data en de “gezondheid” van het instrument. De bevindingen presenteren we dan aan de operators, die met een interactieve interface verder kunnen onderzoeken wat er mis is. De operators kunnen het systeem tegelijkertijd helpen met het leren, zodat de foutoorzaak een volgende keer automatisch gedetecteerd kan worden. Deze interface is gemaakt door Christiaan Meijer van het eScience centrum, en het vervolgonderzoek waar we die domeinkennis mee nemen wordt nu gedaan door Misha Mesarcik.

EDL

ING

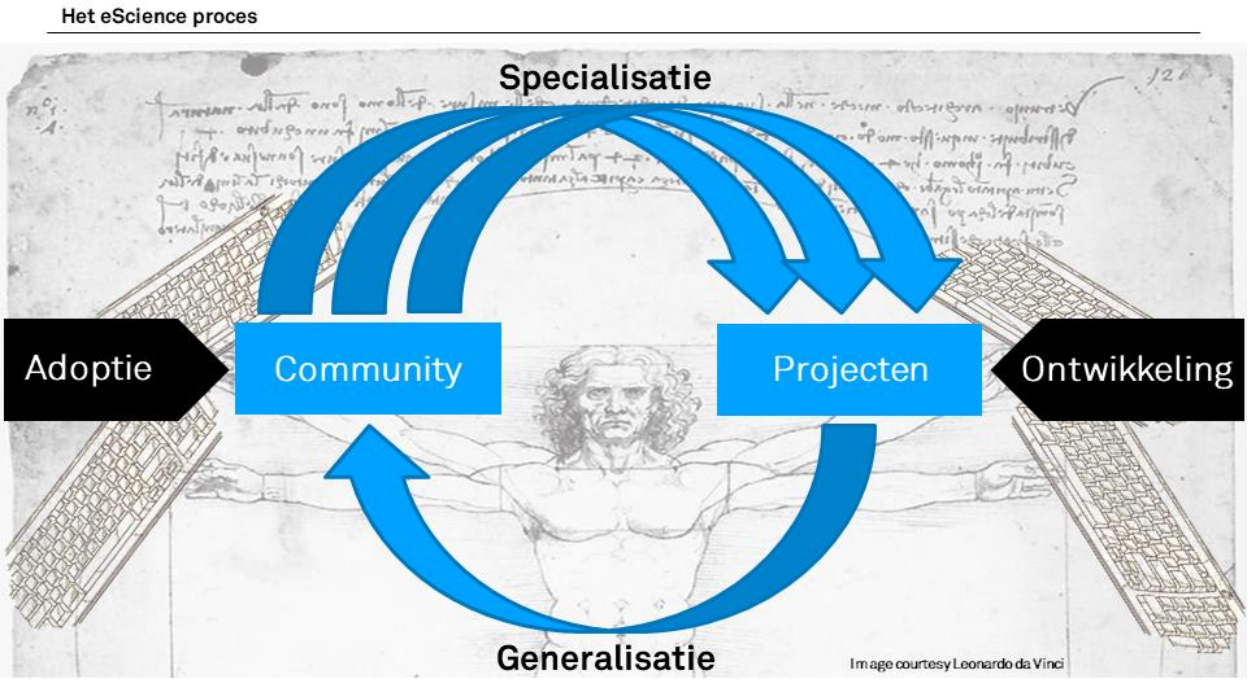
UNIVERSITEIT VAN AMSTERDAM

VU VRIJE UNIVERSITEIT AMSTERDAM

Henk Dreuning
Joost Bosman
Drona Kandhai
Pinar Kahraman
Henri Bal
Mark Hoogendoorn

2.4.2. EDL ING use case

Typisch voor eScience is dat we altijd willen generaliseren. Dat doen we in dit geval ook weer. We proberen dezelfde technologie te verfijnen met een andere use case in een heel andere context, namelijk financiële transacties. Dit EDL deelproject is gemeenschappelijk met de VU en de ING bank, en Henk Dreuning is de PhD student die het onderzoek uitvoert. Ook bij de bank zijn de ICT systemen enorm grootschalig en complex. We willen dezelfde deep learning technologie gebruiken om fouten in het ICT systeem te detecteren. Tegelijkertijd willen we het systeem gebruiken op de stromen van financiële transacties. In dit geval kunnen anomalieën een indicator zijn van mogelijke fraude. Een mooi voorbeeld van maatschappelijke impact met eScience.



2.5. Het eScience proces

Dit maakt de cirkel van mijn verhaal rond. Dit plaatje laat zien hoe het eScience proces in het algemeen werkt. We ontwikkelen nieuwe kennis en software in onze interdisciplinaire projecten **(rechts)**. Een belangrijk aspect van eScience is het hergebruik van de gegenereerde kennis en software. We maken dus een generalisatieslag, waar we de software zo modulair en generiek mogelijk maken, zodat we ook snel nieuwe onderzoeksvragen kunnen beantwoorden, potentieel in andere disciplines **(onder)**. We ontwikkelen open source software, en delen dit met de community. We hergebruiken ook veel bestaande research software. Juist door geen wielen opnieuw uit te vinden en veelgebruikte bestaande software uit te breiden met nieuwe algoritmes die uit ons onderzoek komen, kunnen we een grotere impact realiseren **(adoptie, links)**.

Onze gegeneraliseerde methodes en software passen we altijd weer aan in volgende projecten om goed aan te sluiten bij de domeinwetenschappers met wie we werken. Vaak doen we dat meerdere keren, in verschillende projecten en disciplines **(boven)**. Maatwerk is van groot belang voor acceptatie en impact in het domein. Beide stappen in de cirkel zijn even belangrijk: in de specialisatie stap beantwoorden we de wetenschappelijke vragen, en bevorderen we de acceptatie

en impact; bij de generalisatie maken we de kwaliteits- en efficiency slag. Kortom, wetenschap is specialiseren, en wij als e-scientists zijn gespecialiseerd in het generaliseren en specialiseren. **Deze cirkel dus.** Dankzij het hergebruik en de hoge kwaliteit van de research software levert eScience meer en betere wetenschap op per geïnvesteerde euro.

3. Software-kwaliteit en de impact op de wetenschap

Dan wil ik graag nog 1 laatste punt bespreken.

Buurman en buurman zijn de hoofdpersonen in een van oorsprong Tsjechische tekenfilmserie. In elke aflevering zijn de buurmannen aan het klussen, maar dit gaat altijd op een grappige manier helemaal mis. Vrijwel altijd is de klus aan het einde van een aflevering geklaard, alleen niet op de manier die ze bedacht hadden, er gaat onderweg heel veel fout, en het is enorm inefficiënt. Eigenlijk schrijven wij in de wetenschap software op dezelfde ongestructureerde manier.

In de informatica is software ons instrument. Om goede wetenschap te doen heb je goede instrumenten nodig. Een astronoom maakt geen telescoop van kippengaas en duct-tape. Ze bouwen kwalitatief uitstekende instrumenten, en Nederland doet op deze manier mee aan de wereldtop van het astronomie onderzoek. Deze instrumenten gaan dan ook decennia mee. Ik zou graag zien dat

wij in de informatica hetzelfde gaan doen. Er is een cultuurverandering nodig om te zorgen dat onze instrumentatie van hogere kwaliteit en beter herbruikbaar wordt. Op de lange termijn is dit efficiënter, en genereert het ook meer impact. In mijn optiek is herbruikbare software ook een voorwaarde van de wetenschappelijke methode: onderzoek moet verifieerbaar en reproduceerbaar zijn. Dat is in de informatica maar al te vaak niet het geval.

Ik worstel zelf ook met dit probleem. Vroeger al als AIO en postdoc, toen ik nog zelf veel software schreef, maar ook met mijn promovendi nu. Er is eigenlijk geen tijd om degelijke software te bouwen. De enige oplossing is om expliciet waardering te geven voor het ontwikkelen van software instrumentatie. Wetenschappers moeten software dus gaan zien als een belangrijke onderzoeksbijdrage. We kunnen hier allemaal eenvoudig aan bijdragen: citeer de software die je gebruikt. En softwareontwikkelaars: maak releases, en zorg ervoor dat jouw code vindbaar en citeerbaar is. En, het belangrijkste: wees trots op jouw instrument!

4. Dankwoord

Ik wil graag eindigen met het bedanken van de vele mensen die de afgelopen jaren essentieel zijn geweest voor dit werk.

Allereerst wil ik degenen danken die mijn aanstelling mogelijk hebben gemaakt. Dat waren het college van bestuur, de decaan, de directeur van het Instituut voor Informatica, de bestuursleden van de stichting Nederlands eScience centrum, de leden van het curatorium, en vooral ook professor Cees de Laat en professor Wilco Hazeleger. Bedankt voor deze kans en het vertrouwen.

Dan moet ik natuurlijk beginnen met het bedanken van mijn promovendi. Zij doen al het echte werk. Als u goed heeft opgelet heeft u ze allemaal al langs zien komen, maar hier ziet u ze nogmaals. Graag wil ik ook iedereen bedanken die betrokken is bij de begeleiding van de promovendi. Wat een voorrecht om met jullie allemaal te mogen werken.

Er zijn ook veel mensen met wie ik al heel lang en fijn samenwerk. Te veel om hier nu op te noemen helaas. Ik bedank u daarom graag straks even persoonlijk onder het genot van een drankje tijdens de receptie. Ik wil nu alleen even kort iedereen bedanken van het Systems and Networking Lab van de UvA. Ook dank ook aan iedereen bij het Nederlands eScience centrum. Ook de mensen die inmiddels weg zijn, maar die wel belangrijk zijn geweest. Een bedankje ook voor de support staf van het eScience centrum en van de UVA. And then a special thanks for our eScience research engineers. You are the unsung heroes of science! I am very proud of everything we have accomplished together at the eScience center sofar. Ook veel dank aan SURF en NWO die dit hebben mogelijk gemaakt.

Bedankt ook iedereen van de computersystemen groep van de VU, iedereen bij Astron, en iedereen in de projecten waar ik in de afgelopen jaren bij betrokken ben geweest: onder andere Ibis, GridLab, AstroStream, PROCESS en EDL, en natuurlijk een groot aantal projecten aan het Nederlands eScience centrum. Bedankt voor alle inspiratie!

Dank ook aan mijn co-promotors Aske Plaat en Thilo Kielmann, en promotor Henri Bal. Jullie hebben mijn wetenschappelijke basis gelegd.

Het is ook fijn om altijd terug te kunnen vallen op mijn broer en zus, familie, schoonfamilie en vrienden. Dank jullie wel allemaal.

Dan natuurlijk mijn ouders: Jullie hebben dit uiteindelijk allemaal mogelijk gemaakt door mij altijd alle kansen te geven, en door me te stimuleren om te leren. En, natuurlijk, jullie hebben inmiddels 35 jaar geleden de commodore 64 gekocht, waar dit allemaal mee begonnen is. Dat was nog eens een goede investering! Het is geweldig dat jullie hierbij kunnen zijn, dat is niet bepaald vanzelfsprekend na wat jullie allemaal door hebben gemaakt de afgelopen jaren.

Tenslotte natuurlijk het thuisfront. Patricia, Lucas en Kayleigh, wat zou ik zonder jullie moeten? Bedankt voor jullie steun en liefde.

Ik heb gezegd.