

ICT architectuur SAWEC 2.0

06/04/2021, Maarten Hilferink, Object Vision, in opdracht van PBL.

15/04/2021, Verwerking aanvullingen van Ruud van den Wijngaart

Contents

ICT context	3
Rekencontext:	3
Architectuurcontext:.....	3
Uitwerking rekenstructuur.....	4
Belangrijkste Entiteiten (tabellen) en Attributen	4
In tabelvorm of rekenregels uit te werken	5
Onzekerheid attribuutwaarden	6
Gebruik en bediening, use cases.....	7
Invoer en parameterisatie	7
Uitvoer	8
KEV interactie.....	8
Performance en geheugenbeheer	9
Huidige GeoDMS technieken tbv het snel toepassen van rekenmodellen.	9
Rekenen met arrays	9
Data segmentatie.....	10
Verbeteropties	10
Virtuele data segmenten en pipelining.....	10
Slimmere herschrijving van rekenregels.....	10
Parallellisatie en sampling.....	11
Distributie van gescheiden rekentaken	11
Random distributie van rekenwerk met restricties	11
Dedicated distribution	11
Berekeningen specifiek voor geactiveerde woningen	12
Functionele guidelines tbv parallellisatie.....	12
Structured Dataflows	12
Heterogeneous processing	12
Horizontaal en verticaal schalen	13
Verticaal: rekenproces verdelen over meer cores en RAM op 1 computer.	13
Horizontaal: rekenproces verdelen over meerdere deelprocessen zodat verdeling over meerdere computers mogelijk wordt,	13
Kosten	13
inschatting performance en performance verbeteringen	13
Conclusie en aanbevelingen	15

ICT context

Rekencontext:

- 7 tot 9 miljoen woningen o.b.v. BAG verblijfsobjecten met wonen als gebruiksdoel, met een jaarlijkse mutatie o.b.v. met Ruimtescanner bepaalde nieuwbouw en sloop
- tot 400 optionele maatregelen (hoofdopties: schilmaatregelen in 8 bouwdelen met ieder 3 kwaliteiten, Ruimteverwarming -> InstallatiePerProduct).
- 51 zichtjaren, waarvan 21 voor kalibratie en 30 voor simulatie
- Jaarlijkse activatie van woningen, op basis van voorselectiecriteria op installatie en/of gebouwdeel eigenschappen, waaronder (resterende) levensduur, tbv woning-investeringen obv toepassingscriteria, een discreet keuze model (nested logit choice, S-curves), en volgende ambitietoedeling.
- afweging maatregelen o.b.v. trekking pro rata S-curve op return on investment (positieve cash flow versus duration o.i.d.).

SAWEC 2.0 moet:

- een scenario binnen 1 uur door kunnen rekenen op de Windows 10 Azure omgeving van TNO/PBL,
- kunnen rekenen op een stand-alone Windows 10 machine met 16 GB RAM
- zo mogelijk (dit hangt af van de functionele specificatie van woning overstijgende ambities e.d.) een aselechte steekproef van woningen met pro rata geschaalde budgetten en ambities binnen 2 uur door kunnen rekenen op een stand-alone machine.

Architectuurcontext:

- zie documentatie NEV-Rekensysteem
 - SAWEC dient resultaten te kunnen produceren zonder gebruikers-interventie, dus als batch proces gestart te kunnen worden
 - invoer en uitvoer bestanden als .csv die m.b.v. KEV tools interacteren met de MS Access database op Y:
- performance van interactief werken, mb.v. GeoDmsGUI kan worden verbeterd door:
 - gemakkelijk in te kunnen stellen welke tussenresultaten wel en niet te bewaren ten behoeve van calculation tracing
 - schakelen in random sample-size versus het volledige doorrekenen
 - het kunnen instellen van studiegebieden
- stand alone, of op 1 Azure machine
- schaalbaar naar rekenen met k machines op Azure.

Uitwerking rekenstructuur

Belangrijkste Entiteiten (tabellen) en Attributen

zie ook document 20210222_Woningdataflow.pdf

Woningen in een Dynamische Woningtabel:

- hebben een begin- en eindjaar
- bestaande en nieuwbouwwoningen worden niet onderscheiden
- hebben een gebruiksoppervlakte
- hebben 1 meterkast/voordeur (bestaande woningen) of een aantal meterkasten/voordeuren bij ongedifferentieerde projecties van toekomstige woningen
- hebben een ID uit de BAG of Ruimtescanner output, en een aanvullend volgnummer gerelateerd aan levensduur opsplitsing.
- hebben een aan de ID gerelateerd willekeurig getal tussen 0 en $2^{32}-1$ tbv steekproefselectie en als random-seed t.b.v. het doen van kanstrekkings, zoals woninginvesteringsbeslissing pro rata de uitkomsten van de S-curve toepassing.
- gebouwinformatie en oppervlakken (zie Woningdataflow.pdf)
 - zijn onafhankelijk van de te kiezen woninginvesteringen
 - oppervlakken en de meeste gebouwinformatie kenmerken zijn constant gedurende de levensduur van een woning
- huur/koop/sociaal situatie en bewoners kenmerken zoals huishoudensgrootte en/of inkomenscategorie kunnen dynamisch zijn, maar te overwegen is om een niet erg veranderlijk deel van deze kenmerken ook als statisch te definiëren en bij geprojecteerde veranderingen deze te administreren als ware het een combinatie van sloop en nieuwbouw.
- bronnen voorbereiding
 - Historisch vanaf 2000: alle BAG woningen met start- en eindjaar. Met het BAG snapshot proces zal een historische database van woningen kunnen worden opgebouwd. Hierin zullen woningen die voor 2012 gesloopt zijn waarschijnlijk niet goed zijn opgenomen en bij het ontbreken daarvan zal wellicht een onderschatting van het totaal verbruikt of een overschatting van het verbruik per woning volgen.
 - Heden: BAG + CBS + @@@FO
 - Toekomst: sloop per raster per type: Boolean; nieuwbouw: aantallen per woningtype, aan te vullen met bouwdeel en installatie start eigenschappen; interpolatie naar 1 jaarsperioden
 - stochastisch vastgestelde woningmutaties op grond van regionale verwachtingen uit bijvoorbeeld Ruimtescanner output
- overheveling sociale en commerciële huur en koop.

Woningen in een specifiek zichtjaar

- hebben een dynamische status bestaande uit:
 - een 1..3 kwaliteit per bouwdeel (8x)
 - wel of niet een gas of H2 aansluiting; het hebben en krijgen (neem ik aan?) van een warmtenet aansluiting kan als exogeen beschouwd worden.
 - een installatie per functioneel-product (10x) met een aanschafjaar per installatie
- hebben een verwachte WOZ waarde

- huur/koop/sociaal situatie en bewoners kenmerken zoals huishoudensgrootte en/of inkomenscategorie, voor zover niet meegenomen in de definitie van wat een woning identificeert.

Stamtabellen

- bouwdeel: {RO: Raam onder, RBO Raam boven, DD: deur, PL: paneel, MG: gevel, MS: spouwmuur, DS: Schuin dak, DP: plat dak }
- product: {RWb, RWp, TWb, TWp, KDb, KDp, AS, DK, KK, VT }
- zichtjaar: 2000..2050
- woninginvestering:
 - per bouwdeel (8x): verbetering _LM: 1->2, _LH: 1->3 of _MH: 2->3, of niet van toepassing
 - per functioneel-product (10x); installatie met gerelateerde performance, primaire energiedrager (MT, LT, gas, H2), elektrisch verbruik en investeringskosten
 - heeft technisch en beleidsmatige toepassingscriteria (een investering kan worden opgelegd door in casu niets doen niet van toepassing te laten zijn)
 - Heeft een bandbreedte:
 - _MIN: zelfstandig moment/individuele aanpak
 - _MAX: natuurlijk moment/collectieve aanpak
- Activerings criteria:
 - kunnen doelgroep gerelateerd zijn of vervangingsverwachting van een bouwdeel-kwaliteitsniveau of installatie, waaronder dak en AfgifteSysteem;
 - leiden tot activering van een bouwdeel of product.
 - activeert die woninginvesteringen die (al dan niet) alleen de geactiveerde bouwdeel of installatie verbeteren.

Overige brondata

- (dynamische) klimaatkaarten
- ingroeikaarten warmtenetten.

In tabelvorm of rekenregels uit te werken

- functionele vraag:
 - hoe wordt die berekend voor verschillende bouwdeel kwaliteiten?
 - hoe wordt performance per bouwdeel optie gespecificeerd?
 - rekenregels specifiek voor bouwdelen of te generaliseren tot enkele tabellen ?
- metervraag: gegeven een functionele vraag, berekening conform gegeven installaties en performances, wellicht zoals in Vesta, met toevoeging van dak, ventilatie, en keukenapparatuur
- beleidsmaatregelen (belastingen, subsidies, verplichtingen (?)):
- doelen en ambities (te realiseren na toewijzing individuele woninginvesteringen):
 - hebben een groeps criterium (bijvoorbeeld alle huurwoningen)
 - stellen een minimum of maximum aantal woningen met een bepaald bouwdeel kwaliteit of product-installatie
 - of stellen een maximum individueel of gemiddelde metervraag per energie-drager
 - of stellen een minimum individueel of gemiddelde bouwdeel kwaliteit

- kostentabellen en/of leercurves:
 - voor installaties
 - kosten [EUR/m²] bouwdeelverbeteringen per niveausprong {1->2, 1->3, 2->3} per bouwdeel, zijn afhankelijk van het vertrekpunt (itt installaties)
- gedrag en de verandering daarvan (tgv o.a. overheidsbeleid)
 - verbruiksgedrag(bewoners eigenschappen) * oppervlaktes(woning eigenschappen) / R(kwaliteit bouwdelen) -> functionele vraag
 - rebound ?
 - correlaties ?
 - investerings gedrag ofwel de afweging van alternatieve toegestane woninginvesteringen igv een activatie:
 - nested choices met S-Curve parameters per sub-populatie, zie @@@FO,
 - hoofdkeuze: energiedrager ruimteverwarming, en
 - sub-keuze: gewenste isolatiekwaliteit.
 - gegeven een hoofdkeuze en isolatiekwaliteit, maak een afweging van alternatieve woninginvesteringen op basis van een discounted cash-flow inclusief comfort effecten en een S-curve.
- normeringen en verplichtingen op woningniveau worden geïmplementeerd door activeringscriteria en toepasbaarheid van gerelateerde woninginvesteringen.

Onzekerheid attributwaarden

Binnen de context van 1 run worden alle waarden van attributen (van woningen en opties voor woningen) als vast per tijdstap verondersteld en bepaald door invoerdata, rekenregels en de waarden van voorgaande tijdstappen. SAWEC is daarmee een dynamisch simulatiemodel. Projecties naar de toekomst spelen in deze simulatie een beperkte rol, namelijk bij het evalueren van te overwegen woninginvesteringen, waarbij een investering wordt afgewogen tegen de verwachte (vermindering van) toekomstige kosten. Onzekerheid wordt geoperationaliseerd door tijdens de simulatie pseudo random trekkingen te doen. Dit speelt een rol bij:

- het kiezen van de primaire energiedrager voor ruimteverwarming op basis van een S-curve score op hoofdlijnen, waaronder beste score per energiedrager, maar met een lager gewicht dan in de nested-choice
- het kiezen uit de toegestane woninginvesteringen op basis van de S-curve score per investeringsoptie
- in voorbereidingen (pre-processing):
 - het vaststellen van woningmutaties op grond van regionale verwachtingen (uit bijvoorbeeld Ruimtescanner output) met betrekking tot sloop en nieuwbouw
 - het selecteren van woningen voor transitie huur/koop/sociaal
 - het relateren van (verwachte regionale) inkomensklasse verdeling aan woningen

Aangezien een aantal van bovenstaande stochastische attributen niet als onafhankelijk beschouwd kan worden, is het van belang goed aandacht te besteden aan de organisatie van de trekkingen. Wanneer bijvoorbeeld bekend is dat CBS buurten met veel sociale huurwoningen ook relatief veel huishoudens wonen met lage inkomensklasse, worden verdelingsvraagstukken onderbelicht wanneer in een buurt met 50% huur/koop en 50% laag/hoog inkomen deze attributwaarden onafhankelijk worden gesampled.

Gebruik en bediening, use cases

SAWEC dient allereerst gebruikt te kunnen worden om met extern ingestelde parameters een tijdreeks door te rekenen en resultaten in .csv bestanden te bewaren

- met name binnen het kader van KEV-runs mbv het NEV-Rekensysteem.
- meestal 1 run, bestaande uit het simuleren van alle toekomstige zichtjaren, soms ook het draaien van productie-batches of sensitivity-batches.
- een run dient distribueerbaar te zijn over een op te geven aantal AZURE machines of op een stand-alone Windows 10 Desktop of Workstation.

Aanvullend zal SAWEC gebruikt worden tbv

- gevoeligheidsanalyse: meerdere runs met eigen instellingen. Waarschijnlijk zal het bij het doorrekenen van een groot aantal runs effectiever zijn om niet de woningen maar de runs over verschillende machines te verdelen; een batch van runs kan beter per bundel runs gedistribueerd worden, maar dit heeft bij de implementatie van SAWEC geen prioriteit, kan later alsnog eenvoudig aanvullend geïmplementeerd worden,
- model kalibratie van Activatie en S-curve parameters op historische data. Optimalisatie van een Goodness of Fit van model-uitkomsten en waarnemingen? Dit behoeft mogelijk aanvullende statistische optimalisatie tools, zoals modules tbv multi-nominal logistic regression, beschikbaar in o.a. R. De opzet van model-kalibratie behoeft aanvullende bestudering en ontwerp en eventueel implementatie door iteratief bijgestuurd doorrekenen van een historische situatie mbt GeoDmsRun
- interactief werken tbv exploratie en herleiden van resultaten, en het debuggen van aannames en rekenregels: GeoDmsGui.
- productie van tijdreeksen tbv grafieken in bijvoorbeeld Excel.
 - er kan een lijst indicatoren per zichtjaar worden ge-exporteerd, hiermee kan met een grafiek tool, zoals Excel, een tijdreeks grafiek worden samengesteld.
 - De gebruiker kan vooraf aan een run aangeven van welke indicatoren een tijdreeks tabel gevormd moet worden tbv gemakkelijkere verwerking tot tijdreeks grafieken,.

Invoer en parameterisatie

invoer data bestanden, waaronder parameter files in .csv format omvatten:

- een selectie criterium van woningen: alle, of op basis van kenmerken en/of een bepaald studiegebied
- het aantal rekenmachines. Wanneer op Azure gewerkt wordt, zal er in voorzien zijn dat meerdere rekenmachines gestart kunnen worden die ieder een deel van de woningen simuleren en resultaten daarvan weg schrijven op een gedeelde folder. De aansturende machine zal na gereedkoming de resulterende resultaten samenvoegen.
- alle of een bepaalde random gekozen fractie tbv snellere resultaten. Woning-keuze overstijgende ambities en budgetten zullen pro rata zulke fracties en eventuele distributie over verschillende rekenmachines worden toebedeeld.
- .csv formaat tabellen met woningkenmerken stamtabellen, leercurves,
- .csv .shp of .tif formaat tabellen met brongegevens woningen uit de BAG, Ruimtescanner en gedownsamplde CBS brongegevens

GeoDMS script bestanden, waarvan een gedeelte gemarkeerd is als zijnde eenvoudig wijzigbaar door gebruikers tbv het definiëren van een run .

Ontkoppelde voorbewerkte data, te lezen in een snel binair dataformaat zoals .fss, TIF, en/of NetCDF, waaronder

- resultaten van kalibratie
- een dynamische tabel van historische, bestaande en toekomstige woningen (mbv een begin- en eventueel eind-jaar) met een toegekend random volgnummer tbv sampling en distributie.
- klimaatkaarten met aantal vorstdagen.

Dit dient in het Functioneel Ontwerp of elders verder uitgewerkt te worden met een technische beschrijving van de belangrijkste te lezen .csv bestanden. met opslaglocatie, filenaam, formaat, en attribuut namen

[Uitvoer](#)

Extensieve statistieken van de gesimuleerde woning-samples met ten minste functionele vraag en metervraag per energiedrager, per installatiepakket per jaar, en per koop/huur/sociaal categorie.

Dit dient in het Functioneel Ontwerp of elders verder uitgewerkt te worden met een technische beschrijving van de belangrijkste resulterende .csv bestanden. met opslaglocatie, filenaam, formaat, en attribuut namen

[KEV interactie](#)

- Mbv de GUI van het NEV-RekenSysteem kan worden gekozen voor welk project, scenario, en model.run jaren gerekend moet worden. Mbv environment variabelen zal SAWEC de daarbij behorende data paden voor invoer en uitvoer moeten kunnen bepalen
- In het NEV-Rekensysteem is ook een SAWEC parameter dialoog te activeren. Hierin door de gebruiker gekozen instellingen dienen ook in een parameter file te worden geschreven en door SAWEC gelezen. Specificatie van deze GUI zal in de uitwerking van het Functioneel Ontwerp beschreven moeten worden.
- lezen van KEV energieprijzen, economische parameters, en warmtenetten van NEV-RS in .csv format met TAB separators.
- schrijven van verbruiksgegevens naar NEV: .csv formaat.
- afnemers SAWEC resultaten: SELPE, SAVE-Production, RESolve-E, Competes (elektriciteit)
- voor SELPE: verbruiksgegevens per installatie per jaar, per koop/huur/sociaal categorie, fictief functioneel verbruik.
- bij gedistribueerde simulatie van woningen, is het nodig de extensieve statistieken van de woning-samples te aggregeren en de gedistribueerde aggregaties samen te voegen. Dit kan als postprocessing stap na completering van de gedistribueerde berekening opgezet worden.

Dit dient in het Functioneel Ontwerp of elders verder uitgewerkt te worden.

Performance en geheugenbeheer

Hierbij een overzicht van technieken en methoden om aan de gestelde reketijden te kunnen gaan voldoen.

Allereerst een beschrijving van de huidige architectuur, technieken en mogelijkheden om snel te rekenen en vervolgens mogelijke verbeterpunten.

Huidige GeoDMS technieken tbv het snel toepassen van rekenmodellen.

Rekenen met arrays

Het datamodel van de GeoDMS bestaat uit

- Entiteiten. Een entiteit is een value-range met metriek of projectie informatie, met een eventuele segmentatie van die range (aka tiling).
- Attributen. Een attribuut is een mappings tussen twee entiteiten, geoperationaliseerd als gesegmenteerde arrays van data-elementen.

De data-values van zowel entiteiten als attributen kunnen

- expliciet gedefinieerd worden,
- gerelateerd worden aan een externe bron, of
- voorzien van een rekenregel waarmee de waarden berekend moeten worden.

Een rekenproces bestaat uit het uitvoeren van reken-operaties op arrays hetgeen processing units veel sneller doen dan als voor iedere rekenstap nieuwe instructies verwerkt moeten worden, mede dankzij SIMD vectorisatie mogelijkheden en vergelijkbare optimalisatie technieken.

Iedere instantiatie van een geparametriseerd reken script dient benoemd te worden, net als iedere instantie van een `for_each` repetitie of loop, waarmee iedere berekende entiteit en attribuut een unieke naam heeft en iedere naam een eenduidige data-value. Het ontbreken van object states geeft de voordelen van functioneel programmeren: Referentiele Transparantie memoization en herberekenbaarheid.

Dit value based rekenmodel maakt mogelijk een rekenproces te beschouwen als data-flow diagram van data-values en rekenstappen waarbij het beschikbaar hebben van een data-value voorgesteld kan worden als een pebble op dat diagram en het uitvoeren van een rekenstap kan aanvangen als alle benodigde data-values van een pebble zijn voorzien, resulterend in een pebble op het reken-resultaat. Het bereiken van een rekendoel kan dmv een strategie waarmee steeds een linie tussen bronnen en rekendoel wordt dichtgehouden. Data-values kunnen voor meerdere rekenoperaties gebruikt worden (memoization) of juist on-demand, just in time, opnieuw berekend worden. Ook kan er ingesteld worden dat bepaalde resultaten langer bewaard worden tbv hergebruik in toekomstige rekendoelen, wat vooral handig is als een gebruiker stap voor stap wil nalopen hoe een resultaat-element berekend is (calculation tracing).

De huidige rekenstrategie van de GeoDms is om voor gevraagde resultaten alle benodigde rekenstappen te beschouwen en steeds rekenstappen waarvoor de benodigde data-values beschikbaar zijn in een start-queue te zetten en te starten voor zover het maximum aantal reken-threads, nu gelijk gesteld aan het aantal processing-units, dat toelaat. Steeds wanneer een rekentaak gereed komt, wordt dit opnieuw geëvalueerd. Een data-value wordt bewaard totdat het laatste gebruik in een rekentaak geëindigd is.

Data segmentatie

Teneinde grote datasets beter behapbaar en in meerdere threads bewerkbaar te maken, kan de value-range van een entiteit gesegmenteerd worden. Gerelateerde attributen worden hiermee gesegmenteerd. De meeste rekenoperaties, met name die die per element gedefinieerd zijn, zijn zo opgezet dat deze per data-segment werken en dat de verwerking van verschillende data-segmenten over verschillende threads verdeeld worden.

Zie ook:

- https://www.geodms.nl/GeoDMS_Academy
- https://www.geodms.nl/Value_based_calculating
- https://nl.wikipedia.org/wiki/Functioneel_programmeren
- Taskflow: A Parallel and Heterogeneous Task Programming System Using Modern C++ - Tsung-Wei Huang <https://www.youtube.com/watch?v=MX15huP5DsM&t=2050s>
- Rust programming language concepts [https://en.wikipedia.org/wiki/Rust_\(programming_language\)](https://en.wikipedia.org/wiki/Rust_(programming_language))

Verbeteropties

Virtuele data segmenten en pipelining

De huidige GeoDMS strategie van eenmalig berekenen en tot laatste gebruik bewaren van data-values is niet optimaal voor eenvoudig te (re)genereren data-values, zoals in:

```
attribute<kW> Aansluitwaarde(woning) := const(3[kW], woning);  
attribute<kW> NetCapaciteit (woning)) :=  
    Aansluitwaarde *  
    GelijktijdigheidsFactor[woning_Categorie];
```

Ook is het wenselijk pipelining van data-flow per segment uit te gaan voeren teneinde niet alle data-value segmenten van tussenresultaten tegelijk te hoeven bewaren. Verwerking van meerdere operaties per segment in plaats van alle segmenten per operatie te verwerken zal leiden tot kleiner benodigd geheugen. Voorgesteld wordt om het resultaat van een rekenoperatie niet altijd te laten bestaan uit een reeks array segmenten, maar soms uit een reeks functors die de array segmenten op aanvraag kunnen (her)berekenen, waarbij de gebruikende operatie (die nu een tile-read lock houdt) een shared owner is van het data-value segment.

Slimmere beschrijving van rekenregels.

Voordat rekenregels toegepast worden, vindt eerst toepassing van herschrijfgeregels plaats, waarmee o.a. sommige reken-operaties in termen van andere operaties worden gedefinieerd. Momenteel kunnen deze herschrijfgeregels niet selectief toegepast worden.

Aangezien in Vesta/Mais en SAWEC 2.0 veel attributen woning specifiek moeten kunnen zijn tbv enkele woningmaatregelen maar voor de meeste woningmaatregelen een eenvoudige implementatie van 0 of een constante waarde hebben, is het zinvol hier gebruik van te maken in de verwerking van de rekenregels. Het herkennen van een factor of term met de waarde 0 kan voorkomen dat onnodig gerekend wordt en invulling van zulke specifieke rekenregels in een generiek sjabloon zou moeten kunnen leiden tot het wegsubstitueren van veel woning specifieke berekeningen.

Het in bovenstaand voorbeeld herkennen dat een resultaat berekend kan worden en constant is per categorie van woningen, kan voorkomen dat voor een concrete subset van woningen en woningOpties voor ieder woning dezelfde berekening gemaakt wordt.

Parallellisatie en sampling

Met parallelisatie wordt hier bedoeld: het over meerdere Azure Windows 10 machines verdelen van rekentaken teneinde sneller rekenresultaten te verkrijgen. Ieder machine heeft:

- een lokale kopie van de software, de source data, en lokale opslag voor tussenresultaten (SSD 64 GB),
- lees/schrijf toegang tot gemeenschappelijke opslag,
- en weet via environment variabelen hoeveel machines er ingezet worden en wat het volgnummer van de machine zelf is.

Teneinde rekenwerk aan woningen en woninginvesteringen over meerdere rekenmachines te kunnen verdelen, is een mate van onafhankelijkheid van de keuzes wenselijk. Dit speelt met name bij:

- toepassing van subsidieregelingen met een gelimiteerd budget
- toepassing van ambitie regels, waarbij na individuele investeringsbeslissingen bij een eventueel resterend quotum van de woningen een maatregel wordt toegepast.

Distributie van gescheiden rekentaken

Wanneer SAWEC met een reeks van verschillende scenario-instellingen gedraaid moet worden, zoals bij een gevoeligheidsanalyse, is parallelisatie relatief eenvoudig.

Een gemeenschappelijke lijst van taken en het gedistribueerde oppakken van die taken en wegschrijven van resultaten in een gedeelde folder tbv postprocessing volstaat.

Random distributie van rekenwerk met restricties

Woning overstijgende subsidieregelingen en ambities kunnen beschouwd worden als een op de uitkomsten van toepassing zijnde restrictie met een bijbehorende schaduwprijs die onderscheidt welke woningen wel en niet mee doen. Bepaling van die schaduwprijs is een niet lokale berekening. Toepassing van die prijs wel.

Als n woningen random over k machines worden verdeeld en fractie $p := m/n$ van de woningen in aanmerking komt voor een regeling, is het verwacht aantal woningen dat op een specifieke machine in aanmerking komt voor een regeling: m/k , met als standaarddeviatie $\text{SQRT}(n/k * m/n * (1-m/n)) = \text{SQRT}(m/k * (1-m/n))$, dus iets kleiner dan $\text{SQRT}(m/k)$. Wanneer een schaduwprijs op basis van een random geselecteerde deelverzamelingen worden bepaald door restricties volgens de opsplitsing mee te schalen, zullen een beperkt aantal woningen aan de verkeerde groep worden toegewezen en de werkelijke schaduwprijs zal ergens tussen de k gedistribueerd vastgestelde schaduw prijzen gevonden moeten worden. Het aantal dan centraal opnieuw te beschouwen woningen zijn de woningen die binnen de gevonden schaduwprijs range liggen, naar verwachting $k * \text{SQRT}(m/k) = \text{SQRT}(m*k)$ woningen. Het wordt aanvaardbaar geacht om deze afwijkingen te negeren en regelingen pro rata gedistribueerd toe te passen.

Dedicated distribution

Functionele afwegingen:

- zijn woningen voldoende homogeen, of zouden juist subsets met verschillende mogelijke woninginvesteringen verdeeld moeten worden.

key words: random sampling, task picking

Berekeningen specifiek voor geactiveerde woningen

Afwegingen voor een beperkte subset van alle woningen leiden tot status updates van kenmerken van die woningen. Een snelle operatie die woning attributen bijwerkt is gewenst:

```
status(woning) := actievewoning ?
nieuweStatus(invert(activewoning_sel/woning_rel)) :
VorigeStatus(woning)
```

Functionele guidelines tbv parallellisatie

Tbv segmentatie en parallelle verwerking is het wenselijk dat de berekening van woning gerelateerde kenmerken zo min mogelijk afhankelijk is van de rekenuitkomsten van andere woningen.

- Voor SAWEC 2.0 is al gesteld dat toepassing van gebiedsopties (zoals warmtenetten) als exogeen beschouwd zullen worden.
- Segmentering van verblijfsobjecten zou zodanig moeten geschieden dat appartementen in panden met gemeenschappelijke ketels altijd in dezelfde subset vallen, zodat ook investeringen in panden tbv alle appartementen daarin overwogen kunnen worden.
- in het rekenmodel kan worden opgenomen hoe het machinenummer i en het aantal machines k een rol speelt in de selectie van de te verwerken woningen, zodanig dat dit ook werkt op een stand-alone workstation, dus wanneer $k=1$.
- een synchronisatie functie zal moeten worden toegevoegd dat een proces laat wachten op de gereedkoming van k resultaten (files in een gedeelde netwerk-folder), waarna gedistributeerde berekening zich voort zet, bijvoorbeeld met een volgend jaar of iteratie

Structured Dataflows

Verwerking van huidige dataflow kan leiden tot veel items igv loop unrolling en lang vasthouden van grote tussenresultaten tbv gebruik in daarvan in meerdere eindresultaten. Een manier om beter aan te geven welke (aggregatie) stappen per iteratie uitgevoerd moeten worden voor met volgende rekenstappen verder te gaan, zou kunnen zijn het in de data-flow analyse expliciet rekening te houden met een hiërarchie van taken, waarbij de end-nodes van een sub-diagram eerst uitgerekend moeten zijn voor er verder gegaan wordt met een volgende.

Zie ook: Taskflow: A Parallel and Heterogeneous Task Programming System Using Modern C++ - Tsung-Wei Huang: <https://www.youtube.com/watch?v=MX15huP5DsM&t=2050s>

Heterogeneous processing

Gebruik van GPU's voor eenvoudige vector bewerkingen kunnen de CPU's behoorlijk versnellen en ontlasten

De GeoDMS wordt ontwikkeld met Visual Studio, waarmee array bewerkingen eenvoudig naar GPU's te verplaatsen zijn mbv Accelerated Massive Parallelism (AMP). De GeoDMS rekenkern code is zodanig opgezet dat hier eenvoudig gebruik van kan worden gemaakt.

kanttekeningen

- beperkte winst wanneer data tussen RAM en disk de bottleneck is
- winst neemt toe wanneer door geheugengebruik besparingen er minder via deze bottleneck moet

Zie ook:

- <https://docs.microsoft.com/en-us/cpp/parallel/amp/cpp-amp-overview?view=msvc-160>
- <https://developer.nvidia.com/how-to-cuda-c-cpp>
- Taskflow: A Parallel and Heterogeneous Task Programming System Using Modern C++ - Tsung-Wei Huang <https://www.youtube.com/watch?v=MX15huP5DsM&t=2050s>

Horizontaal en verticaal schalen.

Verticaal: rekenproces verdelen over meer cores en RAM op 1 computer.

Plafond wordt bepaald door:

- hardware: connectedness van fysieke cores en RAM
- en door de mate waarin de software van meerdere cores gebruik kan maken en niet alle threads kluitjesvoetbal spelen.

Bij het huidige aanbod op Azure lijken de rekenkosten bij horizontaal schalen lineair op te lopen tot aan 256 Cores in stappen van [8 cores + 16 GB].

Horizontaal: rekenproces verdelen over meerdere deelprocessen zodat verdeling over meerdere computers mogelijk wordt, vereist onafhankelijkheid of tenminste beperkte afhankelijkheid van de deelprocessen

bijvoorbeeld doorrekening van woning partities of verdeling van meerdere runs

Plafond wordt bepaald door:

- hardware: binnen SAWEC context min of meer onbeperkt.
- zinvolle grootte van een woning partities
- of aantal runs

Kosten

kosten azure [128/256 Cores + 64 GB] ram x 1 uur versus 16/32 x [8 cores + 16 GB] x 2 uur zijn min of meer gelijk.

Zolang een proces met verticaal schalen kan worden versneld, lijkt dit eenvoudiger en effectiever.

inschatting performance en performance verbeteringen

De uiteindelijke rekentijd hangt van veel factoren af en het is lastig daar vooraf harde garanties voor te kunnen geven. Wel kan op basis van nu bekende informatie en kennis over effecten van veranderingen een inschatting te maken,

Uitgangspunten:

- rekentijd Vesta met 7 miljoen bestaande woningen, een beperkt aantal gebouw-opties, geen gebiedsopties, 21 zichtjaren
 - op een Azure machine (GeoDms specificatie): 8 uur.
 - op een snelle thuis-desktop (8 cores, 4.3 GHz DDR4 2133 MHz): 2 uur
- kalibratieresultaten en een dynamische woningtabel zijn ontkoppeld beschikbaar gemaakt mbv voorbewerkingen
- geen gebiedsopties
- rekenen met 400 woninginvesteringen in plaats van 4 gebouwopties: factor 100 langzamer

- alleen doorrekenen van geactiveerde woningen, mits niet meer dan 5% per jaar geactiveerd en selectie en samenvoeg operaties hier goed op werken en niet bij de ambities weer alle woninginvesteringen voor alle andere woningen ook beschouwd hoeven worden: factor 10 sneller
- ambities en subsidiebudgetten zijn goed op te delen over tenminste 256 rekenprocessen: factor 100 sneller
- zonodig kunnen die Azure machines nu, danwel in de nabije toekomst, nog wel zwaarder en sneller geconfigureerd worden: 2 uur in plaats van 8 uur: factor 4 sneller
- conclusie: $8 \text{ uur} * 100 / 10 / 4 = 20 \text{ uur}$ op 1 machine, 1 uur op ongeveer 32 machines en 12 minuten op ongeveer 256 machines.

Conclusie en aanbevelingen

Gestelde rekendoelen zijn haalbaar.

Prioriteitenlijst performance verbetering SAWEC

- functioneel: zorgen dat het rekenwerk niet meer (tussen) resultaten behoeft dan nodig, de activatie zodanig werkt dat niet woninginvesteringen voor een beperkte subset overwogen hoeven te worden en het rekenwerk kan worden gedistribueerd
- software: memory issues recentere GeoDms versies oplossen en tot tot die tijd rekenen met Vesta versie GeoDms, dmv beter systematisch toepassen van moderne C++ programmeer concepten, zoals RAII, std:future's monads in exception handling, etc.
- technisch: verticaal schalen tot aan 64 GB en 400 Processoren
- technisch: horizontaal schalen obv verdeling woningen over meerdere processen op dezelfde of verschillende machines
- software: virtuele data-segmenten en segment-pipelining tbv minder geheugen beslag van tussenresultaten
- software: structured dataflows

(Nog) niet voor SAWEC 2.0 doen:

- software geschikt maken voor GPU processing
- supporten van distributie van gescheiden rekentaken (zoals tbv gevoeligheidsanalyses) tbv run-batch processing