



UvA-DARE (Digital Academic Repository)

Distributed Event-driven Simulation- Scheduling Strategies and Resource Management

Overeinder, B.J.

Publication date
2000

[Link to publication](#)

Citation for published version (APA):

Overeinder, B. J. (2000). *Distributed Event-driven Simulation- Scheduling Strategies and Resource Management*. University of Amsterdam.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

Dutch Summary

Nederlandse Samenvatting

Simuleren is een alledaagse activiteit die de laatste decennia niet meer weg te denken is in onze maatschappij. Simuleren omvat een scala van activiteiten dat gericht is op het verkrijgen van meer inzicht in het systeem dat onderwerp van studie is. Het bestudeerde systeem kan variëren van de aerodynamische eigenschappen van een vliegtuig in ontwikkeling, tot de eigenaardigheden van de dynamica van een proteïne die zich opvouwt in een complexe, drie-dimensionale structuur. Of het bestudeerde systeem is een organisatie of gemeenschap, zoals bijvoorbeeld efficiënt voorraadbeheer van een bedrijf, of de verspreiding van HIV en hepatitis C bij injecterende drugsgebruikers.

Alhoewel de onderzoekdisciplines verschillen, blijft de aanpak van een simulatiestudie min of meer identiek. Een model van een bestaand of theoretisch fysisch systeem wordt ontworpen, experimenten met het model worden uitgevoerd, en de resultaten van de experimenten worden geanalyseerd. Modellen van een fysisch systeem komen voor in alle mogelijke vormen: het meest exacte model is het fysisch systeem zelf, een in grootte geschaald model van het fysisch systeem, of een abstract model beschreven door middel van wiskunde of een formele logische taal. In het algemeen, worden abstracte modellen gerealiseerd in computerprogramma's, ook wel computersimulaties genoemd, zodat de experimenten met het abstracte model uitgevoerd kunnen worden op computers. Het onderwerp van dit proefschrift is computersimulatie en in het bijzonder de efficiënte computersimulatie van systemen, die gekarakteriseerd worden door een heterogeen spatiëel en temporeel gedrag, hetgeen betekent dat veranderingen in het systeem op wisselende plaatsen en op variërende tijden plaats vinden. Systemen met dit gedrag komen algemeen voor in, bijvoorbeeld, populatie dynamica, immunologie, statistische natuurkunde, en informatica. Systemen met heterogeen spatiëel en temporeel gedrag worden het meest nauwkeurig op zogenaamde asynchrone modellen afgebeeld.

Experimenteren met complexe simulaties en de interpretatie van grote hoeveelheden gegenereerde data is een uitdagende onderneming. In dit opzicht worden virtuele omgevingen, die de visualisatie van data en de interactie met de computersimulatie integreren, steeds belangrijker. Een goed voorbeeld is een virtueel laboratorium, waar wetenschappers interactief met een simulatie experimenteren. Als de wetenschappers een wijziging in het simulatiemodel aanbrengen, moet de computersimulatie stop gezet worden, of zelfs terugge-

bracht worden naar een situatie in het (simulatie-) verleden, zodat de wijzigingen doorgevoerd kunnen worden.

De combinatie van complexe simulaties, asynchrone modellen en virtuele omgevingen resulteert in grote applicaties, die voldoende computercapaciteit eisen om interactieve experimentatie mogelijk te maken. Dergelijke vereisten zijn zelfs met de snelste supercomputers niet eenvoudig te realiseren (ook niet door het gedistribueerde karakter van de virtuele laboratorium toepassing). Gedistribueerde en parallelle oplossingsmethoden zijn een passend alternatief voor de virtuele laboratorium toepassing, waar de verzamelde computercapaciteiten gebundeld worden om de benodigde prestatie te leveren.

In onze studie naar gedistribueerde en parallelle oplossingsmethoden, exploiteren we de lokaliteit van data en verwerking van gebeurtenissen met behulp van technieken uit parallelle discrete event simulatie. De verdeling van werk over de gedistribueerde computers wordt gecoördineerd door het Dynamite executiesysteem. De gecombineerde parallelle verwerking van simulatiegebeurtenissen en de verdeling van werk over de gedistribueerde computers is een zeer complexe taak. Daarom zijn de twee (sub-) problemen onafhankelijk van elkaar bestudeerd, maar wel zodanig dat beide componenten geïntegreerd kunnen worden in een omgeving.

Het ontwerp en implementatie van een simulatie-omgeving voor parallelle discrete event simulatie heeft geresulteerd in het APSIS systeem, beschreven in Hoofdstuk 3. De kern van APSIS is het Time Warp optimistische simulatiemechanisme. De belangrijkste verantwoordelijkheid van een simulatiemechanisme is de handhaving van causaliteit, d.i. oorzaak en gevolg. Op een sequentiële computer is dit redelijk eenvoudig te realiseren door de gebeurtenissen in de simulatie op toenemende simulatietijd te sorteren en steeds de gebeurtenis met de eerste simulatietijd te selecteren (de voorste in de rij van gesorteerde gebeurtenissen). Met de parallelle executie van asynchrone simulatiemodellen ligt dit anders. Hier worden gelijktijdig meerdere gebeurtenissen verwerkt door verschillende computers. Het bijhouden van één gesorteerde lijst met gebeurtenissen is niet effectief voor de klasse van asynchrone simulatiemodellen: de parallelle processoren zouden het merendeel van de tijd op elkaar moeten wachten eer zij een gebeurtenis kunnen verwerken. Bij de parallelle executie van asynchrone simulatiemodellen wordt de lijst met simulatiegebeurtenissen over de computers verspreid om zo de lokaliteit van data en verwerking van gebeurtenissen zoveel mogelijk uit te buiten. Maar omdat iedere computer zijn eigen gesorteerde lijst met gebeurtenissen heeft, kan het voorkomen dat de causaliteit in de simulatie geschonden wordt. Bijvoorbeeld: als computer A tijdens de simulatie een nieuwe gebeurtenis met simulatietijd 15 bij computer B wil plaatsen, kan het voorkomen dat computer B al een gebeurtenis met simulatietijd 20 aan het verwerken is. Als een dergelijke causaliteitsfout optreedt, wordt het zogenaamde "rollback and recovery" mechanisme in werking gezet. In het voorbeeld zou computer B zijn simulatietijd terugzetten naar de laatste gebeurtenis direct voor de nieuwe gebeurtenis met simulatietijd 15 ("rollback") en herstelt vervolgens alle veranderingen in de simulatie ten gevolge van de voorbarige verwerking van gebeurtenissen tussen simulatietijd 15 en 20 ("re-

covery”).

De Time Warp methode moet regelmatig de status van de simulatie bewaren om zich te kunnen herstellen van causaliteitsfouten. Bij grote systemen kan de volledige status van de simulaties zeer groot worden en het volledig geheugen in beslag nemen. Wij hebben een meer efficiënte methode geïntroduceerd die alleen de wijzigingen in de status bewaard. Deze incrementele methode is efficiënter in geheugen gebruik, maar tijdens de “recovery” fase moet de oude status van de simulatie stap voor stap gereconstrueerd worden. Verder wordt de APSIS omgeving gecompleteerd door de APSE parallelisme analyse methodiek, zie Hoofdstuk 4. De APSE parallelisme analyse is een waardevolle aanvulling op de APSIS simulatie omgeving, omdat het inzicht geeft in het intrinsieke parallelisme van het simulatiemodel en het parallelisme dat gerealiseerd is door APSIS. Daarbij kan door middel van kritieke pad analyse, de obstakels geïdentificeerd worden die het parallelisme in de simulatie limiteren.

De toepasbaarheid van de Time Warp simulatie-omgeving is geëvalueerd in Hoofdstuk 5. De applicatie is een Ising spin systeem, waarmee de magnetisatie van ferro-metalen bestudeerd kan worden. Ising spin systemen zijn een prototype voor een klasse van applicaties in statistische fysica. De experimenten en de APSE parallelisme analyse geven aan dat de APSIS omgeving in staat is de simulatiegebeurtenissen efficiënt te verwerken over de parallelle processoren. De experimenten tonen ook aan dat er een mate van optimisme controle nodig is. In Time Warp kunnen twee processoren een willekeurige simulatietijd uit elkaar lopen, bijvoorbeeld: simulatietijd 10 op processor A en simulatietijd 2040 op processor B. Als nu een nieuwe gebeurtenis met simulatietijd 15 door processor B verwerkt moet worden, moet processor B een “rollback en recovery” actie van simulatietijd 2040 naar simulatietijd 15 maken. Dergelijke lange rollbacks vereisen substantiële computerrekeningtijden en kunnen zich herhaaldelijk voordoen. Dit kan voorkomen worden door het beperken van het optimisme in de simulatie, bijvoorbeeld door een simulatietijdsinterval vast te stellen waarbinnen gebeurtenissen verwerkt mogen worden. In ons vorig voorbeeld met simulatietijd 10 op processor A en een simulatietijdsinterval van 100, zou processor B zijn gebeurtenissen niet verder dan simulatietijd 110 mogen verwerken.

De effectiviteit van het optimisme controle mechanisme wordt aangetoond in Hoofdstuk 6. In de Ising spin systeem simulatie treden (onverwacht) lange rollback series op rondom de fasetransitie van magneet naar paramagneet. De lengte en frequentie van de rollback series kunnen binnen grenzen gehouden worden door het optimisme controle mechanisme, resulterend in kortere responsietijden. De niet-triviale interferentie tussen Ising spin systeem applicatie en Time Warp methode is mogelijk zelforganiserend kritisch gedrag, waarbij de complexiteit van de berekingen en de complexiteit van de fysica op een niet-lineaire wijze met elkaar vermengd zijn.

De Dynamite omgeving is een experimenteel platform voor onze ideeën betreffende dynamische werkverdeling van rekentaken (processen) over de gedistribueerde of parallelle processoren. In Hoofdstuk 7 beschrijven we het onder-

liggende principe van proces migratie, waarbij de executie van een programma gestopt wordt, verplaatst naar een andere processor en weer opgestart wordt. Een complicerende factor is communicatie tussen parallelle processen. De proces migratie moet transparant zijn, zodat de andere parallelle processen geen notie hebben van de gewijzigde situatie. De transparante proces migratie in Dynamite is geëvalueerd aan de hand van een aantal experimenten met verschillende test applicaties en twee simulatie applicaties. De resultaten van de experimenten tonen de effectiviteit van dynamische proces migratie in Dynamite aan.

In de toekomst worden de APSIS en Dynamite omgeving met elkaar geïntegreerd. Het belangrijkste onderzoeksonderwerp bij de integratie van APSIS met de Dynamite omgeving, is de werkverdelings strategie voor optimistische simulaties over de parallelle processoren. In optimistische simulatie is er een verschil tussen rekenwerk en de het begrip voortgang. Bijvoorbeeld, "rollback en recovery" acties worden als rekenwerk aangemerkt, maar dragen niet bij aan de voortgang van de simulatie. Een van de belangrijkste onderzoeksvragen zal zijn: het vinden van een compacte beschrijving van het executiegedrag van optimistische simulaties en daarop gebaseerd, een effectieve strategie om werk te verdelen over de parallelle processoren.