



UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

User Transparent Parallel Image Processing

Seinstra, F.J.

Publication date
2003

[Link to publication](#)

Citation for published version (APA):

Seinstra, F. J. (2003). *User Transparent Parallel Image Processing*. [Thesis, fully internal, Universiteit van Amsterdam]. Febodruk BV.

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.

Samenvatting*

"Alle intellectuele beroepen bestaan uit het continu verrichten van dingen die, apart genomen, heel eenvoudig zijn."

Willem Frederik Hermans - *Nooit Meer Slapen* (1966)

Om te kunnen voldoen aan de specifieke prestatie-eisen van huidige en toekomstige beeldverwerkingsapplicaties, toont de beeldverwerkingsgemeenschap een grote wens de rekenkracht van parallele en gedistribueerde computersystemen aan te wenden. Helaas bestaat er vooralsnog een enorme discrepantie tussen de vraag naar *eenvoudig te verkrijgen* rekenkracht, en de wijze waarop de aanwezige rekenkracht ook daadwerkelijk wordt aangeboden. De kern van dit probleem bestaat eruit, dat voor beeldverwerkingsonderzoekers geen effectief en laagdrempelig hulpmiddel voor handen is voor het ontwikkelen van parallele applicaties. Bestaande hulpmiddelen vereisen van de gebruiker een inzicht in de complexiteit van parallellisme dat verder gaat dan van beginnende parallele programmeurs verwacht mag worden. Omdat het niet reëel is te eisen dat experts op het gebied van beeldverwerking tevens experts op het gebied van parallel programmeren worden, is het noodzakelijk dat een hulpmiddel wordt ontwikkeld dat gebruikers afschermt van *alle* parallellisatieproblematiek.

Het in dit proefschrift beschreven onderzoek poogt de kloof te dichten tussen de specifieke expertise van beeldverwerkingsonderzoekers, en de additionele kennis die vereist is voor het efficiënt aanwenden van parallele computers. Daartoe behandelt het proefschrift het ontwerp en de implementatie van een softwarearchitectuur die beeldverwerkingsonderzoekers in staat stelt parallele applicaties te ontwikkelen *op een voor de gebruiker volledig transparante wijze* (dat wil zeggen: volledig sequentieel). Het proefschrift onderzoekt de specifieke eisen die aan een dergelijke architectuur gesteld moeten worden, zodat het dienst kan doen als een voor beeldverwerkers acceptabele langetermijnoplossing. Daarnaast geeft het proefschrift een gedetailleerde verhandeling van de verschillende componenten van de ontwikkelde architectuur, alsmede van de daaraan gerelateerde onderzoeksvragen. Eveneens geeft het proefschrift een uitgebreide evaluatie van de door de architectuur geleverde snelheidswinsten, gecombineerd met een vergelijking met handgeoptimaliseerde applicaties.

*Summary in Dutch

Hoofdstuk 2 onderzoekt de mate waarin bestaande hardwarearchitecturen, en daaraan gerelateerde programmeerhulpmiddelen, geschikt zijn voor toepassing in beeldverwerkingsonderzoek. Aan de hand van een lijst van specifieke eisen wordt geconcludeerd dat de klasse van Beowulf-clusters het meest geschikt is, voornamelijk vanwege de positieve prijs-prestatieverhouding. De evaluatie van programmeerhulpmiddelen toont aan dat bibliotheekgebaseerde architecturen een oplossing bieden die het meest aansluit bij de specifieke wensen van de beeldverwerkingsgemeenschap. Dit komt voornamelijk doordat dergelijke architecturen op een redelijk eenvoudige wijze geleverd kunnen worden met een programmeermodel dat de gebruiker volledige transparantie biedt. Echter, vanwege gebrekkige *onderhoudbaarheid* bestaat er op dit moment geen architectuur die ook daadwerkelijk beschouwd kan worden als een acceptabele langetermijnoplossing. Op basis van deze observaties wordt een nieuwe bibliotheekgebaseerde architectuur voor parallel beeldverwerking geïntroduceerd. Door het innovatieve ontwerp voldoet de architectuur zowel aan de eis van onderhoudbaarheid, als aan de eis van gebruikerstransparantie.

Hoofdstuk 3 beschrijft de ontwerpfilosofie van de bibliotheek, die de kern vormt van de ontwikkelde architectuur. Centraal staat de vraag hoe de bibliotheek zodanig geïmplementeerd kan worden dat coderedundantie zoveel mogelijk wordt vermeden, met behoud van efficiëntie. Daartoe wordt het begrip *parallelliseerbare patronen* geïntroduceerd, en getoond hoe parallele implementaties eenvoudig verkregen kunnen worden door concatenatie van operaties die alle *apart* aanwezig zijn in de bibliotheek. Op basis van vier verschillende typen van gegevensbenadering wordt een standaard parallellisatiestrategie gedefinieerd voor elke operatie die overeenkomt met een van de parallelliseerbare patronen. De standaard parallellisatiestrategie is te allen tijde optimaal, omdat het al het aanwezige parallellisme volledig benut met een minimum aan communicatiekosten. De kenmerken van de gepresenteerde ontwerpfilosofie verzekeren langetermijnonderhoudbaarheid van de architectuur, alsook efficiëntie.

Hoofdstuk 4 geeft aan hoe een eenvoudig analytisch prestatiemodel toegepast wordt in het proces van automatische parallellisatie en optimalisatie van beeldverwerkingsapplicaties. Bestaande modellen bevatten in het algemeen een directe complexiteit-nauwkeurigheidrelatie, en zijn daarom voor de gewenste accuratesse niet efficiënt genoeg. Om dit probleem het hoofd te bieden wordt een semi-empirische modelleertechniek geïntroduceerd, die naast eenvoud en overdraagbaarheid tevens voldoende nauwkeurigheid biedt voor de gegeven doelstelling. De techniek is gebaseerd op de definitie van een abstracte machine (*APIPM*), die de typische gedragingen van parallele beeldverwerkingsoperaties beschrijft. Op basis van de gerelateerde APIPM-instructieset wordt een prestatiemodel verkregen dat algemeen toepasbaar is op Beowulf-clusters. De essentie van de modelleertechniek bestaat eruit, dat van bepalende kostenfactoren die Beowulf-clusters niet alle gemeen hebben geabstraheerd wordt in de definitie van de modelparameters. Om elke abstracte modelparameter alsnog te verbinden met een prestatieschatting voor een concrete parallele machine, wordt een klein aantal tijdmetingen uitgevoerd die alle impliciete, doch essentiële, kostenfactoren ondervangen. Een vergelijking van gegenereerde schattingen met experimentele resultaten toont de hoge nauwkeurigheid van de APIPM-gebaseerde modellen voor realistische applicaties.

Hoofdstuk 5 breidt de APIPM-gebaseerde modellen uit om nauwkeuriger schattingen te verkrijgen voor de MPI-primitieven die in de bibliotheek worden toegepast. Bestaande communicatiemodellen (zoals LogP) ondervangen niet alle mogelijke gedragingen van de MPI-operaties. Tot nu toe is het (vaak zeer significante) effect van de schikking van gegevens in het geheugen op de communicatiekosten volledig genegeerd. In de ontwikkelde architectuur is een hogere nauwkeurigheid essentieel om de automatische distributie van beeldgegevensstructuren optimaal uit te voeren. Daarom wordt het nieuwe P-3PC-model geïntroduceerd, dat het typische gedrag van de MPI-primitieven juist representeert. Ten eerste ondervangt het de verschillen in communicatiekosten voor de zender, de ontvanger, alsook het volledige communicatiepad. Daarnaast modelleert het het effect van geheugenschikking, en ondervangt het kosten die niet-lineair afhankelijk zijn van de berichtgrootte. Experimenten die zijn uitgevoerd op twee verschillende Beowulf-clusters tonen aan dat, in vergelijking met gerelateerde modellen, P-3PC in staat is nauwkeuriger schattingen te leveren voor de communicatiekosten zoals die bestaan in typische beeldverwerkingsapplicaties.

Hoofdstuk 6 behandelt de automatische conversie van elke sequentiële beeldverwerkingsapplicatie naar een correcte en efficiënte parallele versie. Daartoe wordt een eindige automaat gedefinieerd, die garandeert dat dit proces te allen tijde correct wordt uitgevoerd. Ten eerste wordt aangetoond dat de automaat leidt tot een opvallend eenvoudige methode voor minimalisatie van de communicatiekosten. Voor verdere prestatieoptimalisaties wordt de automaat toegepast in de constructie van een graaf, die alle mogelijke toestandsveranderingen en parallellisatiestrategieën van een draaiende applicatie weergeeft. Omdat voor elke mogelijke strategie een APIPM-gebaseerde kostenschatting voor handen is, wordt de snelste versie van een applicatie weergegeven door het goedkoopste pad in de graaf. Vanwege het feit dat de automatische optimalisatie van volledige applicaties het centrale, meest essentiële, probleem vormt waarmee de ontwikkelde softwarearchitectuur wordt geconfronteerd, combineert de toegepaste oplossing alle resultaten zoals bereikt in de Hoofdstukken 3, 4, en 5.

Hoofdstuk 7 geeft een inschatting van de effectiviteit van de softwarearchitectuur in het behalen van significante prestatieverbeteringen. Daartoe wordt een beschrijving gegeven van de sequentiële implementatie en automatische parallellisatie van drie voorbeeldapplicaties, die alle veelvoorkomende beeldverwerkingsoperaties bevatten. Op basis van de evaluatie wordt geconcludeerd dat de snelheidswinsten die behaald worden door de parallele applicaties zoals gegenereerd door de ontwikkelde softwarearchitectuur vergelijkbaar zijn met handgeoptimaliseerde implementaties.

Conclusie: de resultaten bereikt met de ontwikkelde softwarearchitectuur tonen aan dat het mogelijk is een oplossing te ontwikkelen voor een ingewikkeld probleem, voornamelijk door het combineren van relevante abstracties, en toepassing van relatief eenvoudige methoden voor de oplossing van elk deelprobleem. Bijgevolg lijkt het er sterk op dat een vergelijkbare strategie toepasbaar is op andere onderzoeksterreinen, voornamelijk daar waar het aantal typische operaties gelimiteerd is — net zoals dat het geval is in beeldverwerkingsonderzoek. Desalniettemin kan alleen de tijd leren of de ontwikkelde softwarearchitectuur ook daadwerkelijk de door ons vermoedde effectieve langetermijnoplossing zal zijn.

