



Universiteit
Leiden
The Netherlands

Transformations for polyhedral process networks

Meijer, S.

Citation

Meijer, S. (2010, December 8). *Transformations for polyhedral process networks*. Retrieved from <https://hdl.handle.net/1887/16221>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/16221>

Note: To cite this publication please use the final published version (if applicable).

Samenvatting

Deze dissertatie beschrijft methoden en technieken voor het analyseren en programmeren van multiprocessor systemen die zijn geïntegreerd in een enkele chip. We richten ons voornamelijk op applicaties voor de verwerking van signalen en beelden in ingebedde multimedia toepassingen. Deze toepassingen kunnen het best worden gekarakteriseerd als een verzameling van rekentaken die data uitwisselen in de vorm van datastromen. In de meeste van deze toepassingen zijn doorstromingsnelheden van cruciaal belang, waardoor rekentaken snel en, indien mogelijk, gelijktijdig moeten worden uitgevoerd. Deze eisen leiden vanzelf tot implementatiestructuren die bestaan uit meerdere, vaak ongelijke, processoren die autonoom rekenen en zijn aangesloten op een communicatie-, synchronisatie-, en geheugeninfrastructuur voor de uitwisseling van data. De complexiteit van zulke ingebedde multi-processor systemen heeft een niveau bereikt waardoor het noodzakelijk is geworden om het programmeren van deze systemen op systematische en automatische wijze uit te voeren.

Voor het efficiënt programmeren van multi-processor systemen heeft het Leiden Embedded Research Center (LERC) een ontwerpmethodologie ontwikkeld die uitgaat van twee principes. Het eerste is gebaseerd op het feit dat toepassingen gespecificeerd worden in termen van datastroom procesnetwerken, in het bijzonder *Polyhedral Proces Netwerken* (PPN), die goed passen bij de beoogde datastroom applicaties. Hierdoor is een ontwerper veel beter in staat, in tegenstelling tot monolitische en sequentiële applicatiebeschrijvingen, om autonome taken toe te kennen aan verschillende processoren van het multi-processor systeem. Het tweede principe heeft als doel multi-processor systemen te creëren die naadloos aansluiten op de eigenschappen van de stroomgebaseerde toepassingen, waardoor de applicaties zo efficiënt mogelijk uitgevoerd kunnen worden. Deze ontwerptechnieken worden volledig ondersteund door het vertaalprogramma Daedalus. Dit is een vertaler die drie hoog-niveau beschrijvingen (de applicatie, het multi-processor systeem, en de toekenning van applicatietaken aan rekeneenheden van het multi-processor systeem) automatisch

omzet naar een laagniveau beschrijving van het systeem. Dit stelt een ontwerper in staat om op volledig automatische wijze een applicatie te implementeren op een multi-processor systeem.

Deze dissertatie richt zich op de beschrijving van applicaties in de vorm van een Polyhedral Proces Netwerk (PPN), en dan met name op het omvormen van PPNs. Het probleem is dat PPNs automatisch afgeleid kunnen worden, maar niet noodzakelijk tot de gewenste doorstroomsnelheden leiden. Het omvormen van het PPN is dan noodzakelijk om het gewenste resultaat te bereiken. Het omvormen van een PPN kan op twee manieren gebeuren: een proces uit het PPN kan opgesplitst worden in meerdere parallele processen, of meerdere processen kunnen samengevoegd worden in één samengesteld proces. In het eerste geval, spreken we van de *process splitting* transformatie die toegepast wordt om de applicatie te versnellen, en in het tweede geval spreken we van de *process merging* transformatie dat toegepast wordt om het aantal processen in het PPN te verminderen indien nodig. Het probleem bestond eruit dat beide transformaties wel gedefinieerd waren, maar de ontwerper wist niet precies hoe deze het best toegepast konden worden. Er zijn namelijk vele verschillende mogelijkheden waarop een bepaalde transformatie toegepast kan worden, en vele verschillende factoren spelen een rol in de uiteindelijke doorstroomsnelheden van applicaties. Om de ontwerper te helpen met het zo efficiënt mogelijk toepassen van transformaties, benoemen we in hoofdstuk 3 de factoren die belangrijk zijn voor de process splitting transformatie, hoe deze geëvalueerd kunnen worden, en een aanpak voor het kiezen van de beste transformatie. In hoofdstuk 4 doen we hetzelfde, maar dan voor de process merging transformatie. Deze analyse is wezenlijk anders dan de process splitting transformatie, omdat het niet lokaal uitgevoerd wordt zoals bij de process splitting, maar globaal voor het hele PPN. Dat wil zeggen dat we voor het samenvoegen van processen een model voor de doorstroomsnelheid definiëren. Dit stelt de ontwerper in staat om een transformatie op een bepaalde manier uit te voeren, de doorstroomsnelheid te evaluëren, en het beste alternatief te kiezen. Daarnaast presenteren we in hoofdstuk 5 een aanpak die beide transformaties combineert. Hierdoor lossen we het probleem op dat de transformaties op vele verschillende mogelijkheden achter elkaar toegepast kunnen worden (in verschillende volgorde). Tenslotte presenteren we in hoofdstuk 6 technieken om PPNs op multi-processor systemen uit te voeren. We beschrijven technieken voor het afbeelden van de verschillende elementen van PPNs op de Intel IXP network processor en de Cell platform.