



Universiteit
Leiden
The Netherlands

Generalized strictly periodic scheduling analysis, resource optimization, and implementation of adaptive streaming applications

Niknam, S.

Citation

Niknam, S. (2020, August 25). *Generalized strictly periodic scheduling analysis, resource optimization, and implementation of adaptive streaming applications*. Retrieved from <https://hdl.handle.net/1887/135946>

Version: 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/1887/135946>

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

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/135946> holds various files of this Leiden University dissertation.

Author: Niknam, S.

Title: Generalized strictly periodic scheduling analysis, resource optimization, and implementation of adaptive streaming applications

Issue Date: 2020-08-25

Samenvatting

Het doel van dit proefschrift is het oplossen van vier onderzoeksproblemen bij het ontwerpen embedded streaming-systemen. Embedded streaming-systemen zijn die systemen die een stroom invoergegevens uit de omgeving verwerken en een stroom van uitvoergegevens genereren voor deze omgeving. Voor velen van deze ingebedde streaming-systemen is de timing een kritische ontwerpvereiste, waarbij correct gedrag afhangt van zowel de juistheid van uitvoergegevens als van het tijdstip waarop de gegevens worden geproduceerd. Een embedded streaming-systeem onderworpen naar zo'n timingvereiste wordt een real-time systeem genoemd. Enkele voorbeelden van real-time embedded streaming-systemen zijn te vinden in verschillende autonome mobiele systemen, zoals vliegtuigen, zelfrijdende auto's en drones.

Om aan de strakke timingvereisten van dergelijke real-time embedded streaming-systemen te kunnen voldoen zijn moderne embedded systemen uitgerust met hardware platforms, de zogenaamde Multi-Processor Systems-on-Chip (MPSoC), die meerdere processors, geheugens, verbindingen en andere hardware-randapparatuur op een enkele chip samenvoegen, om zo te kunnen profiteren van parallele executie. Om de rekencapaciteit van een MPSoC-platform te kunnen benutten moet een streaming-applicatie, die wordt uitgevoerd op het MPSoC-platform, worden beschreven op een parallele wijze, d.w.z. de applicatie wordt gedefinieerd als een set van parallele taken die met elkaar communiceren. De belangrijkste uitdaging is om deze taken ruimtelijk te plannen, d.w.z. de afbeelding van taken op processors, en temporeel, d.w.z. de volgorde van de taakplanning, op het MPSoC-platform zodat aan alle timingvereisten wordt voldaan met een efficiënt gebruik van de beschikbare middelen (de processors, geheugen, energie, etc.) op het platform. Een andere uitdaging is hoe deze toegewezen en geplande applicatietaken op de MPSoC te implementeren en uit te voeren op het platform. Dit proefschrift stelt verschillende technieken voor om de twee bovengenoemde uitdagingen op te lossen.

In het eerste deel van het proefschrift ligt de focus op de eerste boven-

genoemde uitdaging bij het ontwerpen van embedded streaming-systemen. Hier wordt een methode geïntroduceerd om de data-afhankelijke taken in een applicatie, inclusief cyclische data-afhankelijke taken, om te zetten naar real-time periodieke taken. Dit maakt het mogelijk om een verscheidenheid aan harde realtime planning algoritmen voor periodieke taken, van de klassieke real-time planning theorie, toe te passen om dergelijke streamingtoepassingen te plannen met bepaalde gegarandeerde prestaties voor doorvoer en reactietijd. Deze algoritmen kunnen snelle toegangscontrole en planningsbeslissingen uitvoeren voor nieuwe inkomende applicaties in een MPSoC-platform en bieden eigenschappen zoals temporele isolatie en snelle analytische berekening van het minimum aantal processors dat nodig is voor het uitvoeren van de taken in de applicatie.

In het tweede deel van het proefschrift ligt de focus op het efficiënt gebruik maken van componenten op een onderliggend MPSoC-platform bij het plannen van de taken van applicaties op het platform. We introduceren een algoritme om een eerste representatie van een streamingapplicatie, d.w.z. een initiële applicatie graaf, te transformeren in een functioneel equivalente applicatie graaf die minder processors nodig heeft om de gegeven doorvoervereiste te garanderen. Daarnaast onderzoekt dit proefschrift het probleem van energiezuinige planning van streaming-applicaties met doorvoer vereisten op MPSoC-platforms met spannings- en frequentieschaling mogelijkheden. Hiervoor wordt er een nieuw periodiek planningskader geïntroduceerd waarin streaming-applicaties hun uitvoering periodiek kunnen variëren tussen een aantal energiezuinige schema's tijdens runtime om te voldoen aan een doorvoervereiste op de lange termijn. Met behulp van een dergelijke periodieke omschakeling kunnen systeemontwerpers profiteren van het gebruik van dynamische spanning en frequentieschalingstechnieken om de beschikbare extra spelingsijd in het schema van een applicatie efficiënt te gebruiken.

Tot slot, in het derde deel van het proefschrift, ligt de focus op de tweede bovengenoemde uitdaging in het ontwerp van embedded streaming-systemen. Hiervoor wordt een generieke parallelle implementatie- en uitvoeringsmethode voor (adaptieve) streaming-applicaties voorgesteld. De voorgestelde methode kan gemakkelijk kunnen worden gerealiseerd bovenop bestaande besturingssystemen en in combinatie met een breder scala aan taakplanningsmethoden. Een demonstratie van de voorgestelde aanpak voor LITMUS^{RT}, een bestaande real-time uitbreidingen van de Linux-kernel, toont de haalbaarheid aan van deze methode.