Beiträge aus der Informationstechnik

Mobile Nachrichtenübertragung
Nr. 100

Nairuhi Grigoryan

Modeling and Performance Estimation for Multiprocessor System On Chip Architectures

VOGT

Dresden 2023

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Bibliographic Information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.dnb.de.

Zugl.: Dresden, Techn. Univ., Diss., 2022

Die vorliegende Arbeit stimmt mit dem Original der Dissertation „Modeling and Performance Estimation for Multiprocessor System On Chip Architectures“ von Nairuhi Grigoryan überein.

Gesetzt vom Autor

ISBN 978-3-95947-064-3

Jörg Vogt Verlag
Niederwaldstr. 36
01277 Dresden
Germany

Phone: +49-(0)351-31403921
Telefax: +49-(0)351-31403918
e-mail: info@vogtverlag.de
Internet : www.vogtverlag.de

Technische Universität Dresden

Modeling and Performance Estimation for Multiprocessor System On Chip Architectures

M. Sc.

Nairuhi Grigoryan

der Fakultät Elektrotechnik und Informationstechnik der Technischen Universität Dresden

zur Erlangung des akademischen Grades

Doktoringenieur

(Dr.-Ing.)

genehmigte Dissertation

Vorsitzender:	Prof. Dr.-Ing. habil. Christian Georg Mayr
Gutachter:	Prof. Dr.-Ing. Dr. h.c. Gerhard Fettweis
	Prof. Dr.-Ing. Holger Blume

Tag der Einreichung: 28.06.2022
Tag der Verteidigung: 20.12.2022

Nairuhi Grigoryan
Modeling and Performance Estimation for Multiprocessor System On Chip Architectures
Vodafone Chair Mobile Communications Systems
Institut für Nachrichtentechnik
Fakultät Elektrotechnik und Informationstechnik
Technische Universität Dresden
01062 Dresden

Abstract

Nowadays mobile operators face several challenges because of large growth of internet traffic usage, the increase of number of users. There is a need of innovations in almost every area such as e-health, private businesses, automation, etc. 5G supports the variety of new services with different requirements for throughput, latency and reliability. Multicore computing platforms are used to meet the various implementations while allowing scalability and flexibility in the implementation of the base stations. The challenge in this regards is the efficient distribution and processing of signal processing tasks on parallel processors. Multiple processing elements give opportunity to exploit application parallelism by splitting them into many parallel tasks and make the parallel execution possible. But unfortunately these methods also have disadvantages. Moreover, with increasing of the application complexity, the management and synchronization overhead increases disproportionately, which limits the increase in performance and system efficiency. The performance of application highly depends on the methods used to execute it. The overheads resulting from synchronization, management and waiting time in a queue sometimes are not solvable only by choosing the right scheduling algorithm. Therefore another solution is needed. To cope with this problem, the application granularity reduction using task clustering was proposed recently and demonstrated impressive performance improvement. Clustering application before applying scheduling algorithm reduces the granularity of the application and solves the problem concerning synchronization and management overheads. There are different clustering algorithms to be chosen, but they are non deterministic and have high interconnection between tasks from various clusters. Our motivation for this work was to find a clustering algorithm which will suit our problem better and will result in better performance. We present a modification of the CASS-II clustering algorithm which proves to be a satisfactory solution for our applications. In order to analyze and simulate different graphs, we also developed a simulation tool, which enables to run different algorithms for scheduling and clustering very efficiently. It accepts any kind of graphs, and the results can be later analyzed via easy formed visualization functions. We also covered another important topic that is energy consumption. Power control is one of the most important topics in many communication and computation environments. Heat generation, expensive packing and cooling can be result of high power generation. Many researchers are engaged in this problem and suggest various solutions.

Energy-delay tradeoff is examined as a method for energy saving. In this work we investigate dynamic voltage and frequency scaling and suggest a new modified method called Proportional Task Scaling. In order to change energy consumption, the frequency and voltage should be changed together. There are different algorithms that suggest to reduce these parameters differently. The greedy static power management algorithm suggests to reduce frequency and increase the execution time of the first tasks on processing elements. Most of the algorithms use the slack value(difference of deadline and scheduling length) to make this change happen. Our motivation is to reduce even more energy consumption by still meeting the deadline of the application. So, our suggestion is to change parameters not only for the first but the whole application by meeting scheduling requirements and deadline.

Abbreviations

5G	fifth generation
CASS-II	Clustering And Scheduling System II
CP	Critical Path
CRAN	Cloud Random Access Network
DAG	Directed Acyclic Graph
DFE	Data FLow Engine
DLS	Dynamic Level Scheduling
DSP	Digital Signal Processor
DVFS	Dynamic Voltage Frequency Scaling
G-SPM	Greedy Static Power Management
HLFET	Highest Level First with Estimated Time
PE	Processing Element
PHY	Physical Layer
RAN	Radio Access Network
SDN	Sofwtare Defined Network
SDR	Software Defined Radio
TDCA	Task Duplication based Clustering Algorithm

Notations

w	task execution time
CC	communication cost
f	frequency
V	voltage
C	capacitance
$P(f)$	power consumption
P_d	frequency dependent power consumption
P_{ind}	frequency independent power consumption
E	energy consumption
G	graph
sl	slack value
p	processing element
m	makespan of a graph
CP	Critical path of a graph
v	task
$\mu(v)$	execution time of task v
$\lambda(u, v)$	communication cost between tasks u and v
$s(v)$	s-value of task v
$f(v)$	f-value of task v
$b-level(v)$	b-level of task v
$t-level(v)$	t-level of task v

List of Figures

List of Tables

Contents

1 Introduction

1.1 Motivation and Problem Statement

The extreme growth of wireless devices brings with it the need to act respectively. During the last years, cellular operators faced severe challenges such as increasing growth of users and also needs of the users. The data that is transferred and received increases each day. Hardware therefore, should be configured and reconfigured on the go, but installing new hardware to meet all the new standards is always expensive. Also, the cellular operators need to install new base stations to cover the volume of these data. One answer to this growing market was CRAN, which was first developed in 2009 in China [Che+14]. CRAN is a flexible and adaptable platform which enable the operators to adapt easily to these growing market. The operators don't need to install many new hardware units, as they just need to make changes in software. We are witnessing that new technologies transfer high bandwidth applications with high quality of service. With the use of CRANs, the further problem of parallelization can be resolved. The CRAN approach brings notable improvements in capital and operational expenditures (CAPEX and OPEX), as well it makes the operating techniques much easier [KG12]. When implementing CRAN the need for expensive hardware and also renting real estate decreases. Most of the actions are transferred into cloud. CRAN can adapt to the frequent change of network characteristics and utilize the resources more efficiently. Many companies have developed CRAN implementations. They have different parameters and are equipped with different usage modules. The problem here is that most of them provide it as a black box, which has very low level of interference, so the users cannot change the applications easily on software. They are highly optimized for some types of platform but are not implementable to others. So, the challenge is also to provide scalability and efficient use of hardware with changeable implementation of software. Within the scope of this work, we provide one way to gain scalability and flexibility. We provide a method for implementing CRAN that allows also the implementation of different algorithms for task scheduling in CRAN application.

Considering the aformentioned we can also describe the requirements of 5G and beyond:

- Functional split - the partitioning of protocol stack and mapping partitions often to geographically separated processing unit.
- Temporal and spatial parallel processing to cope with huge computational requirements.
- Flexibility - diversity of operation and transmission mode.
- Scalability - highly dynamic (time variable) workload.
- Efficiency - efficient use of processing resources.

The growth in number of users and the increase in data volume is aligned with social development, which in turns brings the need of improvement of other aspects. All these require innovations for private businesses, automation and even e-health. The composite scenarios create other challenging problems for mobile network operators, such as delivering all the necessary data with high speeds and quality. As mentioned in [Ros+14] it will be very difficult to make all the changes happen using only some fixed equipment. What is needed is reliable and flexible software (cloud) which will enable such significant changes in short time. It can be said that flexibility, openness, scalability and also energy efficiency should be the main criteria for the new networks. Cloud technologies have already been used in different areas in order to make the storing and processing easier. This technology is also used nowadays by mobile operators and enables them to answer to many problems described above. 5G networks make these new improvements real. 5G supports different services and applications with different throughput, latency and reliability requirements. For this reason, the implementation of many-core programmable computing platforms is very preferable. Many-core programmable platforms help solve problems concerning design and performance of the networks. There is a great variety of models that help to process modem signals with different characteristics. Many-core platforms give opportunity to exploit the parallelism of the application by partitioning or splitting it to parallel tasks and hence enabling simultaneous execution on multiple cores. By increasing number of the cores in the architecture, maximum possible parallelism can be met easily. But unfortunately all these methods bring also the problem of overheads associated with task synchronization and management, and inherent parallelism which is application specific and usually limited. The scheduling results of the applications highly depend on the methods used to execute the tasks. These overheads limit the flexibility and scalability of the applications. One of the ways to overcome these overheads is applying clustering to the application before scheduling it into processing elements. Clustering enables solving the problem

concerning synchronization and management overheads, and provides better results after scheduling the application.

Another important aspect is energy consumption. Nowadays modern computing platforms consume huge amount of energy during execution. The problem with power consumption has attracted already many research groups to reduce it by changing some parameters of application or architecture. There are two approaches that are used to reduce energy consumption powergating as well as dynamic voltage and frequency scaling [Li16a]. Power gating is switching on/off voltage resources and is well suited for scenarios with slow workload variation(time and energy overhead for switching on/off) and DVFS(Dynamic Voltage and Frequency Scaling) which is well suited for fast workload variation, which is our motivation for choosing this approach for later investigations. By changing dynamic voltage and frequency of the application, we can also change the consumed energy. In some cases we can get better execution results of the application by running it on fully loaded processing elements, and in other cases we not only can meet the required deadlines but also gain in energy consumption. It is much better to come up with the solution for joint minimization of application execution time and energy consumption [Li16a]. For reduction of power consumption, there are also some other techniques suggested such as shutting down some processing elements that are not used or reducing power of partially loaded ones [ZMC03a]. But shutting down and then again turning the processing elements on needs some time for loading and unloading data, which will cause more overhead in its turn. So the algorithms that are designed for energy efficiency and at the same time good performance of the application are better choice. As it was already mentioned above, dynamic voltage scaling is the technique that makes possible to change the power of processing elements during run-time, and which enables to reach energy efficiency without loosing in performance of the overall application.

1.2 Thesis Structure

Chapter 2

Chapter 2 describes the background information of the computational model for multiprocessor signal processing, basics of scheduling and clustering algorithms. The application is described by directed acyclic graphs(DAG), where the tasks are connected by communication costs and cannot be executed until the predecessor nodes finished. Before applying scheduling algorithms the tasks are given some priority levels, which are also described in this chapter. In order to have a better

execution, different scheduling algorithms, which include static and dynamic, are described. For even better results, we apply also clustering algorithms, which minimize the overheads caused by synchronization and management. The details of clustering algorithms as well as the description of overheads can be found here.

Chapter 3

The answer for better performance, which includes scalability and efficient use of hardware resources, predictability and flexibility, is Cloud RAN. The various implementations of CRAN, as well as use cases are described in chapter 3. Here also detailed information can be found about dataflow architectures and system concepts. As it was already described, the application is represented by directed acyclic graph, all this information about the tasks and connectivity can be found also in this chapter. Scheduling the application can be done with a special algorithm or manually. The different results can be found in simulation results section.

Chapter 4

For meeting the growing customer needs, the network operators should design and implement a network with high reliability, low latency and flexibility. For this reason, the multi-core architectures are desirable. Implementation of multi-core architecture is beneficial in terms of short makespan, but it also causes overheads (synchronization and management). One of the answers to this issue is applying a clustering algorithm, and one of the best clustering algorithms is CASS-II, which is described in chapter 4. In order to satisfy all of the needs for our application, we described the new modified version of previously mentioned CASS-II algorithm. From simulation results, it can be drawn when it is better to use each algorithm. Scheduling application also onto heterogeneous architecture is also illustrated here.

Chapter 5

Power and energy management is one of the important aspects in mobile communications. For many architectures, power control is crucial. So the reduction of energy consumption and meeting the deadlines are both critical when designing and executing an application. There are many different ways to manage energy consumption; one of them is dynamic voltage and frequency scaling, which is described in this chapter. Also the greedy static power management algorithm with some modification is presented in details for reaching best energy efficiency and scheduling length pair. In the section of simulation results can be seen that after applying the DVFS method the energy consumption is decreased. However, applying the new clustering methods additionally increases the energy savings significantly. All these methods and steps are described in detail in chapter 5.

Chapter 6

In chapter 6, the conclusion of the dissertation is given and a summary of the findings is presented.