学 位 論 文 内 容 の 要 旨

博士の専攻分野の名称　博士（工学）　氏名　福田エリック駿

学 位 論 文 題 名

A Study on Acceleration Methods of Data Center Applications with Reconfigurable Hardware

（リコンフィギュラブルハードウェアによるデータセンターアプリケーション高速化手法に関する研究）

This thesis is about how to make reconfigurable hardware easy to use for software developers. It has always been difficult for software developers to use and gain high performance with reconfigurable hardware. However, reconfigurable hardware is one of the promising technologies for overcoming the difficulties that general purpose processors, which have been the central concern of computer science for decades, are facing. Enabling reconfigurable technology to be used by varieties of people, including software developers, will have a great impact on pushing the current computing to a new era.

Computers have been developed in view of three metrics: 1) computation speed, 2) energy efficiency, and 3) how easy it is to develop software. Although computers have progressed enormous degrees in these directions, the physical restriction is beginning to prevent them to be further achieved.

The computer community has to find different ways of improving processors. Among the several alternatives that have been proposed, reconfigurable hardware is one of the most possible choice and this is what we try to utilize in this thesis. Reconfigurable hardware promises higher performance and energy efficiency in many application domains, but several problems must be solved before it can be widely utilized. There are two major problems: a) developing applications with reconfigurable hardware more difficult compared to software applications and thus b) the cost of developing the application becomes higher.

In order to solve these problems, we assess where the difficulties lie in the state of the art design method that uses high-level synthesis tool for developing hardware accelerated application systems, and then we propose two methods to overcome the difficulties.

First, we analyze the difficulties when a software developer tries to develop a dedicated hardware with an HLS tool, which is the most straight forward approach for a software developer to develop a hardware. In particular, we implement an operation called window join, a element operator of stream processing, on DRP with CyberWorkBench (CWB). As a result of our step by step implementation, we achieved 2 times of throughput and 19 times of power efficiency compared to software. We further propose five awarenesses that a software developer should have to develop a hardware: 1) I/O, 2) buffer, 3) resource amount, 4) loop, and 5) resource type.

Next, we propose a method to develop an application specific hardware by parsing the code written in the target application specific language to software code that intends high-level synthesis, and then synthesizing hardware with an HLS tool. We take StreamSQL, a description language specific to stream processing, as an example for our method and propose a parser that converts StreamSQL queries to C code that intends to be synthesized to hardware configuration with an HLS tool. This method enables application developers to develop hardware by StreamSQL without having any knowledge of hardware development. The implemented system yeiled twice as high throughput as software and extracted 90% of the chip's I/O band width.

Furthermore, we propose a method to accelerate the performance as a system by processing the data that goes into and comes out of the software that is running on a general purpose processor with a reconfigurable hardware while making no modifications to the software. We accelerate a server that runs memcached, an on-memory key-value store, by caching its functionalities and data to its NIC (Network Interface Card) that is equipped with an FPGA and DRAM. This method does not require any modification to the software memcached, therefore memcached servers that are already in operation can be enhanced. The implemented system showed 10-fold improvement of latency.

The research on enabling software developers to develop and utilize application specific hardware is still on its way. There must be more efforts to be done before a general method to generate dedicated hardware systems to various data center applications. Based on our discoveries in this thesis, we will continue to invent a way to make more developers to utilize hardware systems to accelerate their applications.