

Overview and Analysis of GPU Acceleration for Regular Expressions *

Roman Gajdoš

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
`xgajdosr@stuba.sk`

25. september 2023

Abstract

...

1 Introduction

Pattern matching is widely used in a variety of different domains. Regular expressions have become a prevalent tool for text processing and sanitation due to their flexibility, conciseness, and vast support in most programming languages. They appear in approximately a third of open-source projects. They are employed in technical fields, ranging from database querying, programming language compiling, and web scraping (the process of extracting data or information from internet sites) to network security, such as deep packet inspection, and bioinformatics, among others.

Regular expressions are implemented using finite automata, in either deterministic (DFA) or non-deterministic (NFA) form, each with their respective advantages and drawbacks. Each of them has their own advantages and disadvantages.

In many applications, regular expressions are applied to large amounts of input data, or require a fast response, or both. It stands to reason that efficiency in both memory and speed is the key to optimal use. Here, the question is how to achieve the greatest possible efficiency for a given problem that is addressed by the regular expressions. The processor's capacity to execute multiple expressions simultaneously is notably restricted, even in the current era of multicore processors. However, its frequency and cache memory speed prove excellent for handling small datasets. For tasks that require more extensive parallelism, FPGAs (Field-Programmable Gate Arrays) or ASICs (Application-Specific Integrated Circuits) have been used. The problem is that they are slow to configure and inflexible to change.

In recent years, GPUs with their extensive parallelism, computational capabilities, and high memory bandwidth have become prevalent in numerous

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2023/24, vedenie: MSc. Mirwais Ahmadzai

computing systems. They have scaled at a faster rate than CPUs, providing significant computing power. APIs were created to allow General Purpose Graphics Processing Units (GPGPU) to accelerate processing in supported applications, replacing shading languages and simplifying their use for programmers. Two popular APIs are Compute Unified Device Architecture (CUDA) and Open Computing Language (OpenCL).

In this paper, we investigate a variety of GPU-based regular expression execution methods and conduct a comparative analysis of their strengths and weaknesses. Our research begins with a thorough examination of regular expressions in 2.1. This is followed by a comparison of their representations of finite state automata forms in 2.1.1. We then move into parallel computing platforms, such as CUDA and OpenCL in 2.2.

By combining these findings, our investigation aims to provide a comprehensive overview of GPU-accelerated regular expressions, utilizing previous studies to provide an in-depth comparative analysis in section 3.

2 Background

2.1 Regular Expressions

2.1.1 Finite Automata

2.2 Parallel computing platforms

2.2.1 CUDA

2.2.2 OpenCL

3 Existing solutions

4 Conclusions