CogX Compiler 4.1

Massively scalable computation made easy

Internal

Maintenance

Specification

13 May 2016





Table of Contents

[1. Introduction 3](#_Toc402803863)

[2. Overview 4](#_Toc402803864)

[2.1 Compilation Phases (Parsing, Semantic Checking, Code Generation, Optimization) 4](#_Toc402803865)

[2.2 Primary Data Structures (SyntaxTree, KernelCircuit) 4](#_Toc402803866)

[3. Parser and Semantic Checking 5](#_Toc402803867)

[3.1 Opcode classes 5](#_Toc402803868)

[3.2 Operator classes 5](#_Toc402803869)

[3.3 GPUOperators 5](#_Toc402803870)

[4. Code Generation 6](#_Toc402803871)

[4.1 Purpose 6](#_Toc402803872)

[4.2 Generators 6](#_Toc402803873)

[5. Optimizers 7](#_Toc402803874)

[5.1 Common Subexpression Elimination 7](#_Toc402803875)

[5.2 Redundant Input Elimination 7](#_Toc402803876)

[5.3 HyperKernel Merging 7](#_Toc402803877)

[5.4 Multi-output HyperKernel Merging 7](#_Toc402803878)

[5.5 ProjectFrame – TensorReduceSum Consolidation 7](#_Toc402803879)

[6. Runtime 8](#_Toc402803880)

[6.1 Pipeliner 8](#_Toc402803881)

[6.2 Partitioner 8](#_Toc402803882)

[6.3 Shared Buffer Allocation 8](#_Toc402803883)

[6.4 Model Execution 8](#_Toc402803884)

[7. Platform 9](#_Toc402803885)

[7.1 Abstractions 9](#_Toc402803886)

[7.2 The OpenCL platform 9](#_Toc402803887)

# 1. Introduction

Purpose of the document

# 2. Overview

## 2.1 Compilation Phases (Parsing, Semantic Checking, Code Generation, Optimization)

## 2.2 Primary Data Structures (SyntaxTree, KernelCircuit)

# 3. Parser and Semantic Checking

## 3.1 Opcode classes

## 3.2 Operator classes

## 3.3 GPUOperators

# 4. Code Generation

## 4.1 Purpose

## 4.2 Generators

# 5. Optimizers

## 5.1 Common Subexpression Elimination

## 5.2 Redundant Input Elimination

## 5.3 HyperKernel Merging

## 5.4 Multi-output HyperKernel Merging

## 5.5 ProjectFrame – TensorReduceSum Consolidation

# 6. Runtime

## 6.1 Pipeliner

## 6.2 Partitioner

## 6.3 Shared Buffer Allocation

## 6.4 Model Execution

# 7. Platform

## 7.1 Abstractions

## 7.2 The OpenCL platform