DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Design and Implementation of a Shared Memory Backend for LAIK

Robert Hubinger

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Design and Implementation of a Shared Memory Backend for LAIK

Design und Implementierung eines Shared Memory Backends für LAIK

Author:	Robert Hubinger
Supervisor:	Josef Weidendorfer
Advisor:	Amir Raoofy
Submission Date:	15.09.2022

I confirm that this bachelor's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.09.2022 Robert Hubinger

Acknowledgments

I would like to thank my advisor, Amir Raoofy as well as my supervisor Josef Weidendorfer for their support during the whole process of creating this thesis. Thank you for the countless hours of discussions, guidance and advice.

Abstract

Contents

1 Introduction

2 Background

The understanding of the LAIK library and Inter Process Communication (IPC) is fundamental for understanding the design of the shared memory Backend. This chapter is divided into two parts. In the first, we will briefly introduce the basic features of LAIK, with particular emphasis on action sequences, backend interface, and data storage. In the second part we will cover the necessary basics of inter process communication, especially shared memory and semaphores.

2.1 LAIK

LAIK, which stands for „Leichtgewichtige AnwendungsIntegriete Datenhaltungskomponente", is a library for data management in the HPC environment. Created out of the need for higher flexibility in regard to scheduling and fault tolerance strategies, LAIK provides support for distributing data across parallel applications by controlling the data and its partitioning. The goal of LAIK is to provide fault tolerance mechanisms and load balancing for HPC applications in the most lightweight and performant way possible [1]. As shown in fig.2.1, LAIK sits between the actual application and the library used for communication.

Figure 2.1: LAIK and Communication Backend

2.1.1 Action Sequences

Action sequences are a list of actions which provide the information a node needs to execute a specific chain of actions. Actions are the atomic unit of execution in LAIK, they are predefined procedures which contain the necessary information for their own execution. LAIK creates action sequences based on the applications needs. After a sequence is created it can be optimised before it is executed. When an Action sequence is executed, the corresponding backend performs every Action of the sequence in the determined order. Action sequences can be executed multiple times. When an action sequence isn't needed anymore, it gets deleted.

2.1.2 Backend Interface

As per its specification, LAIK supports different communication libraries to execute the data migration [1]. Application programmers are able to

2.1.3 Data Storage

2.2 Inter Process Communication

2.2.1 Shared memory

2.2.2 Semaphores

3 Design

3.1 Architecture

3.1.1 Initialization

3.1.2 Data Transport

2 Copy Transport

1 Copy Transport

3.2 Secondary Backend Functionality

3.2.1 Integration With Other Backends

3.2.2 Action Substitution

3.3 Standalone Version

4 Implementation

5 Performance Analysis

5.1 Setup

5.2 Results

6 Experiences

7 Summary and Outlook

List of Figures

List of Tables

Bibliography

[1] C. T. Josef Weidendorfer Dai Yang. “LAIK: A Library for Fault Tolerant Distribution of Global Data for Parallel Applications”. en. In: *Konferenzband des PARS’17 Workshops*. Gesellschaft der Informatik. Hagen, 2017, p. 10.