INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Dynamic PGAS Data Structures

Stefan Effenberger

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Masterarbeit

Dynamic PGAS Data Structures

Stefan Effenberger

Aufgabensteller:	Prof. Dr. Dieter Kranzlmüller
Betreuer:	Tobias Fuchs
Abgabetermin:	ADD DATE

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den ADD DATE

. . . .

(Unterschrift des Kandidaten)

Abstract

ADD ABSTRACT

Inhaltsverzeichnis

1 Introduction

Many scientific projects are largely enabled by simulation. Because such simulations often require huge computational capabilites, single compute nodes with a shared-memory architecture cannot provide enough computation power and storage for numerous cases. For this reason, in High Performance Computing (HPC), work is distributed among multiple, interconnected nodes to facilitate the solving of large problems in a timely manner. Since processors cannot directly access the memory of other nodes, the traditional programming model for such systems requires programmers to explicitly distribute data between nodes via message passing. This imposes high demands on the programming skills of scientists who might not have a background in computer science.

Therefore, with the Partitioned Global Address Space (PGAS) programming model, a new approach is proposed: The memory space of individual nodes in a system is unified within a global address space so that each node can directly access the memory of all other nodes. Programmers are still required to keep data access between nodes to a minimum because data transferal over an interconnect is costly. To further reduce the demands on the programmer, distributed data structures that handle data distribution and load balancing are needed.

Furthermore, data-intensive tasks have been gaining a continually growing interest in the scientific community. Traditionally, applications in HPC follow a computation-centric approach by solving numerical algorithms in the fastest possible way. As "Big Data" is becoming increasingly important in scientific projects, a shift towards more data-oriented applications can be observed in recent HPC projects [ZZZ+14]. This trend requires distributed data structures that allow for the storage of large amounts of irregular data and cater to the needs of ever-changing dynamic data.

1.1 Problem statement

1.2 Scope and Objectives

2 Background

2.1 C++ Concepts

2.1.1 Language Concepts

2.1.2 Standard Template Library

2.2 HPC

2.3 Partitioned Global Address Space

2.4 DASH Library

3 Related Work

3.1 Shared Memory

3.1.1 STINGER

3.1.2 Ligra

3.2 Distributed Memory

3.2.1 Parallel Boost Graph Library

3.2.2 STAPL Parallel Graph Library

4 Graph Container Concepts

4.1 Memory Space

4.2 Index Space

4.3 Iteration Space

4.4 Semantics

5 Reference Implementation

6 Case studies

6.1 Static structure

6.1.1 Graph traversal

6.1.2 Shortest path evaluation

6.2 Dynamic Structure

6.2.1 Graph partitioning

6.2.2 De Bruijn Graph construction

7 Evaluation

7.1 Micro-benchmarks

8 Conclusion

8.1 Summary

8.2 Assessment

8.3 Outlook

Abbildungsverzeichnis

Literaturverzeichnis

[ZZZ+14] Zhao, D. ; Zhang, Z. ; Zhou, X. ; Li, T. ; Wang, K. ; Kimpe, D. ; Carns, P. ; Ross, R. ; Raicu, I.: FusionFS: Toward supporting data-intensive scientific applications on extreme-scale high-performance computing systems. In: *2014 IEEE International Conference on Big Data (Big Data)*, 2014, S. 61–70