Due to the ever-increasing demand for scalable architectures and programs in the field of high performance computing new programming models have been established in the last years as alternatives to the message-based programming models. Today, distributed applications can create a shared global address space over all nodes of an HPC system by using asynchronous, one-sided memory accesses to remote memory. Based on this global address space the PGAS programming model has emerged, which enables new ways to develop powerful applications, but also introduces new challenges to ensure program correctness and efficiency.

This thesis contributes to the systematic understanding of parallel distributed applications with a shared global address space. It focuses on the analysis of the interaction of asynchronous and synchronous memory accesses in this address space. The concept of memory access diagrams presented in this thesis opens up a new analysis perspective to the programmer. The underlying task graph model has been enhanced to accurately map the causal relationships between asynchronous memory accesses and other program events. By adapting the model, an algorithm can be specified for the generally NP-complete calculation of synchronization relationships, which performs this calculation in quasi-linear time.

The thesis demonstrates and evaluates the application of the new methods with various examples from research practice. The demonstration illustrates the usefulness of memory access diagrams to visualize the logical causes of programming errors and performance flaws. A further result of the thesis is the prototype of a novel analysis tool, which is already in use in several research institutions.