Parallelizing Irregular Applications through the YAPPA Compilation Framework
Silvia Lovergine, Antonino Tumeo, Oreste Villa, Fabrizio Ferrandi

Motivation
Modern High Performance Computing (HPC) systemsclustered composed of hundreds of nodes integrating multicore processors with advanced cache hierarchies. They reach several petaflops of peak performance, but they are optimized for floating point intensive applications, and regular, localizable data structures. Network interconnection optimized for bulk, synchronous transfers. Many scientific applications are irregular:
- Dynamic, linked data structures (e.g., graphs, unbalanced trees, unstructured grids), irregular control flow, unpredictable and fine-grained communication
- Inherently parallel, but difficult to partition in a balanced way
- Almost no locality
- High synchronization intensity

Complex design challenges and significant programming effort

Our approach
YAPPA (Yet Another Parallel Programming Approach): a compilation framework for the automatic parallelization of irregular applications on modern HPC systems, based on LLVM.
YAPPA builds on the top of GMT (Global Memory and Threading library): a runtime library for irregular applications on distributed memory HPC systems.
YAPPA analyzes a C/C++ application to automatically produce its parallelization.

GTM: runtime library for irregular applications on distributed memory HPC systems
- Built around three main concepts: PGAS data model, latency tolerance through fine-grained software multithreading, and aggregation.
- Partitioned Global Address Space (PGAS) removes need to partition datasets.
- Fine-grained software multithreading tolerates network latencies.
- Data aggregation reduces overheads of fine-grained network accesses.
- Fork/join control model with simple parallel for constructs, eases application development

Example: Breadth First Search (BFS) Algorithm

Parallelizing Irregular Applications through the YAPPA Compilation Framework

Motivation
Modern High Performance Computing (HPC) systems clustered composed of hundreds of nodes integrating multicore processors with advanced cache hierarchies. They reach several petaflops of peak performance, but they are optimized for floating point intensive applications, and regular, localizable data structures. Network interconnection optimized for bulk, synchronous transfers. Many scientific applications are irregular:

- Dynamic, linked data structures (e.g., graphs, unbalanced trees, unstructured grids), irregular control flow, unpredictable and fine-grained communication
- Inherently parallel, but difficult to partition in a balanced way
- Almost no locality
- High synchronization intensity

Complex design challenges and significant programming effort

Our approach
YAPPA (Yet Another Parallel Programming Approach): a compilation framework for the automatic parallelization of irregular applications on modern HPC systems, based on LLVM.
YAPPA builds on the top of GMT (Global Memory and Threading library): a runtime library for irregular applications on distributed memory HPC systems.
YAPPA analyzes a C/C++ application to automatically produce its parallelization. YAPPA requires the programmer to manage synchronization by means of atomic operations

GTM: runtime library for irregular applications on distributed memory HPC systems
- Built around three main concepts: PGAS data model, latency tolerance through fine-grained software multithreading, and aggregation.
- Partitioned Global Address Space (PGAS) removes need to partition datasets.
- Fine-grained software multithreading tolerates network latencies.
- Data aggregation reduces overheads of fine-grained network accesses.
- Fork/join control model with simple parallel for constructs, eases application development

Example: Breadth First Search (BFS) Algorithm

Parallelizing Irregular Applications through the YAPPA Compilation Framework
Silvia Lovergine, Antonino Tumeo, Oreste Villa, Fabrizio Ferrandi

Motivation
Modern High Performance Computing (HPC) systems clustered composed of hundreds of nodes integrating multicore processors with advanced cache hierarchies. They reach several petaflops of peak performance, but they are optimized for floating point intensive applications, and regular, localizable data structures. Network interconnection optimized for bulk, synchronous transfers. Many scientific applications are irregular:

- Dynamic, linked data structures (e.g., graphs, unbalanced trees, unstructured grids), irregular control flow, unpredictable and fine-grained communication
- Inherently parallel, but difficult to partition in a balanced way
- Almost no locality
- High synchronization intensity

Complex design challenges and significant programming effort

Our approach
YAPPA (Yet Another Parallel Programming Approach): a compilation framework for the automatic parallelization of irregular applications on modern HPC systems, based on LLVM.
YAPPA builds on the top of GMT (Global Memory and Threading library): a runtime library for irregular applications on distributed memory HPC systems.
YAPPA analyzes a C/C++ application to automatically produce its parallelization. YAPPA requires the programmer to manage synchronization by means of atomic operations

GTM: runtime library for irregular applications on distributed memory HPC systems
- Built around three main concepts: PGAS data model, latency tolerance through fine-grained software multithreading, and aggregation.
- Partitioned Global Address Space (PGAS) removes need to partition datasets.
- Fine-grained software multithreading tolerates network latencies.
- Data aggregation reduces overheads of fine-grained network accesses.
- Fork/join control model with simple parallel for constructs, eases application development

Example: Breadth First Search (BFS) Algorithm

Parallelizing Irregular Applications through the YAPPA Compilation Framework
Silvia Lovergine, Antonino Tumeo, Oreste Villa, Fabrizio Ferrandi

Motivation
Modern High Performance Computing (HPC) systems clustered composed of hundreds of nodes integrating multicore processors with advanced cache hierarchies. They reach several petaflops of peak performance, but they are optimized for floating point intensive applications, and regular, localizable data structures. Network interconnection optimized for bulk, synchronous transfers. Many scientific applications are irregular:

- Dynamic, linked data structures (e.g., graphs, unbalanced trees, unstructured grids), irregular control flow, unpredictable and fine-grained communication
- Inherently parallel, but difficult to partition in a balanced way
- Almost no locality
- High synchronization intensity

Complex design challenges and significant programming effort

Our approach
YAPPA (Yet Another Parallel Programming Approach): a compilation framework for the automatic parallelization of irregular applications on modern HPC systems, based on LLVM.
YAPPA builds on the top of GMT (Global Memory and Threading library): a runtime library for irregular applications on distributed memory HPC systems.
YAPPA analyzes a C/C++ application to automatically produce its parallelization. YAPPA requires the programmer to manage synchronization by means of atomic operations

GTM: runtime library for irregular applications on distributed memory HPC systems
- Built around three main concepts: PGAS data model, latency tolerance through fine-grained software multithreading, and aggregation.
- Partitioned Global Address Space (PGAS) removes need to partition datasets.
- Fine-grained software multithreading tolerates network latencies.
- Data aggregation reduces overheads of fine-grained network accesses.
- Fork/join control model with simple parallel for constructs, eases application development

Example: Breadth First Search (BFS) Algorithm