Application of HPX to tiled GEMM and QR: A benchmark

September 24, 2019 | Thomas Miethlinger | Jülich Supercomputing Centre

Part I: Introduction

About me

(Thomas Miethlinger)

- Study: Master Physics
- Johannes Kepler University of Linz
- Institute for Theoretical Physics
  Department Many Particle Systems
- Research:
  - Quantum fluids
  - Complex fluids
  - Non-equilibrium statistical mechanics

About the GSP

- Supervisor: Dr. Edoardo Di Napoli
- Co-Supervisor: Dr. Xinzhe Wu
- SimLab Quantum Materials
- Research:
  - Development and maintenance of numerical libraries
  - Design and implementation of high-performance algorithms
  - Development of new mathematical and computational models within a methodological framework

  in the scope of computational materials science and quantum materials.

Part II: Introduction to HPX

Current sitution in high performance computing (HPC)

Currently, speed-up in computing does not stem from higher CPU frequency, but increased parallelism. However, we already face the following challenges in HPC:

- Ease of programming
- Inability to handle dynamically changing workloads
- Scalability
- Efficient utilization of system resources

$\Longrightarrow$ a need for a new execution model: ParalleX, which is implemented by HPX

ParalleX

ParalleX is a new parallel execution model that offers an alternative to the conventional computation models(e.g. message passing):

- Split-phase transaction model
- Message-driven
- Distributed shared memory
- Multi-threaded
- Futures synchronization
- Local Control Objects (LCOs)
- ...

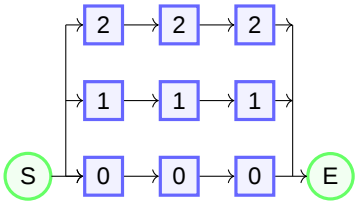
ParalleX focusses on latency hiding instead of latency avoidance.

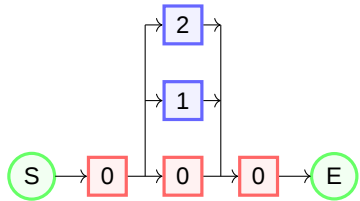
About HPX

- High Performance ParalleX (HPX) is the first runtime system implementation of the ParalleX execution model.
- Development: STE||AR group
  Louisiana State University
  LSU Center for Computation and Technology
- Released as open source under the Boost Software License
- Current version: HPX V1.3.0, released on 23.05.2019
- Aims to be a **C++ standards conforming implementation** of the Parallelism and Concurrency proposals for C++ 17/20/23/...
- This means: HPX is a C++ library that supports **dynamic adaptive resource management** and **lightweight task programming and scheduling** within the context of a **global address space**.

Comparison of HPX and OpenMP

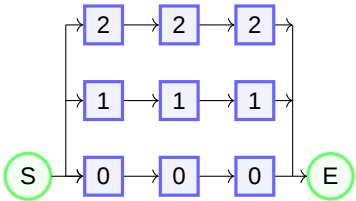
HPX	OpenMP
C++ library	Compiler extension to C and Fortran
Core language: `hpx::C++`	`#pragma omp` directives
Task-based parallelism	Parallel regions (fork-join model)
AGAS (active global address space)	shared memory

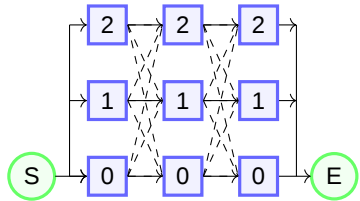




Comparison of HPX and MPI

HPX	MPI
C++ library	Interface specification for C and Fortran
Core language: `hpx::c++`	Core language: MPI_C, MPI_F08
Task-based parallelism	Single program, multiple data (SPMD)
AGAS (active global address space)	Explicit message passing





On learning HPX

An opinion of a non-CS/HPC student

Learning curve on of HPX is quite steep - in the first days quite some dedication, effort and endurance is needed[1].

- Probably the easiest way in the beginning: watch this nice playlist in 1.25x speed on the youtube channel of cscsch (Swiss National Supercomputing Centre)
- Be aware that the API reference is not complete
- Be aware that there exist 4 different “Hello, World!” examples[2]:
  - hello_world_component/*: 3 files; 28, 30 & 55 lines
  - quickstart/hello_world_1.cpp; 22 lines
  - quickstart/hello_world_2.cpp; 24 lines
  - quickstart/hello_world_distributed.cpp; 156 lines

[1] Why is the HPX code repo so big and complicated?

[2] Paths are with respect to `https://github.com/STEllAR-GROUP/hpx/examples/`

Part III: Overview of numerical linear algebra and its applications

Part IV: GEMM

Part V: QR

Application of HPX to tiled GEMM and QR: A benchmark

September 24, 2019 | Thomas Miethlinger | Jülich Supercomputing Centre