

GPI – APPLICATIONS EXCEED YOUR DREAMS

Fraunhofer-Institut für Techno- und Wirtschaftsmathematik ITWM

Fraunhofer-Platz 1
67663 Kaiserslautern
Germany

Contact

Dr. Franz-Josef Pfreundt
Competence Center for High
Performance Computing
Phone +49 631 31600-44 59
pfreundt@itwm.fraunhofer.de

www.itwm.fraunhofer.de
www.itwm.fraunhofer.de/hpc

Business contact

scapos AG
Schloss Birlinghoven
53754 Sankt Augustin
gpi@scapos.com

The most promising programming model of the future is the Partitioned Global Address Space (PGAS). The programmability combined with efficiency is the key to fully utilize the machines of the Post-PetaFlop/s era.

GPI provides a PGAS-API for your standard language. It delivers the full hardware speed directly to the application. The truly asynchronous, one sided and zero copy semantics of GPI allows to overlap communication and computation perfectly, a key requirement for scalable software.

GPI is easy to use and the adoption of existing parallel codes does not require a complete rewrite. In GPI the communication pattern is under full control. This is in contrast to PGAS languages like CoArray Fortran or UPC, where one often ends up with a major rewrite and still depends on compiler magic.

In order to achieve good scalability, one has to rethink and reformulate the communication strategy. The elimination of synchronous communication is beneficial not only for GPI programs, but often improves message passing implementations as well.

GPI coverage of a large class of scientific algorithms

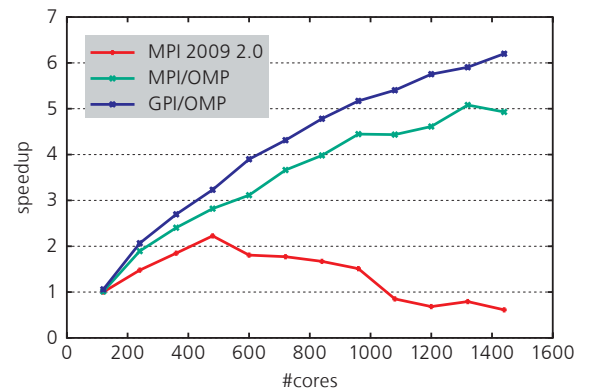
- Computational fluid dynamics (1)
- Stencil codes (2)
- Graph algorithms (3)

Best in class performance in all domains

- Full overlap of communication and computation
- No CPU cycles spent for communication
- No cache disturbance
- Truly asynchronous
- Scalability

Computational Fluid Dynamics¹

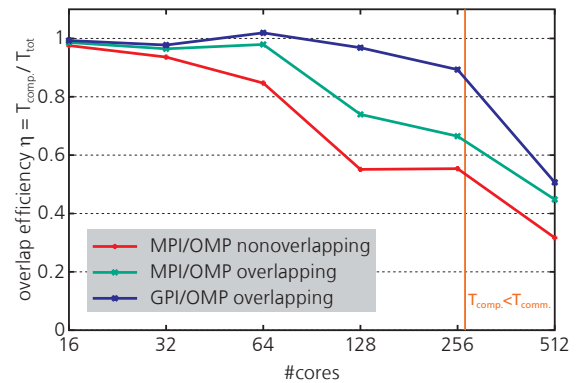
GPI enables strong scalability of the unstructured CFD solver TAU far beyond the capabilities of MPI. TAU is developed by the German Aerospace Research Center (DLR). The kernel of TAU is a finite volume Euler, Navier-Stokes solver on unstructured grids in a domain decomposition. GPI makes a truly asynchronous implementation possible that perfectly overlaps computation and communication. Load imbalances, as seen in the MPI implementation, are completely avoided. The GPI implementation that was developed together with T-Systems² not only scales one order of magnitude higher than the synchronous MPI implementation, it also outperforms the hybrid and overlapping OpenMP/MPI variant clearly.



TAU (F6, 4W Multigrid, 2 million points): speedup
(DLR, Xeon X5670, 16 GiB, QDR Infiniband)

Stencil codes: Berlin Quantum Chromo Dynamics³

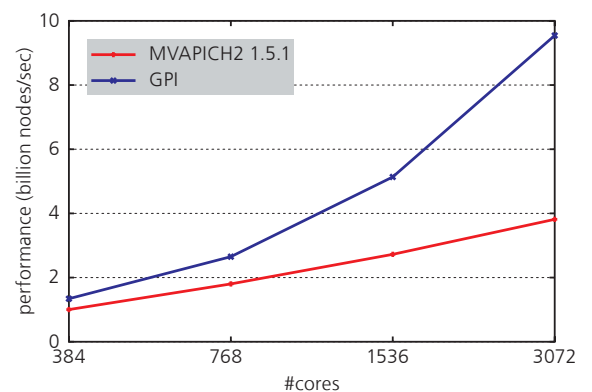
GPI is able to fully hide the communication behind the computation in the theoretically feasible regime as seen for the Berlin Quantum Chromo Dynamics code (BQCD). BQCD is a Hybrid Quantum Monte-Carlo algorithm using domain decomposition of the underlying four dimensional regular space-time grid. The computational kernel of BQCD is the evaluation of a four dimensional nearest neighbor stencil operator. In contrast to the MPI implementation, the GPI implementation shows perfect overlap of communication and computation in the theoretically feasible regime. No cache coherency effects and no CPU load is induced by the communication. There is essentially no difference between the theoretical expectation and the practical measurement.



BQCD (24x24x24x48): covariant derivative D, overlap efficiency (HLRS, Xeon E5440, 16 GiB, DDR Infiniband)

Graph algorithms: Unbalanced Tree Search⁴

GPI outperforms MPI in the Unbalanced Tree Search (UTS) benchmark by a maximum factor of 2.5 in terms of raw performance. The UTS benchmark is designed to evaluate the performance and the ease of programming for parallel applications requiring dynamic load balancing. These applications are characterized by unpredictable communication, unpredictable synchronization and dynamic work granularity. The one sided communication paradigm of GPI perfectly fits these requirements and allows for the fastest UTS implementation ever.



UTS (geometric tree of 270 billion nodes): performance
(Xeon X5670, 16 GiB, QDR Infiniband)

¹ Schwamborn, D.; Gerhold, T.; Heinrich, R.: The DLR TAU-code: Recent applications in research and industry, European conference on computational fluid dynamics, ECCOMAS CFD, pages 1-25

² Jagersküpfer, J. (DLR); Simmendinger, C. (T-Systems Sfr): A Novel Shared-Memory Thread-Pool Implementation for Hybrid Parallel CFD Solvers, EuroPar 2011

³ Nakamura, Y.; Stuben, H.: BQCD – Berlin Quantum Chromo Dynamics Program, PoS, LATTICE 2010: 40, 2010

⁴ Olivier, S.; Huan, J.; Liu, J.; Dinan, J.; Sadayappan, P.; Tseng, C.-W.: UTS: an unbalanced tree search benchmark, Proc 19th intl workshop on languages and compilers for parallel computing (LCPC), New Orleans LA, November 2-4, 2006