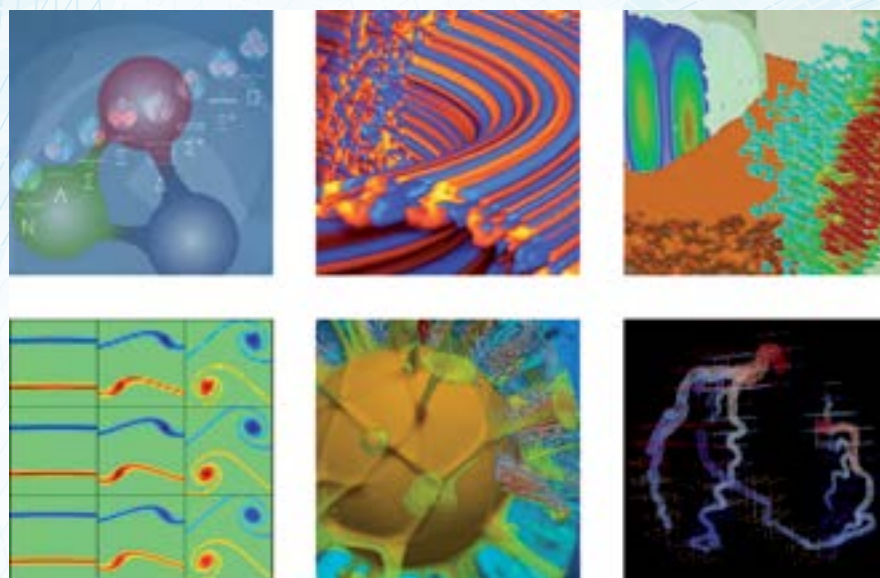


High-Q Club - The highest scaling Codes on JUQUEEN

The trend in supercomputing towards much higher core counts seems inevitable. As a consequence, the users of supercomputing resources once again need to adapt their strategies in programming for these architectures. Whereas in the past we have for example seen vector machines come and go, the MPI standard has become established as the most widely used programming model. With multicore architectures, however, shared memory programming is making something of a comeback.



The latest supercomputer in the Jülich Supercomputing Centre, JUQUEEN, has 16 CPU cores per node and supports up to 64 hardware threads. To help our users migrate their codes and get the most performance for their algorithms, we organised a first porting and tuning workshop earlier this year [1]. Following this workshop and to promote the idea of exascale capability computing, we have established the High-Q Club, a

showcase for codes able to utilise the entire 28-rack BlueGene/Q machine at JSC. The club members comprise a collection of the highest scaling codes on JUQUEEN, through which we intend to encourage other developers to invest in tuning and scaling their codes. We want our users to show that they are capable of using all 458,752 cores, and for example more than 1 million concurrent threads on JUQUEEN.

The diverse membership of the High-Q Club establishes that it is possible to scale real applications to the complete JUQUEEN using a variety of programming languages and parallelisation models, demonstrating individual approaches to reach that goal. High-Q status thus marks an important milestone in application development towards future HPC systems that envisage even higher core counts.

To qualify for membership, developers should submit evidence of the scalability of their codes across all available cores. While we currently do not set a strict minimum efficiency, we do expect the codes to profit from additional cores with an increase in speed. The benchmark used should also be as close as possible to a production scenario: trivial kernels or libraries will not be accepted.

Current members of the High-Q Club are:

dynQCD

dynQCD is a code for simulations in the field of lattice quantum chromodynamics that can be used for different fermion actions. The code is developed at the

University of Wuppertal and the Simulation Laboratory for Nuclear and Particle Physics at JSC and is written in C using pthreads. The BG/Q version in particular makes use of a 4D torus and uses low level SPI calls to the network hardware instead of the MPI library.

Gysela

Gysela is a GYrokinetic SEmi-LAgrangian code for plasma turbulence simulations developed at CEA Cadarache. It is for example used in simulations of the electrostatic branch of the ion temperature gradient turbulence in tokamak plasmas. Gysela is written in Fortran90 and C and uses MPI, OpenMP and pthreads.

PEPC

The Pretty Efficient Parallel Coulomb solver is used for n-body simulations developed within the Simulation Laboratory for Plasma Physics at JSC. PEPC is not restricted to a specific force law or physical problem and for example used for beam-plasma interaction, vortex dynamics, gravitational interaction or molecular dynamics simulations. The code is written in Fortran2003 and C, making use of MPI, OpenMP and pthreads as programming models.

PMG+PFASST

PMG+PFASST combines a parallel multi-grid solver with a time parallel approximation scheme to solve ODEs with linear stiff terms. The two parts have been developed at the Lawrence Berkeley National Lab (PFASST) and the University of Wuppertal (PMG) and have been coupled to one application by developers from the Cross-sectional team Mathematical Methods and Algorithms at JSC and Università della Svizzera italiana. PMG+PFASST is written in Fortran2003 and C with MPI and pthreads.

Terra-Neo

Terra-Neo is used for modeling earth mantle dynamics and is developed specifically for the upcoming heterogeneous exascale computers by using an advanced architecture-aware co-design approach. The development team is built from members of Ludwig-Maximilians-Universität München, Universität Erlangen-Nürnberg, Regionales Rechenzentrum Erlangen and Technische Universität München. They use C++ and Fortran with MPI and OpenMP for the Terra-Neo framework.

walBerla

walBerla is the widely applicable Lattice Boltzmann solver from Erlangen developed by the Universität Erlangen-Nürnberg. Originally, the walBerla framework has been centered around the Lattice-Boltzmann method for the simulation of fluid scenarios but in the meantime evolved to a code that is also suitable for a wide range of applications based on structured grids. It is developed in C++ and builds on MPI and OpenMP as well as CUDA and OpenCL for other architectures.

The updated list of members and more information is available at:

<http://www.fz-juelich.de/ias/jsc/high-q-club>

Reference

- [1] Brömmel, D.
First JUQUEEN Porting and Tuning Workshop, Innovatives Supercomputing in Deutschland, Vol. 1, No. 1, 2013

- Dirk Brömmel
- Paul Gibbon

Jülich
Supercomputing
Centre (JSC)