Extreme-scaling Applications 24/7 on *JUQUEEN* Blue Gene/Q

Dirk BRÖMMEL [a,1], Wolfgang FRINGS [a], Brian J. N. WYLIE [a]

[a] *Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Germany*

Abstract.
Jülich Supercomputing Centre has offered Extreme Scaling Workshops since 2009, with the latest edition in February 2015 giving seven international code teams an opportunity to (im)prove the scaling of their applications to all 458 752 cores of the *JUQUEEN* IBM Blue Gene/Q. Each of them successfully adapted their application codes and datasets to the restricted compute-node memory and exploit the massive parallelism with up to 1.8 million processes or threads. They thereby qualified to become members of the *High-Q Club* which now has over 24 codes demonstrating extreme scalability. Achievements in both strong and weak scaling are compared, and complemented with a review of program languages and parallelisation paradigms, exploitation of hardware threads, and file I/O requirements.

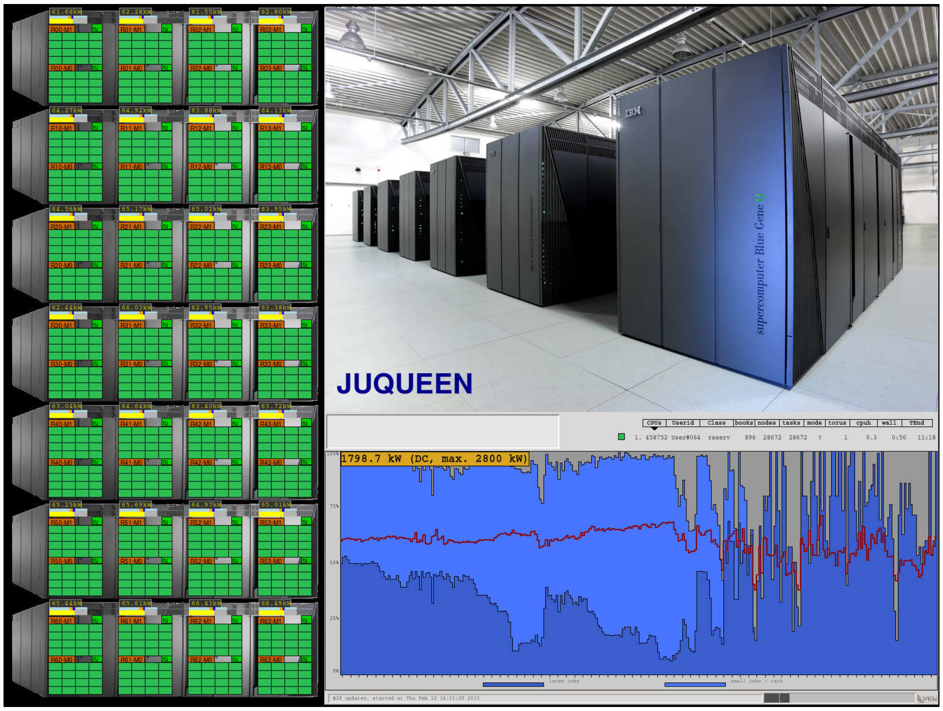
Keywords. HPC applications, *JUQUEEN*, Blue Gene/Q, extreme-scaling workshop, *High-Q Club*.

Introduction

From 5 to 6 February 2015, Jülich Supercomputing Centre (JSC) organised the latest edition of its series of Blue Gene Extreme Scaling Workshops [1]. These workshops started with the 2006 Blue Gene/L Scaling Workshop using JUBL (16 384 cores) and moved to JUGENE for the 2008 Blue Gene/P Porting, Tuning & Scaling Workshop [2], then followed by dedicated Extreme Scaling Workshops in 2009 [3], 2010 [4] and 2011 [5]. These latter three workshops attracted 28 teams selected from around the world to investigate scalability on the most massively-parallel supercomputer at the time with its 294 912 cores. 26 of their codes were successfully executed at that scale, three became ACM Gordon Bell prize finalists, and one participant was awarded an ACM/IEEE-CS George Michael Memorial HPC fellowship. The Leibniz Supercomputing Centre (LRZ) adopted a similar format for workshops in 2013 [6] and 2014 [7] to scale applications on the SuperMUC IBM iDataPlex system, and from 22 participating code teams three succeeded in running on all 14 “thin node” islands (147 456 cores in total).

The focus for the current workshop was on application codes likely to be able to scale during the workshop to run on the full JUQUEEN system [8]. This 28-rack IBM Blue Gene/Q (Figure 1) with 28 672 compute nodes, consisting of 1.6 GHz PowerPC A2 processors each with 16 cores (64 hardware threads) and 16 GB of node memory, has a total of 458 752 cores capable of running 1 835 008 processes or threads. A broad va-

[1] Corresponding author: d.broemmel@fz-juelich.de



	CPUs	Userid	Class	books	nodes	tasks	mode	torus	cpuh	wall	TEnd
🟩 1.	458752	User#064	reserv	896	28672	28672	?	1	0.3	0:50	11:18

Figure 1. *JUQUEEN* Blue Gene/Q as presented by the *LLview* system monitor when running a single job on the full configuration of 4 × 7 racks during the workshop, where the lower-right chart shows the mix of jobs changing to dedicated large jobs (shown darker).

riety of application codes which have demonstrated that they can productively exploit the entire JUQUEEN resources have already been recognised as members of the High-Q Club [9]. The High-Q Club is a collection of the highest scaling codes on JUQUEEN and as such requires the codes to run on all 28 racks. Codes also have to demonstrate that they profit from each additional rack of JUQUEEN in reduced time to solution when strong scaling a fixed problem size or a tolerable increase in runtime when weak scaling progressively larger problems. Furthermore the application configurations should be beyond toy examples and we encourage use of all available hardware threads which is often best achieved via mixed-mode programming combining message-passing with multithreading. Each code is then individually evaluated based on its weak or strong scaling results with no strict limit on efficiency. The workshop thus provided an opportunity for additional candidates to prove their scalability and qualify for membership, or – as was the case for one of the codes – improve on the scaling and efficiency that they had already achieved.

1. Workshop Overview

Seven application teams were invited to work on the scalability of their codes, with dedicated access to the entire JUQUEEN system for a period of 30 hours. Most of the teams' codes had thematic overlap with JSC Simulation Laboratories or were part of an ongoing collaboration with one of the SimLabs. Following earlier tradition, the 2015 Extreme Scaling Workshop was directly preceded by a Porting and Tuning Workshop, offered by JSC as part of the PRACE Advanced Training Centre (PATC) curriculum. Hence most of the application teams were among the 25 new and more experienced users of JUQUEEN who were also present for the prior three days and used the opportunity for initial preparations, performance analyses and tuning tips. Pre-workshop preparations had already made sure that suitable datasets and execution configuration were ready. During both workshops the code teams were supported by JSC Cross-sectional teams and Climate Science, Fluids & Solids Engineering and Neuroscience SimLabs, along with IBM and JUQUEEN technical support.

The seven participating code teams (attendees marked in **bold**) were:

- CoreNeuron *electrical activity of neuronal networks with morphologically-detailed neurons*
  Fabien Delalondre, Pramod Kumbhar, **Aleksandr Ovcharenko** (Blue Brain Project, EPFL), and Michael Hines (Yale University)
- FE^{2}TI *scale-bridging approach incorporating micro-mechanics in macroscopic simulations of multi-phase steels*
  Axel Klawonn and **Martin Lanser** (University of Cologne), **Oliver Rheinbach** (TU Freiberg), Jörg Schröder (University Duisburg-Essen), Daniel Balzani (TU Dresden), and Gerhard Wellein (University Erlangen-Nürnberg)
- FEMPAR *massively-parallel finite-element simulation of multi-physics problems governed by PDEs* — High-Q Club member since Dec. 2014
  Santiago Badia, **Alberto F. Martín**, and Javier Principe (Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Universitat Politècnica de Catalunya)
- ICON *icosahedral non-hydrostatic atmospheric model*
  Catrin Meyer (Forschungszentrum Jülich GmbH, JSC) and **Thomas Jahns** (Deutsches Klimarechenzentrum GmbH)
- MPAS-A *multi-scale non-hydrostatic atmospheric model for global, convection-resolving climate simulations*
  Dominikus Heinzeller (Karlsruhe Inst. of Technology, Inst. of Meteorology and Climate Research) and Michael Duda (National Center for Atmospheric Research, Earth System Laboratory)
- psOpen *direct numerical simulation of fine-scale turbulence*
  Jens Henrik Goebbert (Jülich Aachen Research Alliance) and **Michael Gauding** (TU Freiberg)
- SHOCK *structured high-order finite-difference computational kernel for direct numerical simulation of compressible flow*
  Manuel Gageik and Igor Klioutchnikov (Shock Wave Laboratory, RWTH Aachen University)

A total of 370 'large' jobs were executed (58 on 28 racks, 19 on 24 racks, 6 on 20 racks and 105 on 16 racks) using 12 of the 15 million core-hours reserved for the extreme scaling workshop. Most of the familiar LoadLeveler job scheduling quirks were avoided by deft sysadmin intervention, and a single nodeboard failure requiring a reset resulted in only a short outage when smaller jobs could be executed on the remaining racks.

Table 1. Characteristics of workshop application codes: main programming languages (excluding external libraries), parallelisation including maximal process/thread concurrency (per compute node and overall), and file I/O implementation. (Supported capabilities unused for scaling runs on *JUQUEEN* in parenthesis.)

	Programming			Parallelisation			
Code	Languages			MPI	OMP	Concurrency	File I/O
CoreNeuron	C	C++		1	64	64: 1 835 008	MPI-IO
FE^{2}TI	C	C++		16	4	64: 1 835 008	
FEMPAR			F08	64		64: 1 756 001	
ICON	C		Ftn	1	64	64: 1 835 008	(netCDF)
MPAS-A	C		Ftn	16		16: 458 752	PIO,pNetCDF
psOpen			F90	32	2	64: 1 835 008	pHDF5
SHOCK	C			64		64: 1 835 008	(cgns/HDF5)

2. Parallel Program & Execution Configuration Characteristics

Characteristics of the workshop application codes are summarised in Table 1 and discussed in this section, with scaling performance compared in the following section.

Since Blue Gene/Q offers lower-level function calls for some hardware-specific features that are sometimes not available for all programming languages, a starting point is looking at the languages used. Of the workshop codes, two combine Fortran with C, two use C and C++, and the remaining three exclusively use only either Fortran or C, indicating that all three major programming languages are equally popular (without considering lines of code) as seen with the other High-Q Club codes.

The four hardware threads per core of the Blue Gene/Q chip in conjunction with the limited amount of memory suggest to make use of multi-threaded programming. It is therefore interesting to see whether this is indeed the preferred programming model and whether the available memory is an issue. As a basis, all seven workshop codes used MPI, which is almost ubiquitous for portable distributed-memory parallelisation – for example only one High-Q Club application employs lower-level machine-specific SPI for maximum performance. Three of the workshop codes exclusively used MPI for their scaling runs, both between and within compute nodes. A memory fragmentation issue in a third-party library currently inhibits the use of OpenMP by FEMPAR. On the other hand MPAS-A just started to include OpenMP multi-threading, whereas an earlier investigation with the SHOCK code found this not to be beneficial. The remaining four workshop codes employ OpenMP multi-threading to exploit compute node shared memory in conjunction with MPI. In addition, CoreNeuron has an ongoing effort investigating use of OpenMP-3 tasking and new MPI-3 capabilities (e.g. non-blocking collectives), so these are generally expected to become increasingly important. CoreNeuron is also reorganising data structures to be able to exploit vectorisation. To address MPI overhead, psOpen exploited their own three-dimensional FFT library using non-blocking communication to more than halve communication time by overlapping multiple FFT instances.

The decision for a specific programming model had other implications. ICON needed MPI to be initialised with `MPI_THREAD_MULTIPLE` multi-threading support for an external library, which was determined to result in prohibitive time during model initialisation for `MPI_Allreduce` calls with user-defined reductions of `MPI_IN_PLACE` arguments on communicators derived from `MPI_COMM_WORLD`: the code was therefore

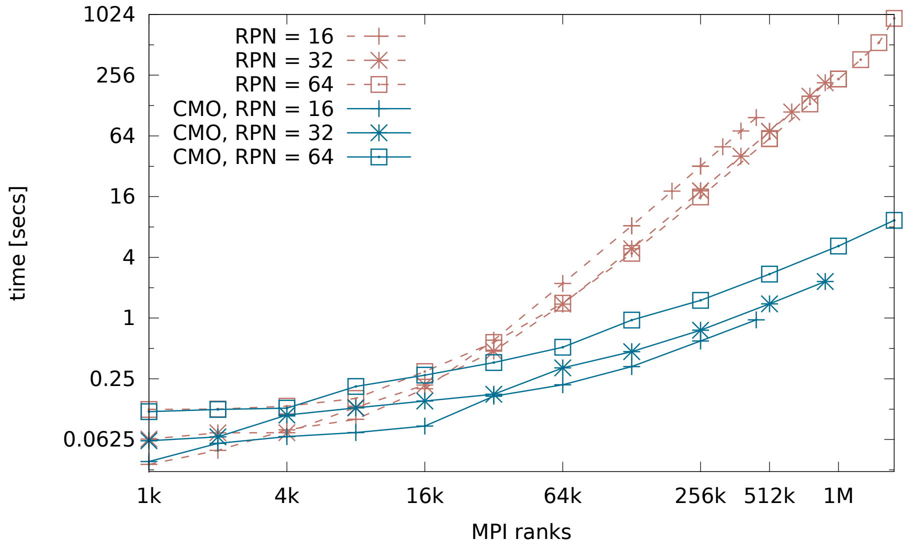


Figure 2. Scaling of wallclock execution time for `MPI_Comm_split` sub-communicator creation on *JUQUEEN* Blue Gene/Q: default and less memory-optimised alternative (CMO).

changed to circumvent this. MPAS-A bootstrapping to set up the grid and neighbourhood halo/ghost cells during its model initialisation was also found to take almost 30 minutes.

Using only MPI means accommodating to the restricted per-process memory. However, two codes in addition had to trade somewhat higher memory requirements for much faster MPI communicator management, a consequence of the vast number of MPI processes possible on JUQUEEN. For FEMPAR and FE^{2}TI the `PAMID_COLLECTIVES_MEMORY_OPTIMIZED` environment variable was critical to reduce the time of `MPI_Comm_split` from 15 minutes down to under 10 seconds. Figure 2 shows the benefit of this as well as different optimisation strategies within the MPI library for different numbers of MPI ranks.

Codes employing shared memory parallelisation also struggled with memory. For CoreNeuron available memory is the limiting factor for larger simulations, with the current limit being 155 million neurons using 15.9 GB of RAM. ICON was able to benefit from a recent reworking to substantially reduce the memory that it needed for large-scale executions, whereas SHOCK started to investigate using single-precision datatypes to reduce its memory requirements. MPAS-A required 1 GB of memory on each process for its regular 3 km mesh simulation (over 65 million grid cells with 41 vertical levels), and could therefore only use a single hardware thread per core, limiting its effective performance.

The other six workshop codes were able to use all four hardware threads of each processor core. FEMPAR and SHOCK used all 64 hardware threads for MPI processes, and in this way FEMPAR was able to increase its efficiency and scalability to 1.75 million processes using $27\frac{1}{2}$ racks of JUQUEEN when employing an additional (fourth-)level of domain decomposition. The other five codes exploited mixed-mode parallelisation with each MPI process having 64 OpenMP threads for CoreNeuron and ICON or only a few OpenMP threads in the case of FE^{2}TI and psOpen.

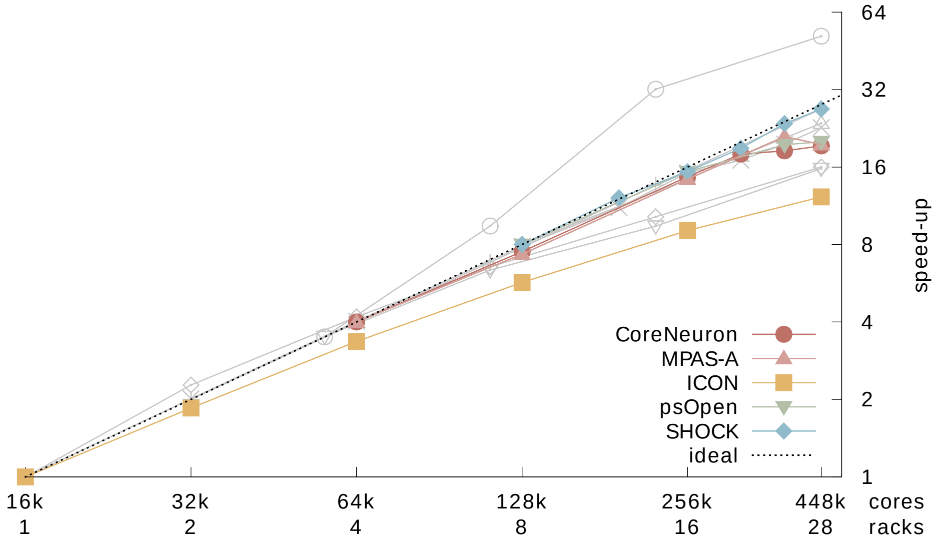


Figure 3. Strong scaling results of the workshop codes with results from existing High-Q Club members included in light grey.

3. Strong and Weak Scaling and Performance

An overview of the results of a scaling workshop entails some form of comparison of achievements in *strong* (fixed total problem size) and *weak* (fixed problem size per process or thread) scaling, put in context of the scalability results from other codes in the High-Q Club.

Figures 3 and 4 show strong and weak scaling results of the workshop codes, including in grey results from a selection of High-Q Club codes. This indicates the spread in execution results and diverse scaling characteristics of the codes. The figures show that the workshop codes not only managed to run on the full JUQUEEN system, but they also achieved very nice scalability and five new codes therefore qualified for High-Q Club status as an outcome of the workshop. Note that in many cases the graphs do not have a common baseline of one rack since datasets sometimes did not fit available memory or no data was provided for 1024 compute nodes: for strong scaling an execution with a minimum of seven racks (one quarter of JUQUEEN) is accepted for a baseline measurement, with perfect-scaling assumed from a single rack to the baseline.

In Figure 3 almost ideal strong-scaling speed-up of 27× on 28 racks is achieved by SHOCK, whereas ICON only achieved a modest 12× speed-up and with the other workshop codes in between.

Even with its heroic dataset of over 65 million grid-points, MPAS-A suffered a performance breakdown in strong scaling going from 24 to 28 racks due to growing communication costs overwhelming diminishing per-rank computation. A similar breakdown was also found with SHOCK when strong scaling with 64 MPI ranks per compute node (but not evident with only 32 rpn). In both cases, larger datasets are expected to avoid this breakdown.

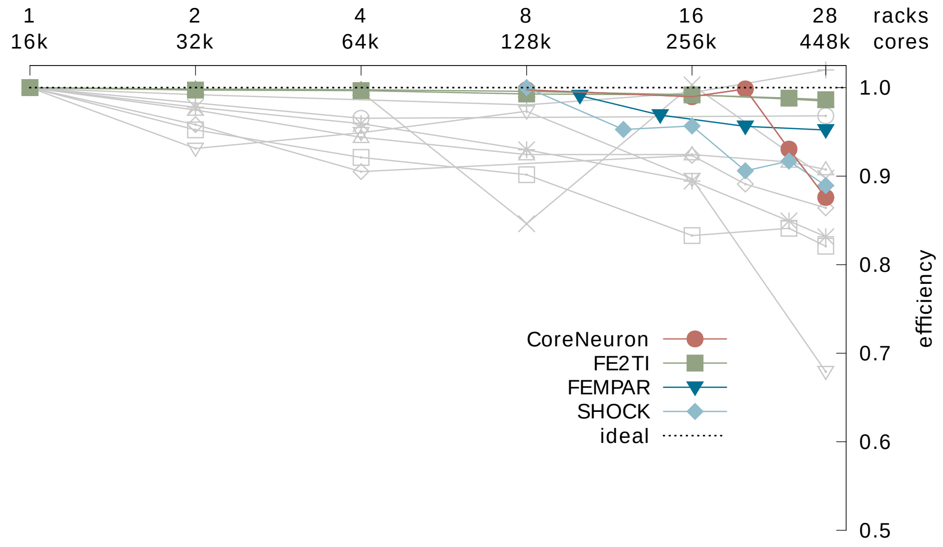


Figure 4. Weak scaling results of the workshop codes with results from existing High-Q Club members included in light grey.

In Figure 4 FE^{2}TI is able to sustain weak scaling efficiency of 99% with 28 racks, whereas for CoreNeuron efficiency drops to a still respectable 88% due to a load-balancing artifact of the input data configuration. Various codes show erratic scaling performance, most likely due to topological effects. SHOCK is characterised by particularly poorly performing configurations with an odd number of racks in one dimension (i.e. 4×3, 4×5 and 4×7 as seen in Figure 1).

4. Parallel Performance Utilities

Managing I/O A critical point attracting increasing attention is performance of file I/O, which is often a scalability constraint for codes which need to read and write huge datasets or open a large number of files. MPAS-A file reading was tuned via environment variables and employing 128 ranks to read and distribute data for the 16 384 tasks in each rack, however, it still needed 20 minutes to load its 1.2 TB of initial condition data using PIO/NetCDF — after taking the better part of a week to transfer this single file from KIT (due to the source institution policy limiting outgoing transfer bandwidth) — while simulation output was disabled for these tests to avoid similar writing inefficiency. Large-scale executions of ICON, psOpen and SHOCK (and various High-Q member codes) using the popular HDF5 and pNetCDF libraries needed to disable file I/O and synthesise initialisation data, whereas CoreNeuron replicated a small dataset to fill memory to 15.9 GB.

SIONlib [10], which was developed to address file I/O scalability limitations, has been used effectively by three High-Q codes and several others are currently migrating to adopt it.

Tools at scale Darshan [11] was engaged with SHOCK and various other codes to investigate I/O performance on JUQUEEN and identify copious reads of small numbers of bytes. The Score-P instrumentation and measurement infrastructure [12] employed by the latest release of Scalasca [13] was used to profile file I/O performance of MPAS-A, however, only MPI call count and timing is currently supported and not measurement of bytes read or written. While Score-P profiles have been produced for applications with one million threads, the largest trace collection configuration currently handled with OTF2 is approximately 655 360 threads (or processes). SIONlib needs to be employed for such traces to avoid file creation limitations and have managable numbers of files.

Custom mappings of MPI process ranks to JUQUEEN compute nodes generated by the Rubik [14] tool were investigated with psOpen and found to deliver some benefits, however, for the largest machine partitions these did not provide the expected reduction in communication times yet suffered from increased application launch/initialisation time.

5. Conclusions

The 2015 JUQUEEN Extreme Scaling Workshop surpassed our expectations and completely achieved its goal: all seven teams succeeded in running and validating their codes on 28 racks within the first 24 hours of access to the full JUQUEEN system. They also demonstrated excellent strong and/or weak scaling which qualified five new members for the High-Q Club: unfortunately, MPAS-A scaling was limited to only 24 racks (393 216 cores). In this case, the dataset used was insufficient to have a performance benefit with 28 racks.

Most optimisations employed by the codes are not specific to Blue Gene (or BG/Q) systems, but can also be exploited on other highly-parallel computer systems. High-Q Club codes have also run at scale on various Cray supercomputers, K computer, MareNostrum-3, SuperMUC and other x86-based computers, as well as on systems with GPGPUs.

Feedback from participants confirmed that the workshop facilitated exchange of ideas that empowered them to identify additional optimisation opportunities that they could exploit. Detailed results for each code are found in chapters contributed to the workshop report [1] by each of the participating teams. These present and discuss more execution configurations and scaling results achieved by the application codes during the workshop.

Acknowledgments

Particular thanks are due to Sandra Diaz, Markus Geimer, Klaus Görgen, Sabine Grießbach, Lars Hoffmann, Michael Knobloch, Alex Peyser, Christoph Pospiech, Michael Rambadt, Michael Schlottke, Michael Stephan, Alexandre Strube and Kay Thust, and the workshop participants themselves who openly shared their own knowledge and expertise.

References

[1] Dirk Brömmel, Wolfgang Frings & Brian J. N. Wylie, JUQUEEN Extreme Scaling Workshop 2015, Technical Report FZJ-JSC-IB-2015-01, Forschungszentrum Jülich, Feb. 2015.
http://juser.fz-juelich.de/record/188191

[2] Bernd Mohr & Wolfgang Frings, Jülich Blue Gene/P Porting, Tuning & Scaling Workshop 2008, Innovatives Supercomputing in Deutschland, inSiDE 6(2), 2008.

[3] Bernd Mohr & Wolfgang Frings, Jülich Blue Gene/P Extreme Scaling Workshop 2009, Technical Report FZJ-JSC-IB-2010-02, Forschungszentrum Jülich, Feb. 2010.
http://juser.fz-juelich.de/record/8924

[4] Bernd Mohr & Wolfgang Frings, Jülich Blue Gene/P Extreme Scaling Workshop 2010, Technical Report FZJ-JSC-IB-2010-03, Forschungszentrum Jülich, May 2010.
http://juser.fz-juelich.de/record/9600

[5] Bernd Mohr & Wolfgang Frings, Jülich Blue Gene/P Extreme Scaling Workshop 2011, Technical Report FZJ-JSC-IB-2011-02, Forschungszentrum Jülich, Apr. 2011.
http://juser.fz-juelich.de/record/15866

[6] Helmut Satzger et al, Extreme Scaling of Real World Applications to >130,000 Cores on SuperMUC, Poster, Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC13, Denver, CO, USA), Nov. 2013.

[7] Ferdinand Jamitzky & Helmut Satzger, 2nd Extreme Scaling Workshop on SuperMUC, Innovatives Supercomputing in Deutschland, inSiDE 12(2), 2014.

[8] Michael Stephan & Jutta Docter, JUQUEEN: IBM Blue Gene/Q supercomputer system at the Jülich Supercomputing Centre. Journal of Large-scale Research Facilities (1), A1, 2015. http://dx.doi.org/10.17815/jlsrf-1-18

[9] The High-Q Club at JSC. http://www.fz-juelich.de/ias/jsc/high-q-club

[10] SIONlib: Scalable I/O library for parallel access to task-local files.
http://www.fz-juelich.de/jsc/sionlib

[11] Darshan: HPC I/O characterisation tool, Argonne National Laboratory.
http://www.mcs.anl.gov/research/projects/darshan/

[12] Score-P: Community-developed scalable instrumentation and measurement infrastructure.
http://www.score-p.org/

[13] Scalasca: Toolset for scalable performance analysis of large-scale parallel applications.
http://www.scalasca.org/

[14] Rubik tool for generating structured Cartesian communication mappings, Lawrence Livermore National Laboratory. https://computation.llnl.gov/project/performance-analysis-through-visualization/software.php

High-Q Club codes

The full description of the High-Q Club codes along with developer and contact information can be found on the web page [9]. The current list also includes:

CIAO *advanced reactive turbulent simulations with overset*
RWTH Aachen University ITV and Sogang University

dynQCD *lattice quantum chromodynamics with dynamical fermions*
JSC SimLab Nuclear and Particle Physics & Universität Wuppertal

Gysela *gyrokinetic semi-Lagrangian code for plasma turbulence simulations*
CEA-IRFM Cadarache

IMD *classical molecular dynamics simulations*
Ruhr-Universität Bochum & JSC SimLab Molecular Systems

JURASSIC *solver for infrared radiative transfer in the Earth's atmosphere*
JSC SimLab Climate Science

JuSPIC *relativistic particle-in-cell code for plasmas and laser-plasma interaction*
JSC SimLab Plasma Physics

KKRnano *Korringa-Kohn-Rostoker Green function code for quantum description of nano-materials in all-electron density-functional calculations*
FZJ-IAS

LAMMPS(DCM) *molecular dynamics simulation with dynamic cutoff method*
Aachen Inst. for Advanced Study in Computational Engineering Science

MP2C *massively-parallel multi-particle collision dynamics for soft matter physics and mesoscopic hydrodynamics*
JSC SimLab Molecular Systems

$\mu\phi$ (muPhi) *water flow and solute transport in porous media, algebraic multi-grid*
Universität Heidelberg

Musubi *multi-component Lattice Boltzmann solver for flow simulations*
Universität Siegen

NEST *large-scale simulations of biological neuronal networks*
FZJ/INM-6 & IAS-6

OpenTBL *direct numerical simulation of turbulent flows*
Universidad Politécnica de Madrid

PEPC *tree code for N-body simulations, beam-plasma interaction, vortex dynamics, gravitational interaction, molecular dynamics simulations*
JSC SimLab Plasma Physics

PMG+PFASST *space-time parallel solver for ODE systems with linear stiff terms*
LBNL, Universität Wuppertal, Università della Svizzera italiana & JSC

PP-Code *simulations of relativistic and non-relativistic astrophysical plasmas*
University of Copenhagen

TERRA-NEO *modeling and simulation of Earth mantle dynamics*
Universität Erlangen-Nürnberg, LMU & TUM

waLBerla *Lattice-Boltzmann method for the simulation of fluid scenarios*
Universität Erlangen-Nürnberg

ZFS *multiphysics framework for flows, aero-acoustics and combustion*
RWTH Aachen AIA and JSC SimLab Fluids & Solids Engineering