Outline

- Navigating modules
- Compiling for the Booster
- Sequential Libraries
- Parallel Libraries and Application Systems:
  - Threaded Libraries
  - MPI parallel Libraries
  - Application Software
- Software for Materials Science
- Software for Computational Engineering
- Further Information
Modules environment

Figure: Current toolchain tree in JURECA and JUWELS
Figure: Current toolchain tree in Booster
Hierarchical modules

- GCCcore/.8.3.0 is preloaded, which enables a lot of base software
- For HPC software you have to load a compiler, to expand the module tree
  ml Intel
- Then you load an MPI version
  ml ParaStationMPI
  Default version is 5.2.2-1 (ie: MPI_THREAD_MULTIPLE is not supported, to get it you need 5.2.2-1-mt)
- Then you can load any other packages, for example
  ml QuantumESPRESSO/6.4.1
Modules environment

- After loading compiler and MPI `ml avail` shows the software available with that combination

- `ml avail name` and `ml help name` will show you details about the `name` package

- Many libraries are available for more than one combination/toolchain

- Write e-mail to `sc@fz-juelich.de` if you want special versions or new software
  - No guarantee the software will be installed

- `$EBROOTNAME` is the root directory where the library is installed
Modules environment

- `ml spider name` shows whether a library is available in the current stage and in which versions.
- `ml spider name/version` shows which environment you have to load before you can load that version.

- Many packages are hidden. To see them use `ml --show-hidden spider name`.
For R, Python and Perl we use bundles

- You might be looking for a software package that is part of a bigger module

- Use `ml key software`
  `ml key numpy will suggest SciPy-Stack`

- You can use then `ml spider` to find out how to load the module
Modules environment

Stages

- The whole software stack in JURECA, the Booster, and JUWELS is updated every 6 months
  - Right when there is an allocation for new projects
- Old stages are still accessible
- To check availability in other stages first type
  JURECA:
  ```
  ml use /usr/local/software/jureca/OtherStages
  ```
  Booster:
  ```
  ml use /usr/local/software/jurecabooster/OtherStages
  ```
  JUWELS:
  ```
  ml use /gpfs/software/juwels/otherstages
  ```
Compiling for the Booster (I)

- Cross-compilation on login nodes not recommended but technically possible
- ml Architecture/KNL
  - This should be the first module you load
  - Then you have the booster software stack available.
  - This doesn’t work on the cluster compute nodes!

- Remember, cross-compiling is not recommended unless you know what you are doing
Recommended way to compile for the Booster

- Start an interactive session, by
  `salloc --partition=develbooster`

- After the allocation is successful start a remote shell from within the `salloc` session and connect it to a pseudo terminal using
  `srun --cpu_bind=none --nodes=1 --pty /bin/bash -i`

  Architecture is switched to KNL automatically then

- In batch scripts add
  `module load Architecture/KNL`
  before all other module commands
Sequential Libraries and Packages (I)

Remember that with intel compilers if you want to link Fortran subroutines with the C linker you need to add -lifcore -lifport

Vendor specific Libraries
- MKL Intel® Math Kernel Library versions as mentioned in general informations, 2019.3.199 on JURECA, Booster, and JUWELS

For more information see http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/SystemDependentLibraries/MKL.html?nn=1035570
Sequential Libraries and Packages (II)

Public domain Libraries
- LAPACK (Linear Algebra PACKage)
- ARPACK (Arnoldi PACKage)
- GSL (Gnu Scientific Library)
- GMP (Gnu Multiple Precision Arithmetic Library)

Commercial library
NAG Fortran and C Library: JURECA Intel compiler only
Contents of Intel® MKL

- BLAS, Sparse BLAS, CBLAS
- LAPACK
- Iterative Sparse Solvers, Trust Region Solver
- Vector Math Library
- Vector Statistical Library
- Fourier Transform Functions
- Trigonometric Transform Functions
Contents of Intel® MKL

- GMP routines
- Poisson Library
- Interface for fftw

Usage of MKL (I)

- Can be loaded with Intel compiler or GCC
- MPI has to be loaded before imkl
- FORTRAN, C, and C++ callable
- Arrays FORTRAN like, i.e. column-first (except cblas)
- Compilation and linking of program name.f calling sequential MKL routines:
  
  ```bash
  ifort name.f -o name -lmkl_intel_lp64 -lmkl_sequential -lmkl_core [-liomp5 -lpthread]
  ```
Usage of MKL (II)

To use CBLAS include mkl.h into source code

Compilation and linking of program name.c calling sequential MKL

```bash
icc name.c -o name -lmkl_intel_lp64 -lmkl_sequential -lmkl_core [-liomp5 -lpthread -lifcore -lifport]
```

For flags to compile with other compilers (PGI, GCC) or multithreaded versions take a look at:

LAPACK

- Part of MKL in libmkl_core.a
- Can be loaded with Intel or GCC on JURECA, Booster, and JUWELS
- In older Stages also part of OpenBLAS with GCC (JURECA only)

Arpack

- ARPACK-NG/3.7.0
- Iterative solver for sparse eigenvalue problems
- Reverse communication interface
- FORTRAN 77
- Calls LAPACK and BLAS routines, MKL necessary
GSL – GNU Scientific Library

- module load Intel GSL/2.5 for icc version on JURECA and JUWELS
- module load GCC/8.3.0 GSL/2.5 for gcc version on JURECA and JUWELS
- Provides a wide range of mathematical routines
- Not recommended for performance reasons
- Often used by configure scripts
GMP - GNU Multiple Precision Library

version 6.1.2 on JURECA

NAG Libraries

- NAG Fortran Mark 26 and NAGC Mark 26 on JURECA only available with Intel compiler
- Please tell us if you really need it
MKL
is multi-threaded when linked with
-lmkl_[intel,gnu,pgi]_threaded
if OMP_NUM_THREADS is not set,
48 threads used on JURECA
68 threads on Booster
96 threads on JEWELS (80 in accelerated nodes)
Usage:
ifort name.f -o name -lmkl_intel_lp64
-lmkl_intel_thread -lmkl_core -liomp5 -lpthread

FFTW 3.3.8 (Fastest Fourier Transform of the West)
MPI, OpenMP and threads version
http://www.fftw.org
Parallel Libraries

MPI Parallelism

- ScaLAPACK (Scalable Linear Algebra PACKage)
- ELPA (Eigenvalue Solvers for Petaflop-Applications)
- Elemental, C++ framework for parallel dense linear algebra
- FFTW (Fastest Fourier Transform of the West)
- MUMPS (MUltifrontal Massively Parallel sparse direct Solver)
- ParMETIS (Parallel Graph Partitioning)
- Hypre (high performance preconditioners)
MPI Parallelism (II)

- PARPACK (Parallel ARPACK), Eigensolver
- SPRNG (Scalable Parallel Random Number Generator)
- SUNDIALS (SUite of Nonlinear and DIfferential/ALgebraic equation Solvers)

Parallel Systems, MPI Parallelism
- PETSc, toolkit for partial differential equations
ScaLAPACK

- part of MKL,
- Parallel BLAS 1-3, PBLAS Version 2
- Dense linear system solvers
- Banded linear system solvers
- Solvers for Linear Least Squares Problem
- Singular value decomposition
- Eigenvalues and eigenvectors of dense symmetric/hermitian matrices

http://www.netlib.org/scalapack/index.html
Usage of ScaLAPACK

Linking a program name.f calling routines from ScaLAPACK, default version, Intel compiler:
mpif77 name.f -lmkl_scalapack_lp64
-lmkl_blacs_intelmpi_lp64 -lmkl_intel_lp64
-lmkl_intel_thread[-lmkl_sequential]
-lmkl_core -liomp5 -lpthread
ELPA
Eigenvalue Solvers for Petaflop-Applications

ELPA uses ScaLAPACK, must be linked together with scalapack

- FORTRAN 95, same data-distribution as ScaLAPACK
- JURECA pure MPI and hybrid version 2018.11.001
- Version with GPU acceleration on JURECA and JUWELS
- Booster and JUWELS special version with AVX512 kernels for ELPA2
- Version for GPU usage on JURECA and JUWELS
Elemental

- C++ framework, two-dimensional data distribution element by element
- http://libelemental.org/about/
- 0.87.7
- Only with Intel compilers available
MUMPS
MUltifrontal Massively Parallel sparse direct Solver

- Solution of linear systems with symmetric positive definite matrices, general symmetric matrices, general unsymmetric matrices
- Real or Complex
- Parallel factorization and solve phase, iterative refinement and backward error analysis
- F90 with MPI and OpenMP since 5.1.1
- current version 5.1.2
- http://graal.ens-lyon.fr/MUMPS/
ParMETIS

Parallel Graph Partitioning and Fill-reducing Matrix Ordering developed in Karypis Lab at the University of Minnesota 4.0.3 on JURECA and JUWELS
http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview
Version with double precision real values available

Hypre

High performance preconditioners
Version 2.15.1 on JURECA and JUWELS, also version with bigint,
http://www.llnl.gov/CASC/hypre/software.html
FFTW

3.3.8 on JURECA and JUWELS, (Intel and GCC modules)

PARPACK

- PARPACK MPI-Version, part of ARPACK-NG/3.7.0
- Must be linked with LAPACK and BLAS
- Reverse communication interface, user has to supply parallel matrix-vector multiplication

https://github.com/opencollab/arpack-ng
http://www.caam.rice.edu/~kristyn/parpack_home.html
The Scalable Parallel Random Number Generators Library for ASCI Monte Carlo Computations version 5.0-14042019:
various random number generators in one library
Version 1.0 separate library for each random number generator
http://sprng.cs.fsu.edu/

Sundials (CVODE)

Package for the solution of ordinary differential equations, Version 4.1.0
https://computation.llnl.gov/casc/sundials/main.html
PETSc

- Portable, Extensible Toolkit for Scientific Computation
- Numerical solution of partial differential equations
- version 3.11.1
- with several other packages included
- complex version and version with 8-Byte integer
- debug versions in Devel Stages only
- http://www.mcs.anl.gov/petsc/
- ml spider petsc
<table>
<thead>
<tr>
<th>Package</th>
<th>JURECA</th>
<th>JUWELS</th>
<th>Booster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abinit</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>ADF</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>Amber</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>CP2K</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>CPMD</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>GPAW</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Gromacs</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>LAMMPS</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Molpro</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>NAMD</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>NWChem</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>QuantumEspresso</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>TURBOMOLE</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>
Software for Computational Engineering

- JURECA and JUWELS
- CFD Package **OpenFOAM**
  - Version 4.1 in Stages/2017b and some older versions in older stages
  - Version 5.0 in Stages/2018a
  - and OpenFOAM-Extend 3.1 and 3.2, only in older stages
- Commercial **FEM Software**
  - **ANSYS**, **LS-DYNA**, **COMSOL** are technically maintained on **JURECA** only
  - **Licenses** must be provided by the **User**!
Further information and JSC-people

http://www.fz-juelich.de/ias/jsc/jureca
http://www.fz-juelich.de/ias/jsc/EN/Expertise/Support/Software/_node.html

Mailto
Supercomputer support:
sc@fz-juelich.de

I. Gutheil: Parallel mathematical Libraries
i.gutheil@fz-juelich.de

D. Alvarez: Software Manager
d.alvarez@fz-juelich.de

B. Körfgen: Physics and Engineering software
b.koerfgen@fz-juelich.de