WIEN2k- hardware/software

- WIEN2k runs on any Linux platform from PCs, Macs, workstations, clusters to supercomputers
  - *Intel **I7** quad (six)-core processors with fast memory bus (1.5-3 Gb/core, Gbit-network, SATA disks). 1000 € /PC,*
  - *with a few such PCs you have a quite powerful cluster (k-parallel)*
  - *60 - 100 atom / cell, requires 2-4 Gb RAM*
  - *installation support for many platforms + compiler*

- Fortran90 (dynamical allocation, modules)
  - *real/complex version (inversion)*
  - *many individual modules, linked together with C-shell or perl-scripts*
- *web-based GUI – w2web (perl)*

- f90 compiler (**ifort**, gfortran), BLAS-library (**mkl**, gotolib), FFTW, perl5, ghostscript (+jpg), gnuplot(+png), Tcl/Tk (Xcrysden), pdf-reader, www-browser, **octave, opendx**

Installation of WIEN2k

- Register via http://www.wien2k.at
- Create your $WIENROOT directory (e.g. ./WIEN2k)
- Download wien2k_13.tar and examples (executables)
- Uncompress and expand all files using:
  - *tar –xvf wien2k_12.tar*
  - *gunzip *.gz*
  - *./expand_lapw*
- This leads to the following directories:
  - *./SRC* *(scripts, ug.ps)*
  - *./SRC_aim* *(programs)*
  - *...*
  - *SRC_templates* *(example inputs)*
  - *...*
  - *SRC_usersguide_html* *(HTML-version of UG)*
  - *example_struct_files* *(examples)*
  - *TiC*

siteconfig_lapw

```
*************************************************
*                    W I E N                    *
*                site configuration             *
*************************************************
   S   specify a system
   C   specify compiler
   O   specify compiler options, BLAS and LAPACK
   P   configure Parallel execution
   D   Dimension Parameters
   R   Compile/Recompile
   U   Update a package
   L   Perl path (if not in /usr/bin/perl)
   Q   Quit
```

D: define NMATMAX (adjust to your hardware/paging!):
NMATMAX=5000 ➔256Mb (real) or 500Mb (complex)
NMATMAX=10000 ➔ 1Gb (real) or 2Gb (complex) ➔ 80-100 atoms/unitcell
NUME=1000 → number of eigenvalues (adjust to NMATMAX)

Compilation

- recommendation: **Intels Fortran compiler** (includes mkl)

  free for non-commercial (but not for academic), www intel.com

  - *`which ifort`* → *tells you if you can use ifort and which version you have*
    - usually installed in **/opt/intel/composerxe-2011..../bin/intel64 (ls ....)**
    - **include ifortvars.csh** and mklvars.csh in your .bashrc/.cshrc file:
      - source /opt/intel/11.0/074/bin/ifortvars.csh intel64
      - source /opt/intel/11.0/074/mkl/tools/environment/mklvarsem64t.csh
  - ***ifort 12*** *(vers. 8.0 and early 12.x buggy, 9.x, 10.0, 11.x ok)*
    - for older versions dynamic linking recommended (depends on ifort version, requires system and compiler libraries at runtime, needs $LD_LIBRARY_PATH)
    - IA32 bit, IA64 bit (Itanium) or **Intel64 (em64t)** -version
    - mkl-library: names change with every version, see:
      http://software.intel.com/en-us/articles/intel-mkl-link-line-advisor
    - 9.x: -L/opt/intel/mkl/lib -lmkl_lapack –lmkl_em64t -lmkl_core (→libmkl_core.so)
    - 10.0: -L/opt/intel/mkl/lib -lmkl_lapack –lmkl
  - *compiler/linker options depend on compiler version + Linux-version !!*
    - -FR (free format) -lguide –lpthread -pthread

compilation

- ***gfortran*** *+* ***gotolib,*** *acml-lib, ATLAS-BLAS*
  - -static linking possible

- siteconfig has support for various ifort versions and gfortran

- it does NOT make sense to invest in new hardware but use a „free“ compiler

userconfig_lapw

- Every user should run userconfig_lapw (setup of environment)
  - *support for tcsh and bash (requires .cshrc or .bashrc)*
  - *sets PATH to $WIENROOT, sets variables and aliases*
    - $WIENROOT, $SCRATCH, $EDITOR, $PDFREADER, $STRUCTEDIT_PATH
    - pslapw: ps –ef | grep lapw
    - lsi: ls –als *.in* lso: ls -als *.output*
    - lss: *.scf* lsc: *.clm*
    - limit stacksize unlimited (otherwise: “segmentation fault”)
    - $OMP_NUM_THREADS (for mkl+multi-core); $LD_LIBRARY_PATH

w2web

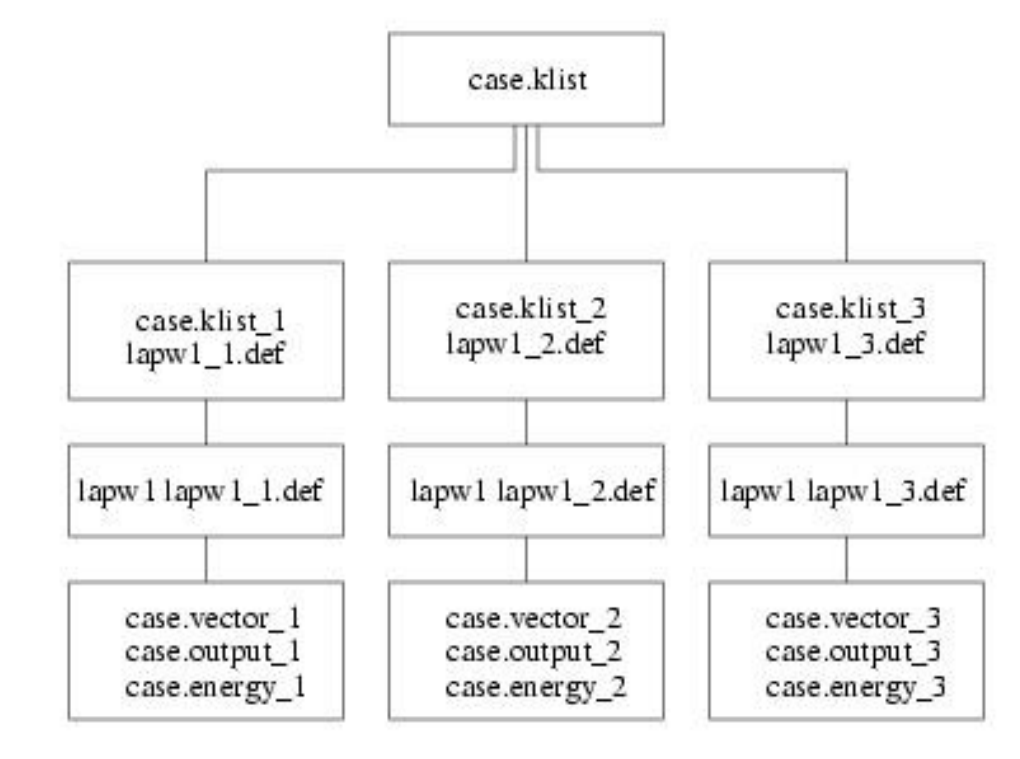
- w2web: acts as webserver on a userdefined (high) port.
  - *define user/password and port. (http://host.domain.xx:5000)*
  - *behind firewall create a „ssh-tunnel“: ssh -fNL 2000:host:2000 user@host*
  - *~/.w2web/hostname/conf/w2web.conf: (configuration file)*
    - deny=*.*.*.*
    - allow=128.130.134.* 128.130.142.10
    - define execution types: NAME=commands (eg.: batch=batch < %f)

k-point Parallelization (lapw1+lapw2)

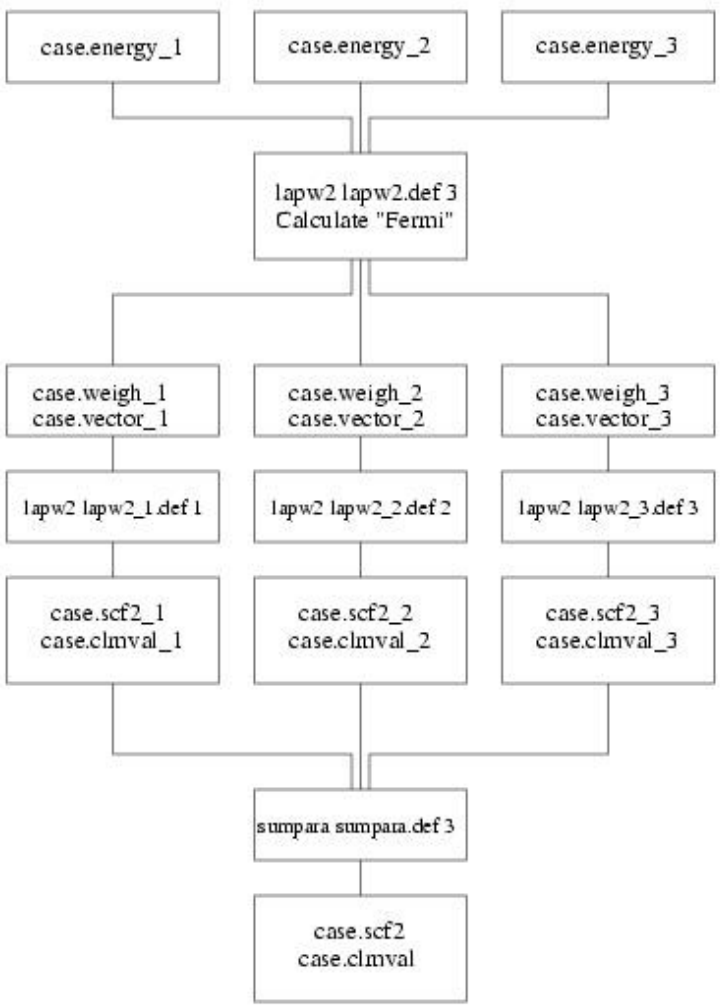
- very efficient parallelization even on loosely coupled PCs (slow network):
  - *common NFS filesystem (files must be accessible with the same path on all machines; use /host1 as data-directory on host1)*
  - *rsh/ssh without password (.rhosts; private/public keys)*
    - ssh-keygen –t rsa
    - append .ssh/authorized_keys on remote host with id_rsa.pub of local host
    - .machines file:
      - 1:host1 (speed:hostname)
      - 2:host2
      - granularity:1 (1:10k+20k; 3: 3+6+3+6+3+6+rest ➔load balancing, not with $SCRATCH, -it
      - extrafine:1 (rest in junks of 1 k)
    - testpara (tests distribution); run_lapw –p
  - *case must fit into memory of one PC !*
  - *high NFS load: use local $SCRATCH directory (only with commensurate k-points/hosts)*
  - *$OMP_NUM_THREADS (parallel diag. with mkl on multi-core CPU)*

Flow of parallel execution

lapw1para

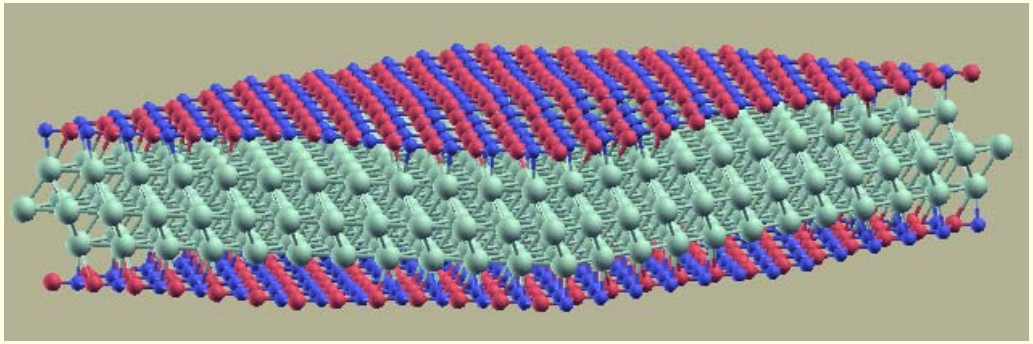


lapw2para



fine-grain mpi-parallelization

- for **bigger** cases (> 50 atoms) and **more** than **4 cores**
- fast network (~~Gbit~~, Myrinet, **Infiniband**, shared memory machines)
- mpi (you need to know which mpi is installed (mpich-1.2, open-mpi, intel-mpi,...)
  - *mpif90 or mpiifort*
- scalapack (included in ifort 11)
  - *llibmkl_blacs_lp64.a or ibmkl_blacs_openmpi_lp64.a or libmkl_blacs_intelmpi_lp64.a*
- FFTW (v. 2 or 3 ; mpi and sequ. version needed, -DFFTW2/3 in Makefiles)
- .machines file:
  - 1:host1:4 host2:4 — 8 mpi-parallel jobs on host1 and host2
  - lapw0:host1:4 host2:4 — 8 parallel jobs; atom-loops only + fft !!!
- simultaneous k-point and mpi-parallelization possible
  - *BN/Rh(111) nanomesh: cell with 1100 atoms*
    - NMAT=45000-80000; 64 cpus, 2h / iteration; scales to at least 512 cores



WIEN2k_13.1

- always use latest version (bug fixes, improved performance, new features)

- eventually: use prebuilt executables from our website !!

Getting help

- *_lapw –h „help switch“ of all WIEN2k-scripts
- help_lapw:
  - *opens usersguide.pdf; Use ^f keyword to search for an item („index“)*
- html-version of the UG: ($WIENROOT/SRC_usersguide/usersguide.html)
- http://www.wien2k.at/reg_user
  - *FAQ page with answers to common questions*
  - *Update information: When you think the program has an error, please check newest version*
  - *Textbook section: DFT and the family of LAPW methods by S.Cottenier*
  - *Mailing-list:*
    - subscribe to the list (always use the same email)
    - full text search of the „digest“ (your questions may have been answered before)
    - posting questions: Provide sufficient information, locate your problem (case.dayfile, *.error, case.scf, case.outputX).
    - „My calculation crashed. Please help.“ This will most likely not be answered.