

UserLibraries



Unknown macro: 'composition-setup'

Contents

- [GotoBLAS](#)
- [PETSc](#)
- [Trilinos](#)
- [VASP](#)

GotoBLAS

Download this patch to build version 1.02 on Blue Gene and test with Cobalt:

- [GotoBLAS-1.02-bgl.patch](#)

Patch GotoBLAS 1.02, compile and submit tests using Cobalt:

```
$ cd GotoBLAS
$ patch -p1 < ../GotoBLAS-1.02-bgl.patch
$ make
$ cd test;make
```

PETSc

Configuration files for various PETSc versions are located on Frost under the directory `/contrib/bgl/petsc`.

For instance, here is one way to build `petsc-3.0.0-p4` on Frost:

```
$ cd petsc-3.0.0-p4/
$ cp /contrib/bgl/petsc/petsc-3.0.0-p4/bgl-ibm-goto_lapack.py config/
$ patch -p0 < /contrib/bgl/petsc/petsc-3.0.0-p4/petsc-3.0.0-p4.patch

$ ./config/bgl-ibm-goto_lapack.py

(set PETSC_ARCH and PETSC_DIR environment variables per configure output)

$ make all
```

Trilinos

example config

```
CC=mpxlc CXX=mpxlc CXXFLAGS="-DMPICH_IGNORE_CXX_SEEK" F77=mpxlf ./configure \
--prefix=/home/voran/local/trilinos \
--enable-triutils \
--enable-teuchos \
--enable-epetraext \
--enable-epetra \
--enable-aztecoo \
--enable-ifpack \
--enable-amesos \
--enable-mpi \
--with-blas=/contrib/bgl/lib/libblas440.a \
--with-lapack=/contrib/bgl/lib/liblapack440.a \
--with-gnumake
```

comment these lines out in packages/triutils/src/Triutils_config.h:

```
/* Define to 1 if you have the <stdio> header file. */
/* #define HAVE_CSTDIO 1 */
/* #define HAVE_CSTDLIB 1 */
```

compile

```
make
make install
```

VASP

The installation instructions for VASP are located here: http://cms.mpi.univie.ac.at/vasp/vasp/Installation_VASP.html

Makefiles that are known to work for VASP 5.2 on Blue Gene/L systems have been placed in the /contrib/bgl/vasp directory on Frost. They are also displayed below for reference. Note that we cannot install VASP system-wide because we are not licensed to do so.

VASP may be built on Frost by following steps like these:


Example VASP build steps

```
$ cd vasp.5.lib
$ cp /contrib/bgl/vasp/makefile.bgl.5.lib Makefile
$ make


$ cd vasp.5.2
$ cp /contrib/bgl/vasp/makefile.bgl.5.2 Makefile
$ make clean
$ make
```

Refer to the VASP documentation for information on configuring experiments: <http://cms.mpi.univie.ac.at/vasp/vasp/vasp.html>

Makefiles listed below for reference:

 Unknown macro: 'toggle-cloak'

Makefile for VASP 5 library

 Unknown macro: 'cloak'

/contrib/bgl/vasp/makefile.bgl.5.lib

```
.SUFFIXES: .inc .f .F
#-----
# Makefile for IBM BG/L
#-----

# C-preprocessor
CPP      = cpp -P -C   $*.F >$*.f
FC=mpixlf90 -g
CC=mpixlc -g

CFLAGS = -O3 -qstrict -qarch=440
FFLAGS = -O3 -qstrict -qarch=440
FREE    = -qfree=f90
NOFREE  = -qfixed

DOBJ =  preclib.o timing.o derrf.o dclock.o diolib.o dlexlib.o drdatab.o

#-----
# general rules
#-----

libdmy.a: $(DOBJ) linpack_double.o
        rm -f libdmy.a
        ar vq libdmy.a $(DOBJ) linpack_double.o

# files which do not require autodouble
lapack_double.o: lapack_double.f
        $(FC) $(FFLAGS) $(NOFREE) -c lapack_double.f
lapack_single.o: lapack_single.f
        $(FC) $(FFLAGS) $(NOFREE) -c lapack_single.f
linpack_double.o: linpack_double.f
        $(FC) $(FFLAGS) $(NOFREE) -c linpack_double.f
linpack_single.o: linpack_single.f
        $(FC) $(FFLAGS) $(NOFREE) -c linpack_single.f

.c.o:
        $(CC) $(CFLAGS) -c $*.c
.F.o:
        $(CPP)
        $(FC) $(FFLAGS) $(FREE) $(INCS) -c $*.f
.F.f:
        $(CPP)
.f.o:
        $(FC) $(FFLAGS) $(FREE) $(INCS) -c $*.f
```



Unknown macro: 'cloak'



Unknown macro: 'toggle-cloak'

Makefile for VASP 5.2 directory



Unknown macro: 'cloak'

/contrib/bgl/vasp/makefile.bgl.5.2

```
.SUFFIXES: .inc .f .F
#-----
# Makefile for RS 6000/ SP2, SP3 and possibly SP4, modified for BG/L
```

```

# you might also try the makefile.hlrn supplied by Bernd Kallies <kallies@zib.de>
#
#
# Attention:
# =====
# present default optimisation is -O3, but you might try -O4 as well
#
# several files are optimized less aggressive (see bottom of the makefile),
# since aggressive optimisation of those files caused VASP to bomb
# in one or the other compiler version
# because it was impossible to keep track of which file is not
# properly compiled by which version, all files that
# are problematic in one or the other version are compiled
# with lower optimisation
#
# ZHEEVX was not working properly with some version
# if you have problems remove
# #define USE_ZHEEVX
# from subrot.F, davidson.F and wavepre_noio.F
#
#-----

# all CPP processed fortran files have the extension .f
SUFFIX=.f

#-----
# fortran compiler and linker
#-----
FC=mpixlf90
FCL=$(FC)

#-----
# C-preprocessor define any of the flags given below
# MPI                generate parallel version
# NGZhalf            charge density reduced in Z direction
# wNGZhalf           gamma point only reduced in Z direction
# CACHE_SIZE         5001    for SP3 and Power 3
#                   32768    for 550,590,3CT
#                   8001     595/397 quad word systems
# scaLAPACK          use scaLAPACK
#
# IBM
# use_allreduce : force mpi_allreduce usage by block of MPI_BLOCK
#-----
# Add -DwNGZhalf for gamma point only

# CPP      = /usr/bin/cpp -P -C -DHOST="\BlueGene\" -DMPI -DNGZhalf \
#           -Duse_collective -Davoidalloc -DscaLAPACK \
#           -DCACHE_SIZE=4000 -DMPI_BLOCK=50000 *.F >*.f

CPP_ = ./preprocess <*.F | /usr/bin/cpp -P -C -traditional >*$(SUFFIX)
CPP  = $(CPP_) -DHOST="\BlueGene\" -DMPI -DNGZhalf \
        -Duse_collective -Davoidalloc \
        -DCACHE_SIZE=2000 -DMPI_BLOCK=50000

#-----
# general fortran flags, none required
#-----

#FFLAGS = -O3 -qstrict -qarch=440d -qtune=440 -qmaxmem=-1
FFLAGS = -g -qfree=f90 -qstrict
#PK 2009/6/3 No opt Works
#FFLAGS = -O0 -g -qstrict -qarch=450d -qtune=450 -qmaxmem=-1

#-----
# optimization:
# optimise for the machine on which the code is compiled
#-----

#OFLAG = -O3 -qstrict -qarch=450d -qtune=450 -qmaxmem=-1
OFLAG = -O3 -qarch=440 -qtune=440 -qmaxmem=-1

```

```

#PK 2009/6/3 No opt works
#OFLAG = -O0 -g -qstrict -qarch=450d -qtune=450 -qmaxmem=-1
OFLAG_HIGH = $(OFLAG)
OBJ_HIGH = none
OBJ_NOOPT = none
DEBUG = -g
INCS =
INLINE = $(OFLAG)

#-----
# maybe one need to set an include path (usually not required)
#-----

#-----
# options for linking
# the following option increases the size of the data frame
# the default makefile does not include support for scaLAPACK
# since problems with scaLAPACK were reported
#
#-----

LIBSCA = /soft/apps/SCALAPACK
LIBBLACS = /soft/apps/BLACS
LIBLOC = /contrib/bgl/lib
# LIBTOOLS = ../lib_bgl
# SCALAPACK = -L$(LIBSCA) -lscalapack -L$(LIBBLACS) -lblacsF77init_MPI-BGP-0 -lblacs_MPI-BGP-0
SCALAPACK =

#ESSL = ../vasp.5.lib/dsygv.o ../vasp.5.lib/dgegv.o -L/opt/ibmmath/essl/4.4/lib -lesslbg
#ESSL = ../vasp.5.lib/dsygv.o ../vasp.5.lib/dgegv.o -L/opt/ibmmath/essl/4.4/lib -lesslbg
#ESSL = -L/soft/apps/ESSL-4.4/lib -lesslbg

BLAS = /contrib/bgl/lib/libgoto.a

#ESSL = ../vasp.4.lib/dsygv.o ../vasp.4.lib/dgegv.o -L/opt/ibmmath/essl/4.3/lib -lesslbg /opt/ibmcmp/xlmass/bg
/4.4/bgl/lib/libmass.a /opt/ibmcmp/xlmass/bg/4.4/bgl/lib/libmassv.a
LIB = -L../vasp.5.lib -ldmy $(SCALAPACK) -L$(LIBLOC) -llapack440 $(BLAS)

#-----
# specify 3d-fft to be used with VASP
#-----
FFT3D = fftmpi.o fftmpi_map.o fft3dfurth.o fft3dlib.o

#-----
# general rules and compile lines
#-----
BASIC= symmetry.o symlib.o lattlib.o random.o

SOURCE= base.o mpi.o smart_allocate.o xml.o \
constant.o jacobi.o main_mpi.o scala.o \
asa.o lattice.o poscar.o ini.o xclib.o xclib_grad.o \
radial.o pseudo.o mgrid.o gridq.o ebs.o \
mkpoints.o wave.o wave_mpi.o wave_high.o \
$(BASIC) nonl.o nonlr.o nonl_high.o dfast.o choleski2.o \
mix.o hamil.o xcgrad.o xcspin.o potex1.o potex2.o \
metagga.o constrmag.o cl_shift.o relativistic.o LDAPU.o \
paw_base.o egrad.o pawsym.o pawfock.o pawlhf.o paw.o \
mkpoints_full.o charge.o dipol.o pot.o \
dos.o elf.o tet.o tetweight.o hamil_rot.o \
steep.o chain.o dyna.o sphpro.o us.o core_rel.o \
aedens.o wavpre.o wavpre_noio.o broyden.o \
dynbr.o rmm-diis.o reader.o writer.o tutor.o xml_writer.o \
brent.o stufak.o fileio.o opergrid.o stepver.o \
chgloc.o fast_aug.o fock.o mkpoints_change.o sym_grad.o \
mymath.o internals.o dimer_heyden.o dvvtrajectory.o vdwforcefield.o \
hamil_high.o nmr.o force.o \
pead.o subrot.o subrot_scf.o pwlhf.o gw_model.o optreal.o davidson.o \
electron.o rot.o electron_all.o shm.o pardens.o paircorrection.o \
optics.o constr_cell_relax.o stm.o finite_diff.o elpol.o \

```

```

hamil_lr.o rmm-diis_lr.o subrot_cluster.o subrot_lr.o \
lr_helper.o hamil_lrf.o elinear_response.o ilinear_response.o \
linear_optics.o linear_response.o \
setlocalpp.o wannier.o electron_OEP.o electron_lhf.o twoelectron4o.o \
ratpol.o screened_2e.o wave_cacher.o chi_base.o wpot.o local_field.o \
ump2.o bse.o acfdt.o chi.o sydmatt.o

```

INC=

```

vasp.bgl: $(SOURCE) $(FFT3D) $(INC) main.o
rm -f vasp.bgl
$(FCL) -o vasp.bgl main.o $(SOURCE) $(FFT3D) $(LIB) $(LINK)
makeparam: $(SOURCE) $(FFT3D) makeparam.o main.F $(INC)
$(FCL) -o makeparam $(LINK) makeparam.o $(SOURCE) $(FFT3D) $(LIB)
zgemmtest: zgemmtest.o base.o random.o $(INC)
$(FCL) -o zgemmtest $(LINK) zgemmtest.o random.o base.o $(LIB)
dgemmtest: dgemmtest.o base.o random.o $(INC)
$(FCL) -o dgemmtest $(LINK) dgemmtest.o random.o base.o $(LIB)
ffttest: base.o smart_allocate.o mpi.o mgrid.o random.o ffttest.o $(FFT3D) $(INC)
$(FCL) -o ffttest $(LINK) ffttest.o mpi.o mgrid.o random.o smart_allocate.o base.o $(FFT3D) $(LIB)
kpoints: $(SOURCE) $(FFT3D) makekpoints.o main.F $(INC)
$(FCL) -o kpoints $(LINK) makekpoints.o $(SOURCE) $(FFT3D) $(LIB)

```

clean:

```

-rm -f *.g *.f *.o *.L *.mod ; touch *.F

```

```

main.o: main$(SUFFIX)
$(FC) $(FFLAGS) $(DEBUG) $(INCS) -c main$(SUFFIX)
xcgrad.o: xcgrad$(SUFFIX)
$(FC) $(FFLAGS) $(INLINE) $(INCS) -c xcgrad$(SUFFIX)
xcspin.o: xcspin$(SUFFIX)
$(FC) $(FFLAGS) $(INLINE) $(INCS) -c xcspin$(SUFFIX)

```

```

makeparam.o: makeparam$(SUFFIX)
$(FC) $(FFLAGS) $(DEBUG) $(INCS) -c makeparam$(SUFFIX)

```

makeparam\$(SUFFIX): makeparam.F main.F

```

#
# MIND: I do not have a full dependency list for the include
# and MODULES: here are only the minimal basic dependencies
# if one structure is changed then touch_dep must be called
# with the corresponding name of the structure
#

```

```

base.o: base.inc base.F
mgrid.o: mgrid.inc mgrid.F
constant.o: constant.inc constant.F
lattice.o: lattice.inc lattice.F
setex.o: setexm.inc setex.F
pseudo.o: pseudo.inc pseudo.F
poscar.o: poscar.inc poscar.F
mkpoints.o: mkpoints.inc mkpoints.F
wave.o: wave.inc wave.F
nonl.o: nonl.inc nonl.F
nonlr.o: nonlr.inc nonlr.F

```

```

$(OBJ_HIGH):
$(CPP)
$(FC) $(FFLAGS) $(OFLAG_HIGH) $(INCS) -c $$$(SUFFIX)

```

```

$(OBJ_NOOPT):
$(CPP)
$(FC) $(FFLAGS) $(INCS) -c $$$(SUFFIX)

```

```

fft3dlib_f77.o: fft3dlib_f77.F
$(CPP)
$(F77) $(FFLAGS_F77) -c $$$(SUFFIX)

```

```

.F.o:
$(CPP)
$(FC) $(FFLAGS) $(OFLAG) $(INCS) -c $$$(SUFFIX)
.F$(SUFFIX):
$(CPP)

```

```
$(SUFFIX).o:
    $(FC) $(FFLAGS) $(OFLAG) $(INCS) -c $$$(SUFFIX)

# special rules
#-----

radial.o: radial.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O2 -c $$$(SUFFIX)

wave.o: wave.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O2 -c $$$(SUFFIX)

metagga.o: metagga.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O2 -c $$$(SUFFIX)

nonl.o: nonl.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O -c $$$(SUFFIX)

paw.o: paw.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O1 -c $$$(SUFFIX)

pseudo.o: pseudo.F
    $(CPP)
    $(FC) $(FFLAGS) $(INCS) -O1 -c $$$(SUFFIX)
```