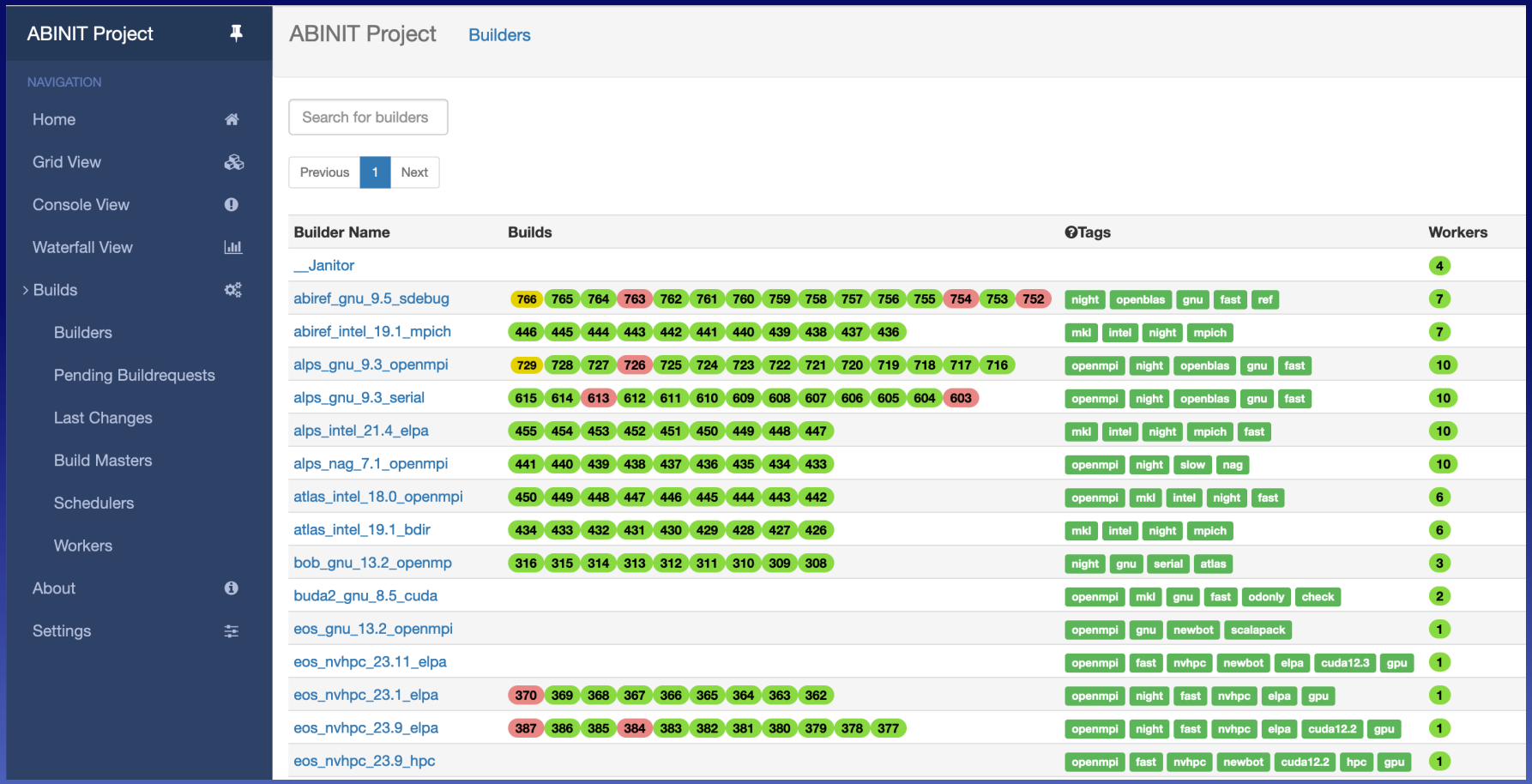
Outline

- Current state of test farm (update since 2022)
- abinit-fallbacks 10.0 : last versions of all fallbacks
- Benchmarks
- Build systems : autotools and cmake

Finally, we have successfully migrated to buildbot version 3 !

https://buildbot.abinit.org/#/builders



Test farm

Slave Matrix

Name	Brand	CPU / Freq (Ghz)	# cores (#TH)	RAM Gb	OS
abiref	HP DL360 G9	Xeon E5-2670v3 / 2.3	2 x 12 (48)	32	CentOS 7.9
alps	HP DL360 G10	Xeon Gold 6230 / 2.1	2 x 20 (80)	64	CentOS 8.3
atlas	Supermicro	Xeon E5-2623v4 / 2.6	2 x 4 (16)	64	CentOS 7.9
bob	Dell R430	Xeon E5-2603v3 / 1.60	2 x 6 (12)	8	**Fedora 39**
buda2*	Supermicro	Xeon Silver 4110 / 2.7	2 x 8 (32)	16	CentOS 7.8
eos*	Dell R7525	AMD EPYC 7643 / 2.3	2 x 48 (192)	256	Ubuntu 22.04
higgs	HP DL 360 G8	Xeon E5-2440 / 2.4	2 x 6 (24)	32	Ubuntu 18.04
minimac	Apple Mac Studio	M1 Ultra / 3.23	20 (16+4)	64	macOS 14.5
scope	HP DL385 G10	AMD EPYC 7502 / 2.5	2 x 32 (128)	96	Ubuntu 18.04
ubu	HP DL360 G9	Xeon E5-2670v3 / 2.3	2 x 12 (48)	32	Ubuntu 16.10

* buda2 is equipped with 3 GPU cards : 2 x K40c and TITAN V

* eos is equipped with 2 GPU cards : 2 x Nvidia A30

Name	Compiler	MPI	MATH	misc
abiref_gnu_9.5_debug	GNU 9.5	OpenMPI 4.0.3	OpenBLAS 0.3.7	many services
abiref_intel_19.1_mpich	intel 19.1	MPICH 3.3.1	mkl 2020	scalapack
alps_gnu_9.3_openmpi	**GNU 9.3**	**Open MPI 4.0.4**	**OpenBLAS 0.3.10**	**Reference**
alps_gnu_9.3_serial	**GNU 9.3**		**OpenBLAS 0.3.10**	**Ref for serial**
alps_intel_21.4_elpa	oneAPI 21.4	intel mpi	mkl 2021.4	elpa 2020.11
alps_nag_7.1_openmpi	NAG 7.1	Open MPI 4.1.2	NetLib 3.9	
atlas_intel_18.0_openmpi	INTEL 18.0	Open MPI 3.0.1	mkl 2018	
atlas_intel_19.1_bdir	INTEL 19.1	Open MPI 3.3.2	mkl 2019	build/ dir
bob_gnu_13.2_openmp	**GNU 13.2**		atlas 3.10	OpenMP n=2
eos_nvhpc_23.1_elpa	**nvhpc 23.1**	Open MPI 3.0.5		**cuda 12.2**
eos_nvhpc_23.9_elpa	**nvhpc 23.9**	Open MPI 4.1.2		**cuda 12.2**
higgs_intel_19.0_serial	INTEL 19.0		mkl 2019	
scope_gnu_12.2_mpich	GNU 12.2	MPICH 4.0.3	OpenBLAS	memory leak
scope_gnu_10.2_paral	**GNU 10.2**	**MPICH 3.3.2**	**OpenBLAS**	**Ref for tparal_24**
scope_gnu_13.2_dep	**GNU 13.2**	**MPICH 4.1.2**	OpenBLAS	check dependency
scope_gnu_12.2_abipy	**GNU 12.2**	**MPICH 4.0.3**	**OpenBLAS**	**check abipy**
ubu_gnu_9.2_openmpi	GNU 9.2	Open MPI 4.0.2	mkl 11.3	
ubu_intel_16.0_mpich	INTEL 16.0	MPICH 3.3.2	mkl 11.3	BigDFT
ubu_intel_16.0_openmp	INTEL 16.0		mkl 11.3	OpenMP n=2

nightly			
Name	Compiler	MPI	MATH
abiref_gnu_9.5_debug	GNU 9.5	OpenMPI 4.0.3	OpenBLAS 0.3.9

- checks build system with less used options (e.g. openmp, exports, cclock)
- checks doc in sources (ROBODoc)
- tests “make distcheck” (tarball creation)
- checks 15 abirules (rules for developers)
  (for ex: “Unused variable”, “Unused dummy argument”, "Nonstandard type declaration”, …)
- checks 10 buildsys (“check-build-examples”, “check-cpp-options”, “check-libpaw”, …)
- checks to find the broken links in docs with linkchecker

odonly				
Name	Compiler	MPI	MATH	misc
higgs_gnu_12.3_cov	GNU 12.3	MPICH 4.1.2	MKL2019/FFTW3	coverage + BigDFT
scope_gnu_10.2_s64	GNU 10.2	MPICH 3.3.2	OpenBLAS 0.3.10	tutoparal with np=64
buda2_gnu_8.5_cuda	GNU 8.5	Open MPI 3.1.4	mkl 2017/magma	

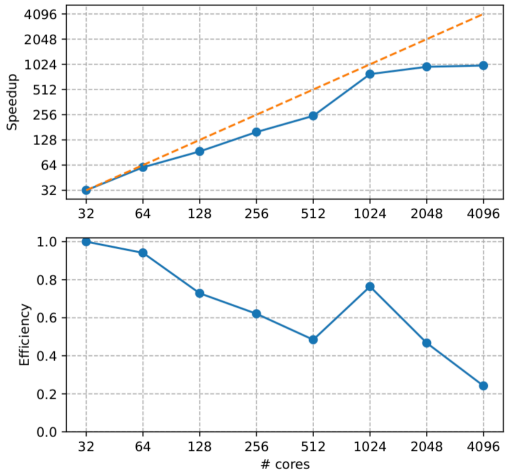
- GNU <= 7 is no longer supported

abinit-fallbacks/release-9.8

current versions	next versions*
[9.8] atompaw = 4.2.0.2 bigdft = abinit-1.7.1.30 hdf5 = 1.10.8 libpsml = 1.1.12 libxc = 6.0.0 lapack = abinit_6.10 netcdf4 = 4.9.0 netcdf4_fortran = 4.6.0 wannier90 = 3.1.0 xmlf90 = 1.5.6	**[10.0]** atompaw = 4.2.0.3 bigdft = abinit-1.7.1.30 hdf5 = 1.14.4-2 libpsml = 2.0.1 libxc = 6.2.2 lapack = abinit_6.10 netcdf4 = 4.9.2 netcdf4_fortran = 4.6.1 wannier90 = 3.1.0 xmlf90 = 1.6.3

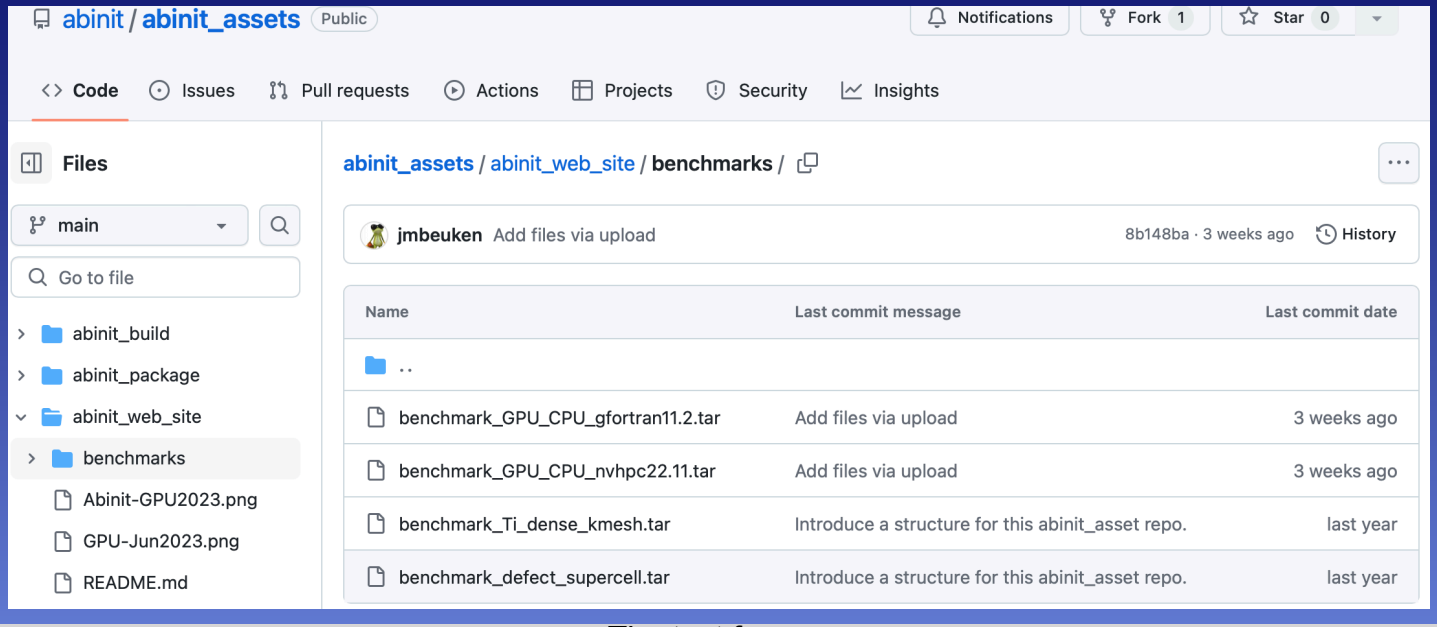
* tested with beuken/develop/4252a3da with eos_gnu_13.2_openmpi

https://www.abinit.org/benchmarks



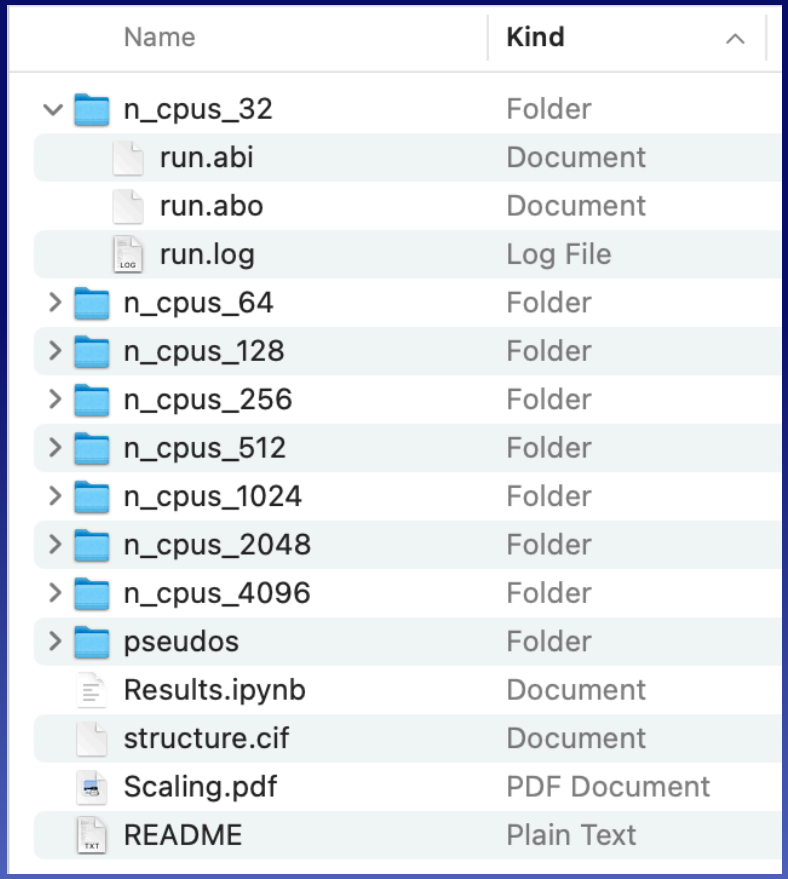
FEATURE TESTED:
Ground-state calculation - self-consistent DFT
SYSTEM TESTED:
Ti metal, 3 atoms/cell I 80x80x80 dense k-mesh | 20 bands
ABINIT VERSION: 9.8.1
COMPILATION OPTIONS:
Gnu 11.3 compiler with -g -02 and MPI
HARDWARE:
Lucia supercomputer AMD EPYC 7513 32C 2.6GHz with Mellanox HDR Infiniband
#245 HPC ranking
ADDITIONAL NOTE:
Until 512 cores only k-point parallelization is used while above there are additional parallelization levels on bands and fft, explaining the improved performance.
SOURCE: LINK

SOURCE : LINK -> https://github.com/abinit/abinit_assets/tree/main/abinit_web_site/benchmarks



abinit_assets/abinit_web_site/benchmarks/benchmark_Ti_dense_kmesh.tar

content of the tar file :



-> can rerun tests to compare

autotools and cmake

AUTOCONF

- improved the build system :
  - better detection of compiler : consequently, simplification and even, deletion of the ac9 file
    -> for examples, on lucia with cray compiler and on lemaitre4 with gnu
  - improved OpenMP management (MT)
  - better management of FCFLAGS, FCFLAGS_EXTRA and FCFLAGS_OPENMP (MT, MV, YP)
- Autoconf 2.69 -> update to 2.71 : remove "Obsolete Macros"
  (done on eos with beuken/develop branch)
  not yet in prod : need to update Autoconf on all builders

CMAKE

- https://docs.abinit.org/installation/#how-to-build-abinit-with-cmake

```
cd $(ABINIT_TOPLEVEL_SOURCE)
# step 1: cmake configure
cmake -S . -B _build/cmake
# step 2: cmake build using 6 threads
cmake --build _build/cmake -j 6
```

no success yet compiling with cmake