# A new LLVM toolchain for EasyBuild

Davide Grassano (@Crivella) CECAM Lausanne - Switzerland

# Outline

- Compilers
- The LLVM project
  - Philosophy
  - Project and runtimes
- Front-ends and commercial toolchains
- Overhaul of the LLVM EasyBlock
  - Challenges & New features
  - Maintainability (EG about changes needed between major versions like offload split in 18->19)
- Introducing an LLVM only toolchain
- Features comparison with GNU
- Where to go from here?



# Compilers - modern architecture

• IR: Any data structure that can represent the program without loss of information so that its execution can be conducted accurately





#### Compilers - GCC workflow





### LLVM - Brief history



"The LLVM Project is a collection of modular and reusable compiler and toolchain technologies."

- 2000 University of Illinois, Vikram Adve, Chris Lattner Research project
   o Low Level Virtual Machine (LLVM)
- 2003 First release of LLVM
- 2005 Apple, Chris Lattner
- 2006 Chris Lattner starts Clang (first release 2007)
- 2011 LLVM is not an acronym anymore but the official project name



# LLVM - Philosophy

- Decoupling (LLVM IR)
  - Completeness of the IR enables total decoupling of the front and back-end
- Modularization
  - Organized as a set of libraries
- Reusability
  - APIs allow to reuse/mix functionalities from several existing components by new tools





#### LLVM - Projects

- LLVM Core: Libraries for middle-end optimizer and back-end code generation based on LLVM IR
- Clang: C/C++ front-end compilers + collections of extra tool for static analysis and more
- LLDB: LLVM integrated debugger based on LLVM Core API/tooling
- libc++ / libc++abi: Implementation of the C++ std library (C++11, C++14, C++17, C++20, C++23, C++2c)
- **libc:** Implementation of the C standard library (C23 + POSIX.1-2024)
- **Compiler-rt**: Low level code generation support routines + RT dynamic testing and sanitizers
- MLIR: Multi-Level IR, improves reusability and compilation for heterogeneous hardware
- **OpenMP:** OpenMP runtime and code generation
- **polly:** Polyhedral optimizer for cache locality and automatic vectorization
- **libclc:** Implementation of OpenCL standard library
- LLD: New linker for LLVM
- BOLT: Post-link code layout optimizer
- FLANG: Fortran front end (<=2003, 2008, 2018, 2023)





Centre Européen de Calcul Atomique et

# LLVM - Flang - history





# LLVM - Workflow example - Flang + OpenMP

- Fortran parser
- MLIR dialects
  - HL-FIR
  - FIR
  - OpenMP
- LLVM IR + OpenMP code-generation









flang -v hw omp.f90 -fopenmp -fopenmp-targets=x86 64-unknown-linux-gnu

Compile the input file for the host to produce a bitcode file. Lower #pragma omp target declarations to offloading entries and create metadata to indicate which entries are on the device.

Compile the input file for the target device using the offloading entry metadata created by the host.

```
"/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/flang" \
   -triple x86 64-unknown-linux-gnu \
   -mrelocation-model pic \
   -pic-level 2 \
   -pic-is-pie 🔪
   -target-cpu x86-64 \
   -fopenmp \
   -resource-dir //home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20 \
   -fopenmp-targets=x86 64-unknown-linux-gnu \
   -mframe-pointer=all \
   -o/tmp/hw omp-eeb2ac.bc \
   -x f95-cpp-input hw omp.f90
"/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/flang" \
   -triple x86_64-unknown-linux-gnu \
   -emit-obj \
   -mrelocation-model pic \
   -pic-level 2 \
   -target-cpu x86-64 \
   -fopenmp \
   -resource-dir /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20 \
   -fopenmp-host-ir-file-path /tmp/hw omp-eeb2ac.bc \
   -fopenmp-is-target-device \
   -mframe-pointer=all \
   -o /tmp/hw omp-a75c3b.o \
   -x f95-cpp-input hw omp.f90
```



flang -v hw omp.f90 -fopenmp -fopenmp-targets=x86 64-unknown-linux-gnu

Link the OpenMP device runtime library and run the backend to create a device object file. Run the backend on the host bitcode file and create a fat object file using the device object file.

```
"/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/clang-offload-packager"
    -o/tmp/hw omp-9b7a32.out \
    --image=file=/tmp/hw omp-a75c3b.o,triple=x86 64-unknown-linux-gnu,arch=,kind=openmp
"/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/flang" \
   -fc1 \
    -triple x86 64-unknown-linux-gnu \
   -emit-obj \
    -mrelocation-model pic \
   -pic-level 2 \
   -pic-is-pie \
    -target-cpu x86-64 \
    -fopenmp \
    -resource-dir /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20 \
    -fembed-offload-object=/tmp/hw omp-9b7a32.out \
    -fopenmp-targets=x86_64-unknown-linux-gnu \
    -mframe-pointer=all \
    -0 /tmp/hw omp-fc8457.0 \
    -x ir /tmp/hw omp-eeb2ac.bc
```



flang -v hw omp.f90 -fopenmp -fopenmp-targets=x86 64-unknown-linux-gnu

Pass the fat object file to the linker wrapper tool and extract the device objects. Run the device linking action on the extracted objects.

```
/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/clang-linker-wrapper" \setminus
   --host-triple=x86 64-unknown-linux-gnu \
   -z relro \
  --hash-style=gnu \
   -m elf x86 64 \
   -pie \
   -dynamic-linker /lib64/ld-linux-x86-64.so.2 \
   -o a.out \
  /lib/x86 64-linux-gnu/Scrt1.o \
   /lib/x86 64-linux-gnu/crti.o \
   /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86 64-unknown-linux-gnu/clang rt.crtbegin.o \
  /tmp/hw omp-fc8457.o \
   -lm \
   -lomp \
   -L/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib \
   /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86 64-unknown-linux-gnu/libclang rt.builtins.a \
   --as-needed \
   -lunwind \
   --no-as-needed \
   -lpthread \
   -lc//home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86_64-unknown-linux-gnu/libclang_rt.builtins.a
   --as-needed \
  -lunwind \
   --no-as-needed \
  /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86 64-unknown-linux-gnu/clang rt.crtend.o //lib/x86 64-linux-gnu/crtn.o \
```



flang -v hw omp.f90 -fopenmp -fopenmp-targets=x86 64-unknown-linux-gnu

Wrap the device images and offload entries in a symbol that can be accessed by the host.

```
"/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/bin/clang" \
   -- no-default-config
   -o/tmp/a.out.x86 64.native-c3cfc2.img \
   --target=x86 64-unknown-linux-gnu \
   -march=native \
   -flto \
   -Wl.--no-undefined \
   /tmp/hw omp-fc8457-x86 64-unknown-linux-gnu--0872bl.o \
   -shared /lib64/ld-linux-x86-64.so.2 \
   -1 FortranRuntime \
   -1 FortranDecimal \
   -l omp \
   -l omptarget \
   -l omptarget.devicertl \
   -L /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib \
   /home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86 64-unknown-linux-gnu/libclang rt.builtins.a \
   -l unwind \
   -l pthread \
   -l c/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/clang/20/lib/x86 64-unknown-linux-gnu/libclang rt.builtins.a \
   -l unwind \
```



flang -v hw omp.f90 -fopenmp -fopenmp-targets=x86 64-unknown-linux-gnu

Add the wrapped binary to the linker input and run the host linking action. Link with libomptarget to register and execute the images.





# LLVM in EasyBuild - Old EasyBlocks

- LLVM (124 lines)
  - Build the llvm-core project only
- Clang (779 lines)
  - No support for shared builds
  - No support for GCCless build
  - Limited support for interplay between subprojects
  - Test-suite available but in general not ran in the respective EasyConfigs
  - Missing supports for new (>17.0.0) projects
  - Most likely non-correct RPATHing for runtimes



## LLVM in EasyBuild - NEW EasyBlock

#### • Fusing LLVM + Clang (1261 lines)

- Allows static/shared builds
- Allows for a `minimal` build for LLVM-core only
- Allow building on top of GCC or in FULL\_LLVM mode
- More detailed project configuration
- Ensure the test-suite is ran and allows for errors (regex or number based)
- More extensive banned\_libraries/sanity checking



### LLVM in EasyBuild - EasyBlock Challenges

• PROs of LLVM test-suite: Highly extensible and configurable



## LLVM in EasyBuild - EasyBlock Challenges

- PROs of LLVM test-suite: Highly extensible and configurable
- CONs of LLVM test-suite: Highly extensible and configurable
  - Fix with CMake parameters where possible
  - Patches to .cfg files where needed
- EG: The test-suite default configurations makes assumption on being able to find system libraries like libatomic



#### LLVM in EasyBuild - Test suite results

#### Host only

#### 

Testing Time: 1275.92s

otal Discovered	Test	ts: 139	007
Skipped		370	(0.27%)
Unsupported		38832	(27.94%)
Passed		99641	(71.68%)
Expectedly Fai	led:	157	(0.11%)
Timed Out			(0.00%)
Failed			(0.00%)

#### All targets

#### warning(s) in tests \*\*\*\*\* Timed Out Tests (2): libomptarget :: x86 64-unknown-linux-gnu :: mapping/declare mapper api.cpp libomptarget :: x86\_64-unknown-linux-gnu-LTO :: mapping/declare\_mapper\_api.cpp Failed Tests (12): BOLT :: X86/register-fragments-bolt-symbols.s BOLT-Unit :: Core/./CoreTests/failed to discover tests from gtest Clang :: CodeGen/PowerPC/ppc-emmintrin.c Clang :: CodeGen/PowerPC/ppc-mmintrin.c Clang :: CodeGen/PowerPC/ppc-pmmintrin.c Clang :: CodeGen/PowerPC/ppc-smmintrin.c Clang :: CodeGen/PowerPC/ppc-tmmintrin.c Clang :: CodeGen/PowerPC/ppc-xmmintrin.c Clang Tools :: modularize/ProblemsCoverage.modularize Flang :: Driver/atomic.f90 Flang :: Driver/gcc-toolchain-install-dir.f90 ompd-test :: api tests/test ompd get icv from scope.c

#### Testing Time: 1509.63s

Total Discover	ed Tests:	1392	222
Skipped			(0.02%)
Unsupported		5366	(3.85%)
Passed	: 133	3546	(95.92%
Expectedly Fa	ailed:		(0.19%)
Timed Out			(0.00%)
Failed		12	(0.01%)



# LLVM in EasyBuild - EasyBlock Challenges

- RPATHING (multi-stage build)
  - Easybuild creates wrappers scripts for every compiler that takes the content of LD\_LIBRARY\_PATH and transform its to a sequence of -WI,-rpath=XXX options





#### LLVM - Toolchain





#### Testing the toolchain: --try-toolchain

• Do we need to recreate every EC with the new toolchain?



#### Testing the toolchain: --try-toolchain

- Do we need to recreate every EC with the new toolchain?
- *eb* SOME\_PACKAGE.*eb* --*robot* --*try*-*toolchain*=TC\_NAME,TC\_VERS
- Attempt to install package and dependencies by
  - Replacing the package toolchain with the specified one
  - Replace the dependencies toolchain/subtoolchain with the equivalent one



#### Testing the tool

- Attempt to instal
  - 0
  - Ο

== Temporary log file in case of crash /home/crivella/.local/easybuild/logs/easybuild-kn8dgl8n.log Dry run: printing build status of easyconfigs and dependencies CFGS=/home/crivella/Documents/GIT/easybuild-easyconfigs/easybuild/easyconfigs \* [x] (no easyconfig file found) (module: M4/1.4.19--20.1.0)[x] (no easyconfig file found) (module: CMake/3.29.3c-20.1.0) [x] (no easyconfig file found) (module: pkgconf/2.2.0--20.1.0)[x] (no easyconfig file found) (module: Autotools/20231222--20.1.0) [X] (no easycontig file found) (module: Autotoous/2023) [X] (no easycontig file found) (module: 2t)(b)(3.1-14) [X] (no easycontig file found) (module: Python/3.12.3-[X] SCF65/m/M/M/4-1.4.9.ec (module: MA1-4.9) [X] (no easycontig file found) (module: hwloc/2.10.0-L] [X] (no easycontig file found) (module: hwloc/2.10.0-L] [X] (scf65/b/Bison/Bison-3.8.2.eb (module: Bison/3.8.2) [X] SCF65/b/Bison/Bison-3.8.2.eb (module: Bison/3.8.2) -20.1.0)-20.1.0)-20.1.0) -20.1.0)\* [x] \$CFGS/f/flex/flex-2.6.4.eb (module: flex/2.6.4) (no easyconfig file found) (module: libevent/2.1.12--20.1.0 (no easyconfig file found) (module: OpenBLAS/0.3.27-L -20.1.0)\* [x] \$CFGS/z/zlib/zlib-1.3.1.eb (module: zlib/1.3.1) \* [x] \$CFGS/b/binutils/binutils-2.42.eb (module: binutils/2.42) \* [x] \$CFGS/g/GCCcore/GCCcore-13.3.0.eb (module: GCCcore/13.3.0) eb SOME\_PACK \* (x) 6(r0 easyconfig file found) (module: 1)/1/8.1.8-6(ccore-13.3.0) \* (x) 5(r65/z/z/1)/10/11/8.1.8-6(ccore-13.3.0) \* (x) 5(r65/z/z/1)/11/11/11/8.1.8-6(ccore-13.3.0) eb (module: 1)/11/11/11/8.1.8-6(ccore-13.3.0) \$CFG5/z/zlib/2lib-1.3.1-GCCcore-13.3.0.eb (module: zlib/1.3.1-GCCcore-13.3.0)
(no easyconfig file found) (module: libfabric/1.21.0-LLVMtc-20.1.0) \* [x] \$CFGS/h/help2man/help2man-1.49.3-GCCcore-13.3.0.eb (module: help2man/1.49.3-GCCcore-13.3.0) 
 Perption
 Strip
 Strip<</th>
 Strip<</th>
 Strip<</th>
 \* [x] \$CFGS/m/M4/M4-1.4.19-GCCcore-13.3.0.eb (module: M4/1.4.19-GCCcore-13.3.0) [x] (no easyconfig file found) (module: PRRTE/3.0.5--20.1.0) \* [x] \$CFGS/e/expat/expat-2.6.2-GCCcore-13.3.0.eb (module: expat/2.6.2-GCCcore-13.3.0) \* [x] (no easyconfig file found) (module: UCC/1.3.0--20.1.0)\* [x] \$CFGS/t/Tcl/Tcl-8.6.14-GCCcore-13.3.0.eb (module: Tcl/8.6.14-GCCcore-13.3.0) \* [x] \$CFGS/p/pkgconf/pkgconf-1.8.0.eb (module: pkgconf/1.8.0) \* [x] \$CFGS/p/Perl/Perl-5.38.2-GCCcore-13.3.0.eb (module: Perl/5.38.2-GCCcore-13.3.0) \* [x] \$CFGS/s/SQLite/SQLite-3.45.3-GCCcore-13.3.0.eb (module: SQLite/3.45.3-GCCcore-13.3.0) \* [x] \$CFGS/a/Autoconf/Autoconf-2.72-GCCcore-13.3.0.eb (module: Autoconf/2.72-GCCcore-13.3.0) [x] \$CFGS/a/Automake/Automake-1.16.5-GCCcore-13.3.0.eb (module: Automake/1.16.5-GCCcore-13.3.0) \* [x] \$CFGS/l/libffi/libffi-3.4.5-GCCcore-13.3.0.eb (module: libffi/3.4.5-GCCcore-13.3.0) \$CFGS/l/libtool/libtool-2.4.7-GCCcore-13.3.0.eb (module: libtool/2.4.7-GCCcore-13.3.0) \$CFGS/g/gettext/gettext-0.22.5.eb (module: gettext/0.22.5) \$CFGS/a/Autotools/Autotools-20231222-GCCcore-13.3.0.eb (module: Autotools/20231222-GCCcore-13.3.0) [x] \$CFGS/x/XZ/XZ-5.4.5-GCCcore-13.3.0.eb (module: XZ/5.4.5-GCCcore-13.3.0) \* [x] \$CFGS/l/libxml2/libxml2-2.12.7-GCCcore-13.3.0.eb (module: libxml2/2.12.7-GCCcore-13.3.0) \* [x] \$CFGS/l/libiconv/libiconv-1.17-GCCcore-13.3.0.eb (module: libiconv/1.17-GCCcore-13.3.0) \* [x] \$CFG5/g/gettext/gettext-0.22.5-6CCcore-13.3.0.eb (module: gettext/0.22.5-6CCcore-13.3.0)
\* [x] \$CFG5/z/zlib/zlib-1.2.13.eb (module: zlib/1.2.13) \* [x] \$CFGS/b/binutils/binutils-2.40.eb (module: binutils/2.40) \* [x] \$CFGS/p/Perl/Perl-5.38.0.eb (module: Perl/5.38.0) \* [x] \$CFGS/o/OpenSSL/OpenSSL-3.eb (module: OpenSSL/3) \* [X] \$CFGS/p/Python/Python-3.12.3-GCCcore-13.3.0.eb (module: Python/3.12.3-GCCcore-13.3.0) \* [X] \$CFGS/c/cURL/cURL-8.7.1-GCCcore-13.3.0.eb (module: cURL/8.7.1-GCCcore-13.3.0) [x] \$CFGS/l/libarchive/libarchive-3.7.4-GCCcore-13.3.0.eb (module: libarchive/3.7.4-GCCcore-13.3.0) [x] \$CFGS/c/CMake/CMake-3.29.3-GCCcore-13.3.0.eb (module: CMake/3.29.3-GCCcore-13.3.0) (x) \$CFGS/p/psutil/psutil-6.0.0-GCCcore-13.3.0.eb (module: psutil/6.0.0-GCCcore-13.3.0) \* [x] \$CFGS/g/git/git-2.45.1-GCCcore-13.3.0.eb (module: git/2.45.1-GCCcore-13.3.0) \* [x] \$CFGS/1/LLVM/LLVM-20.1.0.eb (module: LLVM/20.1.0) \* [x] \$CFGS/1/L c-20.1.0.eb (module: (20.1.0)\* [x] \$CFGS/f/FlexiBLAS/FlexiBLAS-3.4.4--20.1.0.eb (module: FlexiBLAS/3.4.4-L LVMtc-20.1.0) \* [x] \$CFGS/f/FFTW/FFTW-3.3.10c-20.1.0.eb (module: FFTW/3.3.10-LLVM c-20.1.0) \* [x] \$CFGS/b/binutils/binutils-2.42c-20.1.0.eb (module: binutils/2.42--20.1.0)-20.1.0.eb (module: UCX/1.16.0-\* [x] \$CFGS/u/UCX/UCX-1.16.0c-20.1.0) \* [x] \$CFGS/o/OpenMPI/OpenMPI-5.0.7c-20.1.0.eb (module: OpenMPI/5.0.7-L -20.1.0) \* [x] \$CFGS/l/ -2025a.eb (module: 1/2025a) \* [x] \$CFGS/f/FFTW.MPI/FFTW.MPI-3.3.10--2025a.eb (module: FFTW.MPI/3.3.10-lo -2025a) \$CFGS/s/ScaLAPACK/ScaLAPACK-2.2.0--2025a-fb.eb (module: ScaLAPACK/2.2.0--2025a-fb) \* [X] oss/lfoss-2025a.eb (module: \$CFGS/1/L1 /2025a) [x] \$CFGS/q/QuantumESPRESS0/QuantumESPRESS0-7.4-1 -2025a.eb (module: QuantumESPRESSO/7.4-1 -2025a Temporary log file(s) /home/crivella/.local/easybuild/logs/easybuild-kn8dgl8n.log\* have been removed. Temporary directory /tmp/eb-kslvte7b has been removed.

#### NAME.TC VERS

#### ivalent one

- Add the TC definition in framework (LLVMtc, lfbf, lolf, lompi, lfoss)
- Start by creating LLVMtc
  - Need an *EMPTY* compatibility **binutils** package
- zlib
  - **Id.IId** does not enable version scripts be default (needs -WI,--undefined-version)
- FFTW
  - Enabling quad precision in the config file is tied on the compiler description being GCC >4.6
  - As LLVM is not recognized trying to build with quad precision will throw an error
  - NEW EC file with quad\_precision disabled (anyway disabled by default for the MPI version)



- UCX
  - Compatibility problem with v<1.17.0 (apply patch or use newer version)
- libpciaccess
  - Patch -Werror out of options or pass -Wno-int-conversion
- HDF5 (WIP)
  - Configure script runs **flang** and uses the output to determine input options for CFLAGS
    - *-target-feature -lwp* ends up being interpreted as a non-existing library by clang
    - This is emitted only in case of **-march=native** (temporary fix disable this)
  - Weird one: double quotes in the compiler version string (extracted from **mpicc+clang**) gets injected into a template file for h5cc without being escaped



- ScaLAPACK:
  - Needs -std=c89 to avoid an undeclared function error
- ELPA
  - Another weird one (!precompiler)



- ScaLAPACK:
  - Needs -std=c89 to avoid an undeclared function error
- ELPA
  - Another weird one (!precompiler)

```
if (wantDebug) write(error_unit,*) "elpa_cholesky_&
&MATH_DATATYPE&
#if REALCASE == 1
   &: Error in dpotrf: ",info
#endif
#if COMPLEXCASE == 1
   &: Error in zpotrf: ",info
#endif
success = .false. ! "
```



#### QuantumESPRESSO

- Remove HDF5 and ELPA deps as not ready yet
- Remove qmcpack plugin that requires HDF5



#### QuantumESPRESSO

- Remove HDF5 and ELPA deps as not ready yet
- Remove qmcpack plugin that requires HDF5
- Error while building base libraries

/home/crivella/.local/easybuild/build/QuantumESPRESS0/7.4/lfoss-2024a/qe-7.4/UtilXlib/mp.f90:2341:30: error: Element of assumed-shape array may not be associated with a dummy argument 'sendbuf=' array CALL MPI\_GATHERV( alldata(1,displs(myid+1)+1), recvcount(myid+1), my\_column\_type, &



#### QuantumESPRESSO

- Remove HDF5 and ELPA deps as not ready yet
- Remove qmcpack plugin that requires HDF5
- Error while building base libraries -> Bump OpenMPI version and rebuild dependent software
- Error while building d3q plugin (missing **flang** feature)

error: loc("/home/crivella/.local/easybuild/build/QuantumESPRESS0/7.4/lfoss-2025a/qe-7.4/external/d3q/src/d3\_shuffle.f90":190:1): /home/crivella/.local/easybuild/build/LLVM/20.1.0-rc1/systemsystem/llvm-project-20.1.0-rc1.src/flang/lib/Lower/CallInterface.cpp:1115: not yet implemented: VOLATILE in procedure interface LLVM ERROR: aborting



#### QuantumESPRESSO

0	Remove HE 945	554 pw_workflow_relax_relax	= 653.46 sec*proc (3 tests)
	945	555 pw_workflow_scf_dos	= 45.56 sec*proc (3 tests)
0	Remove gr <sup>945</sup>	556 pw_workflow_vc-relax_dos	= 74.55 sec*proc (3 tests)
	945	557 pw_workflow_vc-relax_scf	= 33.55 sec*proc (3 tests)
0	Error while 194	558 systemcp	= 1548.79 sec*proc (45 tests)
	945	559 systempw	= 8591.45 sec*proc (291 tests)
0	Error while <sup>94</sup>	ofo unit	= 57.31 sec*proc (40 tests)
	945		
0	Success <sup>94</sup>	bb2 lotal lest time (real) = 3	380.55 Sec
	943		average and the second and and and and and and and and and a
	941	$504 = 2025 \cdot 05 \cdot 14 \cdot 14 \cdot 52 \cdot 21 \cdot 040$	) qualitumespresso, py:400 into total tests passed s/o out of 5/0 (100.00%)
	941	$505 = 2025 \cdot 05 \cdot 14 \cdot 14 \cdot 52 \cdot 21 \cdot 040$	build_tog.py:322 INTO (took 30 mins 20 Sets)
	54.	False skin test sten: Fal	se ski sanity check False)
	9.45	$567 = 2025 - 03 - 14 - 14 \cdot 32 \cdot 21 - 041$	build log py-22 TNF0 installing
	945	$568 = 2025 \cdot 03 \cdot 14 \cdot 14 \cdot 32 \cdot 21 \cdot 041$	easylock pv:2228 TNF0 Starting install sten
	945	$569 = 2025 \cdot 03 \cdot 14 \cdot 14 \cdot 32 \cdot 21 \cdot 041$	easy conting villa 2 INFO Generating template values
	945	570 = 2025 - 03 - 14 + 14 : 32 : 21 . 041	templates.pv:297 DEBUG config: QuantumESPRESSO EasvConfig @ /home/crivella/Documents/GIT/easvb
		4-lfoss-2025a.eb	
	0.41	71 2025 02 14 14 22 21 042	templeter au 222 DEDUC version found in accuration in 7.4

94571 == 2025-03-14 14:32:21,042 templates.py:333 DEBUG version found in easyconfig is 7.4







## Testing the toolchain: CP2K

- BOOST: x86\_64-pc-linux vs x86\_64-unkown-linux-gnu target triple
- Libinit: replace cstd with extra\_cxxflags to pass --std=c++11 also to clang++
- Catch2: Needs version >2 (used 3.8)
- Rust (WIP): clash with symbols of internal LLVM compilation
  - Possible solution: Link against already build LLVM
  - Use Rust version made to be compatible with LLVM 20.x
- SciPy-Bundle:
  - Punch the MESON build system around to use an LLVM based compiler and manually remove non-existing **flang** flags
- Libxsmm: No configure (remove/add flags with patches to the Makefile)



#### Testing the toolchain: CP2K

- Test version 2024.3
- With LLVM 20.x: disable SciPy based dependencies due to skipping Rust
- Stricter checks on OMP sections with default none
  - Patch to ensure all variable appear as either private/shared
- Error due to missing feature

error: loc("/home/crivella/.local/easybuild/build/CP2K/2024.3/lfoss-2025a/cp2k-2024.3/obj/Linux-x86-64lfoss/psmp/transport.F90":419:7): /home/crivella/.local/easybuild/build/LLVM/20.1.0/system-system/llvmproject-20.1.0.src/flang/lib/Optimizer/CodeGen/TargetRewrite.cpp:544: not yet implemented: passing argument or result on the stack in indirect calls



## FLANG: Missing feature

- Grepping through the flang source code
  - 339 missing features (as of 20.1.0 and 20.1.1)



(loc, "polymorphic vector subscripts"); converter.getCurrentLocation(), include/flang/Optimizer/Support/Utils.h: TODO(loc, include/flang/Optimizer/Support/Utils.h: TODO(loc, include/flang/Optimizer/Builder/DirectivesCommon.h: lib/Lower/Bridge.cpp: TODO(loc, "generate f lib/Lower/Bridge.cpp: TODO(loc, "create p loc, "VOLATILE in procedure interface"); loc, "coarray: dummy argument coarray in procedure interface"); loc, "assumed shape dummy argument with VALUE attribute"); TODO(loc, "derived type"); interface.converter.getCurrentLocation(), (loc, "generate fir.dt component for length parametrized derived " loc, (loc, "create polymorphic host associated copy"); 'ODO(loc, "array component shape depending on length parameters"); (loc, "get length parameters from derived type BoxValue"); getLoc(), "lowering symbol to HLFIR"); getLoc(), "coarray: lowering a reference to a coarray object"); toLocation(), "coarray: ChangeTeamConstruct"); getLoc(), "coarray: lowering a reference to a coarray object"); toLocation(), "coarray: ChangeTeamStmt"); loc, "StaticDataObject::Pointer substring with kind > 1"); (toLocation(), "coarray: EndChangeTeamStmt"); getLoc(), "Designate derived type with length parameters in HLFIR"); toLocation(), "coarray: CriticalConstruct"); loc, toLocation(), "coarray: CriticalStmt"); O(loc, "compute character length of automatic character component " qetLoc(), "BOZ"); toLocation(), "coarray: EndCriticalStmt"); getLoc(), "lowering type parameter inquiry to HLFIR"); (loc, "RANK(\*) with parameterized derived type selector"); loc, "stride inquiry"); loc, "RANK(\*) with polymorphic selector"); lib/Lower/ConvertExprToHLFIR.cpp: lib/Lower/OpenACC.cpp: T0D0(l lib/Lower/OpenACC.cpp: T0D0(l lib/Lower/OpenACC.cpp: T0D0(loc lib/Lower/OpenACC.cpp: T0D0(loc) lib/Lower/OpenACC.cp) T0D0(loc) lib/Lower/OpenACC.c loc, "structure constructor for derived type with length parameters " loc, "procedure pointer assignment"); loc, "automatic character component in structure constructor"); (loc, "polymorphic pointer assignment in FORALL"); loc, "Unsupported boxed type in OpenACC privatization"); loc, "polymorphic pointer assignment in FORALL"); loc, "Unsupported boxed type in OpenACC firstprivate"); loc, "Device resident allocatable derived-type component"); loc, "Unsupported boxed type for reduction"); loc, "reduction add type"); loc, "assignment to a whole allocatable inside FORALL"); loc, "reduction mul type"); loc, "pointer assignment inside FORALL"); loc, "reduction operator"); loc, "pointer assignment inside FORALL"); loc, "Unsupported boxed type in OpenACC reduction"); loc, "assignment to pointer result of function reference"); operandLocation, "reduction with unsupported type"); loc, "derived-type finalization with array assignment"); currentLocation, "DO CONCURRENT with locality spec"); clauseLocation, "OpenACC collapse force modifier"); (loc, "assignment to derived type allocatable with " operandLocation, "OpenACC Global Ctor from parser::Name"); (loc, "sequence association of length parameterized derived type " (clauseLocation, "clause on declare directive"); (loc, "derived type argument passed by value"); loc, "coarray: allocation of a coarray object"); loc, "setting derived type params in allocation"); callContext.loc, loc, "derived type length parameters in allocate"); loc, "coarray: allocation of a coarray object"); loc, "component with length parameters in structure constructor"); loc, "deferred length type parameters."); loc, "parent component in structure constructor"); loc, "path to a POINTER, ALLOCATABLE, or other component that requires " loc, "Creation of very large array constants"); loc, "copy derived type with length parameters"); (loc, "initial-data-target with derived type length parameters"); qetLoc(), "BOZ"); (loc, "initial-data-target with derived type length parameters"); loc, "component with length parameters in structure constructor"); loc, loc, "polymorphic component in derived type assignment"); loc, "allocatable components in derived type assignment"); loc, "procedure pointer globals"); loc, "procedure pointer component in derived type assignment"); loc, "global"); // Something else loc, "rank inquiry on assumed rank"); genLocation(converter, sym), loc, "stride inquiry"); loc, "assumed-rank variable in procedure implemented in Fortran"); getLoc(), "type parameter inquiry"); loc, getLoc(), "character array concatenate"); (loc, "support for assumed rank entities"); getLoc(), loc, "CHARACTER MIN and MAX with dynamically optional arguments" lib/Lower/ConvertExpr.cpp: getLoc(), "non explicit semantics::Bound implementation"); loc, "creating temporary for derived type with length parameters"); converter.getCurrentLocation(), "coarray: CHANGE TEAM construct"); loc, "polymorphic array temporary"); (converter.getCurrentLocation(), "coarray: CHANGE TEAM statement"); loc, "creating polymorphic temporary"); converter.getCurrentLocation(), "coarray: END CHANGE TEAM statement"); loc, "parametrized derived type optional scalar argument copy-in"); (converter.getCurrentLocation(), "coarray: FORM TEAM statement"); (loc, "assumed type actual argument"); [loc, "passing a polymorphic entity to an OPTIONAL " converter.getCurrentLocation(), "co-array address"); loc, "passing to an OPTIONAL CONTIGUOUS derived type argument " converter.getCurrentLocation(), "co-array value"); loc, "passing an assumed rank entity to an OPTIONAL " (loc, "defined IO procedure pointers"); loc, "pass by value in non elemental function call"); loc, "character descriptor to contiguous buffer"); getLoc(), (loc, "DISPOSE not part of the runtime::io interface"); loc, "assignment to whole allocatable array inside FORALL"); loc, "io-control-spec contains a reference to a non-integer, " loc, "gather rhs LEN parameters in assignment to allocatable");

lib/Lower/OpenMP/OpenMP.cpp: TOD0(converter. TOD0(convert converter.getCurrentLocation(), "OpenMPDeclareMapperConstruct"); loc, "CHARACTER does not have constant LEN"); converter.getCurrentLocation(), "OpenMPCancelConstruct"); loc, "user-defined elemental assigment from expression with rank"); converter.getCurrentLocation(), "OpenMPCancelConstruct"); getLoc(), "coarray: reference to a coarray in an expression"); converter.getCurrentLocation(), "OpenMPDepobjConstruct"); (getLoc(), converter.getCurrentLocation(), "OpenMPAllocatorsConstruct"); (loc, "lhs of pointer assignment returned unexpected value"); loc, "OpenMP atomic compare"); loc, "invalid box conversion in elemental computation"); (clauseLocation, name + " clause is not implemented yet"); (loc, "TARGET of pointer assignment with runtime size/shape"); converter.getCurrentLocation(), "OpenMPUtilityConstruct"); loc. "array (as element) assignment"); converter.getCurrentLocation(), "OpenMPDispatchConstruct"); getLoc(), "getting shape from polymorphic array in elemental " (converter.getCurrentLocation(), "OpenMPExecutableAllocate"); loc, "polymorphic array temporary"); (converter.getCurrentLocation(), (loc, "character array expression temp with dynamic length"); (loc, loc, "derived type array expression temp with LEN parameters"); (loc, "Unsupported boxed type for reduction/privatization"); loc, "creating an array temp where the element type has " TODO(loc, loc, "non trivial optional elemental intrinsic array " currentLocation, loc, "intrinsic function with inquired argument"); (clauseLocation, "Triplet indexing in map clause is unsupported"); loc, "assumed type actual argument"); TODO(loc, "lastprivate clause with CONDITIONAL modifier"); (loc, location, "OMPD target data MapOperand BoxType"); getLoc(), "procedure designator"); currentLocation, "OmpAllocateClause ALIGN modifier"): loc, "parentheses on argument in elemental call"); currentLocation, "DEPOBJ dependence-type"); loc, "concat on unexpected extended values"); currentLocation, "Do Concurrent in Worksharing loop construct"); getLoc(), "expected vector to have an extent"); (clauseLocation, "OMPD\_target Device Modifier Ancestor"); (loc, "shape must be recovered from box"); converter.getCurrentLocation(), loc, "shape must be recovered from box"); converter.getCurrentLocation(), loc, "optional arguments in user defined elemental procedures"); (converter.getCurrentLocation(), getLoc(), "substring of static object inside FORALL"); (converter.getCurrentLocation(), (loc, "dynamic sized type"); (currentLocation, "Map type modifiers (other than 'ALWAYS')" loc, "deep copy on allocatable members"); currentLocation, (loc, "PDT size"); currentLocation, (loc, "PDT offset"); clauseLocation, "PRESENT modifier is not supported yet"); loc, "unhandled array constructor expression"); clauseLocation, "Mapper modifier is not supported yet"); getLoc(), "array expr type parameter inquiry"); clauseLocation, "Iterator modifier is not supported yet"); getLoc(), "array expr descriptor inquiry"); (loc, "passing type bound procedure (extension)"); getLoc(), "structure constructor"); (loc, "Procedure pointer with intrinsic target."); loc, "other component type"); loc, "reference to parent component"); (loc, "array (as element) assignment"); (loc, "threading length parameters in field index op"); (loc, "assignment in a FORALL involving a designator with a " loc, "coarray: reference to coarray object with vector subscript in " (loc, "assignment in a FORALL involving a designator with a " O(loc, "host associated derived type allocatable or pointer with " (loc, "need to adjust typeparameter(s) to reflect the final " converter.genLocation(sym.name()), loc, "assignment of derived type"); converter.getCurrentLocation(), "coarray: NOTIFY WAIT runtime"); getLoc(), "coarray: reference to a coarray in an expression"); converter.getCurrentLocation(), "coarray: EVENT POST runtime"); loc, "non elemental user defined array assignment inside FORALL"); converter.getCurrentLocation(), "coarray: EVENT WAIT runtime"); Lib/Lower/NLfirIntrinsics.cpp: TOD Lib/Lower/NDpenMP/ReductionProcessor.cpp: Lib/Lower/OpenMP/ReductionProcessor.cpp: Lib/Lower/OpenMP/OpenMP.cpp: TODO(loc, " Dib/Lower/OpenMP/OpenMP.cpp: TODO(loc loc, "hlfir transformational intrinsic dynamically optional " converter.getCurrentLocation(), "coarray: LOCK runtime"); loc, "Reduction of some types is not supported"); converter.getCurrentLocation(), "coarray: UNLOCK runtime"); loc, "Reduction of some identifier types is not supported"); converter.getCurrentLocation(), "coarray: SYNC ALL runtime"); loc, "Reduction of some intrinsic operators is not supported"); converter.getCurrentLocation(), "coarray: SYNC IMAGES runtime"); (loc, "Unsupported boxed type in OpenMP reduction"); converter.getCurrentLocation(), "coarray: SYNC MEMORY runtime"); loc, "OpenMP genCombiner for unsupported reduction variable type"); converter.getCurrentLocation(), "coarray: SYNC TEAM runtime"); (currentLocation, "Reduction modifiers are not supported"); (converter.getCurrentLocation(), "assumed rank expression types"); currentLocation, (converter.genLocation(component.name()), currentLocation, "Unsupported intrinsic proc reduction"); converter.genLocation(component.name()), "procedure components"); currentLocation, "Unexpected reduction type"); (loc, "parameterized derived types"); loc, "Reduction of some intrinsic operators is not supported"); (converter.getCurrentLocation(), "derived type length parameters"); (converter.getCurrentLocation(), (loc, "copy element of dynamic size"); converter.getCurrentLocation(), loc, "unhandled dynamic type parameters"); loc, "OMPD ordered"); loc, "fir.rebox assumed rank codegen with fir.ref<fi loc, "Scope construct"); (loc, "registration of managed variables"); loc, "Taskloop construct"); global->getLoc(), "support for procedure pointers"); loc, "Composite TASKLOOP SIMD"); (loc, "hlfir::designate load of pointer or allocatable"); (loc, "Unhandled directive " + llvm::omp::getOpenMPDirectiveName(dir)); loc, "hlfir.designate with a parametrized derived type loc, "Unhandled loop directive (" + loc. converter.getCurrentLocation(), "OpenMPDeclarativeAllocate"); (bufferizedExpr.getLoc(), "general extract storage case"); converter.getCurrentLocation(), "OpenMPDeclareReductionConstruct"); anal autract cta

loc, "polymorphic type with length parameters in HLFIR";<u>lib/Optimizer/CodeGen/Co</u>g loc, "derived type with type parameters"); en.cpp: loc, "codegen of optimized chained concatenation of more loc, "compute size of derived type with type parameters"); loc, "finalizing polymorphic temporary in HLFIR"); (loc, "need a Fortran entity to create a box"); loc, loc, "hlfir.expr conversion"); forallOp.getLoc(), "FORALL construct or silib/Optimizer/CodeGen/Cod loc, "fir.array nested inside other array and/or derived type"); shape.getLoc(), "read fir.shape to get extents"); loc, "unsupported combination of coordinate operands"); shape.getLoc(), "read fir.shape to get lower bounds"); firEnd.getLoc(), "fir.end codegen"); loc, "bounds of expressions in hlfir"); loc, "get extent from HLFIR expr without producer holding the shape"); caseOp.getLoc(), "fir.select case codegen with character type"); loc, "inquire PDTs length parameters of hlfir.expr"); unboxproc.getLoc(), "fir.unboxproc codegen"); loc, "inquire PDTs length parameters in HLFIR"); loc, "create temporary for assumed rank polymorphic"); loc, "multiple results not supported yet"); loc, "createStackTempFromMold for polymorphic type"); loc, "ABI of fir.dispatch with character arguments"); loc, "intrinsic module procedure: " + llvm::Twine(name)); lib/Optimizer/CodeGen/TargetRewrite.cpp: "coarray: intrinsic " + llvm::Twine(name)); loc, loc, loc. "intrinsic: " + llvm::Twine(name.upper())); loc, "multiple results not supported yet"); "cannot outline call to intrinsic " + llvm::Twine(name) + loc, loc, "cannot outline call to intrinsic " + llvm::Twine(name) + loc, "COMPLEX(KIND=2): for " + context + " type"); loc, "get length parameters of derived type"); lib/Optimizer/CodeGen/Target.cpp: lib/Optimizer/CodeGen/Target.cpp: "COMPLEX(KIND=3): " + context + " type"); loc, loc, "intrinsic module procedure: " + func); "COMPLEX(KIND=10): " + context + " type"); loc, "intrinsic: len trim for character array"); loc. loc, "Distinguigh dummy procedure arguments"); loc. "complex for this precision for " + context + " type"); loc. "intrinsic: reshape requires computing rank of result"); "intrinsic: min and max for CHARACTER"); "passing VALUE BIND(C) derived type for this target"); loc, loc, loc, "box value is missing type parameters"); lib/Optimizer/CodeGen/Target.cpp: lib/Optimizer/CodeGen/Target.cpp loc, "returning BIND(C) derived type for this target"); loc, "mutable box value is missing type parameters"); (loc, "passing vector argument to C by value"); loc, "character compare from descriptors"); loc, "updating mutablebox of derived type with length parameters"); (loc, "unsupported component type for BIND(C), VALUE derived " loc, "read allocatable or pointer derived type LEN parameters"); loc, "passing struct as a real > 128 bits in register"); loc, "update allocatable derived type length parameters"); loc, "creating unallocated fir.box of derived type with length " loc, "passing vector argument to C by value is not supported"); loc, "pointer assignment to derived with length parameters"); loc, "unsupported component type for BIND(C), VALUE derived " loc, "procedure pointer assignment"); lib/Optimizer/CodeGen/Target.cpp loc, "pointer assignment to derived with length parameters"); "handle complex argument types"); loc. loc, "procedure pointer assignment"); loc, "reallocation of derived type entities with length parameters").lib/Optimizer/CodeGen/Target.cpp: loc, "handle complex return types"); loc, "automatic allocation of derived type allocatable with loc, "handle complex argument types"); loc, lib/Optimizer/CodeGen/Target.cpp: lib/Optimizer/CodeGen/Target.cpp: loc, "handle complex return types"); loc, "read fir.box with length parameters"); loc, "generate code to get LEN type parameters"); loc, "unsupported complex type(not IEEEsingle, IEEEdouble, " loc, "derived type with type parameters"); (loc, "unsupported dynamic extent sequence type as a structure " loc, "allocatable and pointer components non deferred length " loc, "array component shape depending on length parameters"); loc, "unsupported vector width(must be 128 bits)"); loc, "get character component length from length type parameters"); lib/Optimizer/CodeGen/Target.cpp: lib/Optimizer/CodeGen/Target.cpp (loc, "unsupported component type for BIND(C), VALUE derived " loc, loc, "get length parameters from derived type BoxValue"); loc, loc, "LEN of character must be computed at runtime"); (loc, "floatType with width exceeding 128 bits is unsupported"); loc, "concatenating non contiguous character array into a scalar"); loc, "character array assignment"); loc, "unsupported component type for BIND(C), VALUE derived " loc, "creating inlined temporary stack for derived types"); loc, "unsupported empty struct type for BIND(C), ' embox.getLoc(), "embox conversion for fir.class type"); op.getLoc(), "did not find allocation function"); (mlir::UnknownLoc::get(ctx), "target not implemented"); boxprochost.getLoc(), "fir.boxproc host codegen"); (loc, "computing size of a component"); globalLen.getLoc(), "fir.global len codegen"); lenp.getLoc(), "fir.len\_param\_index codegen"); crivella@crivella-desktop:~/Documents/GIT/llvm-project/flang [(llvmorg-20.1.0) ? (6)] loc, "fir.allocmem codegen of derived type with length parameters"); \$ grep -R -E "^ \*TODO\(" \* | wc -l xbox.getLoc(), embox.getLoc(), 339 (loc, "generate call to calculate size of PDT"); crivella@crivella-desktop:~/Documents/GIT/llvm-project/flang [(llvmorg-20.1.0) ? (6)] loc, "fir.embox codegen of derived with length parameters");

\$

(loc, "reboxing descriptor of derived type with length parameters");

emboxproc.getLoc(), "fir.emboxproc codegen");

#### Feature comparison with GNU - language standard

C++

=	$\frown$	D	ГD	Λ	NL
	U			A	N

	LLVM	GNU
c++11	DONE	DONE
c++14	DONE	DONE
c++17	ALMOST	ALMOST
c++20	ALMOST	EXPERIMEN TAL
c++23	EXPERIME NTAL	EXPERIMEN TAL
c++2c	WIP	EXPERIMEN TAL

	LLVM	GNU
1995	DONE	DONE
2003	~DONE	DONE
2008	ALMOST	ALMOST
2018	EXPERIME NTAL	EXPERIMEN TAL
2023	WIP	WIP



#### Feature comparison with GNU - targets

- LLVM (20.1.0)
  - o aarch32/64 (82 CPUs, 308 feat)
    - upto v9.6
  - o amdgcn (73 GPUs, 209 feat)
    - upto gfx1201
  - avr (315 CPUs, 37 feat)
  - bpf (6 CPUs, 3 feat)
  - loongarch (6 CPUs, 20 feat)
  - mips (19 CPUs, 53 feat)
  - ppc (39 CPUs, 84 feat)
  - o r600 (19 GPUs, 16 feat)
  - o risc-v32/64 (39 CPUs, 276 feat)
  - sparc (40 CPUs, 53 feat)
  - o spirv32/64
  - wasm32/64 (4 CPUs, 17 feat)
  - x86/86\_64 (129 CPUs, 196 feat)
    - upto diamondrapids (AMX-AVX512)
    - upto znver5
  - xcore

- GCC (14.2)
  - o aarch32/64 (84 CPUs)
    - upto v9.5 (more for 15.0)
  - amdgcn (26 GPUs)
    - upto gfx1153 (more for 15.0)
  - loongarch (5 CPUs)
  - mips (92 CPUs)
  - ppc (55 CPUs)
  - risc-v32/64 (18 CPUs)
  - x86/86\_64 (112 CPUs)
    - upto diamondrapids (AMX-AVX512)
    - upto znver5



#### Where to go from here? EasyBlock - missing projects

- DONE
  - o bolt
  - $\circ$  clang
  - clang-extra-tools
  - compiler-rt
  - flang
  - Ild
  - Ildb
  - o Ilvm-core
  - openmp
  - offload
  - libcxx
  - libcxxabi
  - $\circ$  libunwind
  - mlir
  - polly

- Missing/untested
  - libc
  - libclc
  - Ilvm-libgcc
  - o pstl
  - SPIR-V (external)



#### Where to go from here? EasyBlock - maintainability

- More testing
  - More systems (especially with GPU and offloading)
  - Cross-compilation
  - More project and parameters combination
- Smarter version control list of enabled targets and devices
- Hope the CMake variables and project structure keeps stable



#### Where to go from here? Flags - Optimization

- Toolchain tested with same optimization options as the FOSS one
- Test LLVM specific optimization
  - Test if defaults for option not used is the same as in GCC
  - Test target specific optimizations and effect of march/mtune
  - Test the polyhedral optimizer (Polly)



#### Where to go from here? - Standalone vs + GCCcore

#### • Standalone

- No leftover GCC dependencies
- With a more mature libc could allow for a more system independent toolchain?
- Needs to redo all the GCCcore EC files
- On top of GCCcore
  - Reuse existing GCCcore dependencies
  - Risc mixing of OpenMP and compiler runtime libraries
    - Seems that calling a libgomp compiled libraries from a libomp linked executable works as intended



1 #include <stdio.h></stdio.h>		#include <stdio.h> #include <omp.h></omp.h></stdio.h>
2 #include <omp.h> 3</omp.h>		long long int func(long long int n);
<pre>4 long long int func(long long int n 5 long long int a = 0; 6</pre>	n) [[	int main(int argc, char **argv) { long long int b = 0;
<pre>7 #pragma omp critical 8 { 9 printf ("FUNC: omp_in_para</pre>	allel() = %s, I am thread %d\n",	<pre>printf ("Outside parallel: omp_in_parallel() = %s, omp_max_threads() = %d\n",</pre>
10 omp_in_parallel() 11 }	<pre>? "True":"False", omp_get_thread_num()); 13 14 15 16 17 17 17 17 17 17 17 17 17 17 17 17 17</pre>	#pragma omp parallel
12	ction(+:a) 16 n; i++) {	<pre>t printf ("MAIN: omp_in_parallel() = %s, I am thread %d\n",</pre>
15   a += i; 16 } 17   18   return a:	ncies	<pre>#pragma omp parallel for schedule( static ) reduction(+:b) for (int i = 0; i &lt; 4; i++) {     b += func(i*i*8000*8000); </pre>
19 <b>J</b>		
	a more manure upc could allow for a more syste	<pre>printt("D: %Llo(n", D); ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;</pre>
	Da.So	
<ul> <li>On top</li> <li>Ret</li> </ul>	<pre>/home/crivella/.local/easybuild/software/LLVM/20.1.0-all/lib/x86_64-unknow /home/crivella/test/hw/test_gcc_llvm_crossopenmp/liba.so [runpath] // /lib/x86_64-linux-gnu/libgomp.so.1 [ld.so.conf]</pre>	n-linux-gnu/ <b>libomp.so</b> [LD_LIBRARY_PATH]
<ul> <li>Risc r</li> </ul>	MIXING Of Running with 8 threads	
	Seems the MAIN: omp_in_parallel() = True, I am thread 0 MAIN: omp_in_parallel() = True, I am thread 3	• • • • • • • • • • • • • • • • • • •
	<pre>MAIN: omp_in_parallel() = True, I am thread 2 MAIN: omp_in_parallel() = True, I am thread 1 MAIN: omp_in_parallel() = True, I am thread 4 MAIN: omp_in_parallel() = True, I am thread 6 MAIN: omp_in_parallel() = True, I am thread 5 MAIN: omp_in_parallel() = True, I am thread 7 FUNC: omp_in_parallel() = True, I am thread 0 FUNC: omp_in_parallel() = True, I am thread 3 FUNC: omp_in_parallel() = True, I am thread 2 FUNC: omp_in_parallel() = True, I am thread 4 A = 0 A =</pre>	996maxresident)k
EUM - Julich 2025 Davide Grassano - CECAM	Centre Européen de Calcul Atomique et Moléculaire	

#### Where to go from here? - Standalone vs + GCCcore

- Standalone
  - No leftover GCC dependencies
  - With a more mature libc could allow for a more system independent toolchain?
  - Needs to redo all the GCCcore EC files
- On top of GCCcore
  - Reuse existing GCCcore dependencies
  - Risc mixing of OpenMP and compiler runtime libraries
    - Seems that calling a libgomp compiled libraries from a libomp linked executable works as intended
    - The opposite seems to lead to undefined behavior and the OMP part tied to libomp being ignored/not working







# Thanks for your attention. **Questions?**



# Compilers - Intermediate Representation (IR)

- Any data structure that can represent the program without loss of information so that its execution can be conducted accurately
- Requirements / design choices
  - Correctness
  - Clear/defined semantics
  - Decoupled/orthogonal operations
    - GPUs might benefit from non-orthogonal operations (eg fused multiply-add)
  - Completeness
- Forms
  - Textual (Debugging)
  - Bytecode (Persistence/exchange + possible compatibility)
  - In-memory (analysis/passes)



#### Compilers - Static Single Assignment (SSA)

• IR where each variable is assigned once





# Compilers - Static Single Assignment (SSA)

- IR where each variable is assigned once
- Enables/facilitates
  - Constant propagation
  - Dead-code elimination
  - Global value numbering
  - Branch predictions
    - Value range propagation
    - Sparse conditional constant propagation
  - Register allocation
  - Strength reduction



# Compilers - Static Single Assignment (SSA)

- Used by
  - Mono
  - WebKit
  - o Swift
  - Erlang
  - o LLVM
  - GNU Compiler collection
  - GO
  - IMB's java virtual machine
  - Mozilla JavaScript engine
  - Chrome JavaScript engine
  - PyPy
  - Android runtime

- Standard ML compiler
- LuaJIT
- HHVM PHP JIT compiler
- libFirm
- Oracle's JVM
- Microsoft Visual C++ compiler
- SPIR-V
- IBM XL compilers
- NVIDIA CUDA



#### Example of LLVM middle end optimizations



