How to run the LLVM-Test Suite on GPUs

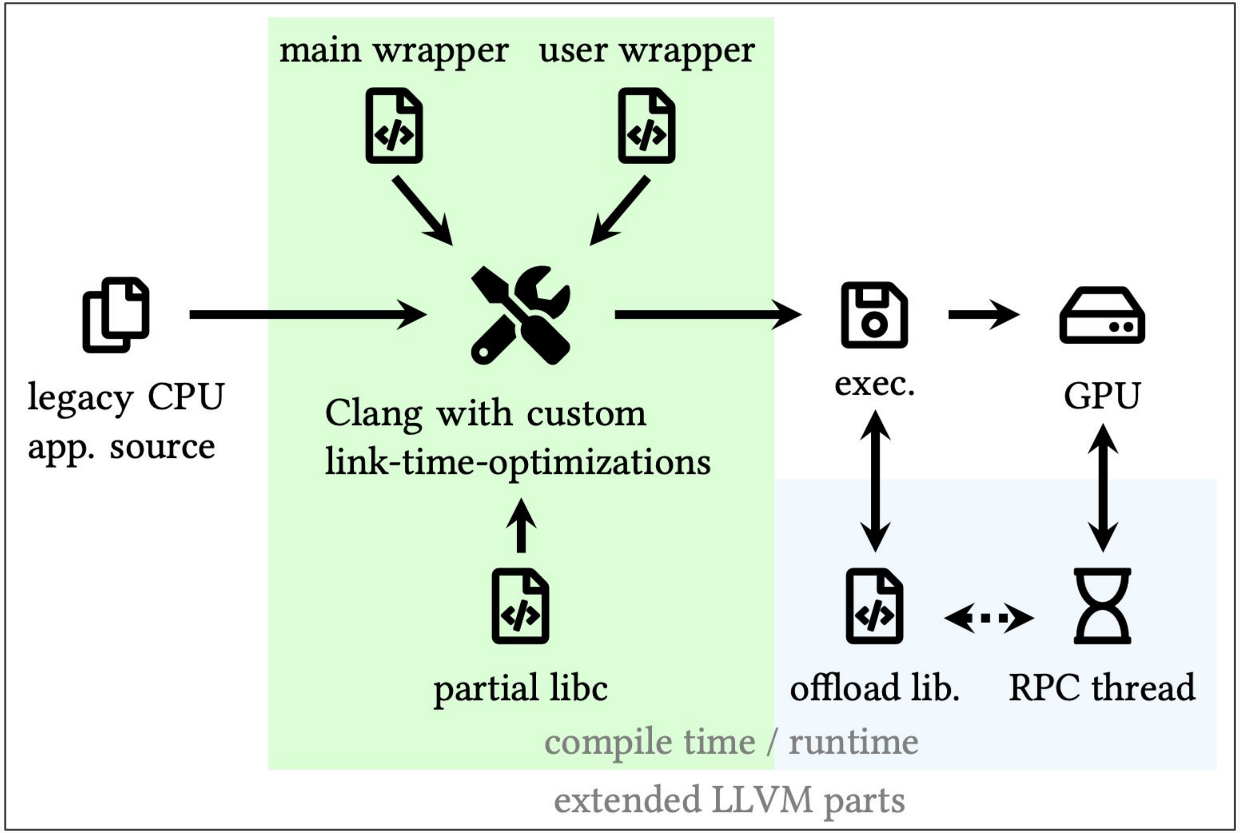
and what you’ll find

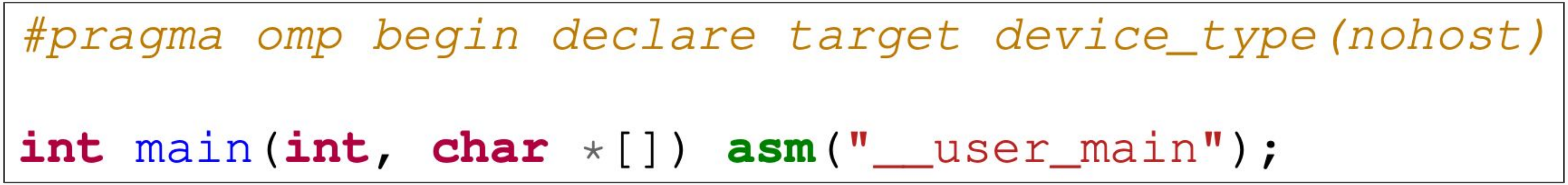
Johannes Doerfert <jdoerfert@llnl.gov>
Shilei Tian <shilei.tian@stonybrook.edu>

Prepared by LLNL under Contract DE-AC52-07NA27344.

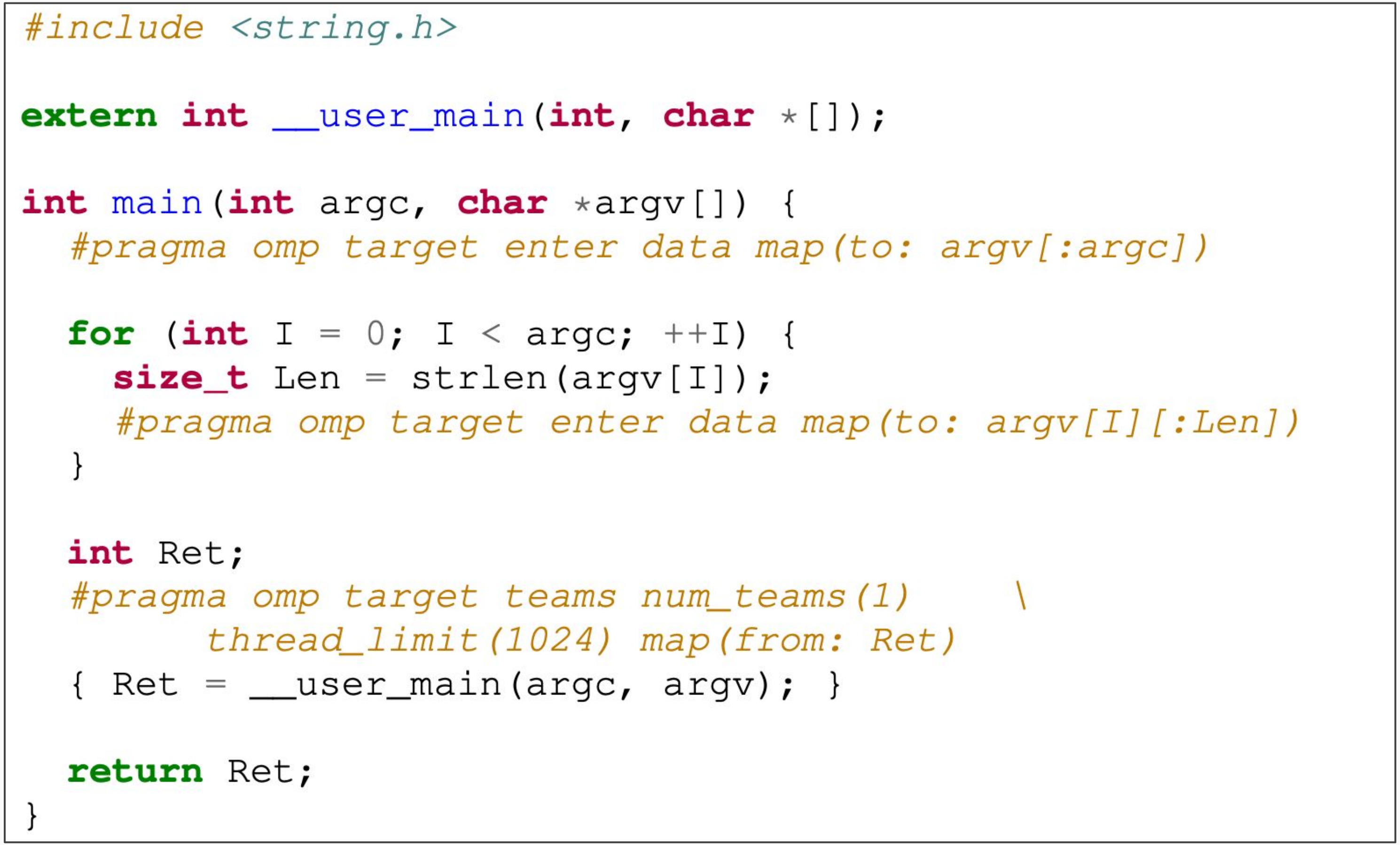
Direct GPU Compilation

GPU First — Execution of Legacy CPU Codes on GPUs



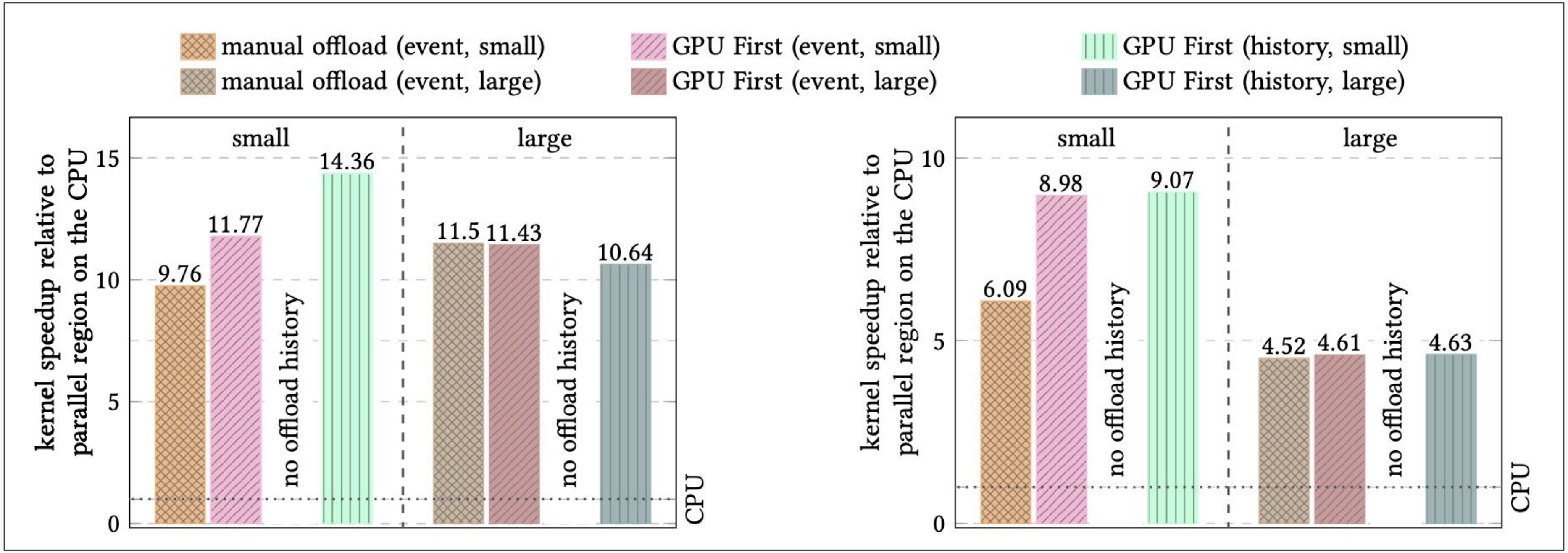


```
#pragma omp begin declare target device_type(nohost)

int main(int, char *[]) asm("__user_main");
```



```c
#include <string.h>

extern int __user_main(int, char *[]);

int main(int argc, char *argv[]) {
  #pragma omp target enter data map(to: argv[:argc])

  for (int I = 0; I < argc; ++I) {
    size_t Len = strlen(argv[I]);
    #pragma omp target enter data map(to: argv[I][:Len])
  }

  int Ret;
  #pragma omp target teams num_teams(1)     \
        thread_limit(1024) map(from: Ret)
  { Ret = __user_main(argc, argv); }

  return Ret;
}
```

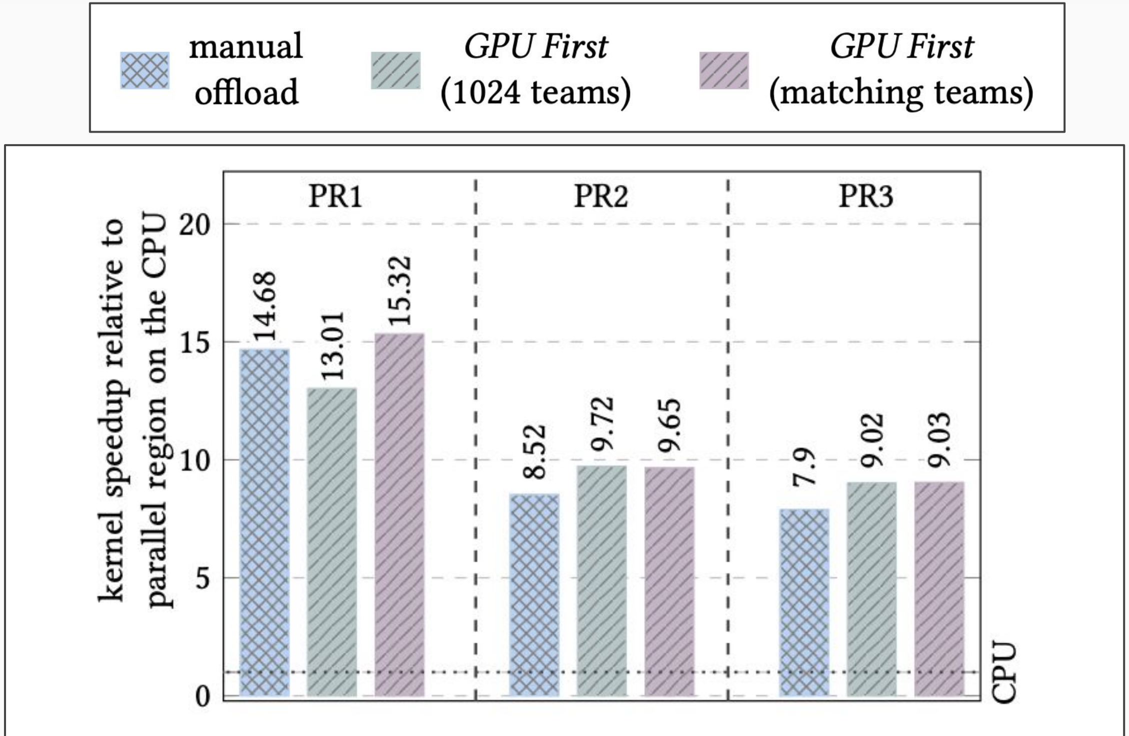
GPU First — Execution of Legacy CPU Codes on GPUs



XSBench

RSBench

GPU First — Execution of Legacy CPU Codes on GPUs



(b) Relative performance of the three parallel regions (PR1, PR2, PR3) in the hypterm micro benchmark (from HeCBench) when executed on the GPU instead of the CPU.

LLVM-Test Suite on the GPU

Pass Percentage (out of 2007 tests)

86.7%

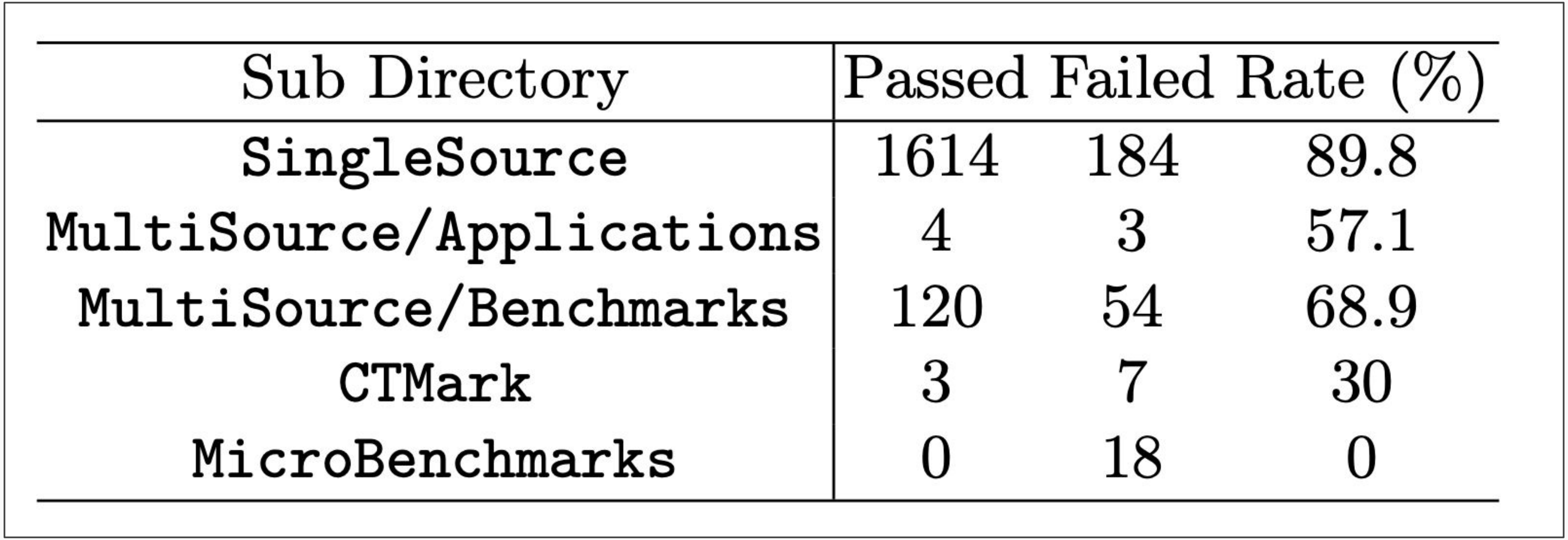
Pass Percentage (out of 2007 tests)

System + Setup

An NVIDIA A100 Tensor Core GPU (40GB) with AMD EPYC 7532 processors (32C/32T) and 256 GB DDR4 RAM.

```
$ cmake -G Ninja -S llvm-test-suite -DTEST_SUITE_RUN_BENCHMARKS=OFF \
  -DCMAKE_C_COMPILER=clang-gpu -DCMAKE_CXX_COMPILER=clang-gpu++
$ ninja
$ llvm-lit -v .
```

Results



Sub Directory	Passed	Failed	Rate (%)
`SingleSource`	1614	184	89.8
`MultiSource/Applications`	4	3	57.1
`MultiSource/Benchmarks`	120	54	68.9
`CTMark`	3	7	30
`MicroBenchmarks`	0	18	0

Problems — [1/3] Broken Tests

```
SingleSource/Benchmarks/SmallPT/smallpt.cpp

#pragma omp parallel for schedule(dynamic, 1) private(r)
 fprintf(stderr,"Rendering (%d spp)\n",samps*4);
```

```
MultiSource/Applications/sgefa/driver.c

char *malloc();
```

Problems — [2/3] Compiler Bugs

```
FAILED: CTMark/ClamAV/CMakeFiles/clamscan.dir/libclamav_readdb.c.o
...
clang-17: llvm-project/clang/lib/CodeGen/CodeGenFunction.cpp:2188:
clang::CodeGen::CodeGenFunction::VlaSizePair
clang::CodeGen::CodeGenFunction::getVLASize(const clang::VariableArrayType*):
Assertion ‘vlaSize && "no size for VLA!"’ failed.
```

```
MultiSource/Applications/ClamAV/libclamav_readdb.c
static int cli_loaddbdir_l(const char *dirname, struct cl_engine **engine, unsigned int *signo, unsigned int options)
{
        DIR *dd;
        struct dirent *dent;
        /* To sort the files in the temp dir to get repeatable results */
        const unsigned MAX_DIRENTS = 20;
        struct dirent dents[MAX_DIRENTS];
       ...
```

Problems — [3/3] Compiler Limitations

extern globals

```
extern std::ostream cout;  // <iostream>
extern char *optarg;       // <unistd.h>
```

C++ exceptions

```
throw
catch
```

variadic user functions

```
__builtin_va_arg
```

unsupported types

```
long double
```

inline assembly

```
asm(...)
```