Precision and Performance Analysis of LLVM's C Standard Math Library on GPUs

By Anton Rydahl, Joseph Huber, Ethan McDonough, and Johannes Doerfert

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

About Me

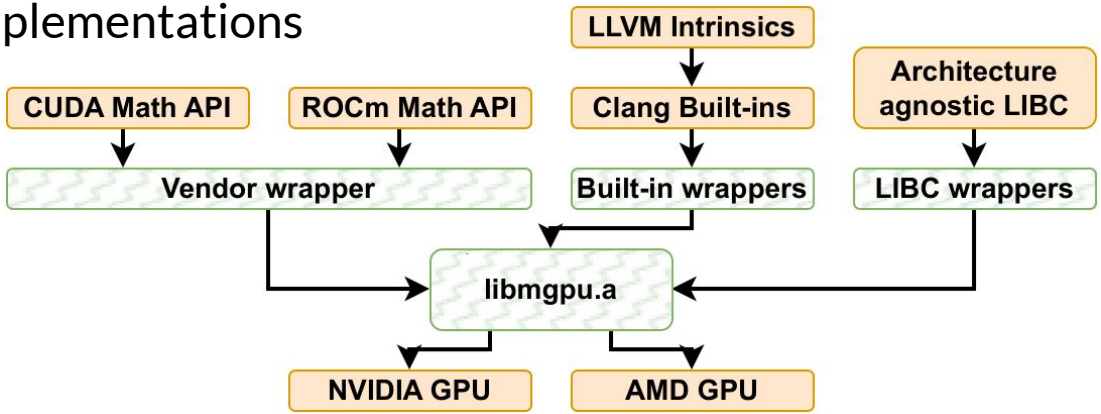
- MSc in Mathematical Modelling and Computation
- Interning at Lawrence Livermore National Laboratory
- Working on libc, libc++, and OpenMP in LLVM

Background

- LLVM's *libc* is being developed for GPUs
- Clang uses vendor libraries
- **Explore what LLVM infrastructure can be reused on GPUs**

GPU Math Libraries

- NVIDIA's CUDA Math
- AMD's HIP Math
- LLVM intrinsics
- Target agnostic implementations



GPU Math Libraries

```
$ llvm-ar x libmgpu.a remquof.cpp.o
$ llvm-objdump --offloading remquof.cpp.o
remquof.cpp.o:          file format elf64-x86-64
OFFLOADING IMAGE [0]:
kind            llvm ir
arch            gfx906
triple          amdgcn-amd-amdhsa
producer        none
OFFLOADING IMAGE [1]:
kind            llvm ir
arch            gfx90a
triple          amdgcn-amd-amdhsa
producer        none
OFFLOADING IMAGE [2]:
kind            llvm ir
arch            sm_70
triple          nvptx64-nvidia-cuda
producer        none
OFFLOADING IMAGE [3]:
kind            llvm ir
arch            sm_80
triple          nvptx64-nvidia-cuda
producer        none
```

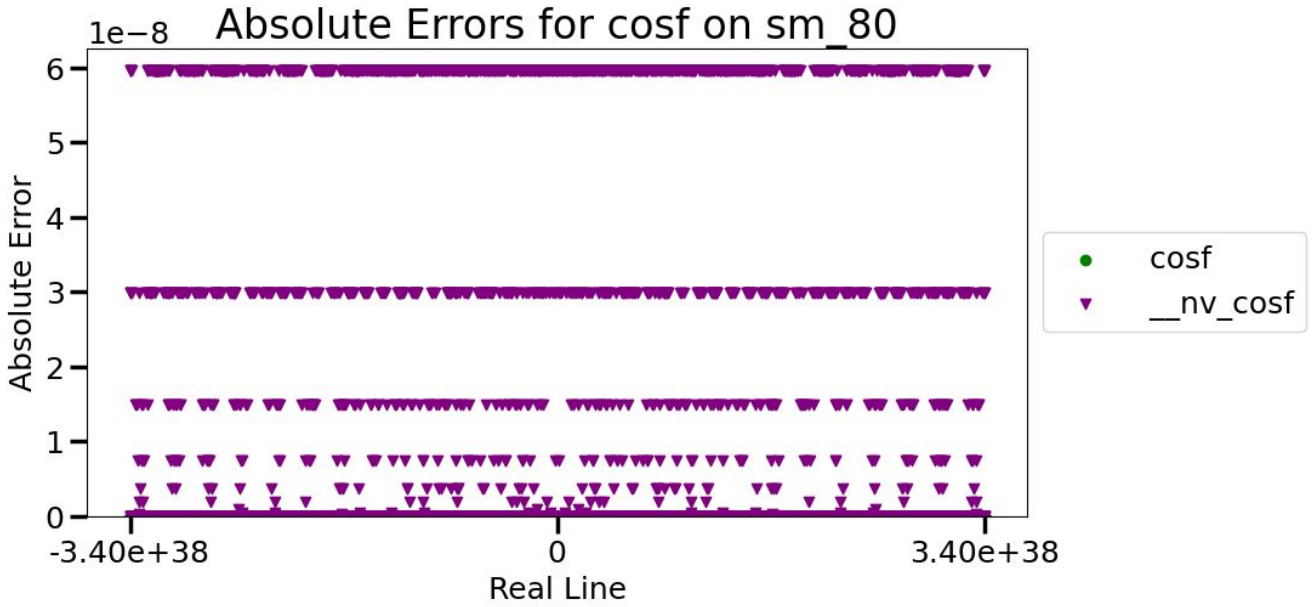
Correctness

- Exhaustive search for univariate functions with 32 bit data types
  - Upper bound on error

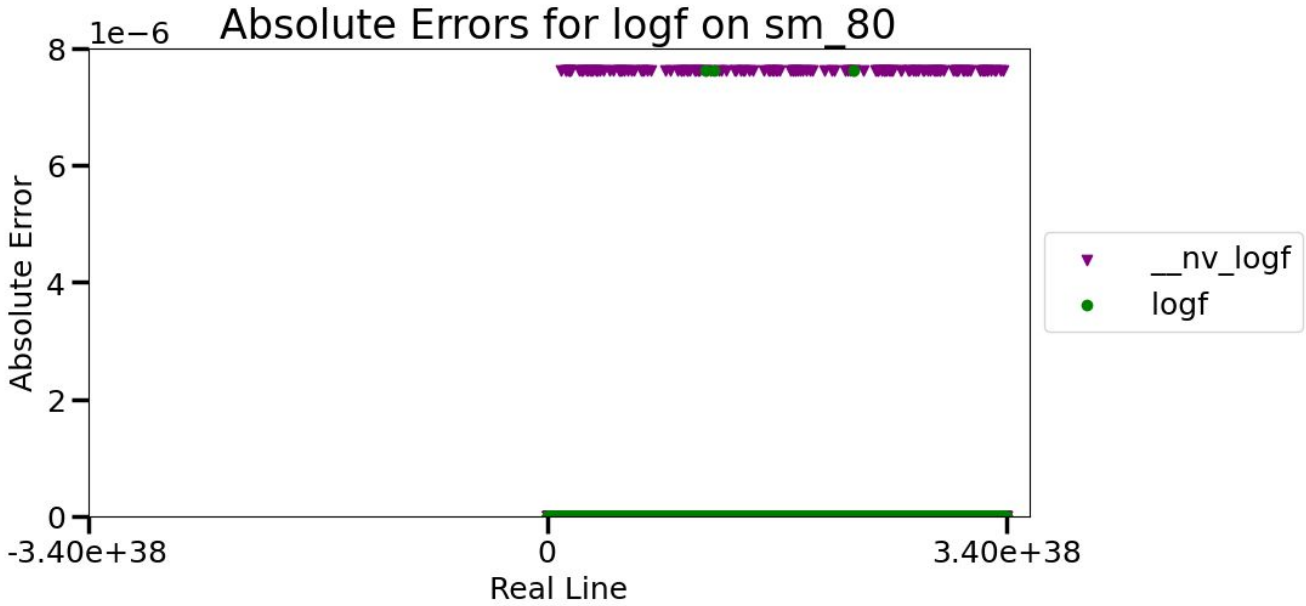
- Uniformly distributed input for 64 bit data
  - Lower bound on error

- Comparing against the GNU MPFR library
  - Arbitrary precision

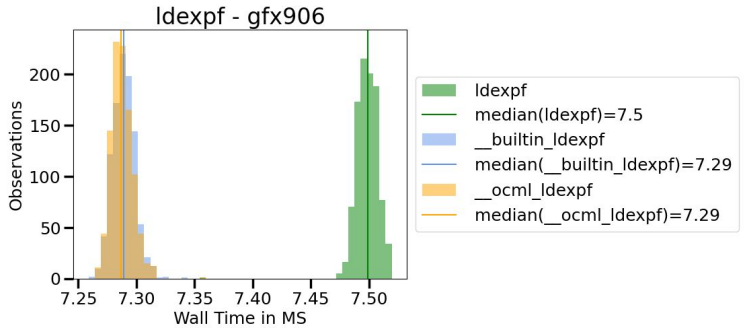
Correctness

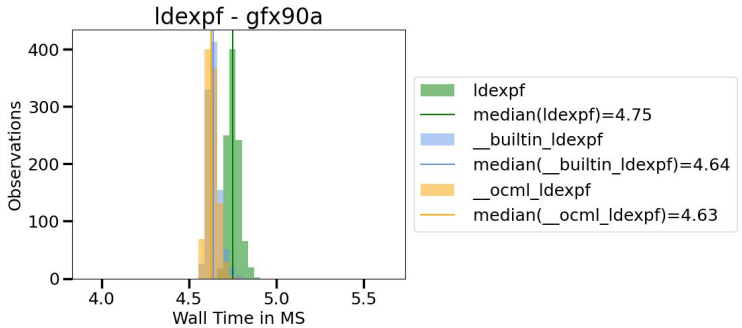


Correctness

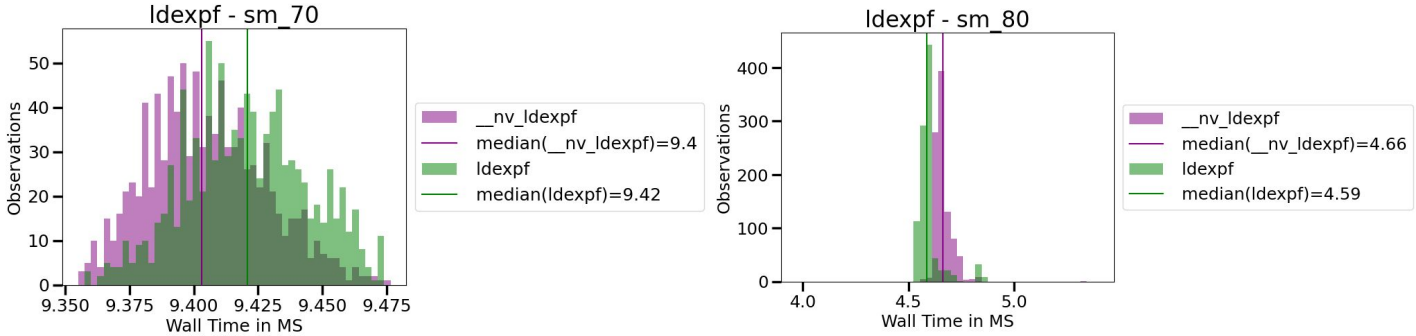


Timings


ldexpf - gfx906
Observations
0
50
100
150
200
7.25
7.30
7.35
7.40
7.45
7.50
Wall Time in MS
ldexpf
median(ldexpf)=7.5
__builtin_ldexpf
median(__builtin_ldexpf)=7.29
__ocml_ldexpf
median(__ocml_ldexpf)=7.29


ldexpf - gfx90a
Observations
0
100
200
300
400
4.0
4.5
5.0
5.5
Wall Time in MS
ldexpf
median(ldexpf)=4.75
__builtin_ldexpf
median(__builtin_ldexpf)=4.64
__ocml_ldexpf
median(__ocml_ldexpf)=4.63

Timings


ldexpf - sm_70
Observations
0
10
20
30
40
50
9.350
9.375
9.400
9.425
9.450
9.475
Wall Time in MS
__nv_ldexpf
median(__nv_ldexpf)=9.4
ldexpf
median(ldexpf)=9.42
ldexpf - sm_80
Observations
0
100
200
300
400
4.0
4.5
5.0
Wall Time in MS
__nv_ldexpf
median(__nv_ldexpf)=4.66
ldexpf
median(ldexpf)=4.59

Fastest Correct Set of Functions

- HIP
- CUDA
- LLVM Libc
- LLVM Builtin

Function	gfx906	gfx90a	sm_70	sm_80
log	L	L	C	C
log10	L	H	C	C
log1p	L	L	L	L
log2	L	H	C	C
logb	H	H	C	L
logbf	H	H	B	C
nearbyint	H	B	C	B
nearbyintf	H	H	B	B
nextafter	B	L	L	C
nextafterf	L	L	C	C
pow	H	H	C	C
powf	H	H	C	C
remainder	H	H	B	B
remainderf	H	H	B	B

Fastest Correct Set of Functions

- HIP
- CUDA
- LLVM Libc
- LLVM Builtin

Function	gfx906	gfx90a	sm_70	sm_80
log	L	L	C	C
log10	L	H	C	C
log1p	L	L	L	L
log2	L	H	C	C
logb	H	H	C	L
logbf	H	H	B	C
nearbyint	H	B	C	B
nearbyintf	H	H	B	B
nextafter	B	L	L	C
nextafterf	L	L	C	C
pow	H	H	C	C
powf	H	H	C	C
remainder	H	H	B	B
remainderf	H	H	B	B

Results of the Analysis

- Given a tolerance, we can find an optimal set of mathematical functions
  - Depends on the target architecture
  - 7 times faster than CUDA Math on sm_80 on average
    - Influenced by outliers
    - Sensitive to inlining
  - 5 % faster than HIP Math on gfx906