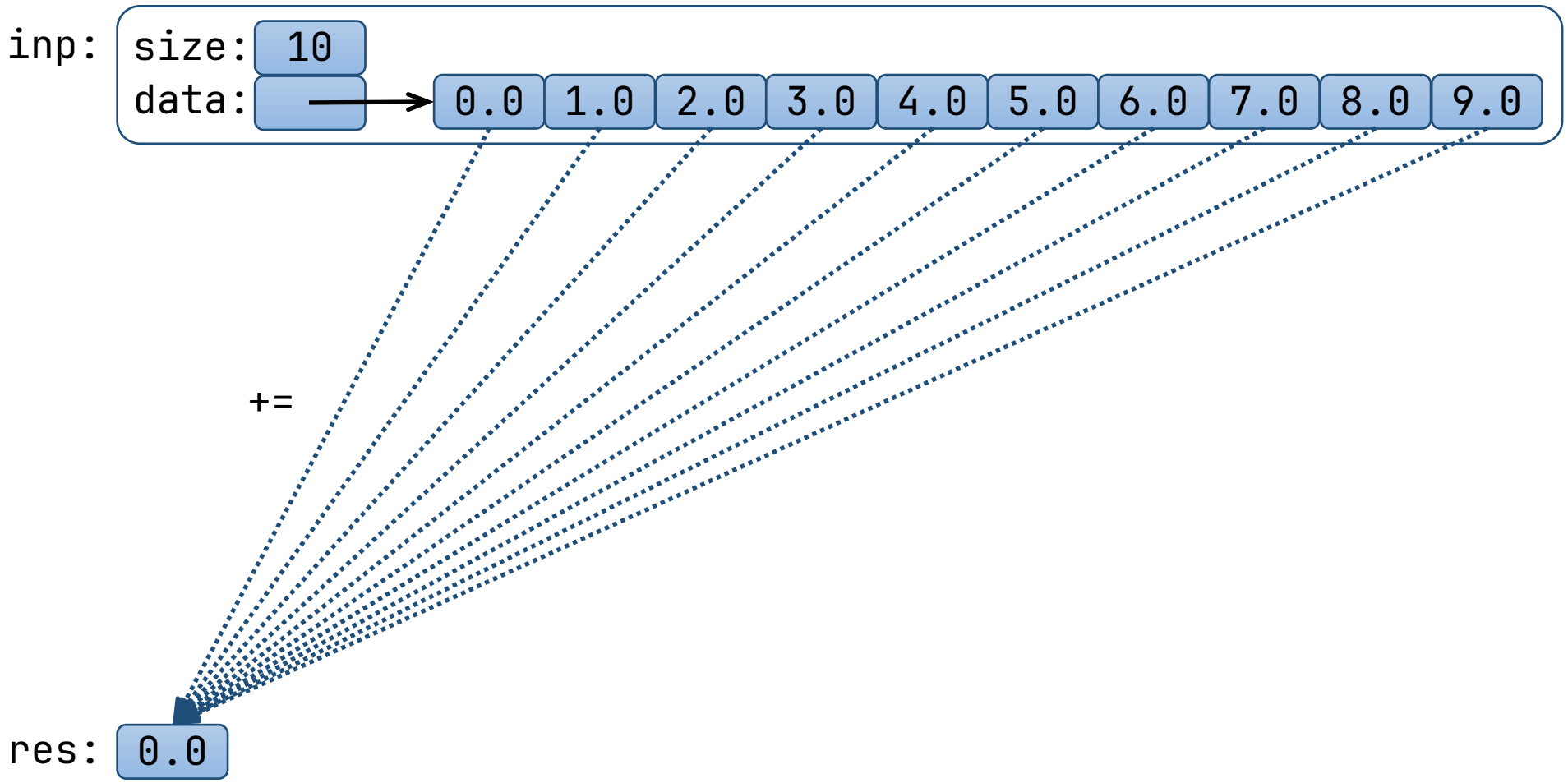
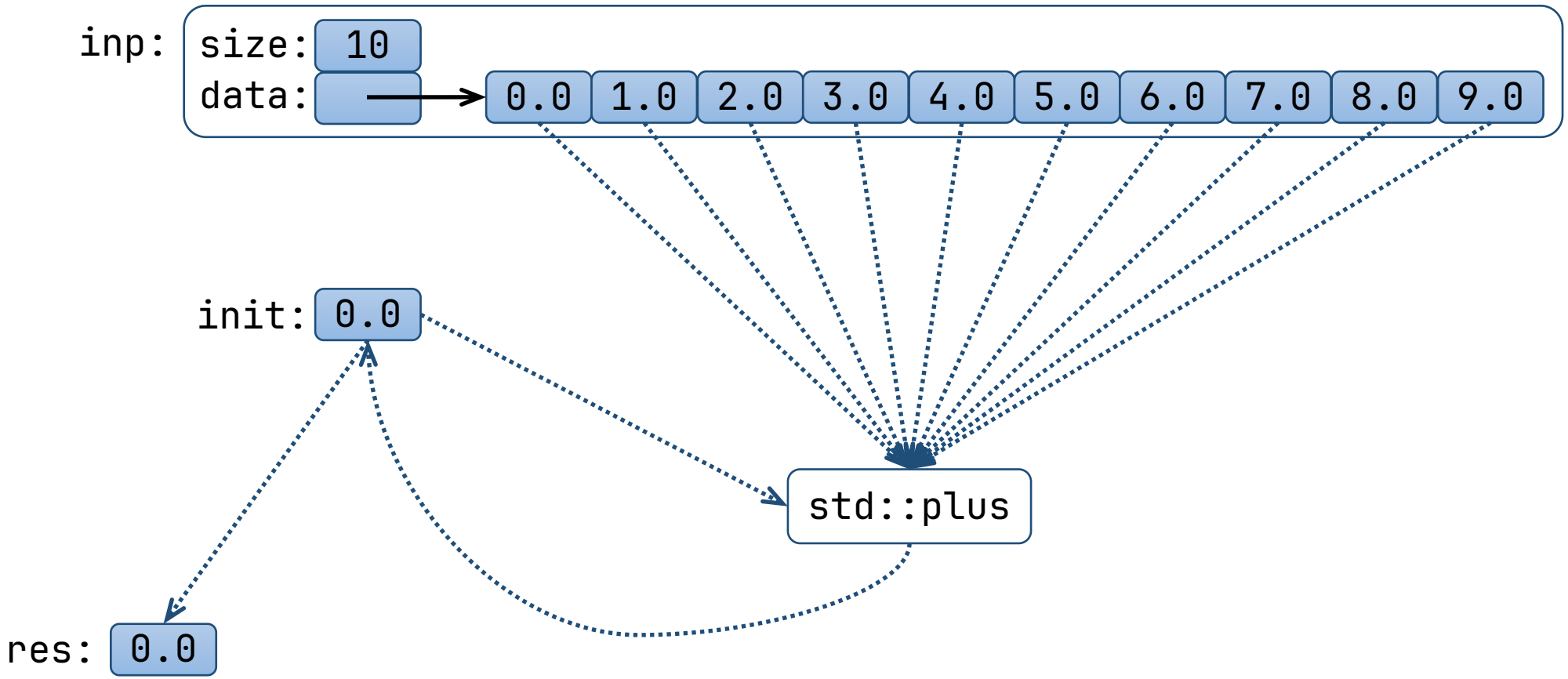
Summation Benchmarks

Marius Mikučionis <marius@cs.aau.dk>

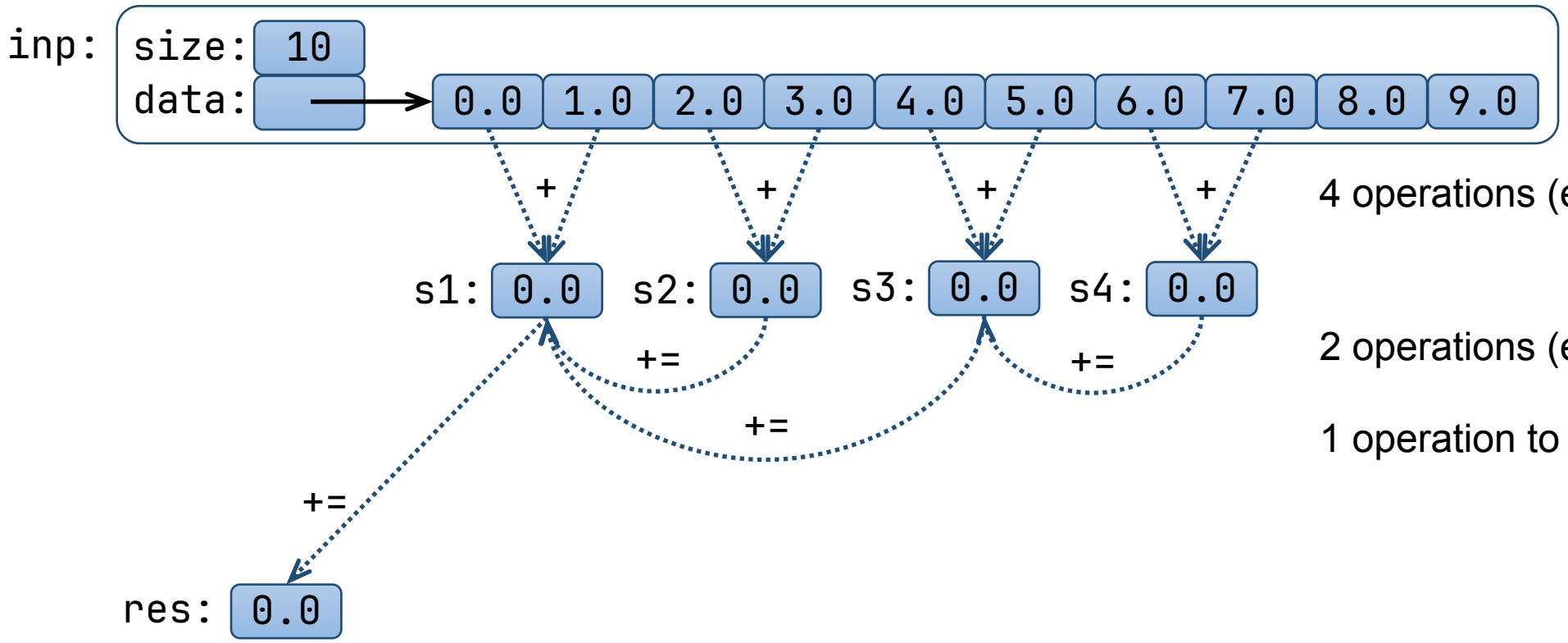
sum_loop: addition over simple loop



sum_accumulate: addition using std::accumulate



sum_vec4: vectorized addition using 4 registers

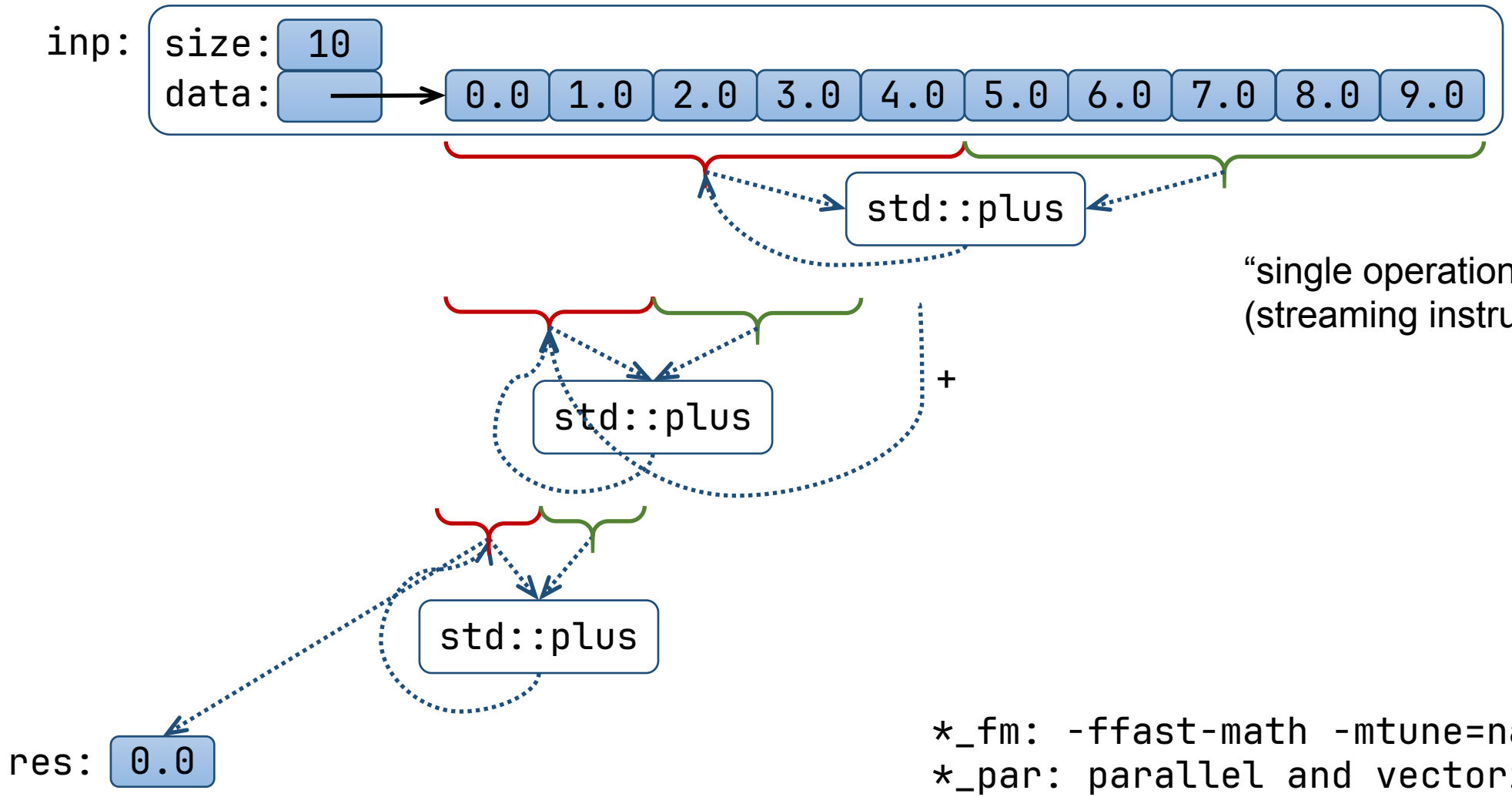


4 operations (executed in parallel)

2 operations (executed in parallel)

1 operation to add all up

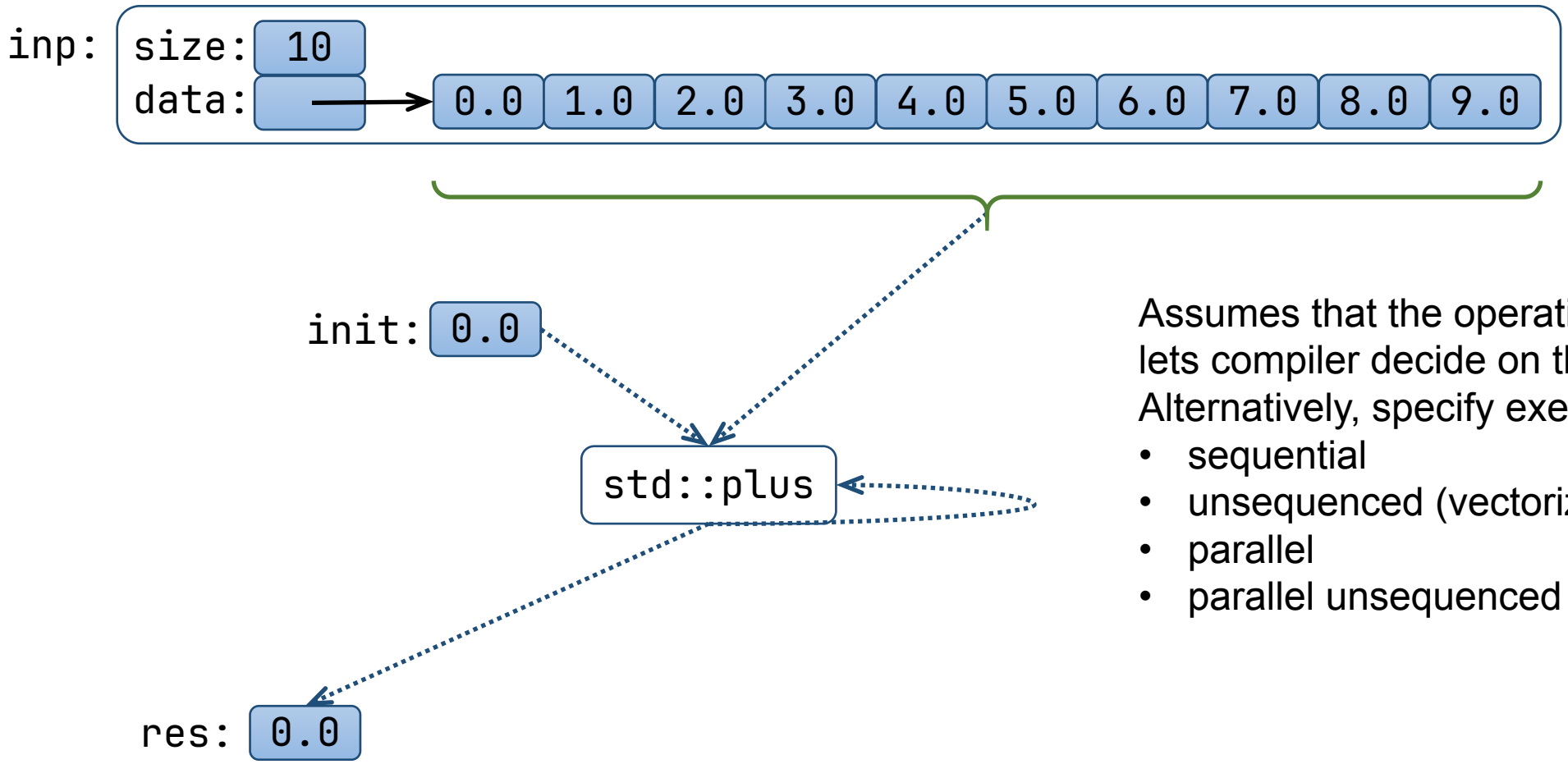
sum_transform: addition using std::transform over ranges



"single operation over many data"
(streaming instructions?)

```
*_fm: -ffast-math -mtune=native
*_par: parallel and vectorized execution
*_simd: Single Instruction Multiple Data
```

sum_reduce: addition using std::reduce



Assumes that the operation is commutative, lets compiler decide on the optimal order.
Alternatively, specify execution order:

- sequential
- unsequenced (vectorized)
- parallel
- parallel unsequenced

```
*_fm: -ffast-math -mtune=native
*_par: parallel and vectorized execution
*_simd: Single Instruction Multiple Data
```

Conventional Project Directory Structure

```
include
    output.hpp
    sum.hpp
source / src
    output.cpp
    sum_loop.cpp
    sum_accumulate.cpp
    sum_vec4.cpp
    sum_transform.cpp
    sum_reduce.cpp
    CMakeLists.txt
tests
    output_test.cpp
    sum_test.cpp
    sum_bm.cpp
    CMakeLists.txt
cmake
    doctest.cmake
    benchmark.cmake
    tbb.cmake
    eve.cmake
CMakeLists.txt
```

Library interface: public *declarations* in headers (no c/cpp files here)
Good place to start looking what's "in the box"
Hopefully that's ***all*** you need to know ***how to use*** the library, see also *Tests*

Implementation details:

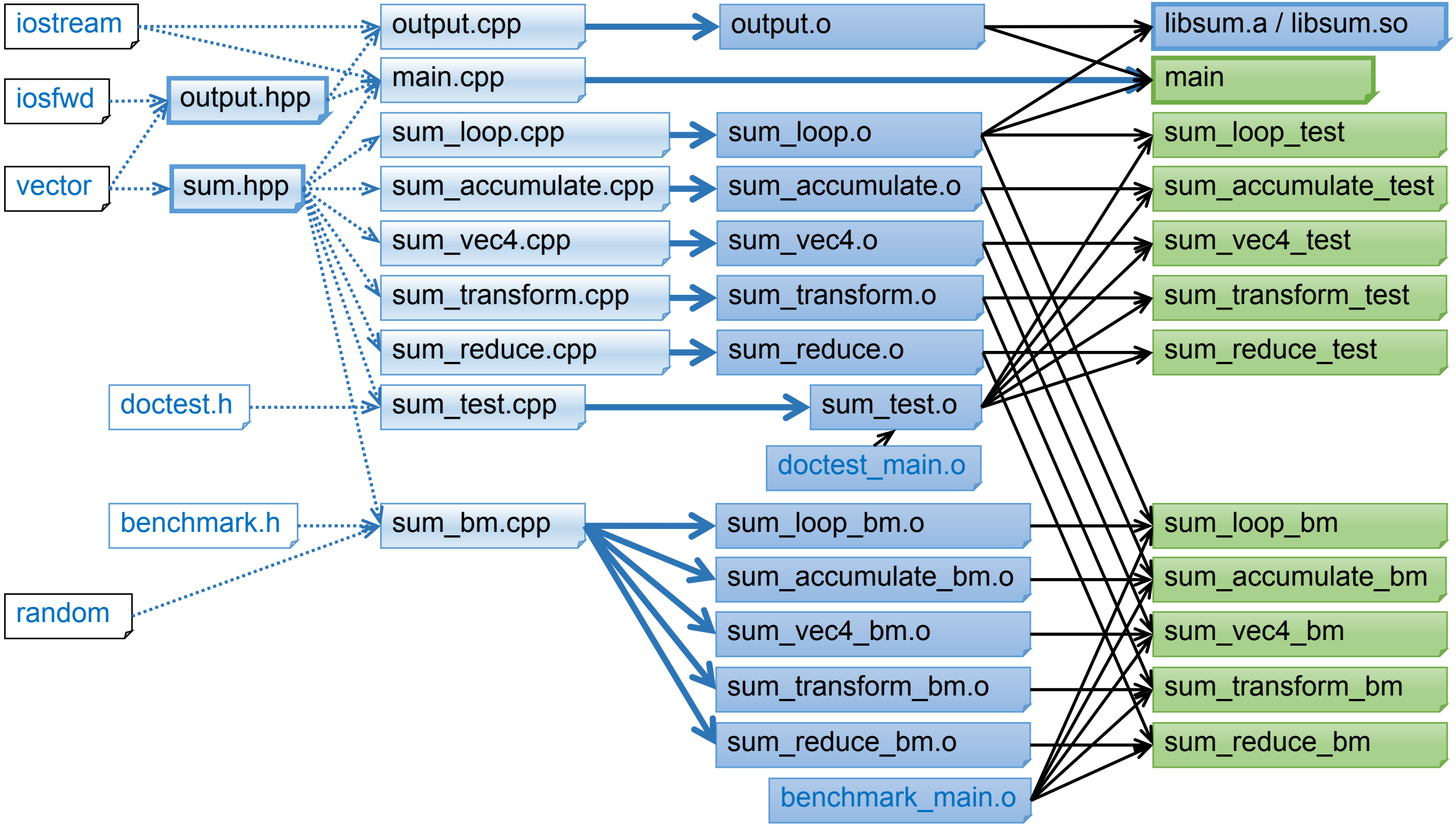
- private *declarations* (h/hpp headers) and
- *definitions* (implementation in c/cpp files)
- *building* and *linking* scripts

Tests and ***benchmarks***:

- private *declarations (h/hpp headers)* and
- *definitions* (c/cpp files)
- *building* and *linking* scripts

Build configuration and ***deployment scripts***:

- *dependency checks* for *system* and *third-party* tools and libraries
- *third-party* tool and library installation scripts
- library *features*, compiler and linker *options*
- library *installation* scripts


iostream
iosfwd
vector
doctest.h
benchmark.h
random
output.hpp
sum.hpp
output.cpp
main.cpp
sum_loop.cpp
sum_accumulate.cpp
sum_vec4.cpp
sum_transform.cpp
sum_reduce.cpp
sum_test.cpp
sum_bm.cpp
output.o
sum_loop.o
sum_accumulate.o
sum_vec4.o
sum_transform.o
sum_reduce.o
sum_test.o
doctest_main.o
sum_loop_bm.o
sum_accumulate_bm.o
sum_vec4_bm.o
sum_transform_bm.o
sum_reduce_bm.o
benchmark_main.o
libsum.a / libsum.so
main
sum_loop_test
sum_accumulate_test
sum_vec4_test
sum_transform_test
sum_reduce_test
sum_loop_bm
sum_accumulate_bm
sum_vec4_bm
sum_transform_bm
sum_reduce_bm