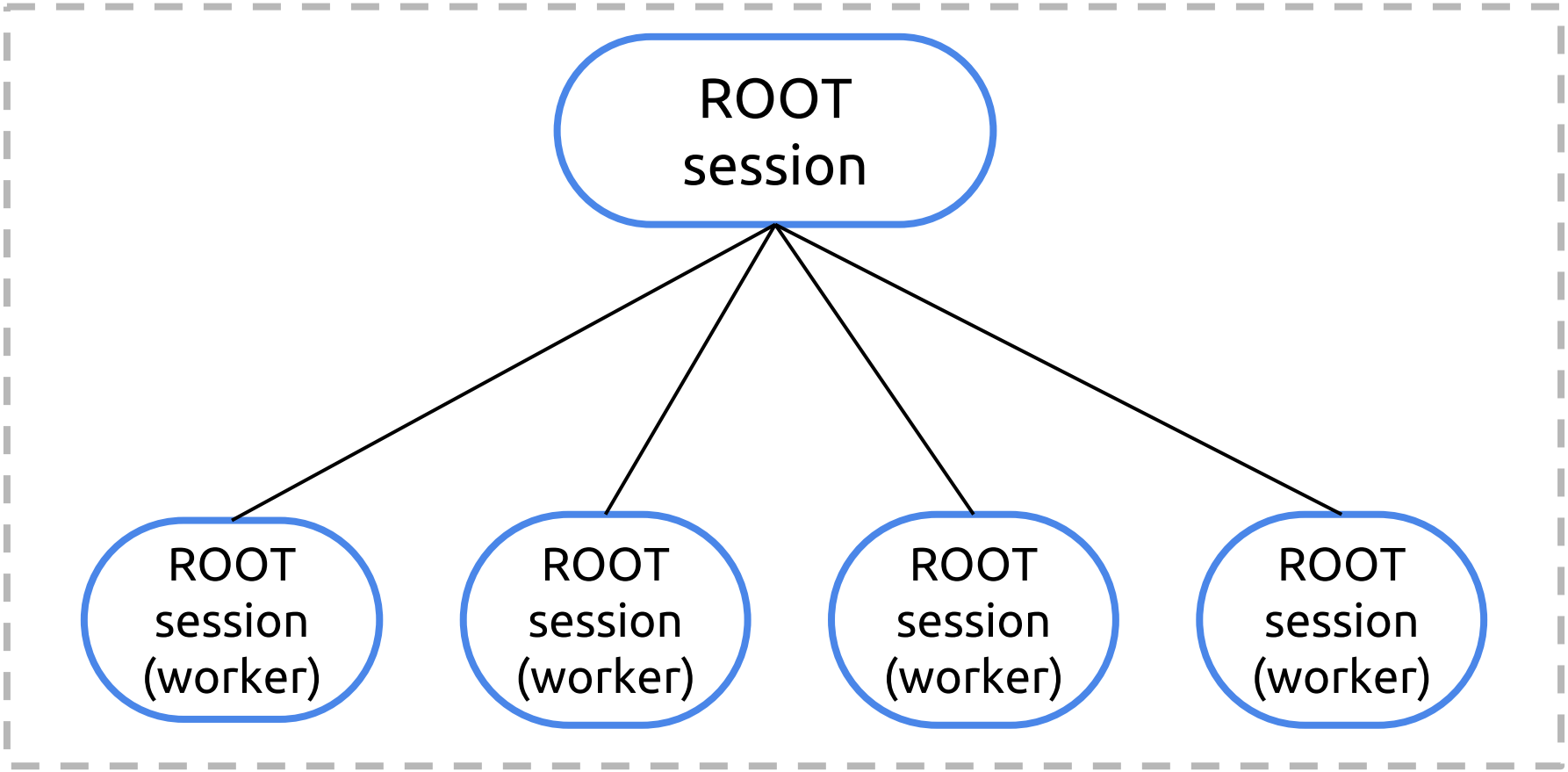
Enhancements to multiprocessing in ROOT

Enrico Guiraud

supervisor:
Gerardo Ganis

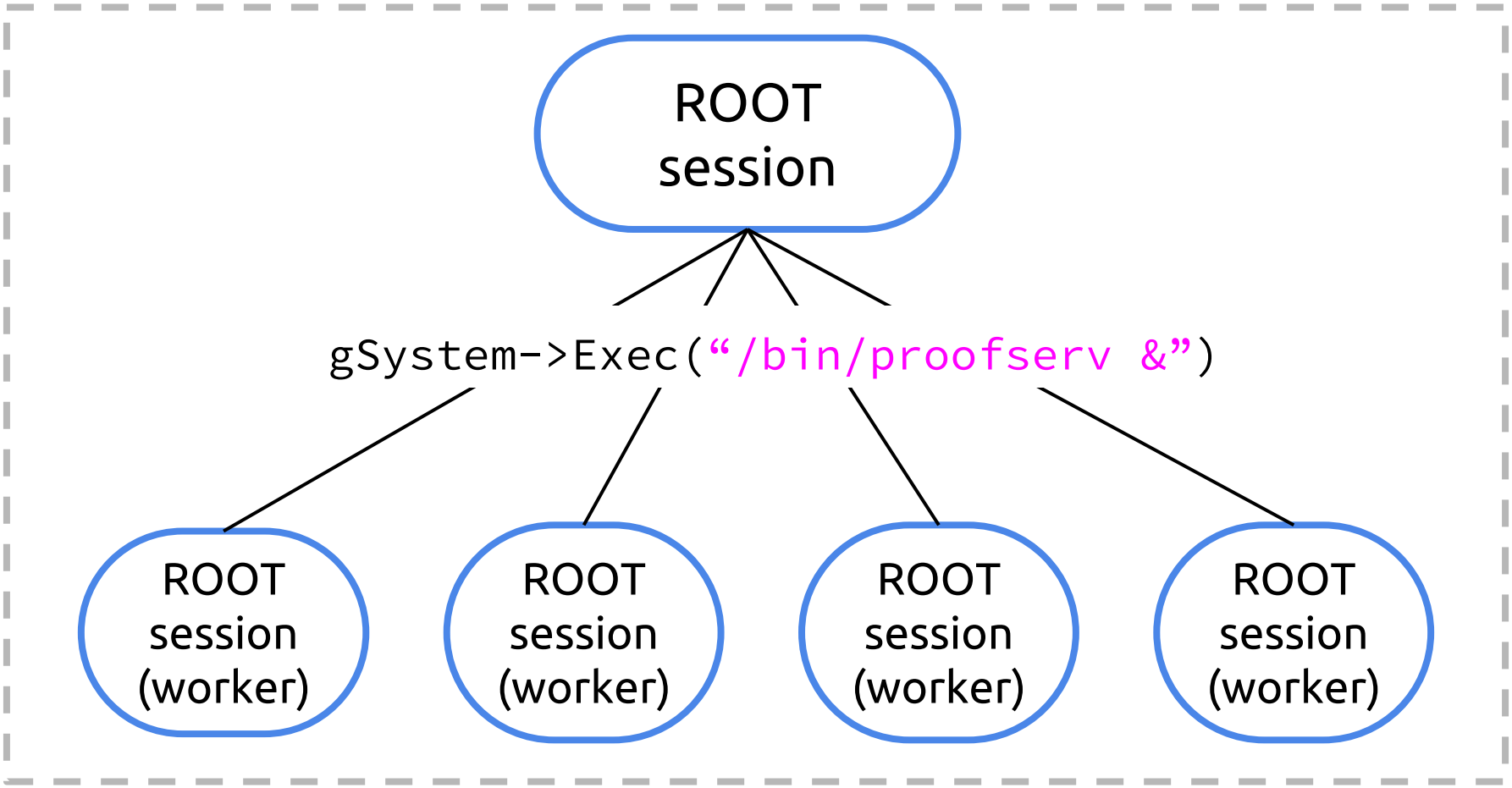
https://github/bluehood/root

New lightweight framework for multi-process applications



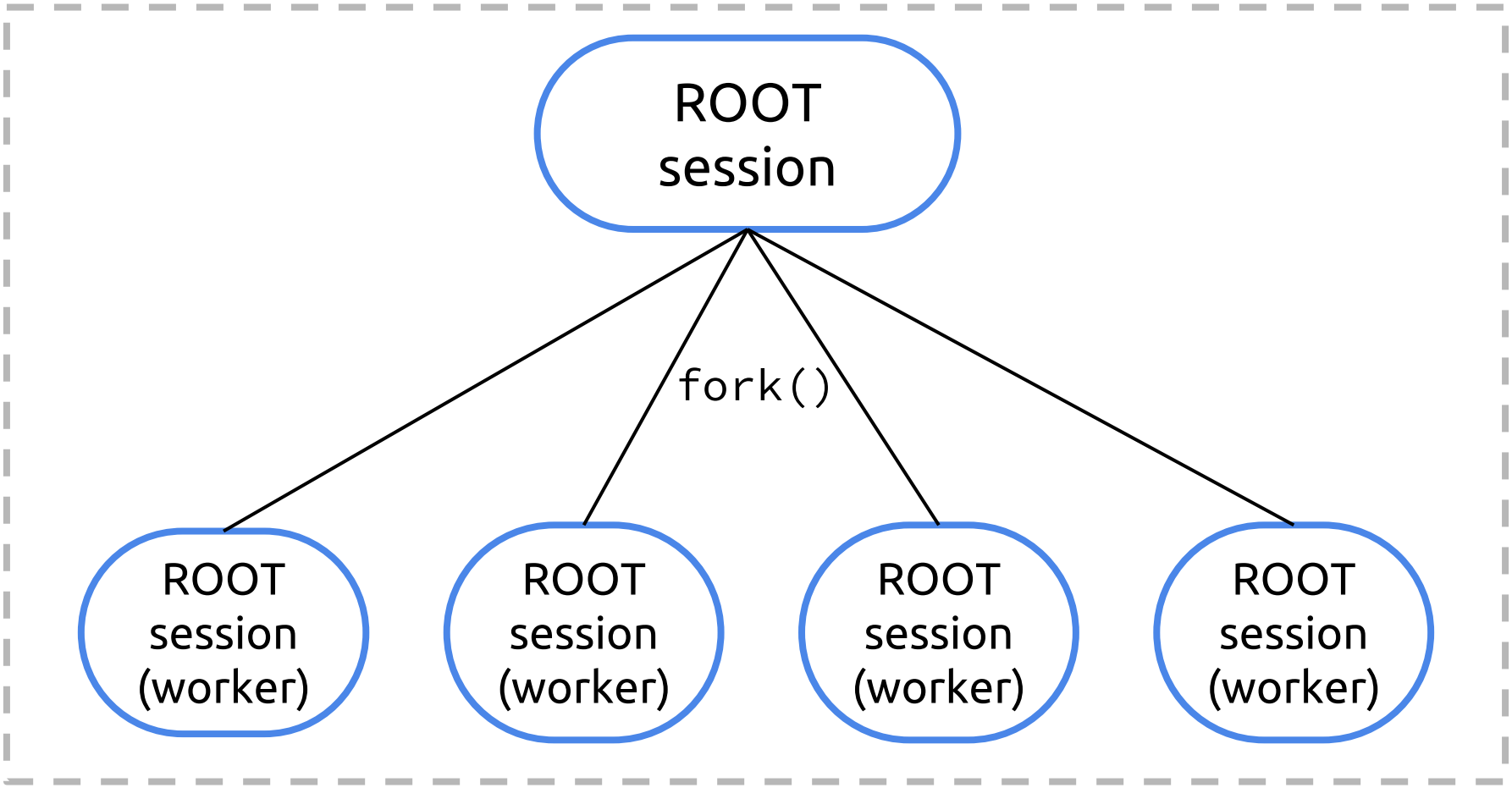
https://github/bluehood/root

The *old* way to create workers



https://github/bluehood/root

...the *new* way to create workers



https://github/bluehood/root

Advantages of the new approach

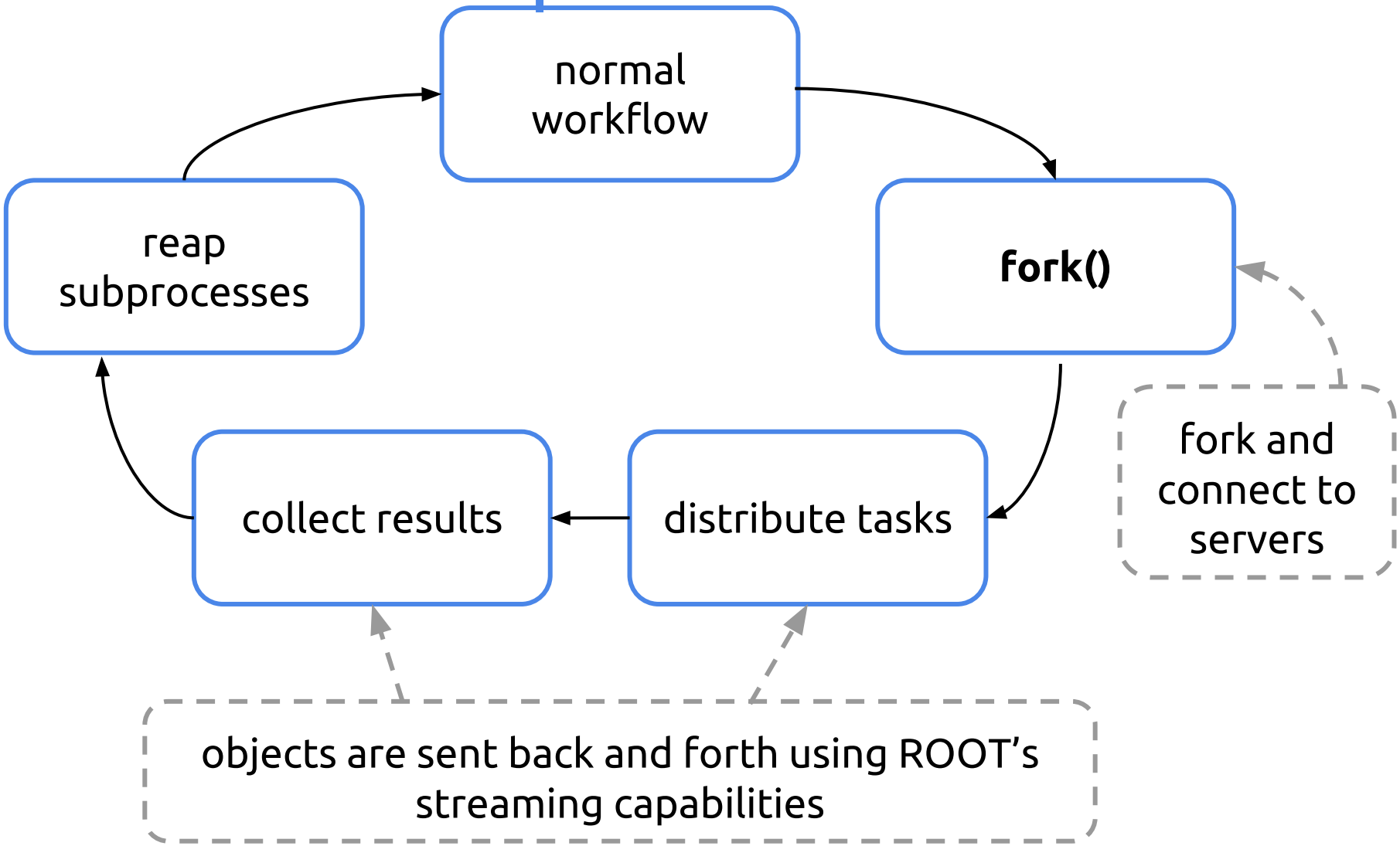
- ➔ versatility: workers have access to the session's environment
- ➔ small footprint: workers' memory is copy-on-write
- ➔ starting and stopping workers is cheap and fast

Disadvantage

- ➔ portability

https://github/bluehood/root

A new type of workflow is possible!



https://github/bluehood/root

An application: the Map function

```
TPool pool

auto result = pool.Map( func, arguments )
```

https://github/bluehood/root

An application: the Map function

defaults to n. of cores

```
TPool pool(8)

auto result = pool.Map( func, arguments )
```

https://github/bluehood/root

An application: the Map function

defaults to n. of cores

```
TPool pool(8)
```

C/C++ function

loaded macro

functor class

std::function

lambda expr.

```
auto result = pool.Map( func, arguments )
```

https://github/bluehood/root

An application: the Map function

defaults to n. of cores

```
TPool pool(8)
```

C/C++ function

loaded macro

functor class

std::function

lambda expr.

```
auto result = pool.Map( func, arguments )
```

std::*container*

initializer list

TCollection*

https://github/bluehood/root

An application: the Map function

defaults to n. of cores

```
TPool pool(8)
```

C/C++ function

loaded macro

functor class

std::function

lambda expr.

```
auto result = pool.Map( func, arguments )
```

std::vector

or

TObjArray

std::*container*

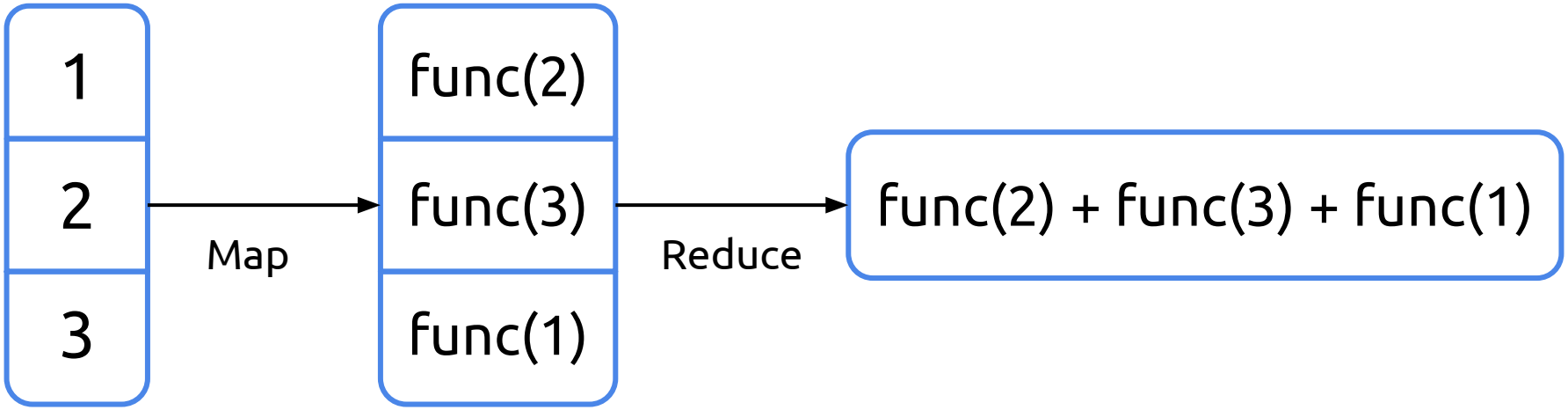
initializer list

TCollection*

https://github/bluehood/root

An application: the Map function

What about Reduce?



https://github/bluehood/root

An application: the Map function

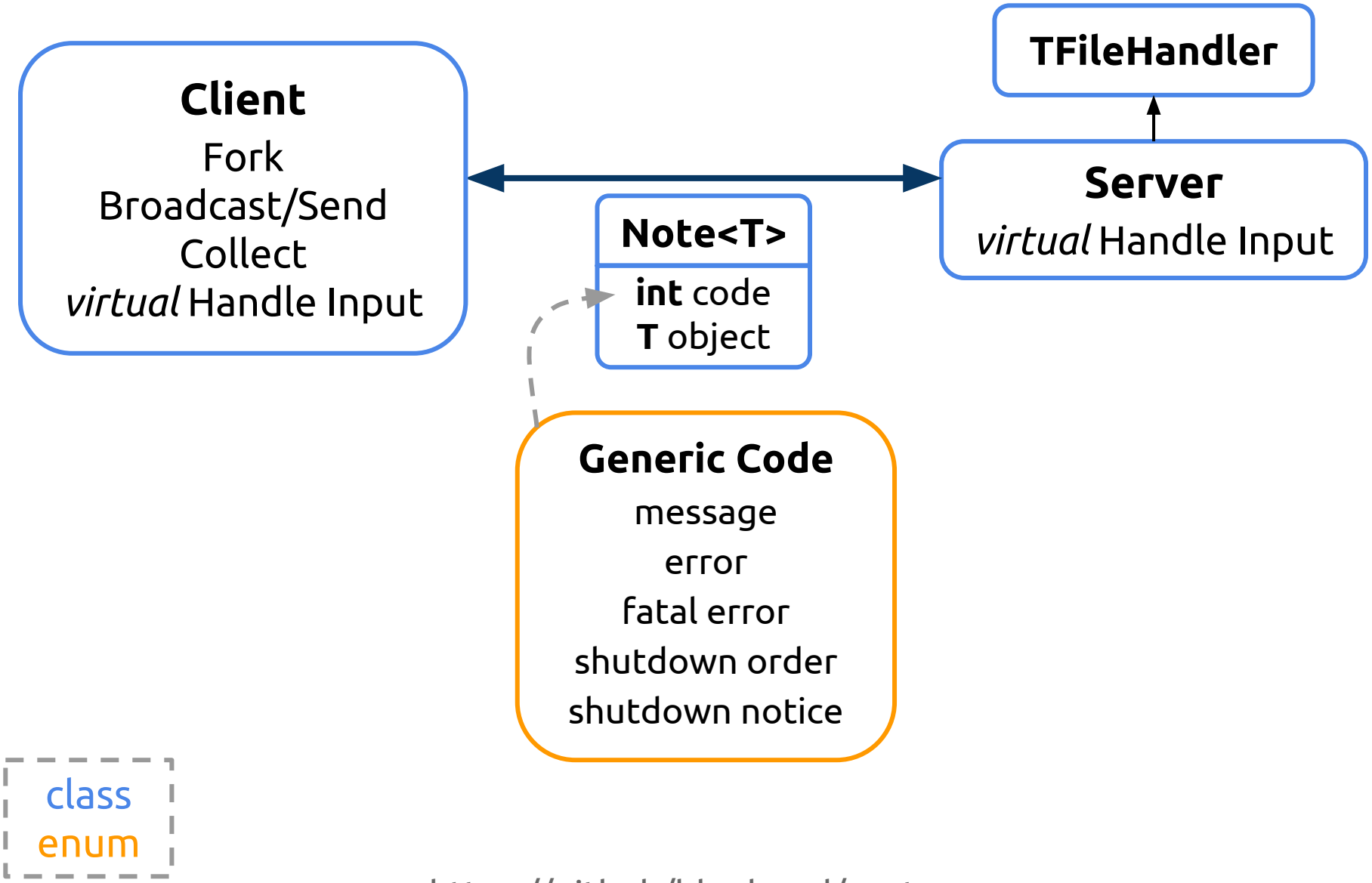
What about Reduce?

MapReduce(func, args, reduce_func)


IT COMPILES!
IT COMPILES!
memegenerator.net

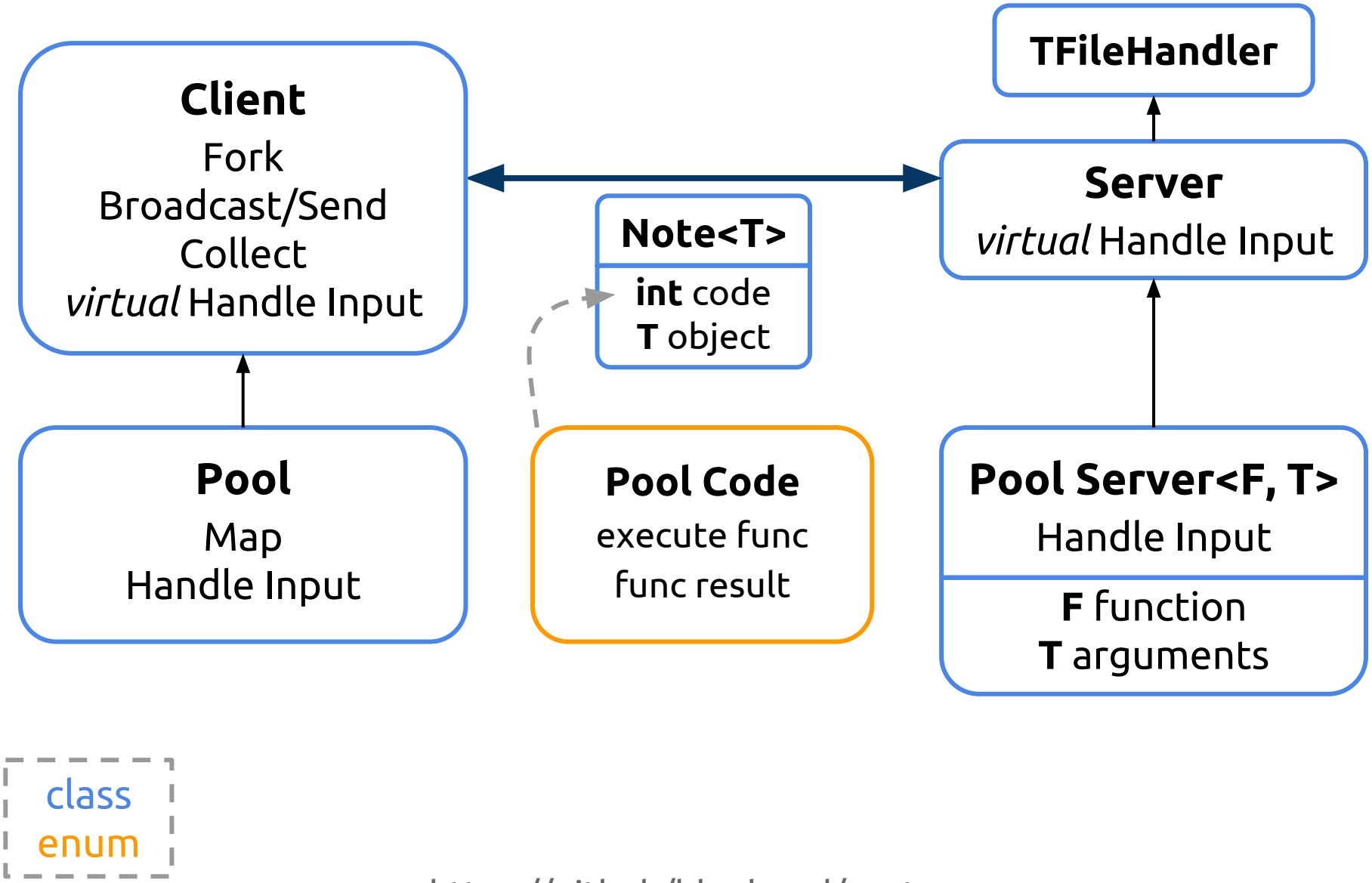

Thank you for your time!
Thank you for having me *here*!

Implementation details



https://github/bluehood/root

Implementation details: Map



https://github/bluehood/root

An application: the Map function

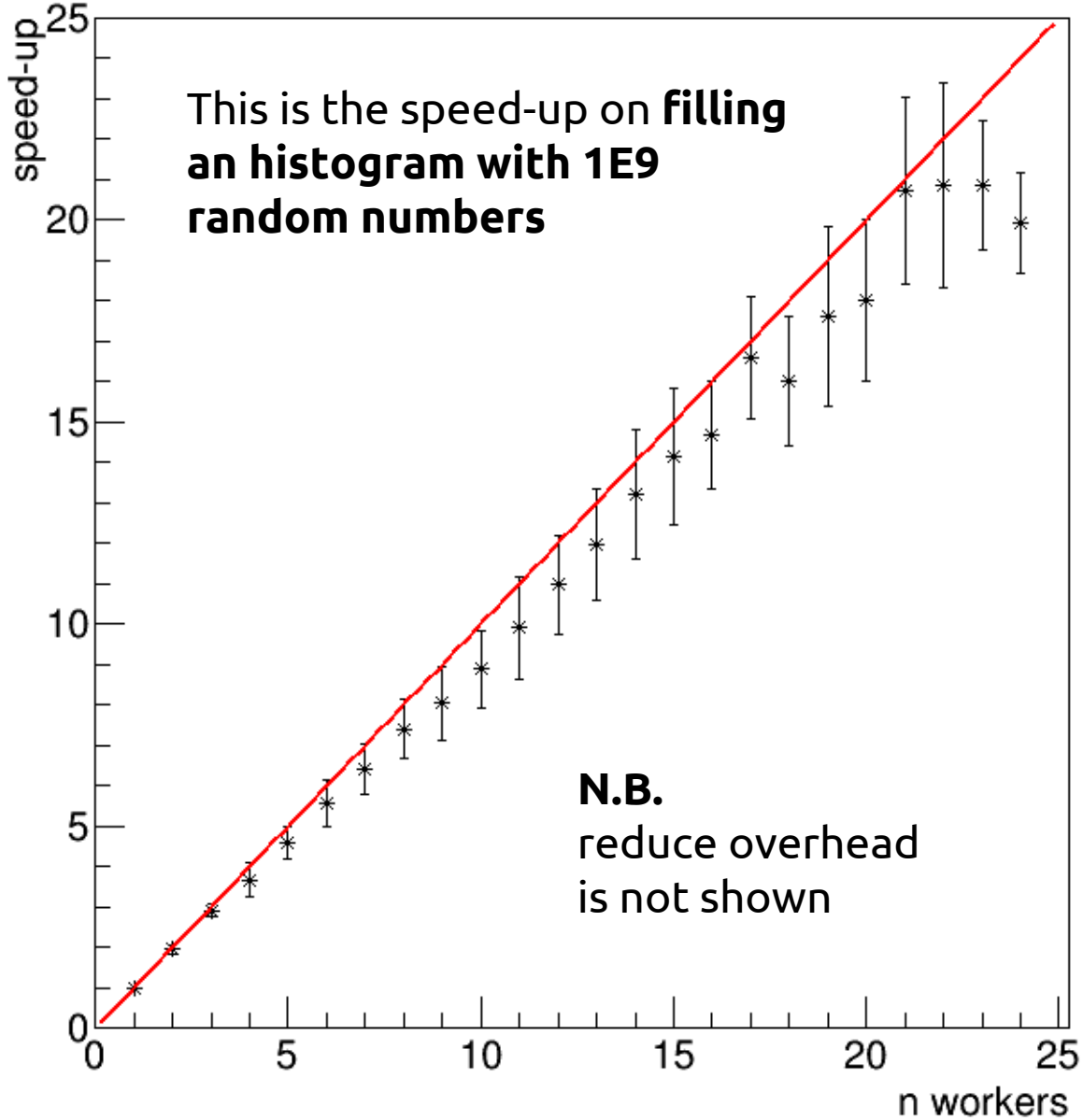
```
TPool  pool(8)

.L  myMacro.cxx+

auto res = pool.Map(
        [ ] (string f) { return myMacro(“opt”, 12, f); },
        { “file1”, “file2”, “file3” }
        )
```

https://github/bluehood/root

Map speed-up


This is the speed-up on **filling an histogram with 1E9 random numbers**
N.B.
reduce overhead
is not shown
speed-up
n workers
0
5
10
15
20
25

MapReduce speed-up