

# Smoothing Data Movement Between Memory and Storage for Reverse Time Migration

Tariq Alturkestani

PhD advisor: David Keyes

King Abdullah University of Science and Technology

June 17, 2019

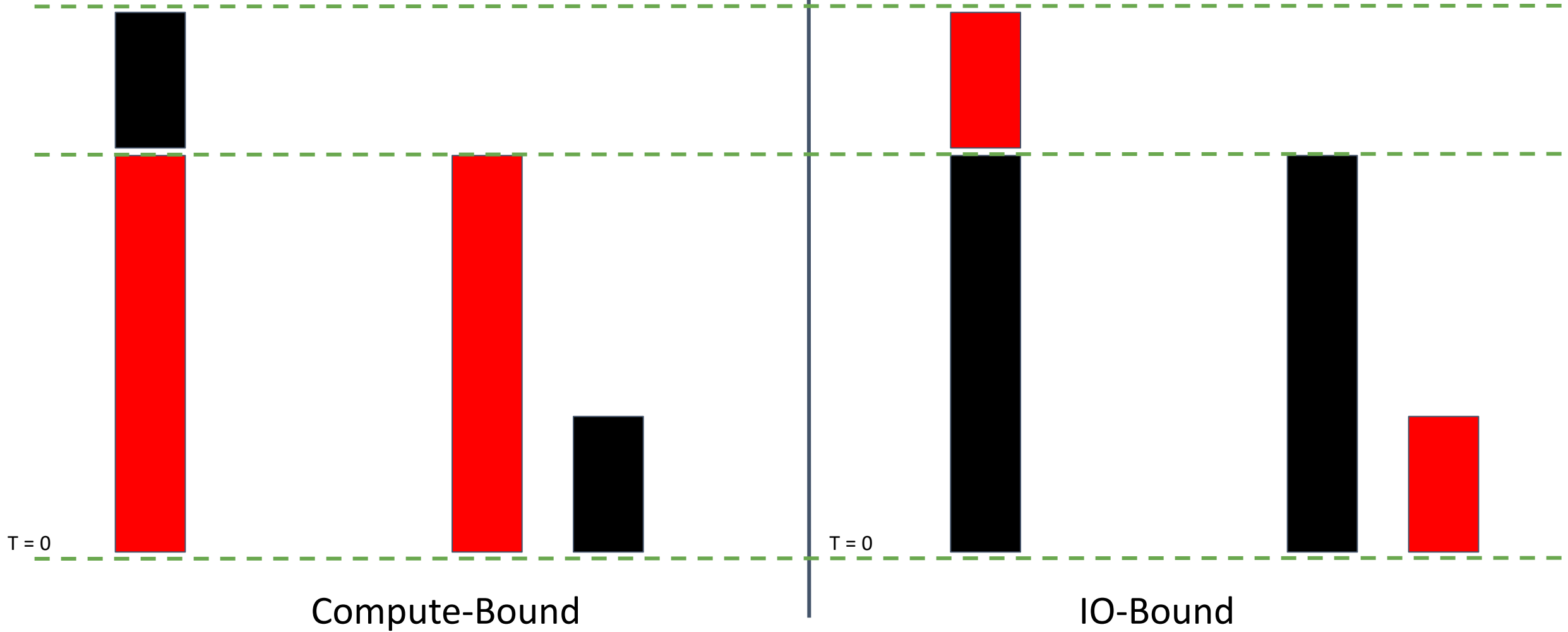
# What is being overlapped?



Overlapped



Overlapped



$$\text{Max Speedup} = (\text{Kernel time} + \text{IO time}) / \max(\text{Kernel time}, \text{IO time})$$

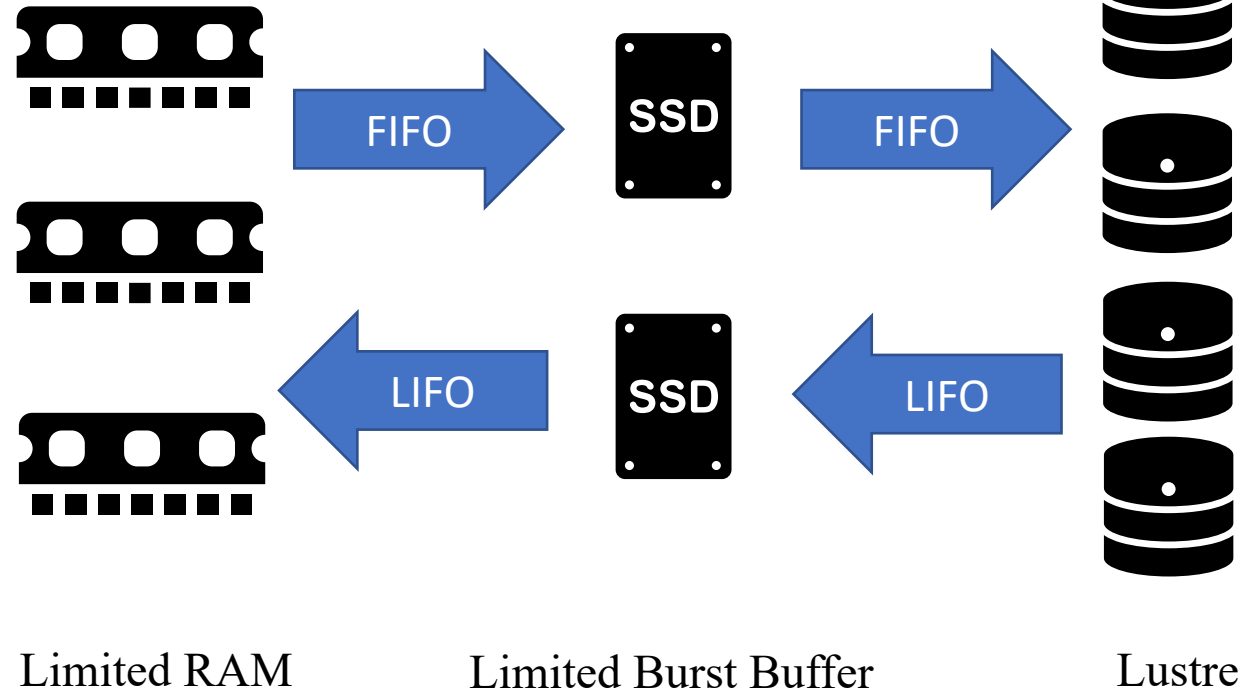
Legends: Compute IO

# Proposed Solution: Multilayer Buffer System (MLBS)

Application Space

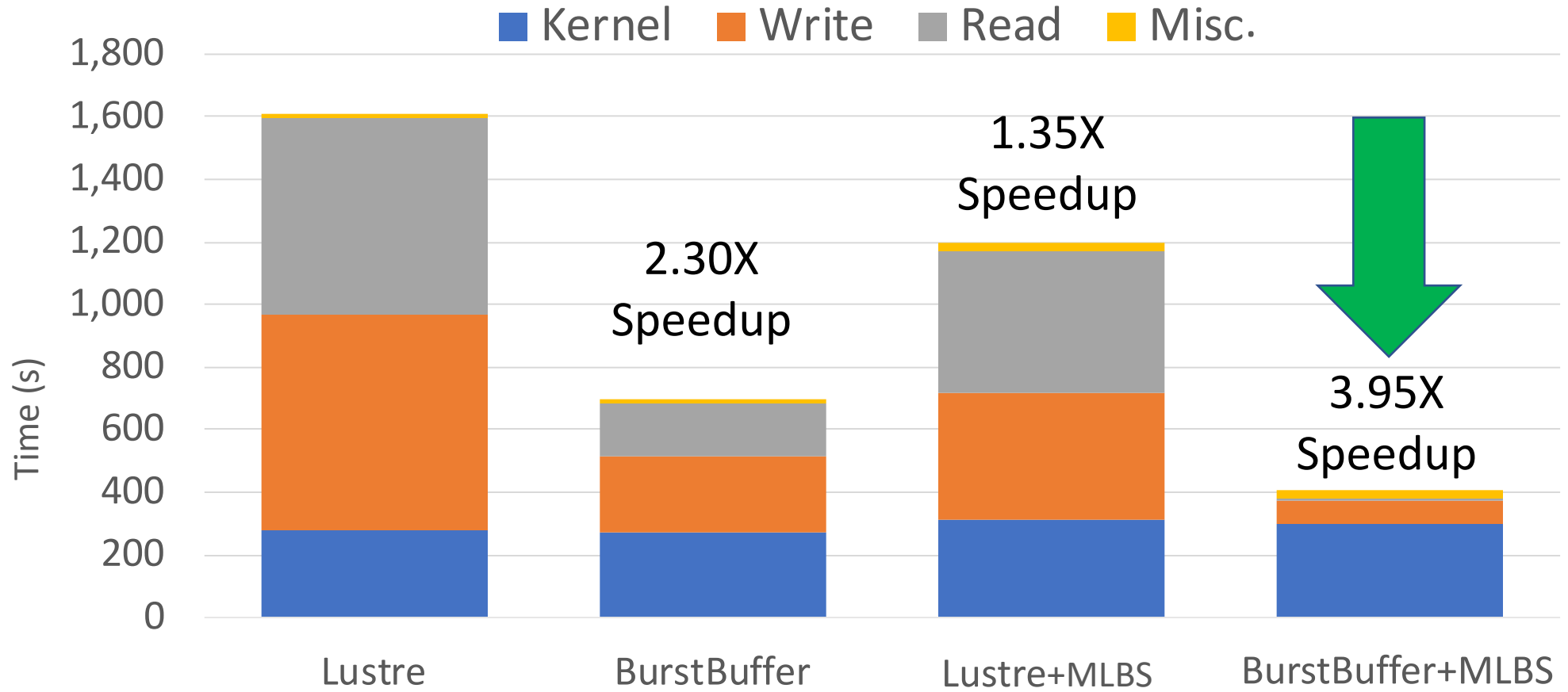
```
function adjoint():  
  for (i=0 -> N)  
    compute()  
    MLBS.write()  
  
  for (i=0 -> N)  
    MLBS.read()  
    compute()
```

MLBS Helper Engine

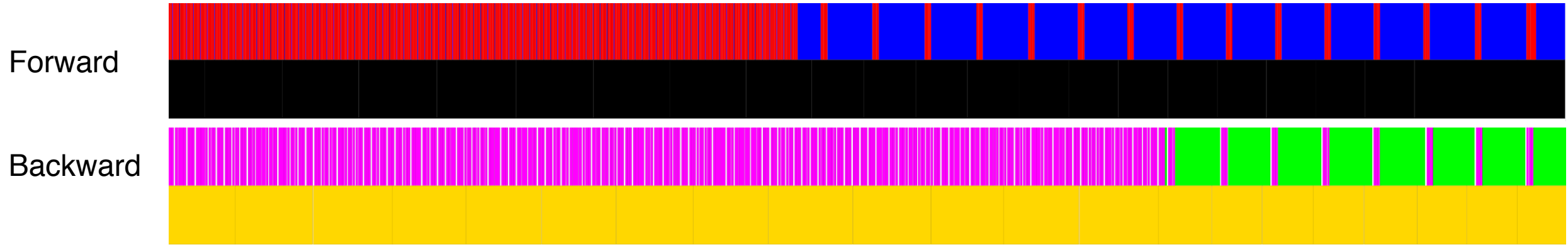


- Fast parallel copy
- Pointer swap

Without MLBS,  
the RTM app  
spends **%82**  
of its wall time  
on **I/O**



I/O Traces



# PhD Road Map

Extend:

MLBS framework to include storage objects

MLBStore to support domain decomposition and parallel I/O

MLBStore to asynchronously overlap I/O and compute

MLBS to support any out-of-core application

