OAK RIDGE National Laboratory

GPU Kernel Performance Deep Dive

Youngsung Kim and Sarat Sreepathi

Oct. 29, 2020

2020 ESMD-E3SM PI Meeting

ORNL is managed by UT-Battelle, LLC
for the US Department of Energy

U.S. DEPARTMENT OF ENERGY

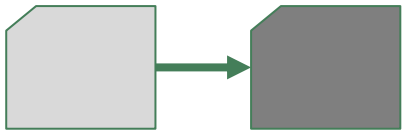
The Key Activities in Performance Optimizations

Static & Dynamic Analyses

Modifications

"Why is this slow(fast)?"

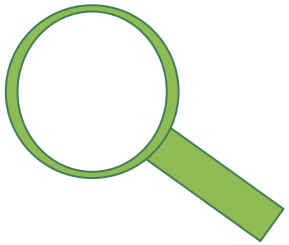
"I know how to fix this"



The Key Activities in Performance Optimizations

Static & Dynamic Analyses

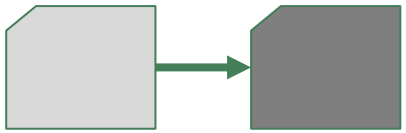
"Why is this slow(fast)?"



- H/W architecture
- Interactions between H/W and S/W
- Profiling tools, compiling tools

Modifications

"I know how to fix this"



- Algorithms and domain knowledge
- Interactions between S/W and H/W
- Compiling tools, testing

(Example)

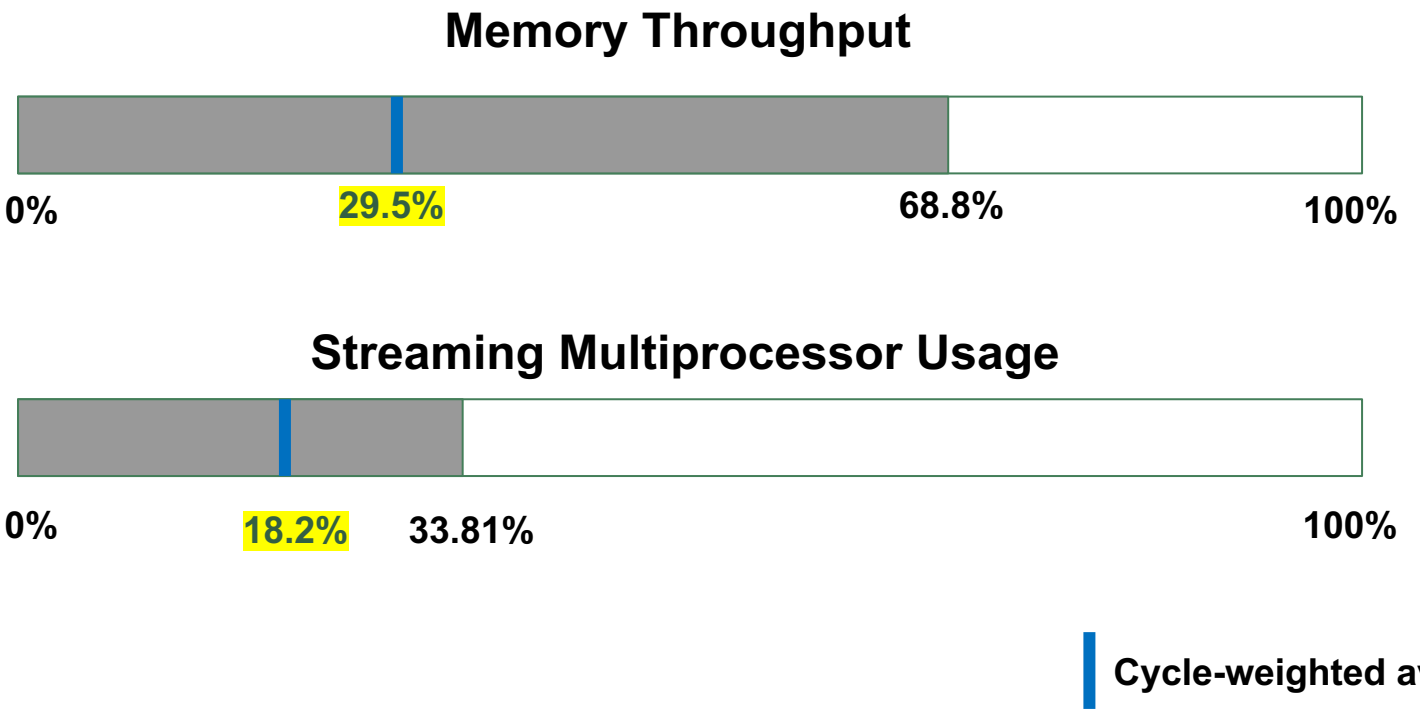
Analysis : too high L1 cache misses ⟺ Modification : Let's try tiling technique

Performance Optimization Setup

- Target Application: **SAM++**
  - A GPU port in C++ of System for Atmospheric Modeling (SAM) using a GPU porting framework, **YAKL**
  - Solve a 2-D or 3-D CRM(Cloud Resolving Model)
  - More computationally expensive than traditional code
  - 409 columns on a GPU
- Test System
  - H/W: Oak Ridge National Lab, OLCF Summit
    Nvidia Voltas, 80 SMs/GPU, 16GiB HBM,
  - S/W: gcc/6.4.0 cmake/3.17.3 cuda/11.0.2 netcdf/4.6.2
- GPU Performance Profilers
  - **Nsight-systems**: System level kernel launch analysis
  - **Nsight-compute**: Kernel level performance analysis

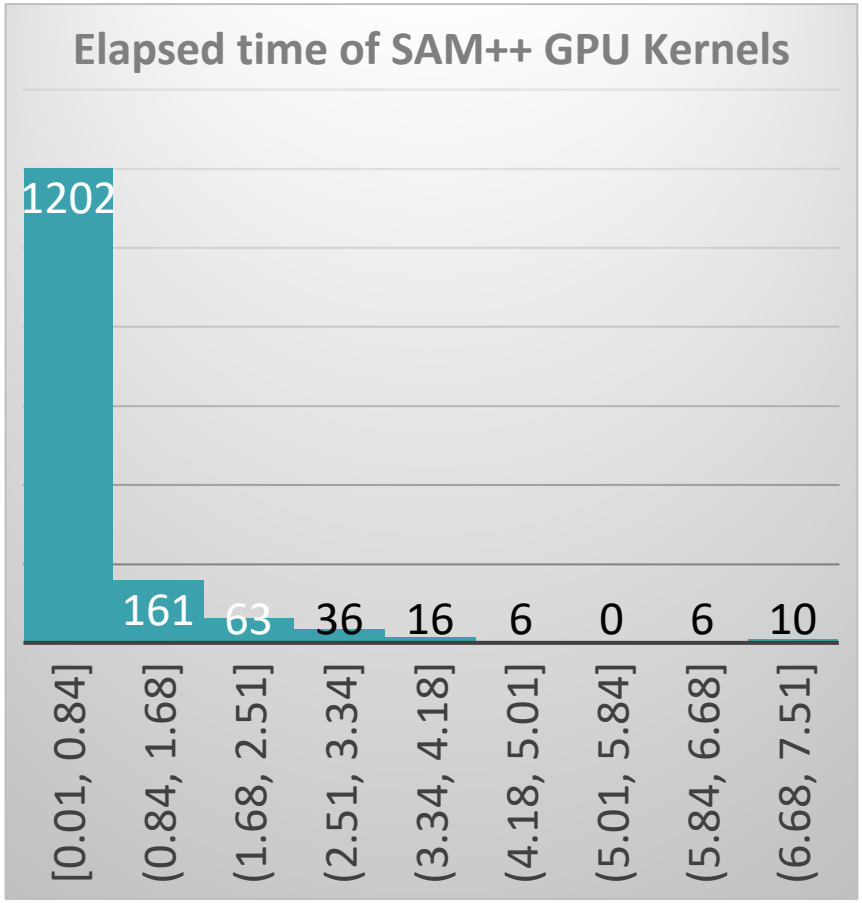
Performance Analysis - Overview

- Code size : 9,758 code lines without comments
- Profiling limited to first 1,500 launches of cpp2d

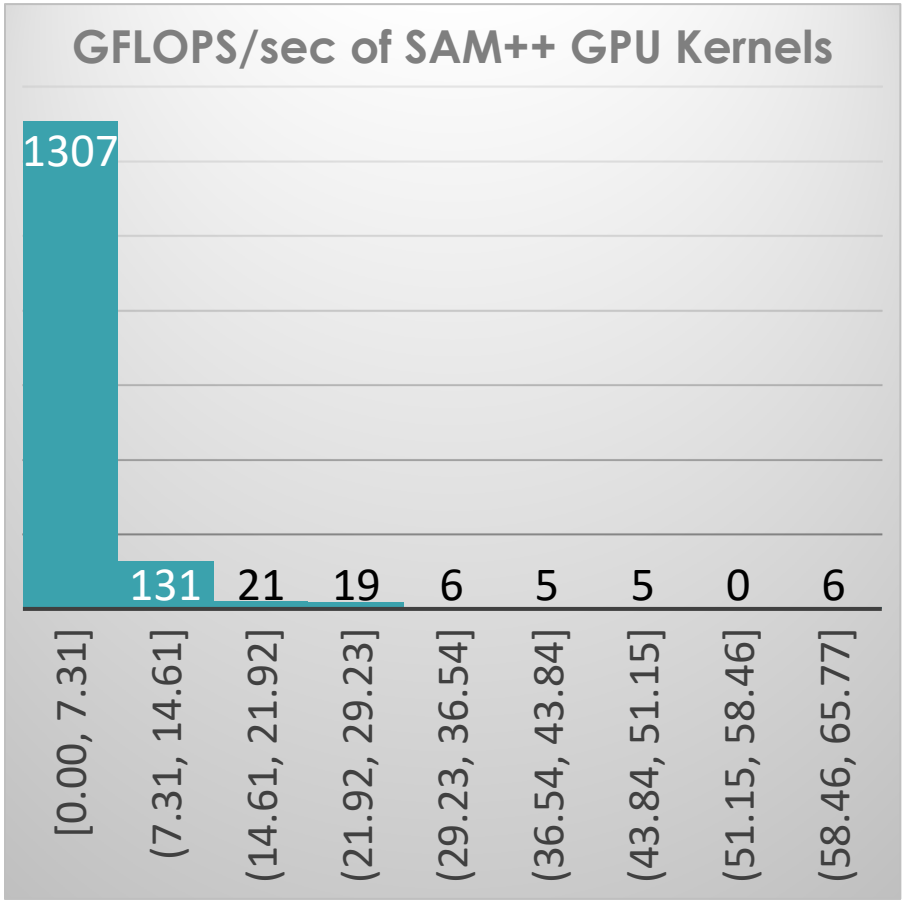


Performance Analysis – Computations

How long do the kernels run?



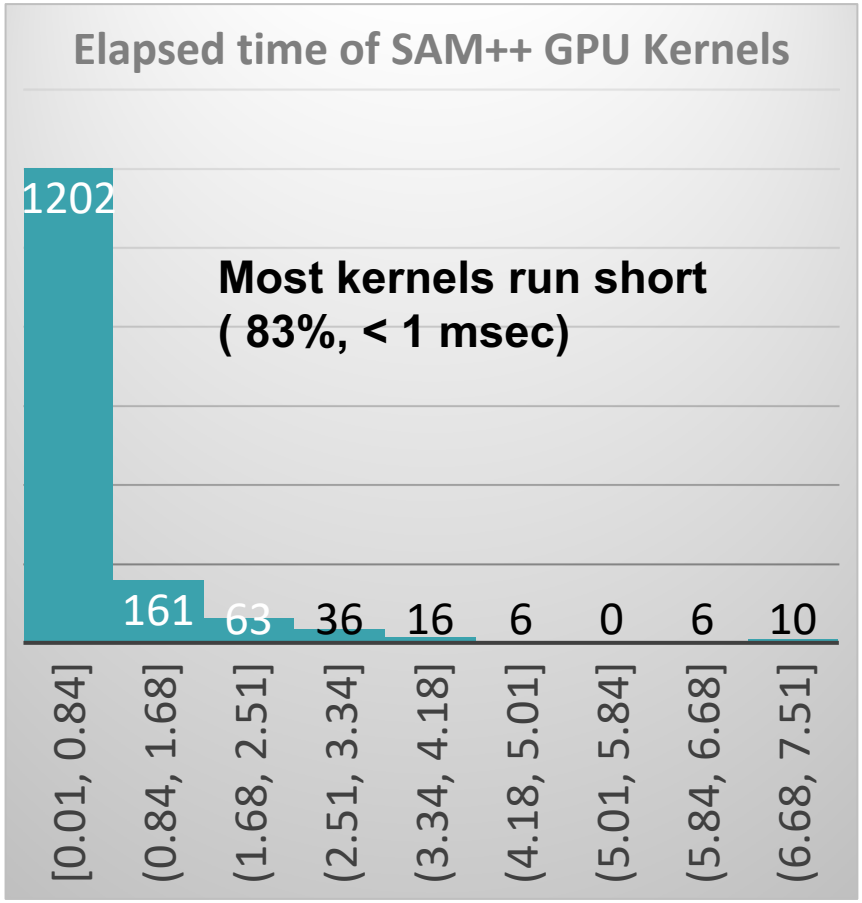
How fast(FLOPs) do the kernels run?



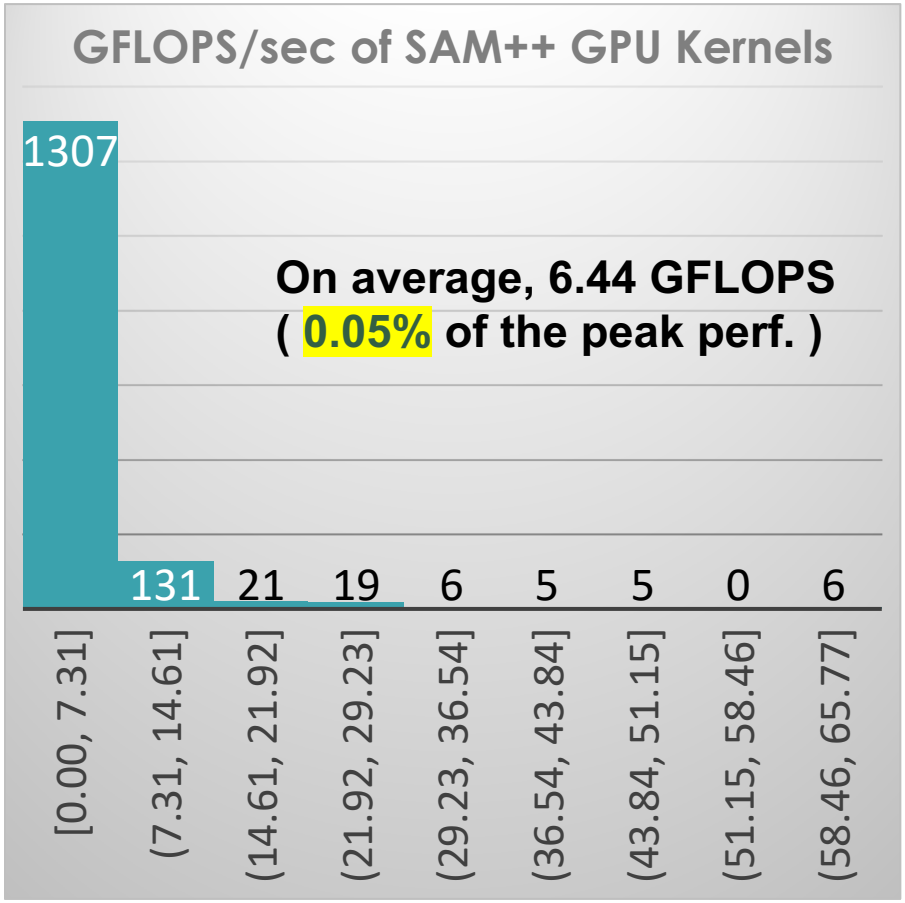
OAK RIDGE National Laboratory

Performance Analysis – Computations (FLOPS)

How long do the kernels run?



How fast(FLOPs) do the kernels run?



Performance Analysis - Key findings

Register usage close to maximum limit

255 used out of 256

Register Data movement and **integer** instructions are dominant

MOV : 67%
IADD3 : 13%
DADD : 1%
Total: 268,574,707 instructions

Usage of registers are increased rapidly on accessing **elements in array**

If (qn(k,j,i,icrm)…)

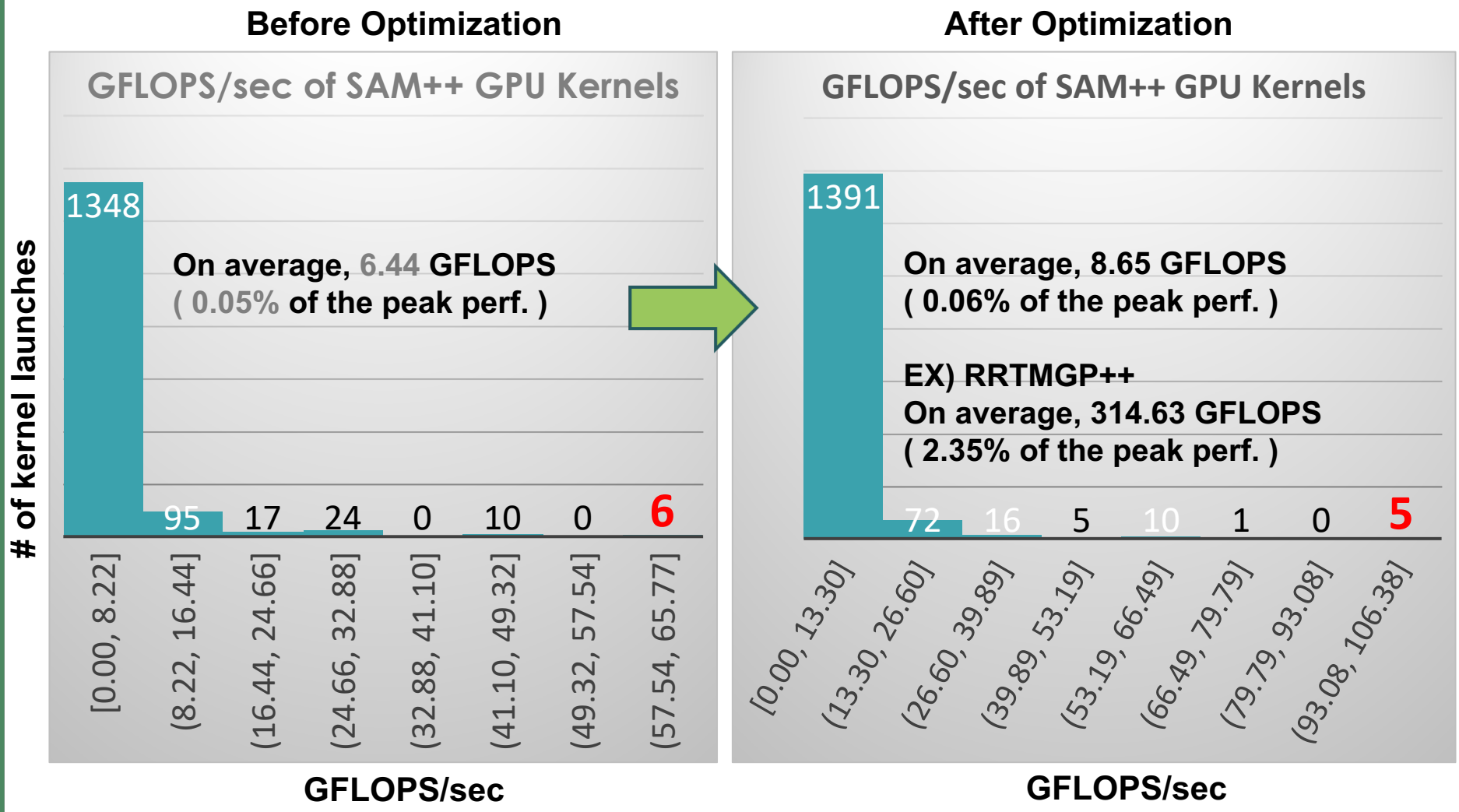
Strongly indicates that "register spill" problem exists due to array index calculations

Using slower device global memory instead of faster registers

Code modifications based on the key findings

- YAKL is a framework used in SAM++ for GPU porting. Array index calculations are done inside of YAKL
- The core developer (Matthew Norman) of YAKL made following modifications
  - switch to **unsigned int** instead of **size_t**
  - change the looping strategy to use **cheaper integer modulo**
  - create a **SimpleBounds class** that uses fewer registers for loops

Speed-ups – Computations (FLOPS)



Speed-ups

Register usage

255 => 255

Executed instructions

MOV : 67% => 55%
IADD3 : 13% => 16%
DADD : 1% => 1%
Total 268,574,707 => 197,559,738(-25%) instructions

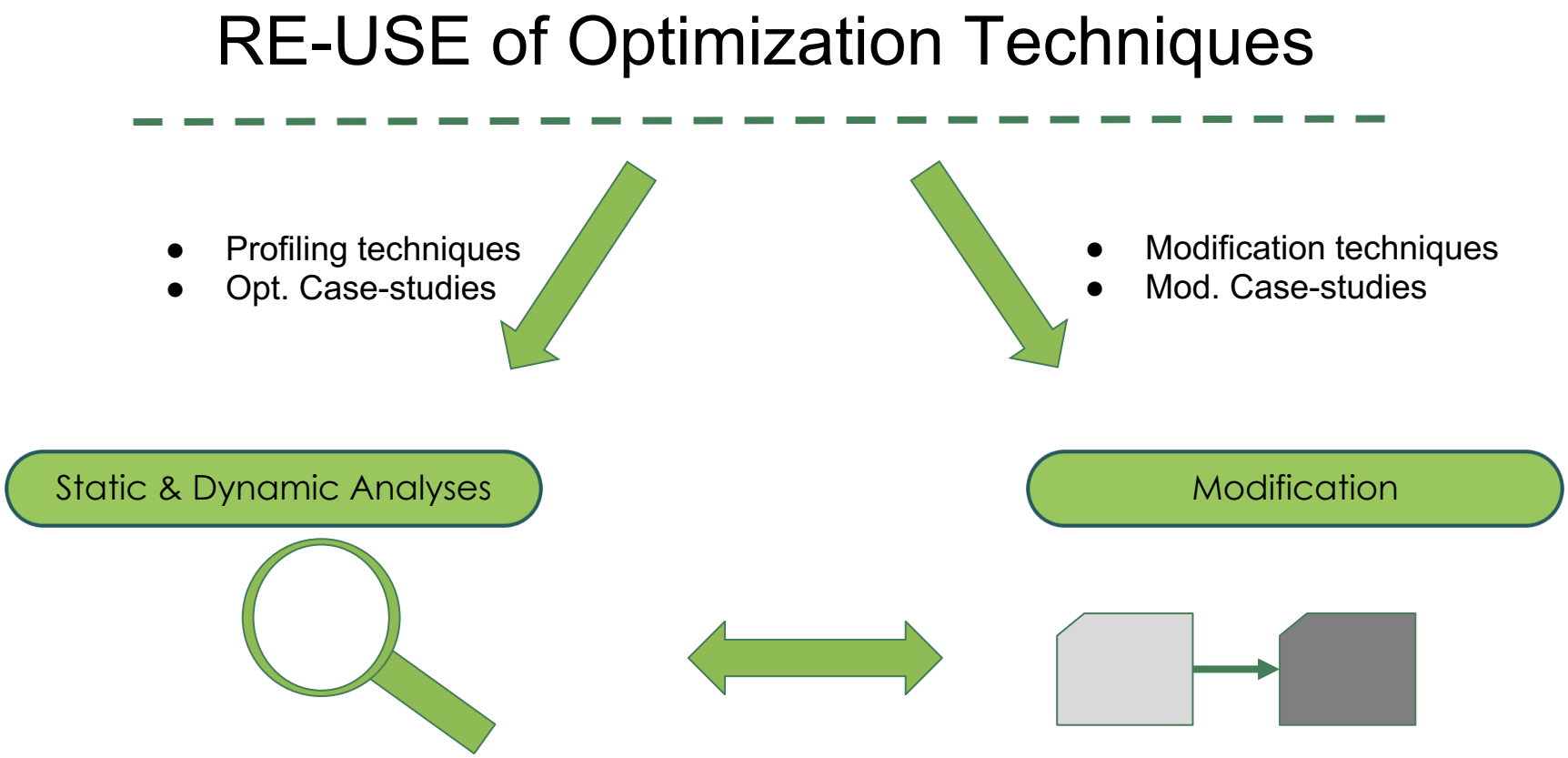
elements in array

If (qn(k,j,i,icrm) + qp(ind_qp, k…))

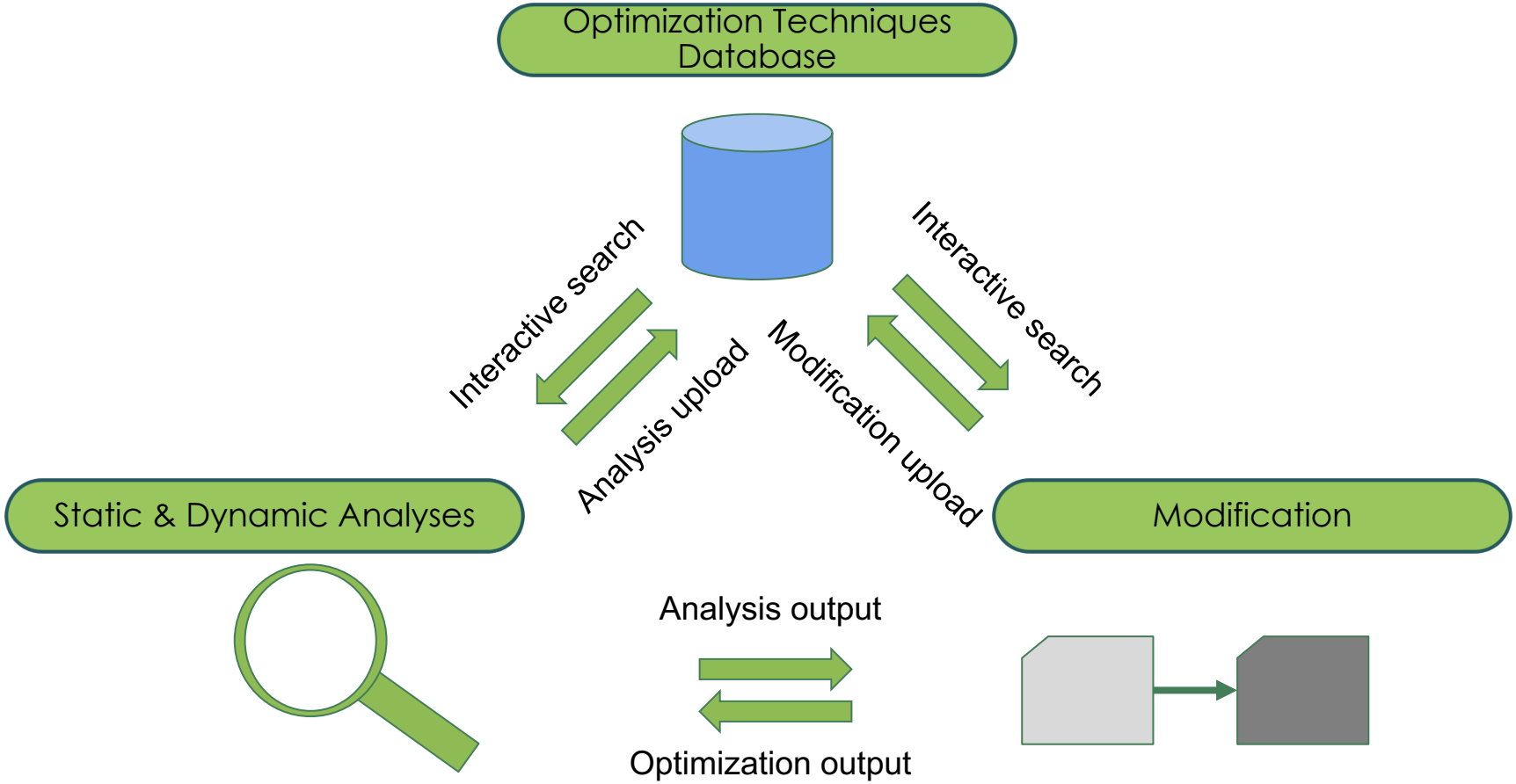
Reduced MOV and IADD3 instructions

Total run time: 9.16 sec => 6.18 sec, 1.5X speed-up

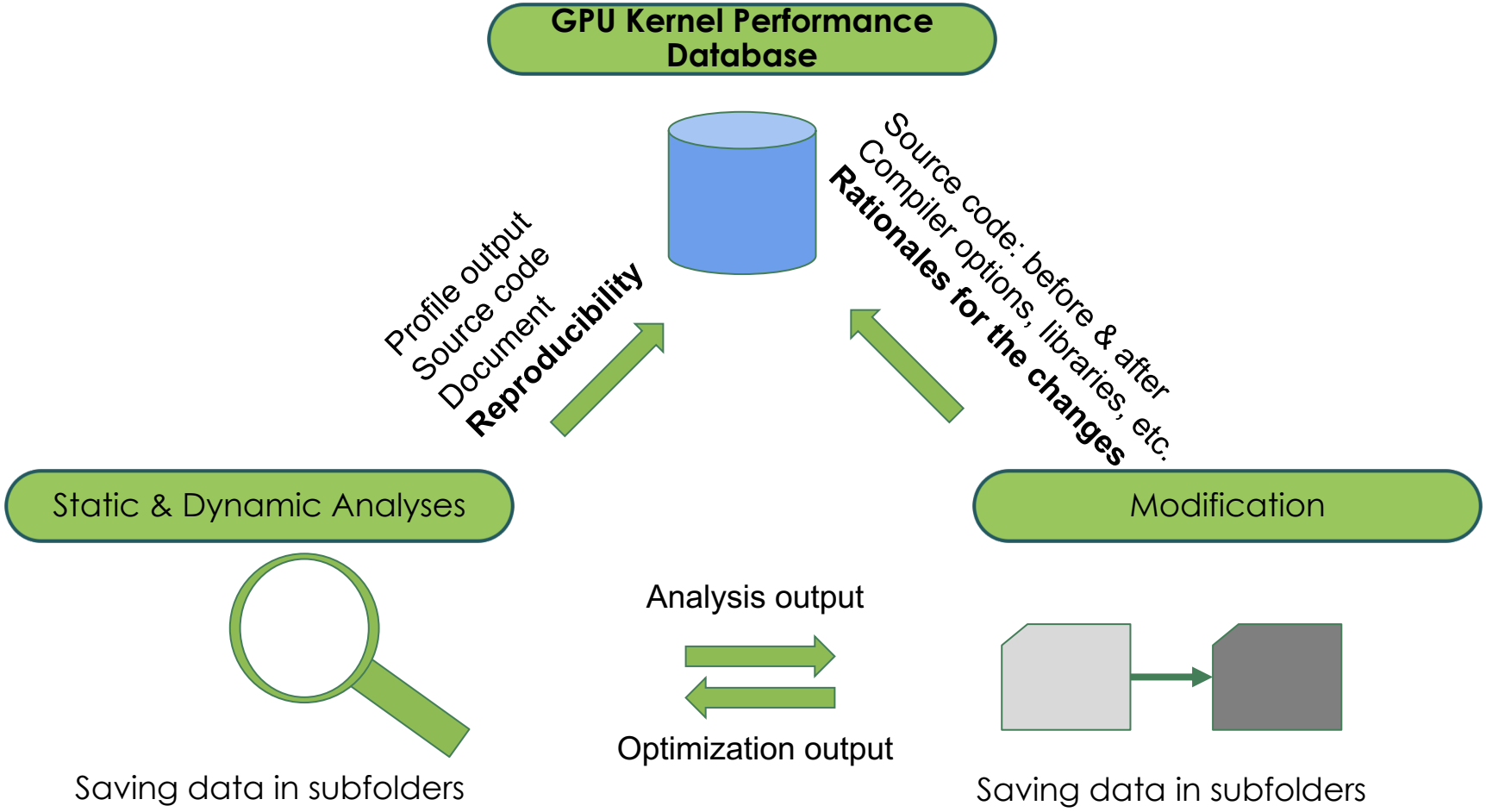
The Key Activities in Performance Optimizations - Revisited



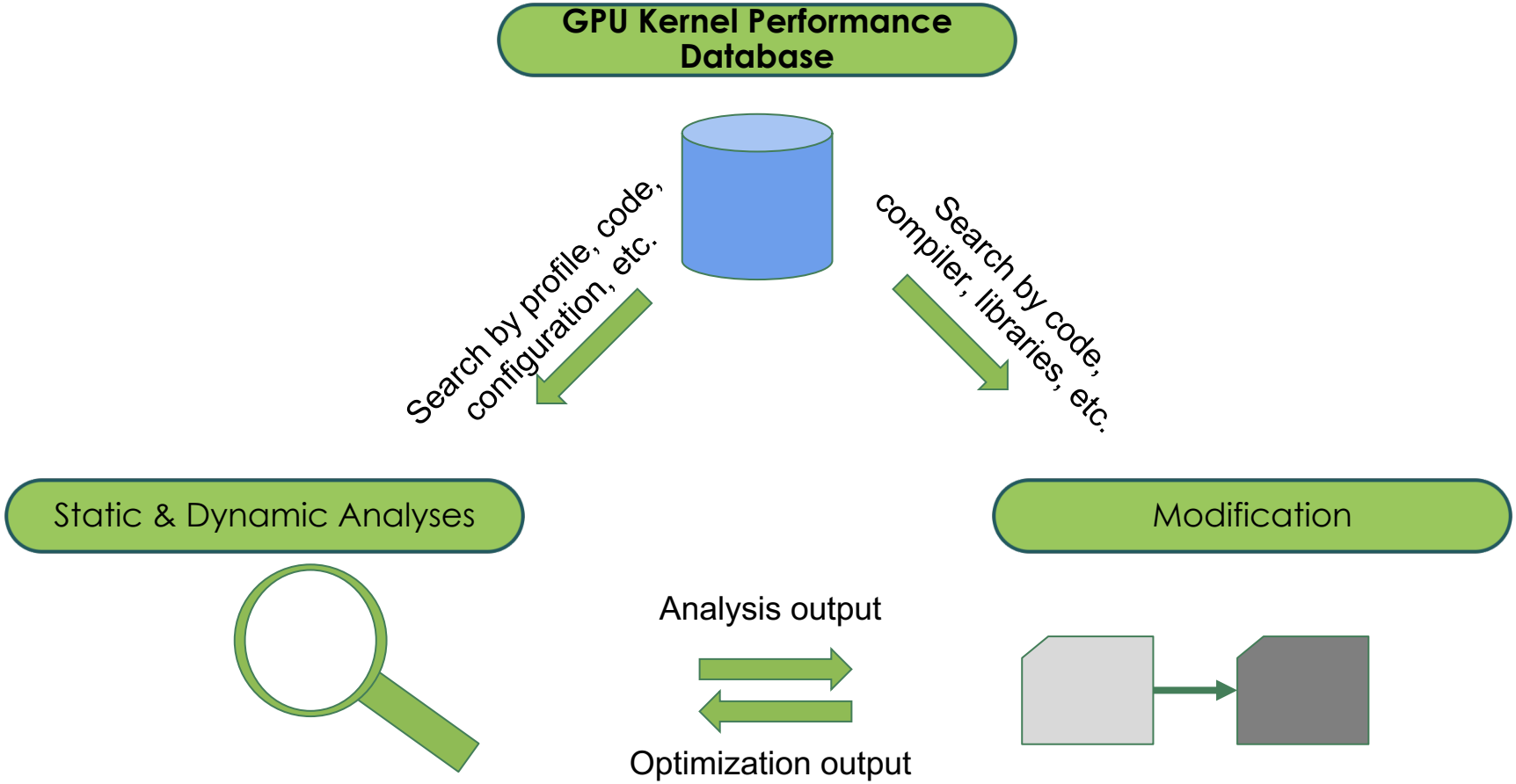
Scaling Up Performance Optimization



Scaling Up Performance Optimization - Through GPU Kernels



Scaling Up Performance Optimization - Through GPU Kernels



Conclusions

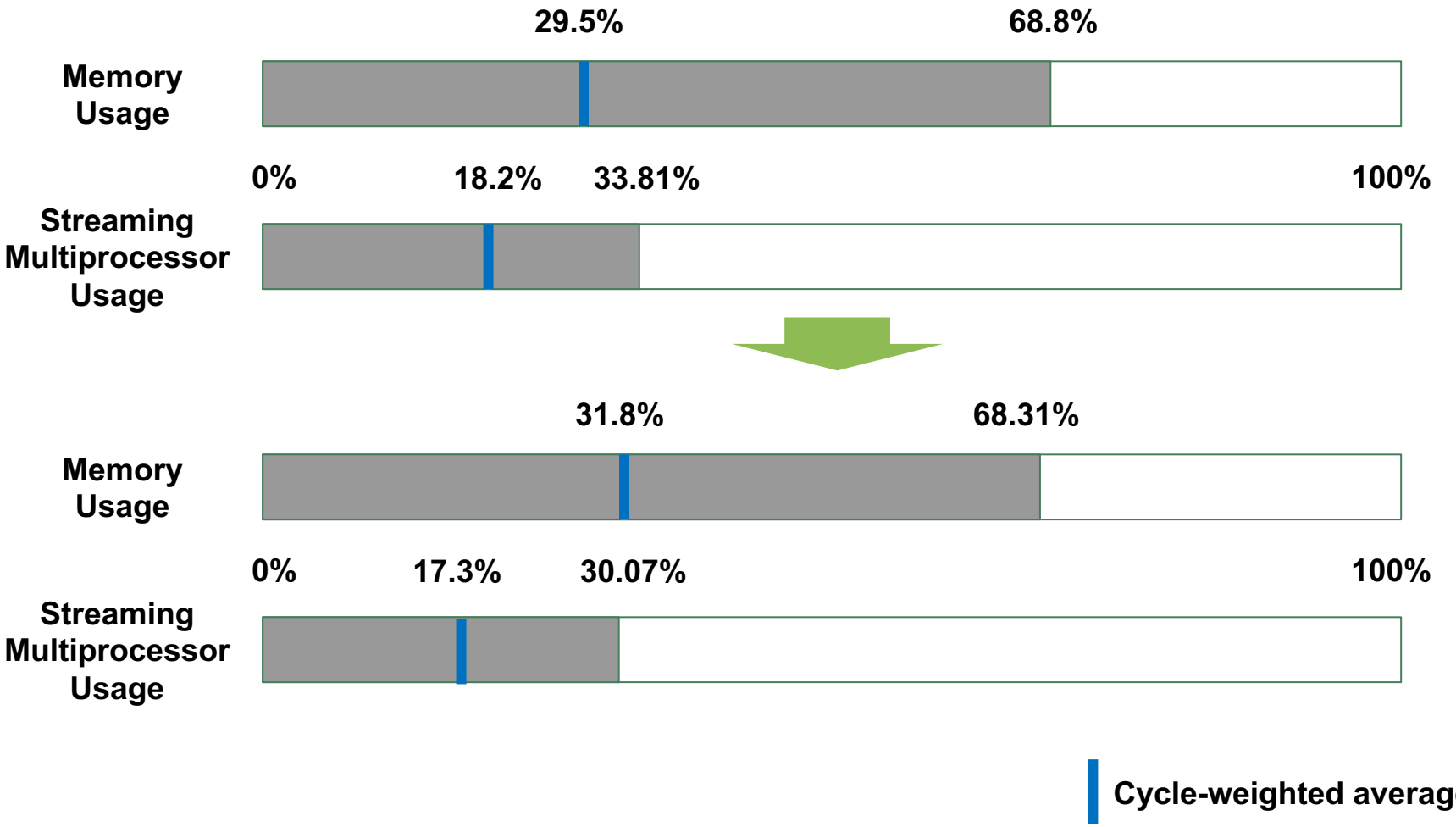
- GPU Kernel performances of SAM++ are analyzed using Nvidia Nsight Profilers
- From the analyses, "register spill" issue is identified, and YAKL is updated to reduce register usage
- Performance analysis requires different skill set from actual modification, but they are inter-dependent
- Future: prototype "GPU Kernel Performance Database" for scaling up performance optimization

Q&A

THANK YOU

kimy@ornl.gov

Speed ups - GPU Usage



Analysis Upload Subfolders

Folder name	Content
Goals	GFLOPS, Elapsed time, Memory Usage, I/O, …
Setup	Reproducibility: script, code, system, compiler, …
Measurement	performance metrics
Analysis	A tightly related subset of performance metrics with causal relationship to the goal(s)
Conclusions	root cause of the performance
Verification	check if this analysis is useful

Modification Upload Subfolders

Folder name	Content
Analysis	Corresponding performance analysis
Setup	Reproducibility: code, system, compiler, …
Changes	Code, compiler, library: before & after
Reasoning	Reasons of the changes
Speed-up	Performance change

Original

- ==155521== NVPROF is profiling process 155521, command: ./cpp2d
- File : input.nc
- Samples: 409
- crm_nx : 32
- crm_ny : 1
- crm_dx : 1000.0000000000000
- crm_dt : 5.0000000000000000
- plev : 30
- Reading the data
- Running the CRM
- Elapsed Time: 9.160443301999991
- Writing output data
- ==155521== Profiling application: ./cpp2d

Optimized

- ==57325== NVPROF is profiling process 57325, command: ./cpp2d
- File : input.nc
- Samples: 409
- crm_nx : 32
- crm_ny : 1
- crm_dx : 1000.0000000000000
- crm_dt : 5.0000000000000000
- plev : 30
- Reading the data
- Running the CRM
- Elapsed Time: 6.178564213999997
- Writing output data
- ==57325== Profiling application: ./cpp2d