

Deep Learning and GPU Programming Workshop

12 – 15 July 2021

MODULE ONE: INTRODUCTION

Dr. Volker Weinberg | LRZ | 12.07.2021





MODULE OVERVIEW

Topics to be covered

- Introduction to parallel programming
- Common difficulties in parallel programming
- Introduction to OpenACC
- Parallel programming in OpenACC



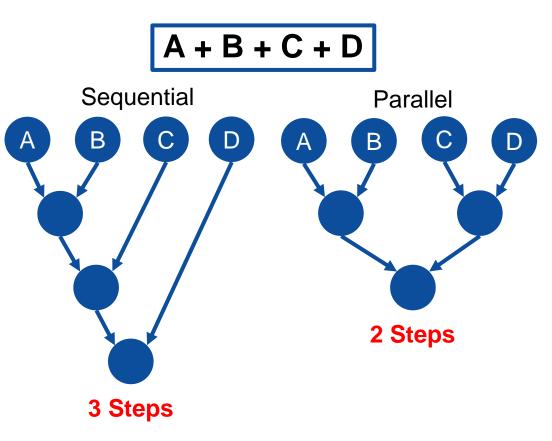
INTRODUCTION TO PARALLEL PROGRAMMING



WHAT IS PARALLEL PROGRAMMING?

"Performance Programming"

- Parallel programming involves exposing an algorithm's ability to execute in parallel
- This may involve breaking a large operation into smaller tasks (task parallelism)
- Or doing the same operation on multiple data elements (data parallelism)
- Parallel execution enables better performance on modern hardware





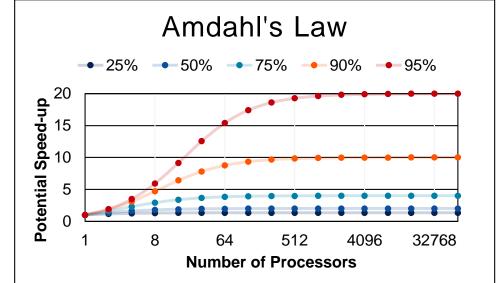
AMDAHL'S LAW



AMDAHL'S LAW

Serialization Limits Performance

- Amdahl's law is an observation that how much speed-up you get from parallelizing the code is limited by the remaining serial part.
- Any remaining serial code will reduce the possible speed-up
- This is why it's important to focus on parallelizing the most time consuming parts, not just the easiest.





APPLYING AMDAHL'S LAW

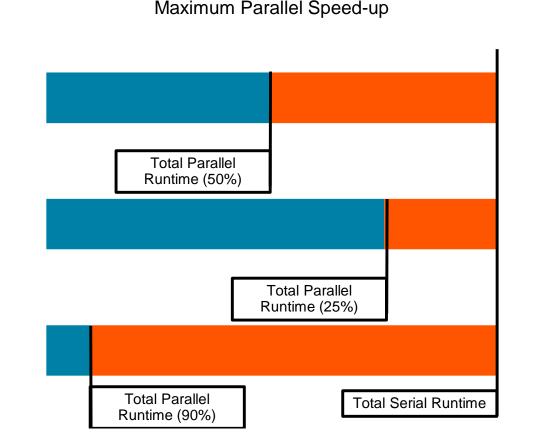
Estimating Potential Speed-up

- What's the maximum speed-up that can be obtained by parallelizing 50% of the code?
- 1 / (100% 50%) = 1 / (1.0 0.50) = 2.0X
- What's the maximum speed-up that can be obtained by parallelizing 25% of the code?

1 / (100% - 25%) = 1 / (1.0 - 0.25) = 1.3X

What's the maximum speed-up that can be obtained by parallelizing 90% of the code?

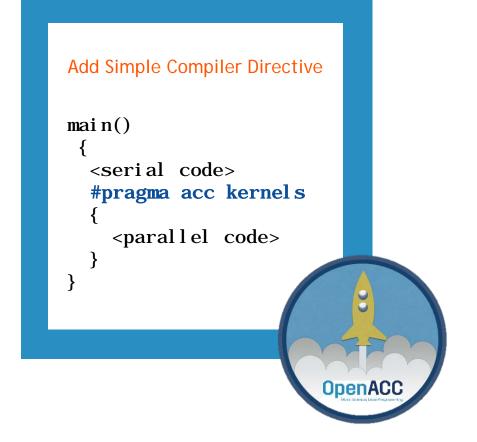
1 / (100% - 90%) = 1 / (1.0 - 0.90) = 10.0X



INTRODUCTION TO OPENACC



OpenACC is a directivesbased programming approach to parallel computing designed for performance and portability on CPUs and GPUs for HPC.





STANDARDS-BASED PARALLELISM

MPI standard **OpenACC** standard **OpenMP** standard The OpenACC^a **OpenMP** Application Programming Interface MPI: A Message-Passing Interface Standard Version 2.7 OpenACC-Standard.on **Application Programming Interface** November, 2018 **Specification Version 5.0** Edited by Michael Klemm and Bronis R. de Supinski OpenMP Architecture Review Board | openmp.org

https://www.mpi-forum.org/docs/

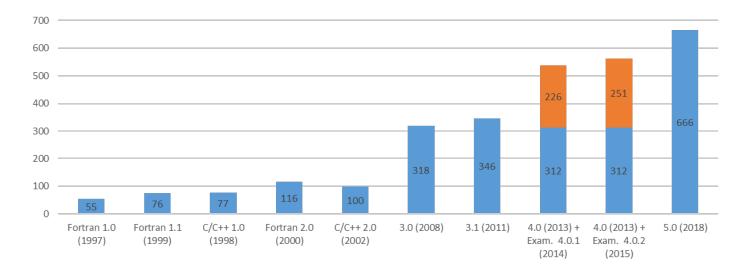


https://www.openmp.org/specifications/

https://www.openacc.org/specification

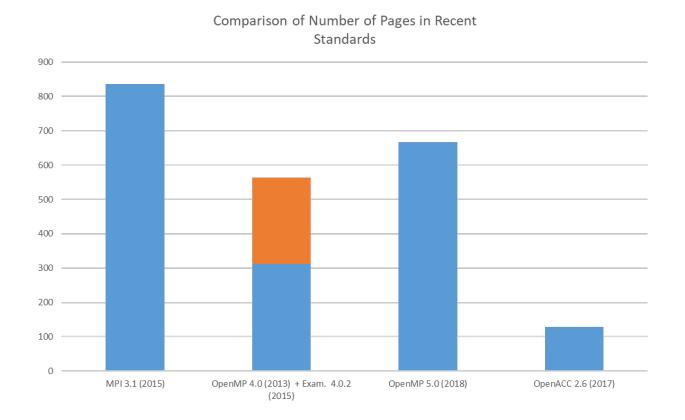
DEVELOPMENT OF OPENMP STANDARD

Number of Pages in OpenMP Standard



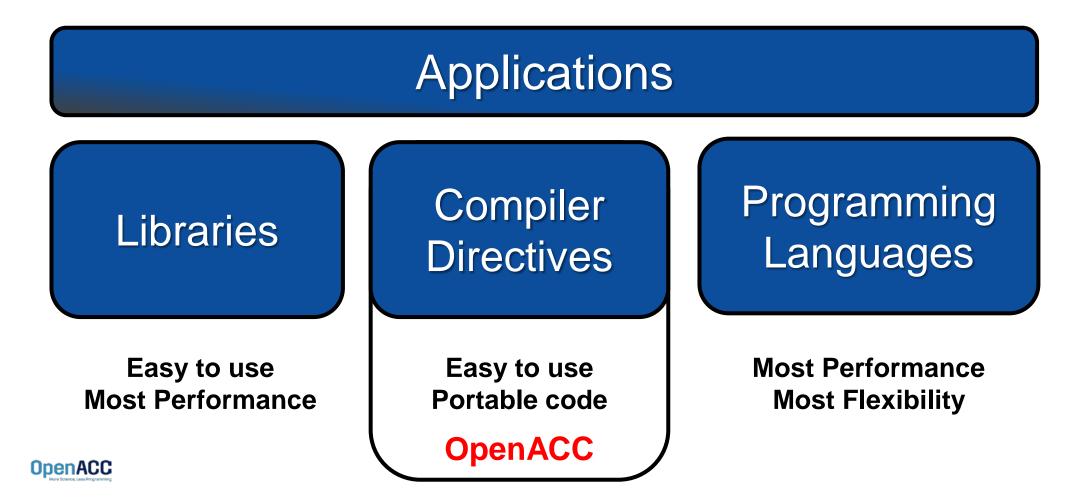


COMPLEXITY OF RECENT STANDARDS





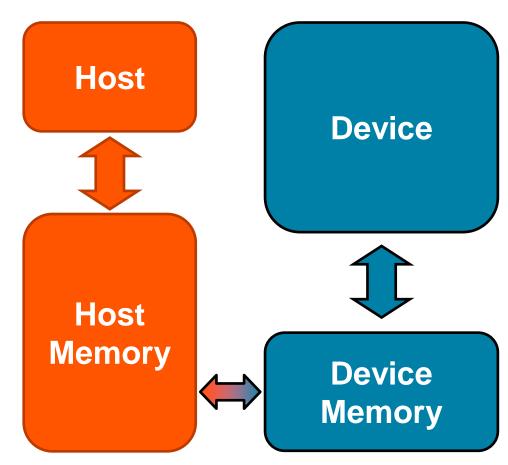
3 WAYS TO ACCELERATE APPLICATIONS



OPENACC PORTABILITY

Describing a generic parallel machine

- OpenACC is designed to be portable to many existing and future parallel platforms
- The programmer need not think about specific hardware details, but rather express the parallelism in generic terms
- An OpenACC program runs on a host (typically a CPU) that manages one or more parallel devices (GPUs, etc.). The host and device(s) are logically thought of as having separate memories.





OPENACC Three major strengths

Incremental	Single Source	Low Learning Curve



Incremental

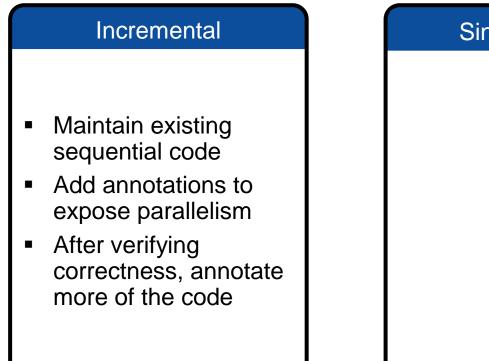
- Maintain existing sequential code
- Add annotations to expose parallelism
- After verifying correctness, annotate more of the code

```
Begin with a working
                                        sequential code.
   Enhance Sequential Code
#pragma acc parallel loop
for(i = 0; i < N; i++)
{
    < loop code >
                                  Parallelize it with OpenACC.
}
#pragma acc parallel loop
for(i = 0; i < N; i++)
{
                                    Rerun the code to verify
    < loop code >
                                        correct behavior,
                                    remove/alter OpenACC
```

code as needed.



OpenACC



Low Learning Curve Single Source

Supported Platforms POWFR Sunway x86 CPU x86 Xeon Phi **NVIDIA GPU** PEZY-SC

Single Source Rebuild the same code on multiple architectures Compiler determines how to parallelize for the desired machine Sequential code is maintained

The compiler can **ignore** your OpenACC code additions, so the same code can be used for **parallel** or **sequential** execution.



Incremental

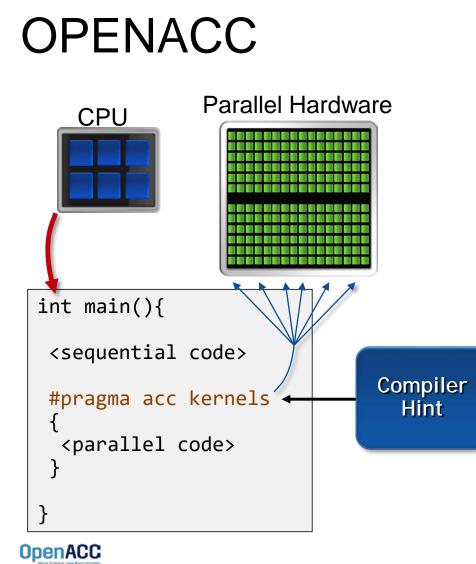
- Maintain existing sequential code
- Add annotations to expose parallelism
- After verifying correctness, annotate more of the code

Single Source

- Rebuild the same code on multiple architectures
- Compiler determines how to parallelize for the desired machine
- Sequential code is maintained

Low Learning Curve





The programmer will give hints to the compiler about which parts of the code to parallelize.

The compiler will then generate parallelism for the target parallel hardware.

Low Learning Curve

- OpenACC is meant to be easy to use, and easy to learn
- Programmer remains in familiar C, C++, or Fortran
- No reason to learn low-level details of the hardware.

Incremental

- Maintain existing sequential code
- Add annotations to expose parallelism
- After verifying correctness, annotate more of the code

Single Source

- Rebuild the same code on multiple architectures
- Compiler determines how to parallelize for the desired machine
- Sequential code is maintained

Low Learning Curve

- OpenACC is meant to be easy to use, and easy to learn
- Programmer remains in familiar C, C++, or Fortran
- No reason to learn low-level details of the hardware.



EXPRESSING PARALLELISM WITH OPENACC

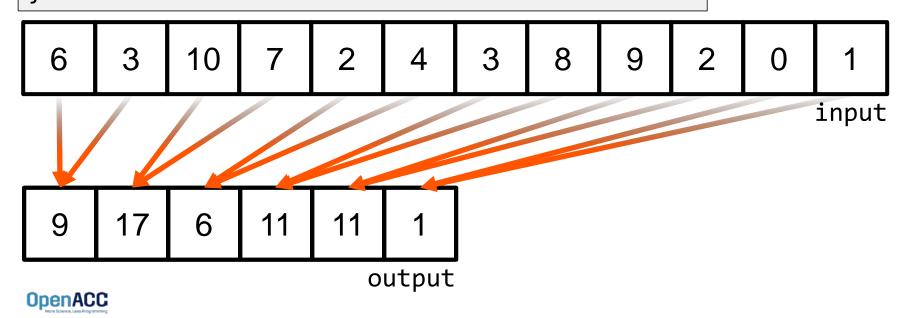


CODING WITH OPENACC

Array pairing example- serial

void pairing(int *input, int *output, int N){

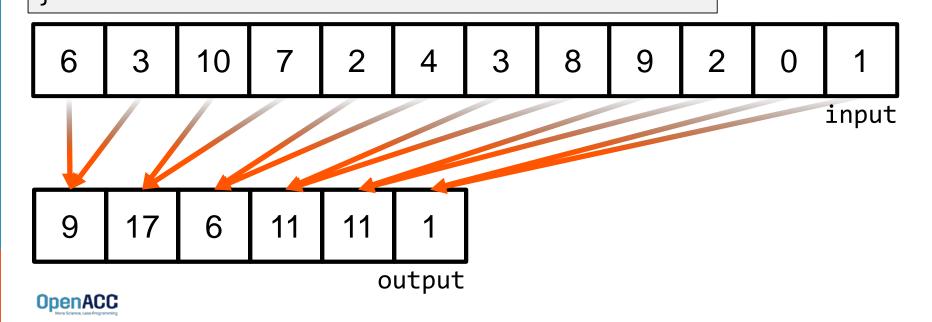
```
for(int i = 0; i < N; i++)
    output[i] = input[i*2] + input[i*2+1];</pre>
```



CODING WITH OPENACC

Array pairing example - parallel

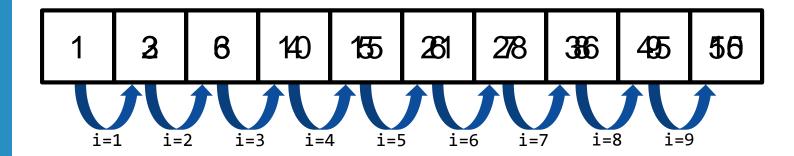
```
void pairing(int *input, int *output, int N){
    #pragma acc parallel loop
    for(int i = 0; i < N; i++)
        output[i] = input[i*2] + input[i*2+1];</pre>
```



DATA DEPENDENCIES

Not all loops are parallel

void pairing(int *a, int N){



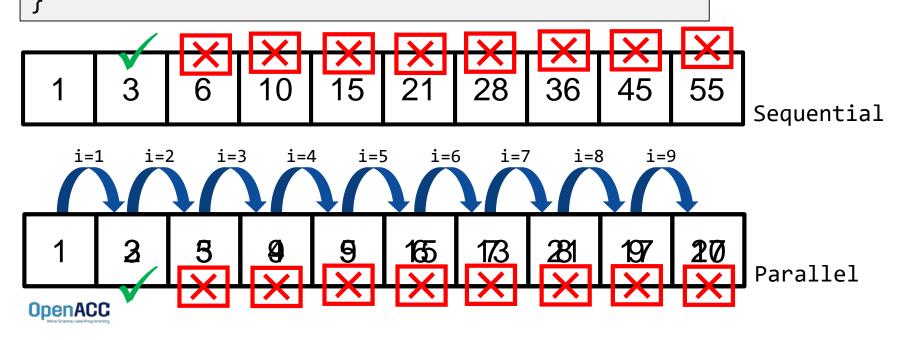


DATA DEPENDENCIES

Not all loops are parallel

void pairing(int *a, int N){
 #pragma acc parallel loop
 for(int i = 1; i < N; i++)
 a[i] = a[i] + a[i-1];
}</pre>

If we attempted to parallelize this loop we would get wrong answers due to a *forward dependency*.



MODULE 1 REVIEW



CLOSING SUMMARY Module One: Introduction

- Parallel programming is a technique of utilizing modern hardware to do lots of work all at once.
- Amdahl's law is the gravity of parallel programming, break this law at your own peril.
- Not all loops are parallel, but often can be rewritten to be parallelizable
- OpenACC is a high level model for generating parallel code from serial loops



OPENACC RESOURCES

Guides • Talks • Tutorials • Videos • Books • Spec • Code Samples • Teaching Materials • Events • Success Stories • Courses • Slack • Stack Overflow

Resources



🗱 slack

https://www.openacc.org/community#slack

https://www.openacc.org/resources		https://www.openaco
OpenACC Methodese (see Programme)	Search Q About Tools News Events Resources Spec Community	OpenACC
Resources A complete library of OpenACC materials that includes a collection of video tutorials, guides, online courses, books and more.		Success Stories Applications across multiple domains have been accelerated with Open are sharing their results and opperferces.
Cuides Introduction to OpenACC Quick Guides OpenACC Programming and Best Practices Guide OpenACC 235 API Reference Card	Books Parallel Programming with OpenACC	
Tutorials Video tutorials to help start with OpenACC and advance your skills	Programming Massively Parallel Processon, Third Edition: A Hands-on Approach	Manaraban se oning Grid and Quark t. B an church fair codei la faire data disen simulation an church fair codei la faire data disen simulation
Compilers	and Tools	Eve
https://www.or	penacc.org/tools	https://www.oper
OpenACC	About Tools News Events Resources Spec Community	OpenACC Marchana Carl Parentes
Downloads & Tools OperACC complies, profilers and debuggers are designed and available	to download from multiple vendors and academic organizations.	Events The OpenAct Community organizes a variety of events throughout the year, at conferences to workshops, hackathons, online courses and ther Group m around the world to learn OpenAct programming and to participate in activ User Group.
Contract Cry Inc for more information. Contact Cry Inc for more information. Contact Cry Inc for more information.	Open Source Compilers	Hackathons Hackathons are five day intensive hands on mentioning assions. They are designed to scientify a policitions to GPUs ang literarias, OpenACC, CUOA and other to testby the Calk Ridge Lakenoing Computing Facility (ICC/) at the Calk Ridge Nations the M literational and registration relates (searco with the Surgeona Attacous)

Success Stories

cc.org/success-stories





THANK YOU

