OPENACC: MORE SCIENCE LESS PROGRAMING

Sunita Chandrasekaran

Assistant Professor, Computer & Information Sciences

University of Delaware, Newark, DE, USA

Director of OpenACC User Adoption

June 07, 2018

1st HPC Summer School 2018

Universidad de los Andes, Bogotá, from June 5th to 9th

OpenACC
More Science, Less Programming

ACK: NVIDIA TEACHING KIT

OPENACC DIRECTIVES ARE DESIGNED FOR

1

Scientists to help do

more science and less programming

2

Performance portability on

CPUs, GPUs and other platforms

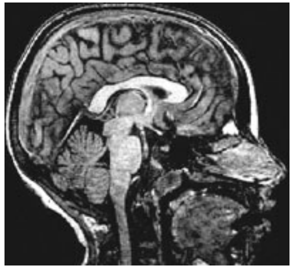
OpenACC

Simple | Powerful | Portable

Fueling the Next Wave of Scientific Discoveries in HPC

```
main()
{
  <serial code>
  #pragma acc kernels
  //automatically runs on GPU
  {
    <parallel code>
  }
}
```

University of Illinois
PowerGrid- MRI Reconstruction



70x Speed-Up
2 Days of Effort

RIKEN Japan
NICAM- Climate Modeling

7-8x Speed-Up
5% of Code Modified

8000+

Developers

using OpenACC

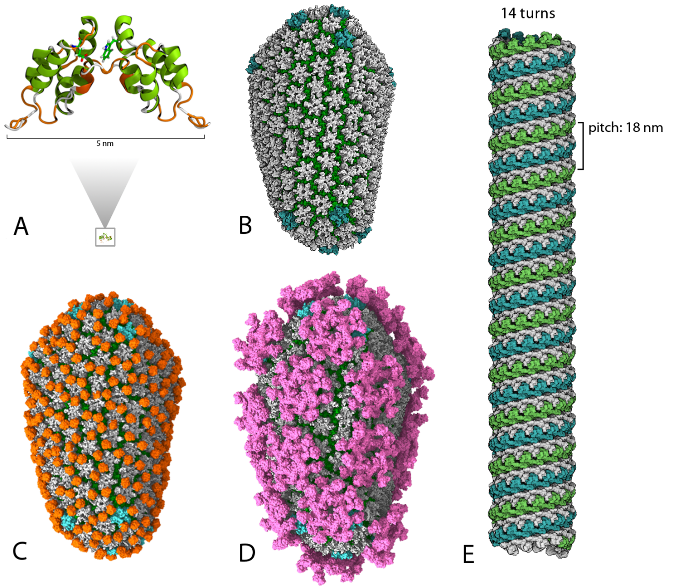
http://www.cray.com/sites/default/files/resources/OpenACC_213462.12_OpenACC_Cosmo_CS_FNL.pdf
http://www.hpcwire.com/off-the-wire/first-round-of-2015-hackathons-gets-underway
http://on-demand.gputechconf.com/gtc/2015/presentation/S5297-Hisashi-Yashiro.pdf
http://www.openacc.org/content/experiences-porting-molecular-dynamics-code-gpus-cray-xk7

OpenACC

Simple | Powerful | Portable

Fueling the Next Wave of Scientific Discoveries in HPC

```
main()
{
  <serial code>
  #pragma acc kernels
  //automatically runs on GPU
  {
    <parallel code>
  }
}
```



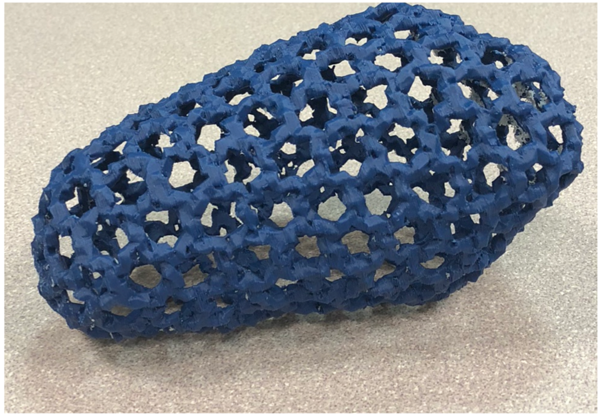
University of Delaware

Accelerating Chemical Shift of Protein Structures

67x Speed-Up

14 months effort

Undergraduate students with minimum parallel programming background



OPENACC DIRECTIVES

Manage Data Movement

Initiate Parallel Execution

Optimize Loop Mappings

```
#pragma acc data copyin(a,b) copyout(c)
{
  ...
  #pragma acc parallel
  {
  #pragma acc loop gang vector
      for (i = 0; i < n; ++i) {
          z[i] = x[i] + y[i];
          ...
      }
  }
  ...
}
```

OpenACC
Directives for Accelerators

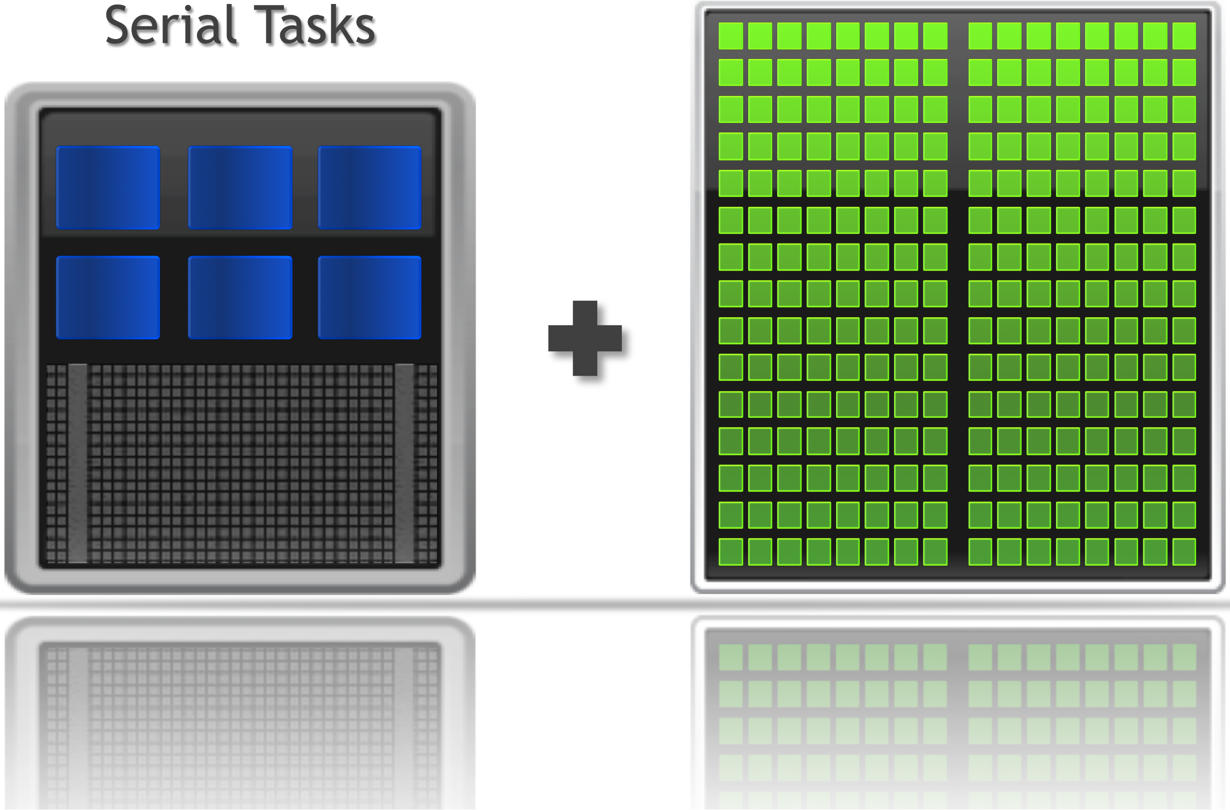
- Incremental
- Single source
- Interoperable
- Performance portable
- CPU, GPU, MIC

ACCELERATED COMPUTING

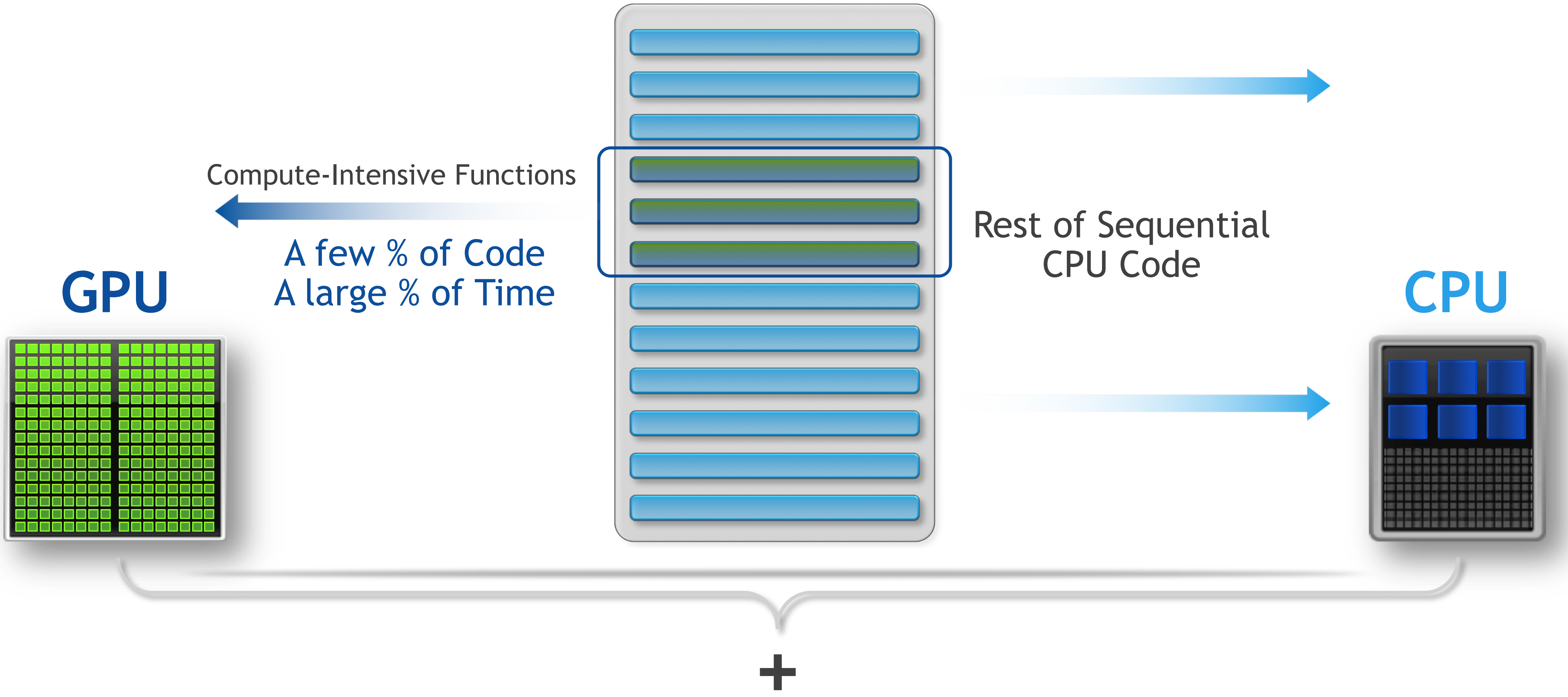
10X PERFORMANCE & 5X ENERGY EFFICIENCY FOR HPC

CPU
Optimized for
Serial Tasks

GPU Accelerator
Optimized for
Parallel Tasks



WHAT IS HETEROGENEOUS PROGRAMMING?



PORTABILITY & PERFORMANCE

Portability

Performance

- Accelerated Libraries
- High performance with little or no code change
- Limited by what libraries are available
- Compiler Directives
- High Level: Based on existing languages; simple, familiar, portable
- High Level: Performance may not be optimal
- Parallel Language Extensions
- Greater flexibility and control for maximum performance
- Often less portable and more time consuming to implement

CODE FOR PORTABILITY & PERFORMANCE

Libraries

- Implement as much as possible using portable libraries

Directives

- Use directives for rapid and portable development

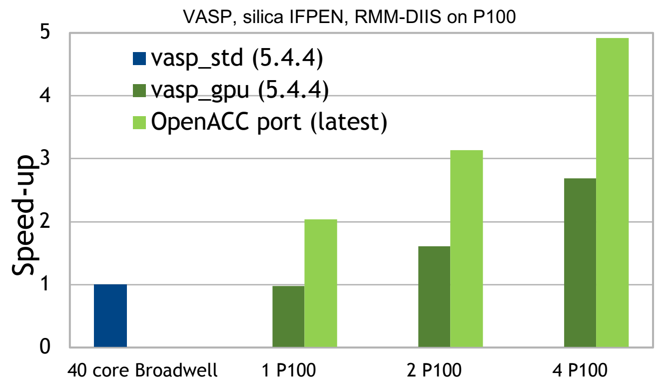
Languages

- Use lower level languages for important kernels

OPENACC GROWING MOMENTUM

Wide Adoption Across Key HPC Codes

ANSYS Fluent, Gaussian, VASP



* OpenACC port covers more VASP routines than CUDA, OpenACC port planned top down, with complete analysis of the call tree, OpenACC port leverages improvements in latest VASP Fortran source base

3 of Top 5 HPC Applications Use OpenACC

GTC

XGC

ACME

FLASH

LSDalton

CAAR Codes Use OpenACC

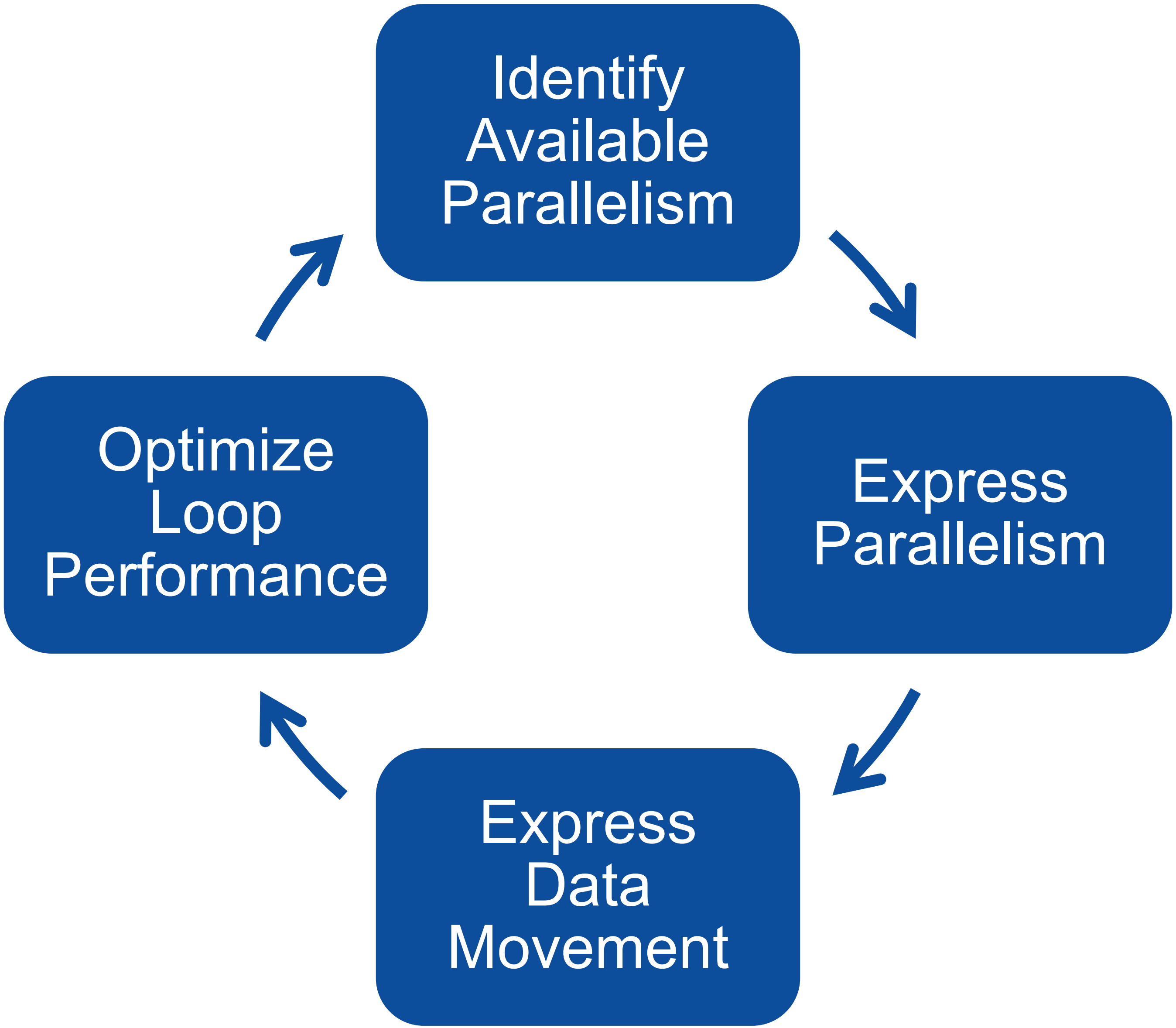
Key Codes Globally

COSMO, IFS(ESCAPE), NICAM, ICON, MPAS

Gordon Bell Finalist

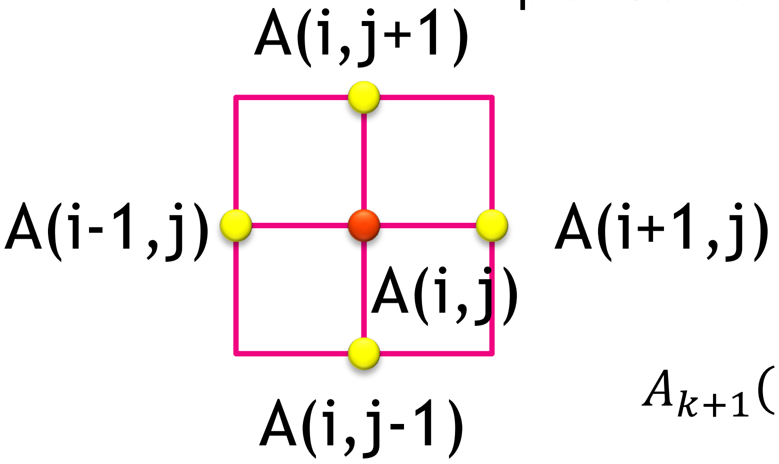
CAM-SE on Sunway Taihulight

OpenACC Dominates in Climate & Weather


Identify Available Parallelism
Express Parallelism
Express Data Movement
Optimize Loop Performance

EXAMPLE: JACOBI ITERATION

- Iteratively converges to correct value (e.g. Temperature), by computing new values at each point from the average of neighboring points.
- Common, useful algorithm
- Example: Solve Laplace equation in 2D: $\nabla^2 f(x, y) = 0$



$$A_{k+1}(i, j) = \frac{A_k(i-1, j) + A_k(i+1, j) + A_k(i, j-1) + A_k(i, j+1)}{4}$$

OpenACC
More Science, Less Programming

JACOBI ITERATION: C CODE

```
while ( err > tol && iter < iter_max ) {
  err=0.0;


  for( int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {

      Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                           A[j-1][i] + A[j+1][i]);

      err = max(err, abs(Anew[j][i] - A[j][i]));
    }
  }


  for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
      A[j][i] = Anew[j][i];
    }
  }

  iter++;
}
```

Iterate until converged

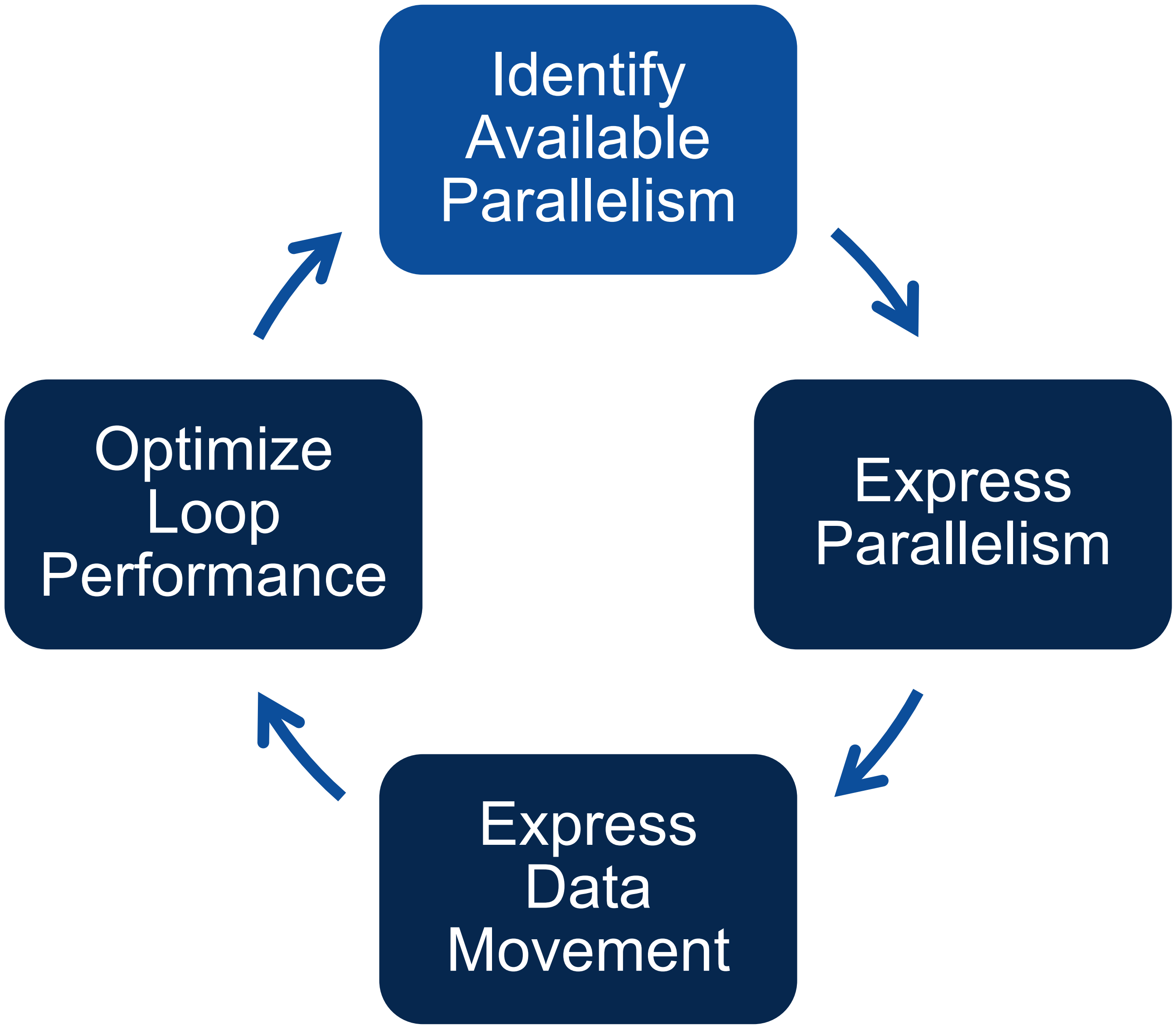
Iterate across matrix elements

Calculate new value from neighbors

Compute max error for convergence

Swap input/output arrays

OpenACC
More Science, Less Programming


Identify Available Parallelism
Express Parallelism
Express Data Movement
Optimize Loop Performance

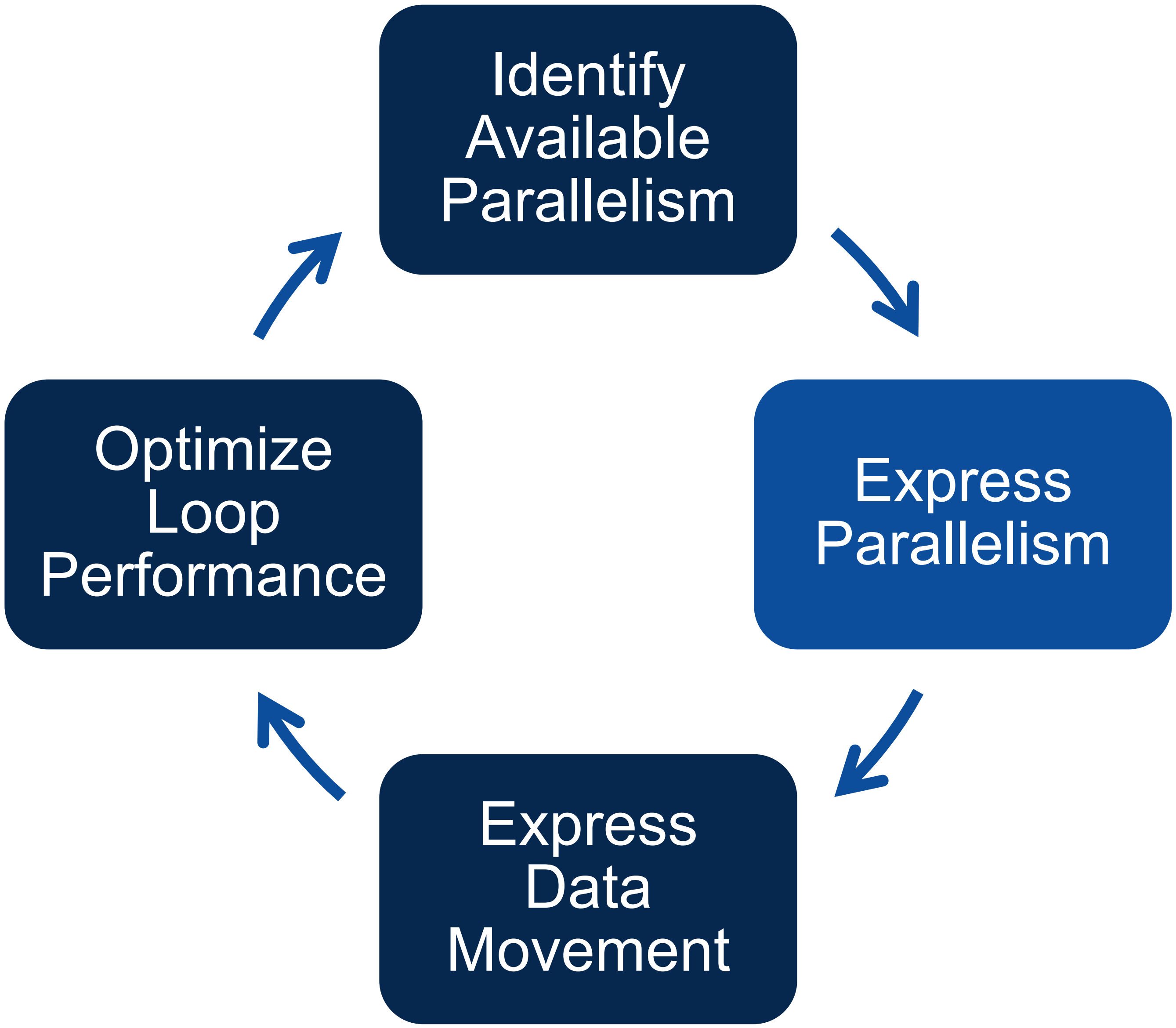
IDENTIFY PARALLELISM

```
while ( err > tol && iter < iter_max ) {
  err=0.0;


  for( int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {

      Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                           A[j-1][i] + A[j+1][i]);

      err = max(err, abs(Anew[j][i] - A[j][i]));
    }
  }


  for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
      A[j][i] = Anew[j][i];
    }
  }

  iter++;
}
```

Data dependency between iterations.

Independent loop iterations

Independent loop iterations


Identify Available Parallelism
Express Parallelism
Express Data Movement
Optimize Loop Performance

OPENACC KERNELS DIRECTIVE

The kernels directive identifies a region that may contain *loops* that the compiler can turn into parallel *kernels*.

```
#pragma acc kernels
{
for(int i=0; i<N; i++)
{
  x[i] = 1.0;
  y[i] = 2.0;
}

for(int i=0; i<N; i++)
{
  y[i] = a*x[i] + y[i];
}
}
```

kernel 1

kernel 2

The compiler identifies 2 parallel loops and generates 2 kernels.

PARALLELIZE WITH OPENACC KERNELS

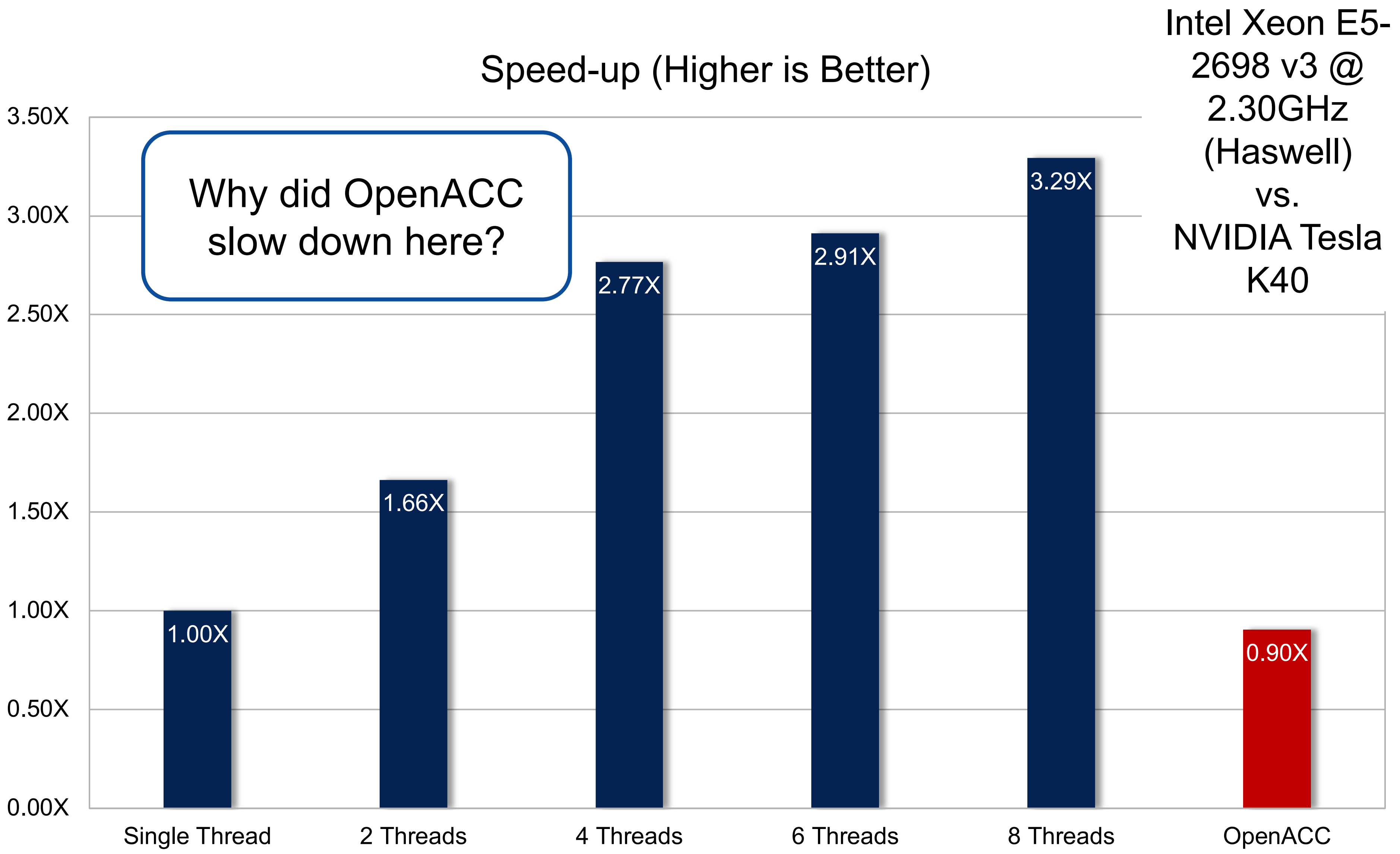
```
while ( err > tol && iter < iter_max ) {
  err=0.0;

#pragma acc kernels
{
  for( int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {

      Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                           A[j-1][i] + A[j+1][i]);

      err = max(err, abs(Anew[j][i] - A[j][i]));
    }
  }

  for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
      A[j][i] = Anew[j][i];
    }
  }
}
  iter++;
}
```

Look for parallelism within this region.

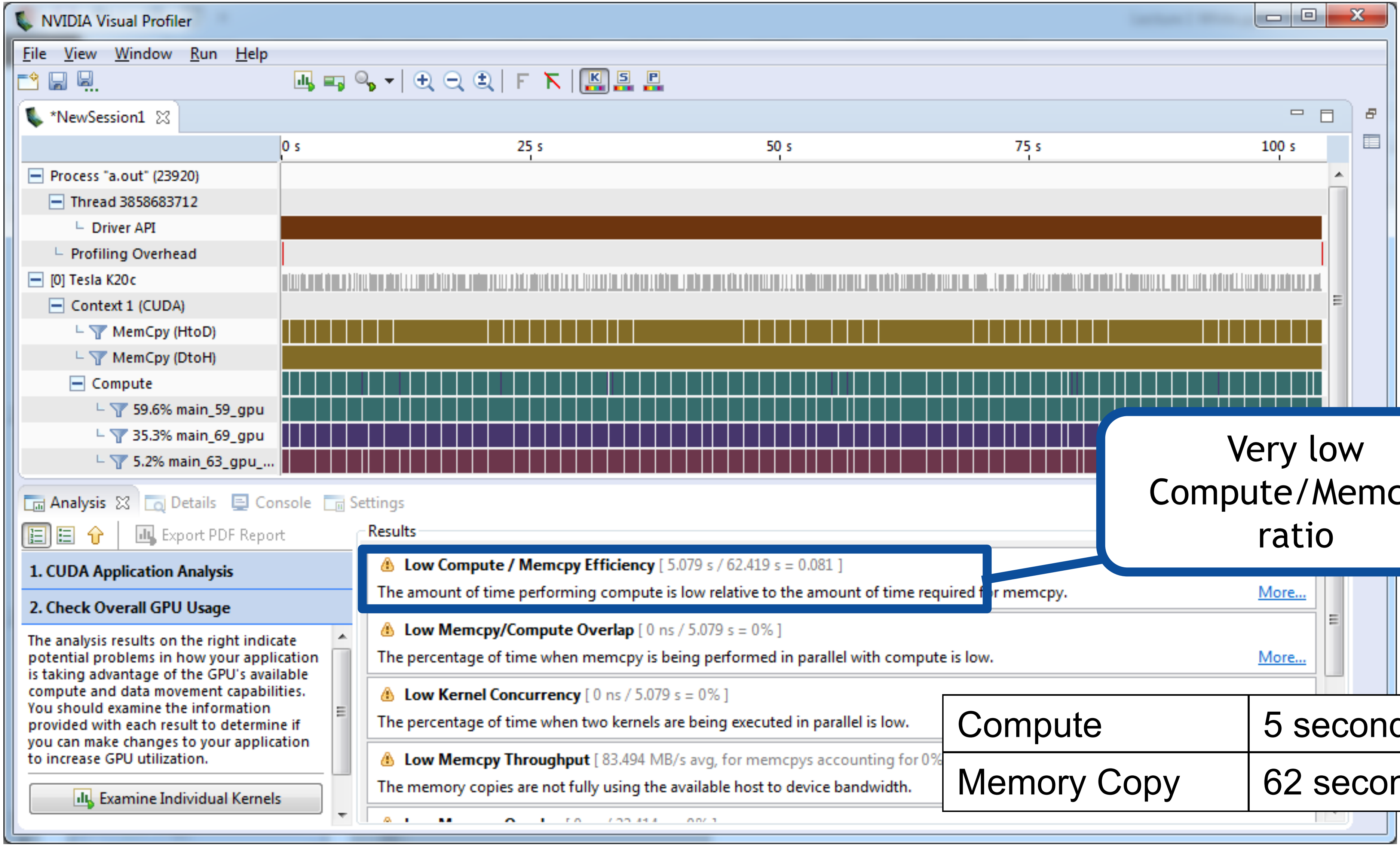
BUILDING THE CODE

```
$ pgcc -fast -ta=tesla -Minfo=all laplace2d.c
main:
     40, Loop not fused: function call before adjacent loop
         Generated vector sse code for the loop
     51, Loop not vectorized/parallelized: potential early exits
     55, Generating copyout(Anew[1:4094][1:4094])
         Generating copyin(A[:][:])
         Generating copyout(A[1:4094][1:4094])
         Generating Tesla code
     57, Loop is parallelizable
     59, Loop is parallelizable
         Accelerator kernel generated
         57, #pragma acc loop gang /* blockIdx.y */
         59, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
         63, Max reduction generated for error
     67, Loop is parallelizable
     69, Loop is parallelizable
         Accelerator kernel generated
         67, #pragma acc loop gang /* blockIdx.y */
         69, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
```

OpenACC
More Science, Less Programming

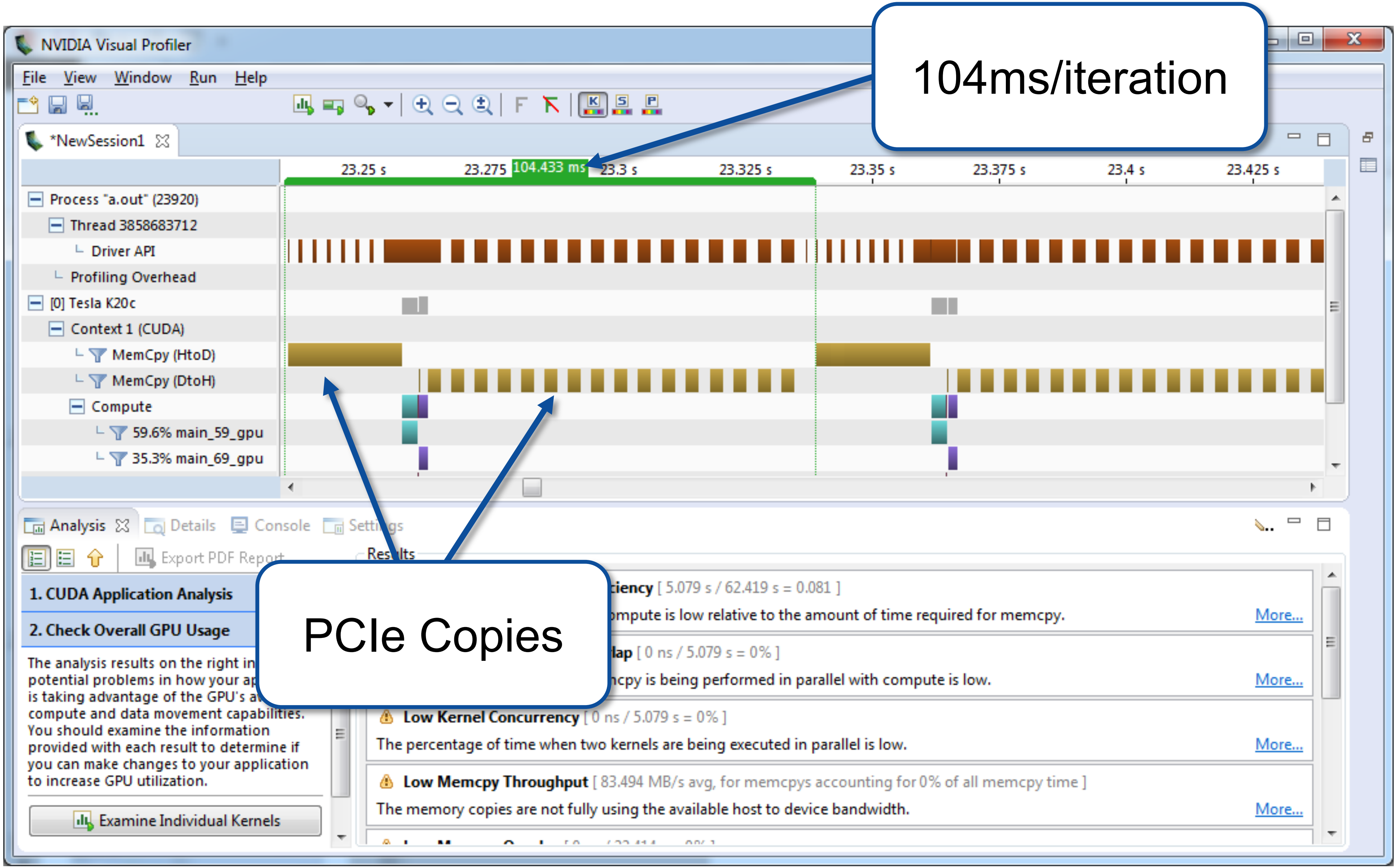


Speed-up (Higher is Better)
Intel Xeon E5-2698 v3 @ 2.30GHz (Haswell) vs. NVIDIA Tesla K40
Why did OpenACC slow down here?
3.50X
3.00X
2.50X
2.00X
1.50X
1.00X
0.50X
0.00X
1.00X
1.66X
2.77X
2.91X
3.29X
0.90X
Single Thread
2 Threads
4 Threads
6 Threads
8 Threads
OpenACC



Very low Compute/Memcpy ratio

Compute	5 seconds
Memory Copy	62 seconds


104ms/iteration
PCIe Copies
NVIDIA Visual Profiler
File View Window Run Help
*NewSession1
23.25 s
23.275
104.433 ms
23.3 s
23.325 s
23.35 s
23.375 s
23.4 s
23.425 s
Process "a.out" (23920)
Thread 3858683712
Driver API
Profiling Overhead
[0] Tesla K20c
Context 1 (CUDA)
MemCpy (HtoD)
MemCpy (DtoH)
Compute
59.6% main_59_gpu
35.3% main_69_gpu
Analysis
Details
Console
Export PDF Report
Results
1. CUDA Application Analysis
2. Check Overall GPU Usage
The analysis results on the right in
potential problems in how your ap
is taking advantage of the GPU's a
compute and data movement capabilities.
You should examine the information
provided with each result to determine if
you can make changes to your application
to increase GPU utilization.
Examine Individual Kernels
[5.079 s / 62.419 s = 0.081]
mpute is low relative to the amount of time required for memcpy.
[0 ns / 5.079 s = 0%]
cpy is being performed in parallel with compute is low.
Low Kernel Concurrency [0 ns / 5.079 s = 0%]
The percentage of time when two kernels are being executed in parallel is low.
Low Memcpy Throughput [83.494 MB/s avg, for memcpys accounting for 0% of all memcpy time]
The memory copies are not fully using the available host to device bandwidth.
More...

EXCESSIVE DATA TRANSFERS

```
while ( err > tol && iter < iter_max )
{
  err=0.0;
```

A, Anew resident on host

C
o
p
y

```
#pragma acc kernels
```

A, Anew resident on accelerator

```
 for( int j = 1; j < n-1; j++) {
   for(int i = 1; i < m-1; i++) {
     Anew[j][i] = 0.25 * (A[j][i+1] +
                          A[j][i-1] + A[j-1][i] +
                          A[j+1][i]);
     err = max(err, abs(Anew[j][i] -
                           A[j][i]);
   }
 }
 ...
```

These copies happen every iteration of the outer while loop!

C
o
p
y

A, Anew resident on accelerator

A, Anew resident on host

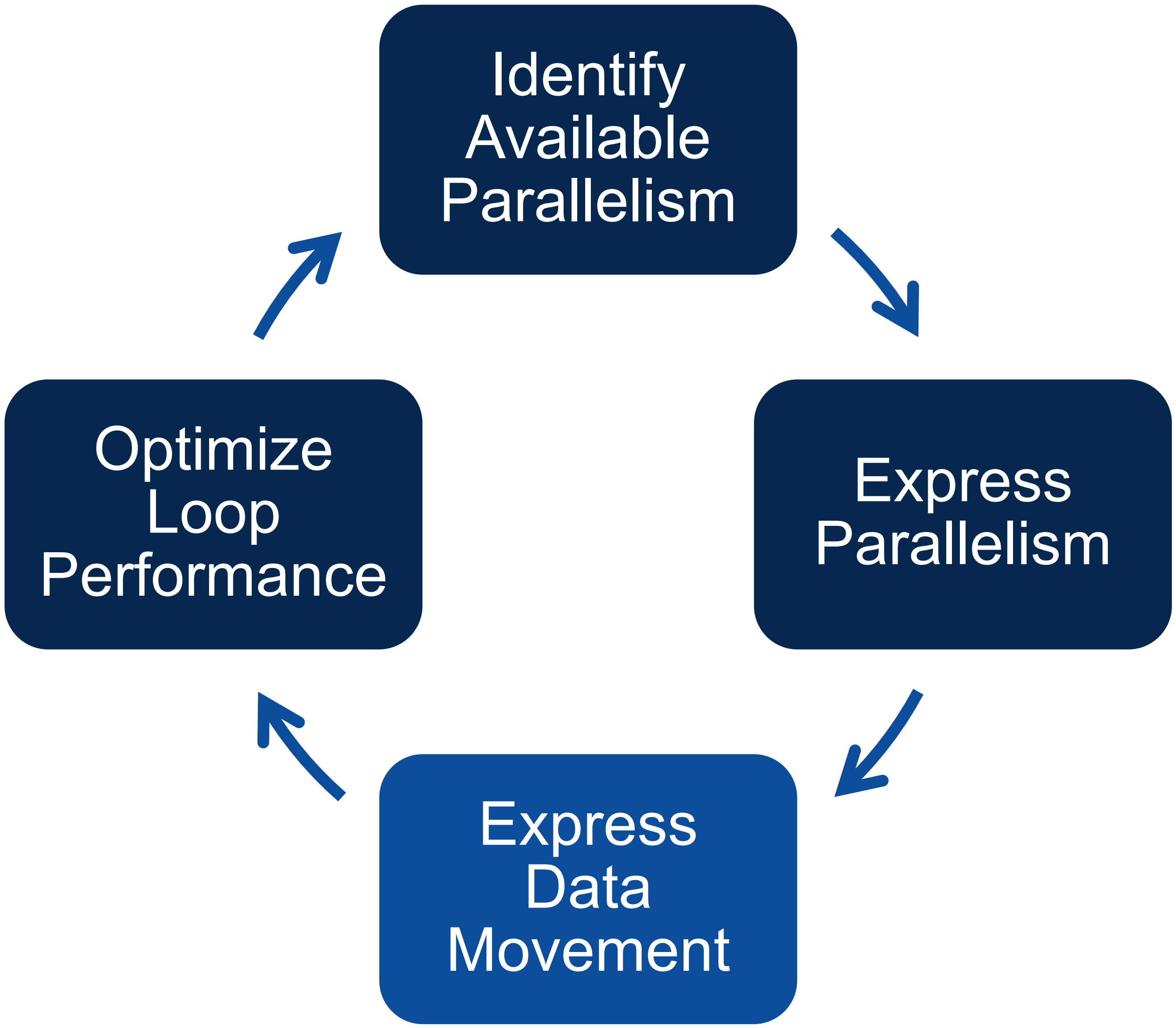
```
      ...
}
```

IDENTIFYING DATA LOCALITY

```
while ( err > tol && iter < iter_max ) {
  err=0.0;

#pragma acc kernels
  {
  for( int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {

      Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                           A[j-1][i] + A[j+1][i]);

      err = max(err, abs(Anew[j][i] - A[j][i]));
    }
  }

  for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
      A[j][i] = Anew[j][i];
    }
  }
  }

  iter++;
}
```

Does the CPU need the data between these loop nests?

Does the CPU need the data between iterations of the convergence loop?


Identify Available Parallelism
Express Parallelism
Express Data Movement
Optimize Loop Performance

DATA REGIONS

- The **data** directive defines a region of code in which GPU arrays remain on the GPU and are shared among all kernels in that region.

```
#pragma acc data
{
#pragma acc kernels
...

#pragma acc kernels
...
}
```

Data Region

Arrays used within the data region will remain on the GPU until the end of the data region.

OpenACC
More Science, Less Programming

DATA CLAUSES

`copy ( list )`	Allocates memory on GPU and copies data from host to GPU when entering region and copies data to the host when exiting region.
`copyin ( list )`	Allocates memory on GPU and copies data from host to GPU when entering region.
`copyout ( list )`	Allocates memory on GPU and copies data to the host when exiting region.
`create ( list )`	Allocates memory on GPU but does not copy.
`present ( list )`	Data is already present on GPU from another containing data region.
`deviceptr( list )`	The variable is a device pointer (e.g. CUDA) and can be used directly on the device.

OpenACC
More Science, Less Programming

EXPRESS DATA LOCALITY

```
#pragma acc data copy(A) create(Anew)
while ( err > tol && iter < iter_max ) {
  err=0.0;
#pragma acc kernels
{
  for( int j = 1; j < n-1; j++) {
    for(int i = 1; i < m-1; i++) {

      Anew[j][i] = 0.25 * (A[j][i+1] + A[j][i-1] +
                           A[j-1][i] + A[j+1][i]);

      err = max(err, abs(Anew[j][i] - A[j][i]));
    }
  }

  for( int j = 1; j < n-1; j++) {
    for( int i = 1; i < m-1; i++ ) {
      A[j][i] = Anew[j][i];
    }
  }
}
  iter++;
}
```

Copy A to/from the accelerator only when needed.

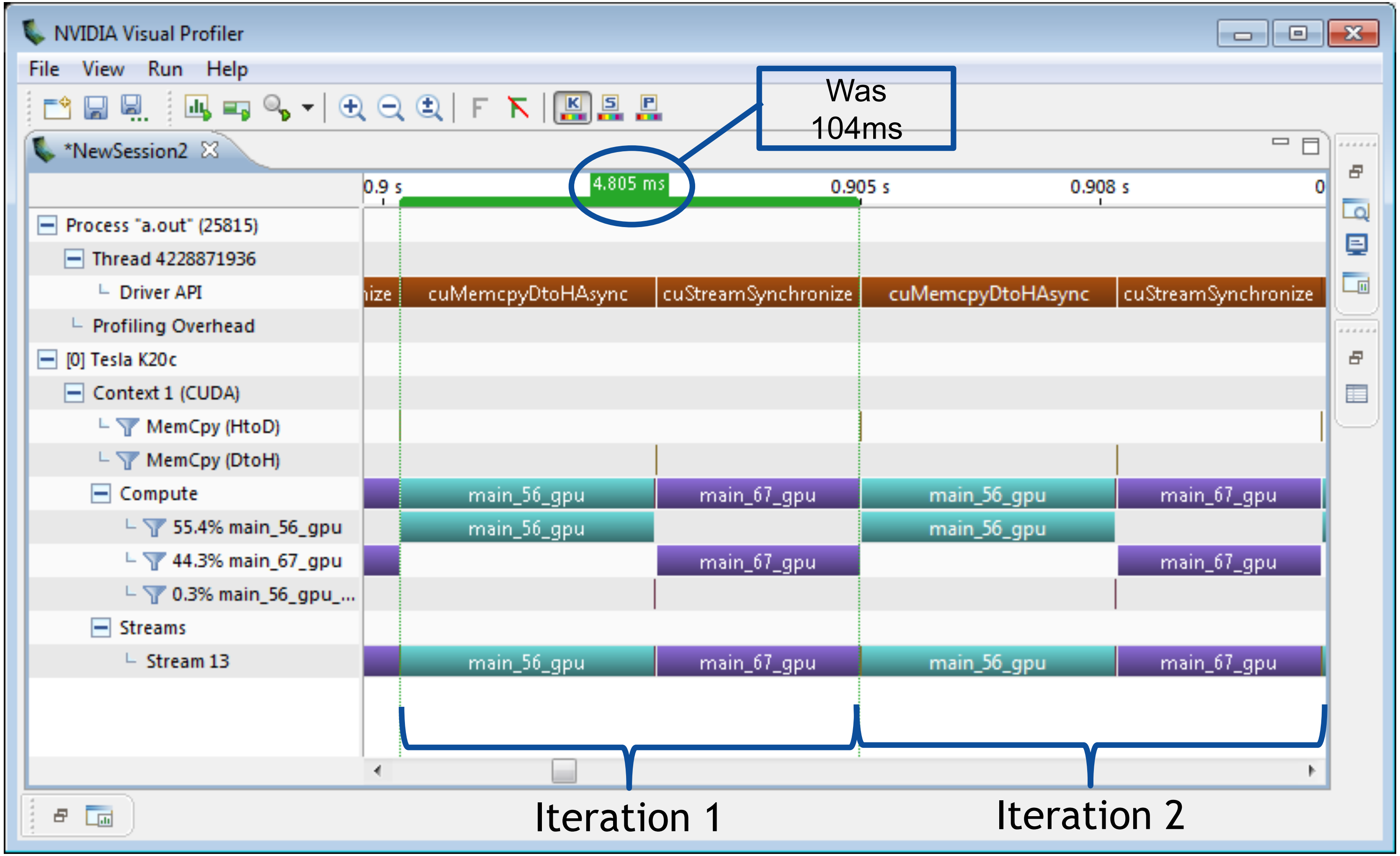
Create Anew as a device temporary.

REBUILDING THE CODE

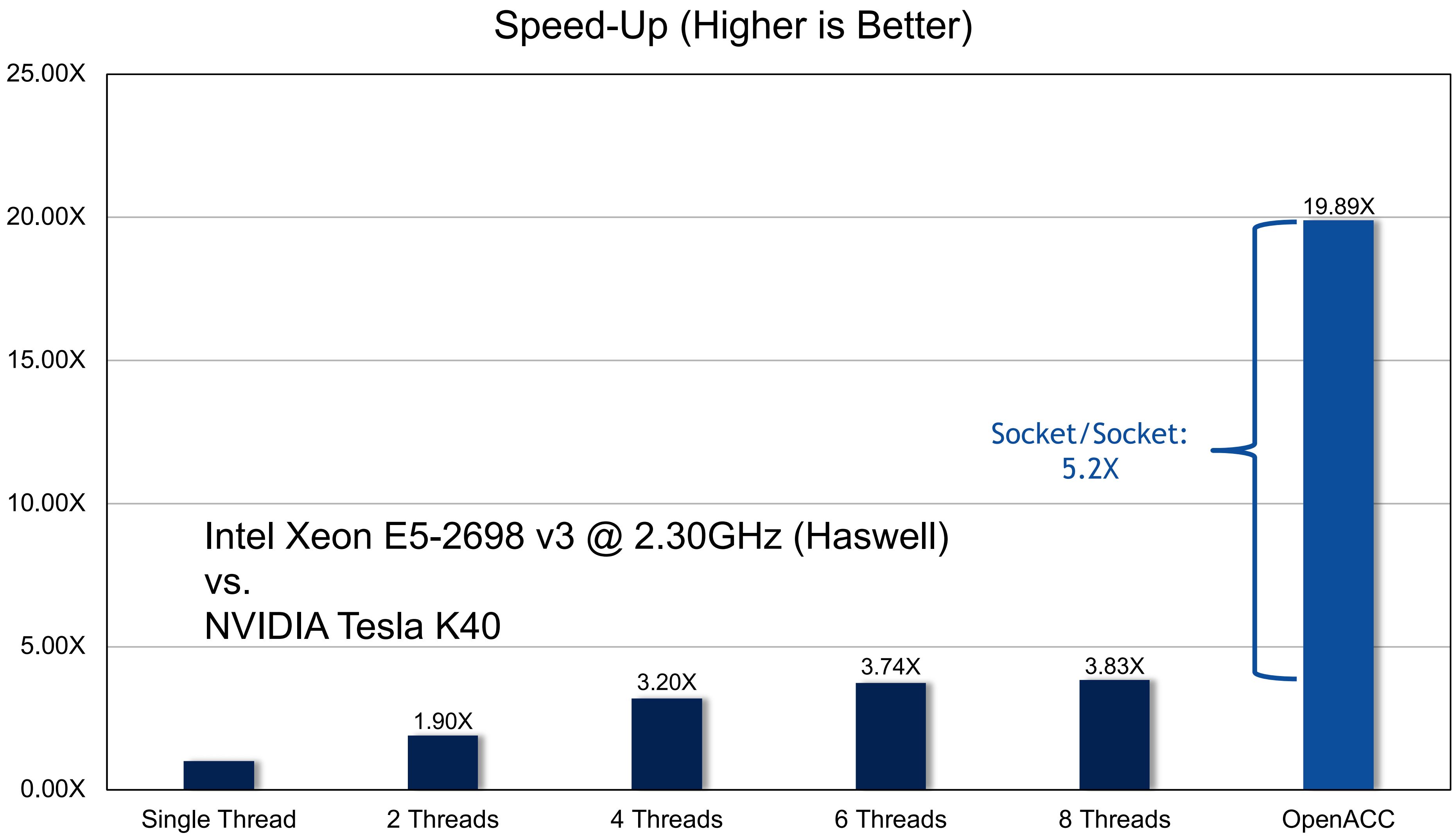
```
$ pgcc -fast -acc -ta=tesla -Minfo=all laplace2d.c
main:
     40, Loop not fused: function call before adjacent loop
         Generated vector sse code for the loop
     51, Generating copy(A[:][:])
         Generating create(Anew[:][:])
         Loop not vectorized/parallelized: potential early exits
     56, Accelerator kernel generated
         56, Max reduction generated for error
         57, #pragma acc loop gang /* blockIdx.x */
         59, #pragma acc loop vector(256) /* threadIdx.x */
     56, Generating Tesla code
     59, Loop is parallelizable
     67, Accelerator kernel generated
         68, #pragma acc loop gang /* blockIdx.x */
         70, #pragma acc loop vector(256) /* threadIdx.x */
     67, Generating Tesla code
     70, Loop is parallelizable
```

OpenACC
More Science, Less Programming

VISUAL PROFILER: DATA REGION



OpenACC


Speed-Up (Higher is Better)
25.00X
20.00X
15.00X
10.00X
5.00X
0.00X
19.89X
Socket/Socket:
5.2X
Intel Xeon E5-2698 v3 @ 2.30GHz (Haswell)
vs.
NVIDIA Tesla K40
1.90X
3.20X
3.74X
3.83X
Single Thread
2 Threads
4 Threads
6 Threads
8 Threads
OpenACC

INTRODUCING THE NEW OPENACC TOOLKIT

- Free Toolkit Offers Simple & Powerful Path to Accelerated Computing

OpenACC
Directives for Accelerators

http://developer.nvidia.com/openacc

PGI Compiler
Free OpenACC compiler for academia

NVProf Profiler
Easily find where to add compiler directives

GPU Wizard
Identify which GPU libraries can jumpstart code

Code Samples
Learn from examples of real-world algorithms

Documentation
Quick start guide, Best practices, Forums

DOWNLOAD THE OPENACC TOOLKIT

- Go to https://developer.nvidia.com/openacc

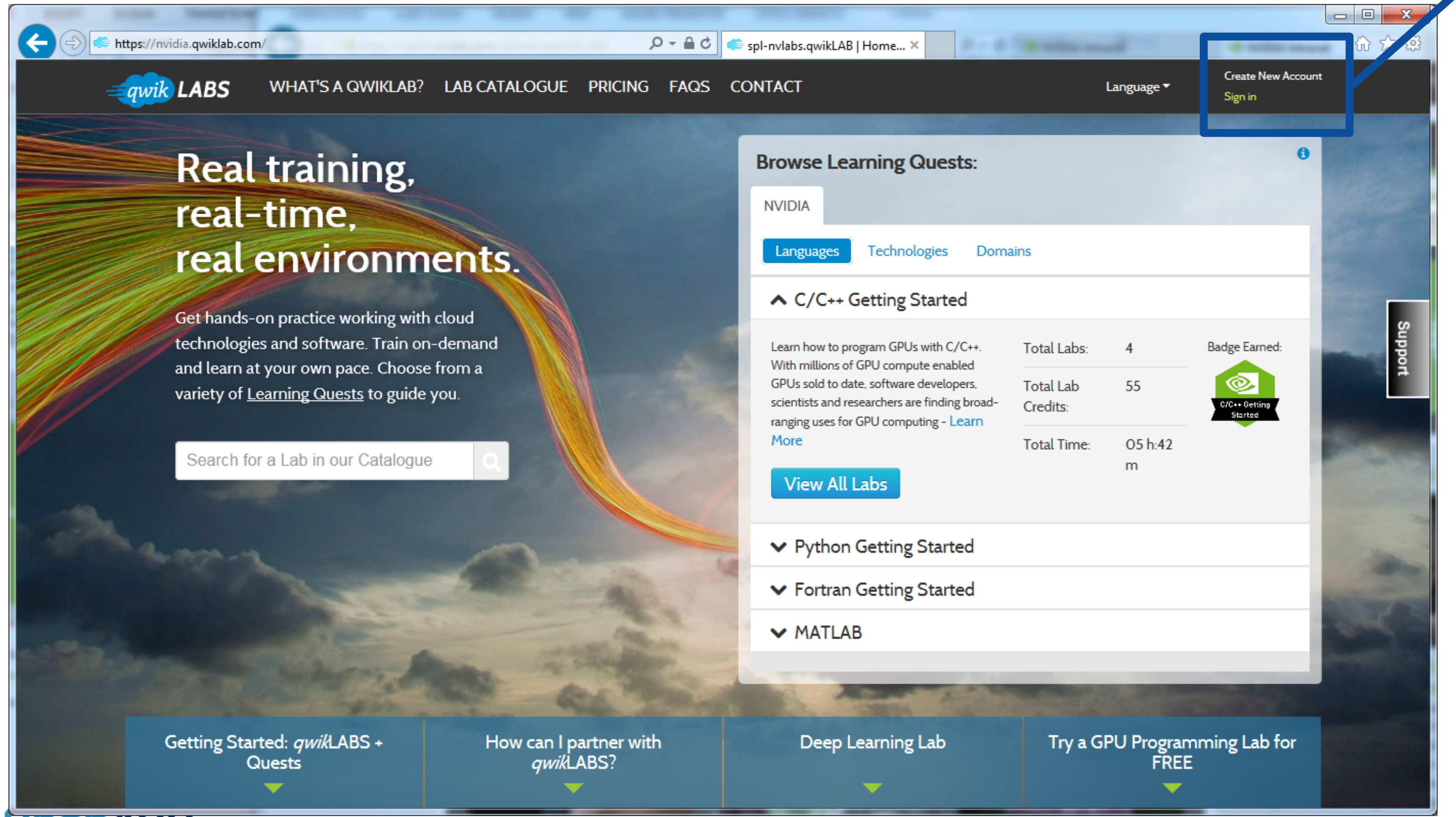
OpenACC
More Science, Less Programming

WINDOWS/MAC DEVELOPERS

- The OpenACC Toolkit is only available on Linux, however…
- The PGI compiler is available on Mac and Windows from http://www.pgroup.com/support/trial.htm
  - You should still register for the OpenACC Toolkit to get the 90 day license.
- The CUDA Toolkit contains the libraries and profiling tools that will be used in this course.
  - https://developer.nvidia.com/cuda-zone
- The OpenACC Programming Guide is available from http://bit.ly/openacc-guide
  - Obtaining all examples and guides from the toolkit will still require downloading the full OpenACC toolkit.

GETTING ACCESS

- Go to nvidia.qwiklab.com, log-in or create an account



Sign In or Create a New Account

OpenACC
More Science, Less Programming

WHERE TO FIND HELP

- OpenACC Course Recordings - https://developer.nvidia.com/openacc-course
- OpenACC on StackOverflow - http://stackoverflow.com/questions/tagged/openacc
- OpenACC Toolkit - http://developer.nvidia.com/openacc

- Additional Resources:

- Parallel Forall Blog - http://devblogs.nvidia.com/parallelforall/
- GPU Technology Conference - http://www.gputechconf.com/
- OpenACC Website - http://openacc.org/

OPENACC RESOURCES

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

FREE Compilers

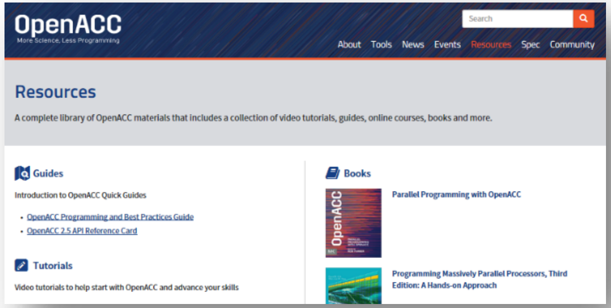
PGI Community EDITION

slack

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

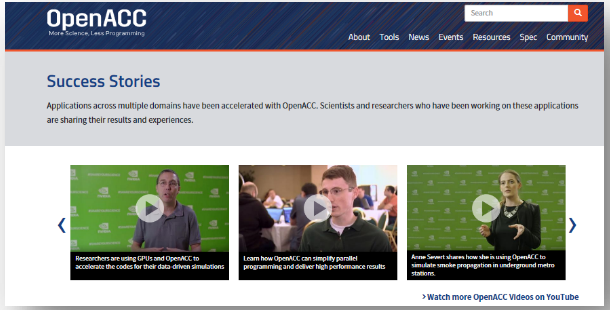
Resources

https://www.openacc.org/resources



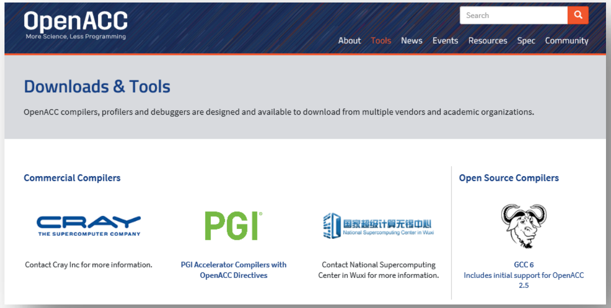
Success Stories

https://www.openacc.org/success-stories



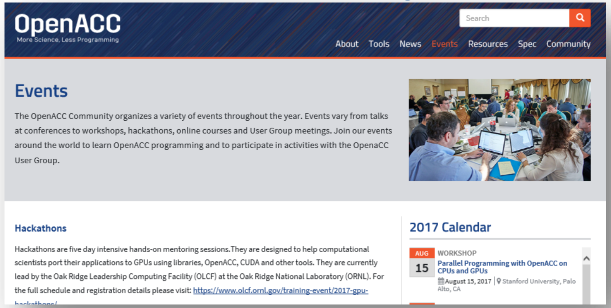
Compilers and Tools

https://www.openacc.org/tools



Events

https://www.openacc.org/events



OPENACC SLACK COMMUNITY

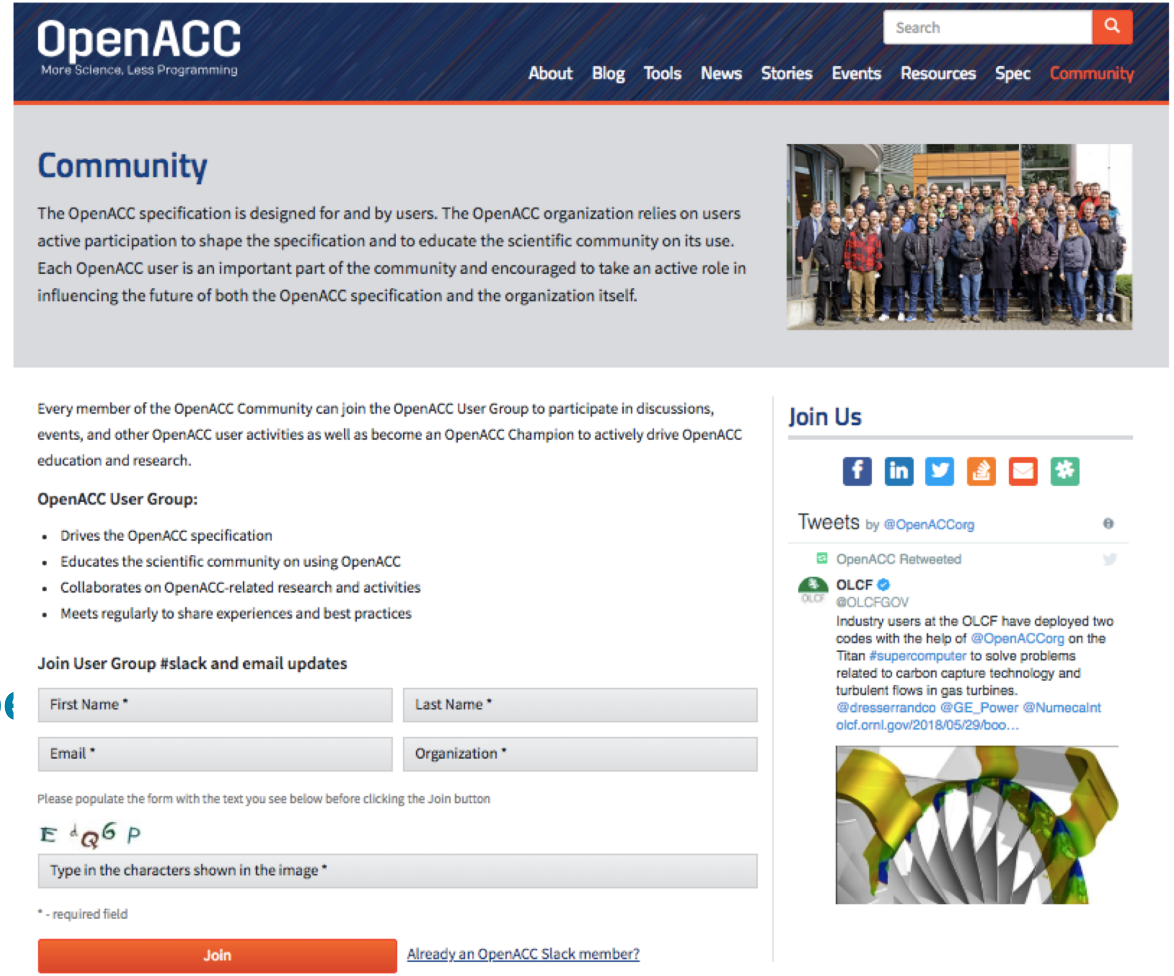
Sign up!

Join 340 other slack members !

Get your answers from

scientific developers, compiler develope

and users ☺


OpenACC
More Science, Less Programming

Search

About Blog Tools News Stories Events Resources Spec Community

Community

The OpenACC specification is designed for and by users. The OpenACC organization relies on users active participation to shape the specification and to educate the scientific community on its use. Each OpenACC user is an important part of the community and encouraged to take an active role in influencing the future of both the OpenACC specification and the organization itself.

Every member of the OpenACC Community can join the OpenACC User Group to participate in discussions, events, and other OpenACC user activities as well as become an OpenACC Champion to actively drive OpenACC education and research.

OpenACC User Group:

- Drives the OpenACC specification
- Educates the scientific community on using OpenACC
- Collaborates on OpenACC-related research and activities
- Meets regularly to share experiences and best practices

Join User Group #slack and email updates

First Name * | Last Name *

Email * | Organization *

Please populate the form with the text you see below before clicking the Join button

Type in the characters shown in the image *

* - required field

Join | Already an OpenACC Slack member?

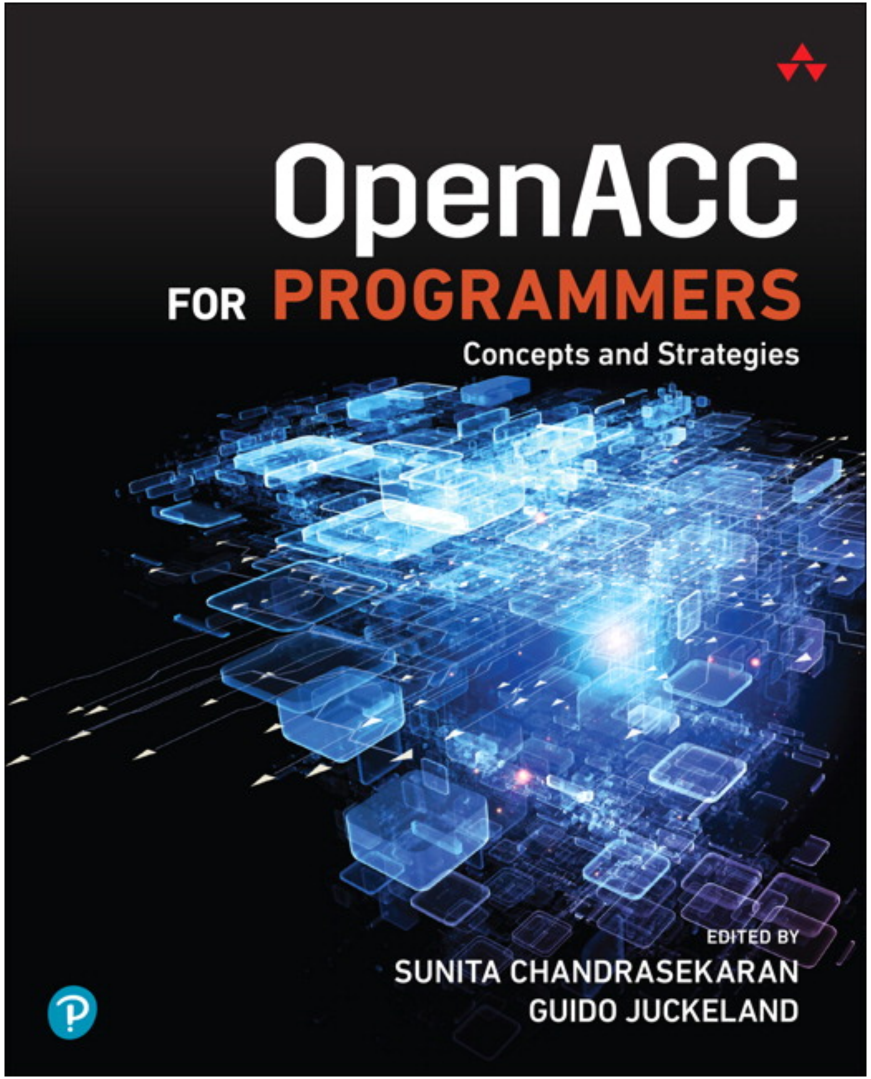
Join Us

Tweets by @OpenACCorg

OpenACC Retweeted

OLCF
@OLCFGOV

Industry users at the OLCF have deployed two codes with the help of @OpenACCorg on the Titan #supercomputer to solve problems related to carbon capture technology and turbulent flows in gas turbines. @dresserrandco @GE_Power @Numecalnt olcf.ornl.gov/2018/05/29/boo…



- An OpenACC Textbook recently published – November 2017
- Available via Amazon and other sites
- Example codes, chapter solutions available

https://github.com/OpenACCUserGroup/openacc_concept_strategies_book

Read More at

WWW.OPENACC.ORG

OpenACC
More Science, Less Programming