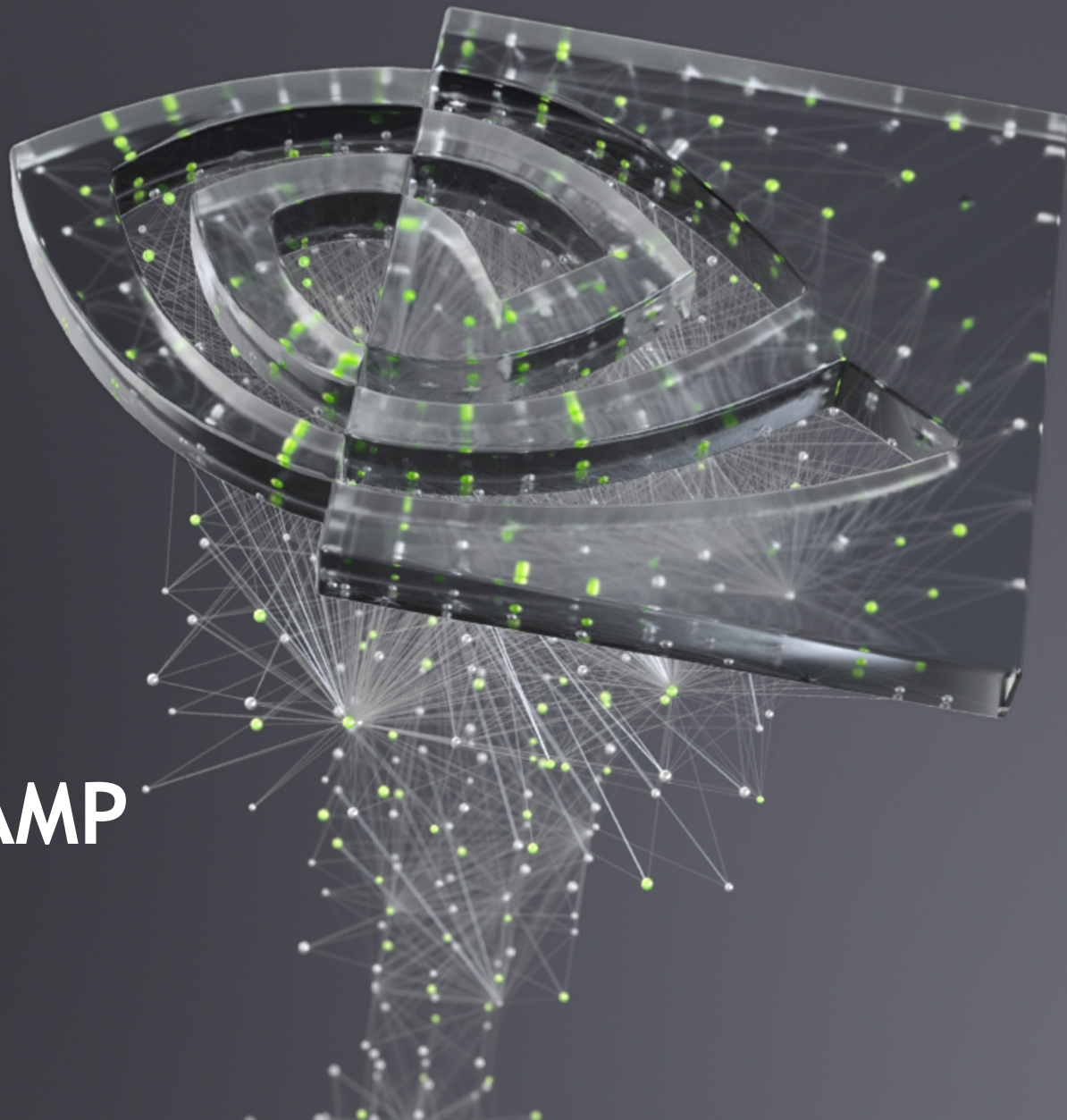


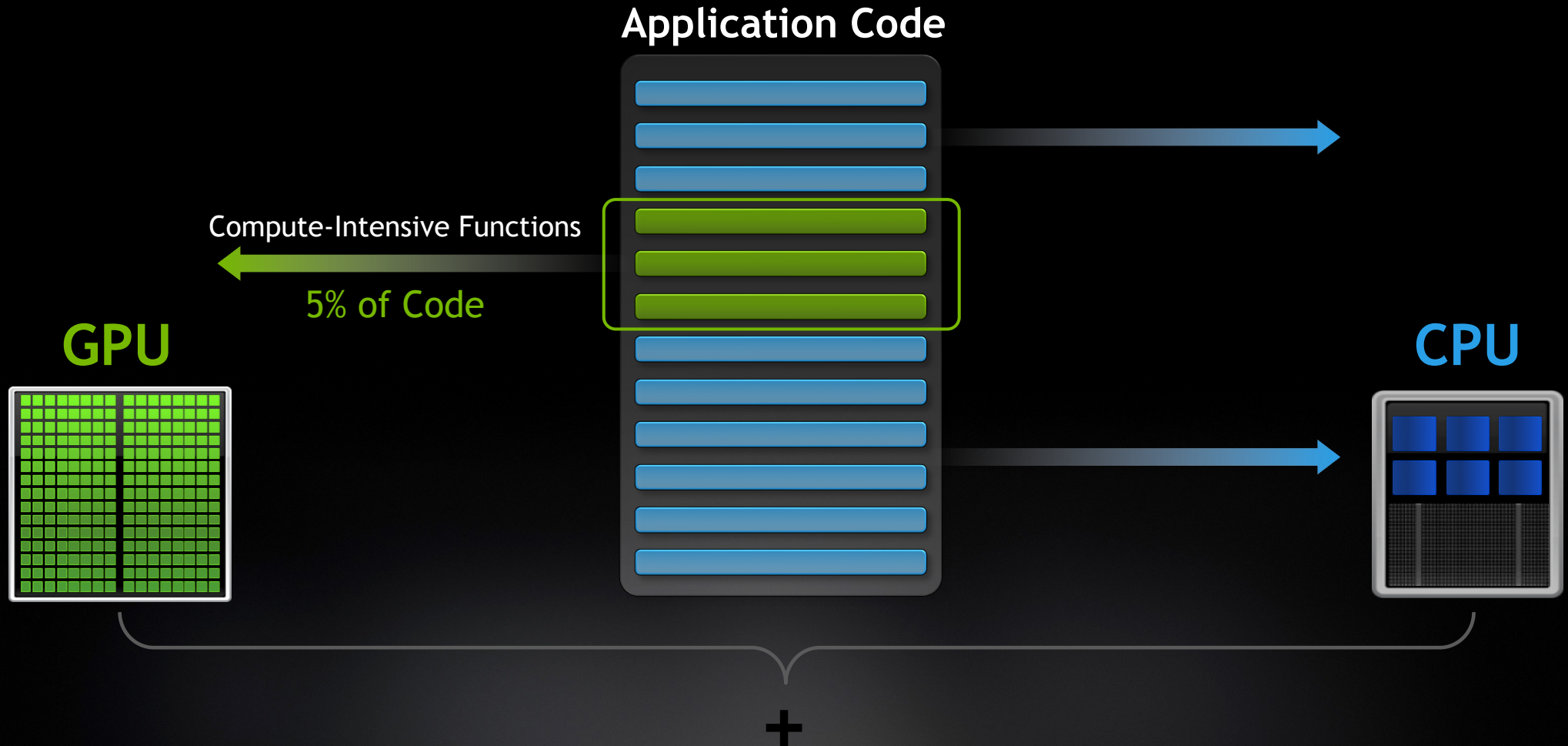


N-WAYS GPU BOOTCAMP

CONCLUSION

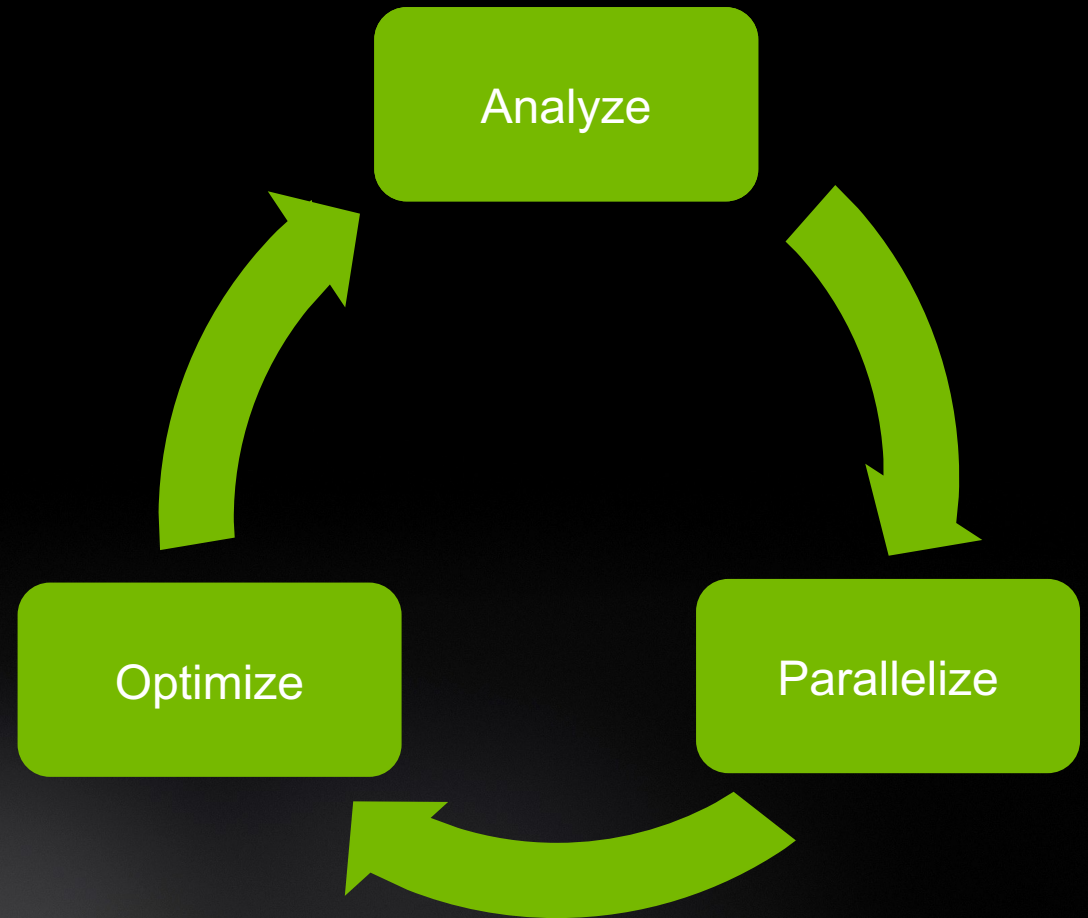


How GPU Acceleration Works



development CYCLE

- **Analyze** your code to determine most likely places needing parallelization or optimization.
- **Parallelize** your code by starting with the most time consuming parts and check for correctness.
- **Optimize** your code to improve observed speed-up from parallelization.



GPU Programming in 2023 and beyond

Math Libraries | Standard Languages | Directives | CUDA

```
std::transform(par, x, x+n, y, y,  
  [=] (float x, float y) {  
    return y + a*x;  
  });
```

```
do concurrent (i = 1:n)  
  y(i) = y(i) + a*x(i)  
enddo
```

**GPU Accelerated
C++ and Fortran**

```
#pragma acc data copy(x,y)  
{  
  ...  
  std::transform(par, x, x+n, y, y,  
    [=] (float x, float y) {  
      return y + a*x;  
    });  
  ...  
}
```

**Incremental Performance
Optimization with Directives**

```
__global__  
void saxpy(int n, float a,  
  float *x, float *y) {  
  int i = blockIdx.x * blockDim.x +  
    threadIdx.x  
  if (i < n) y[i] += a*x[i];  
}  
  
int main(void) {  
  cudaMallocManaged(&x, ...);  
  cudaMallocManaged(&y, ...);  
  ...  
  saxpy<<<(N+255)/256,256>>>(...,x, y)  
  cudaDeviceSynchronize();  
  ...  
}
```

**Maximize GPU Performance with
CUDA C++/Fortran**

GPU Accelerated Math Libraries

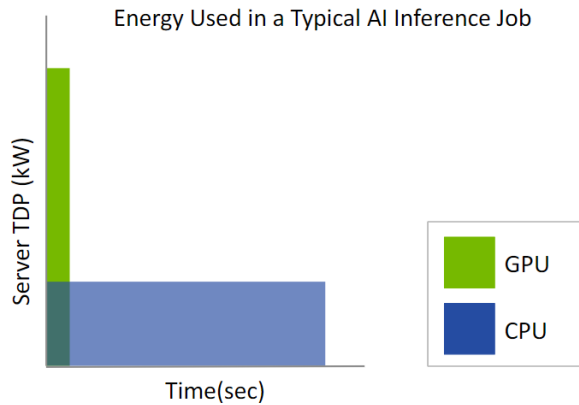
MATERIALS

- NWAYS bootcamp
 - [openhackathons-org/nways_accelerated_programming \(github.com\)](https://github.com/openhackathons-org/nways_accelerated_programming)
- OPENACC API
 - [API \(openacc.org\)](https://openacc.org)
- CUDA best practice guide
 - [CUDA Best Practices \(nvidia.com\)](https://nvidia.com)
- CUDA Programming guide
 - [CUDA C++ Programming Guide \(nvidia.com\)](https://nvidia.com)

ENERGY SAVING

GPUs typically have **higher peak active power consumption** than CPUs

However, they complete workloads **significantly faster**

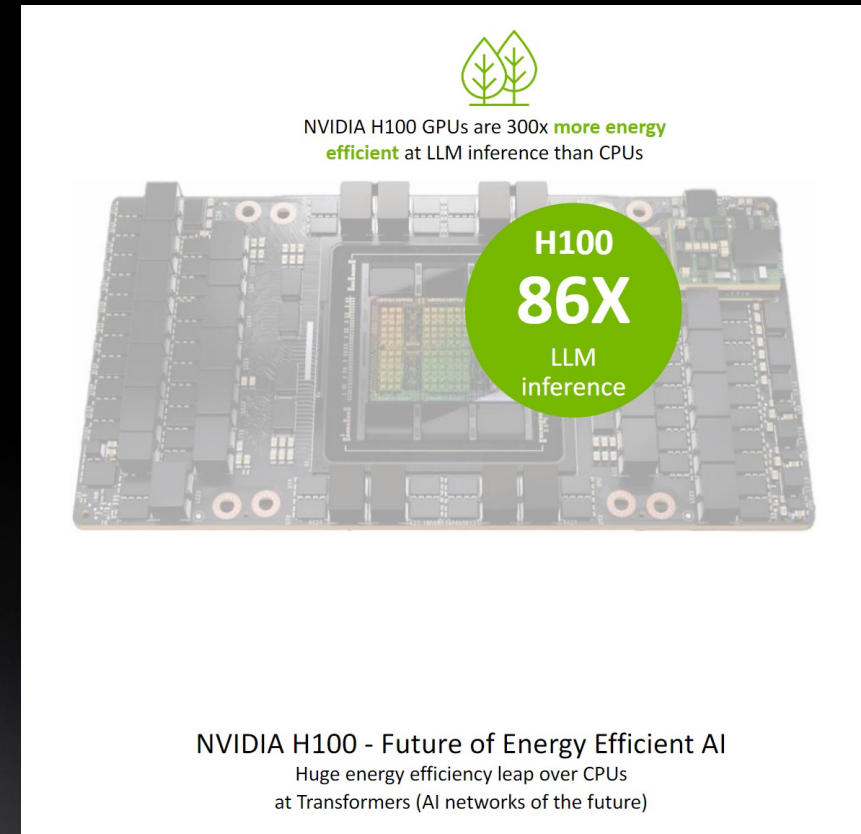
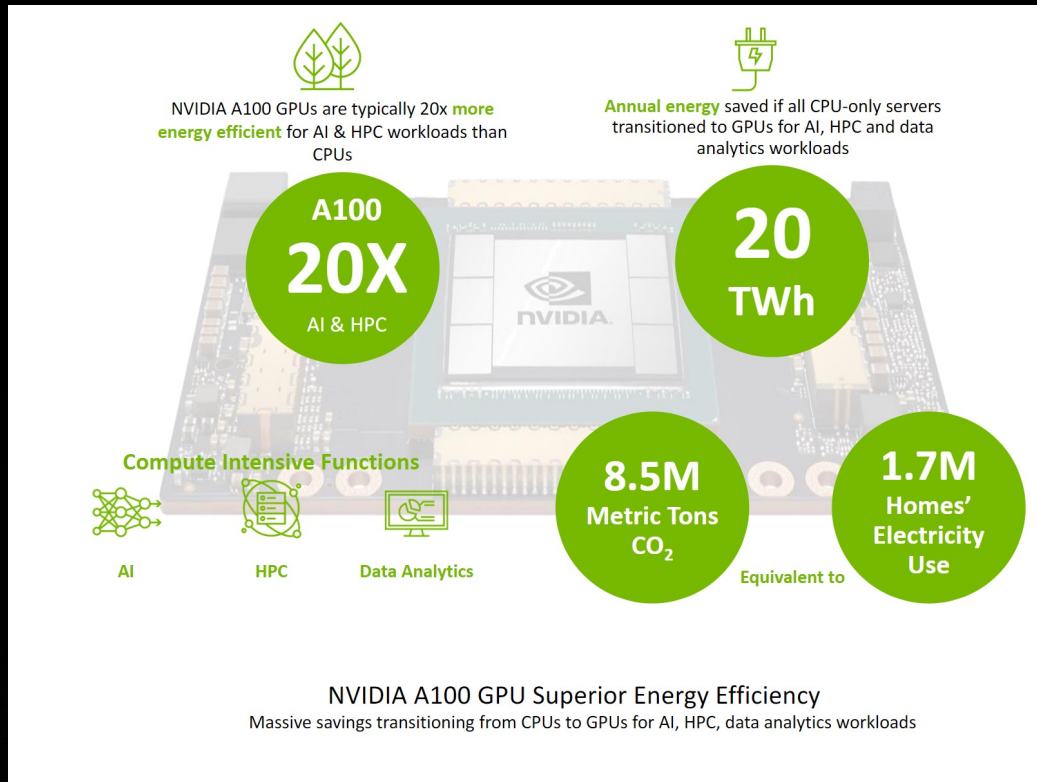


GPUs Use Less Energy in AI

Rapid workload completion allows return to idle state quickly

- Speed up is the most important advantage for GPU.
- Energy saving is another advantage for GPU computing in comparison to CPU.

ENERGY SAVING





Open Hackathons provide exciting opportunities for scientists to accelerate their AI or HPC research under the guidance of expert mentors from national laboratories, universities and industry leaders in a collaborative environment. Representing distinguished scholars and preeminent institutions around the world, these teams of mentors and attendees work together to realize performance gains and speedups using a variety of programming models, libraries and tools.

The goal of the Open Hackathon is for computational scientists to port, accelerate and optimize their scientific applications to modern computer architectures, including CPUs, GPUs and other computing technologies. Participating teams should leave the event either with their applications accelerated and/or optimized on the latest supercomputing hardware or a clear roadmap of the next steps needed to leverage these resources.

This Hackathon is open to everyone looking to take their projects to the next level; however, priority acceptance will be given to TWS/NCHC affiliated scientists and their collaborators.

If you would like to be notified when the call for applications is open, please click the button below.

NOTIFY ME

Important Event Dates

NCHC Open Hackathon 2023 Application
Deadline
October 04, 2023

NCHC Open Hackathon 2023 Team/Mentor
Meeting
November 27, 2023

NCHC Open Hackathon 2023 Day 1
November 27, 2023

NCHC Open Hackathon 2023 Day 2
December 06, 2023

2023 11/27-12/08 OpenACC Hackathon

<https://www.openhackathons.org/s/siteevent/a0C5e000005VZLiEAO/se000160>



THANK YOU



nvidia.