The Ultimate CUDA Development GPU

Introducing GeForce GTX TITAN

The Ultimate CUDA Development GPU

2688	4.5	1.27	288
CUDA Cores	Teraflops Single Precision	Teraflops Double Precision	GB/s Memory Bandwidth

GTX TITAN

Personal Supercomputer on Your Desktop

1 Teraflop < $1000

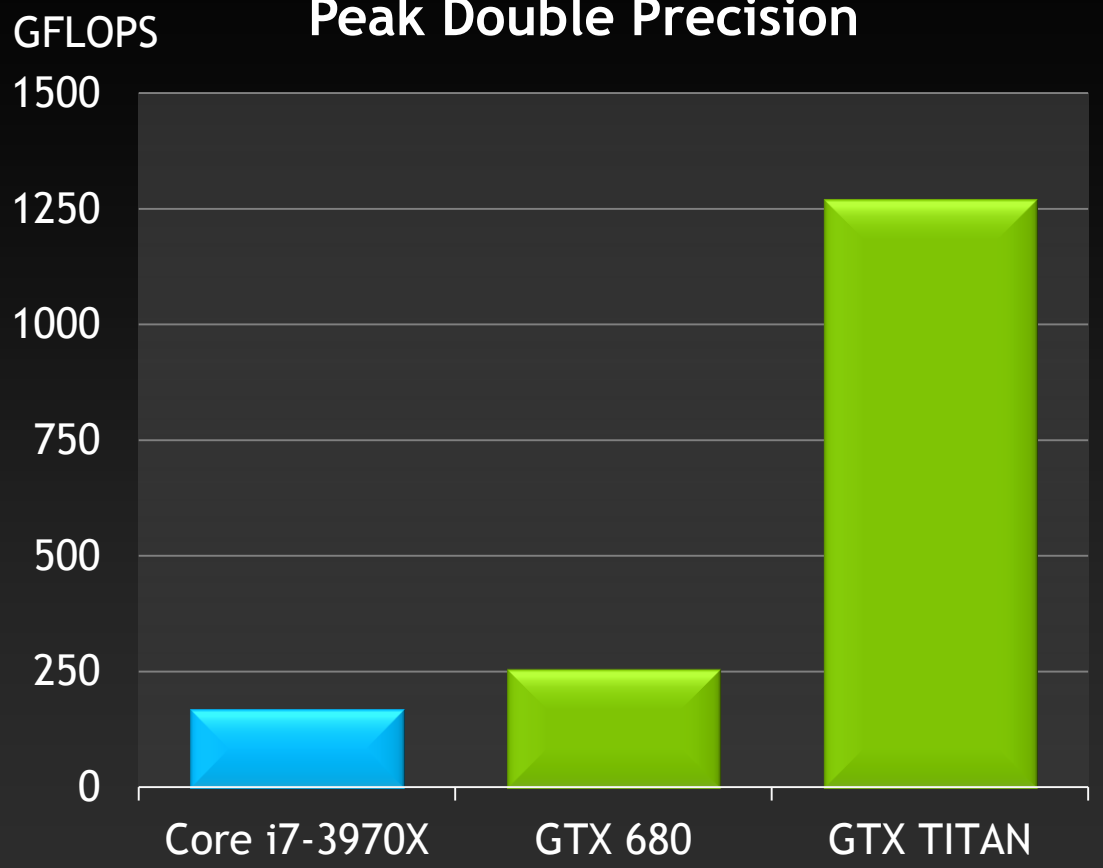
Develop Anywhere

Ease of Programming with

New Kepler Architecture

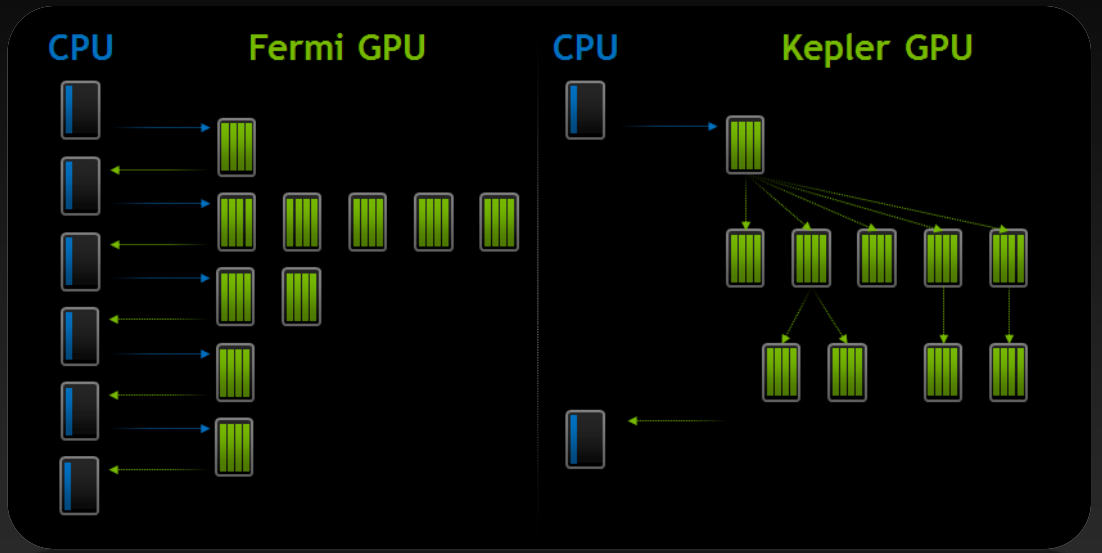
The Best of Kepler in a PC

Peak Double Precision



Boosts PC with 8x More Performance

Dynamic Parallelism



More Science, Less Coding

Dynamic Parallelism Makes Parallel Programming Easier

Quicksort

```
__device__ WorkStack stack;
__global__ void quicksort(int *data, int left, int right)
{
    int nleft, nright;

    // Partitions data based on pivot of first element.
    // Returns counts in nleft & nright
    partition(data+left, data+right, data[left], nleft, nright);

    // If a sub-array needs sorting, push it on the stack
    if(left < nright)
        stack.push(data, left, nright);
    if(nleft < right)
        stack.push(data, nleft, right);
}

__host__ void launch_quicksort(int *data, int count)
{
    // Launch initial quicksort to populate the stack
    quicksort<<< ... >>>(data, 0, count-1);

    // Loop more quicksorts until no more work exists
    while(1)
    {
        // Wait for all sorts at th
        cudaDeviceSynchronize();

        // Copy our stack from the
        WorkStack stack_copy;
        stack_copy = CopyFromDevice

        // Count of things on stack
        if(stack_copy.size() == 0)
            break;

        // Pop the stack and launch
        while(stack_copy.size())
        {
            WorkStack elem = stack_
            cudaStream_t s;
            cudaStreamCreate(&s);
            quicksort<<< ..., s >>>
        }
    }
}
```

Before Kepler

```
__global__ void quicksort(int *data, int left, int right)
{
    int nleft, nright;
    cudaStream_t s1, s2;

    // Partitions data based on pivot of first element.
    // Returns counts in nleft & nright
    partition(data+left, data+right, data[left], nleft, nright);

    // If a sub-array needs sorting, launch a new grid for it.
    // Note use of streams to get concurrency between sub-sorts
    if(left < nright) {
        cudaStreamCreateWithFlags(&s1, cudaStreamNonBlocking);
        quicksort<<< ..., s1 >>>(data, left, nright);
    }
    if(nleft < right) {
        cudaStreamCreateWithFlags(&s2, cudaStreamNonBlocking);
        quicksort<<< ..., s2 >>>(data, nleft, right);
    }
}

__host__ void launch_quicksort(int *data, int count)
{
    quicksort<<< ... >>>(data, 0, count-1);
}
```

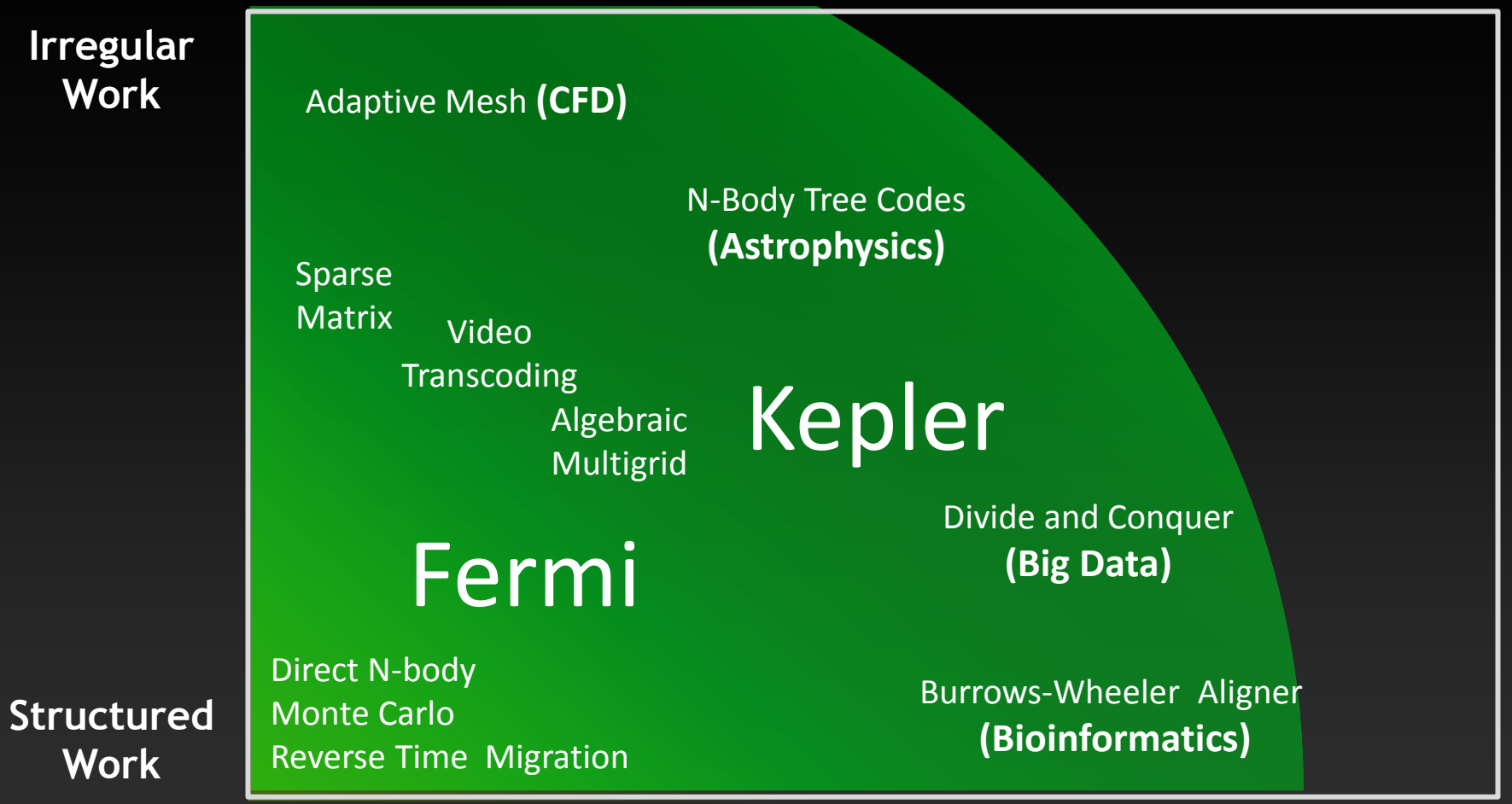
With Kepler

No complex CPU & GPU interaction

Easier code in half the lines

Easier porting for existing codes

2x More Applications, More Customers



Comparing GTX TITAN and Tesla K20X

Features	GeForce GTX TITAN	Tesla K20X
Core/Mem clock	837MHz/3GHz (clocks may vary when double precision is on)	732MHz/2.6GHz
Peak Single Precision	~4.5 Tflops	3.95 TFlops
Peak Double Precision	~1.27 Tflops (estimate)	1.32 TFlops
Memory size	6 GB	6 GB
Memory BW (ECC off)	288 GB/s	250 GB/s
PCIe	Gen 3 only on Ivy Bridge Gen 2 on Sandy Bridge	Gen 2
CUDA Features	Dynamic Parallelism, Hyper-Q For CUDA Streams GPUDirect Peer to Peer	Dynamic Parallelism, Hyper-Q Proxy for MPI and CUDA Streams, GPUDirect Peer to Peer, and RDMA
GPU monitoring	None	NVML/NVSMI, OOB, InfoROM, NVHealthmon, TCC
Cluster monitoring	None	Bright Computing, Ganglia
ECC Features	No ECC	DRAM, Internal Caches & Reg Files
Total Board Power	250W	235W

Tesla Advantage: Built for Deployment

Performance

- Fastest DP of 1.31TFLOPS on Tesla K20X
- Optimized for Infiniband with NVIDIA GPUDirect™
- Hyper-Q for accelerating MPI based workloads
- Tuning and optimization support from NVIDIA experts

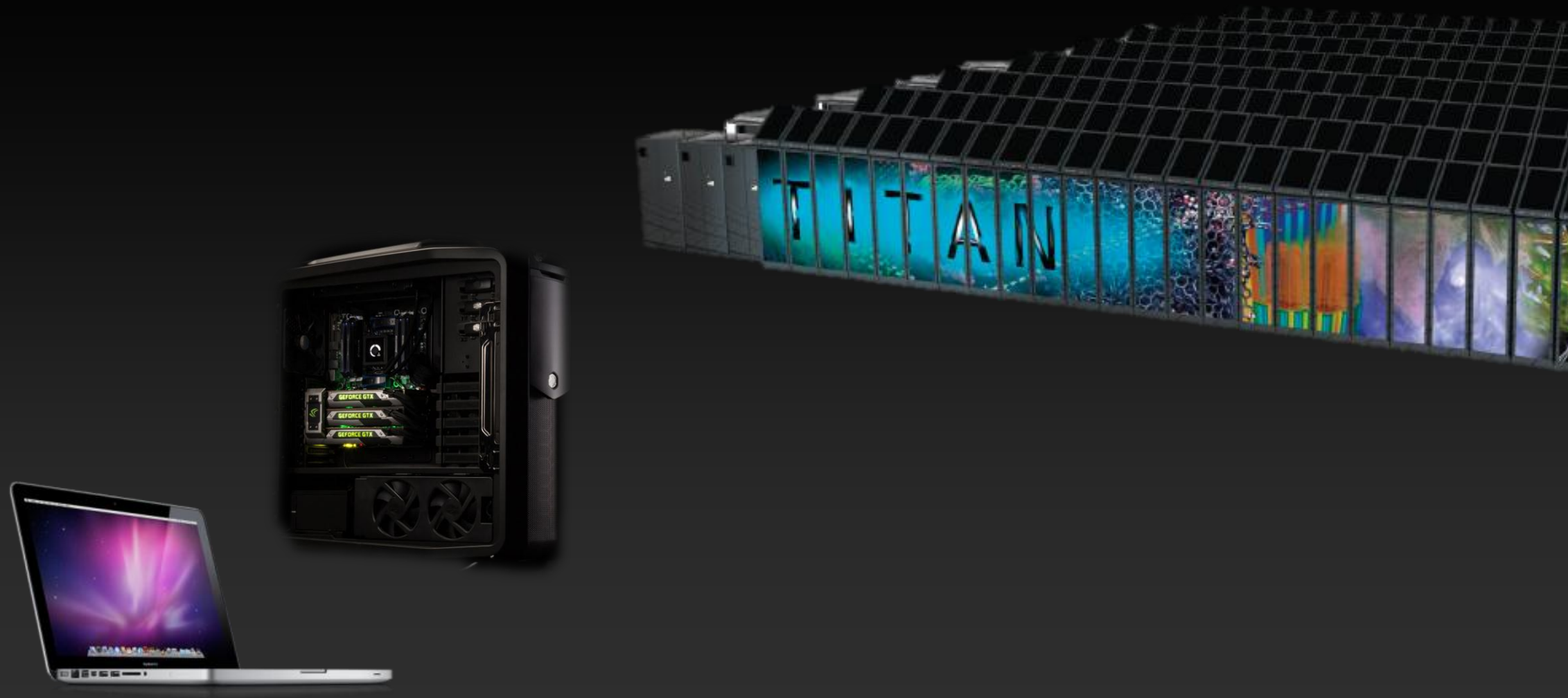
Reliability

- ECC protection
- Tested to run 24/7 with real-world workloads
- 3 year warranty and support for bugs/feature requests
- ISVs certify only on Tesla
- NVIDIA technical support
- Longer life cycle for continuity and cluster expansion

Built for HPC

- Integrated solutions & support from OEMs and channel
- Tools for GPU Management (Nvhealthmon, nvsmi/nvml)
- Tools for Cluster Management
- Enterprise OS support
- Solution expertise provided by CUDA engineers and technical staff
- Designed, tested and optimized for cluster deployment

Develop with GeForce, Deploy with Tesla



Call to Action

- Promote GTX Titan with OpenACC to new developers
  - Increases # of GPU enabled workloads
- Continue recommending Tesla for workloads in production environment
  - Tesla Advantage: Performance + Reliability + Built for HPC
- Refer to GTX Titan and Tesla FAQ for differentiation
  - GPUDirect, Hyper-Q, GPUBoost. GPUMonitoring


Thank You
NVIDIA