1st ASTERICS-OBELICS International School

6-9 June 2017, Annecy, France.

H2020-Astronomy ESFRI and Research Infrastructure Cluster
(Grant Agreement number: 653477).

GPU Programming

Valeriu Codreanu

SURFsara

Outline

Yesterday's lecture

- Introduction to the GPU ecosystem
- The GPU HW architecture
- GPU programming
- GPUs & High-performance Libraries
- GPU Debugging & Profiling
- GPUs & Python

Today's lecture

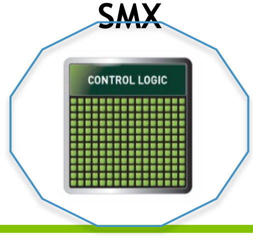
- Volta architectural features
- Unified memory
- Multi-GPU/GPUDirect RDMA

CUDA Parallel Computing Platform

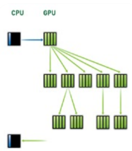
Programming Approaches	Libraries "Drop-in" Acceleration	OpenACC Directives Easily Accelerate Apps	Programming Languages Maximum Flexibility
Development Environment		Nsight IDE Linux, Mac and Windows GPU Debugging and Profiling	CUDA-GDB debugger NVIDIA Visual Profiler
Open Compiler Tool Chain		Enables compiling new languages to CUDA platform, and CUDA languages to other architectures	

Hardware Capabilities

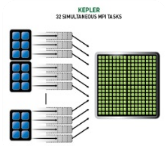
SMX



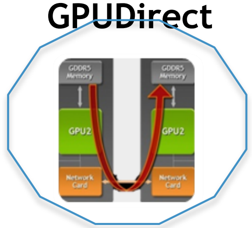
Dynamic Parallelism



HyperQ



GPUDirect



NVIDIA Volta GV100 architecture

21B transistors
815 mm2

80 SM5120 CUDA Cores
640 Tensor Cores

16 GB HBM2
900 GB/s HBM2
300 GB/s NVLink

NVIDIA Volta GV100 SM architecture

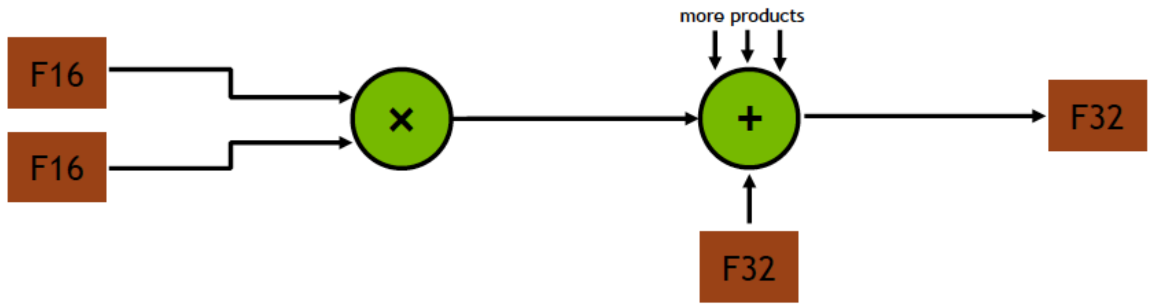
Tensor Core

$$D = \underbrace{\begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix}}_{\text{FP16}} \underbrace{\begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix}}_{\text{FP16}} + \underbrace{\begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}}_{\text{FP16 or FP32}}$$

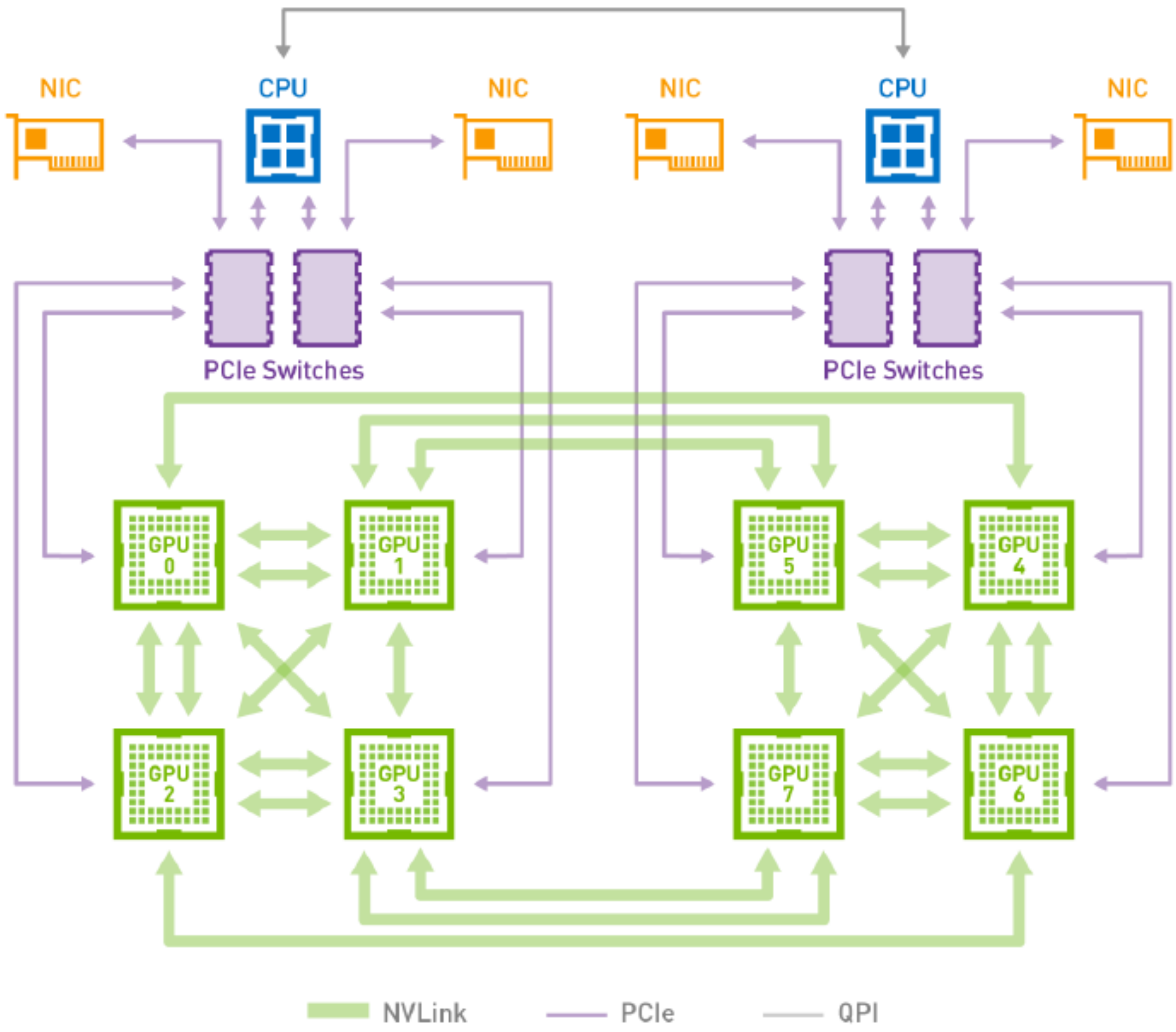
FP16 or FP32

$$D = AB + C$$

FP16 storage/input | **Full precision product** | **Sum with FP32 accumulator** | **Convert to FP32 result**



Volta NVLink architecture

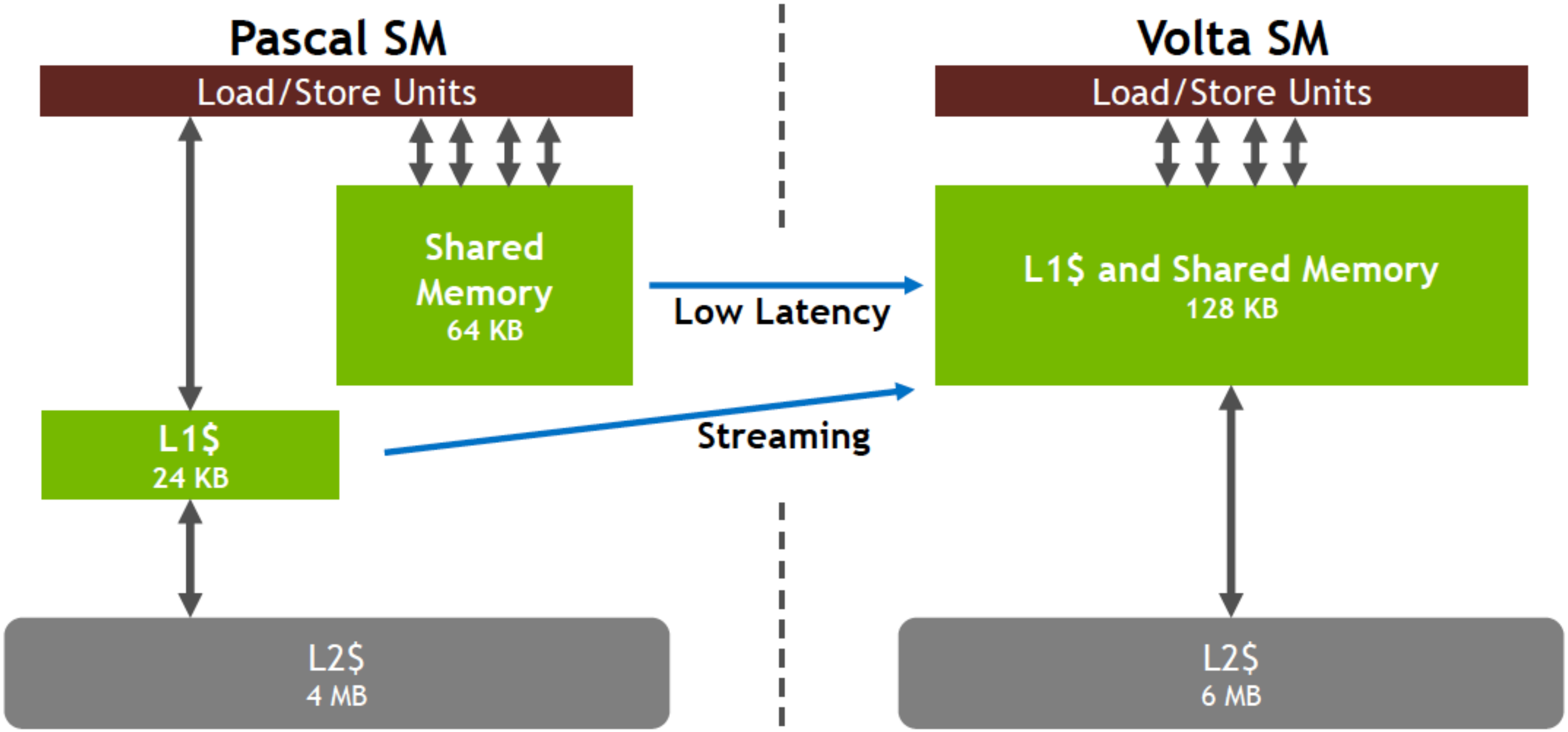


300GB/sec

50% more links

28% faster links

Volta SM memory hierarchy



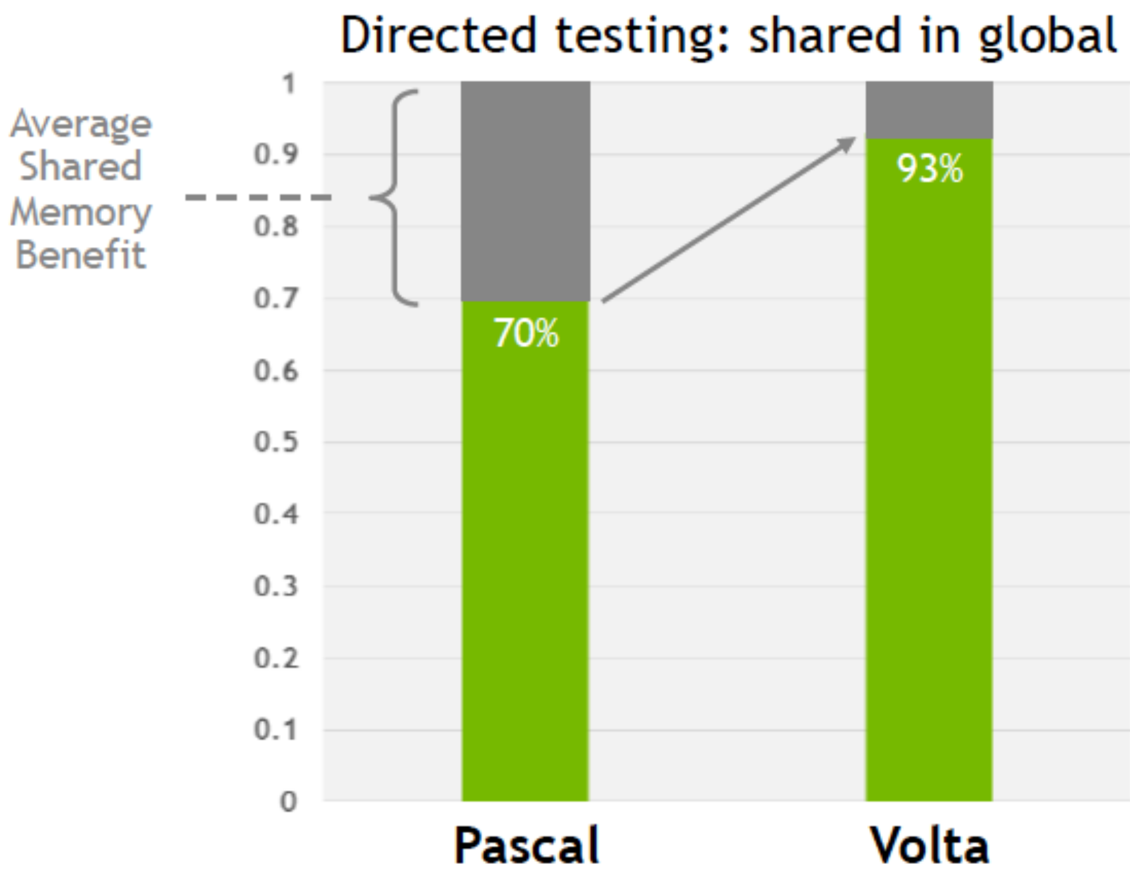
Increasing (easily) achievable performance

Cache: vs shared

- Easier to use
- 90%+ as good

Shared: vs cache

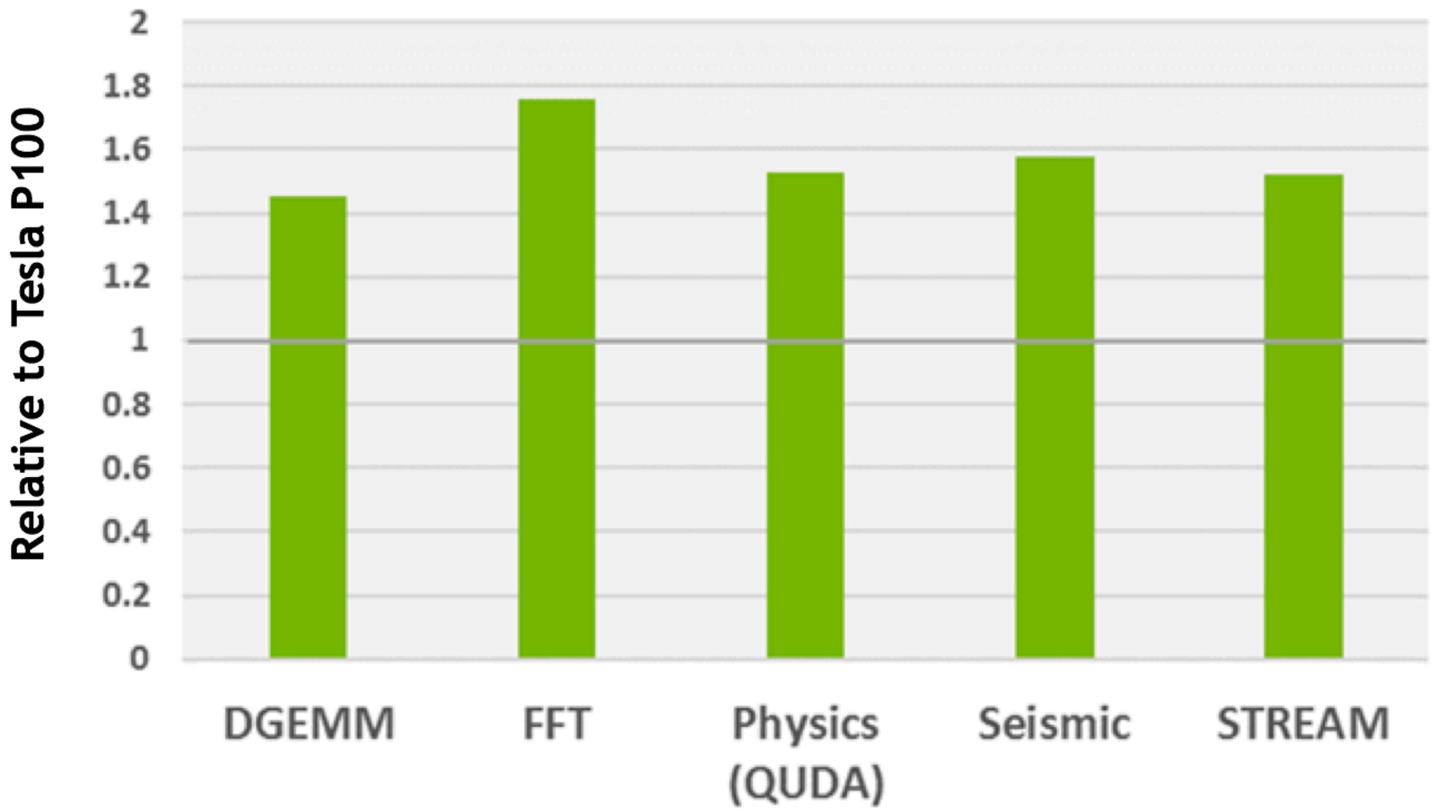
- Faster atomics
- More banks
- More predictable



Volta performance specs

	P100	V100	Ratio
Training acceleration	10 TOPS	120 TOPS	12x
Inference acceleration	21 TFLOPS	120 TOPS	6x
FP64/FP32	5/10 TFLOPS	7.5/15 TFLOPS	1.5x
HBM2 Bandwidth	720 GB/s	900 GB/s	1.2x
NVLink Bandwidth	160 GB/s	300 GB/s	1.9x
L2 Cache	4 MB	6 MB	1.5x
L1 Caches	1.3 MB	10 MB	7.7x

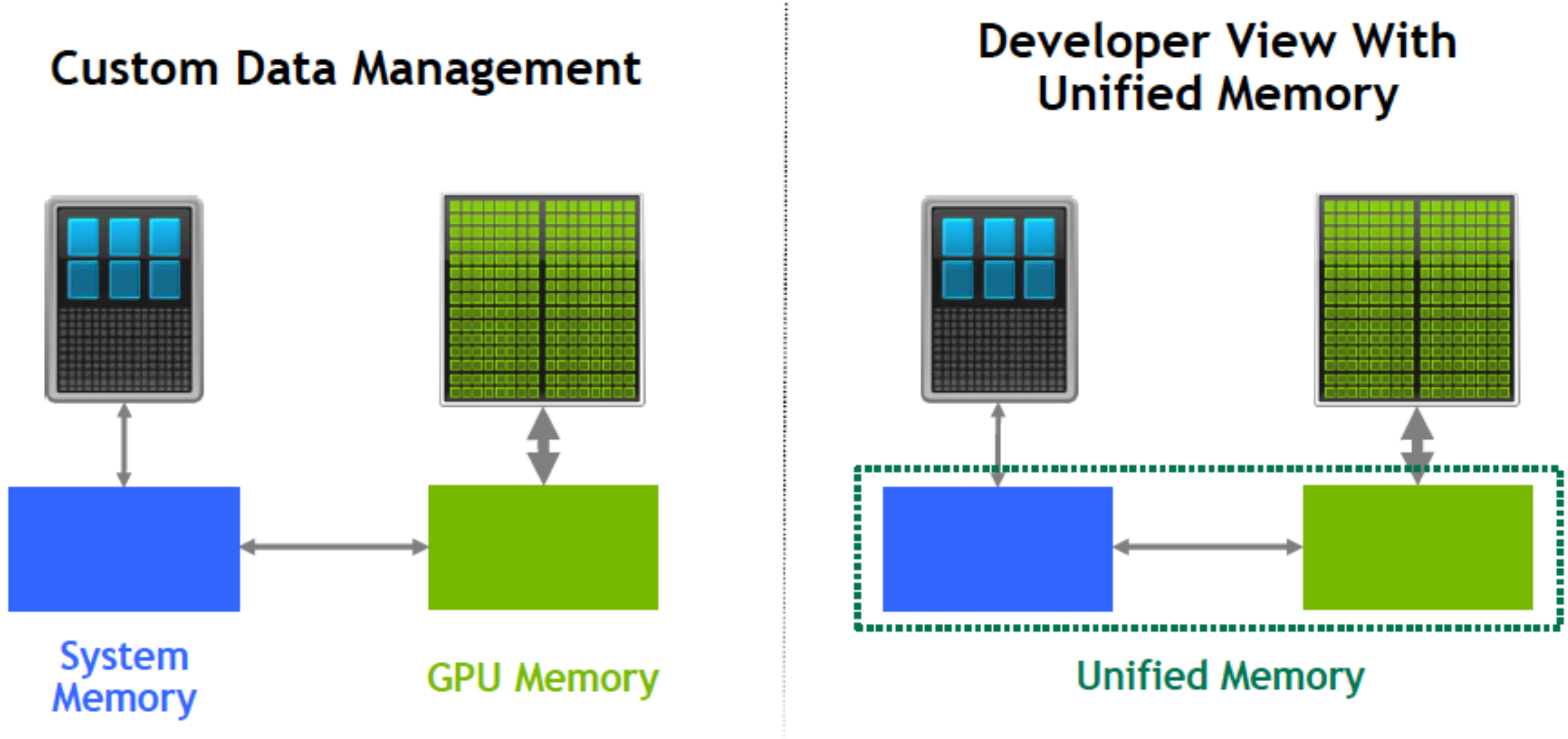
Volta HPC Application Performance



System Config Info: 2X Xeon E5-2690 v4, 2.6GHz, w/ 1X Tesla P100 or V100. V100 measured on pre-production hardware.

Unified Memory

Unified memory



Unified memory programming

CPU code

```
void sortfile(FILE *fp, int N) {
  char *data;
  data = (char *)malloc(N);

  fread(data, 1, N, fp);

  qsort(data, N, 1, compare);


  use_data(data);

  free(data);
}
```

GPU code with Unified Memory

```
void sortfile(FILE *fp, int N) {
  char *data;
  cudaMallocManaged(&data, N);

  fread(data, 1, N, fp);

  qsort<<<...>>>(data,N,1,compare);
  cudaDeviceSynchronize();

  use_data(data);

  cudaFree(data);
}
```

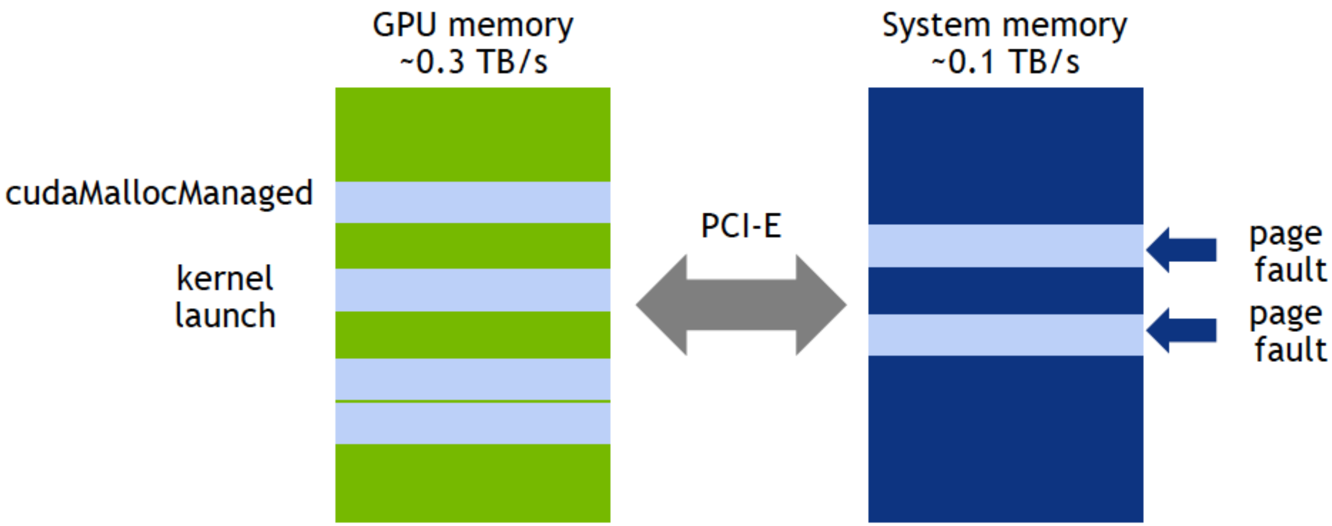
Unified memory execution

```
cudaMallocManaged(&ptr, ...);
*ptr = 1;
qsort<<<...>>>(ptr);
```

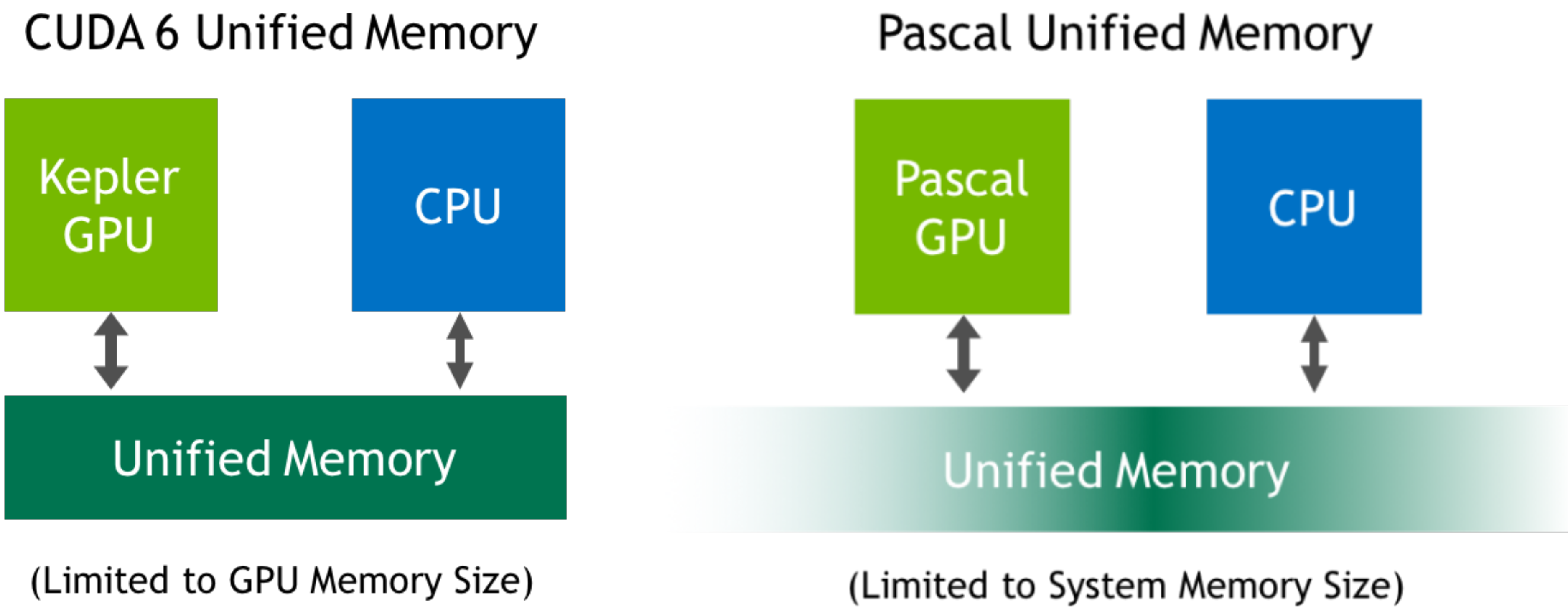
← Pages are populated in GPU memory

← **CPU page fault**: data migrates to CPU

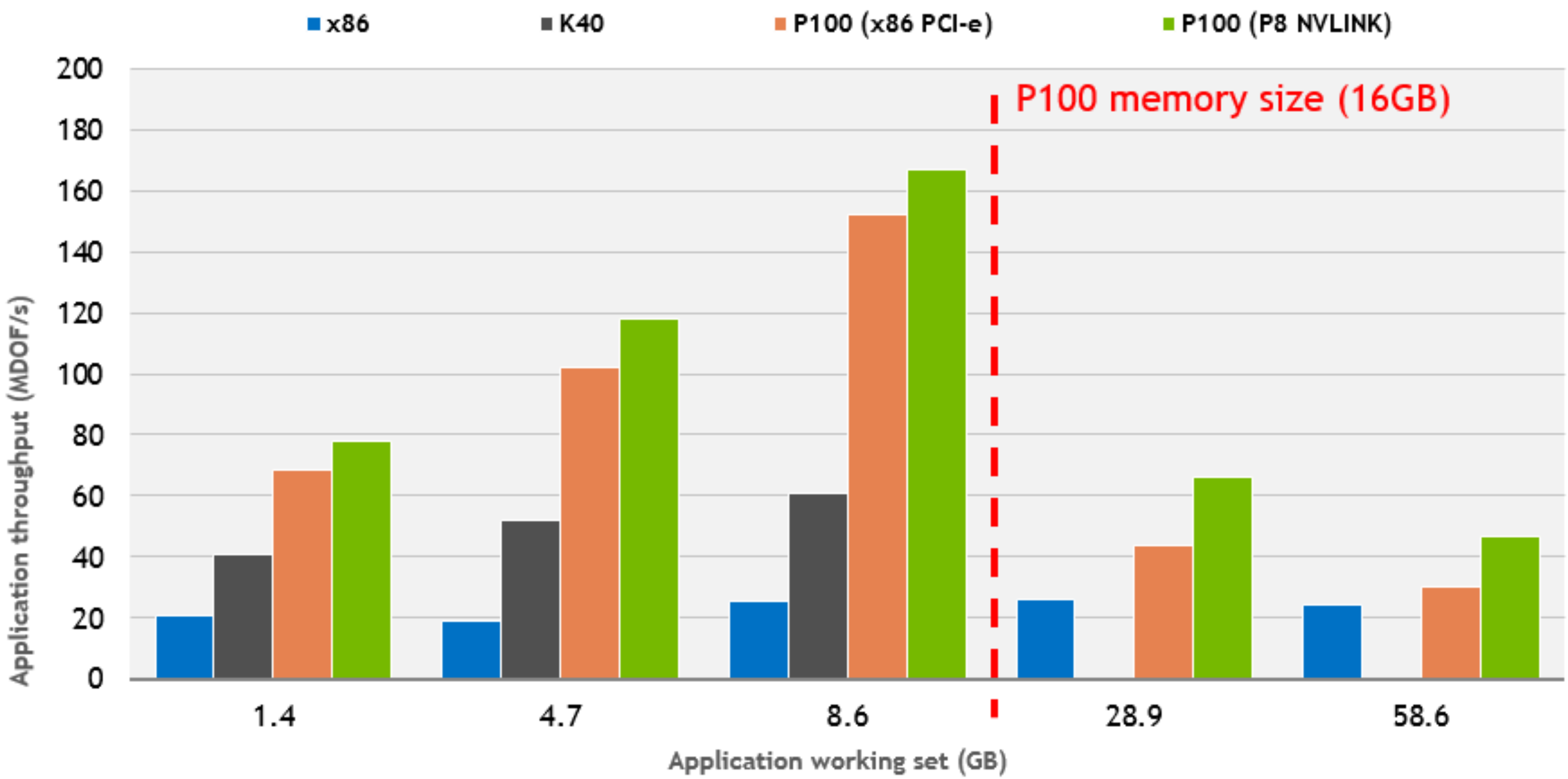
← **Kernel launch**: data migrates to GPU



Unified memory oversubscription



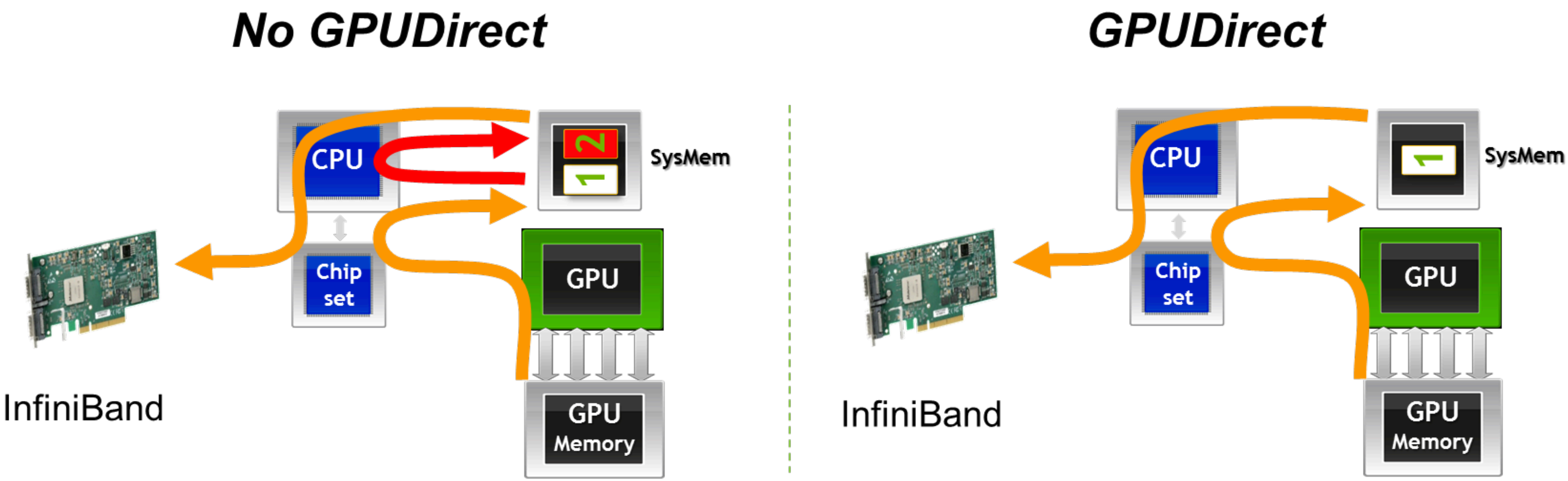
Unified memory oversubscription (Pascal)



HPGMG AMR proxy performance

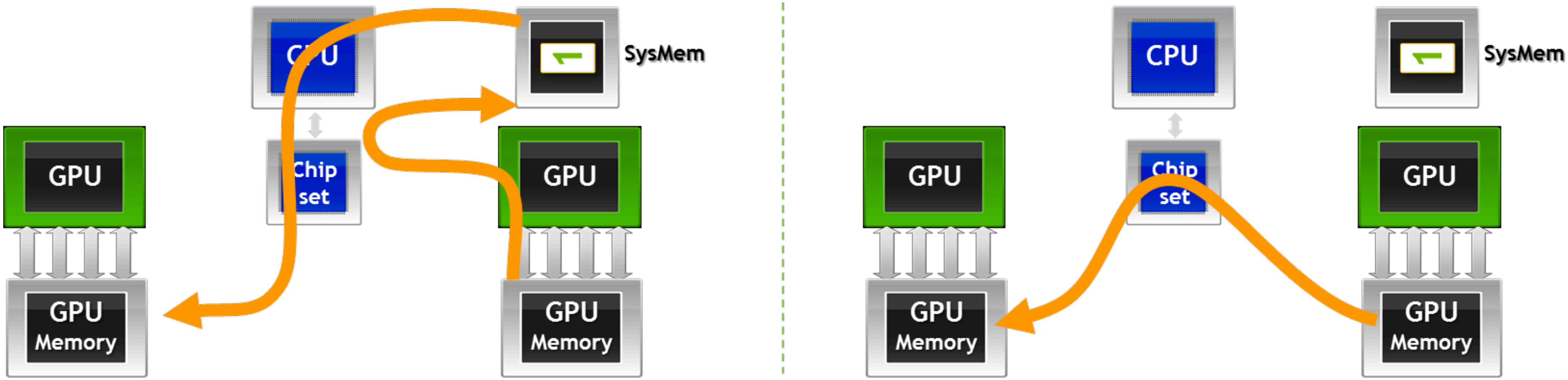
Multi-GPU programming

GPUDirect (CUDA3.1)



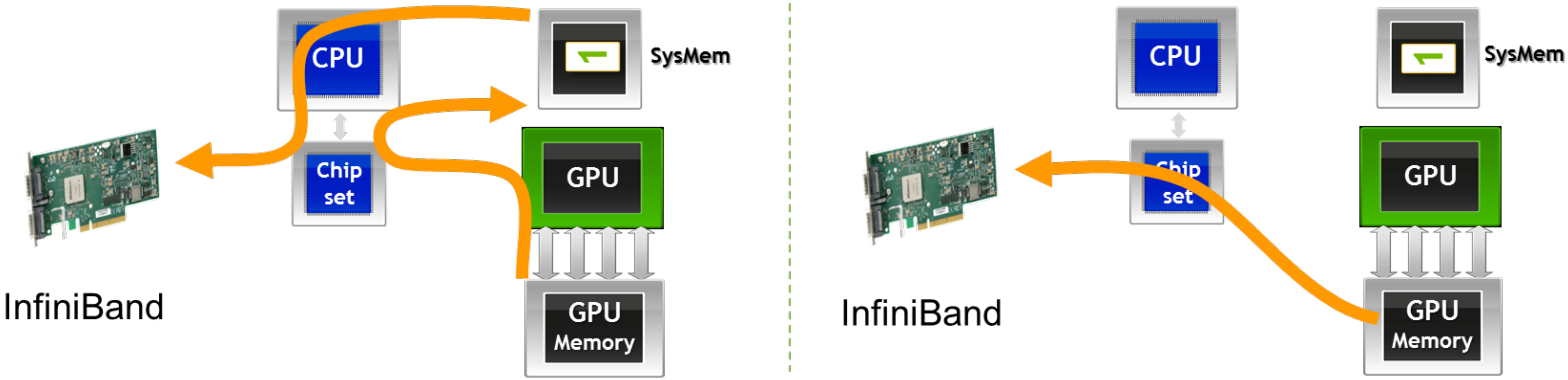
GPUDirect P2P (CUDA 4)

No GPUDirect P2P | **GPUDirect P2P**



GPUDirect RDMA (CUDA 5)

No GPUDirect RDMA | **GPUDirect RDMA**



GPUDirect RDMA example

```
//without GPUDirect RDMA
//MPI rank 0
cudaMemcpy(s_buf_h,s_buf_d,size,cudaMemcpyDeviceToHost);
MPI_Send(s_buf_h,size,MPI_CHAR,1,100,MPI_COMM_WORLD);

//MPI rank 1
MPI_Recv(r_buf_h,size,MPI_CHAR,0,100,MPI_COMM_WORLD, &status);
cudaMemcpy(r_buf_d,r_buf_h,size,cudaMemcpyHostToDevice);



//with GPUDirect RDMA
//MPI rank 0
MPI_Send(s_buf_d,size,MPI_CHAR,1,100,MPI_COMM_WORLD);

//MPI rank 1
MPI_Recv(r_buf_d,size,MPI_CHAR,0,100,MPI_COMM_WORLD, &status);
```

Next-gen systems

TITAN VS SUMMIT

Compute System Comparison

ATTRIBUTE	TITAN	SUMMIT
Compute Nodes	18,688	~3,400
Processor	(1) 16-core AMD Opteron per node	(Multiple) IBM POWER 9s per node
Accelerator	(1) NVIDIA Kepler K20x per node	(Multiple) NVIDIA Volta GPUs per node
Memory per node	32GB (DDR3)	>512GB (HBM+DDR4)
CPU-GPU Interconnect	PCI Gen2	NVLINK (5-12x PCIe3)
System Interconnect	Gemini	Dual Rail EDR-IB (23 GB/s)
Peak Power Consumption	9 MW	10 MW

Peak node performance: > 40TFlops vs 1.4 TFlops
Peak system performance: 150-300 PFlops vs. 20 PFlops

THANK YOU FOR YOUR ATTENTION

www.prace-ri.eu

Acknowledgement

H2020-Astronomy ESFRI and Research Infrastructure Cluster (Grant Agreement number: 653477).