Computational Science on Many-Core Architectures

360.252

Karl Rupp

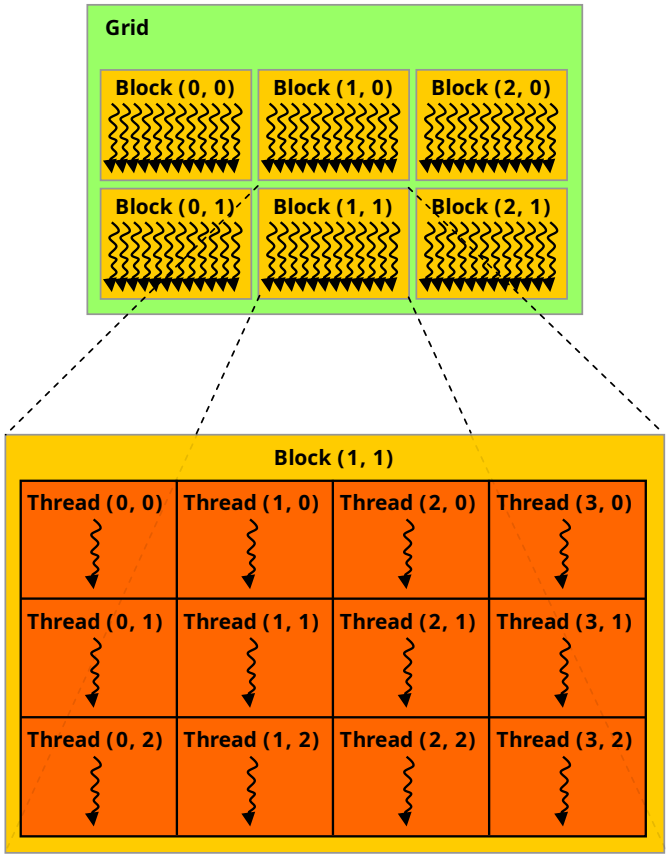
Institute for Microelectronics, TU Wien
http://www.iue.tuwien.ac.at/

Zoom Channel 941 8518 8102
Q&A on Wednesday, October 19, 2022

Thread Indexing

2D Indexing

- Optional feature to organize threads in multiple index dimensions
- Convenience feature, no performance difference
- Indexing within blocks can be different from indexing of blocks in grid



https://upload.wikimedia.org/wikipedia/commons/5/5b/Block-thread.svg

Thread Indexing

Example: Adding Matrices

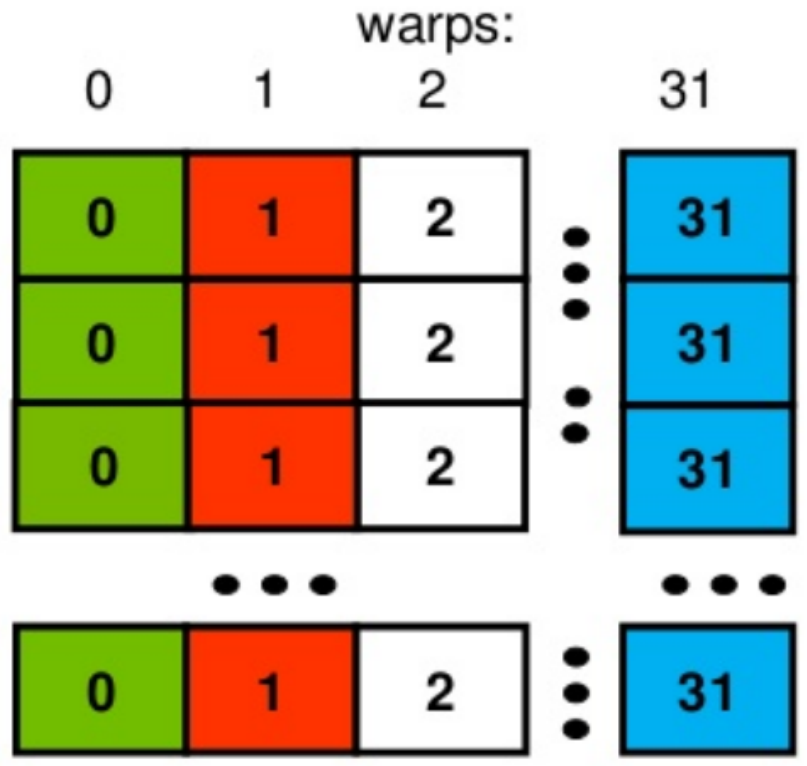
```
__global__
void add_matrices(double *A, double *B, double *C, int N, int M)
{
  int x = blockIdx.x * blockDim.x + threadIdx.x; // row index
  int y = blockIdx.y * blockDim.y + threadIdx.y; // column index
  int idx = y * M + x;  // global index in row-major matrix

  if (x < M && y < N)
    A[idx] = B[idx] + C[idx];
}

int main() {
  double *d_A, *d_B, d_C; // row-major matrices on GPU, N x M
  ...
  dim3 block(4,16);          // 4 threads per blocks in x-dimension,
                             // 16 threads per blocks in y-dimension
  dim3 grid(M/4+1,N/16+1); // enough blocks in x- and y-dimension
  add_matrices<<<grid,block>>> (d_A,d_B, d_C, N, M);
  ...
}
```

Shared Memory: Banking

Note on Shared Memory Banks

- Shared memory is organized in 32 banks of 32/64 bits each

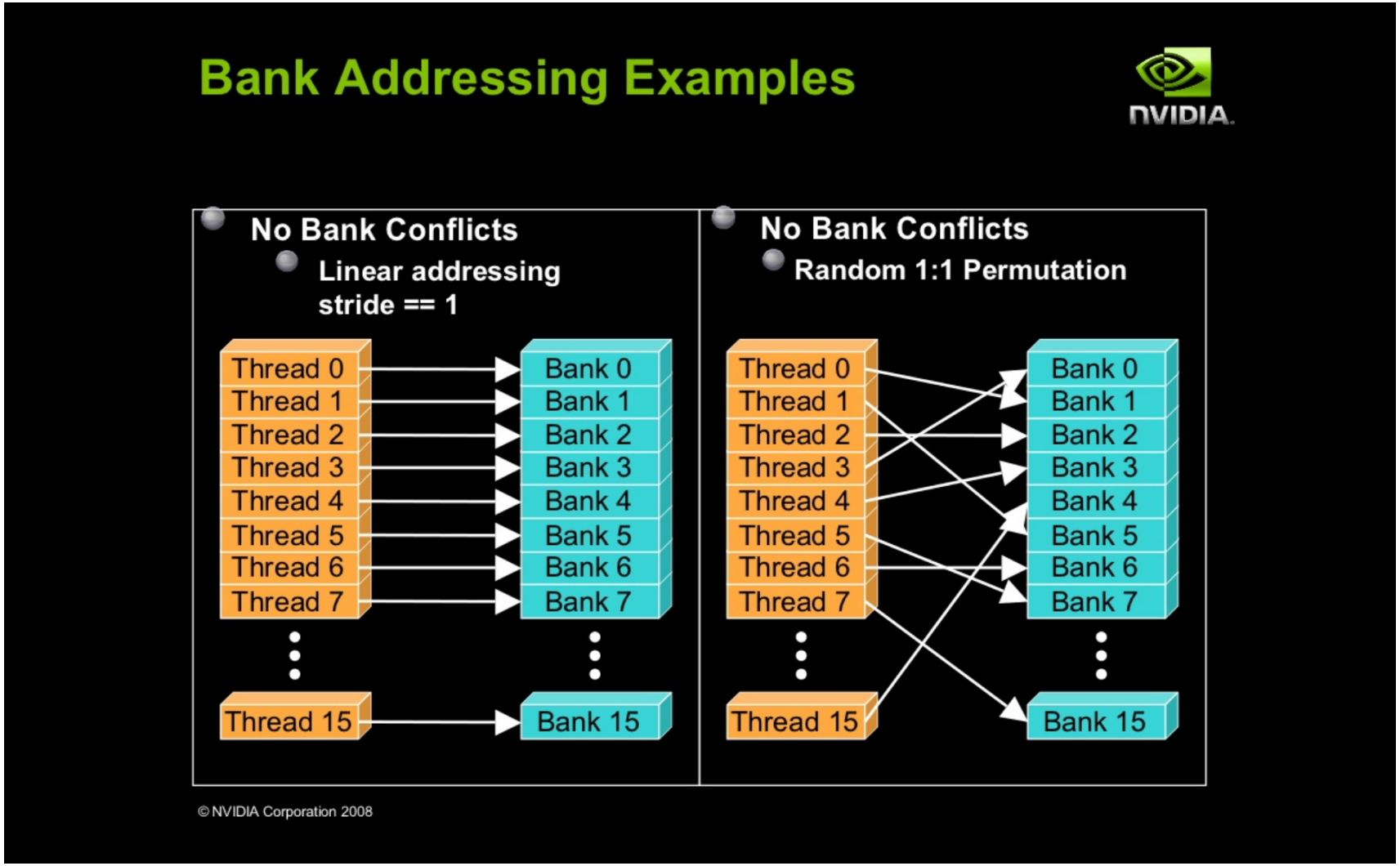
Bank 0
Bank 1
...
Bank 31



Shared Memory: Banking

The Good

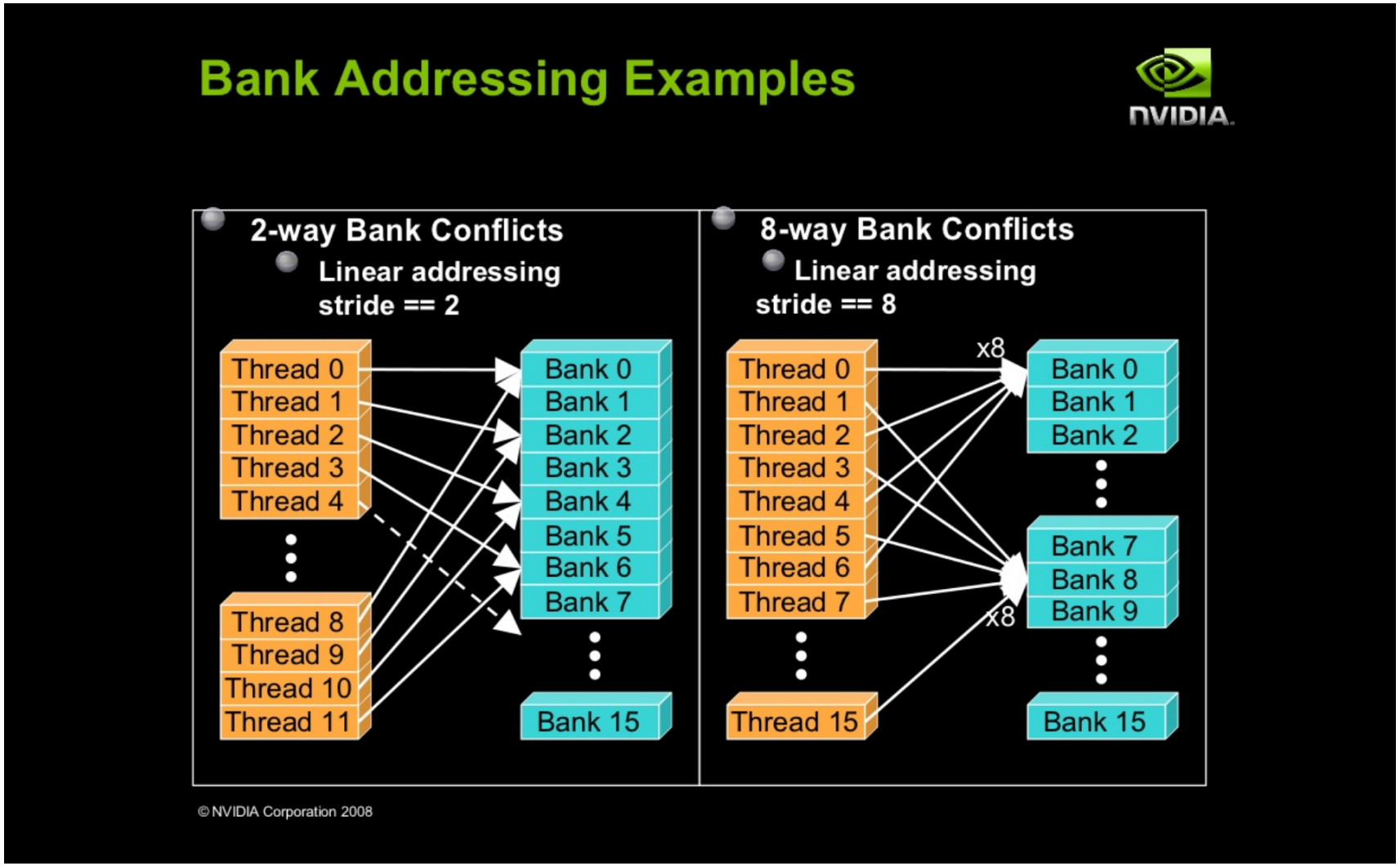
- Access to different banks is parallel



Shared Memory: Banking

The Bad

- Access to the same bank is serialized



Shared Memory: Banking

A Workaround: Padding

- Instead of

```
__shared__ double tile[TILE_DIM][TILE_DIM];
```

use

```
__shared__ double tile[TILE_DIM][TILE_DIM+1];
```