SPADE: A Flexible and Scalable Accelerator for SpMM and SDDMM

Gerasimos Gerogiannis, Serif Yesil*, Damitha Lenadora, Dingyuan Cao, Charith Mendis and Josep Torrellas

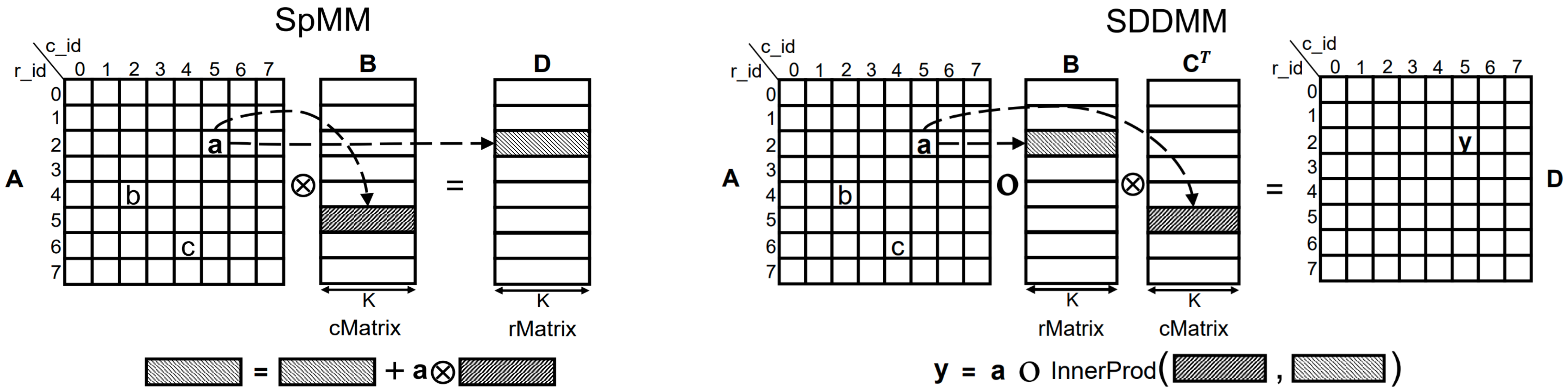
University of Illinois at Urbana-Champaign

*Now at NVIDIA.

SpMM and SDDMM

- Two important operations in sparse matrix computations:
  - SpMM: Sparse Matrix – Dense Matrix Multiplication
  - SDDMM: Sampled Dense Matrix – Dense Matrix Multiplication
- Applications in machine learning, graph neural networks (GNNs), atmospheric modeling, aerodynamic design, linear algebra solvers …
- Unique mixture of sparse and dense operands
- Heavily memory-bound for real-world graphs

SpMM and SDDMM



- The non-zeros of the sparse matrix drive the accesses to the dense matrices and lead to irregular access patterns
- Reuse of dense rows depends on sparsity pattern

Pitfalls in Designing an Accelerator for SpMM and SDDMM

- Low arithmetic intensity makes host-accelerator data transfers and virtual address remapping very costly (>97% of execution time) ☹

- Current accelerator designs are inflexible and fail to adapt to varying sparsity patterns of the input matrix (e.g., road graphs, social network graphs, scientific graphs) ☹

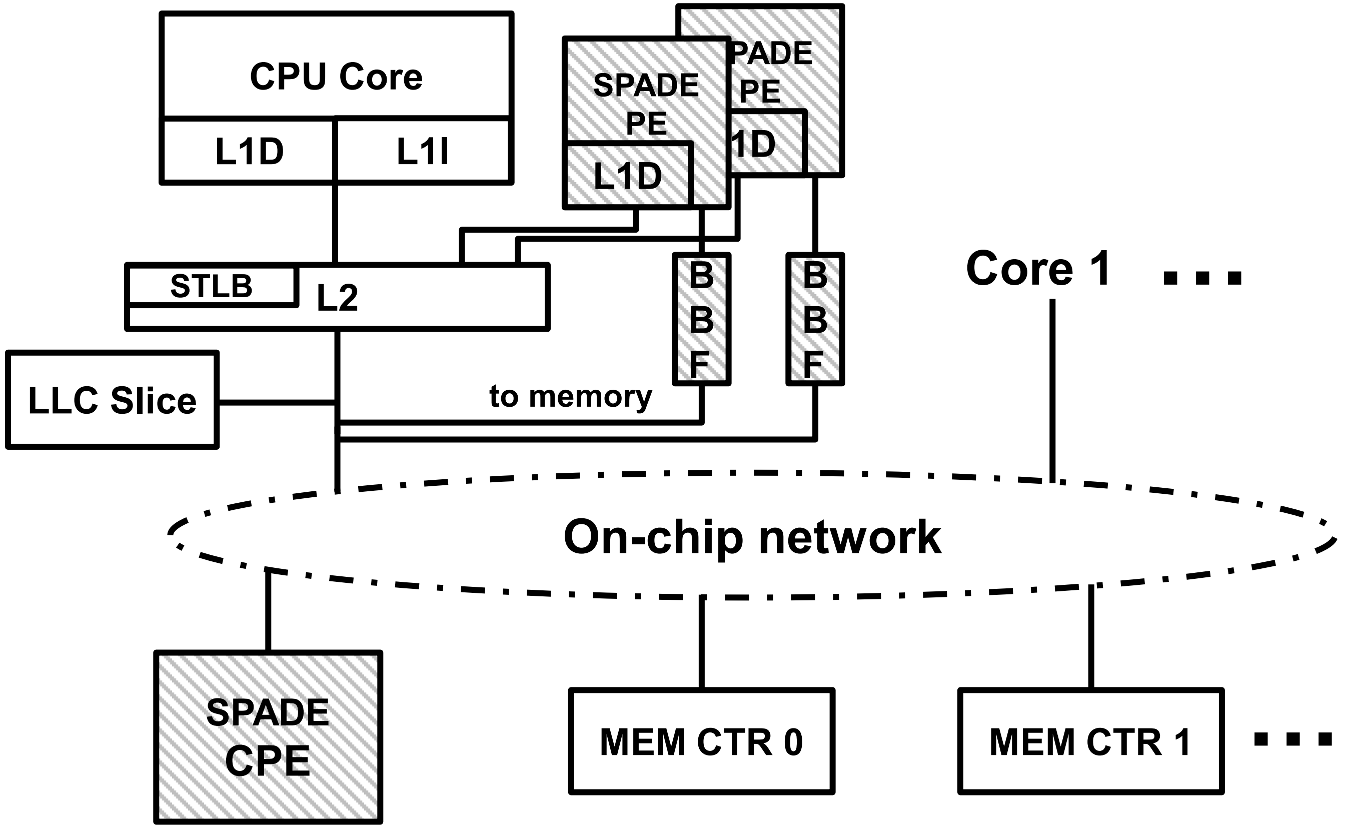
Addressing these Pitfalls with SPADE

- To eliminate host-accelerator data transfer and address remapping cost:
  ✓ An accelerator architecture tightly integrated with the cores of a CPU multicore

- To accommodate varying sparsity patterns:
  ✓ A high-level *Tile ISA* and various flexibility knobs in the architecture

- To accommodate the memory-bound nature of the problems and scale-up:
  ✓ An accelerator pipeline designed for latency tolerance

Tight integration with the cores of a CPU multicore



- SPADE PEs:
  - Decoupled, out-of-order vector engines
  - Reuse a core's L2, LLC and STLB
- SPADE CPE:
  - Control engine that assigns tiles to PEs
- BBF:
  - Bypass Buffer: allows for optional cache bypassing
- Low area cost:
  - All SPADE hardware: 2.5% of host area

SPADE ←→ CPU mode transitions

- Cache hierarchy state and STLB entries are reused in place

- Especial cache operations during SPADE ←→ CPU mode transitions

  - SPADE mode → CPU mode:
    SPADE L1 caches and BBFs are written back (to L2 and memory) and invalidated

  - CPU mode → SPADE mode:
    CPU L1 caches are written back to L2 and invalidated
    Data that the upcoming SPADE cycle will access through BBFs is written back to memory and inv

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

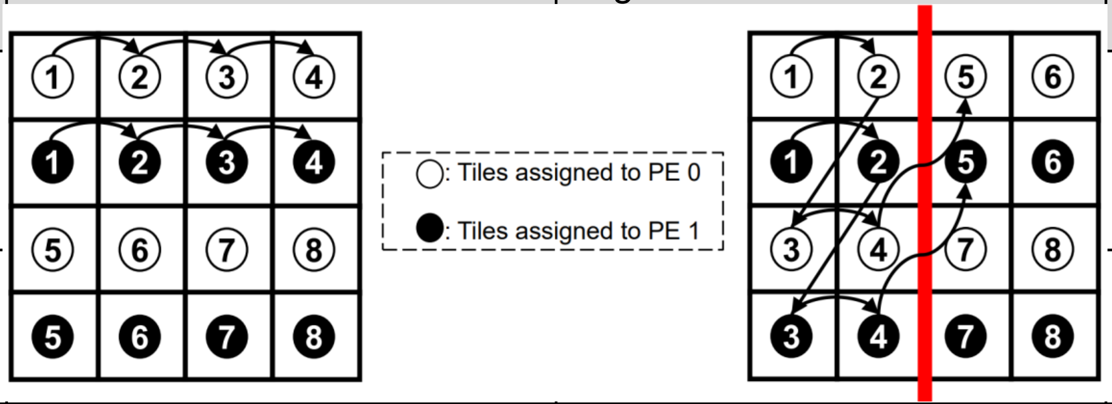
Tile ISA

Instructions issued by the CPE to the PEs

Instruction	What it does	Arguments	Notes
Initialization	Initializes the PEs	Operation to perform (SpMM or SDDMM), base virtual addresses of sparse and dense matrices etc.	Broadcasted to all PEs
Tile operation	Executes SpMM or SDDMM on a tile of the sparse matrix	Tile information such as location of first non-zero and number of non-zeros in the tile	Assigned to a single PE
Scheduling Barrier	Pauses tile scheduling by the CPE until all previous tiles have been completed		Times the tile execution for data reuse and to limit cache pressure
WriteBack&Invalidate	Informs the PEs to write-back and invalidate L1s and BBFs		Broadcasted to all PEs
Termination	Signals the termination of the SPADE mode execution		Broadcasted to all PEs

Tile ISA

Instructions issued by the CPE to the PEs

Instruction	What it does	Arguments	Notes
Initialization			Broadcasted to all PEs
Tile operation			Assigned to a single PE
Scheduling Barrier	Pauses tile scheduling by the CPE until all previous tiles have been completed		Times the tile execution for data reuse and to limit cache pressure
WriteBack&Invalidate	Informs the PEs to write-back and invalidate L1s and BBFs		Broadcasted to all PEs
Termination	Signals the termination of the SPADE mode execution		Broadcasted to all PEs

SPADE is designed for flexibility

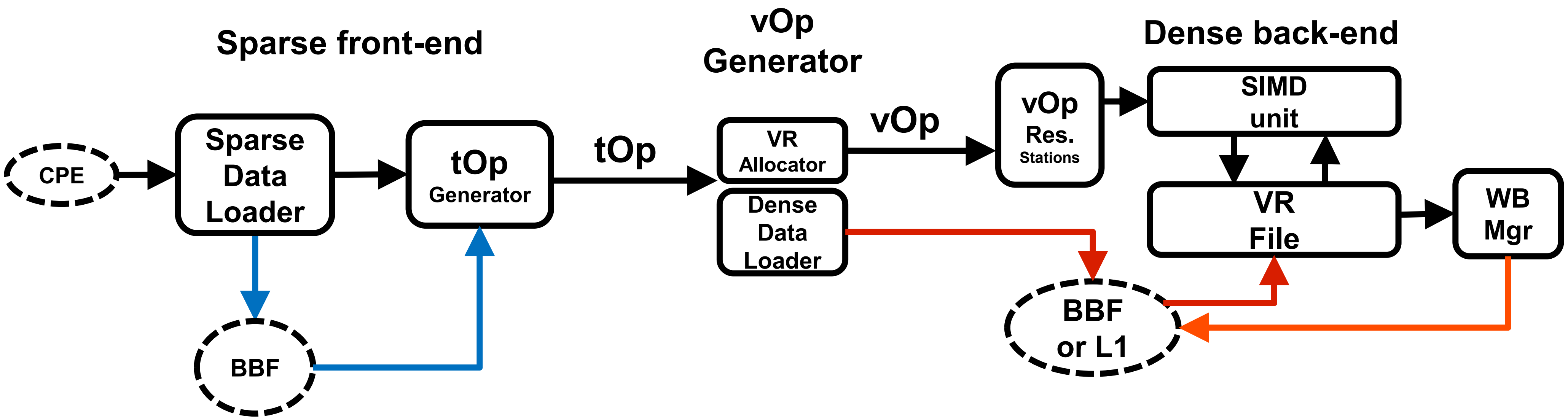
- ✓ Accepts tiles of any size
- ✓ Dense structures can optionally bypass the caches
- ✓ Scheduling barriers change timing of tile scheduling for best cache use

- Each of these knobs is tuned based on the input matrix:
  - Depending on the input sparse matrix, bypassing can increase runtime by up to 170% or decrease it by up to 33%
  - Depending on the input sparse matrix, barriers can increase runtime by up to 80% or decrease it by up to 57%

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

SPADE pipeline

Sparse requests overlap with dense requests and computation for latency tolerance



- Sparse Data Loader: issues read requests for sparse data
- tOp Generator: generates one tuple operation (tOp) per non-zero
- VR Allocator: allocates VRs and breaks down tOps in vector-sized operations (vOps)
- Dense Data Loader: issues read requests for dense data
- vOp Reservation Stations: support OoO execution in back-end
- WB Manager: periodically writes back Vector Register to the memory subsystem

Evaluation

- Simulation-based evaluation using SST and DRAMSim3
- Baselines:
  - 56-core Intel Icelake CPU
  - NVIDIA V100 GPU
  - Scaled-up idealized version of the *Sextans* SpMM accelerator
- Benchmarks: 10 large graphs from SparseSuite
- Prototyped a simplified SPADE in a chip and taped it out using TSMC 65nm

Evaluation Highlights

- SPADE delivers high average speedups:
  - Over a 56-core Intel Icelake CPU: 2.3x
  - Over an NVIDIA V100 GPU:
    - 1.3x without considering host-GPU data transfer overhead
    - 43.4x considering data transfer overhead
  - Over an idealized scaled-up *Sextans* SpMM accelerator
    - 2.5x without considering host-accelerator data transfer overhead
    - 52.4x considering data transfer overhead
- Scales well from 224 to 1792 PEs

Conclusion

- SPADE is an SpMM/SDDMM accelerator tightly integrated in a CPU multicore
- Eliminates host-accelerator data transfer and address remapping overheads
- Provides architectural flexibility knobs to exploit diverse sparsity patterns
- Delivers substantial speedups over CPUs, GPUs and other accelerators at a low area and power cost

SPADE: A Flexible and Scalable Accelerator for SpMM and SDDMM

Gerasimos Gerogiannis, Serif Yesil*, Damitha Lenadora, Dingyuan Cao, Charith Mendis and Josep Torrellas

University of Illinois at Urbana-Champaign

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

*Now at NVIDIA.