A Compilation Framework for Irregular Memory Accesses on the Cell Broadband Engine

MAINAK CHAUDHURI

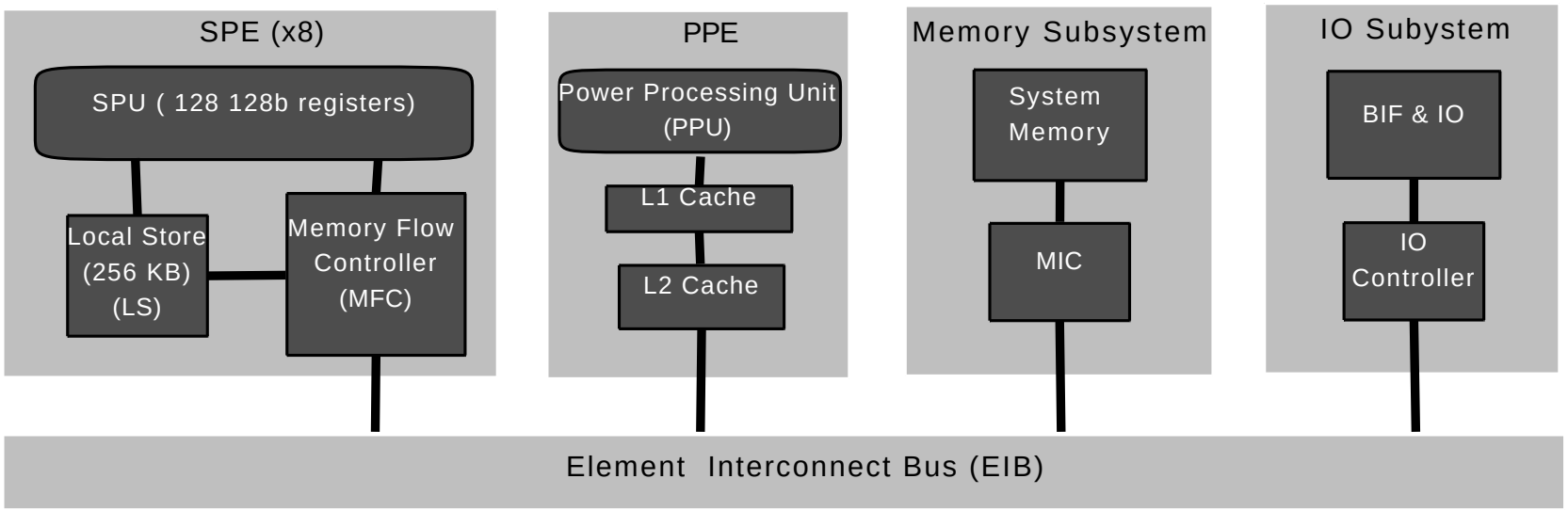
Department of Computer Science and Engineering, IIT Kanpur

Pramod Bhatotia – IBM Research
Sanjeev Aggarwal – IIT Kanpur

Outline

Cell Broadband Engine Overview

Cell Architecture



The Cell Broadband Engine

Compiling Irregular Accesses on the Cell Processor

Irregular Array Accesses

Irregular Array Access

An array access is irregular if no closed-form expression, in terms of the loop indices and constants, for the subscripts of the accessed array is available at compile-time.

```
do t =
  do i = num_edges
    n1 = left[i]
    n2 = right[i]
    force = (x[n1] − x[n2])/4
    y[n1] += force
    y[n2] += force
```

Compiling Irregular Accesses on the Cell Processor

Irregular Array Accesses

Irregular Array Access

An array access is irregular if no closed-form expression, in terms of the loop indices and constants, for the subscripts of the accessed array is available at compile-time.

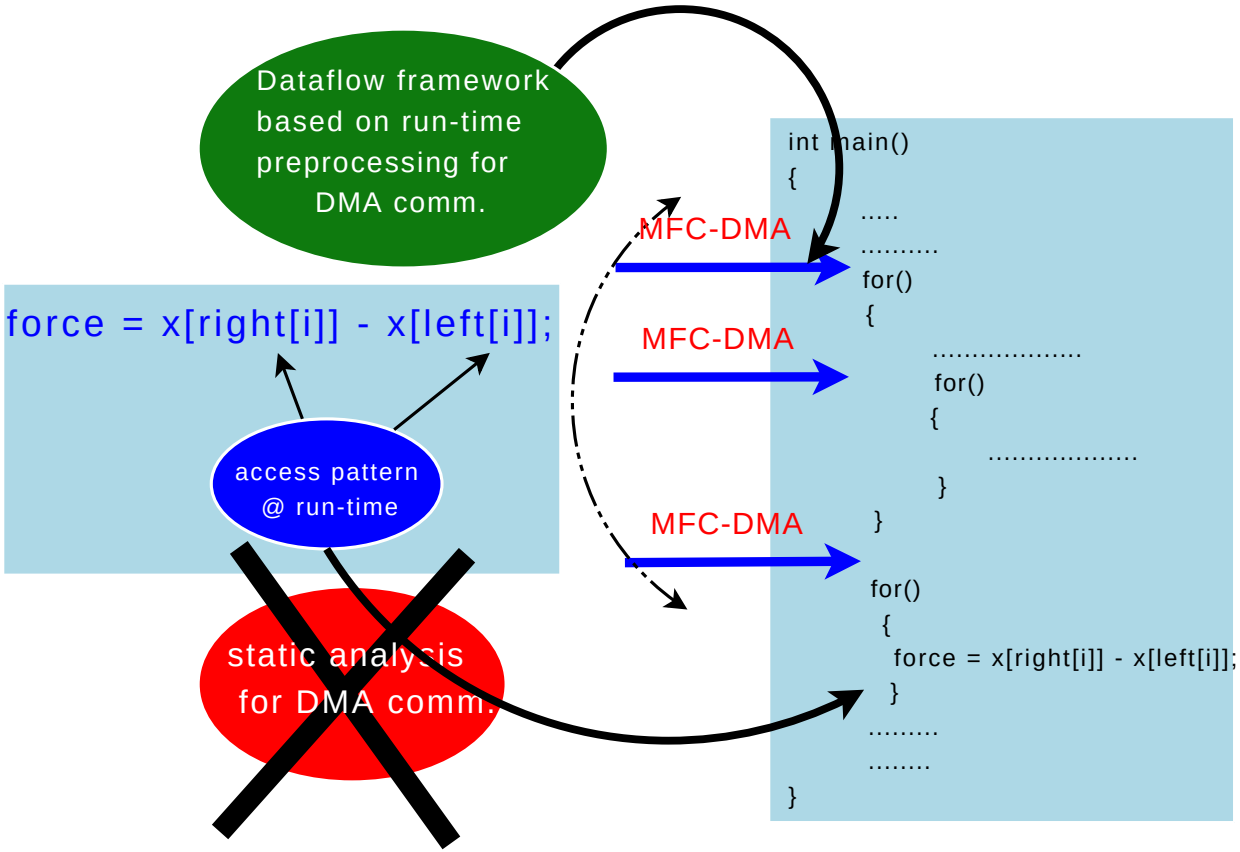
```
do t =
  do i = num_edges
    n1 = left[i]
    n2 = right[i]
    force = (x[n1] - x[n2])/4
    y[n1] += force
    y[n2] += force
```

Compiling Irregular Accesses

Compiling Irregular Accesses

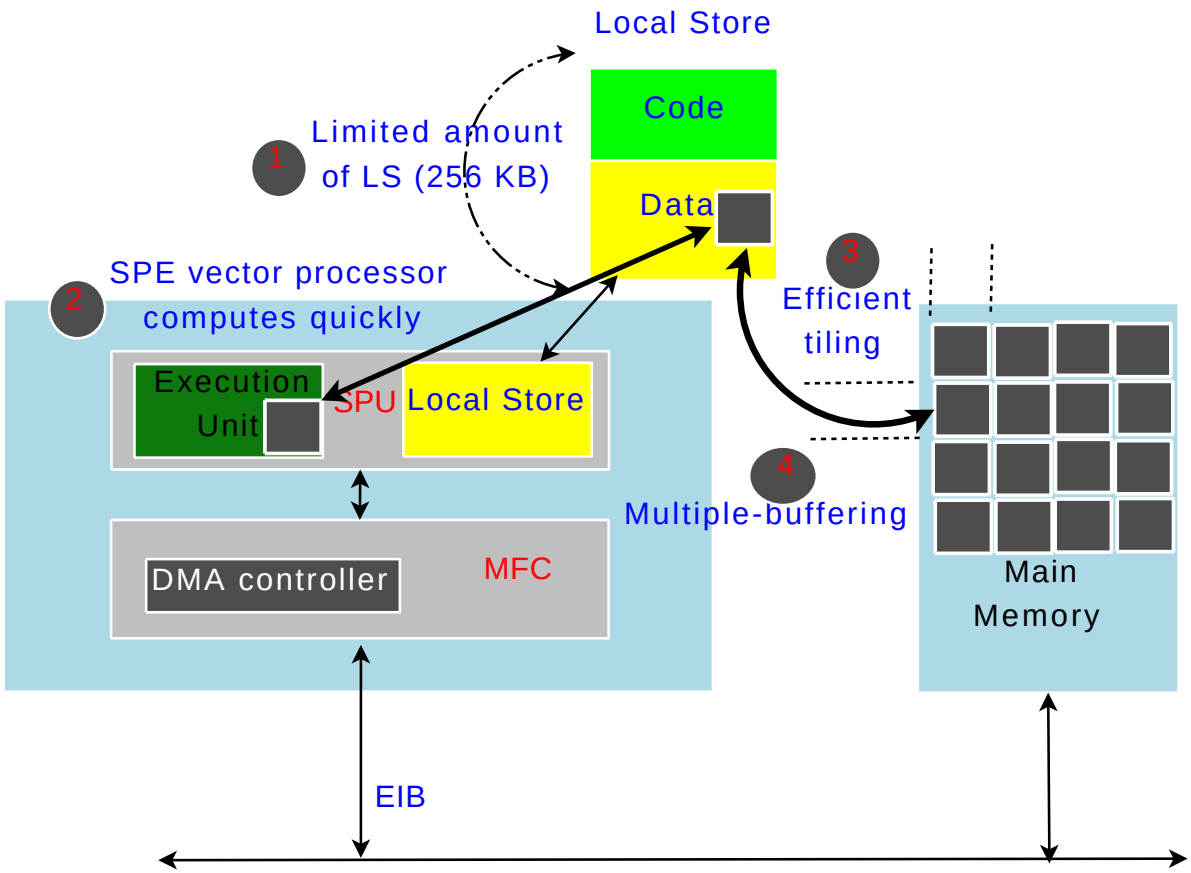
Compiling Irregular Accesses on the Cell Processor

Compiler Analysis for Irregular Array Accesses



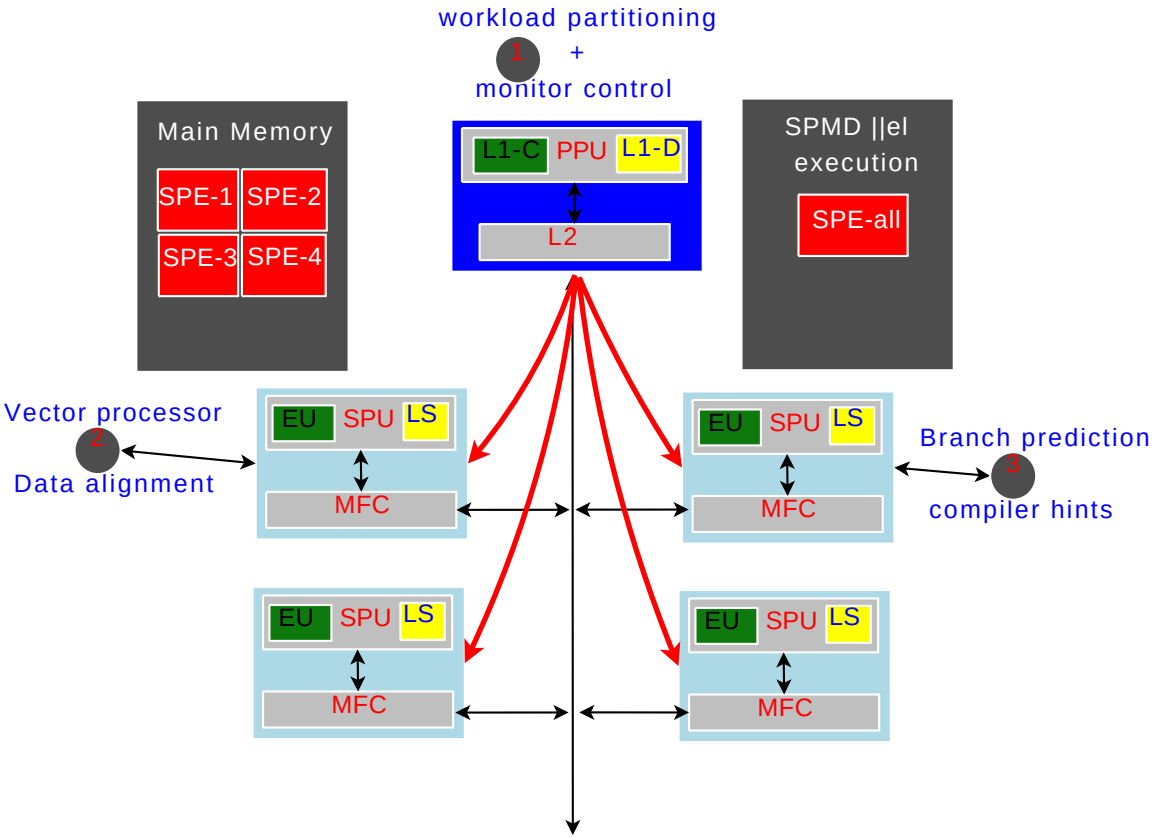
Compiling Irregular Accesses on the Cell Processor

Explicit Dynamic Memory Management



Compiling Irregular Accesses on the Cell Processor

Cell Architecture Specific Challenges



Need for Run-time Parallelization

Sparse Updates

- *Sparse update refers to reduction operations on some elements of an array within a loop where the access pattern of the array in the loop may be irregular.*

$$for(i = 1\ to\ n)$$
$$A[B[i]] = A[B[i]] + X;$$

Flow

S1: A(B(i))=...
S2: ...=A(B(i))

Anti

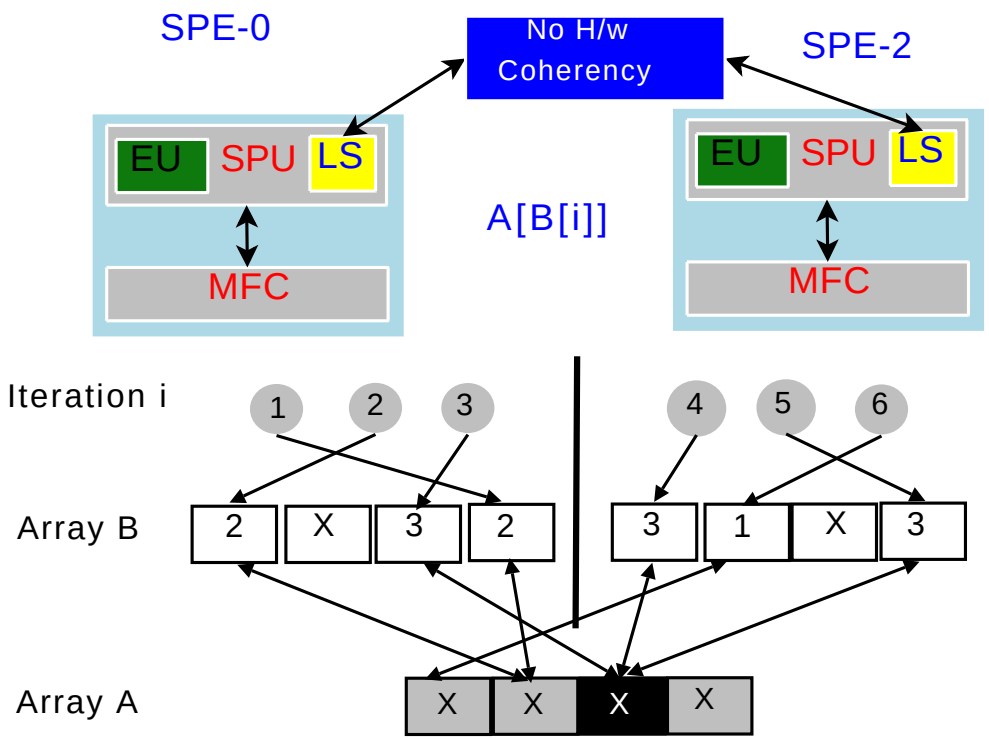
S1: ...=A(B(i))
S2: A(B(i))=...

Output

S1: A(B(i))=...
S2: A(B(i))=...

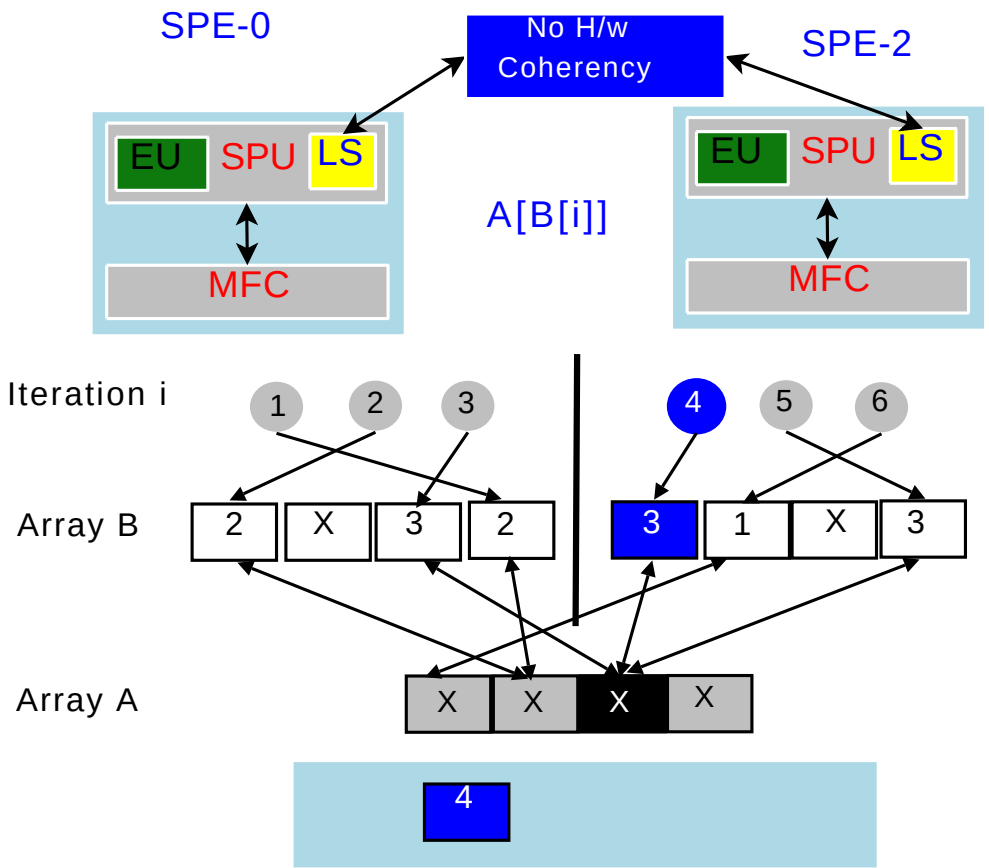
Compiling Irregular Accesses on the Cell Processor

Run-time Parallelization



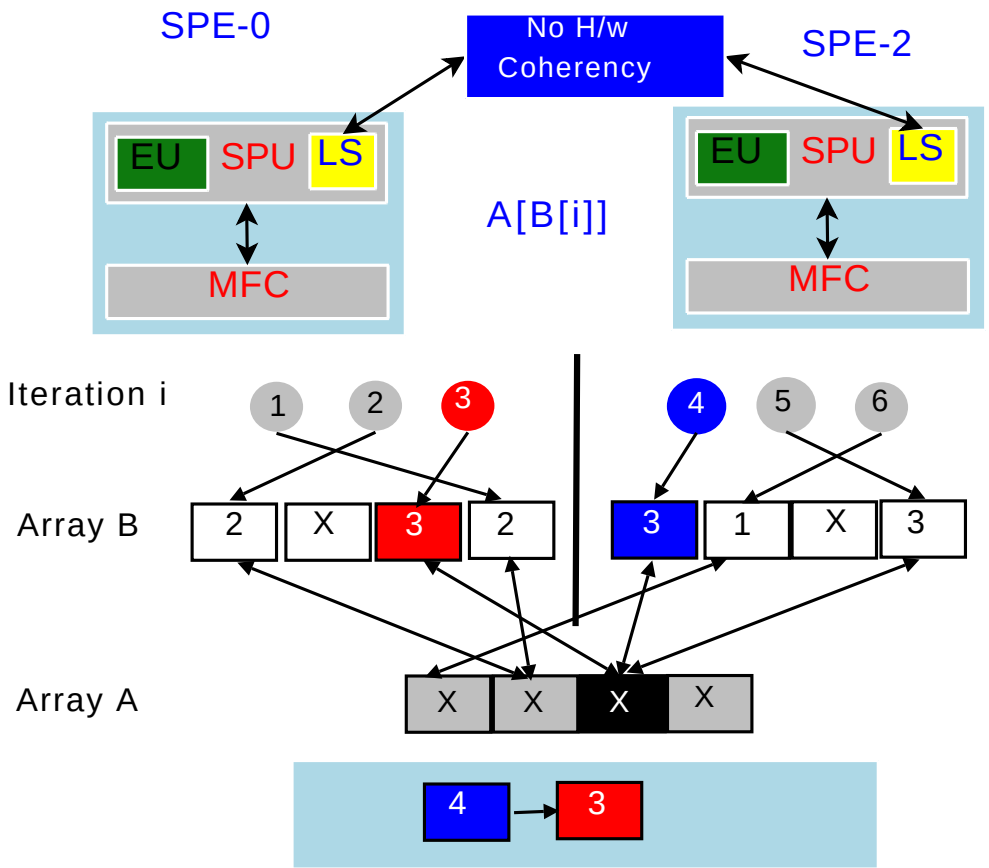
Compiling Irregular Accesses on the Cell Processor

Run-time Parallelization



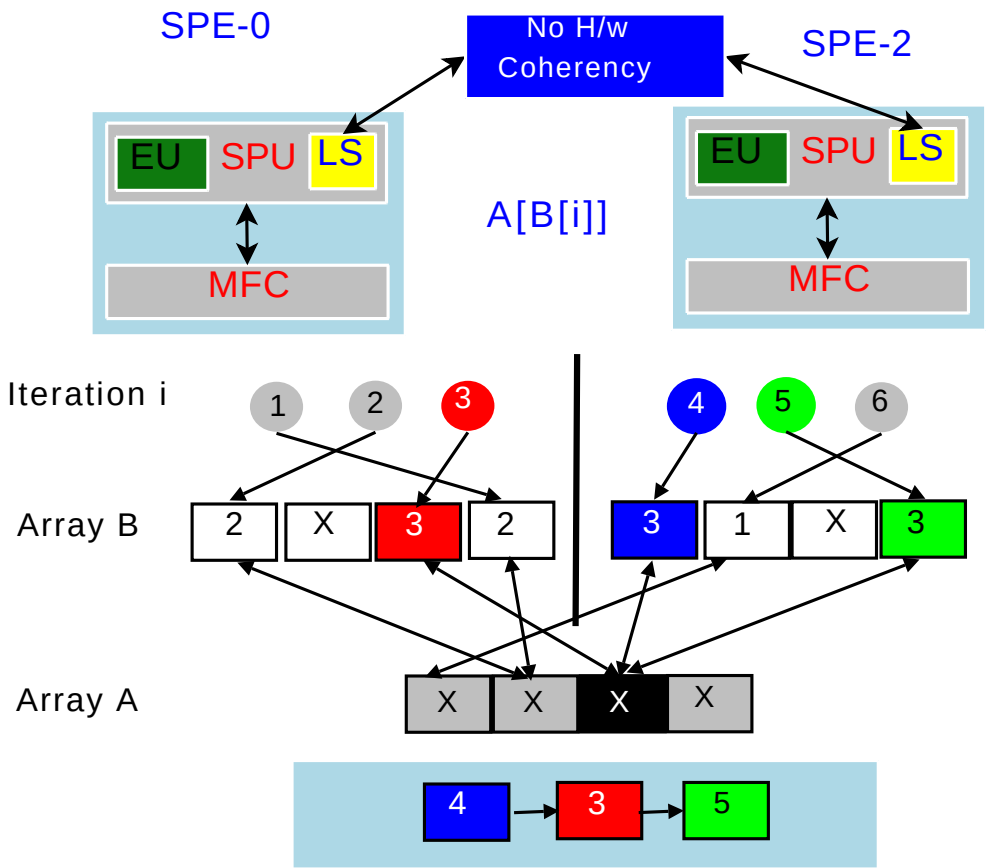
Compiling Irregular Accesses on the Cell Processor

Run-time Parallelization

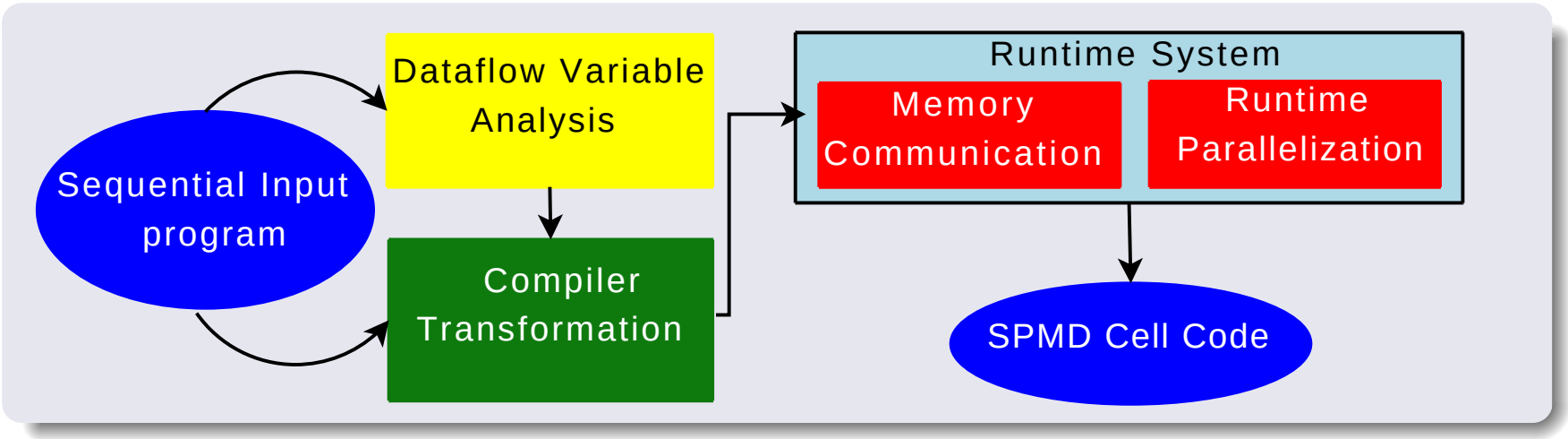


Compiling Irregular Accesses on the Cell Processor

Run-time Parallelization



Overview of the System



Parallelizing Compiler Framework

Dataflow Analysis

Dataflow Analysis

Dataflow Analysis for Memory Communication

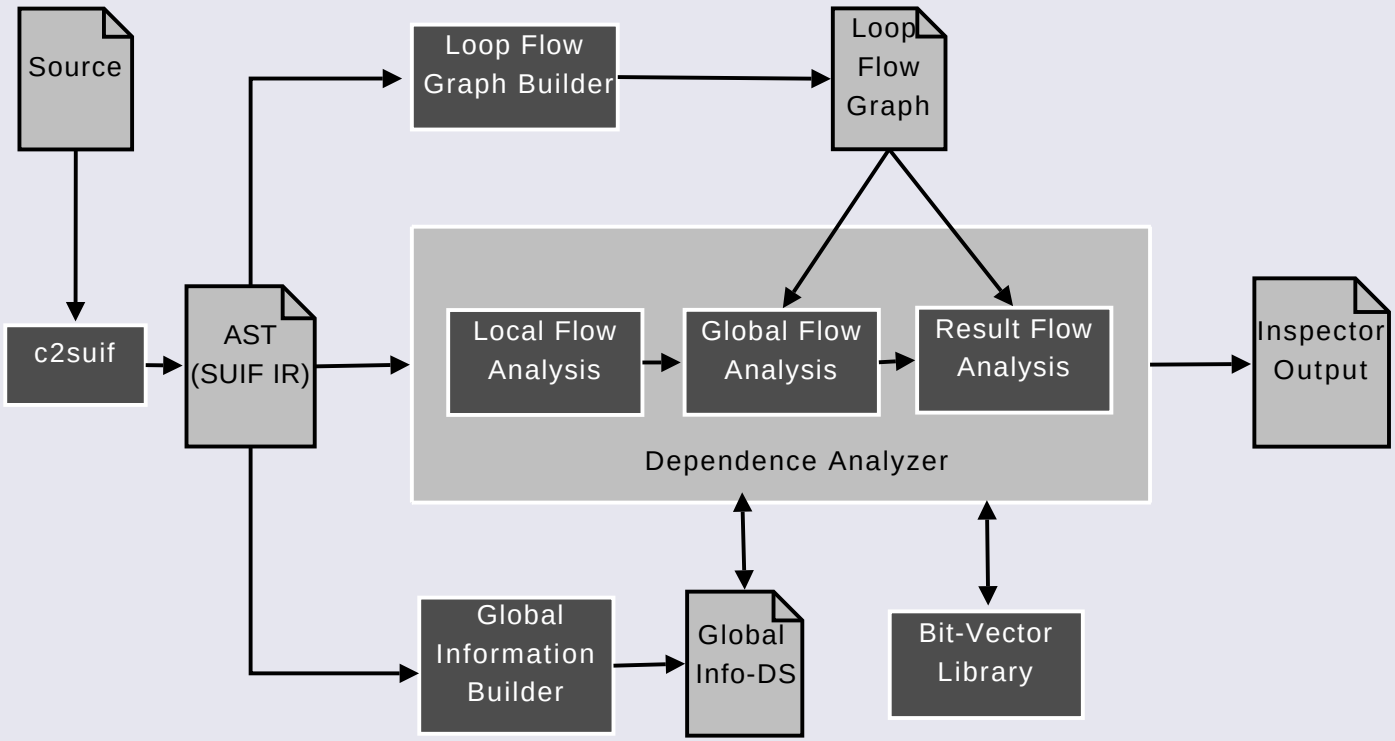


Figure: Dataflow analysis for memory communication

Compiler Transformations and Code Generation

Compiler Transformations and Code Generation

Compiler Transformations and Code Generation

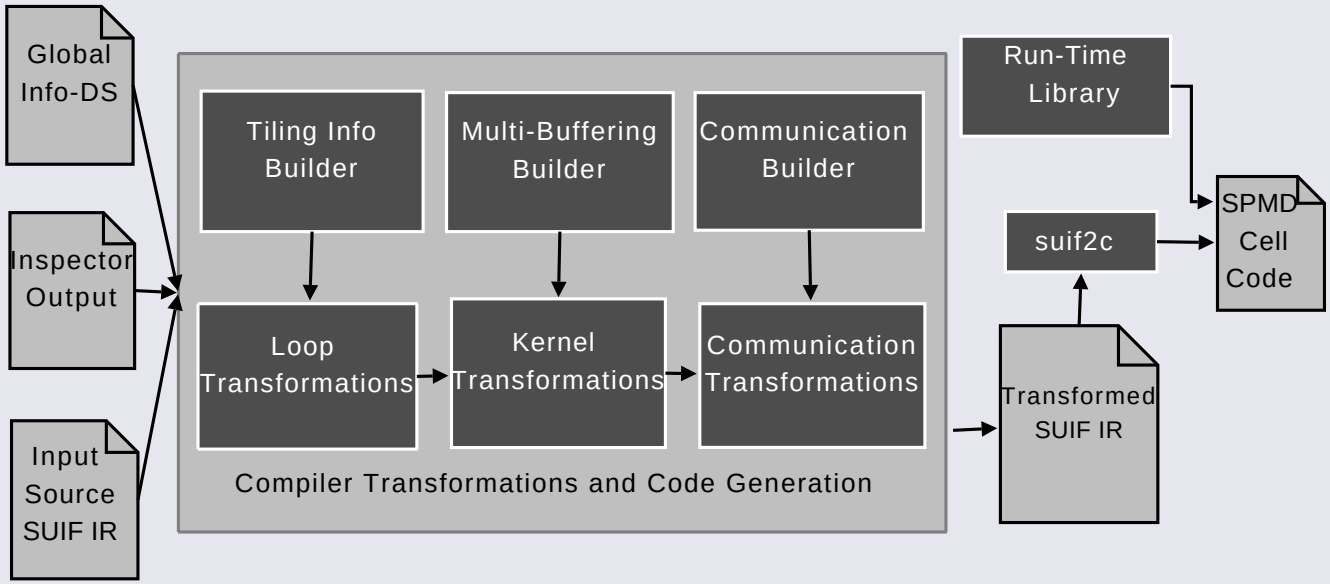
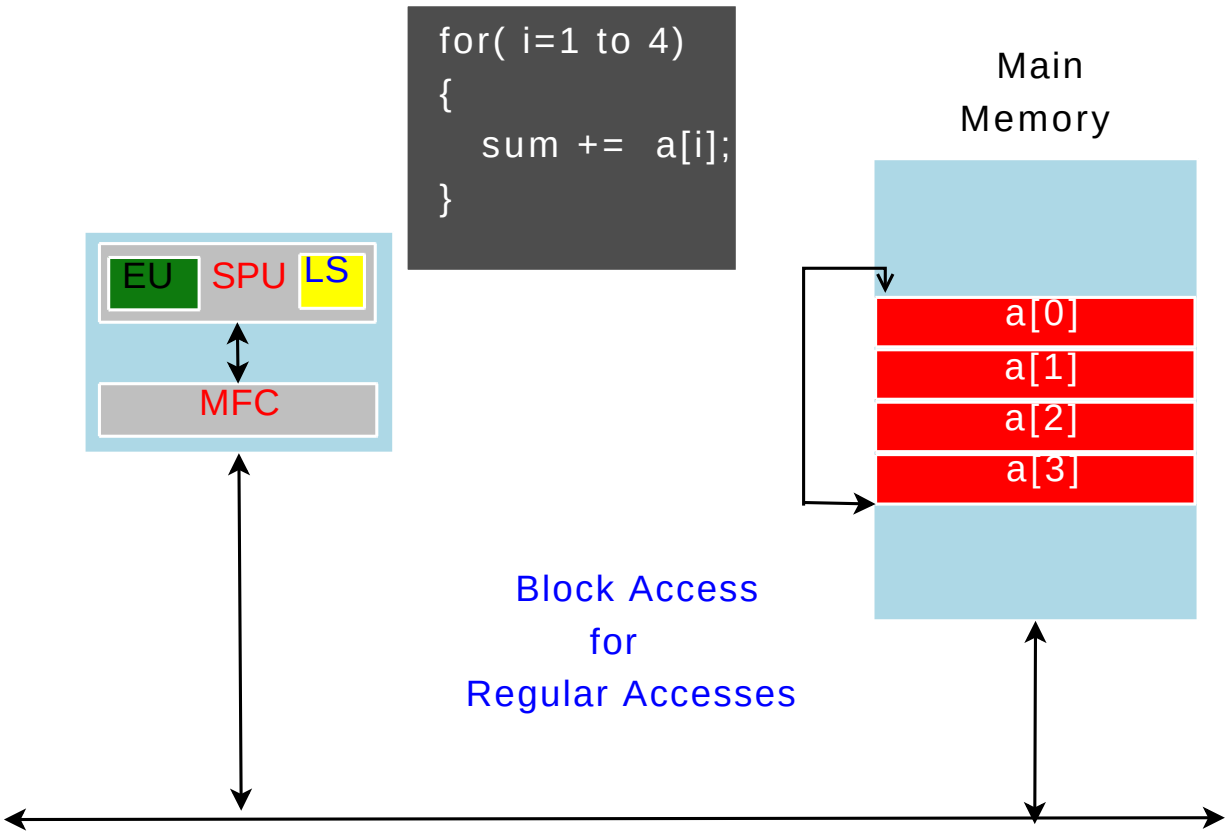


Figure: Compiler transformations and code generation

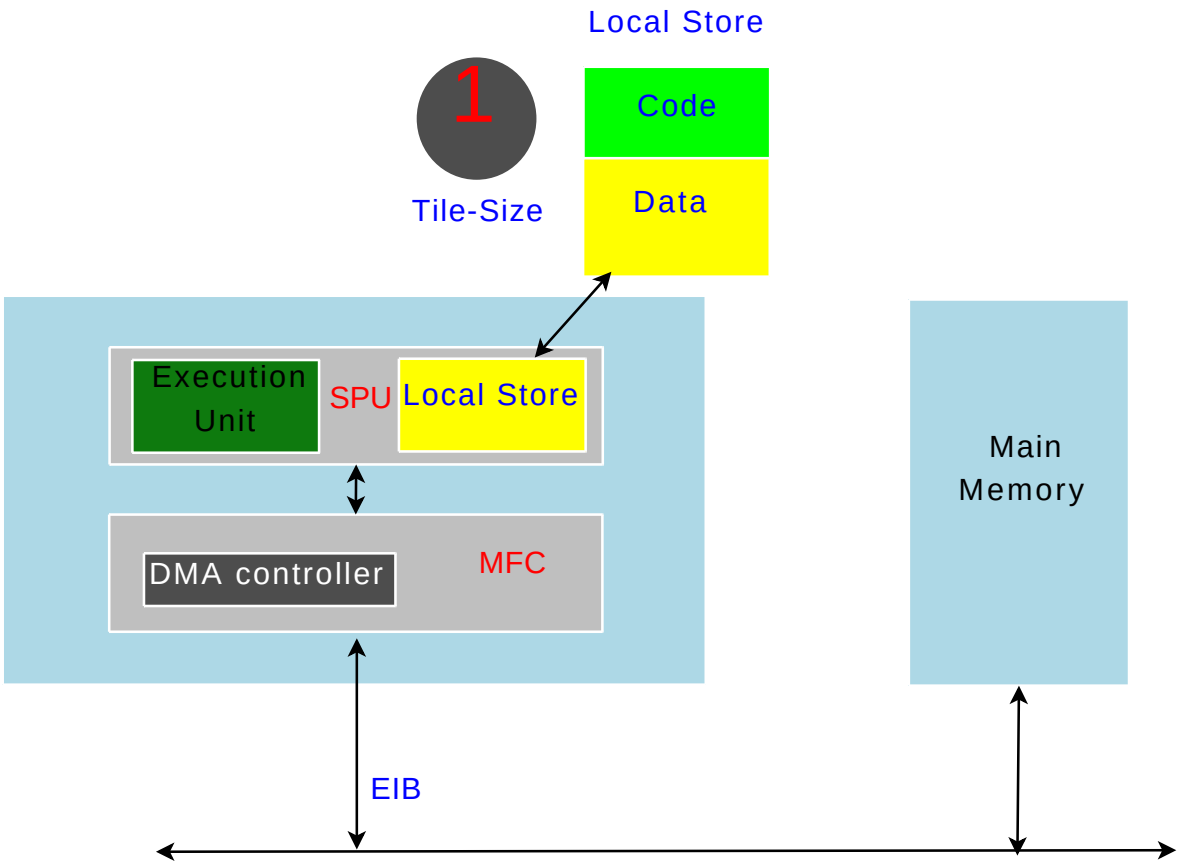
Compiler-Directed Communication Mechanism

Compiler-Directed Communication Mechanism

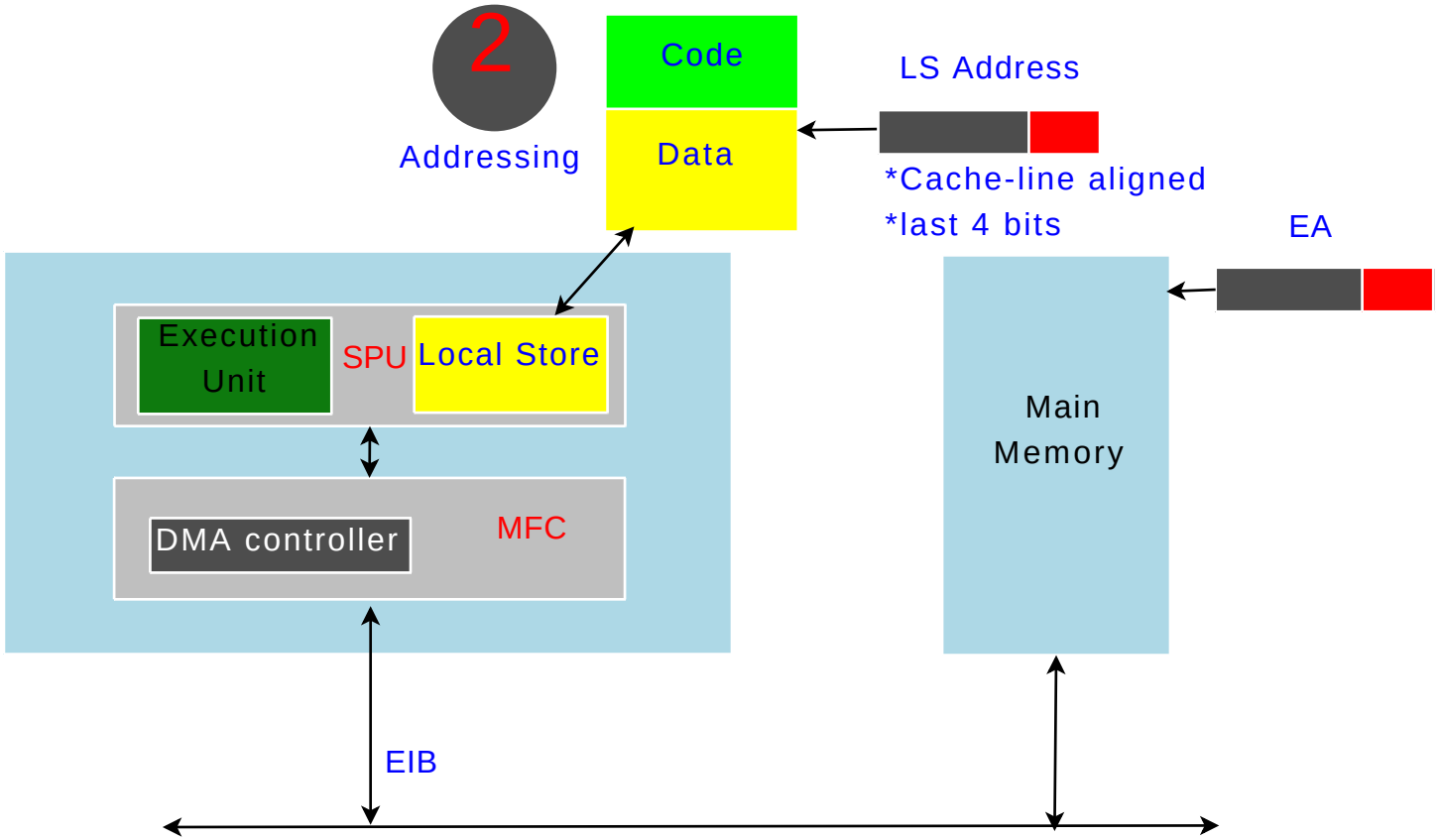
Compiler-directed Communication Mechanism
Block Access Method for Regular Accesses



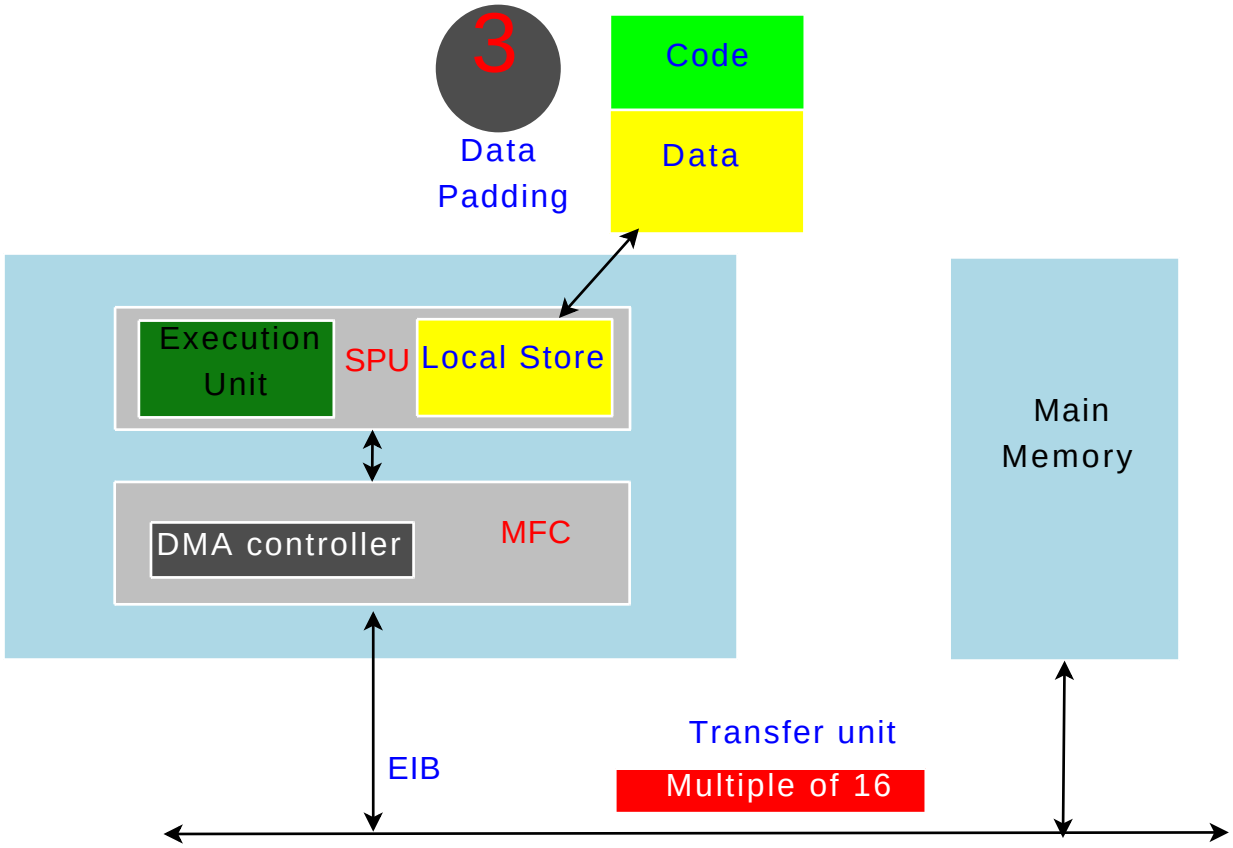
Compiler-directed Communication Mechanism
Block Access Method for Regular Accesses



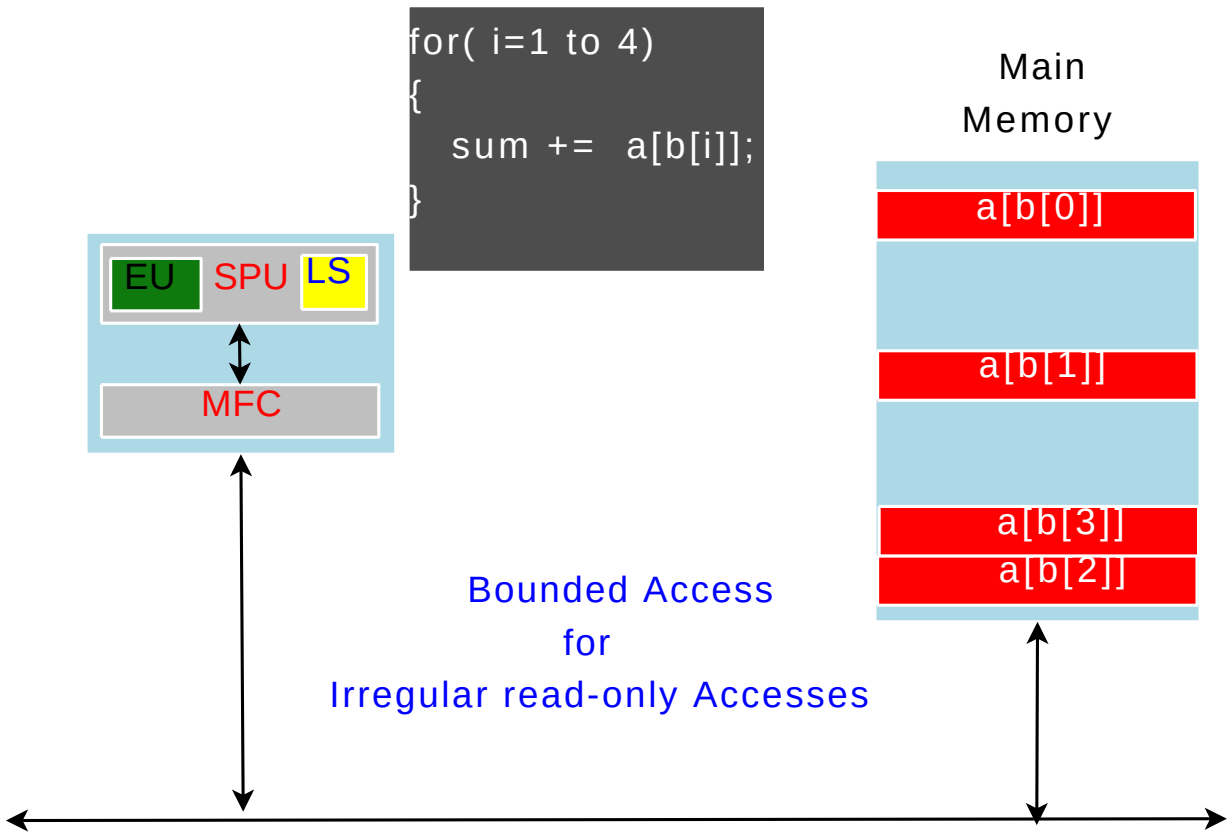
Compiler-directed Communication Mechanism
Block Access Method for Regular Accesses



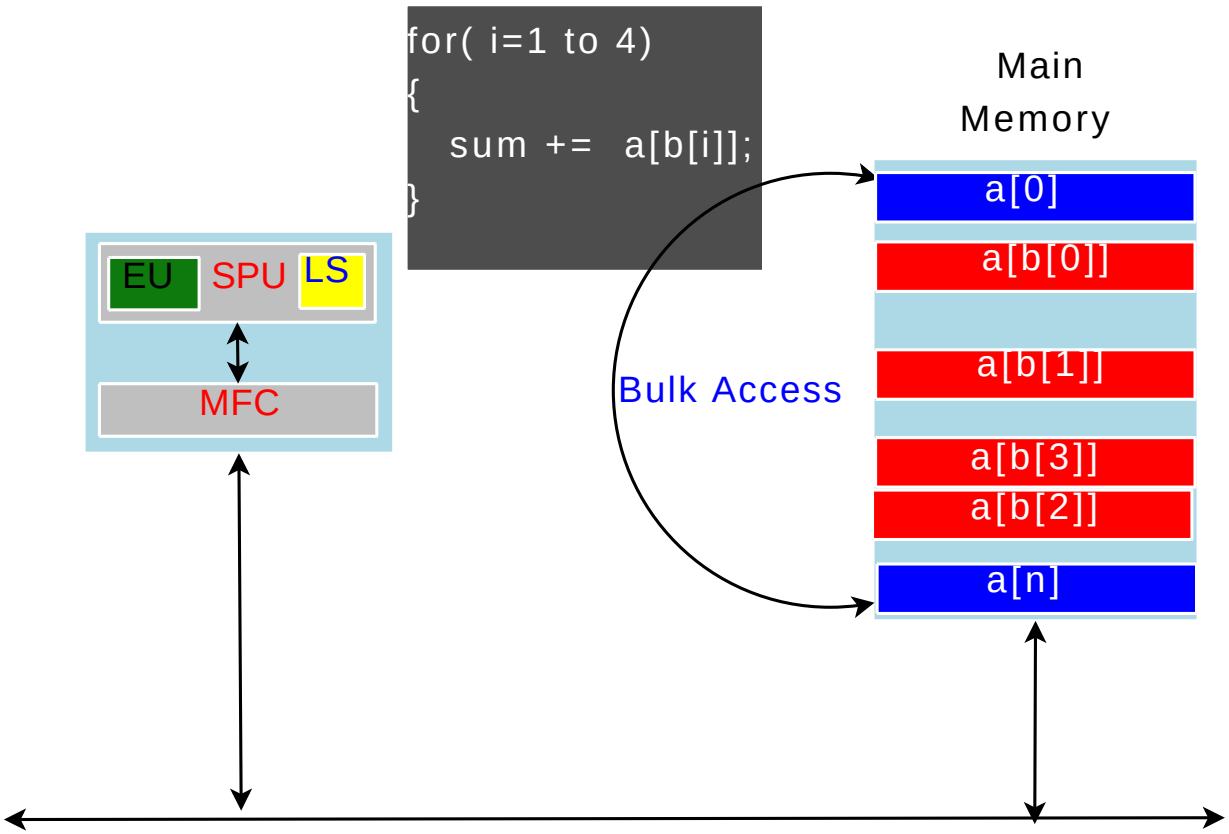
Compiler-directed Communication Mechanism
Block Access Method for Regular Accesses



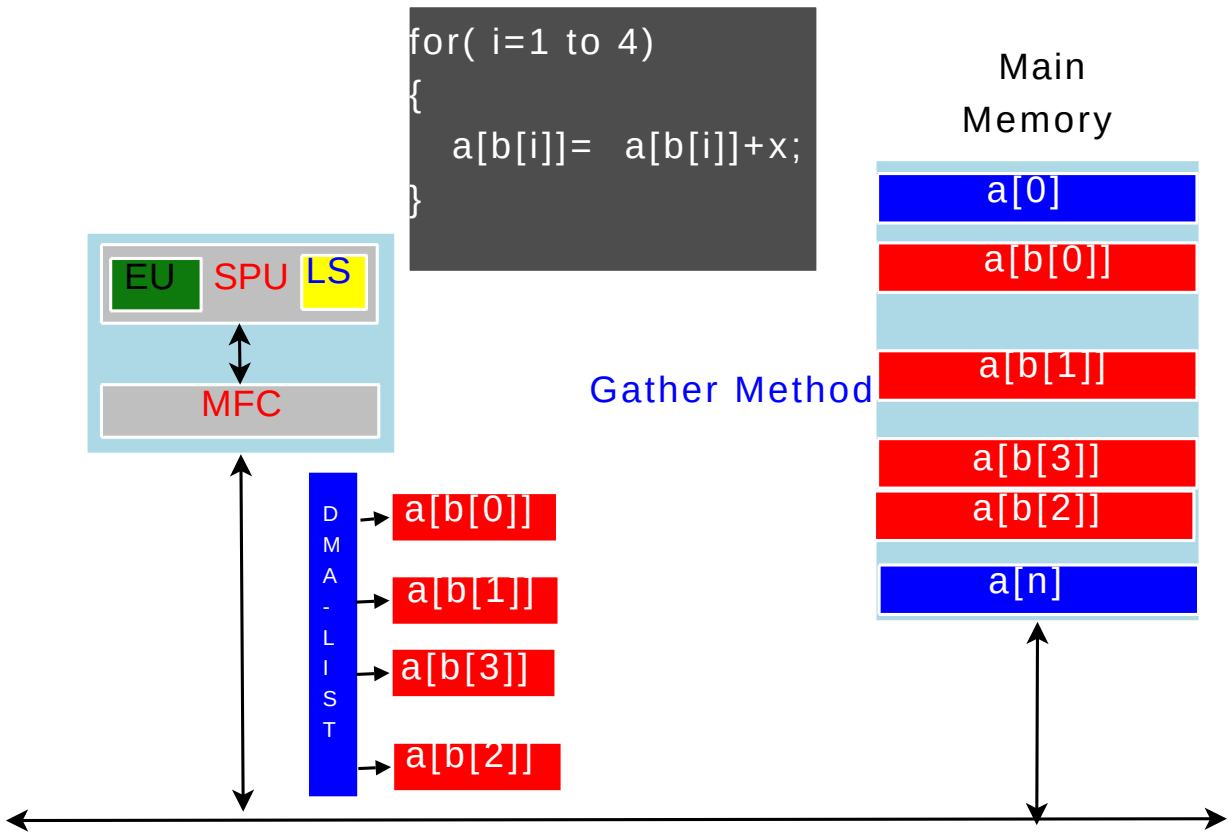
Compiler-directed Communication Mechanism
Bounded Method for Irregular Read Accesses



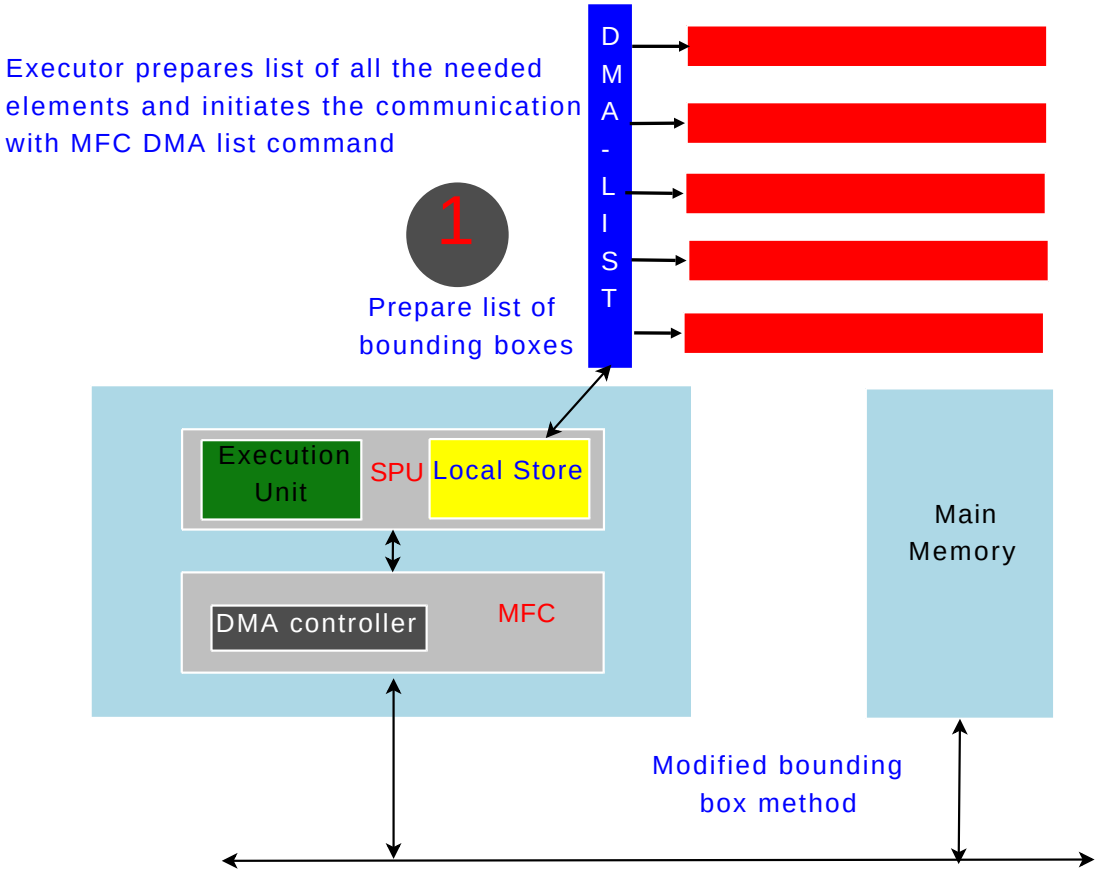
Compiler-directed Communication Mechanism
Bounded Method for Irregular Read Accesses



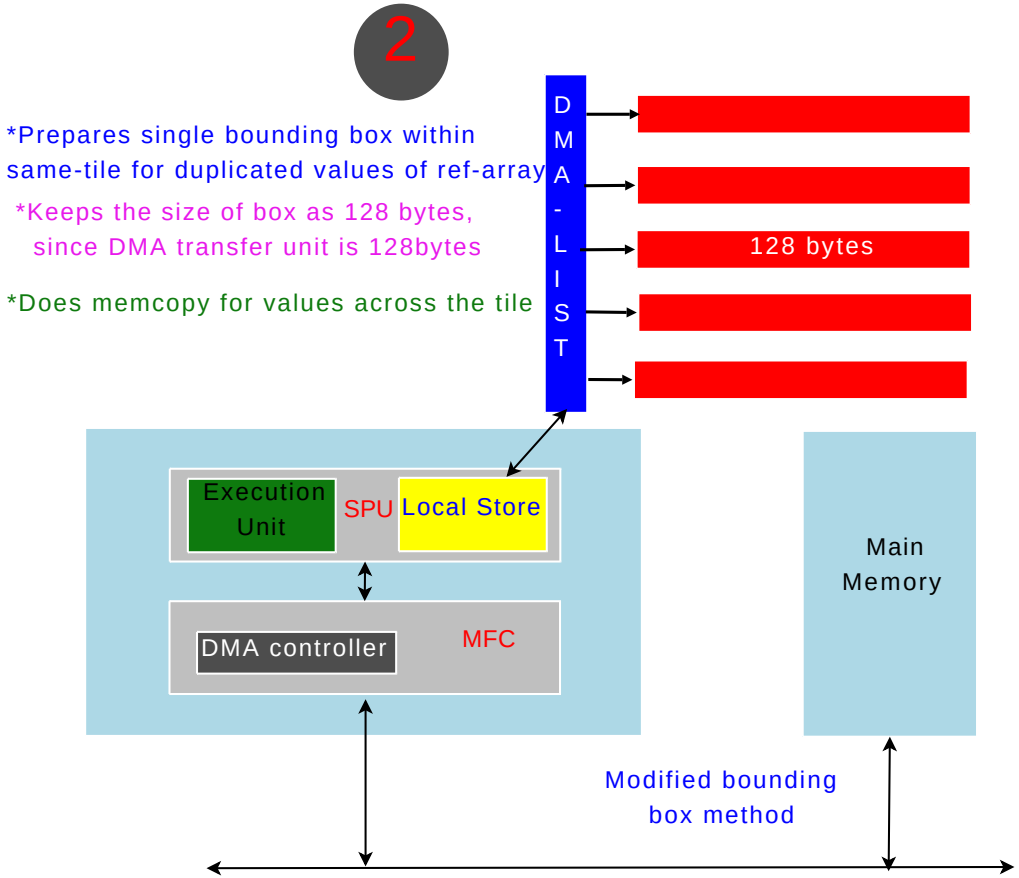
Compiler-directed Communication Mechanism
Bounded Method for Irregular Read Accesses



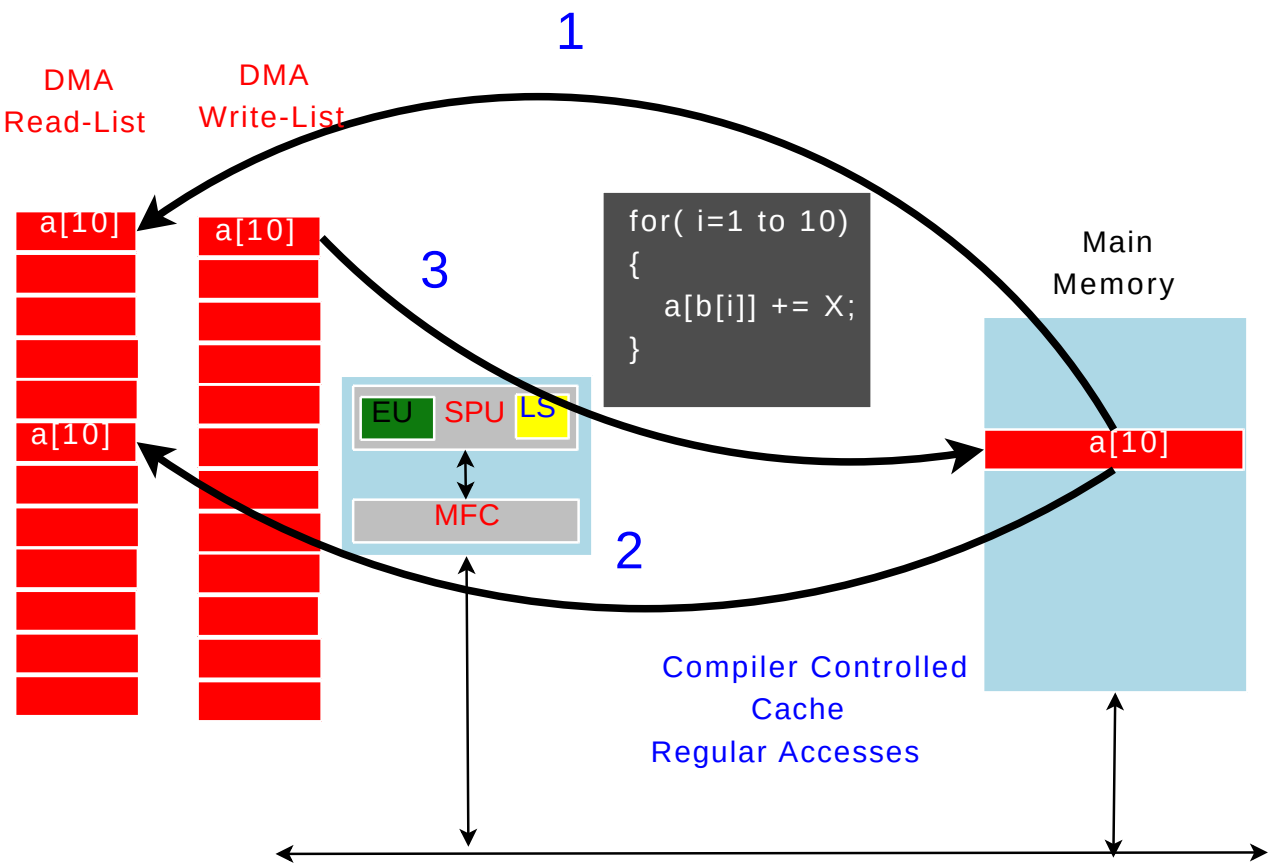
Compiler-directed Communication Mechanism
Bounded Method for Irregular Read Accesses



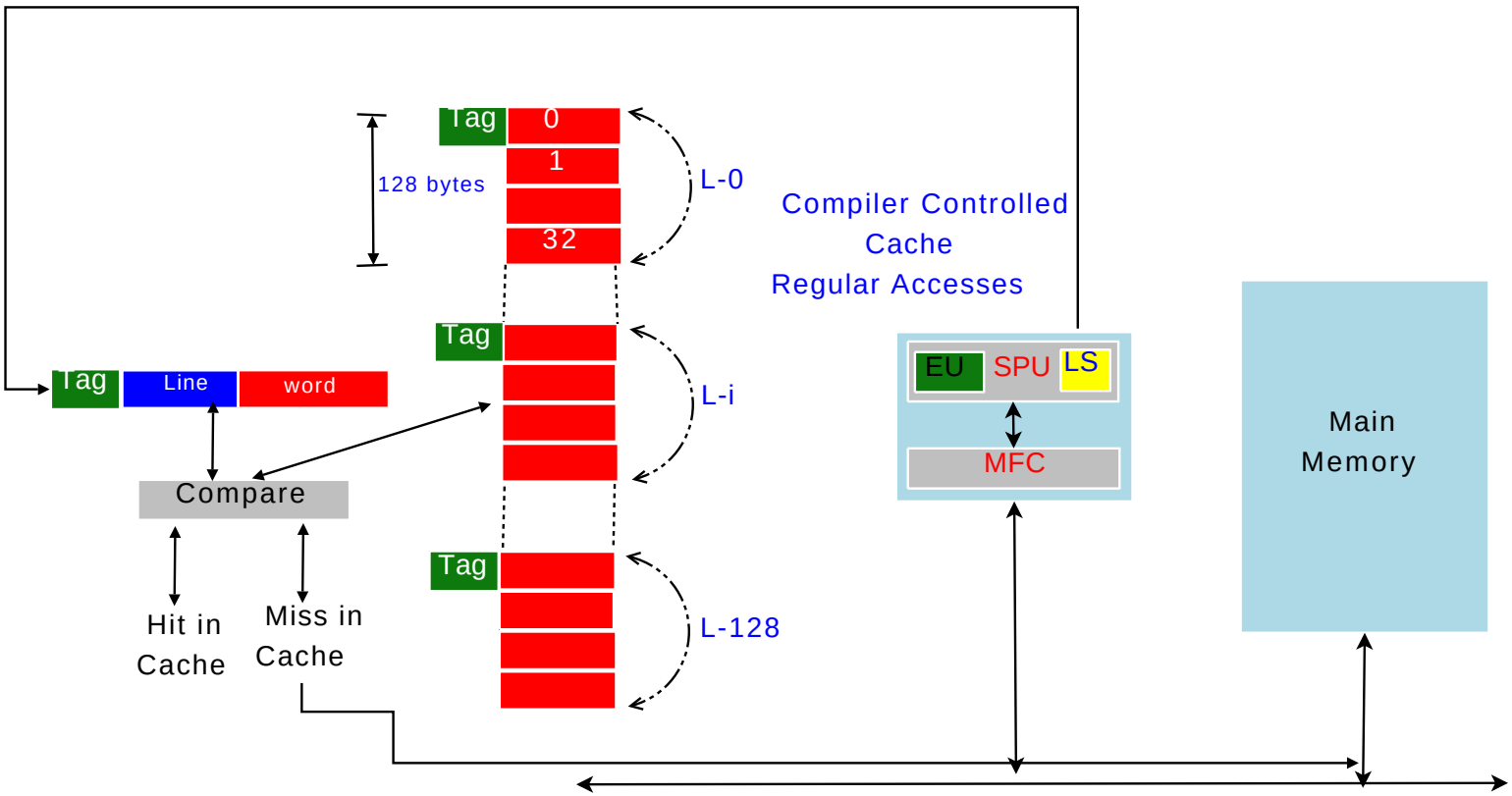
Compiler-directed Communication Mechanism
Bounded Method for Irregular Read Accesses



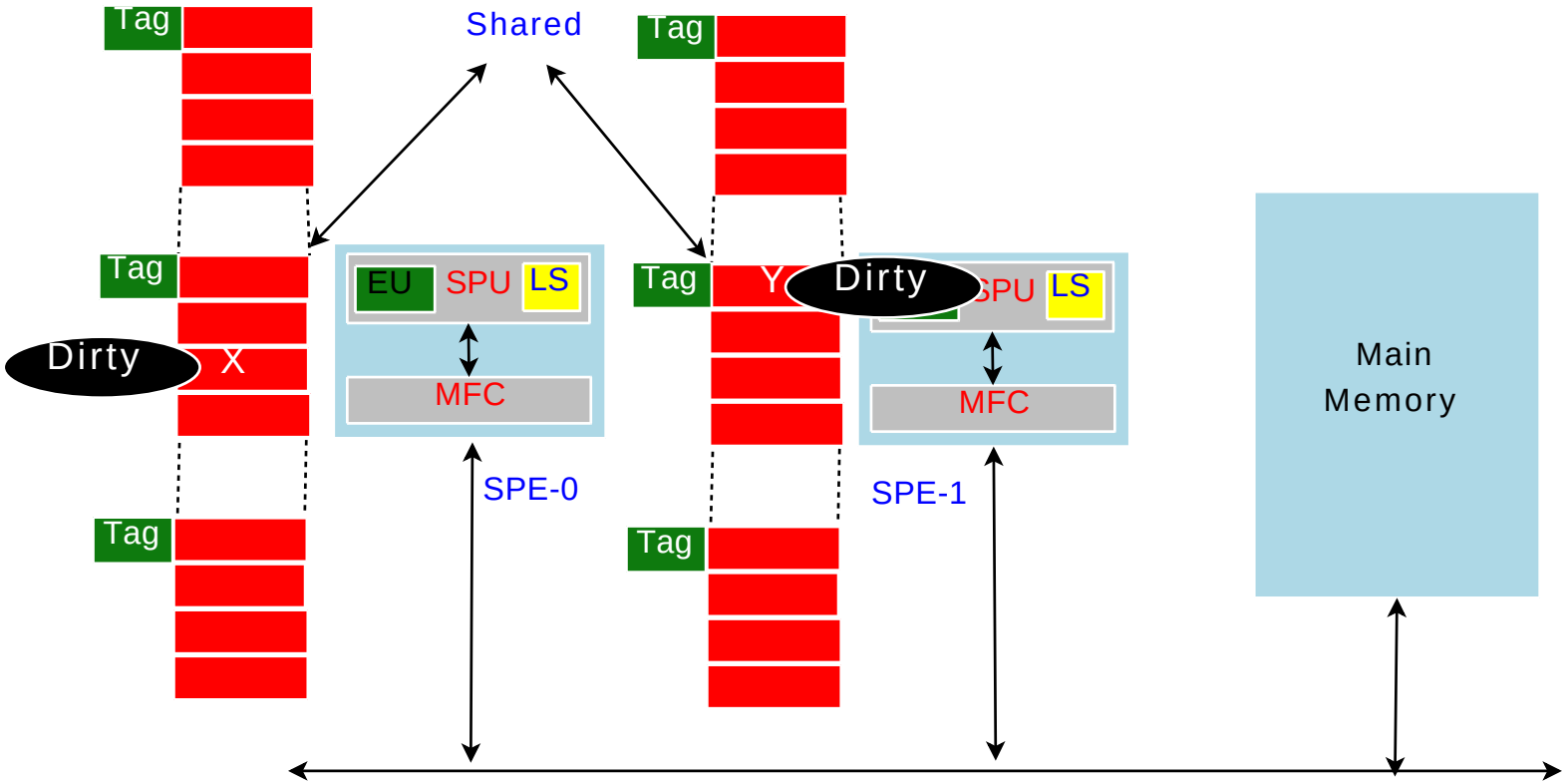
Compiler-directed Communication Mechanism
Compiler Controlled Cache for Sparse Updates



Compiler-directed Communication Mechanism
Compiler Controlled Cache for Sparse Updates



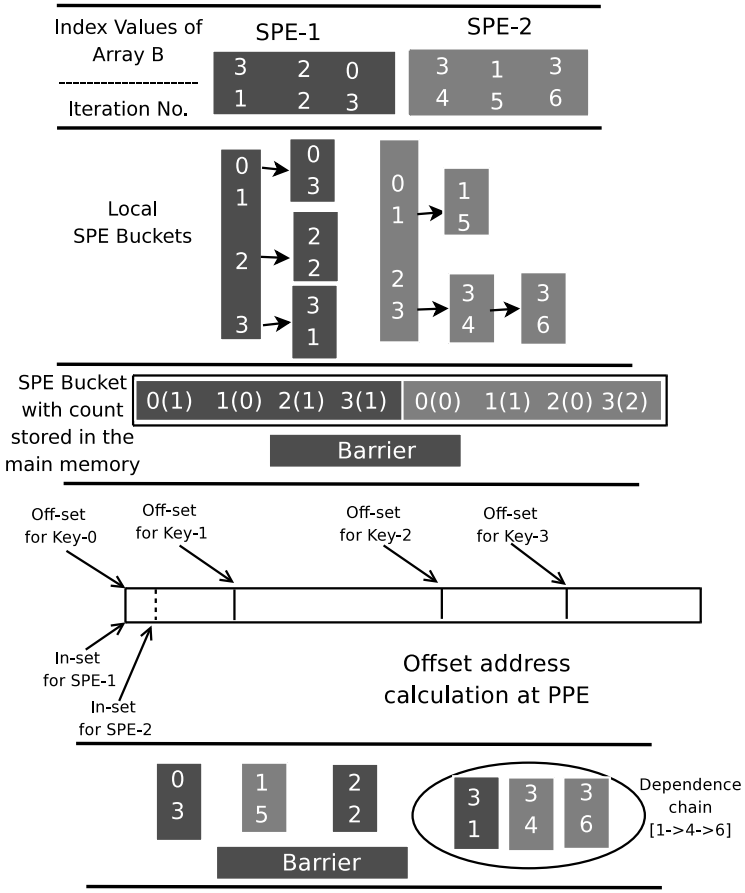
Compiler-directed Communication Mechanism
Compiler Controlled Cache for Sparse Updates



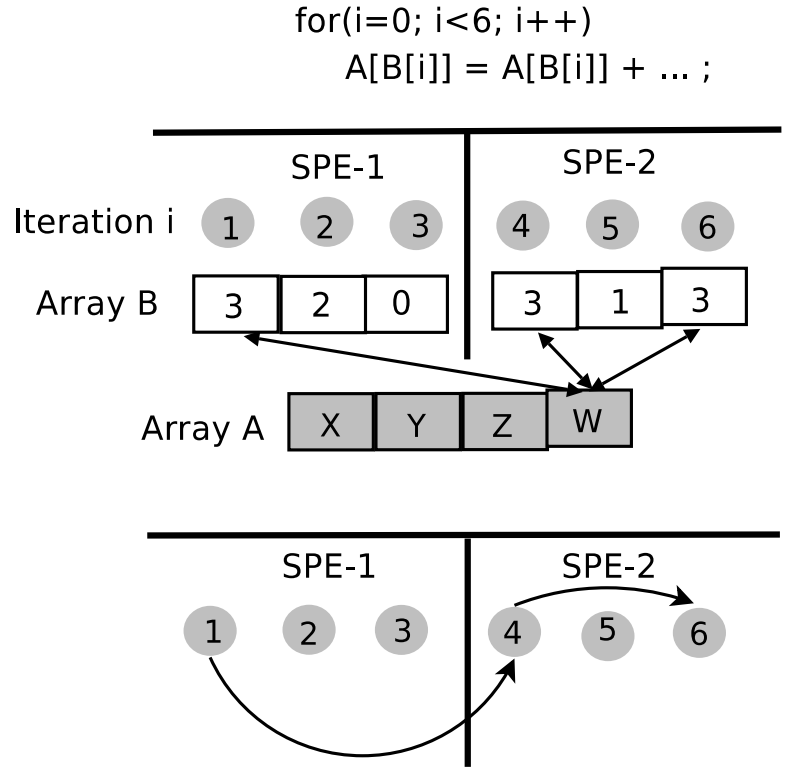
Run-time Parallelization

Run-time Parallelization

Parallelization of Irregular Reduction Loops



Parallel Construction of the Dependence Chain



Serial execution dependence chain [1->4-> 6]

Experiment and Results

Experiments and Results

Iterative Partial Differential Equation (PDE) Solver

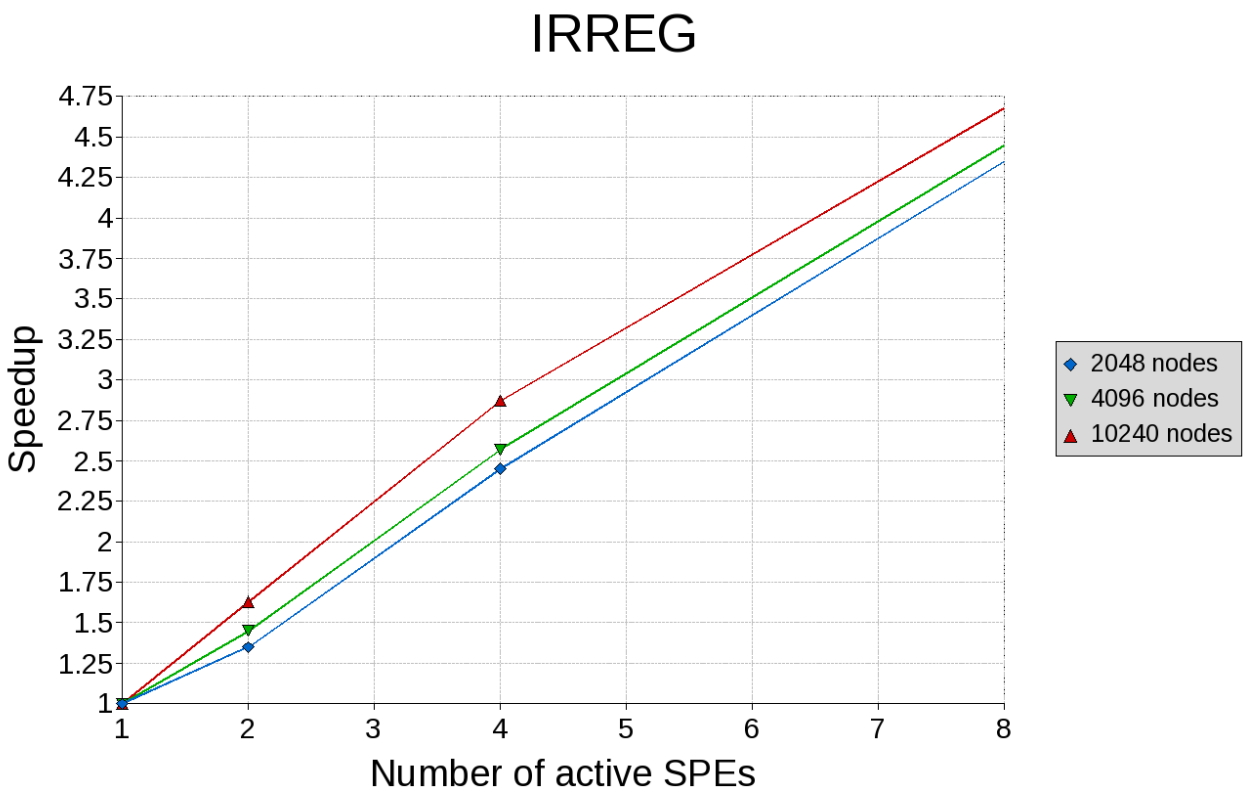


Figure: Speedup for IRREG

Molecular Dynamics Code

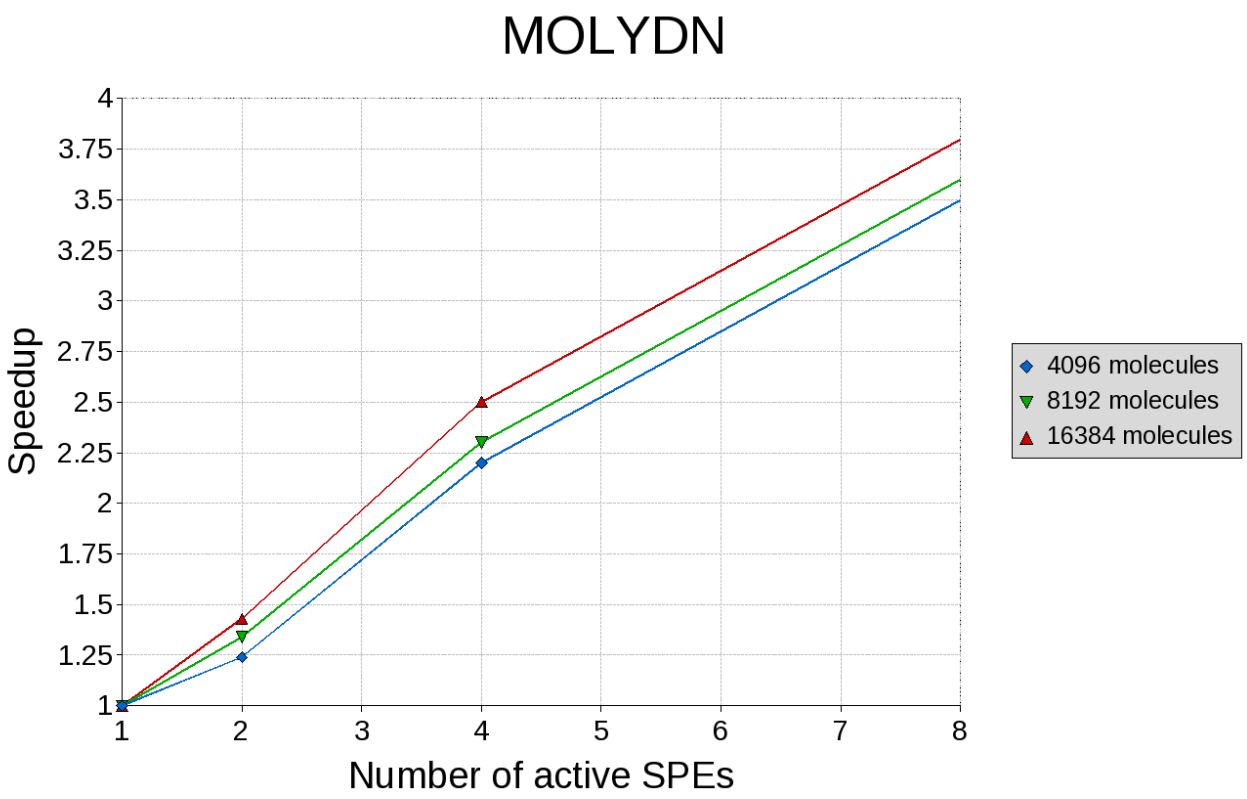


Figure: Speedup for MOLDYN

Nonbonded Force Calculations

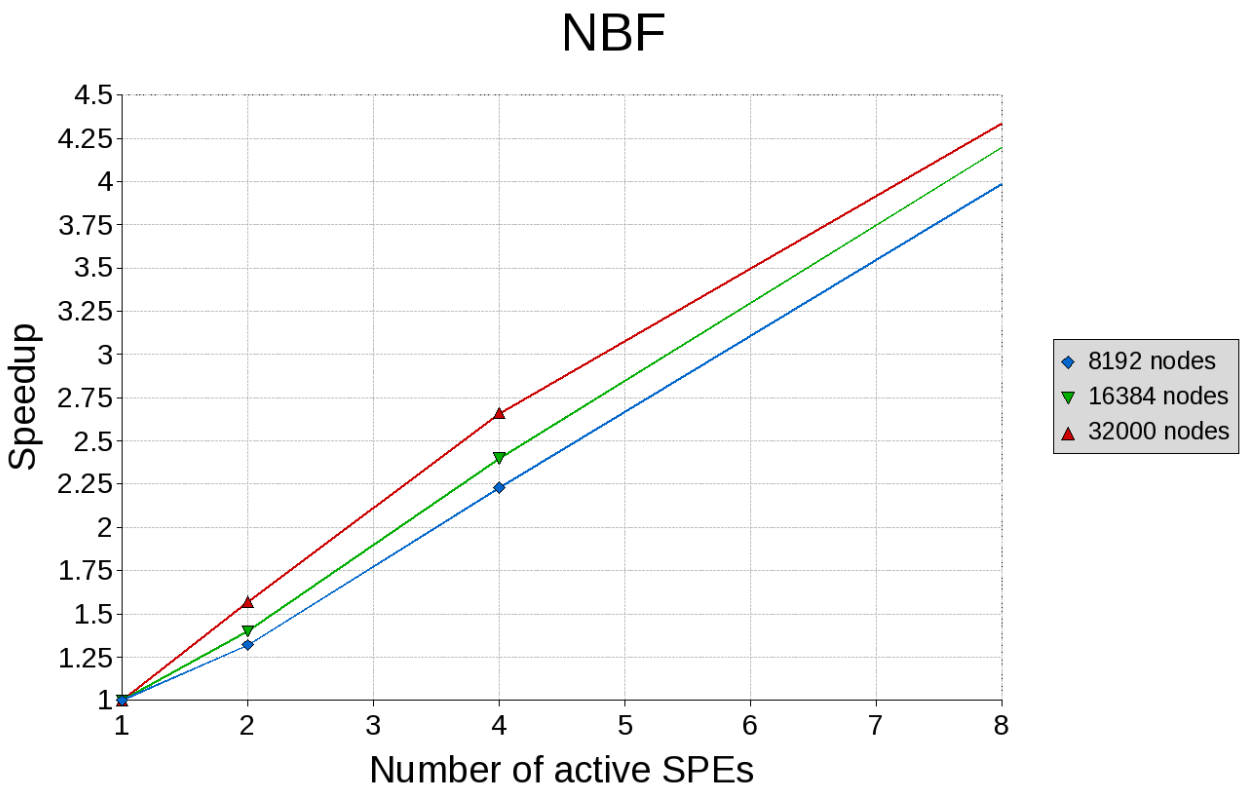


Figure: Speedup for NBF

Summary

Summary

Summary

- Dataflow analysis framework for determining the program points for memory communication.
- Compiler transformation for run-time data processing and actual computation.
- Builds the memory communication schedules.
- Run-time parallelization of loops judiciously partitions the data and computational work.

Thank You

Thank You