A Parallelizing Compiler for Multicore Systems

José M. Andión, Manuel Arenaz, Gabriel Rodríguez and Juan Touriño

17th International Workshop on Software and Compilers for Embedded Systems (SCOPES 2014)
June 10-11, 2014 — Schloss Rheinfels, Sankt Goar, Germany

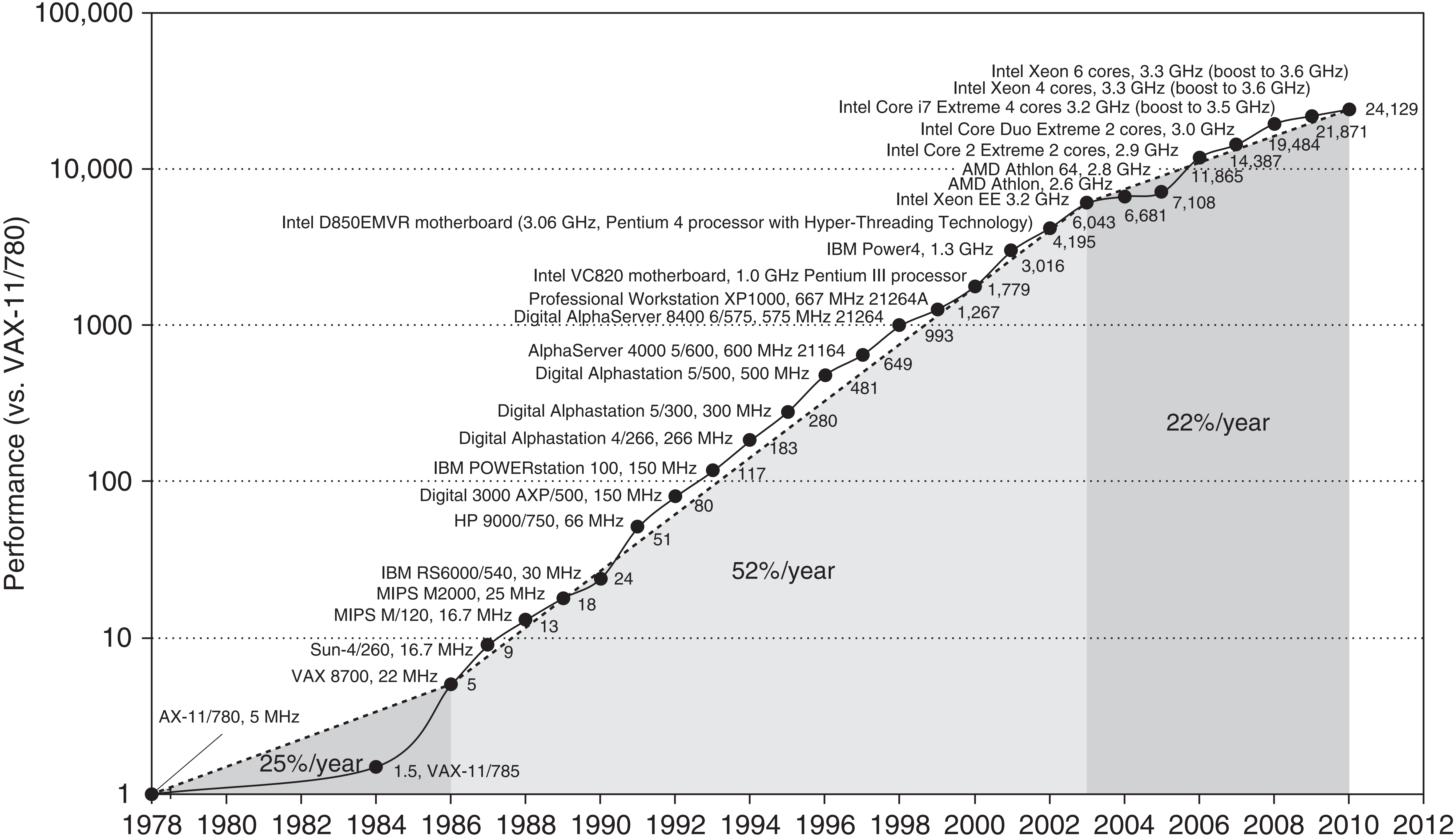
COMPUTER ARCHITECTURE GROUP
UNIVERSITY OF A CORUÑA
GAC.UDC.ES
UNIVERSIDADE DA CORUÑA

Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions



The Parallel Challenge

David A. Patterson and John L. Hennessy. Computer Organization and Design: The Hardware/Software Interface. Elsevier, 2014.

The Parallel Challenge

- libraries
- compiler directives
- programming languages
- parallelizing compilers

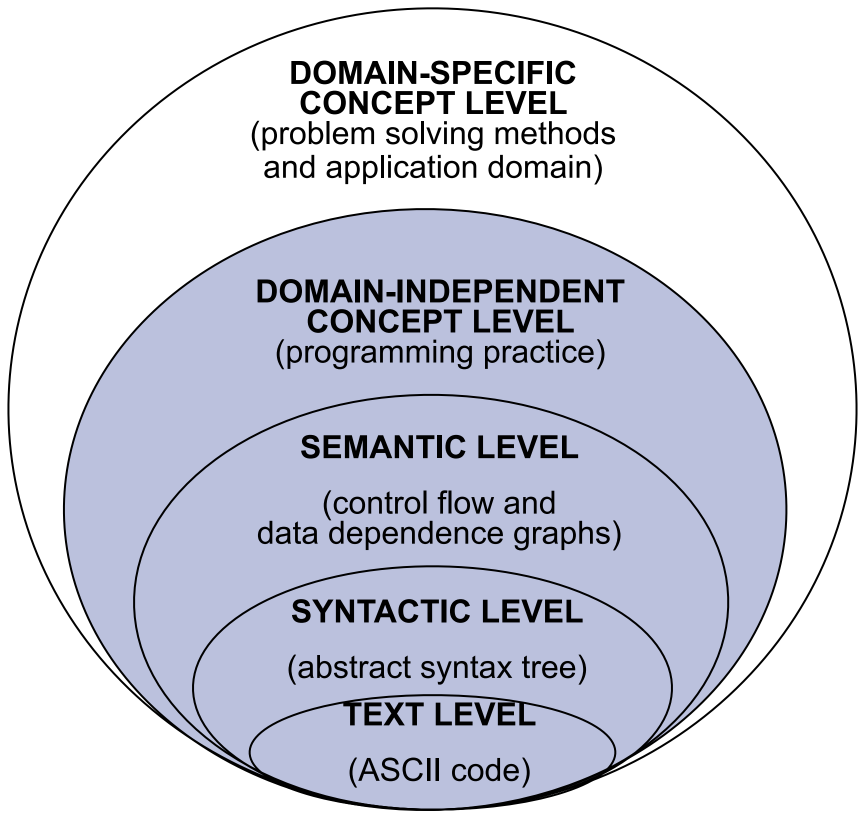
Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

diKernel: Domain-Independent Computational Kernel



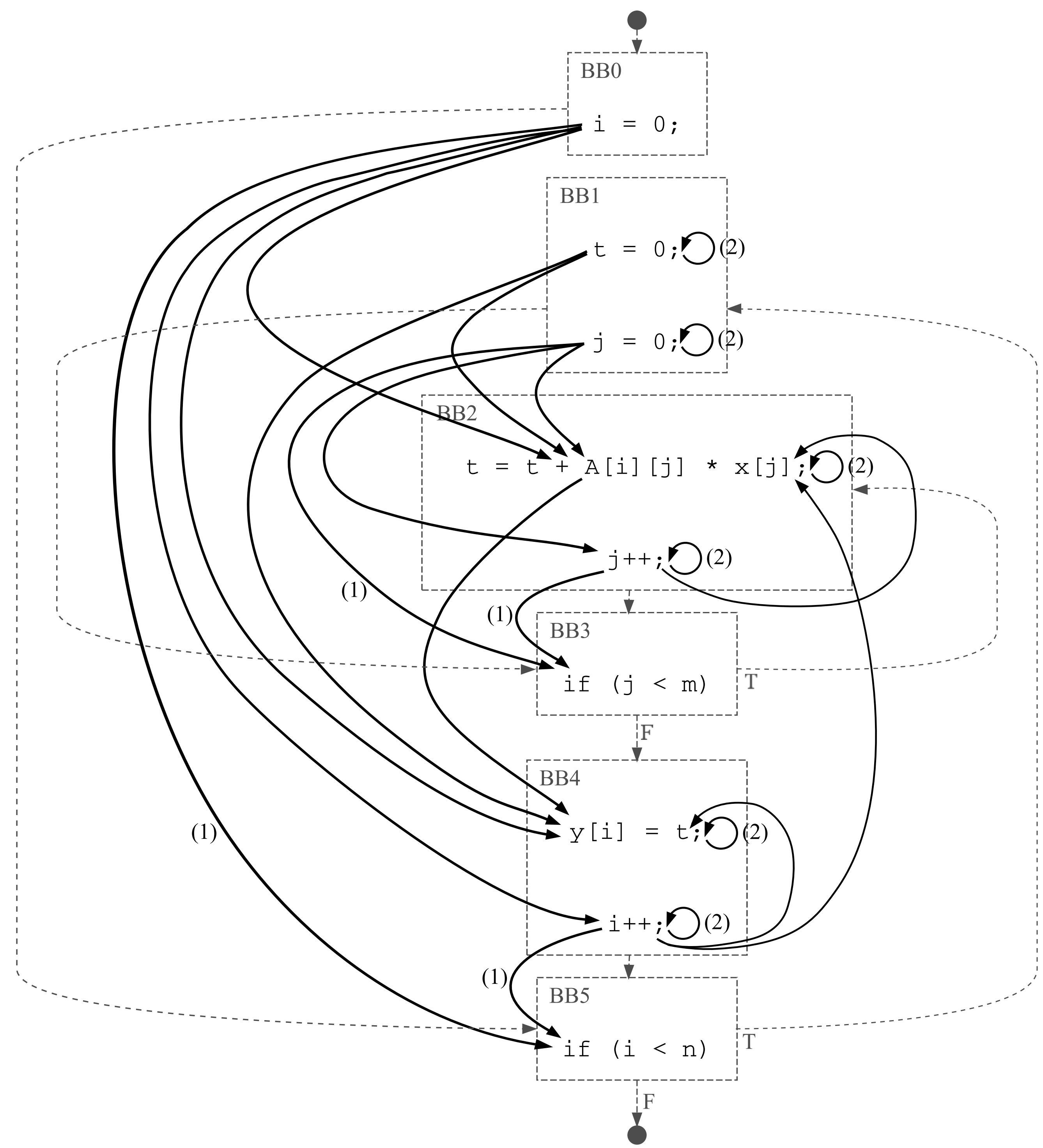
- Characterizes the computations carried out in a program without being affected by how they are coded

- Exposes multiple levels of parallelism

M. Arenaz et al. XARK: An Extensible Framework for Automatic Recognition of Computational Kernels. ACM Transactions on Programming Languages and Systems, 30(6), 2008.

Standard statement-based IR

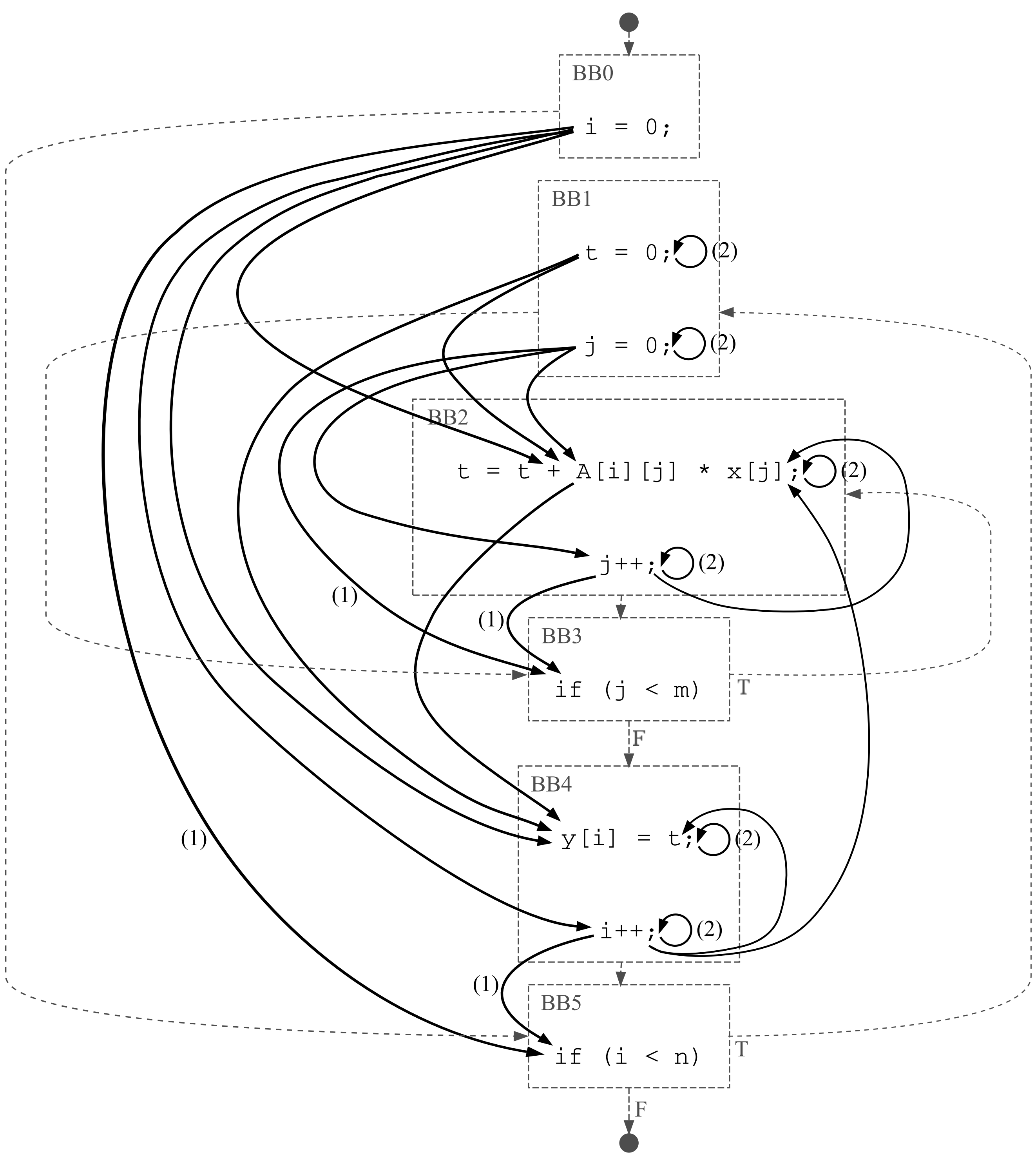
```
1.for (i = 0; i < n; i++) {
2.    t = 0;
3.    for (j = 0; j < m; j++) {
4.        t = t + A[i][j] * x[j];
5.    }
6.    y[i] = t;
7.}
```

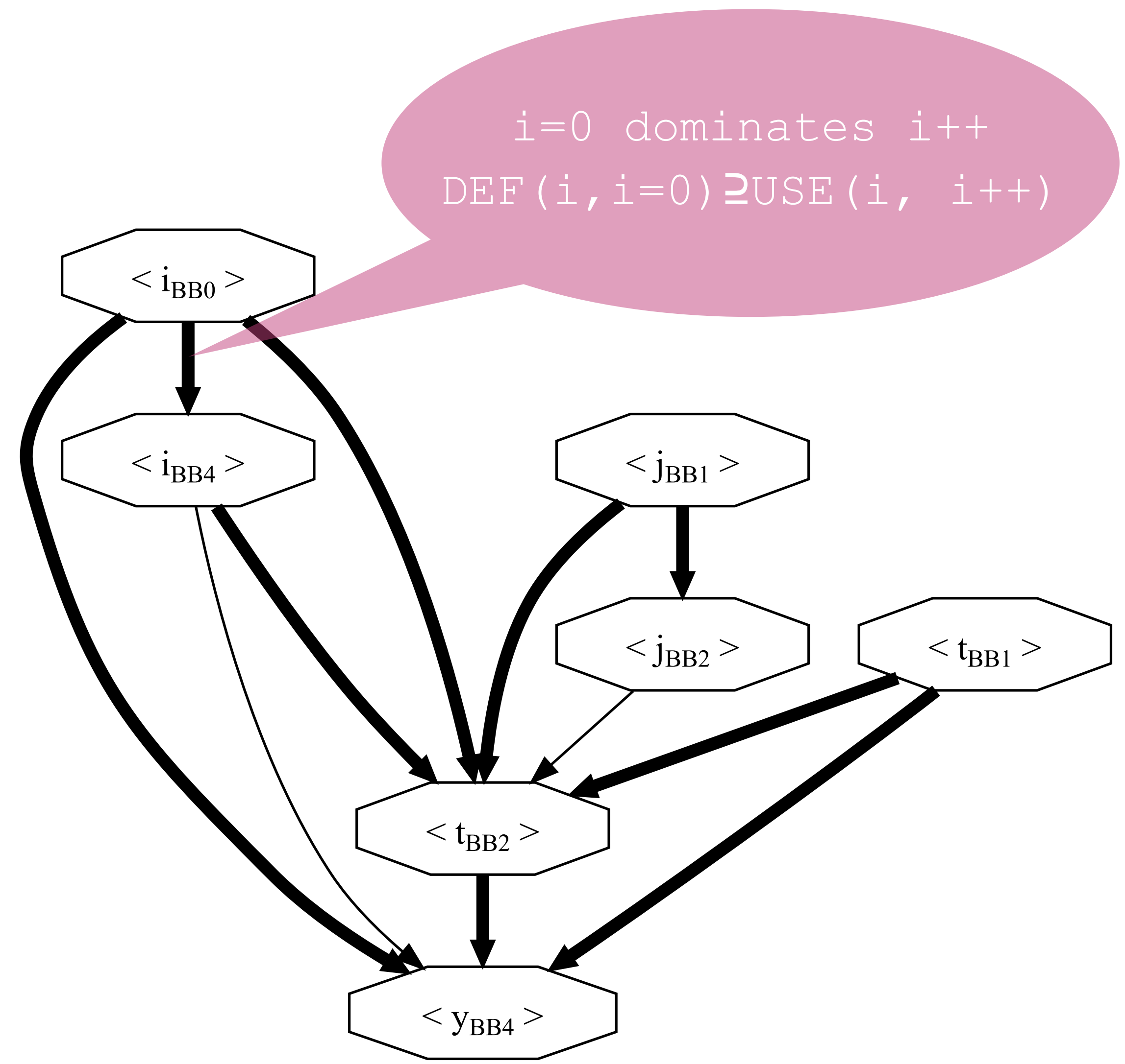


J.M. Andión et al. A Parallelizing Compiler for Multicore Systems. SCOPES 2014.

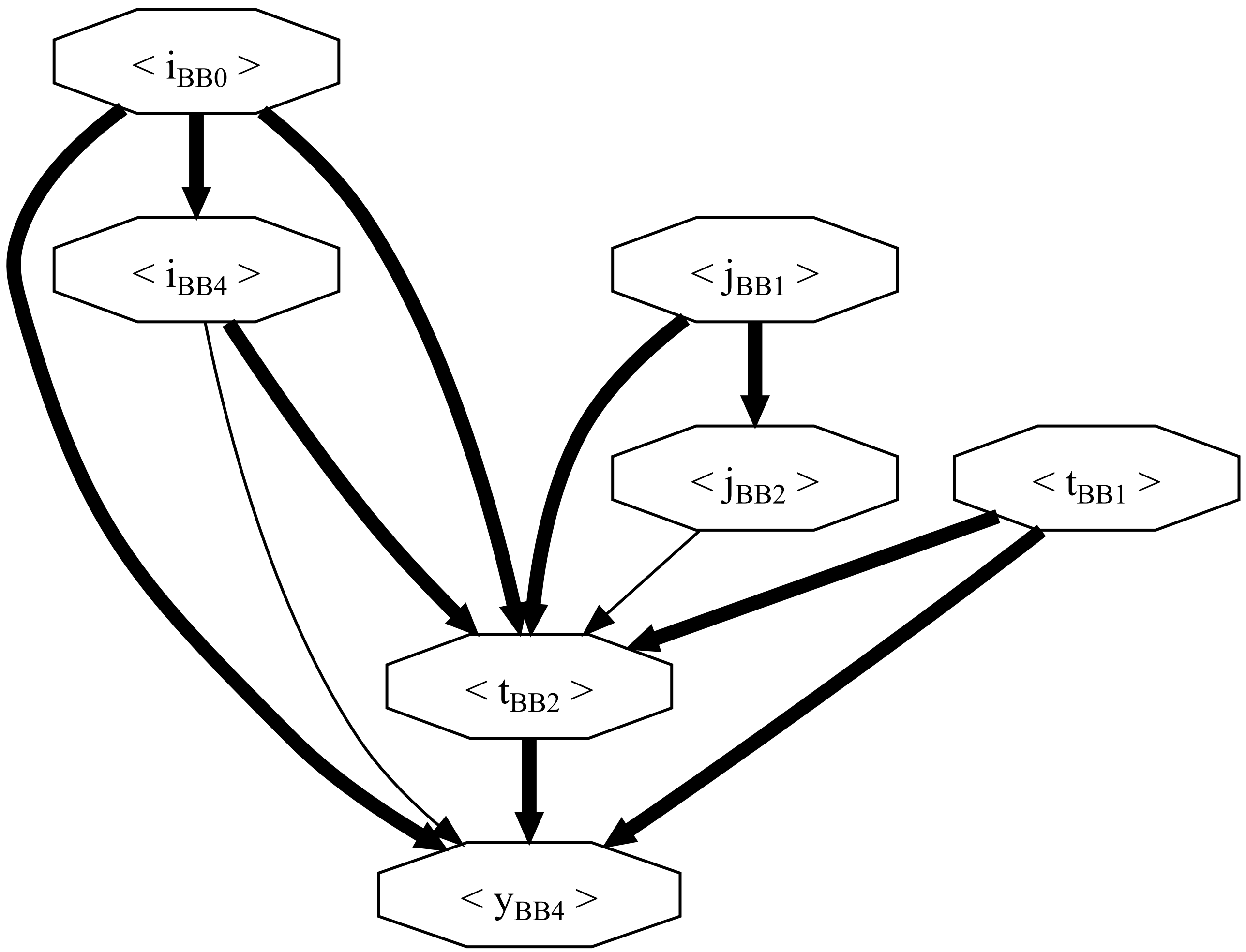
COMPUTER ARCHITECTURE GROUP UNIVERSITY OF A CORUÑA GAC.UDC.ES UNIVERSIDADE DA CORUÑA

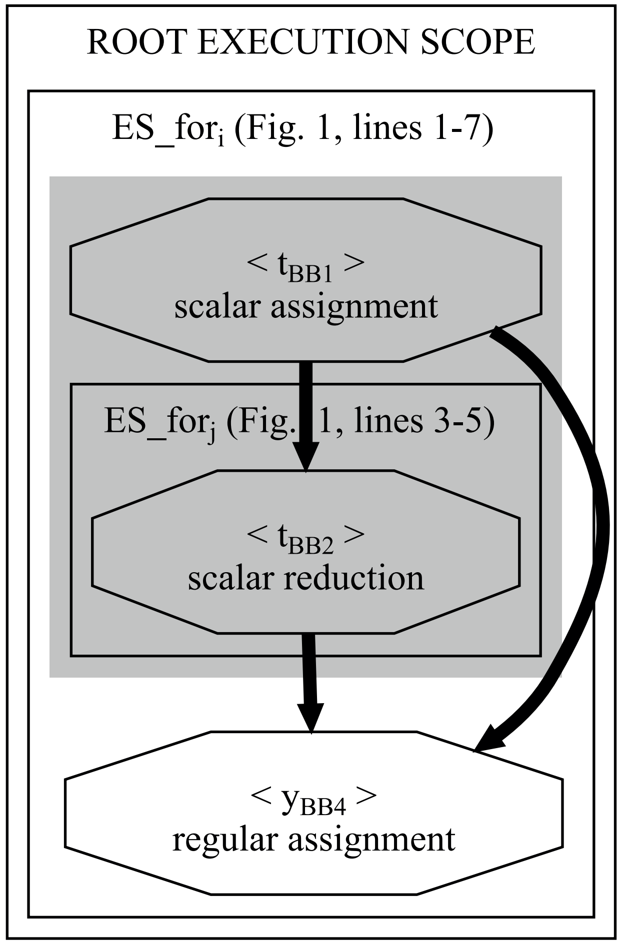
Building the KIR (I)





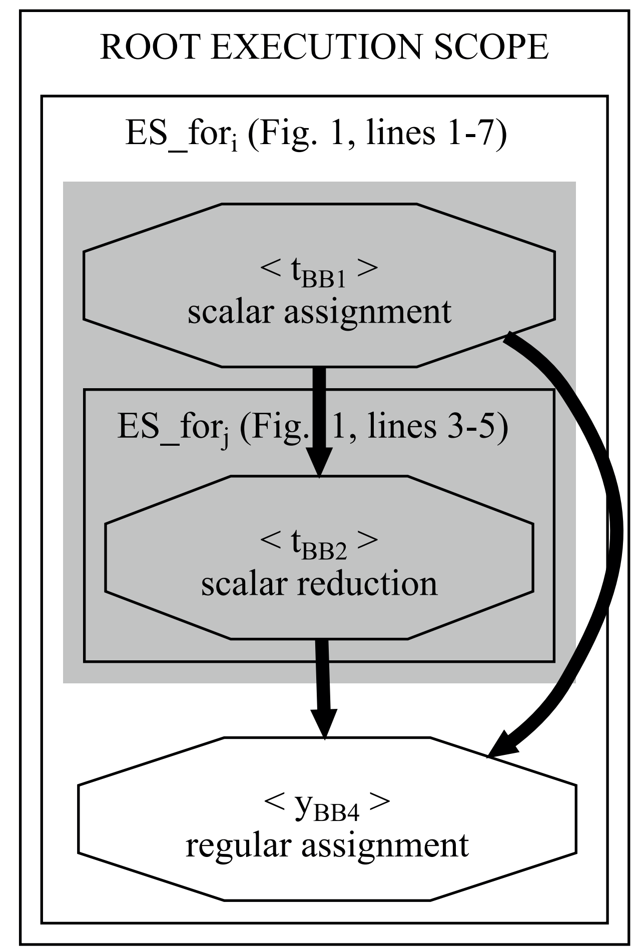
Building the KIR (II)





Building the KIR (and III)

```
1.for (i = 0; i < n; i++) {
2.    t = 0;
3.    for (j = 0; j < m; j++) {
4.        t = t + A[i][j] * x[j];
5.    }
6.    y[i] = t;
7.}
```



Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

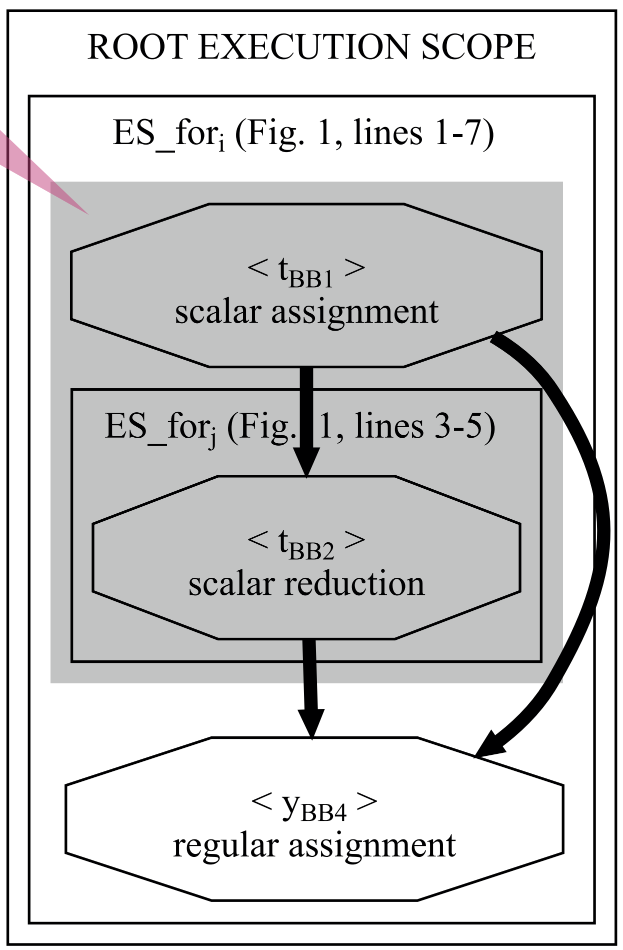
Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

Automatic Partitioning driven by the KIR (I)

t is a privatizable scalar variable

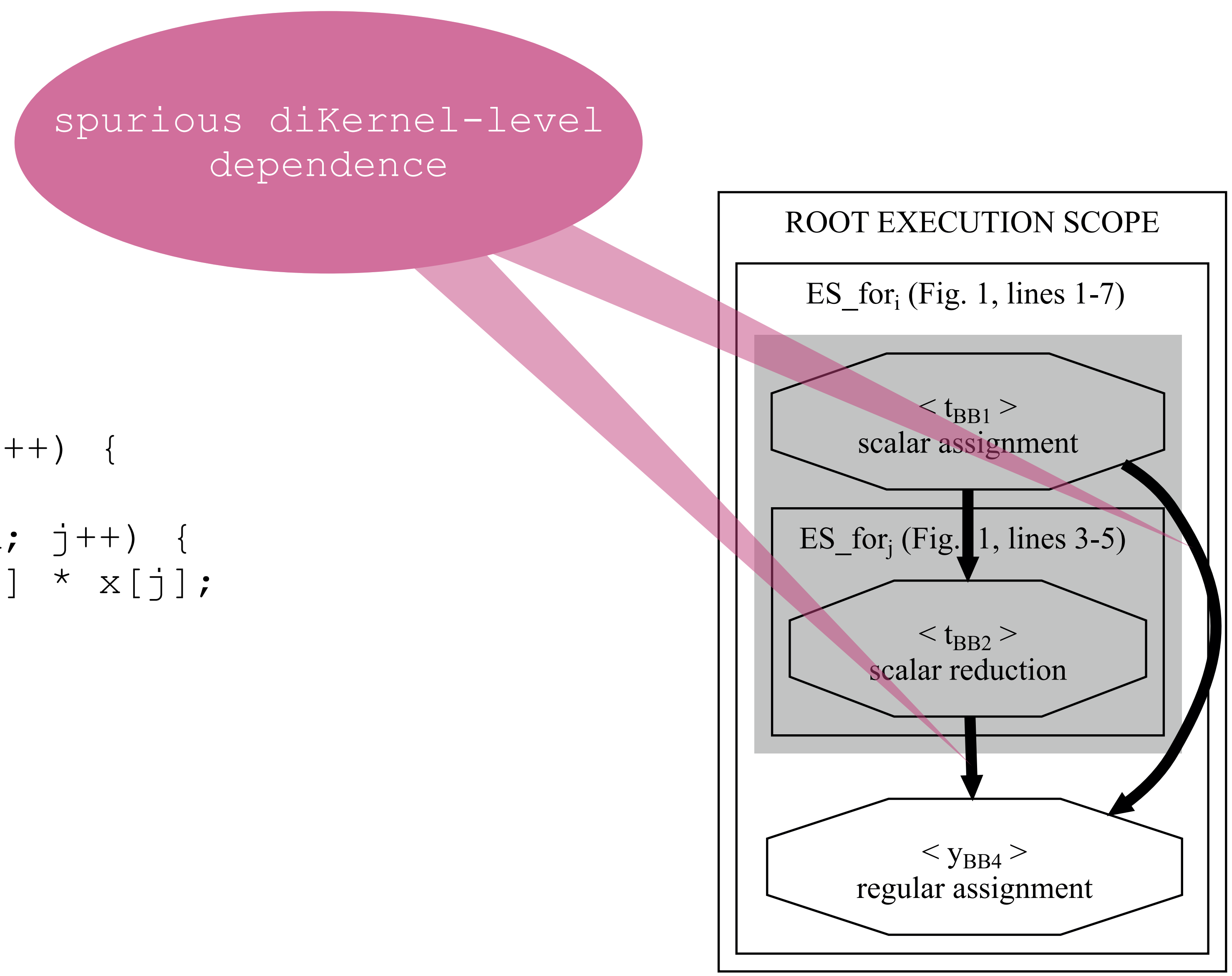
```
1.for (i = 0; i < n; i++) {
2.   t = 0;
3.   for (j = 0; j < m; j++) {
4.      t = t + A[i][j] * x[j];
5.   }
6.   y[i] = t;
7.}
```



UNIVERSIDADE DA CORUÑA

Automatic Partitioning driven by the KIR (II)

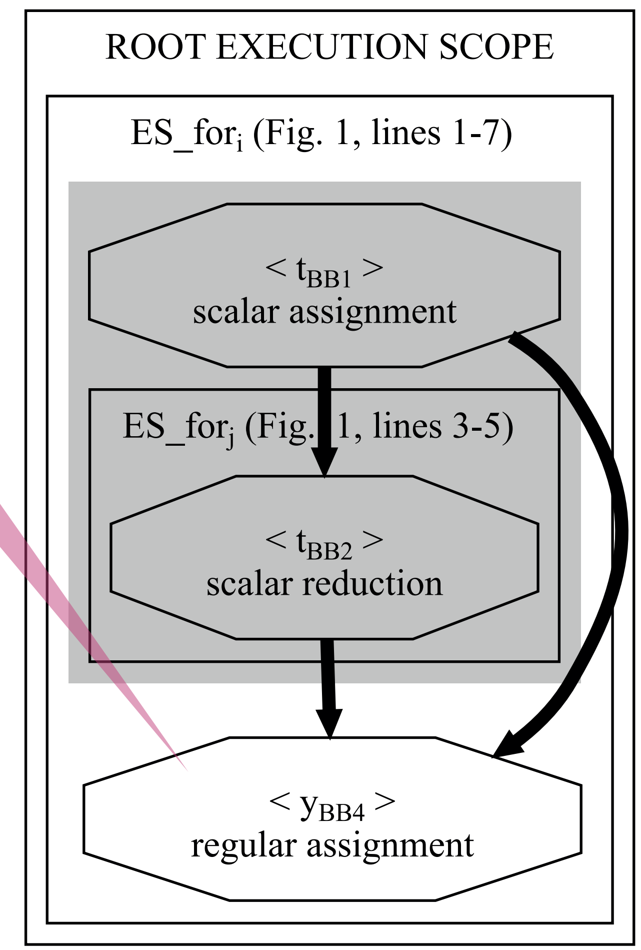
```
1.for (i = 0; i < n; i++) {
2.    t = 0;
3.    for (j = 0; j < m; j++) {
4.        t = t + A[i][j] * x[j];
5.    }
6.    y[i] = t;
7.}
```



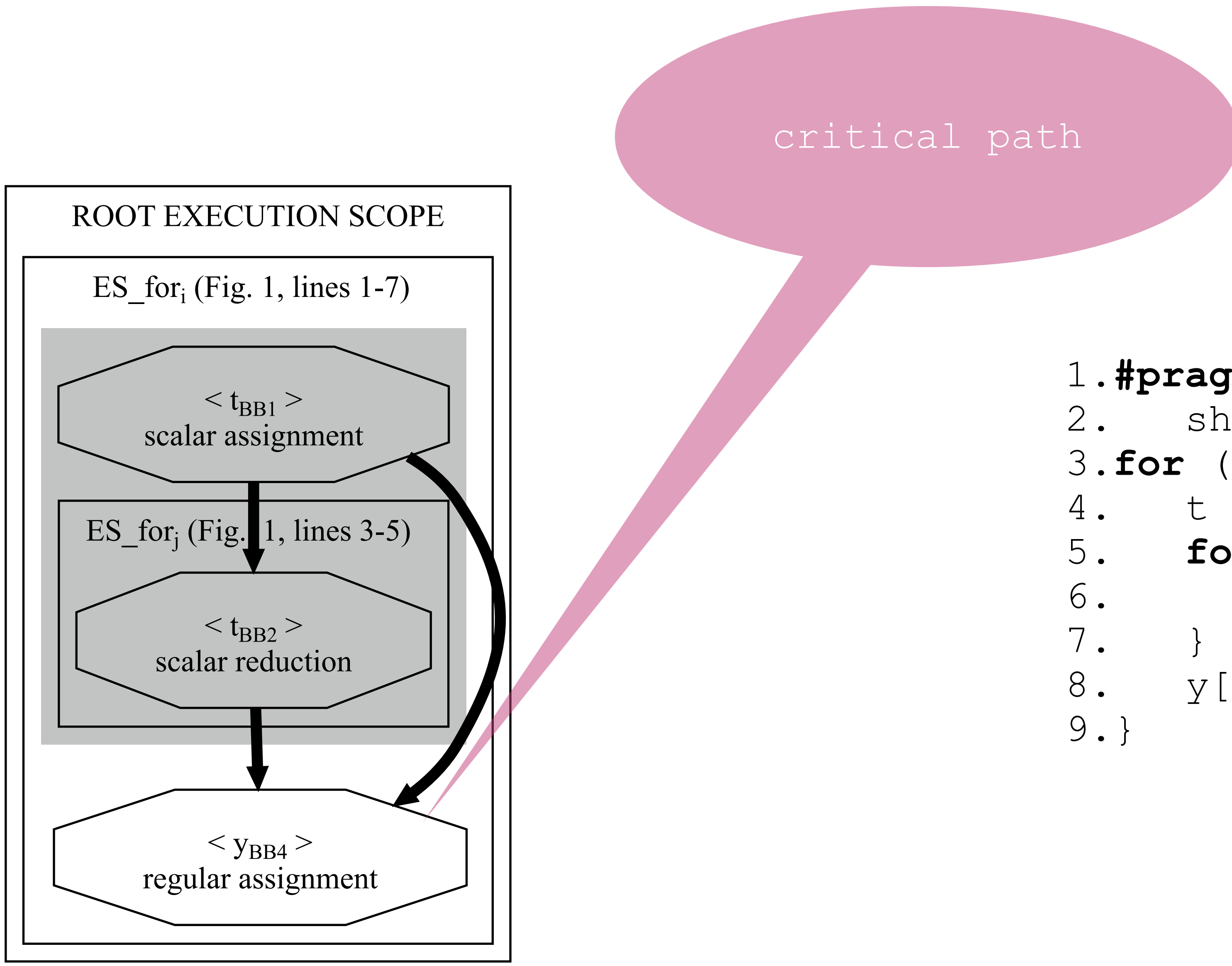
Automatic Partitioning driven by the KIR (III)

critical path

```
1.for (i = 0; i < n; i++) {
2.    t = 0;
3.    for (j = 0; j < m; j++) {
4.        t = t + A[i][j] * x[j];
5.    }
6.    y[i] = t;
7.}
```



Automatic Partitioning driven by the KIR (and IV)




```
1.#pragma omp parallel for
2.    shared(A,x,y) private (t,i,j)
3.for (i = 0; i < n; i++) {
4.    t = 0;
5.    for (j = 0; j < m; j++) {
6.        t = t + A[i][j] * x[j];
7.    }
8.    y[i] = t;
9.}
```

Outline

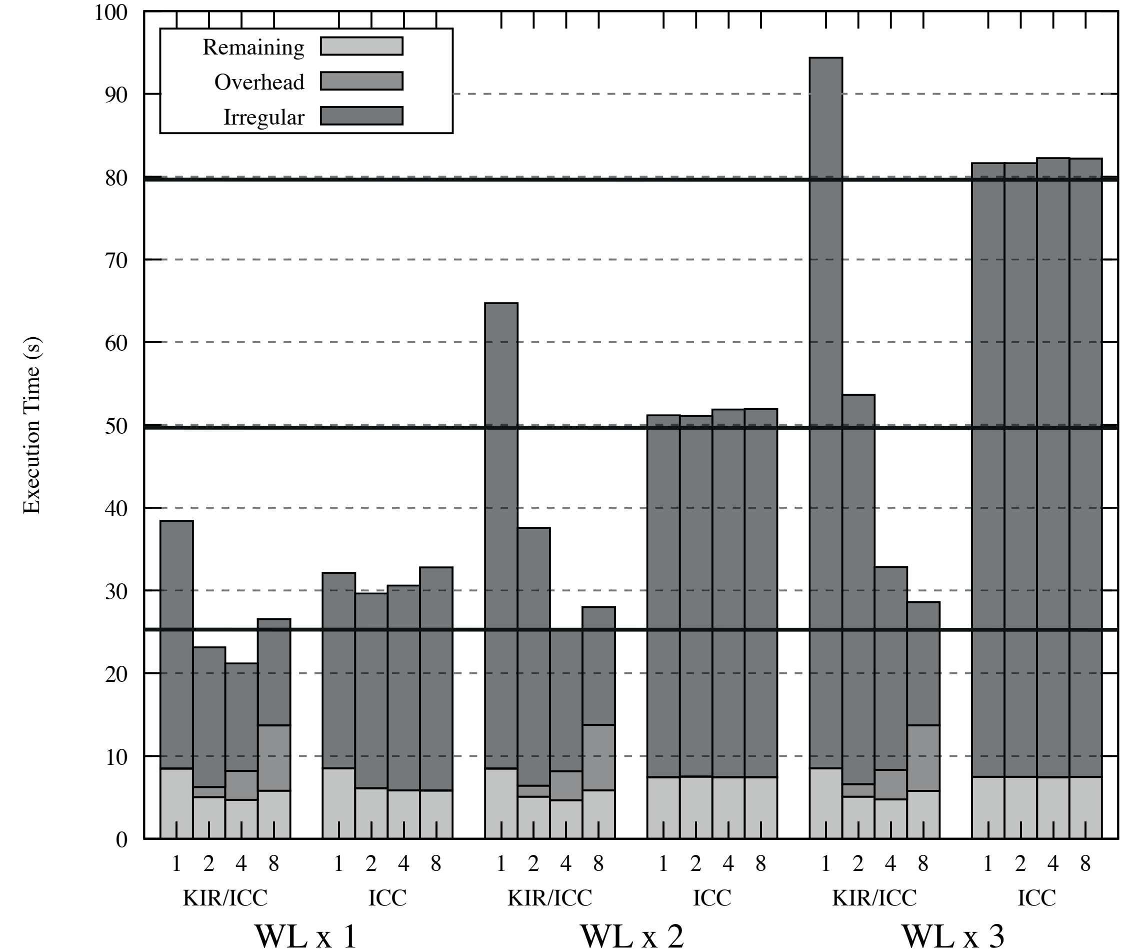
- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

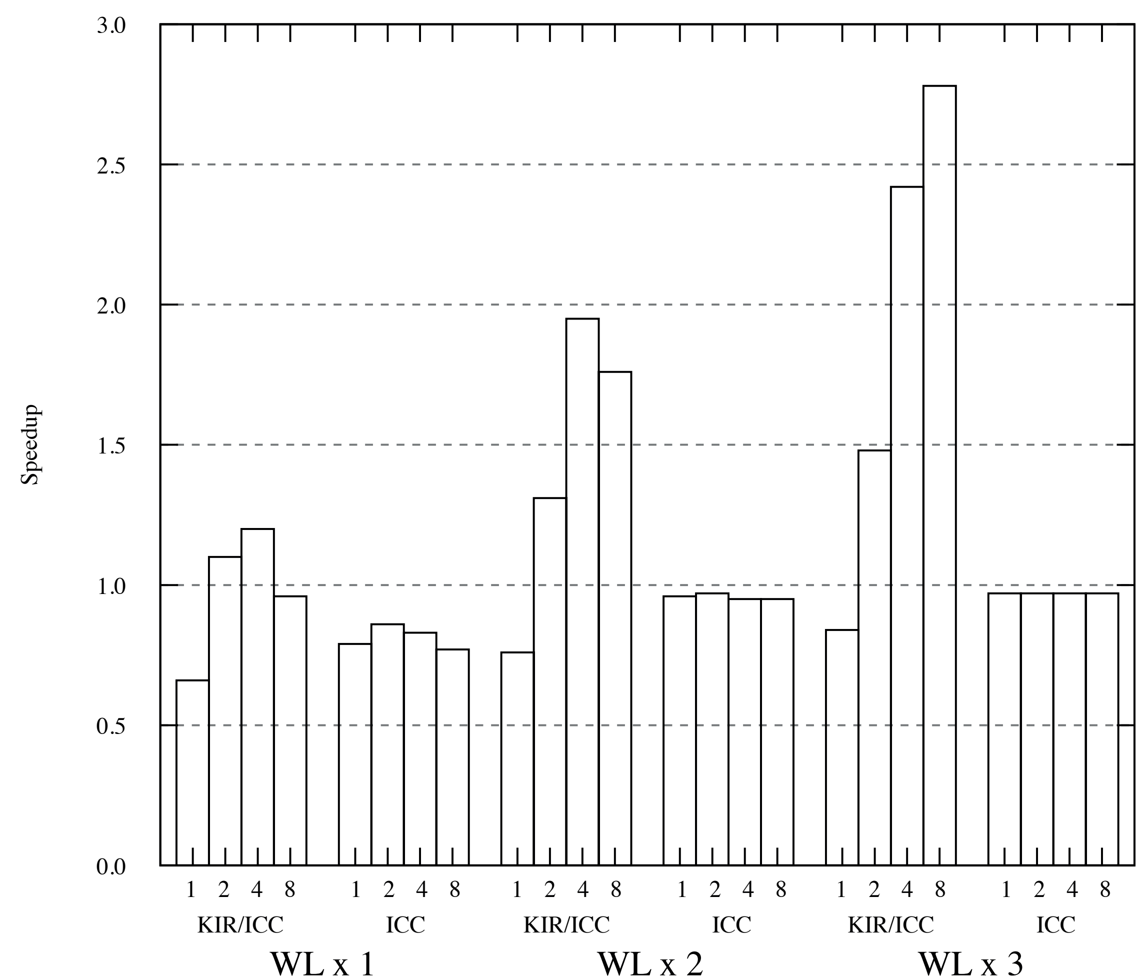
Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

Experimental Evaluation

- Built on top of GCC 4.4.0
- EQUAKE from SPEC CPU2000 on 2 Intel Xeon E5520 quad-core processors
- The Intel compiler is unable to parallelize this case study properly while our approach reduces the execution time
- More results on *J.M. Andión et al. A Novel Compiler Support for Automatic Parallelization on Multicore Systems. Parallel Computing, 39(9), 2013.*





Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- Conclusions

Outline

- Motivation: The Parallel Challenge
- KIR: A diKernel-based IR
- Automatic Partitioning driven by the KIR
- Experimental Evaluation
- **Conclusions**

1. The KIR: a diKernel-based IR
    - diKernels
    - diKernel-level dependences
    - execution scopes
2. Automatic Partitioning Technique
    - coarse-grain parallelism
    - global OpenMP parallelization strategy

Future Work

- Locality exploitation techniques
- Fine-grain parallelism
- Many-core architectures such as GPUs

J.M. Andión et al. Locality-Aware Automatic Parallelization for GPGPU with OpenHMPP Directives. HLPP 2014 & International Journal of Parallel Programming (to appear)

A Parallelizing Compiler for Multicore Systems

José M. Andión, Manuel Arenaz, Gabriel Rodríguez and Juan Touriño

17th International Workshop on Software and Compilers for Embedded Systems (SCOPES 2014)
June 10-11, 2014 — Schloss Rheinfels, Sankt Goar, Germany

COMPUTER ARCHITECTURE GROUP
UNIVERSITY OF A CORUÑA
UNIVERSIDADE DA CORUÑA