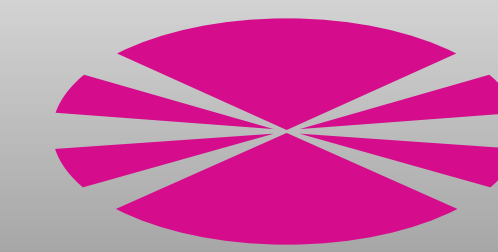


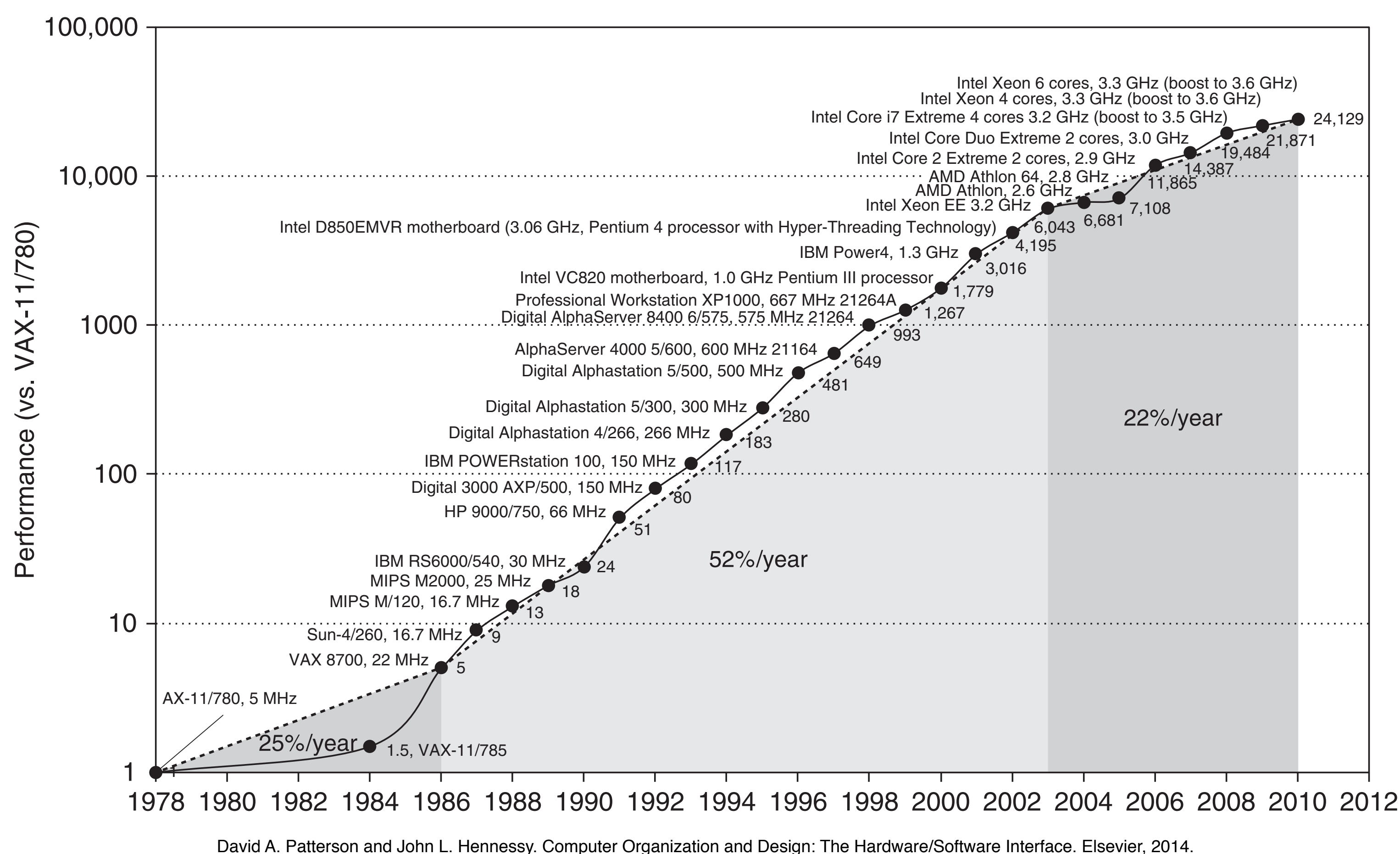
A Parallelizing Compiler for Multicore Systems

José M. Andión, Manuel Arenaz, Gabriel Rodríguez and Juan Touriño
 {jandion,manuel.arenaz,grodriguez,juan}@udc.es



UNIVERSIDADE DA CORUÑA

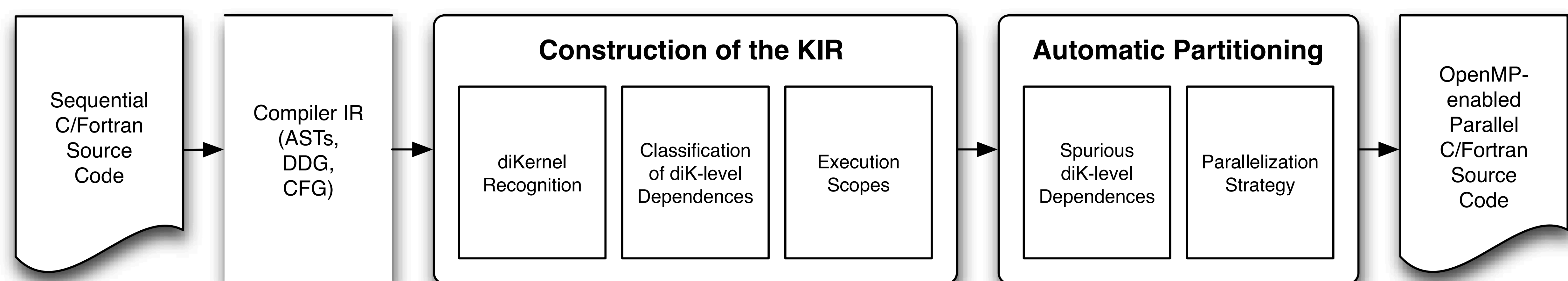
Motivation



The Parallel Challenge

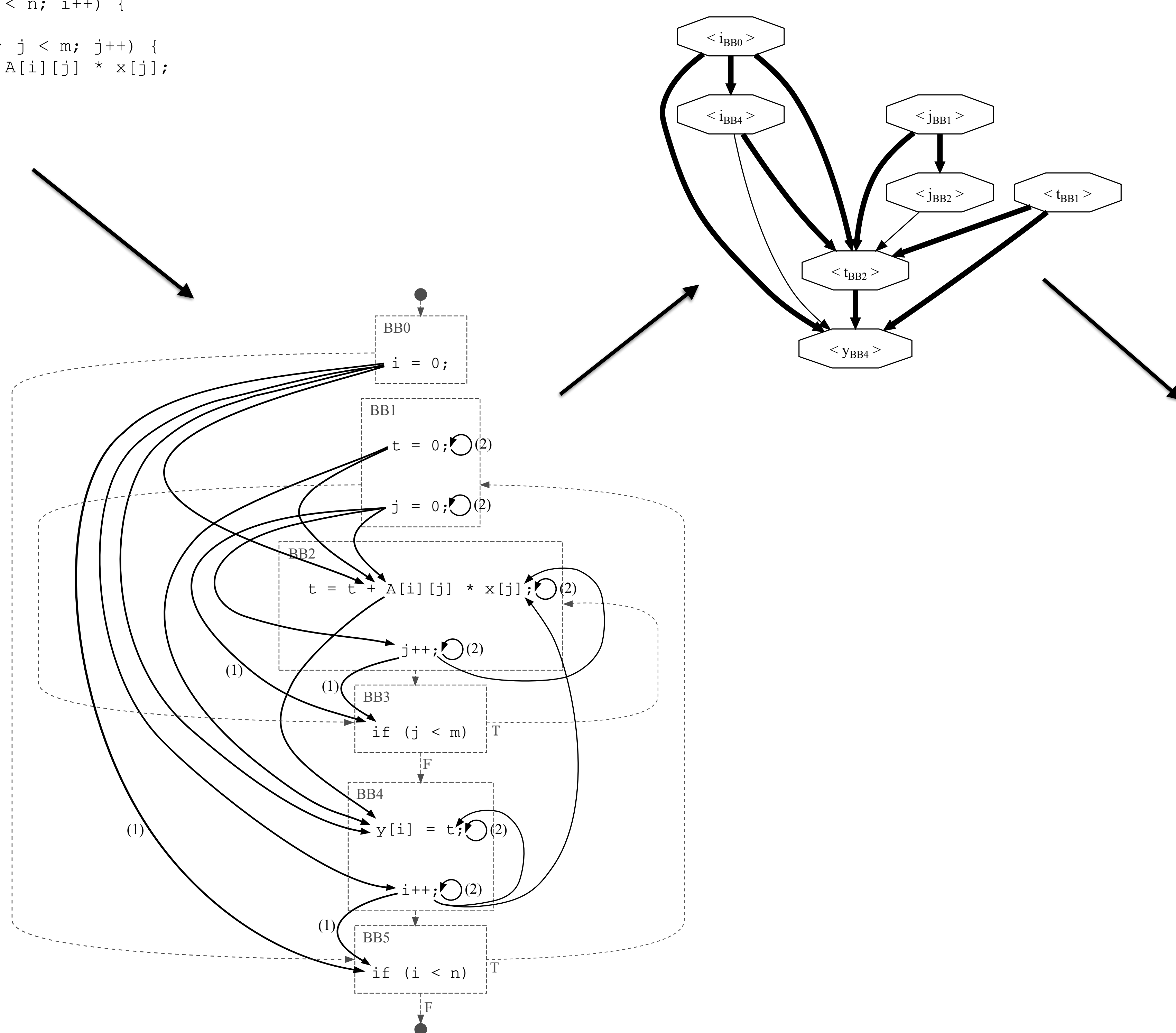
- libraries: MPI, CUDA...
- compiler directives: OpenMP, OpenACC...
- programming languages: PGAS...
- parallelizing compilers: GCC, ICC, PLUTO...

Automatic Parallelization driven by diKernels



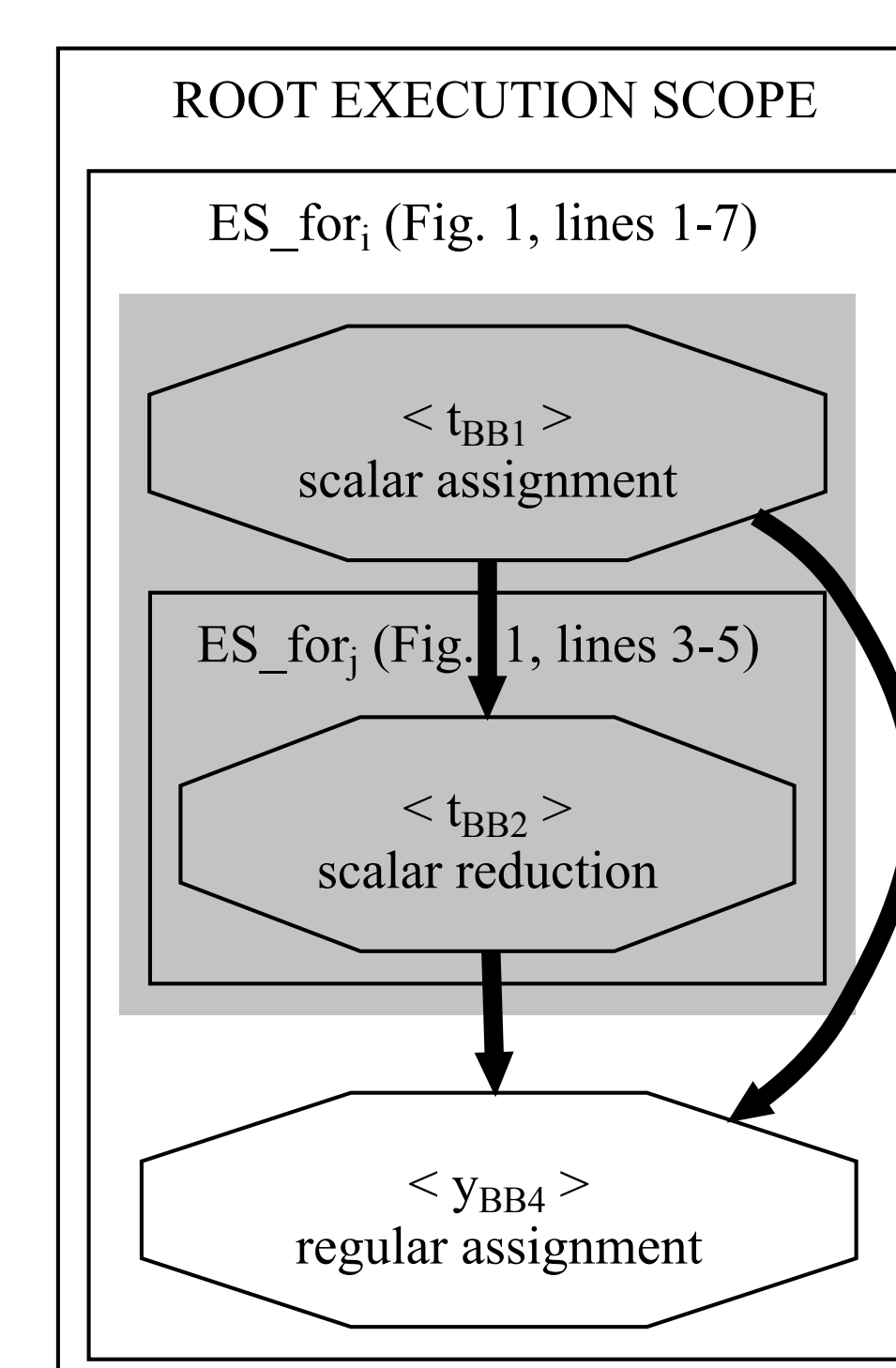
```

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. }
    
```



```

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



Experimental Evaluation

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

