A Task Graph Representation for Flexible Hardware/Software Partitioning

Tiago Santos, João Bispo, João M. P. Cardoso

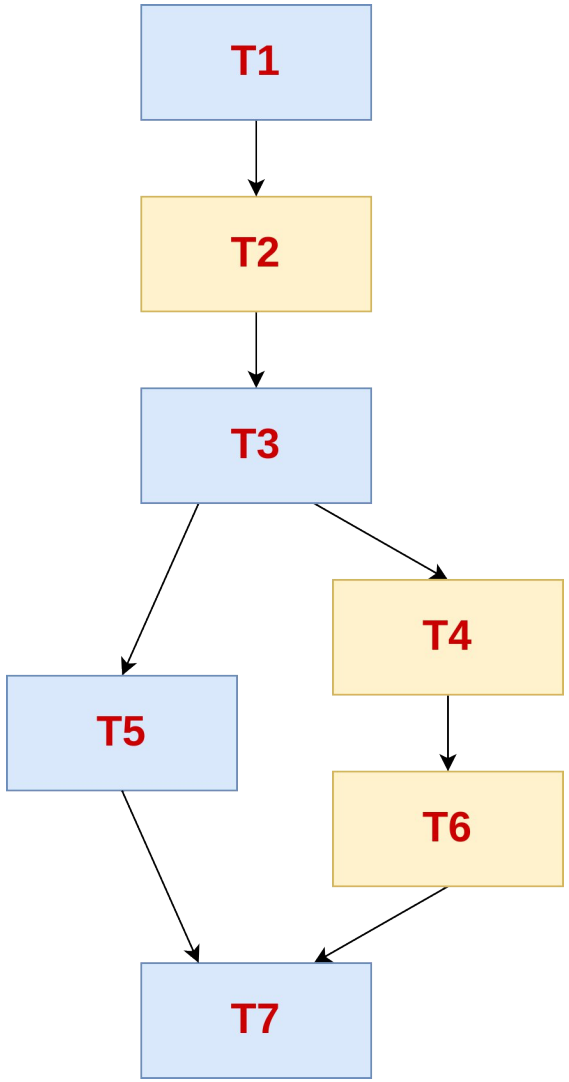
HiPEAC Workshop on Reconfigurable Computing (WRC' 24)
17th of January 2024, Munich, Germany

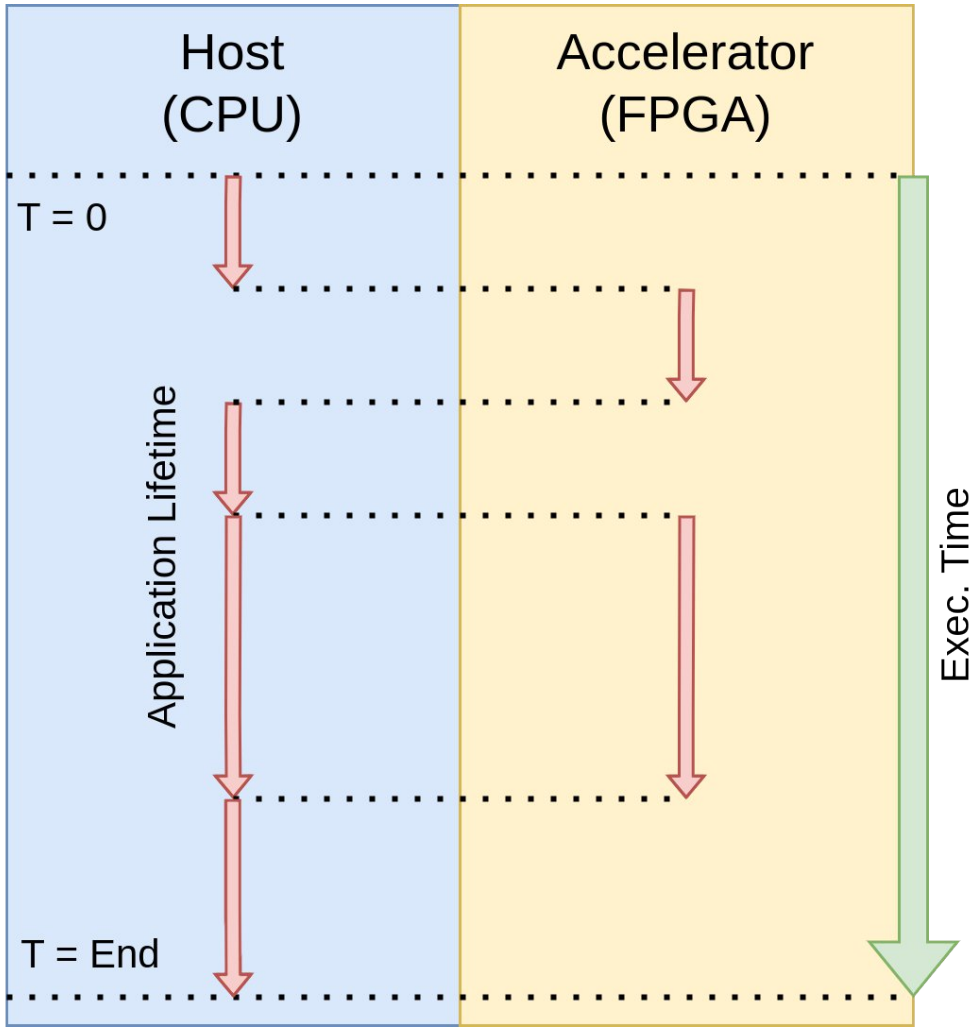
U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

INESCTEC

SPeCS
Special-Purpose Computing
Systems, languages and tools

Hardware/Software Partitioning





7 tasks $\rightarrow 2^7$ possible solutions, and more if accounting for variants/versions of each task or clusters!

Task Granularity for Holistic HW/SW Partitioning

```
(...)
```

Possible code region for offloading?

```
convolve2d(output, filter, image_gray);


filter[0] = 1;
filter[1] = 2;
//...
filter[7] = -2;
filter[8] = -1;


convolve2d(output, filter, temp_buf);
```

```
combthreshold(image_gray, temp_buf, output);

(...)
```

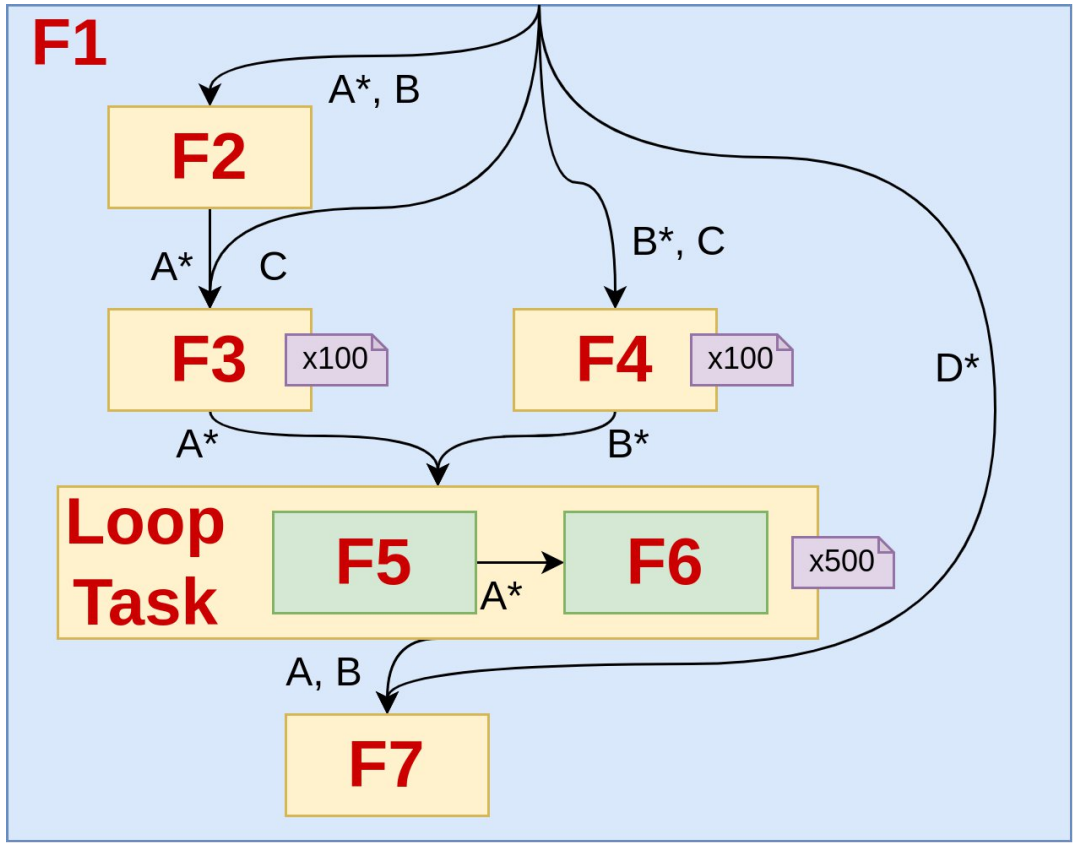
How can we enable tasks with flexible granularity? Mix up statements, loop bodies, functions...

How can we merge and split tasks while keeping as much as possible the original source code?

How can we expose data communication, and create clusters of tasks across shared data?

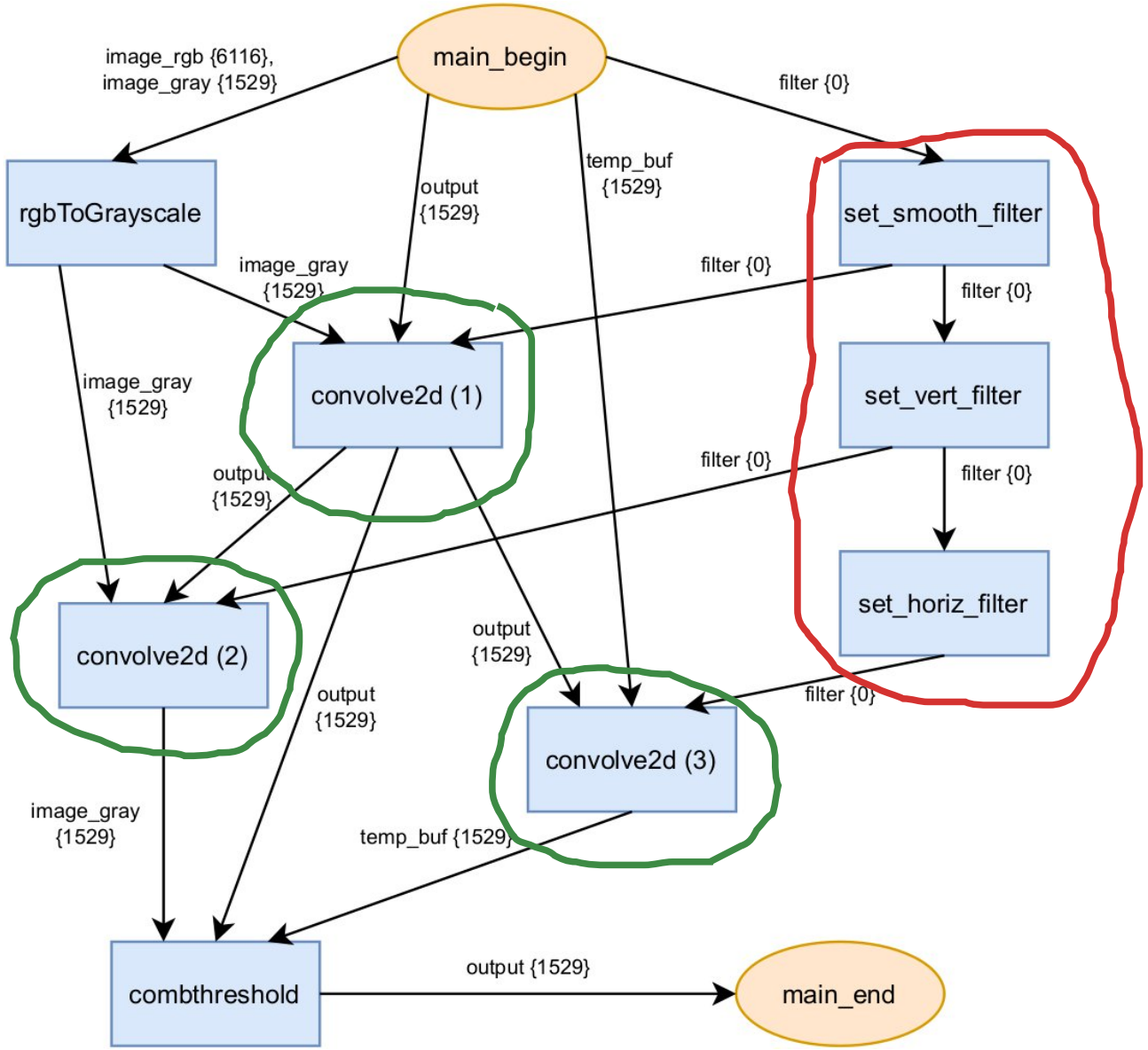
Our Task Graph Formulation

```
void F1(int *A, int *B, int *C, int *D) {
    F2(A, B);

    for (int i = 0; i < 100; i++) {
        F3(A, C[i]);
        F4(B, C[i]);
    }

    for (int i = 0; i < 500; i++) {
        F5(A, B);
        F6(A);
    }

    F7(A, B, D);
}
```



AST Transformation	Task Graph Transformation
Function Inlining	Merging Tasks
Function Outlining	Splitting a Task

Exposing Data Communication



Multigraph with one data item per edge, with its size and communication cost

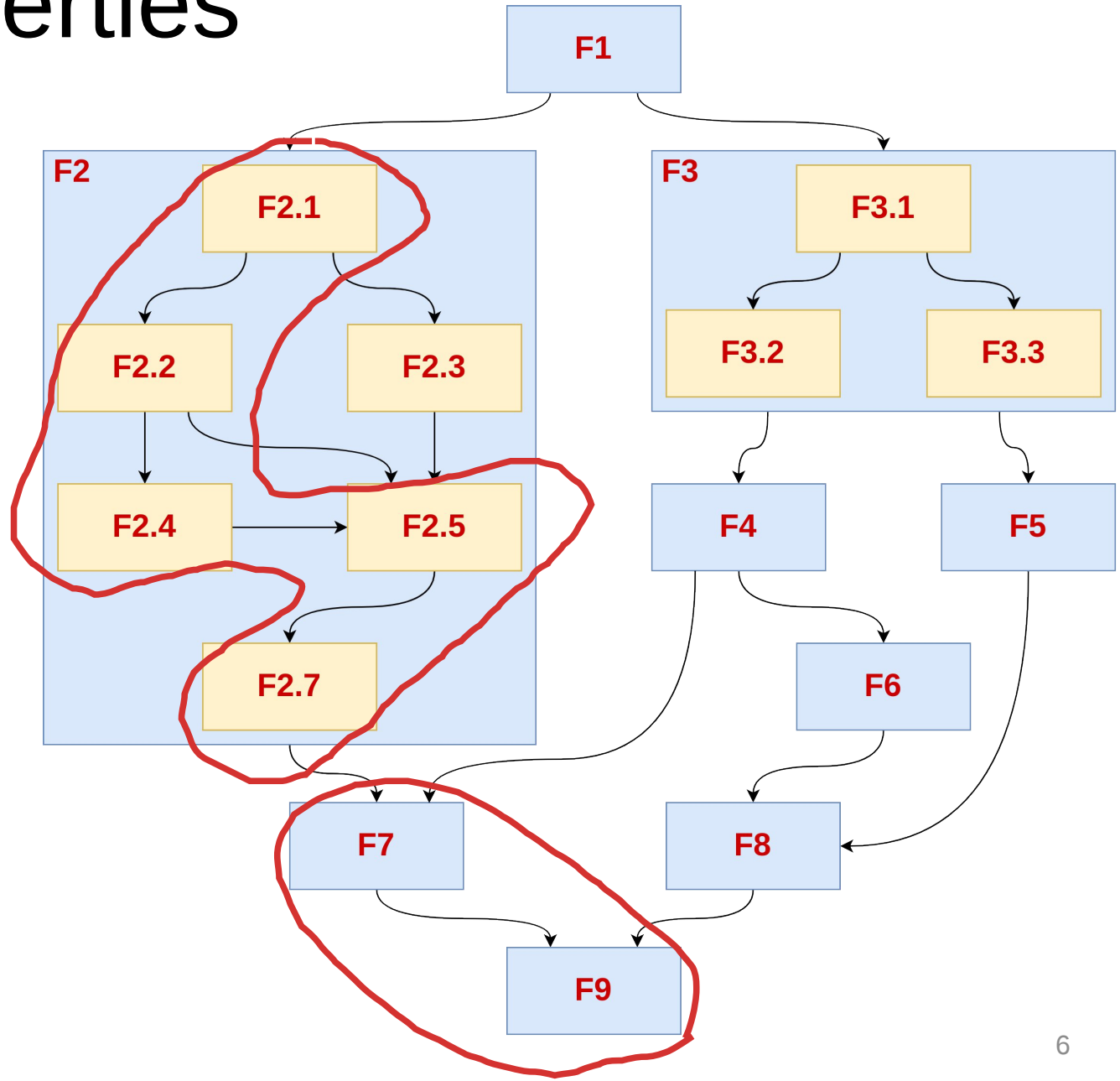
Find the cluster of tasks that modify the same data item

Find nearby tasks that use copies of the data item

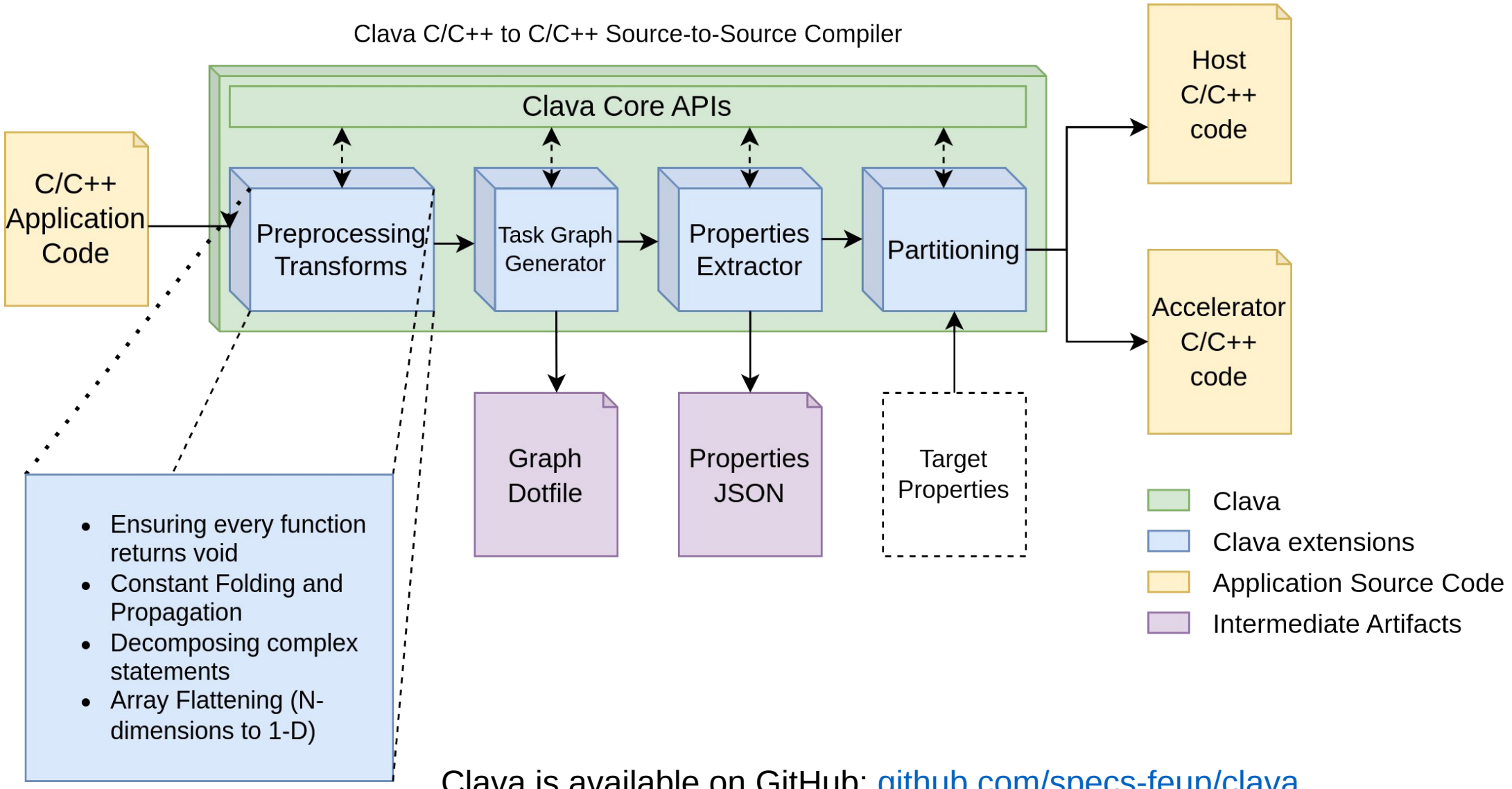
Task Graph Properties

- **Subgraph level of parallelism**: ratio of the subgraph's critical path and its total number of tasks

- **Dataflow regions**: pairs of tasks with a producer-consumer relationship may suggest the presence of a dataflow region



Tool Flow



Clava is available on GitHub: github.com/specs-feup/clava
Task Graph extension will be open-sourced in the very near future!

Concluding Remarks

Hierarchical Task Graph as an overlay of the AST

Enables creation of arbitrary clusters with flexible granularity while preserving source code

Exposes data communication and traceability between tasks

Experimental results for **Rosetta** and **MachSuite** show potential in parallelizing tasks and exploiting dataflow regions

Ongoing work: perform partitioning and code optimization as a single-pass holistic process

Thank you for listening!

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

INESCTEC

SPeCS
Special-Purpose Computing
Systems, languages and tools