

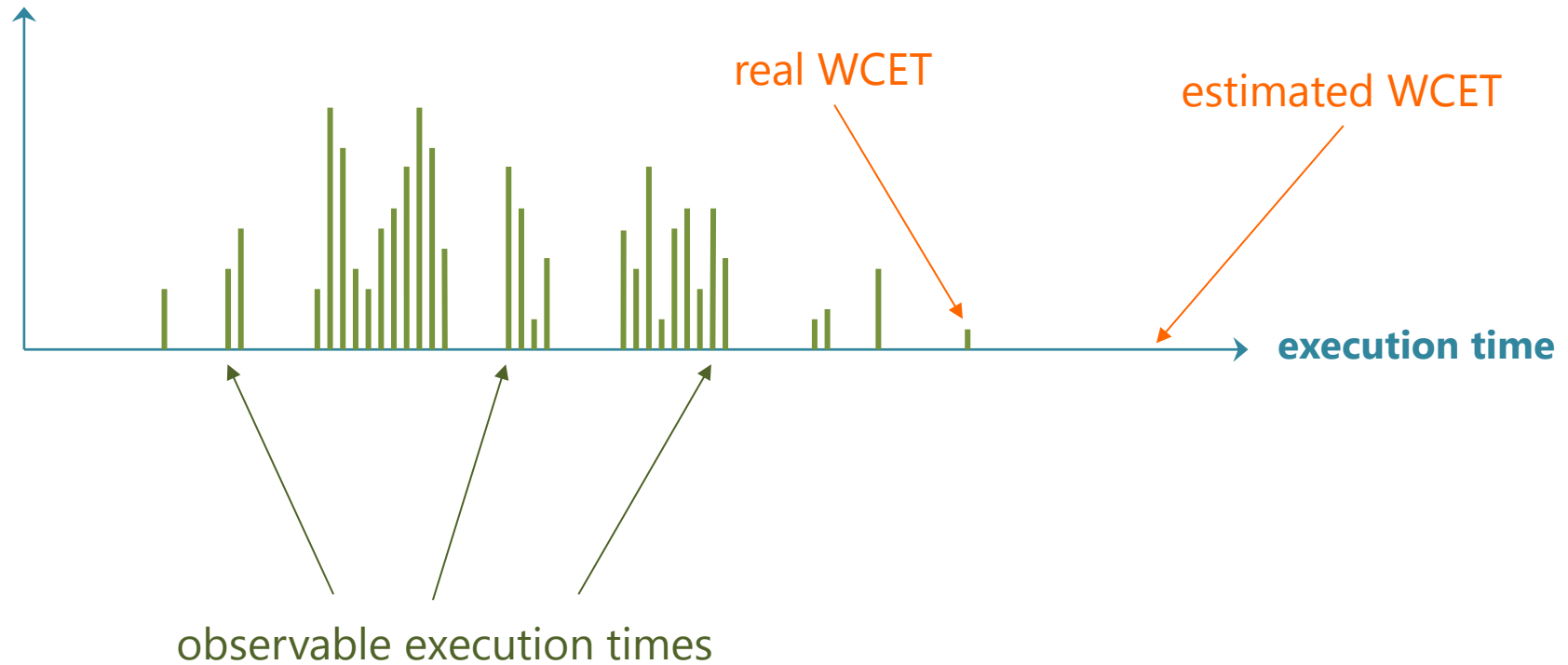
Static WCET analysis of parallel applications

Christine Rochange

University of Toulouse, France

parMERASA Dissemination Event,
Barcelona, 2014-09-23

distribution



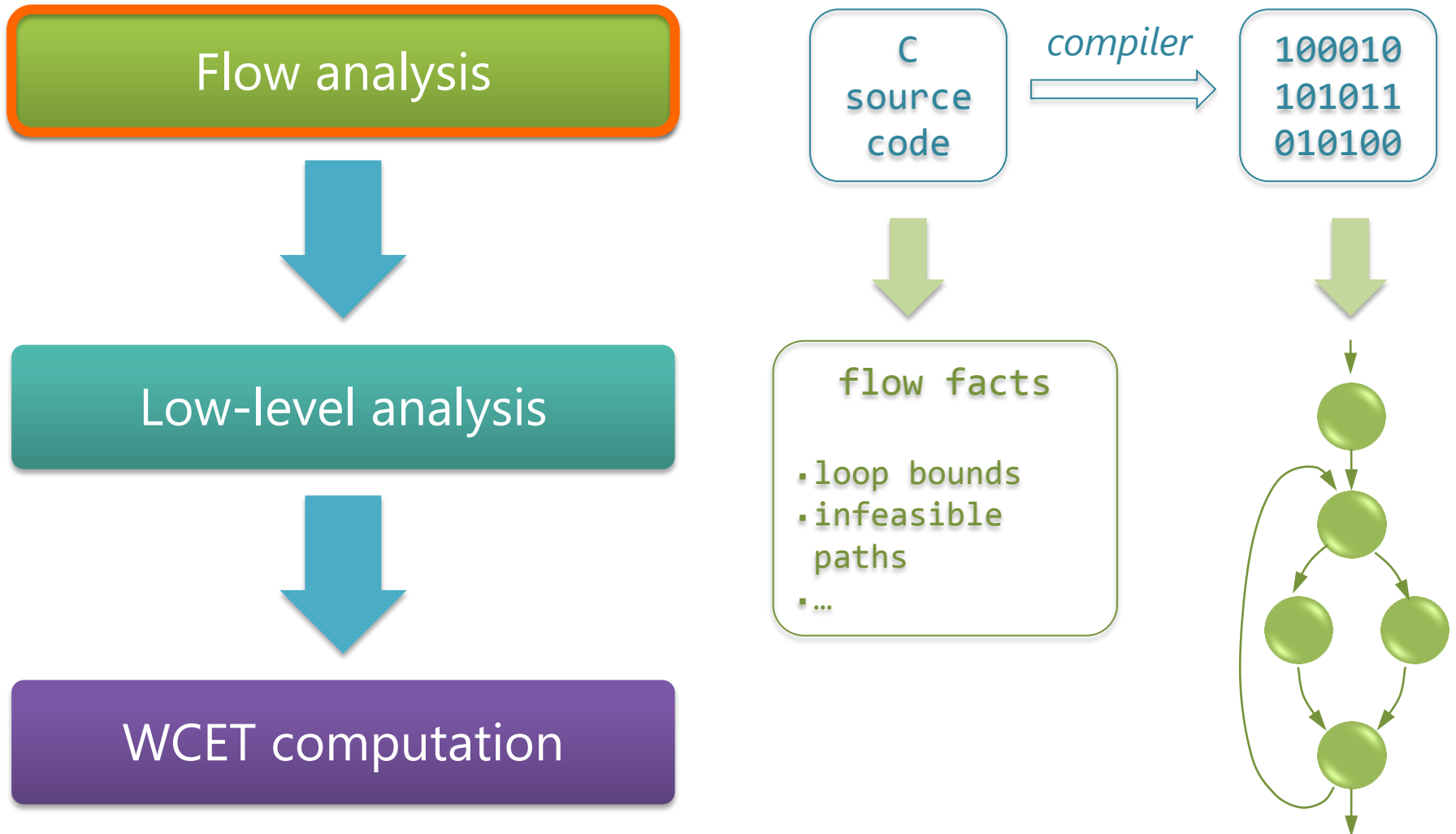
Flow analysis

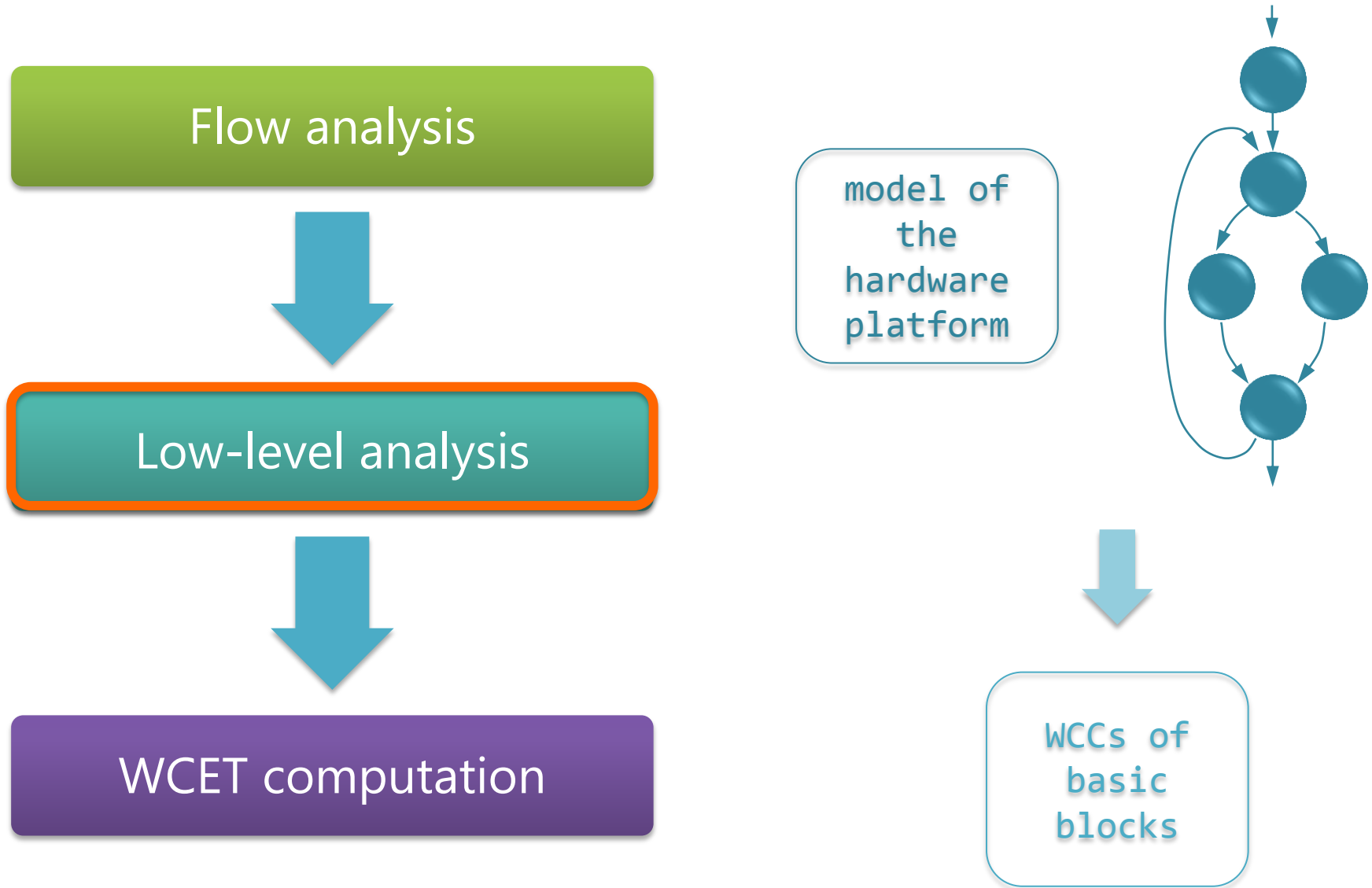


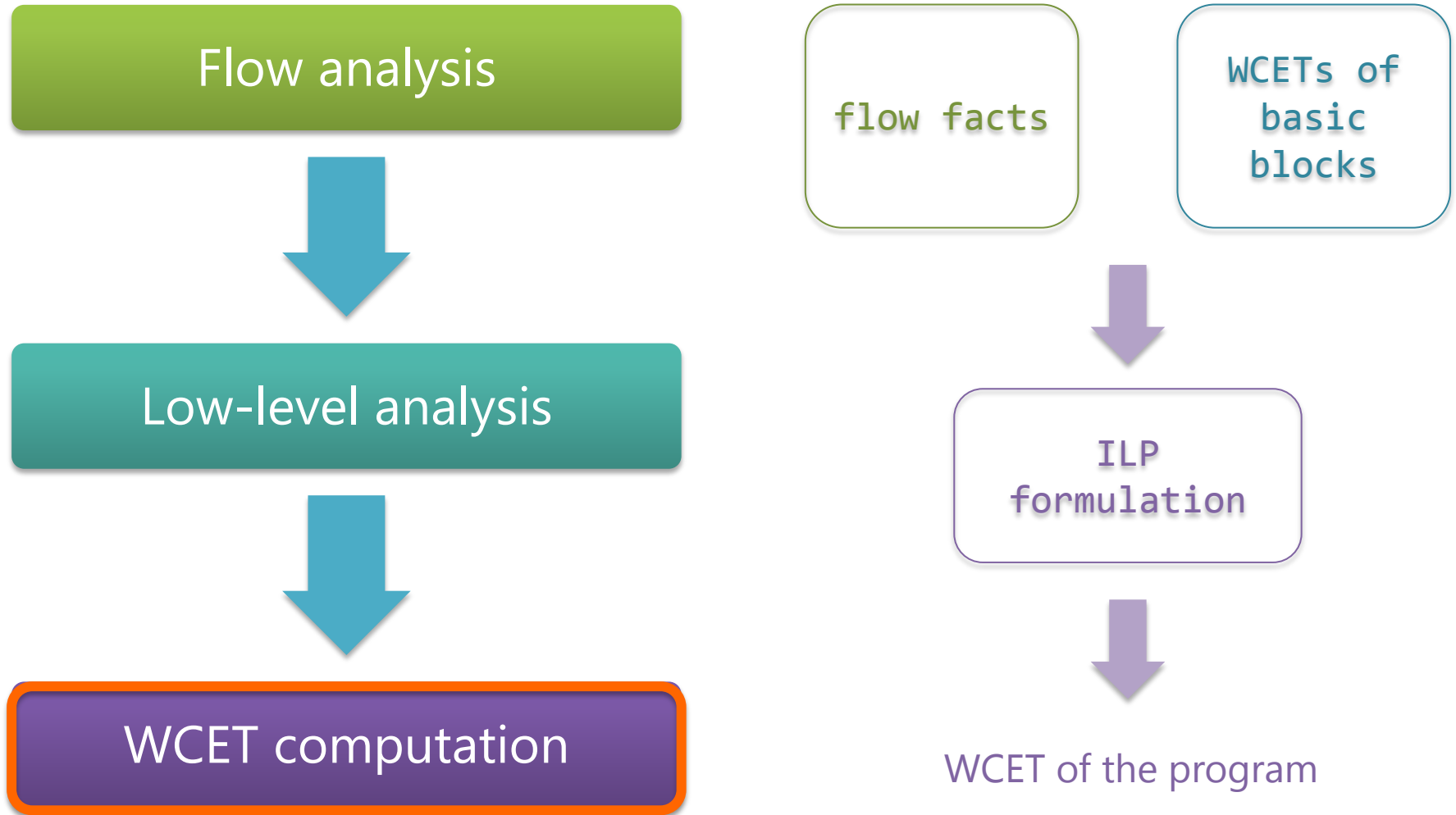
Low-level analysis



WCET computation







- **Contention for shared resources**

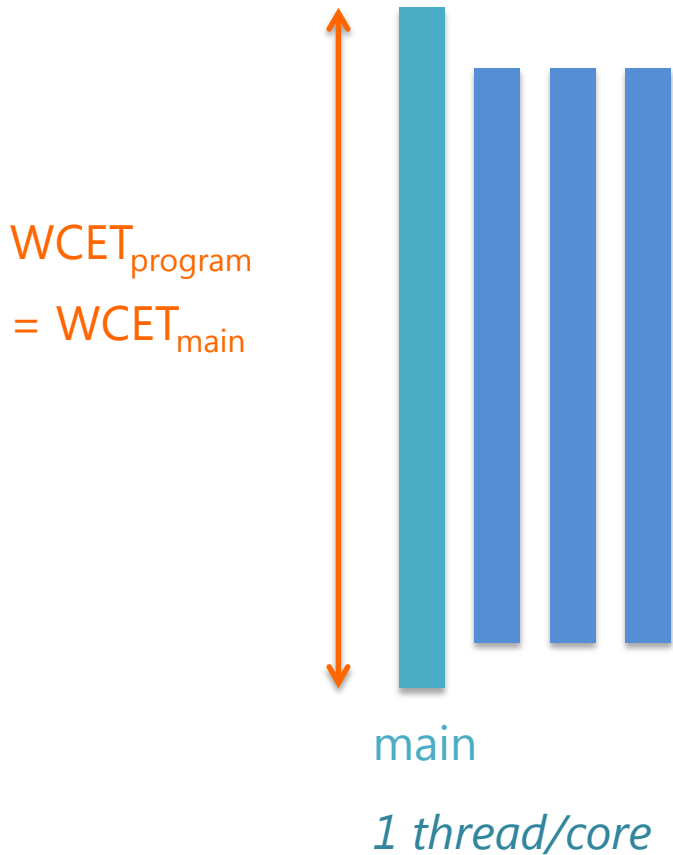
- network and memory
 - latency?
- shared cache
 - impact of interferences/pollution on cache behaviour?

- **parMERASA architecture**

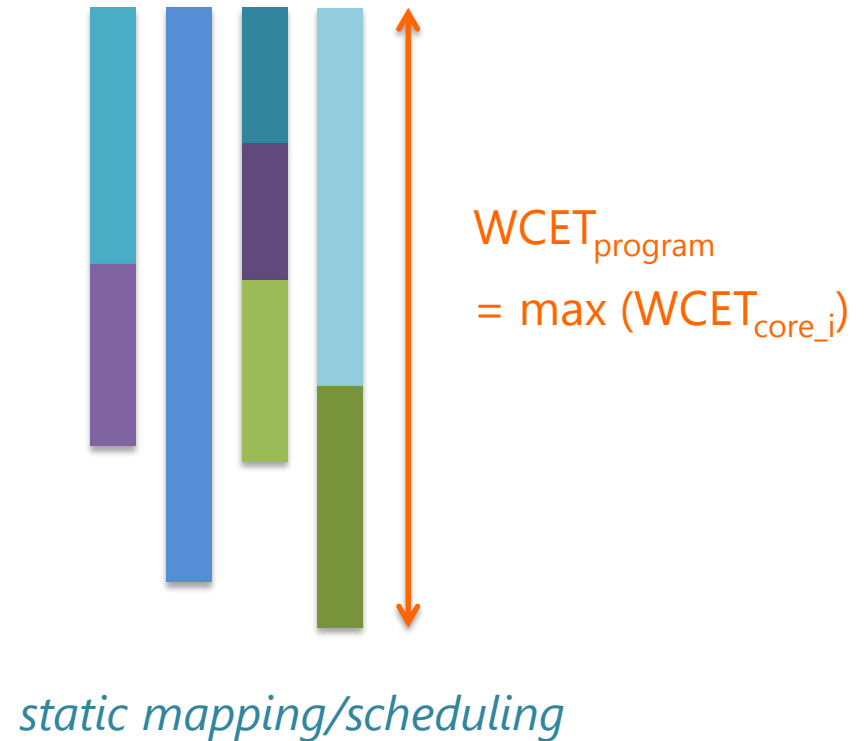
- bounded latencies, private caches

**time
composability**

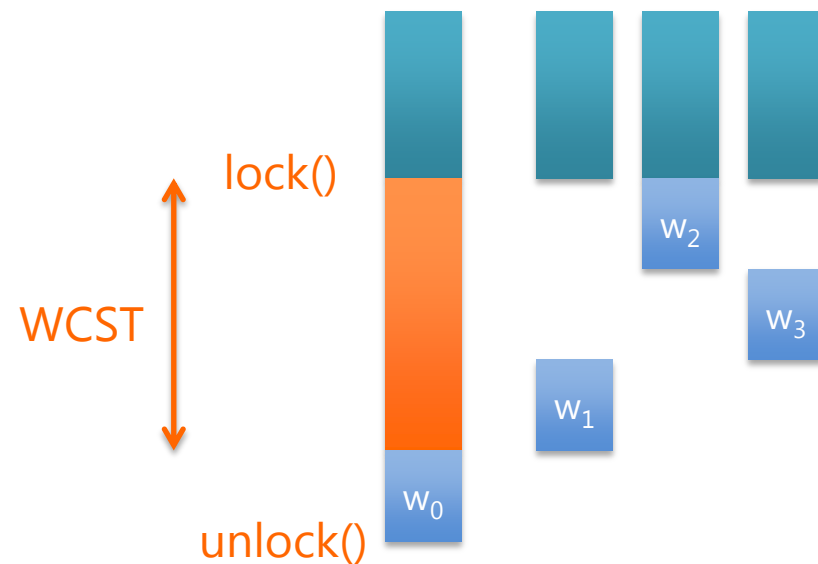
▪ Data parallelism



▪ Task parallelism

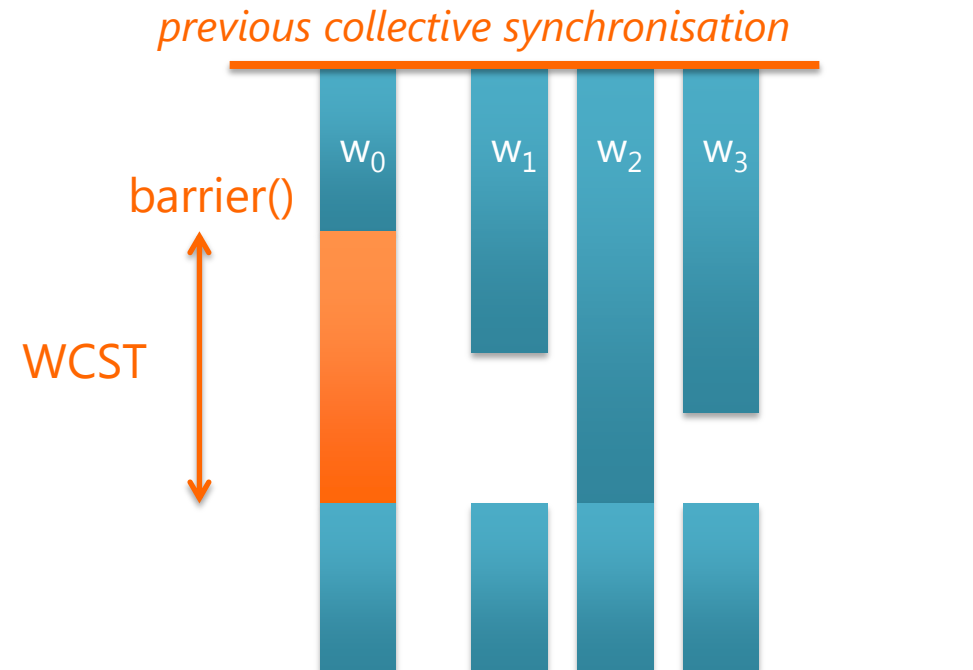


Critical section

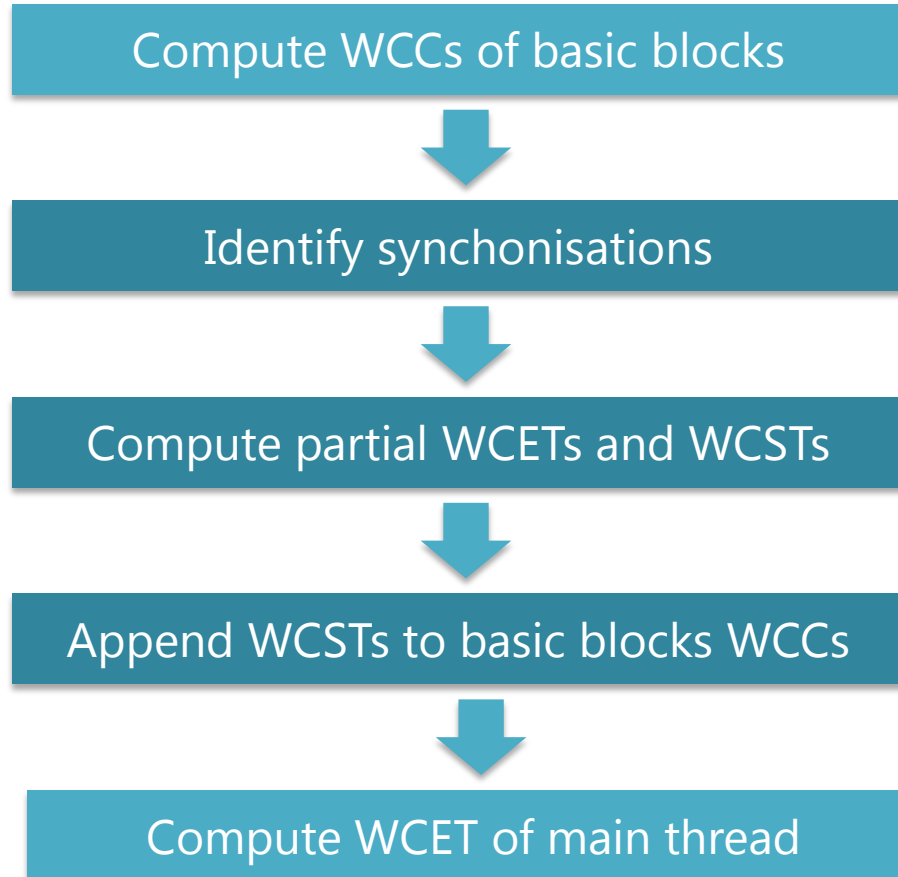


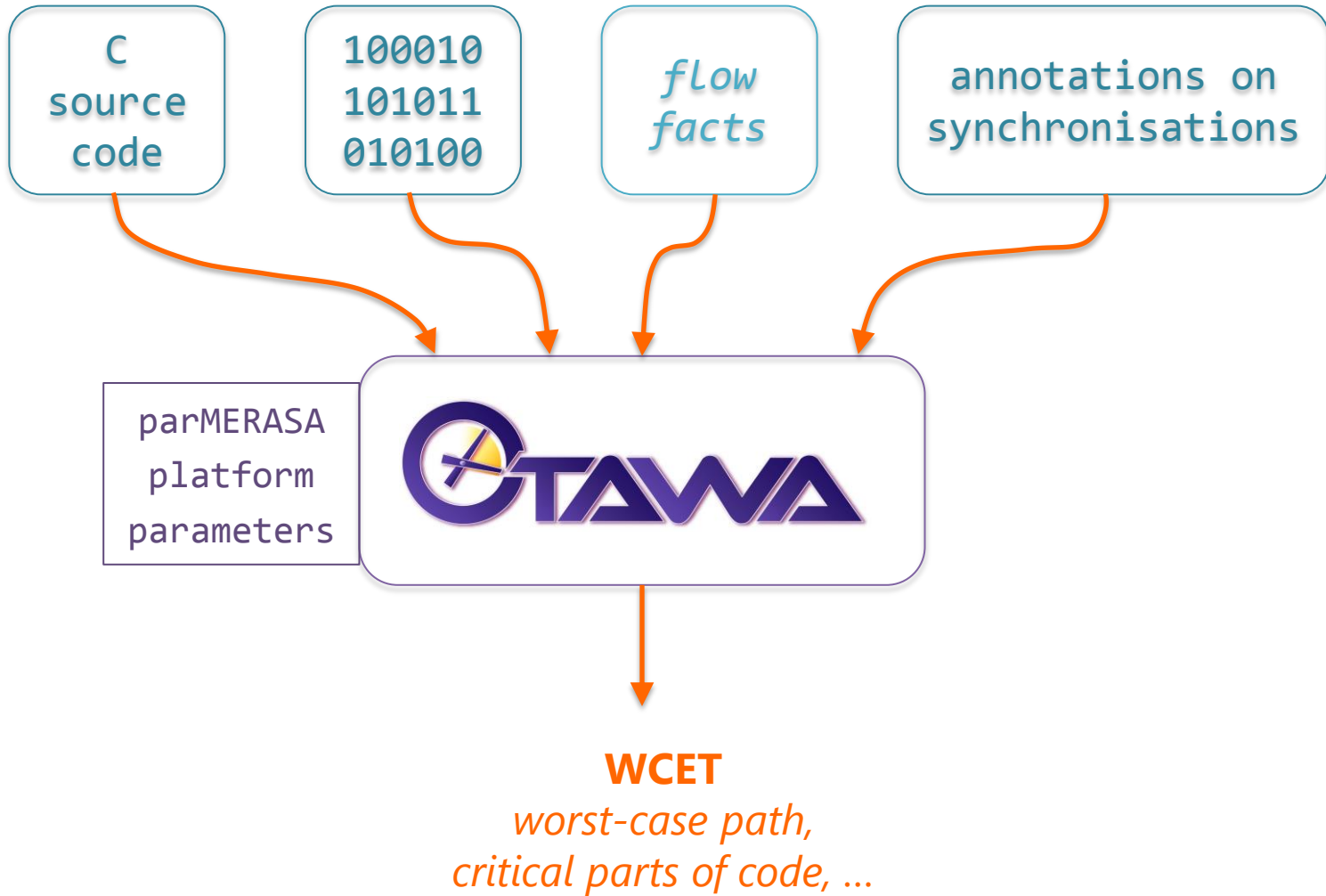
$$WCST = \sum_{\text{possible contenders}} w_i$$

Barrier



$$WCST = \max (0, w_1-w_0, w_2-w_0, w_3-w_0)$$





▪ Facilitate flow analysis


- well structured code
- boundable loops
- *etc.*



▪ Favour well structured parallelism

- use the **design pattern catalogue**
 - ensures analysability
 - facilitates the generation of annotations
- prefer **collective synchronisations**
 - e.g. barriers instead of point-to-point signal/wait



- **Static WCET analysis of parallel applications is feasible!**
 - **time-predictable platform** (hardware + system software)
 - static mapping/scheduling of threads
 - easier for well structured programs
 - use recommended **design patterns & skeletons**
 - **tight** WCET estimations can be achieved
 -  available as open-source software