

METHODOLOGY

- ANALYSIS

1) - analyze the problem

- understand what's going on
- " / extract all parallelism related features
- (evaluate) alternatives
produce

2) evaluate the different choices/alternatives

- ↳ { a) use abstract perf models
- b) assuming unknown parameters the same in all cases
- c) commit choices only in case of evident advantage(s)

3) choose proper development tools/ programming frameworks

3.a) the tools/frameworks must support the choices made

3.b (!!)) the tools/frameworks must guarantee software reuse

4) sequential wrappers

- parametric
- reusable in different skeleton contexts
- reuse as much as possible the (tuned) existing code

5) Prototype "structural" solutions

- far from complete solution
- capturing all parallelism related details
- provide feedback to performance models
- evaluate alternatives left from 1) & 2)

6) Complete solution

↳ choose the best among the alternatives

↳ refine the prototype

↳ provide all the business logic code/params

↳ provide all low level (non functional) params that have been left unspecified (up to now)

7. (short) : seq debugging (for correctness is "guaranteed")

7) (Time) tuning → could require "restructuring" of your application

- using performance models

- or (if available) the "hints" from the framework (e.g. the "BLAMES" of Scaadium calcium)