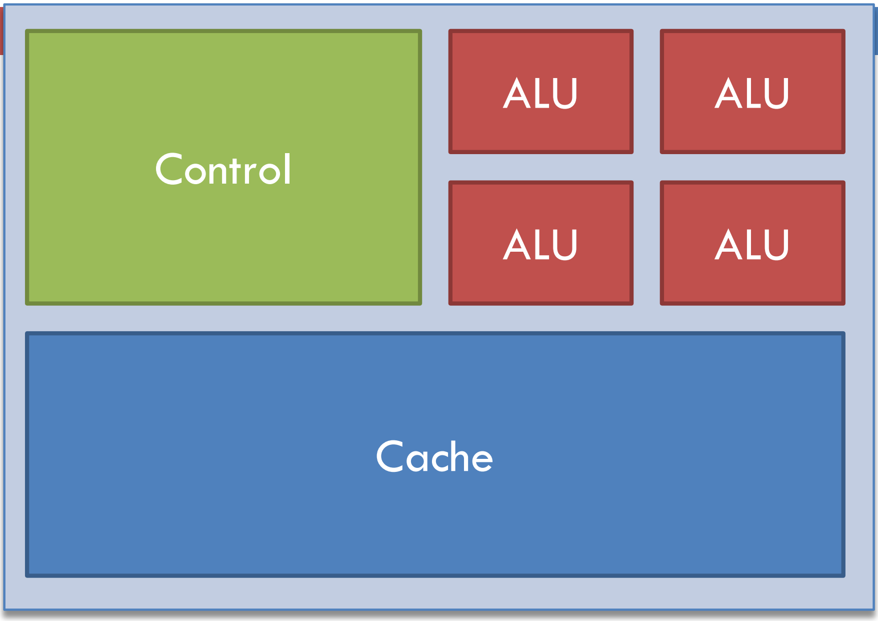
CPU + GPU HETEROGENEOUS COMPUTING (THE LAB)

Ana Lucia Varbanescu, University of Amserdam, NL
a.l.varbanescu@uva.nl

With significant contributions by:
Jie Shen @ Delft Unviersity of Technology
Stijn Heldens @ U Twente

CPU + GPU

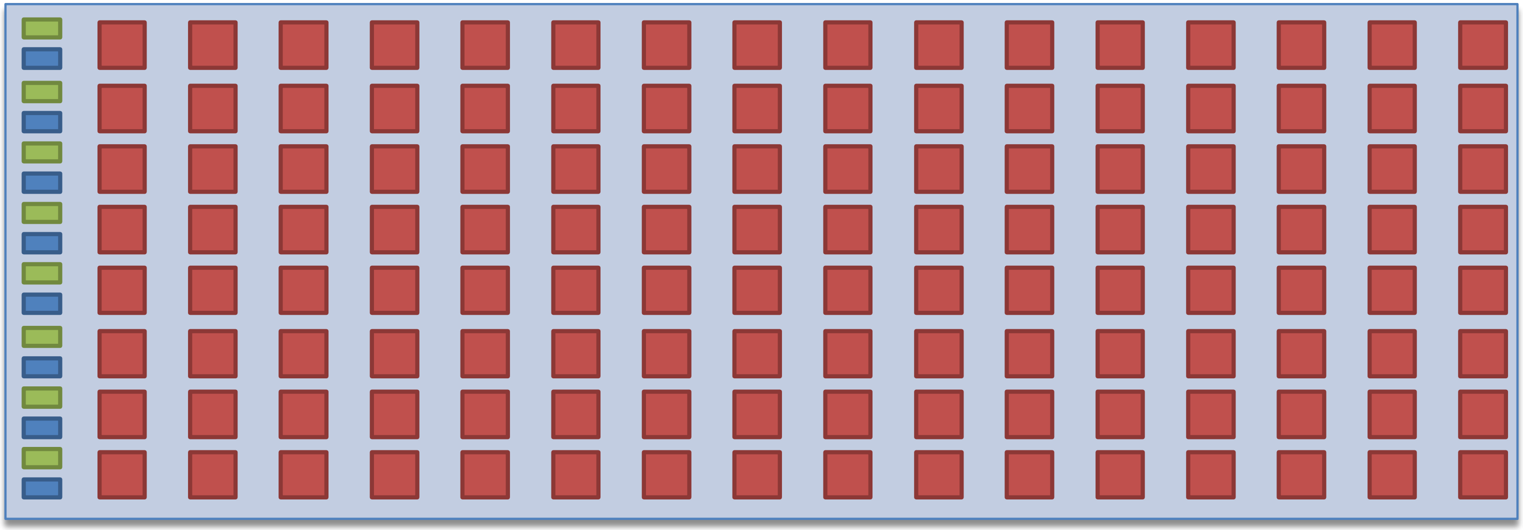


CPU

Low latency, high flexibility.
Excellent for irregular codes with
limited parallelism.

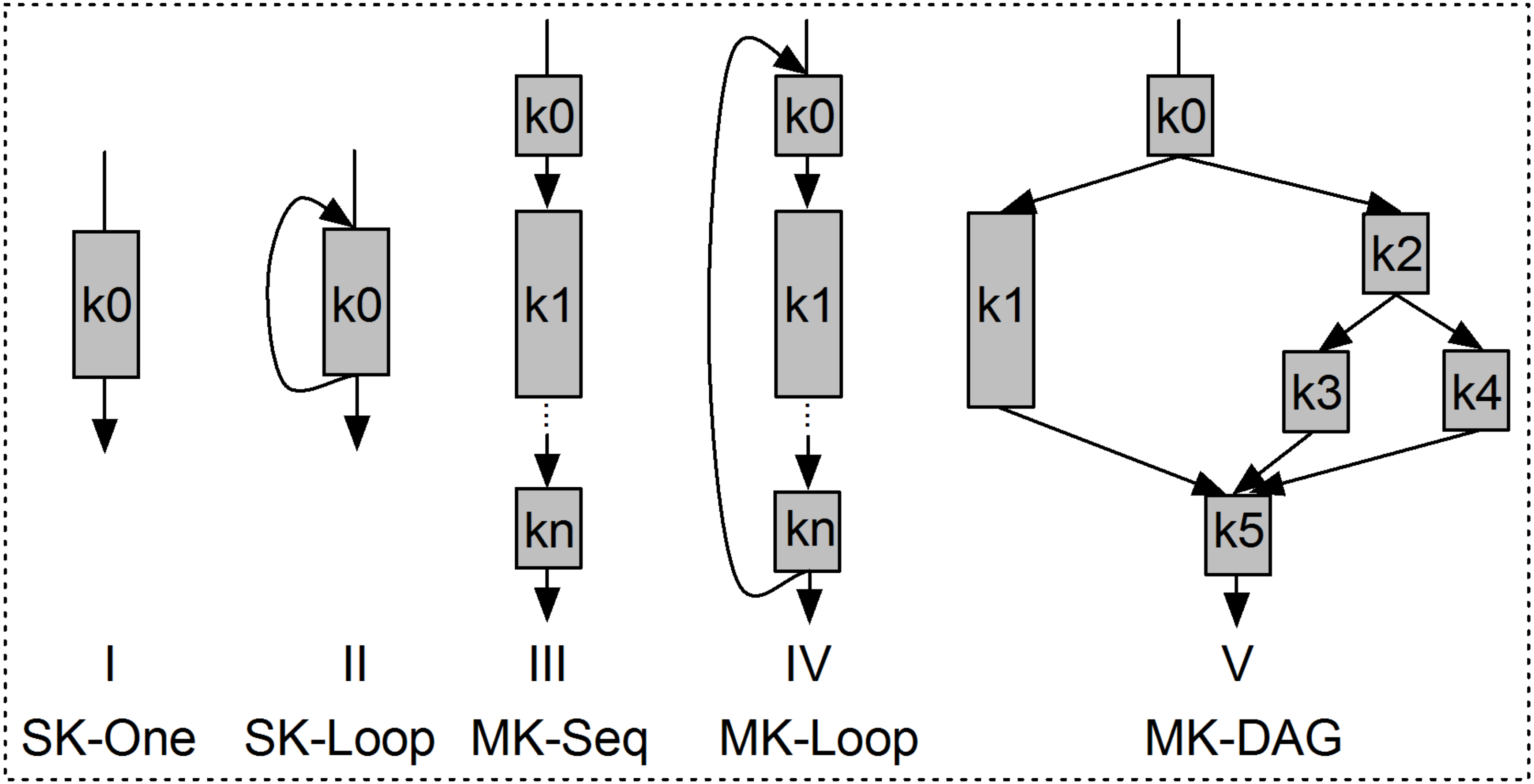
GPU

High throughput.
Excellent for massively parallel workloads.



Workload

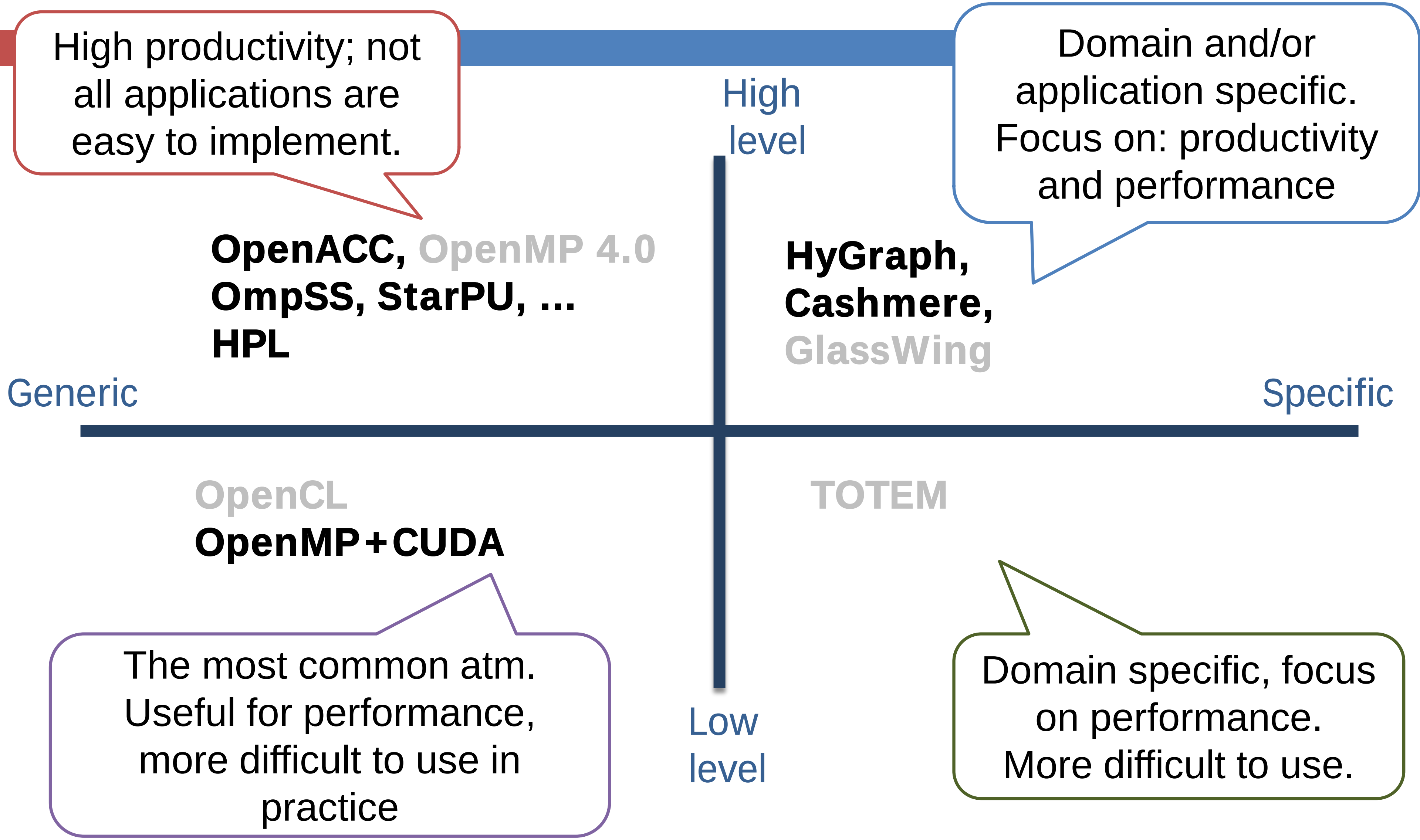
- DAG (directed acyclic graph) of “kernels”



Two main challenges

- How to **program** the workload in a "user-friendly" and "heterogeneity-friendly" way
  - Unified programming models ?
  - Low- or high-level ?
  - …
- How to **split** the work between the different processors
  - Static
  - Dynamic
  - Hybrid … ?

(1) Heterogeneous Computing PMs



High productivity; not all applications are easy to implement.

Domain and/or application specific. Focus on: productivity and performance

High level

OpenACC, OpenMP 4.0
OmpSS, StarPU, …
HPL

HyGraph,
Cashmere,
GlassWing

Generic

Specific

OpenCL
OpenMP+CUDA

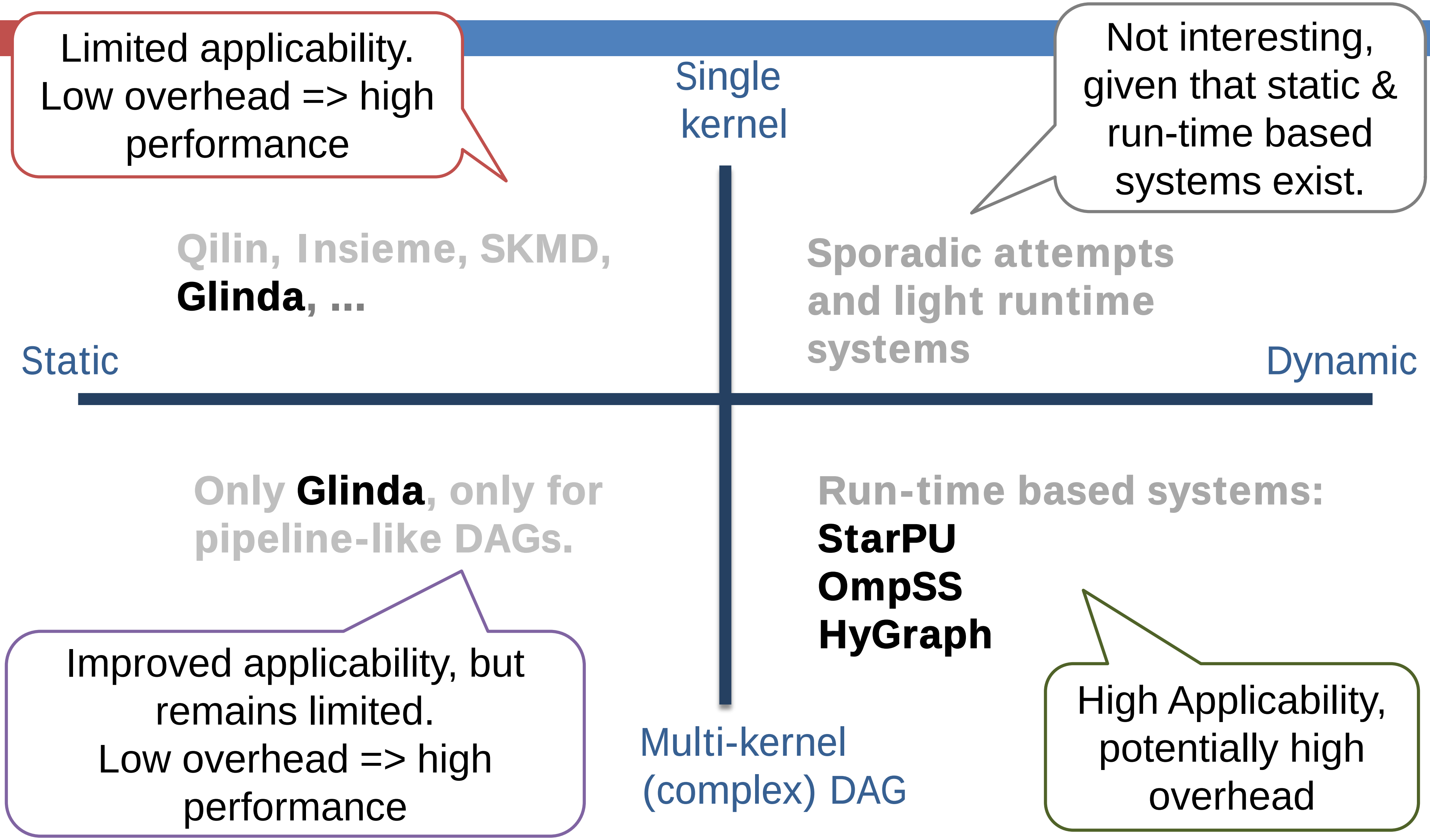
TOTEM

The most common atm. Useful for performance, more difficult to use in practice

Low level

Domain specific, focus on performance. More difficult to use.

Heterogeneous Computing today



Limited applicability.
Low overhead => high performance

Single kernel

Not interesting, given that static & run-time based systems exist.

Qilin, Insieme, SKMD, **Glinda**, ...

Sporadic attempts and light runtime systems

Static

Dynamic

Only **Glinda**, only for pipeline-like DAGs.

Run-time based systems:
StarPU
OmpSS
HyGraph

Improved applicability, but remains limited.
Low overhead => high performance

Multi-kernel (complex) DAG

High Applicability, potentially high overhead

The lab

- Three application (classes)
  - Single-kernel: matrix multiplication
  - Multi-kernel: image processing pipeline
  - Domain-specific: graph processing

- Given code templates
  - Fill in the blanks
  - Compare & improve
  - Reflect

Practicalities

My email:
A.L.Varbanescu@uva.nl

- Where ?
  - DAS5 (http://www.cs.vu.nl/das5/)
    - Ask me for *temporary* accounts
- What ?
  - Code + basic report
- When ?
  - Deadlines given earlier
- What if …
  - "I want my own project?" – OK on a case-by-case basis
  - "I don't know programming?" – it will be difficult