

# Final projects

Marco Danelutto

Dept. Computer Science

Univ. of Pisa

---



*Master Degree (Laurea Magistrale) in  
Computer Science and Networking  
Academic Year 2009-2010*





# Project HowTo

---

- **Alone or in a 2 person group**
- **Project theme is partially specified**
  - *up to the student(s) refine and complete specifications*
  - *could be made considering the on going achievements*
- **Programming language/framework choice up to the student**
  - *in one of the languages I know and I can work with:*
    - C, C++, Java
    - MPI, OpenMP, POSIX/TCP, ProActive/GCM, Muesli, SkeTo, eSkel, muskel



# Project “classes”

---

- **Different kinds of projects**

- *Implementation of RTS for structured programming (RTS)*
- *Implementation of application using a structured programming framework (APPL)*
- *Comparison of existing frameworks (COMP)*
- *Lesson on a structured programming framework (LESS)*

- **Implementation of RTS for structured programming**
  - *not a complete framework*
  - *a significant subset*
- **Activities**
  - *Design phase: fix features of the RTS, fix framework*
  - *Implementation phase: implement the RTS, provide suitable performance models*
  - *Presentation phase: demo sample code*

- **Implementation of application using a structured programming framework**
  - *application from your favorite domain*
  - *with open source code available*
  - *parallelization is your responsibility*
- **Activities**
  - *Design phase: fix application, fix the skeleton/pattern framework*
  - *Implementation phase: implement the application, provide suitable performance models*
  - *Presentation phase: demo application*

- **Comparison of existing frameworks**
  - *pick up two comparable frameworks*
  - *compare them on a specific set of applications*
- **Activities**
  - *Design phase: fix synthetic application set, fix framework set*
  - *Implementation phase: implement the application, compare results, compare models*
  - *Presentation phase: demo synthetic applications*

- **Lesson on a structured programming framework**
  - *on one of the structured programming frameworks not illustrated during the course*
- **Activities**
  - *Design phase: individuate a framework*
  - *Implementation phase: install, make experiments, evaluate scalability, compare performance models*
  - *Presentation phase: lesson with slides, including a demo of the framework usage*

# RTS assignments

---

- **RTS.1: Pipeline**

- *implementation of a run time support for pipeline computations, including stage merge/split, with performance models for multicore NOW*

- **RTS.2: Mapreduce**

- *implementation of a run time support for mapreduce computations, with performance models for multicore NOW*

- **RTS.3: Divide&Conquer**

- *implementation of a run time support for divide&conquer computations, with performance models for multicore NOW*



# APPL assignments

---

- **APPL.1: Image processing application**
  - *stream of images to be processed with a known graphic algorithm suitable for data parallel implementation*
- **APPL.2: Eight queens**
  - *find a “safe” placement of  $n$  queens on a  $n$  by  $n$  board*
- **APPL.3: sequence alignment**
  - *find sequence alignment (exact, approximated)*
- **APPL.x: AOB**
  - *propose an application, proceed after approval*



# COMP assignments

---

- **COMP.1: Muesli vs. SkeTo**
  - *requires C++ background*
- **COMP.2: Calcium vs. Muskel**
  - *requires Java background*
- **COMP.3: Calcium vs. Muesli**
  - *requires both C++ and Java background*
- **Muesli vs. TBB**
  - *requires C++ and MPI background*



# LESS assignments

---

- **LESS.1: SkeTo**
  - *C++ MPI*
- **LESS.2: eSkel**
  - *C++ MPI*
- **LESS.3: OcamIP3L**
  - *ML POSIX/TCP*
- **LESS.4: TBB**
  - *C++*

# Project target architecture

---

- **Computers at the Polo Fibonacci**
  - *new dual core PCs in Aula I*
    - AXT1 to AXT20 (possibly changing names to AXTI<sub>n</sub>)
  - *new dual core PCs in Aula H*
    - AXTH1 to AXTH20
  - *old single core in Aula M*
    - FUJIM1 to FUJIM40
- **In case of multicore targetting (to be agreed)**
  - *access to ottavinareale.di.unipi.it (dual quad core)*



# What has to be prepared

---

- **Report**

- *Complete design of the project*
  - including all the choices left unspecified and fixed by yourself
- *High level view of what's been implemented/studied/compared*
- *Relevant details*
  - implementation issues and relative solutions
  - detailed performance models
- *All the code used*
  - properly commented



# What has to be prepared (2)

---

- **“Critic” approach**
  - *illustrate alternatives and choices made*
  - *motivate results achieved*
  - *motivated differences with expected behaviour*
    - if any

# What has to be prepared (3)

---

- **Demo**

- *Sample code running*

- on your implementation (RTS)
- on the frameworks chosen (COMP)
- on the framework presented (LESS)
- implementing the application chosen (APPL)

- *With possibility to vary parameters*

- parallelism degree, size of input tasks, ...

- **Discussion / demo / result presentation**
  - *RTS, COMP, APPL*
    - about 20 minutes, accessing machines through a terminal, if needed, with possibility to have a X display (local)
  - *LESS*
    - 1 hour lesson (45mins) with slides, to be given to other persons (colleagues, researchers, ...)
- **After the presentation**
  - *1 or 2 questions related to the course material*
  - *to be answered individually*





# Project assignment

---

- **First phase**

- *by December 8*

- ordered list of preferences to the professor by email
    - Email subject: SPM final project
    - Email body:
      - name of the student(s) in the group
      - first choice project (description)
      - other choices, in order of preference
      - each choice: which class, which number, 10 lines describing whatever you think is interesting



# Underspecified project items

---

- **E.g. APPLx**
  - *by December 8*
    - choose the application
    - check with professor feasibility (during lessons intervals?)
    - prepare a short description of application, including source of source code and hints for parallelizations
      - to be included in the email



# Project assignment (2)

---

- **Second phase**
  - *by December 15*
    - I'll give you an assignment
    - most likely the one you proposed
    - unless there is no distribution at all among proposed items
      - in this case I'll go through the “second” choices



# What if first term skipped ?

---

- **Assignment is valid for the whole academic year**
  - *I'll check more and more to test you're not just re-submitting existing solutions*
- **In case you present a project and I'll not judge it "sufficient"**
  - *I can ask you more on the chosen subject*
  - *or to change assignment for the next term*