Computação Paralela

Mest. Engenharia Computacional
Mest. Int. Engenharia Computacional

Ano letivo 2022/2023

Rui Costa, Nuno Lau

Apresentação

Computação Paralela (2022/2023)

Escolaridade:

1h TP / semana 2h P / semana

Docentes (aulas TP e P):

Rui Costa
americo.costa@ua.pt
Física (13.3.32.2)

Nuno Lau
nunolau@ua.pt
IEETA (IRIS Lab / 2.07)

Página web em elearning.ua.pt

login: utilizador universal

Slides adaptados da bibliografia

Objetivos

- Identificar e classificar os diferentes modelos e os diferentes níveis da programação paralela
- Reconhecer programas suscetíveis de ser paralelizados e escolher a melhor abordagem
- Desenvolver, depurar e otimizar um programa paralelo
- Utilizar eficazmente clusters computacionais

Programa (Aulas TeoricoPráticas)

1. Computação paralela de alto desempenho
2. Modelos de programação paralela
3. OpenMP
4. CUDA
5. MPI
6. Otimização

Programa (Aulas Práticas)

As aulas práticas seguem uma filosofia do *saber fazer* e visam a realização de pequenos trabalhos sobre os diferentes assuntos.

- Paralelização com threads
- Extensões multimédia de processadores
- OpenMP
- CUDA
- MPI

Avaliação

- **Componente TeoricoPrática**
  - Teste final (durante época de exames)
  - Nota mínima: 8
- **Componente Prática**
  - 2 trabalhos
  - 50%.TP1 + 50%.TP2
  - Nota mínima: 8
- **Nota final**
  - NF = 40%.CTP + 60%.CP

Trabalhos

- Efetuar uma implementação precisa dos problemas propostos
- Os trabalhos são desenvolvidos também fora das aulas práticas
- **O plágio será fortemente penalizado**

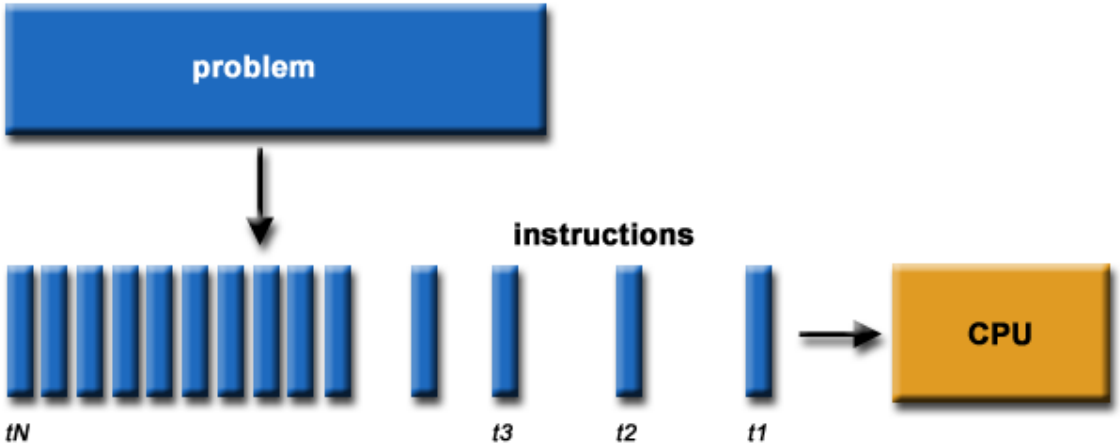
Bibliografia

- Rauber and G. Rünger, Parallel Programming: for Multicore and Cluster Systems, 2nd edition, Springer, 2013.
- Hager and G. Wellein, Introduction to High Performance Computing for Scientists and Engineers, CRC Press, 2010.
- J. Quinn, Parallel Programming: in C with MPI and OpenMP, McGraw-Hill, 2003.
- S. Pacheco, Parallel Programming with MPI, Morgan Kaufmann, 1997.
- Cheng, M. Grossman, and T. McKercher, Professional CUDA C Programming, Wrox, 2014.

Questões?

What is Parallel Computing? (1)

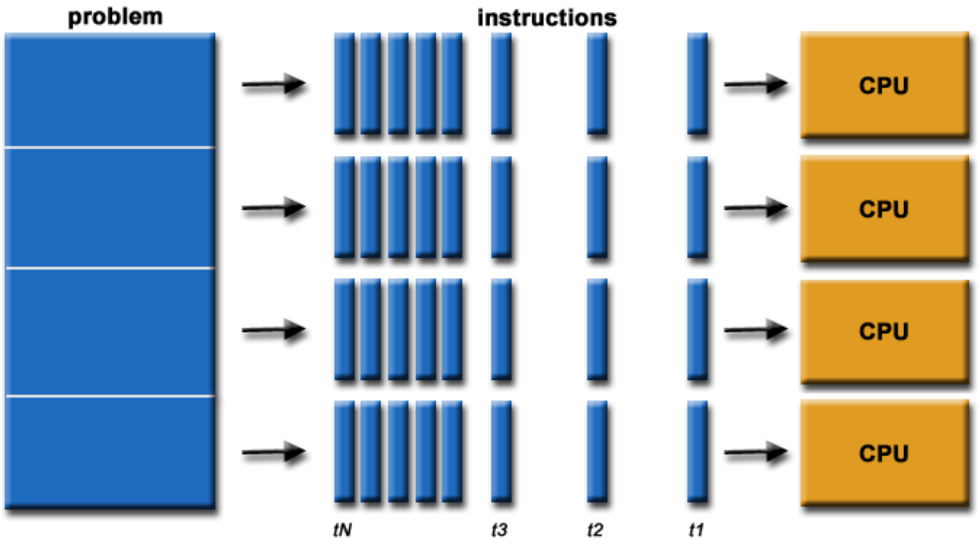
- Traditionally, software has been written for ***serial*** computation:
  - To be run on a single computer having a single Central Processing Unit (CPU);
  - A problem is broken into a discrete series of instructions.
  - Instructions are executed one after another.
  - Only one instruction may execute at any moment in time.



Acknowledgement: This lecture is based on slides by Florent Nolot, "Introduction to Parallel Computing", Université de Reims Champagne-Ardenne

What is Parallel Computing? (2)

- In the simplest sense, ***parallel computing*** is the simultaneous use of multiple compute resources to solve a computational problem.
  - To be run using multiple CPUs
  - A problem is broken into discrete parts that can be solved concurrently
  - Each part is further broken down to a series of instructions
- Instructions from each part execute simultaneously on different CPUs/Cores



Parallel Computing: Resources

- The compute resources can include:
  - A single computer with multiple processors/cores
  - A single computer with (multiple) processor(s) and some specialized computer resources (GPU, FPGA …)
  - An arbitrary number of computers connected by a network
  - A combination of the above

Parallel Computing: The computational problem

- The computational problem usually demonstrates characteristics such as the ability to be:
  - Broken apart into discrete pieces of work that can be solved simultaneously
  - Execute multiple program instructions at any moment in time
  - Solved in less time with multiple compute resources than with a single compute resource

Parallel Computing: what for? (1)

- Parallel computing is an evolution of serial computing that attempts to emulate what has always been the state of affairs in the natural world:
  - many complex, interrelated events happening at the same time, yet within a sequence.
- Some examples:
  - Planetary and galactic orbits
  - Weather and ocean patterns
  - Tectonic plate drift
  - Rush hour traffic in Paris
  - Automobile assembly line
  - Daily operations within a business
  - Building a shopping mall
  - Ordering a hamburger at the drive through.

Parallel Computing: what for? (2)

- Traditionally, parallel computing has been considered to be "the high end of computing" and has been motivated by numerical simulations of complex systems and "Grand Challenge Problems" such as:
  - Weather and climate
  - Chemical and nuclear reactions
  - Biological, human genome
  - Geological, seismic activity
  - Mechanical devices - from prosthetics to spacecraft
  - Electronic circuits
  - Manufacturing processes

Parallel Computing: what for? (3)

- Commercial applications are providing an equal or greater driving force in the development of faster computers. These applications require the processing of large amounts of data in sophisticated ways. Example applications include:
  - Parallel databases
  - Data mining
  - Machine learning
  - Oil exploration
  - Web search engines, web based business services
  - Computer-aided diagnosis in medicine
  - Management of national and multi-national corporations
  - Advanced graphics and virtual reality, particularly in the entertainment industry
  - Networked video and multi-media technologies
  - Collaborative work environments
- Ultimately, parallel computing is an attempt to maximize the infinite but seemingly scarce commodity called time

Why Parallel Computing? (1)

- This is a legitime question!
- Parallel computing is complex on any aspect!

- The **primary reasons** for using parallel computing:
  - **Save time and/or money**
    - wall clock time
    - using multiple "cheap" computing resources instead of paying for time on a supercomputer
  - **Solve larger problems**
  - **Provide concurrency** (do multiple things at the same time)