SPD 2013 –14   Course Introduction

Strumenti di programmazione per sistemi paralleli e distribuiti (SPD)

~

Programming Tools for Distributed and Parallel Systems

M. Coppola   massimo.coppola@isti.cnr.it

Course structure

- Programming Tools for Parallel and Distributed Systems (SPD)
  - 2nd term (Feb. 2014- May. 2014)
  - **6** credits
    - Note: old SPD, 9 credits, can still be taken *if you have it on your study plan*
  - 48hours : ~36 lessons, ~12 laboratory
  - Final test: lab project + oral examination
    - Includes discussing the project
  - New Course pages not yet active
  - Old Course wiki :

www.cli.di.unipi.it/doku/magistraleinformaticanetworking/spd

Overview

Description and Analysis of parallel and distributed programming platforms and models, to tackle problems of daunting size, scale and performance requirements

Parallelism at different levels of scale

- *Theoretical foundations*
- Standards for platforms and programming systems
- State-of-the-art solutions
- Practical use
- *Applications*

Course topics

- Parallel programming tools & platforms for HPC
  - HPC and also large scalable systems: Clouds
- Many different parallelism levels
  - Many-core systems
  - Multiprocessor systems
  - Clusters
  - Clouds (& Grids)

Message Passing and Shared Memory

- MPI – Message Passing Interface
  - message passing standard
  - Cluster and Cloud computing
  - linked library
  - Support for several languages
    - C, C++, Fortran + several more from 3rd parties
- TBB – Intel-Thread Building Blocks library
  - C++ template library
  - shared memory
  - multiple threads
  - aims at multi-core CPUs

High-Level Parallel Prog. Frameworks

- ASSIST and other Structured Parallel Programming approaches
  - High-Level SPP language for Clusters/Clouds
  - dynamic and autonomic management
  - BSP – based approaches

- GPGPU
  - General Purpose GPU programming
  - Exploit Many-core on-chip parallelism targeted at graphics for general purpose programs
    - High-level approaches tied to GPU implementors : CUDA and Brooks+
  - High-level approaches: OpenCL
    - APU development: soon to merge with standard programming?

Execution environments

- Ordinary multicore CPUs
- Large compact multicore CPUs (Intel Phi)
- General purpose computing enabled GPUs
- Clouds, Clusters, multi / many-core systems
  - Contrail
    - Federation of Clouds
- Specific Cloud platforms:
  - OpenNebula
    - Open Source European Platform
    - Implementation and APIs
  - OpenStack

Links to other courses

- **SPA** is a prerequisite
  - High-performance Computing Systems and Enabling Platforms
- **SPM** Distributed systems: paradigms and models
  - SPM theoretical foundations, surveys of systems
  - SPD focuses on few programming systems + lab time
- **CPA** Complements of Distributed Enabling Platforms
  - CPA focuses on Cloud/Grid platforms, related programming tools
- LPD Laboratory of Distributed Software Engineering
  - Tools and methodology for distributed software design and development
- ALP Parallel & Distributed Algorithms
  - ALP provides basics of parallel algorithmic cost models

QoS an SLA in {networking, virtualization, services**}**

- SRT Real time systems
  - Real-time theory has applications to SLA
- P2P Peer to Peer Systems
- Network Optimization Methods
  - QoS Routing and Scheduling

Some Related M. Thesis Proposals

- On SPD topics
  - Research oriented [R], implementation [I], or both [IR}

QoS control in Federated Clouds: Service Level Agreement, Quality of Protection, Hypervisor enhancement for QoS in the Contrail project	R
Algorithms and heuristics resource management in Cloud Federations in the Contrail project	R I
Software Virtual Machines integration in Cloud platforms	IR
Distributed JIT compilation for multicore CPUs	IR
Automatic Data Extraction and Analysis from distributed Medical Administrative Databases	R I

  - Other joint proposals may pop out in collaboration with P2P (Prof. Laura Ricci) and other courses related to SPD

Timetable

- 4 hours per week (standard)
  - Starting on 24/02/2014
  - Some lessons will be skipped due to work constraints in March (e.g. March 13)
- We will need to set up at least 4 extra lessons, possibly in May
- We need to get
  - non conflicting time slot for all WIN students, as well as
  - slots which do not clash with fundamental courses of the other two C.S. curricula.

Laboratory

- First exercises with laptops
- Ok for development with most of the programming tools (MPI, TBB, GPGPU, etc…)
- For testing, options are
  - Labs of the C.S. Dept. used as a cluster
  - Labs in scuola S.Anna
  - Intel PHI boards at the C.S. Dept.
  - Other devices on a case-by-case

Final test

- Coding an individual project
  - Agree topic with the teacher, write 2-page summary
  - Project will use at least one of the frameworks and tools presented
    - E.g. MPI, or TBB+MPI, or OpenCL + TBB …
  - Submit project proposal (before) and a written report (after) about the project
    - explaining problem, approach,
    - explain work done, code results, analyze test results
  - Discuss project
    - may be in seminar form with the class, if so agreed
- Plus oral test
  - About any topic in the course program

Timetable proposal

- Initial timetable (week of 24/02/2014)
  - Mon 14-16 (N1)
  - Thu 14-16 (N1)
- Confirmed new timetable (starting the week of 3/3/2014)
  - Tue 11-13 (L1)
  - Wed 16-18 (N1)
- Candidate time slot if needed
  - Fri 14-16
(either at Polo Fibonacci or CNR/S.Anna buildings, if teaching rooms unavailable)
- Question time
  - Thu 16-18 at CNR buildings, Room C33 (entrance 19)