

## *HPC Course*

Attendance, two weeks:

lectures and labs: 3.0 ECTS

Get Lab attendance sheet signed !

Project, finished Fall '10: 4.5 ECTS

Support:

Lab assistant, Project advisor, Examiner

10-08-15

HPC Course Summer '10

1

## HPC Intro Project

For some application and HPC architecture of your choice:

- Develop efficient program for non-trivial problem
- Demonstrate and report how efficient it is.

Project: 4.5 ECTS = 3 weeks of work *incl.* report writing

***Deadline for reports: Nov 1, 2010.***

10-08-15

HPC Course Summer '10

2

The project is **not** about:

- *Substantial* development of *new* code.
- *Scientific results obtained with code*

**So:**

Prioritize analysis and measurements!

Demonstrate use of tools (profiling, vampir, ...) and simple performance model.

*NO TIME* for development of new **significant** code.

**Examples:**

- \* Parallelize a code you know and/or work with; choose interesting part.
- \* Write a simple code for key algorithm of bigger solution process
- \* Write a simple code for a simple problem

10-08-15

HPC Course Summer '10

3

### **Now – during lab-afternoons**

- Discuss with instructors & course participants, form groups of size  $G$ .
- Define project and choose supervisor: Michael, Jesper,...
- Write very short synopsis, check with supervisor !
- Submit synopsis to **CyberChair** before end of HPC course
- Start the work **ASAP**:
- Finish the work; Get in touch with supervisor !!
- Submit report to **Cyberchair**.

The report will be graded and sent back with comments; you may have to complete some parts and hand in again. **We need email and paper mail address!**

- KTH and SU students: LADOK
- Other students
  - You, for registration with your advisor etc.
  - NGSSC, if relevant

10-08-15

HPC Course Summer '10

4

1. Develop initial version of program;
  2. Develop Performance model = theoretical prediction:  
time =  $f(\text{problem size } N, \text{ #processors } P, \text{ problem partitioning parameters, ...})$   
Try to assess the *communication* and *computation* times separately.
  3. *Measure* performance, e.g.  $t = f(N, P, \dots)$ , for different problem partitionings, if relevant  
x = wall clock time start to finish, or CPUtime, ...
- |               |   |   |       |     |
|---------------|---|---|-------|-----|
| Size \ # proc | 1 | 2 | 4 ... | $n$ |
| $N_1$         | x | x | x     | x   |
| $N_2$         | x | x | x     | x   |
| ...           |   |   |       |     |
| $N_M$         | x | x | x     | x   |
4. If suitable, plot “speedup” and/or “efficiency”, MFLOPS?, ...
    - Make several measurements to discover variations – discuss sources of variability. (interactive nodes, dedicated, ...)
    - Compare w. prediction;
    - Identify “bottlenecks” by profiling tools; find remedy & make changes
    - Check improvement by measurements
    - Write report with description of problem, algorithm, and design decisions, pertinent graphs of measurements and profiling, “before and after”.

10-08-15

HPC Course Summer '10

5

### Single processor performance

Algorithm:  
BLAS etc. library  
Memory hierarchy  
Disk - main - cache - register;  
Organization of loops  
data layout  
index strides  
Compute vs. save  
Compiler directives (“-O3”)

### Multi-processor performance

Algorithm: Communication !  
Latency vs. bandwidth  
# messages vs. size  
  
***Problem partitioning***  
***Load balancing***

10-08-15

HPC Course Summer '10

6

## Other

- Group size  $G$ :  $2 \leq G < 5$  recommended.
  - Requirements for grade  $\geq C$  increase with  $G$ .**
- Proposed deadline schedule (  $w$  = week number)
  - < 0922 First iteration: status report, quick feedback from advisors
  - < 1024 Second (final ?) iteration, results, quick feedback/grading
  - ----- 1101 -----
  - > 1210 ... evaluation may take a while
- **2011-01-01 Absolute deadline: evaluation turnaround time may be very long**
- **REDUCE AMBITIONS IN NEW CODE DEVELOPMENT!**
  - Reasonable idea:
    - executed (=coded), modeled, measured, interpreted, and reported
    - “Embarrassingly parallel” OK, but performance model (?)
- **Report:**
  - Background, formulas, relevant problem sizes, ...:
  - Performance model and measurements.
  - Graphs, and textual description of what the graphs show, what we learn from them
  - **Interpretation: WHY these numbers?**

10-08-15

HPC Course Summer '10

7

## Example report

Compute electric field in  $N$  gridpoints  $\mathbf{g}_j$ ,  
radiated by  $K$  photonic crystal rods located in  $\mathbf{r}_k$ ,  $k = 1, \dots, K$   
using  $M$ :th order Bessel expansion with coefficients

$$C_{km}, m = -M, \dots, M$$

*Assumption:*  $MK \ll N$

1. Compute  $\mathbf{c} = \{C_{km}\}$  to satisfy field self-consistency  
condition: solve (small) linear system

$$\mathbf{A}\mathbf{c} = \mathbf{s} \quad \mathbf{A}: MK \times MK$$

2. Compute field in the  $N$  gridpoints:

for  $j = 1:N$

$$E(\mathbf{g}_j) = \sum_{k=1}^K \left( \sum_{m=-M}^M C_{mk} B^m(|\mathbf{g}_j - \mathbf{r}_k|) \right)$$

10-08-15

HPC Course Summer '10

8

“Embarrassingly parallel” algorithm:

Assign  $N/P$  gridpoints to each of the  $P$  processors!

1. Set up and solve  $\mathbf{A}\mathbf{c} = \mathbf{s}$  on proc 1.
2. Proc 1 sends  $\mathbf{c}$  to all other processors (“broadcast”)
3. All processors compute assigned grid points
4. All processors send their assigned gridpoints to proc. 1
5. Proc 1 plots the field

Performance model:

$$T = c_1(MK)^3 + n\tau_F N/P + (P-1)(\tau_s + N/P \tau_b)$$

$$= \text{Const.} + n\tau_F N/P + P\tau_s$$

$$\text{Min. for } P = \sqrt{\frac{n\tau_F N}{\tau_S}}$$

Measure  $T$  for  
different  
 $N$  and  $P$  !