

## **OpenMP Spring Bonus** (Gives bonus points for the final Exam!)<sup>1</sup>

Please send codes + brief report + timings before 02.05.2020 to spataro@unical.it  
 Mail subject : [APSD Spring Bonus – Name Surname]”  
**(NO mails considered with different email subject!)**

1. Simple embarrassingly parallel 2D Matrix computation (es: A and B are 1000 x 1000 double value matrices):

- a.  $A(i, j) = 15*i^2 + 5*\pi *j^6$
- b.  $B(i, j) = (11/5)*A(i,j)$

2. Given three vectors A, B and C, compute (if needed adopt padding, critical sections, reduction, etc):

- $C = A + B$ .
- Scalar product between C and B.

3. Calculation of PI

- A. Standard method (numerical integration + critical section + padding)
- B. Reduction
- C. Monte Carlo

4. Find an element in a vector (Extra BONUS - WOW! be careful !). HINT: Use `cancel` clause...

5. Implement a simple version of the Game of Life (no need of graphic output). Remember slides for implementation hints (main and support matrixes, etc).

- [http://en.wikipedia.org/wiki/Conway's\\_Game\\_of\\_Life](http://en.wikipedia.org/wiki/Conway's_Game_of_Life)
- <http://www.youtube.com/watch?v=XcuBvj0pw-E>
- <http://pmav.eu/stuff/javascript-game-of-life-v3.1.1/>

\*\*\*\*\*

After implementing the versions adopting OpenMP instructions (pragmas, `parallel for`, etc), take timings and speed-up by:

- a) **varying the dimensions** of the considered data structures/point (depends on the problem). Start by choosing an appropriate problem dimension.
- b) **varying the number** of threads .
- c) changing worksharing construct scheduling (static, dynamic, etc) where applicable.

For Example (Sum Vec – Static Scheduling):

Timings (seconds)

Threads	1000000	10000000	100000000	1000000000	
1					
2					
4					

Speed-up (Ts/Tp)

Threads	1000000	10000000	100000000	1000000000	
1					
2					

---

<sup>1</sup> Write a brief one/two page report of what was done + spreadsheet file. Send everything to spataro@unical.it by May 2<sup>nd</sup>, 2020. Please use appropriate multi-core machines paying attention to the number of adopted threads! (i.e., 8 for a 4+4 hyperthread i7)

|