



24<sup>th</sup> Summer  
School on  
PARALLEL  
COMPUTING

## Debugging and Profiling: Exercises

V. Ruggiero ([v.ruggiero@ Cineca.it](mailto:v.ruggiero@ Cineca.it))  
Roma, 17 July 2015

SuperComputing Applications and Innovation Department





# Outline

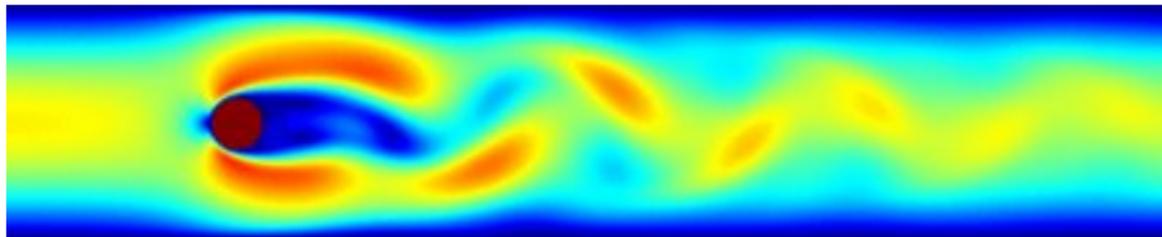
GPROF

GDB

TOTALVIEW

## Esercitazione:gprof

- ▶ Benchmarking e Profiling di un codice cinetico (Lattice Boltzmann)
  - ▶ Scaricabile gratuitamente da <http://wiki.palabos.org/numerics:codes> nelle versioni C, C++, Fortran90, Matlab, etc.
  - ▶ Flusso attraverso un cilindro





## Esercitazione

- ▶ Come scala il codice al variare delle dimensioni?

Dimensioni	250*50	500*100	1000*200

- ▶ Quali sono le quattro subroutine più impegnative?

Subroutine	Tempo	%	Num. chiamate



# Outline

GPROF

GDB

TOTALVIEW



## Esercitazione:GDB

- ▶ TEST1: semplice bug per familiarizzare con i comandi
- ▶ TEST2: altro semplice bug per familiarizzare con i comandi
- ▶ TEST3: calcolo di un elemento della successione di Fibonacci.
  - ▶ Input: un numero che rappresenta la posizione dell'elemento della successione di cui si vuole il valore
  - ▶ Output: stampa a schermo del valore dell'elemento richiesto
- ▶ TEST4: sorting
  - ▶ Input: un intero che rappresenta il numero di interi che si vogliono ordinare ed i relativi valori (in C i valori da ordinare)
  - ▶ Output: i valori in ordine crescente
- ▶ TEST5: Crivello di Eratostene (ricerca dei numeri primi)
  - ▶ Input: un numero intero che rappresenta il limite di ricerca
  - ▶ Output: L'elenco dei numeri primi inferiori e uguali a quello fornito come limite



# Outline

GPROF

GDB

TOTALVIEW



## Esercitazione: TOTALVIEW

- ▶ **hang**: il programma genera un intero di valore 10 col processo di rank 0 e lo invia agli altri processi
- ▶ **gather**: eseguire il programma con 4 processi. Il processo con rank=1 usa la funzione gather per raccogliere il risultato ricevuto dagli altri processi.

L'output corretto:

```
0 0 204 204 408 408 612 612
```

- ▶ **scatter**: Eseguire il programma con 3 processi. Un processo genera un array di 3 elementi e "scattera" uno degli elementi ad ognuno degli altri processi.

L'output corretto:

```
Scatter data rank =          1 : irecv=          2
Scatter data rank =          2 : irecv=          3
Scatter data rank =          0 : irecv=          1
```