deep se
dependable evolvable pervasive software engineering group

Exercise Session (MPI) – MonteCarlo method

Federica Filippini

Politecnico di Milano

federica.filippini@polimi.it

POLITECNICO DI MILANO

Goal

- Implement a **parallel function** to approximate definite integrals on limited intervals $\Omega \subset \mathbb{R}$.

- Prototype:

```
std::pair<double, double>
montecarlo (const std::function<double (double)>& f,
            unsigned long N);
```

- **Assumptions:**
  - N is known only on the master node
  - Ω = [-1, 1]

MonteCarlo method

- Recalls N observation from the random variable X ~ $U(\Omega)$
- Estimates the integral with

$$Q_N = |\Omega| \, \bar{Y}_N$$

where Y = f(X) and |Ω| is the measure of Ω.

- By the law of large numbers,

$$\lim_{N \to +\infty} Q_N = I = \int_\Omega f(x) \, \mathrm{d}x$$

- Variance of the quadrature formula:

$$\mathrm{Var}(Q_N) = \frac{|\Omega|^2}{N} \sigma^2$$

where σ^2 = Var(Y), which can be estimated through S_N^2.