Evaluating the Efficiency of a Computation

26.4.2020

The efficiency E of a computation on a given hardware system is defined as the quotient of the sustained performance P_s of the computation and the theoretical peak performance P_p of the hardware system:

$$E = \frac{P_s}{P_p} \quad \text{with } 0 \leq E \leq 1.$$

Sustained Performance

As discussed in class, the sustained performance P_s of a computation is given as the quotient of the work performed and the (measured) time needed for performing this work. In our context, it is common practice to characterize the work by the number of floating-point operations.

In particular, computing the LU factorization of an $n \times n$ matrix requires $\frac{2}{3}n^3 + O(n^2)$ floating-point operations. For our purposes, it suffices to consider the highest-order term.

Theoretical Peak Performance

The theoretical peak performance of a hardware system is given as the quotient of the maximum number of (floating-point) operations which can be performed simultaneously within a clock cycle and the duration of a clock cycle.

In the specific case of our hardware system harris, you find relevant information at https://harris.taa.univie.ac.at/. If you run your code on a single core of harris, at most four floating-point operations can be performed simultaneously. This leads to

$$P_p = \frac{4}{\frac{1}{2{,}2} 10^{-9}} \left[\frac{flop}{s}\right] = 8{,}8 \left[\frac{Glop}{s}\right]$$

If you run your code on several cores, the theoretical peak performance has to be multiplied by the number of cores.