SALVATORE DI GIROLAMO <DIGIROLS@INF.ETHZ.CH>

DPHPC: Balance Principles & SIMD

Recitation session
Deriving a Balance Principle

- **Concept of balance:** a computation running on some machine is efficient if the compute-time dominates the I/O time. [Kung, 1986]

- **Deriving a balance principle:**
  - Algorithmically analyze the parallelism
  - Algorithmically analyze the I/O behavior (i.e., number of memory transfers)
  - Combine these two analyses with a cost model for an abstract machine.

- **Goal:** say precisely and analytically how
  - Changes to the architecture might affect the scaling of a computation
  - Identify what classes of computation might execute efficiently on a given architecture

b) Assume a single-core system with an LRU data cache, a peak performance of \( \pi = 4 \) single precision floating point operations/cycle, and a memory bandwidth of \( \beta = 8 \) bytes/cycle.

- What is the ridge point in the roofline point of the above described system? (2pt)

- Consider the following function operating on a matrix \( A \) of \( n^2 \) floats. \( A \) is stored in row-major order. Assume that the cache size \( \gamma \) is much smaller than \( n \) (\( \gamma \ll n \)) and that a cache block has size equal to 8 floats (a float is 8 bytes). No elements of \( A \) are initially in cache (i.e., cold cache). What is the operational intensity of the following code? Is it compute or memory bound on this system? Justify your answer. (4pt)

```c
void foo(float A[n][n]){
    for (int j=0; j<n; j++){
        for (int i=1; i<n; i++){
            A[0][j] = A[0][j] + A[i][j];
        }
    }
}
```
c) Assume a program with an operational intensity of $I = \Theta(\sqrt{\gamma})$ that is balanced with respect to a given architecture (single-core). If the peak performance ($\pi$) doubles every 2 years and the memory bandwidth ($\beta$) doubles every 4 years, with which yearly rate does the cache size need to increase in order to keep the balance? (4pt)
Vectorizing the Vandermonde Matrix Determinant Computation

\[
\begin{bmatrix}
1 & 1 & \cdots & 1 \\
x_1 & x_2 & \cdots & x_N \\
x_1^2 & x_2^2 & \cdots & x_N^2 \\
x_1^{N-1} & x_2^{N-1} & \cdots & x_N^{N-1}
\end{bmatrix}
\]

\[V_n = \prod_{1 \leq i < j \leq n} (x_j - x_i)\]