

# Optymalizacja Kodu Na Różne Architektury

## Zadanie 2

### 1 Ogólne zasady zaliczenia

- Zadanie oceniane jest w skali od 0-50 punktów.
- Warunkiem koniecznym jest indywidualne oddanie zadania.
- Ocenie podlega m.in. stopień zrozumienia mechanizmów odpowiedzialnych za obserwowane zmiany w wydajności.
- Weryfikacja posiadanego modelu procesora i **dopasowanie do jego specyfikacji zoptymalizowanego kodu** (cache i jednostki wektorowe, w tym ich typ i generacja, oraz inne) jest konieczne do uzyskania pozytywnej oceny.
- Zadania umieszczone w systemie Moodle/UPEL po terminie będą bezwzględnie oceniane na maksymalnie 25 punktów.

### 2 Zadanie

Proszę o wykonanie optymalizacji algorytmu faktoryzacji Choleskiego wzoru na ćwiczeniu 2, ćwiczeniu 3 oraz poniższym instruktażu.

**<https://github.com/flame/how-to-optimize-gemm>**

Proszę o umieszczenie w systemie UPEL/Moodle sprawozdania w formie pdf. Mile widziane będą zwięzłe sprawozdania.

Można użyć jako startowego kodu do optymalizacji funkcji **chol** z drugiego laboratorium. Proszę napisać prosty algorytm weryfikujący, czy dla

zadanego rozmiaru macierzy nie zmienia się zwracane przez optymalizowaną funkcję rozwiązanie. Proszę uwzględnić rzeczywistą złożoność obliczeniową przy szacowaniu GFLOPS-ów. Mile widziane będą pomiary liczników procesora (PAPI) dające rzeczywisty odczyt mocy obliczeniowej uzyskiwanej przez procesor.

**Bardzo proszę o podkreślanie wniosków w sprawozdaniu, oraz dopasowanie rozwiązania do możliwości posiadanego modelu procesora.**