



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA W KRAKOWIE

# **Wybrane metody automatycznej generacji kodu na procesory wielordzeniowe**

**Student realizujący: Piotr Listkiewicz**  
**Opiekun pracy: dr inż. Marcin Pietroń**

## Wstęp

- **Komputery osobiste mają coraz większe ilości procesorów**
- **Większość programów wykorzystuje ten fakt w małym stopniu**
- **Niewygodne by programista musiał samodzielnie zrównoleglać przy każdej możliwości**
- **W nowoczesnych kompilatorach są zaimplementowane pewne mechanizmy automatycznej detekcji fragmentów kodu do zrównoleglenia i generacji**

## Cel pracy

- **Analiza metod automatycznej detekcji kodu który można zrównoleglić**
- **Wytworzenie narzędzia umożliwiającego automatyczną generację kodu w wersji równoległej**
- **Obecnym celem jest implementacja narzędzia dla języka C , które przekształca go do postaci OpenMP**
- **Jeżeli pierwsza faza projektu poszłaby pomyślnie, to w drugiej implementacja narzędzia przekształcającego bytectomy javy do postaci równoległej przy pomocy wątków**

## Wykrywanie potencjalnej równoległości

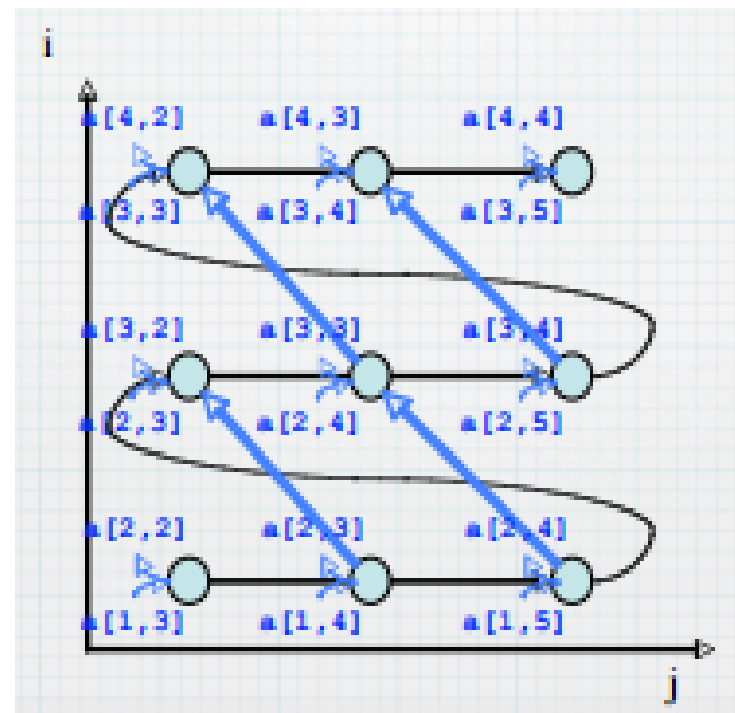
- **W pętłach występuje główna „złożoność” programów – warto skupić się na nich**
- **Ale nie każdą pętlę opłaca się zrównoleglić**
- **Potrzebujemy informacji o zależnościach występujących w pętłach by poprawnie wykryć sposób zrównoleglania**

## Testowanie zależności

- Wykrywanie zależności występujących w pętli pomiędzy iteracjami i instrukcjami
- Jest kilka możliwych rodzajów zależności pomiędzy instrukcjami
- Na ich podstawie buduje się graf zależności

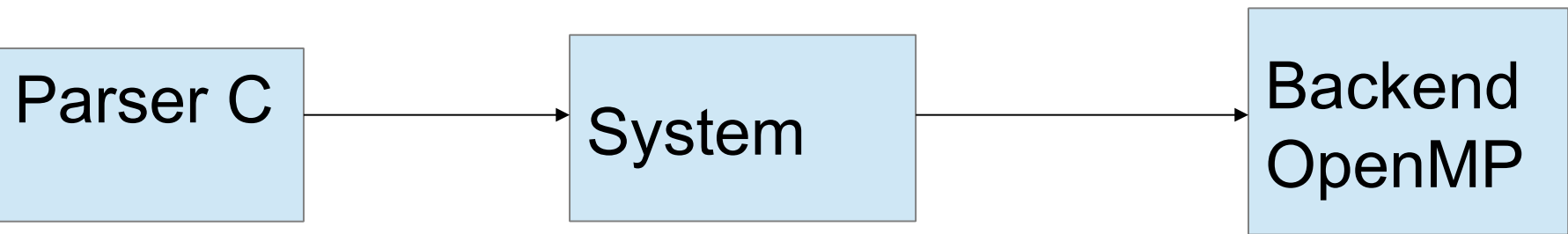
```

for i = 2 to 4
  for j = 2 to 4
1:   a[i,j] = a[i-1,j+1];
  
```



# Metody testowanie zależności

- **Range Test**
- **GCD Test**
- **Ohmega Test**



## Harmonogram prac

- **Parsing kodu w C**
- **Przerobienie drzewa rozbioru na własną reprezentację**
- **Podpięcie/Napisanie algorytmów testowania zależności**
- **Wstawienie dyrektyw w OpenMP na podstawie analizy zależności pętli**
- **Testowanie rozwiązania**



Pytania?