

MPI CA3DMM raport  
Maksymilian Grochowski 417883

UWAGA : Moje rozwiązanie używa biblioteki cblas zatem proszę o uruchamianie go z włączoną obsługą tej biblioteki na prometheusie

Moje rozwiązanie opiera się na podstawowej implementacji CA3DMM bez użycia podwójnych buforów. Używam w niej częściowych komunikatorów MPI aby wysyłać wiadomości tylko do koniecznych procesów, a także wiadomości zbiorowych z MPI typu Allgather i Sendrecv i Reduce. Pobieranie danych z generatora jest wykonywane przez wszystkie procesy używane w algorytmie równomiernie. Dodatkową optymalizacją jest użycie biblioteki cblas do wykonywania mnożenia macierzy wewnątrz algorytmu cannonna.

Ilość FLOP-ów w algorytmie zakładając  $p_n = p_m = p_k = P^{1/3}$  (co implikuje jedną grupę cannonna) i ilość flo-pów potrzebnych do wykonania zwykłego mnożenia macierzy to  $n * m * k$  to tak naprawdę w każdej  $k$  warstwie  $n/p_n * m * k / p_k * p_n$  mnożąc razy  $k$  warstw wychodzi  $O(n * m * k)$  trzeba dodać do tego koszt zsumowania wyników i zebrania danych to  $O(m * n * p_k + m * k + n * k)$

Ilość przesyłanych danych to  $O(n/p_n * m * k / p_k) / \sqrt{p_n * p_m} * (p_n * p_m)^2 * p_k$  na algorytm cannonna plus  $O(n * m * p_k)$  na zsumowanie wyników, w tym przypadku nie ma obecnie kosztu redystrybucji macierzy wynosi ona ilość cannon grup razy wielkość jednej z macierzy.

Numeryczną intensywnością rozwiązania jest iloraz pierwszej i drugiej części.