

Gajda Krzysztof 78275

Podany zostanie algorytm równoległy dla obliczania sumy n liczb. Jego idea jest następująca: dzielimy zbiór n liczb na $\lceil n/2 \rceil$ par, których sumę obliczamy równoległe i postępujemy w ten sam sposób dla uzyskanych $n/2$ liczb (gdy n jest nieparzyste jedna liczba jest bez pary).

```
k:=n;
for i:=1 to  $\lceil \log n \rceil$  do
begin for j:=1 to  $\lfloor k/2 \rfloor$  parallel do  $a_j := a_j + a_{k+1-j}$ ;
k:= $\lfloor k/2 \rfloor$ ;
end (a1 podaje wynik).
```

Łatwo zauważyć, że złożoność algorytmu wynosi $O(\log n)$ z użyciem $\lceil n/2 \rceil$ procesorów. Można wykazać, że minimalna liczba procesorów potrzebna do równoległego obliczenia sumy n liczb w koszcie $O(\log n)$ szacuje się przez $52(n/\log n)$. Nietrudno pokazać algorytm wykorzystujący $O(n/\log n)$ procesorów. Podzielmy zbiór liczb na $O(n/\log n)$ grup po $O(\log n)$ elementów i równoległe dla każdej grupy obliczmy sekwencyjnym algorytmem ich sumę w koszcie $O(\log n)$. Uzyskane w ten sposób $O(n/\log n)$ liczb zsumujemy algorytmem 3.1, uzyskując złożoność całości $O(\log n)$.