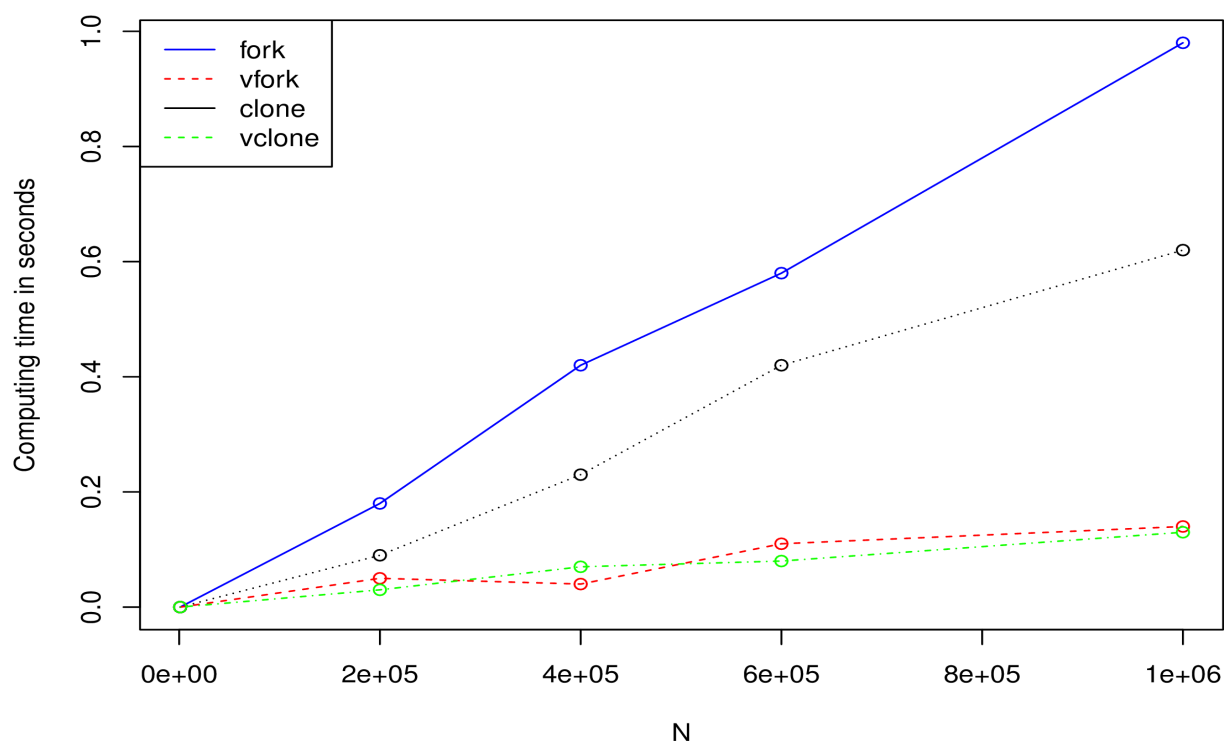
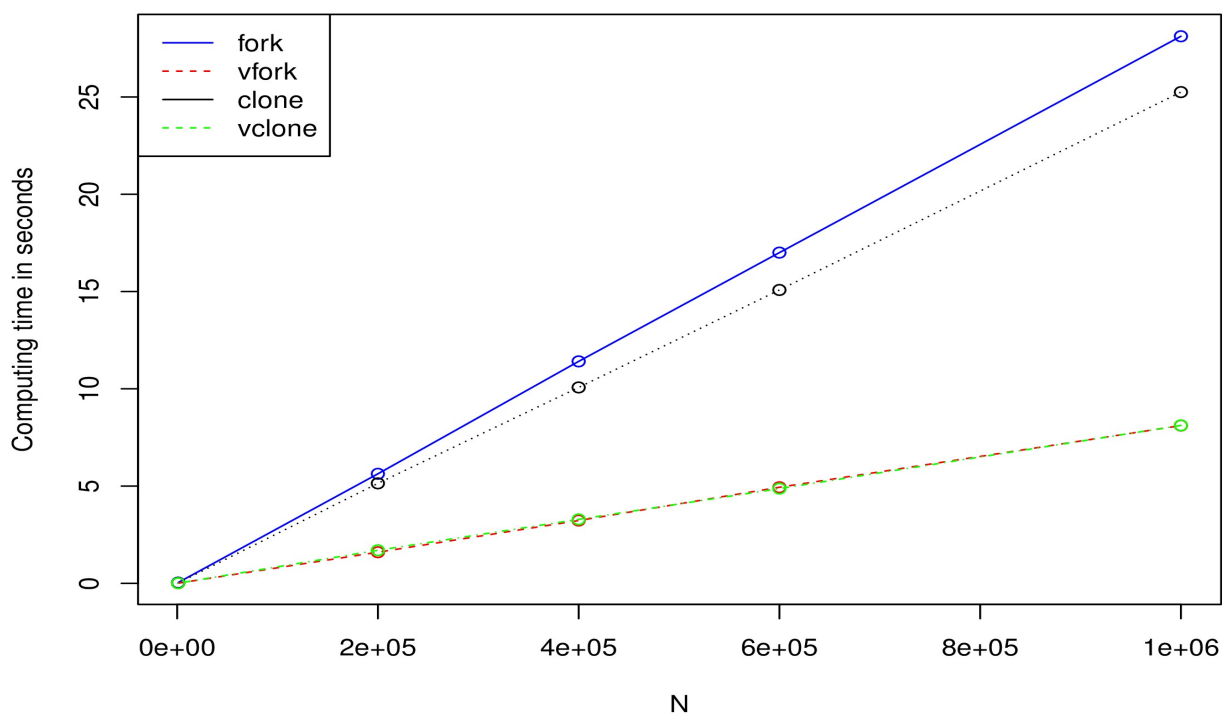


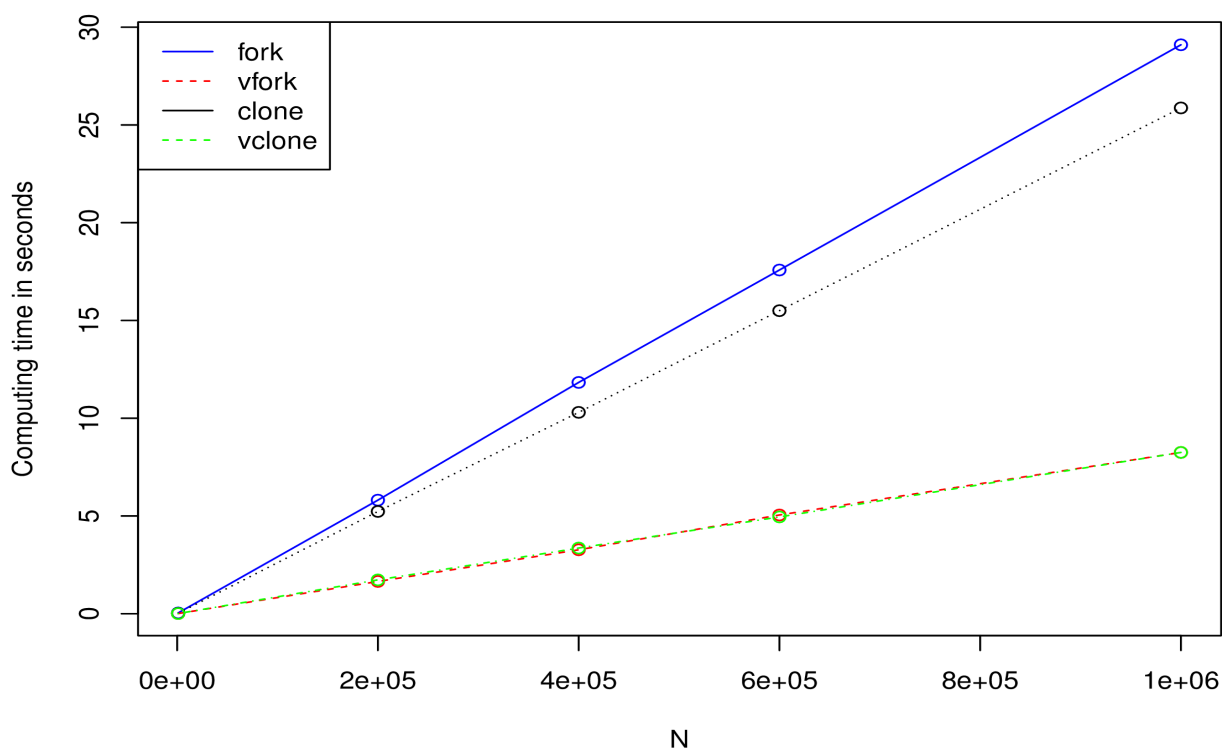
Rys 1. Czasy rzeczywiste procesu macierzystego



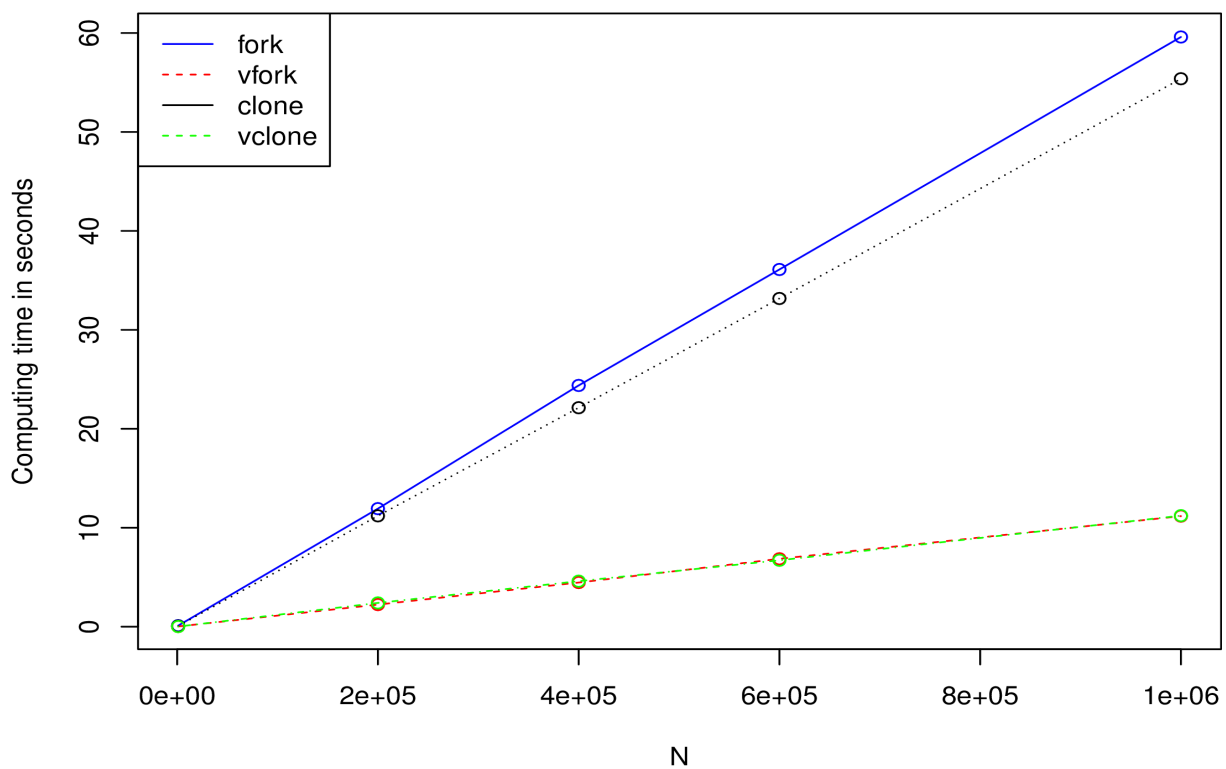
Rys 2. Czasy użytkownika procesu macierzystego



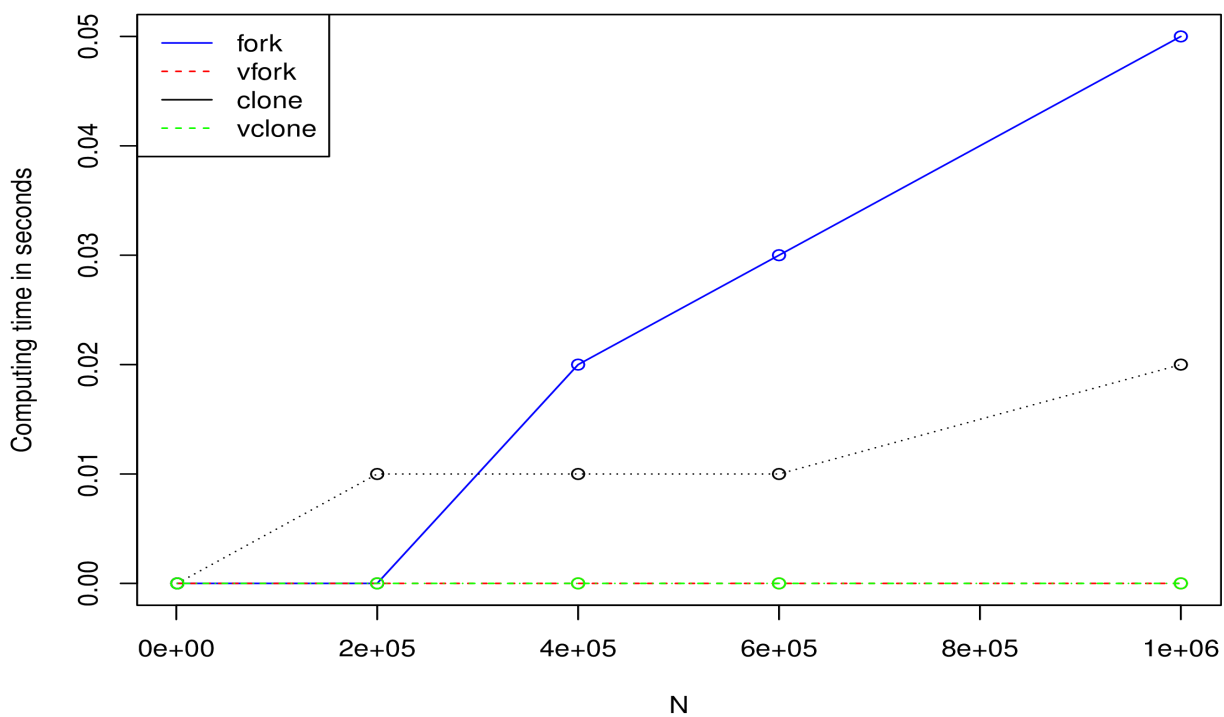
Rys 3. Czasy systemowe procesu macierzystego



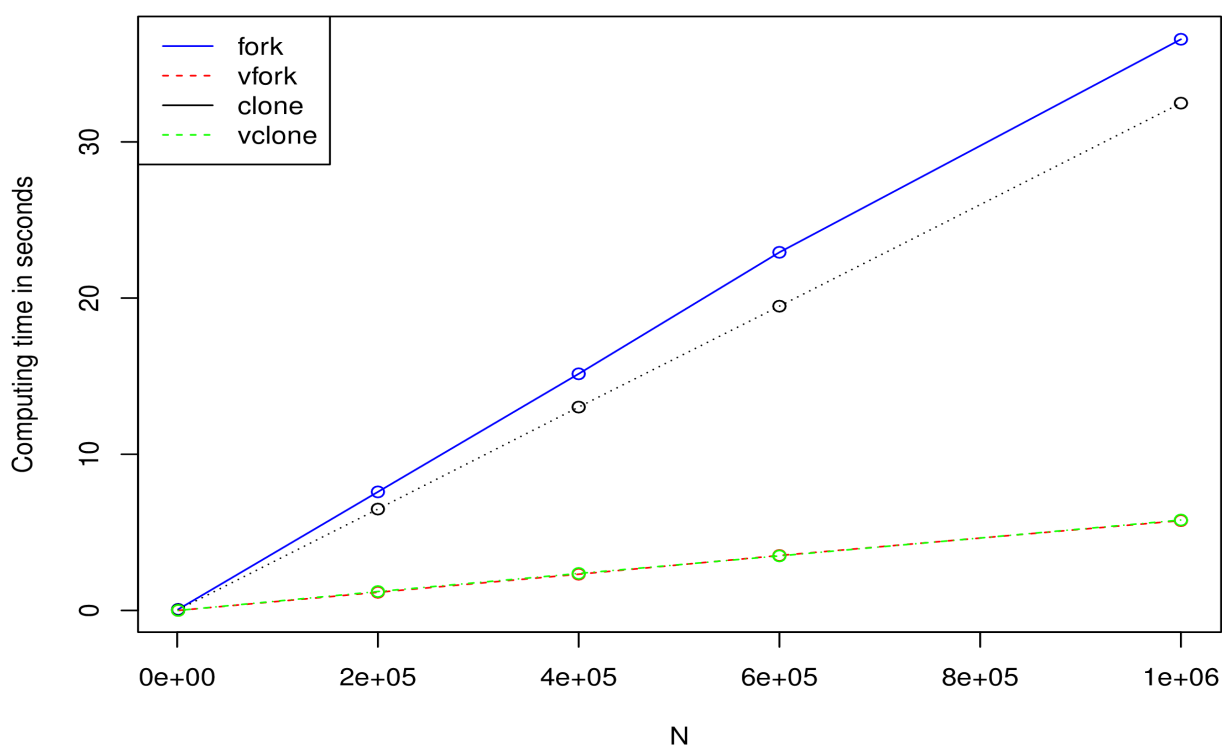
Rys 4. Suma czasu użytkownika i systemowego procesu użytkownika



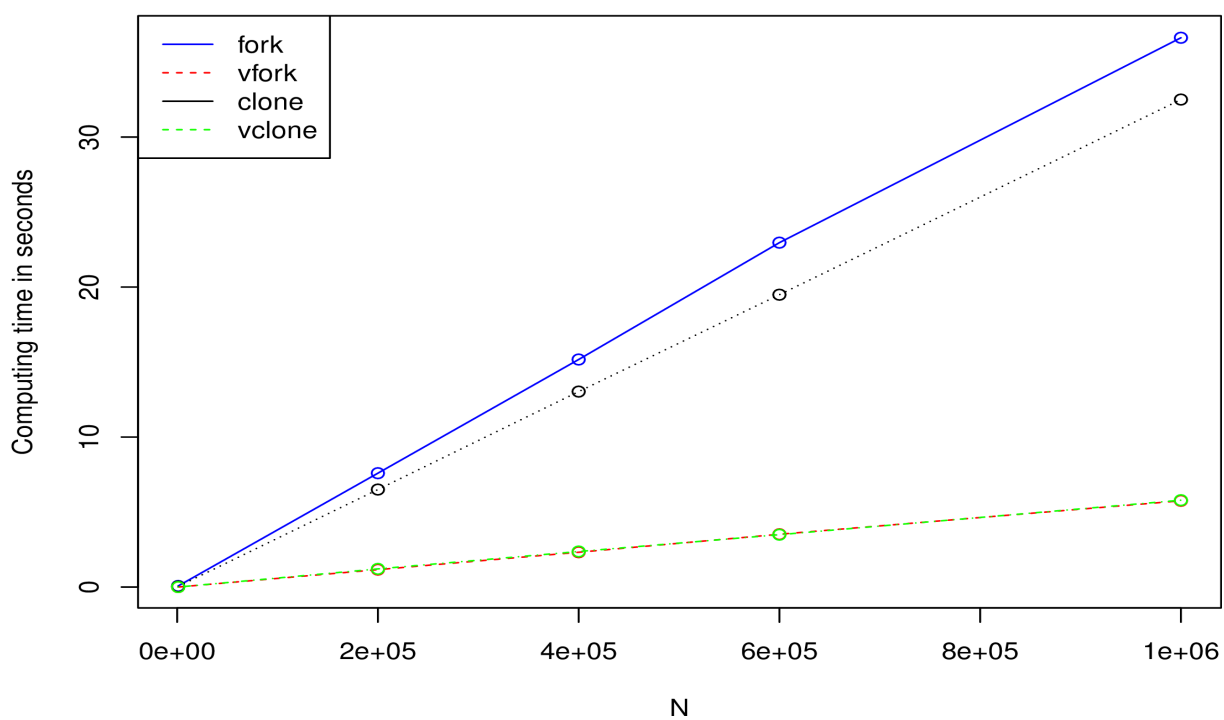
Rys 5. Czasy rzeczywiste procesów potomnych



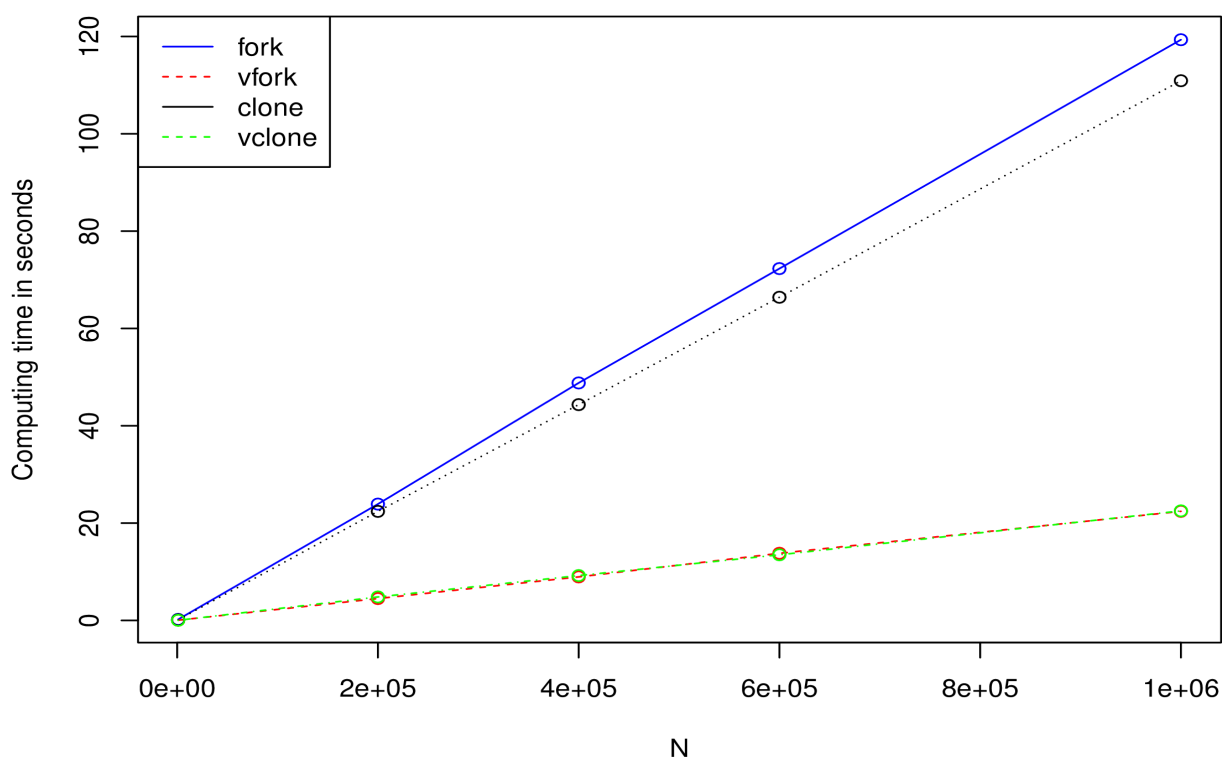
Rys 6. Czasy użytkownika procesów potomnych



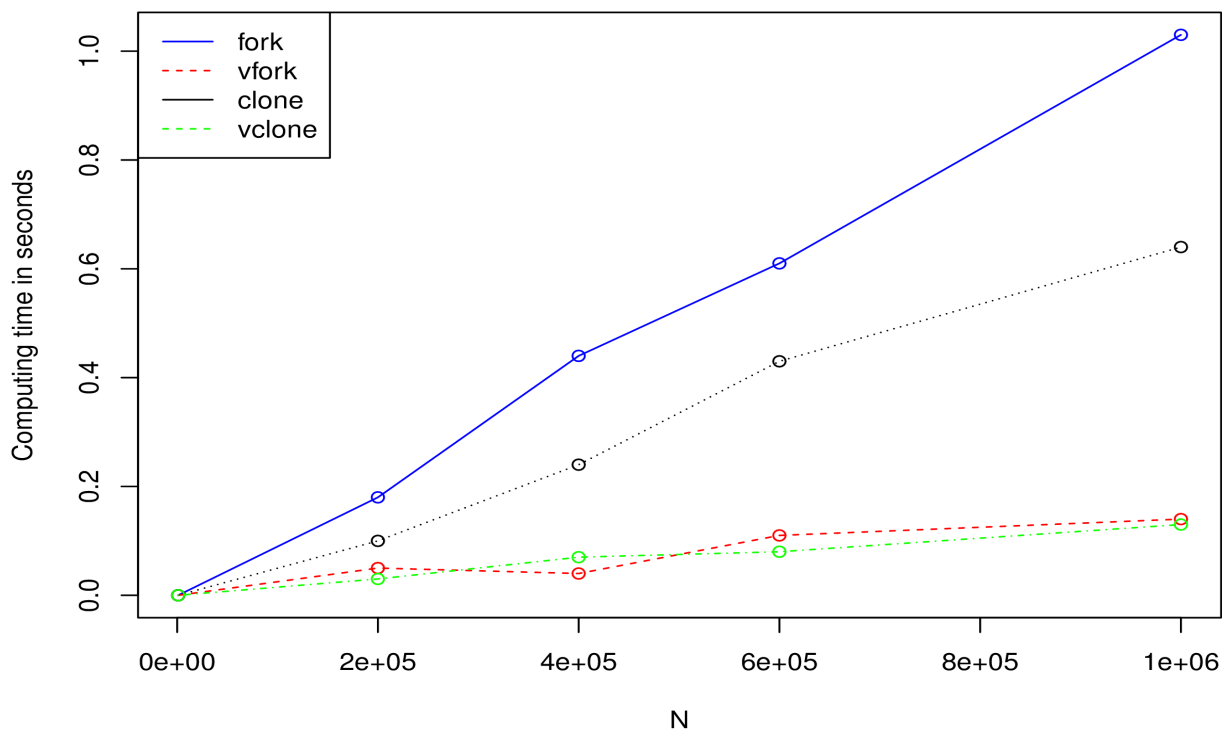
Rys 7. Czasy systemowe procesów potomnych



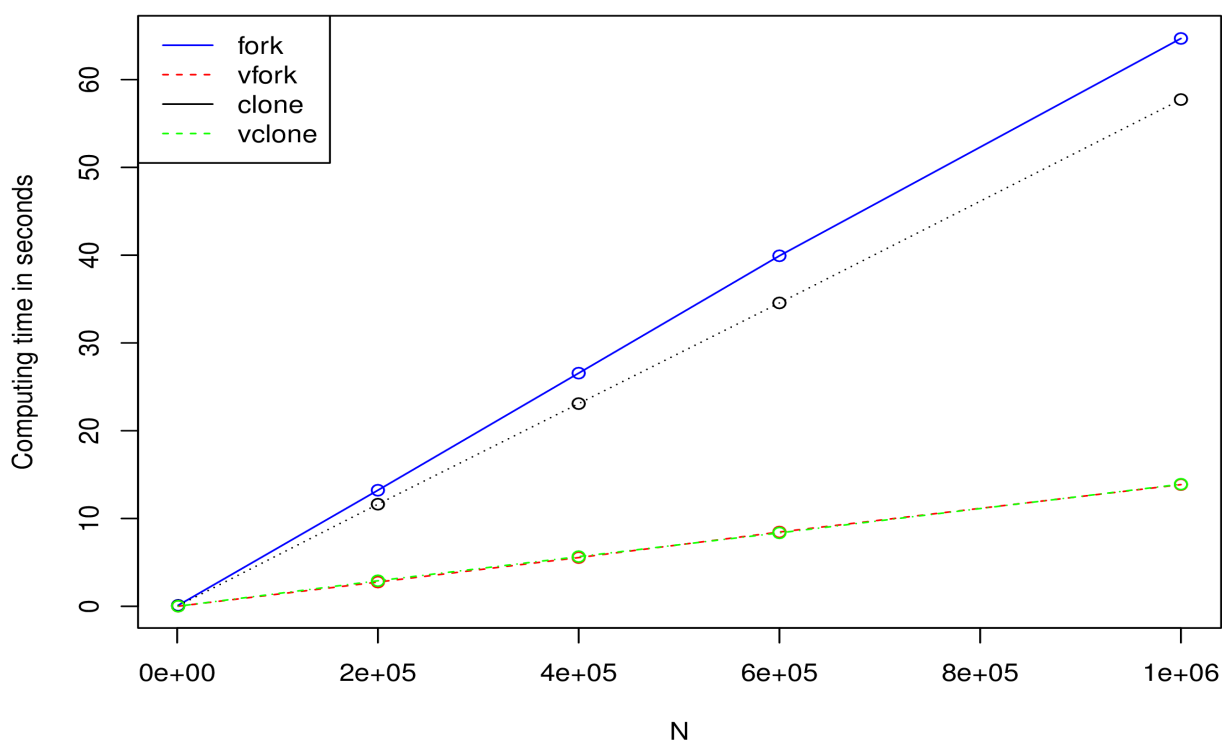
Rys 8. Suma czasów użytkownika i systemowego procesów potomnych



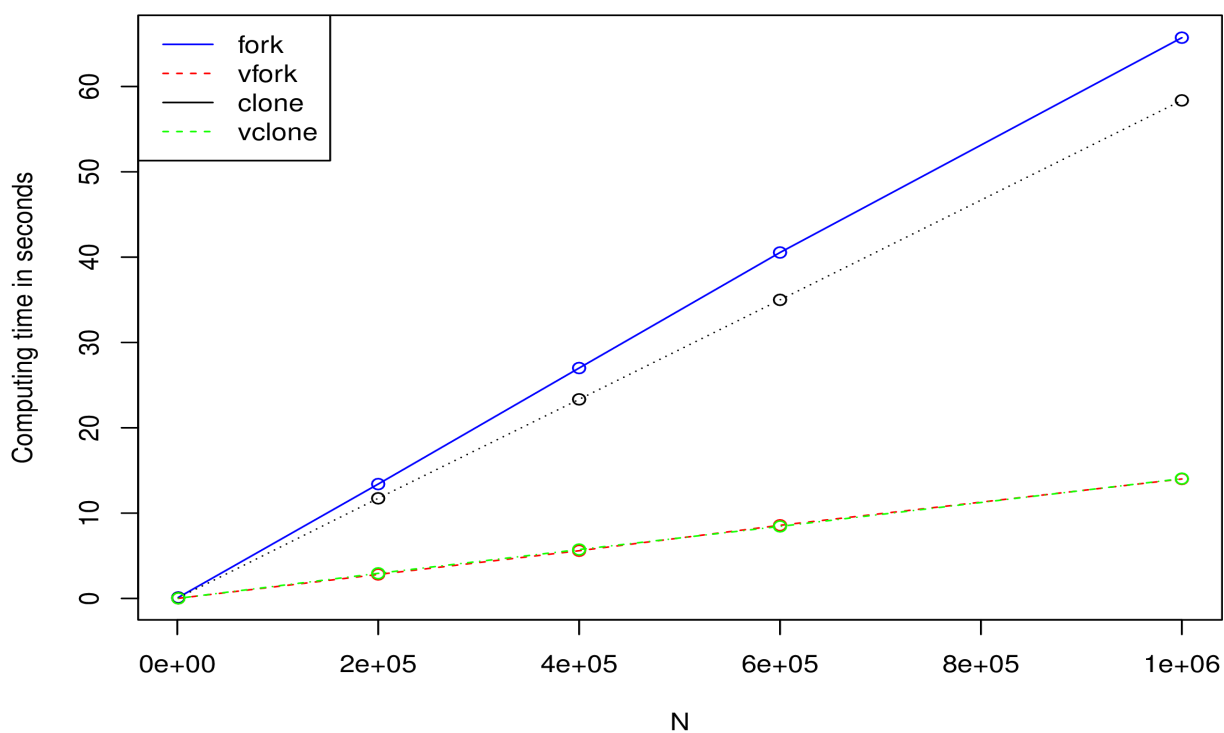
Rys 9. Suma czasów rzeczywistych procesu macierzystego i procesów potomnych



Rys 10. Suma czasów użytkownika procesu macierzystego i procesów potomnych



Rys 11. Suma czasów systemowych procesu macierzystego i procesów potomnych



Rys 12. Suma czasów użytkownika i systemowych procesu macierzystego i procesów potomnych

Komentarz:

1. Na wszystkich wykresach widać, że najwolniejszą metodą tworzenia procesu jest fork, zaraz za nią jest funkcja clone z argumentami odpowiadającymi funkcji fork. Funkcja vfork oraz clone z argumentami odpowiadającymi funkcji vfork są porównywalne wydajnościowo. Spowodowane jest to tym, że funkcja vfork nie tworzy kopii pamięci procesu.
2. Zauważamy, że tworzenie procesu dotyczy przede wszystkim pracy jądra systemu, stąd czasy systemowe są dużo większe od czasów użytkownika.
3. Czasy wykonania w procesie macierzystym są nieznacznie większe od sumy czasów procesów potomnych. Oznacza to, że program większa część działania programu polega na tworzeniu i działaniu wewnątrz procesów potomnych.