Sprawozdanie z mierzenia opóźnienia i przepustowości w klastrze

Krystian Życiński

2020

1 Wstęp

1.1 Cel projektu

Celem projektu było zapoznanie z komunikacją P2P w MPI i zmierzenie za jej pomocą opóźnienia oraz przepustowości połączeń w klastrze. Należało przetestować dwa różne typy komunikacji, dokonać odpowiednich pomiarów i storzyć na ich podstawie wykresy. Dodatkowo wyniki powinny uwzględniać następujące konfiguracje:

1. Komunikacja na 1 nodzie (pamięć współdzielona),
2. Komunikacja na 1 nodzie (bez pamięci współdzielonej, przez sieć),
3. Komunikacja między 2 nodami na tym samym hoście fizycznym (przez sieć),
4. Komunikacja między 2 nodami na różnych hostach fizycznych (przez sieć).

1.2 Sposób wykonania

Do implementacji algorytmów został wykorzystany język Python oraz biblioteka `mpi4py`. Przy mierzeniu przepustowości zostały wykorzystane pythonowe *bytearray* o odpowiednich rozmiarach. Do dokładnego zmierzenia czasu zostały wykorzystane funkcje z MPI, mianowicie `MPI_Barrier` do zapewnienia, że oba procesy są rozpoczęte, oraz `MPI.Wtime()` do zmierzenia samego czasu. Wyniki zostały zapisywane w plikach *.csv*. Programy zostały uruchomione w następujących konfiguracjach:

1. vnode-02
2. vnode-02
3. vnode-02, vnode-03
4. vnode-06, vnode-09

Przed uruchomieniem każdej konfiguracji klastry zostały sprawdzone z użyciem funkcji *htop*, aby uniknąć dokonania obliczeń na obciążonym środowisku.

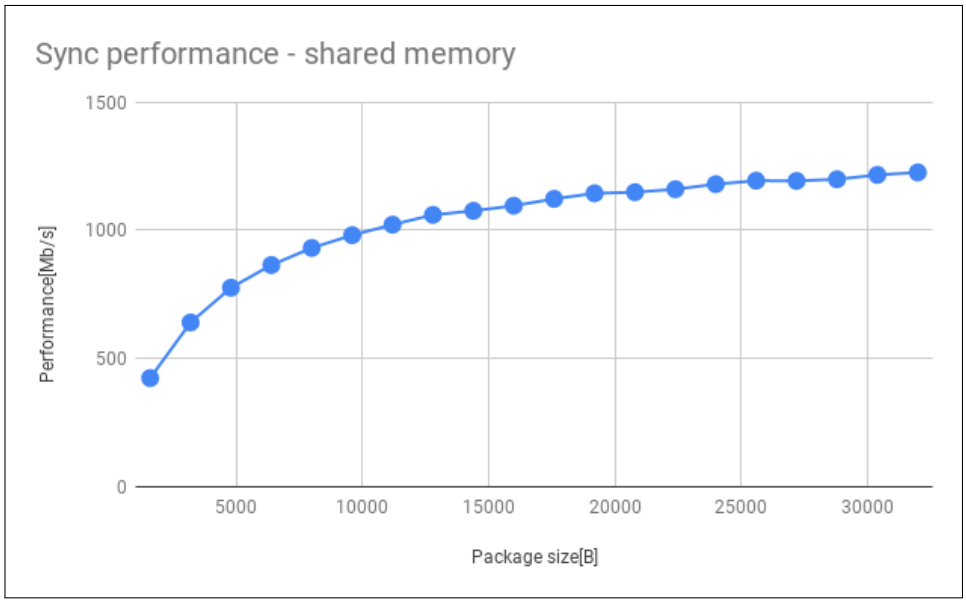
Każdy eksperyment został przeprowadzony 20 razy z różnymi wartościami parametrów. Eksperymenty polegały na wymianie 50000 wiadomości między procesami z różnym rozmiarem wysyłanych danych (rozmiary widać na wykresach).

Do wykonania eksperymentu wybrałem metodę wysyłania synchronicznego oraz metodę wysyłania bufora.

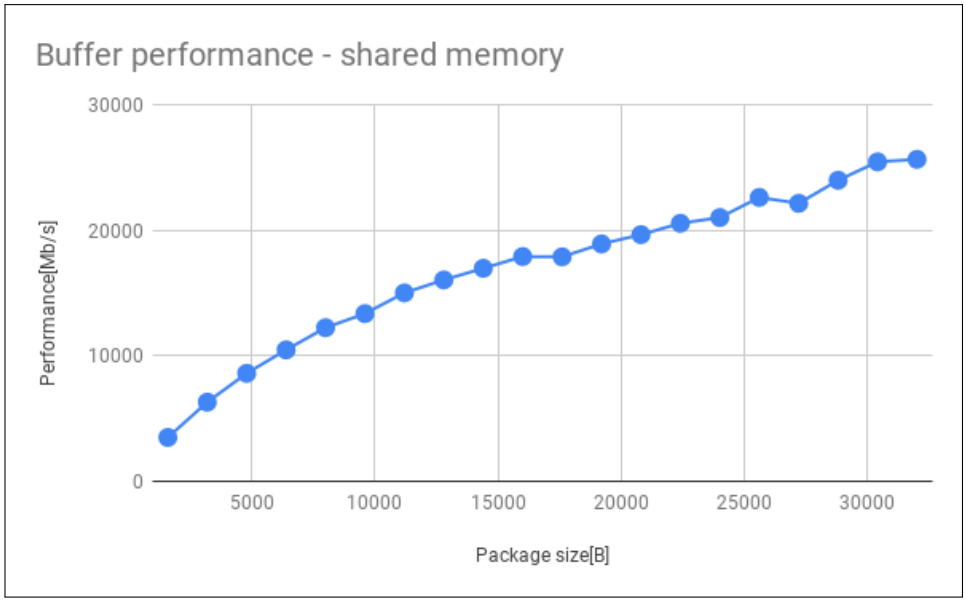
2 Wyniki obliczeń

2.1 Komunikacja na 1 nodzie—pamięć współdzielona

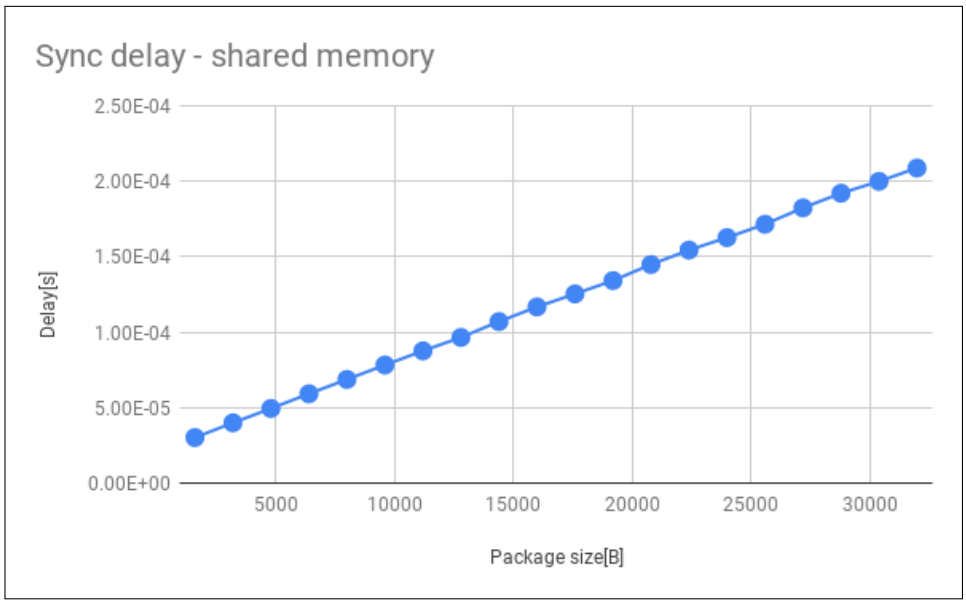
Jak można się spodziewać pamięć współdzielona daje najlepsze możliwe rezultaty:



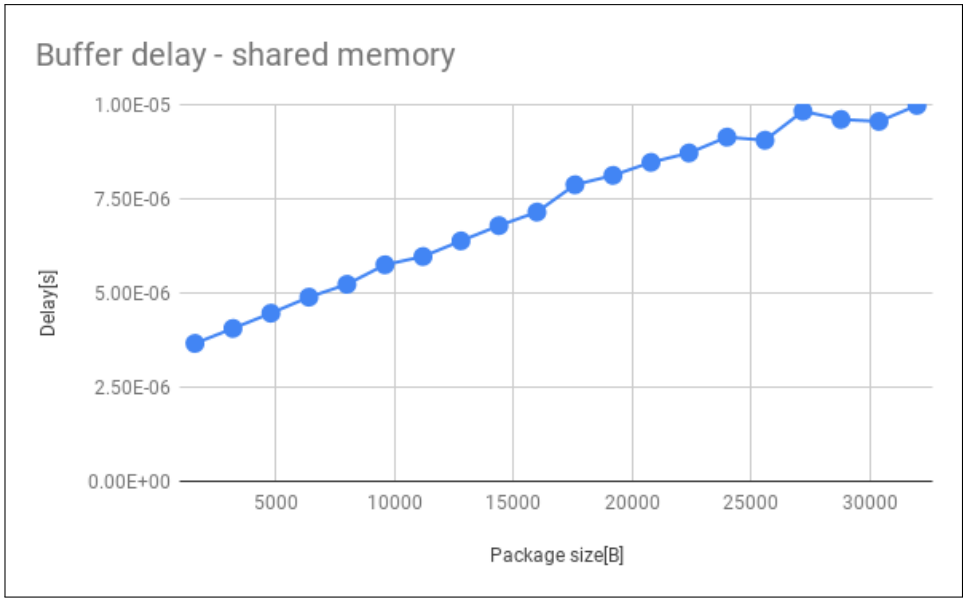
Wykres 1: Przepustowość metody sync dla pamięci współdzielonej



Wykres 2: Przepustowość metody buforowania dla pamięci współdzielonej



Wykres 3: Opóźnienie metody sync dla pamięci współdzielonej



Wykres 4: Opóźnienie metody buforowania dla pamięci współdzielonej

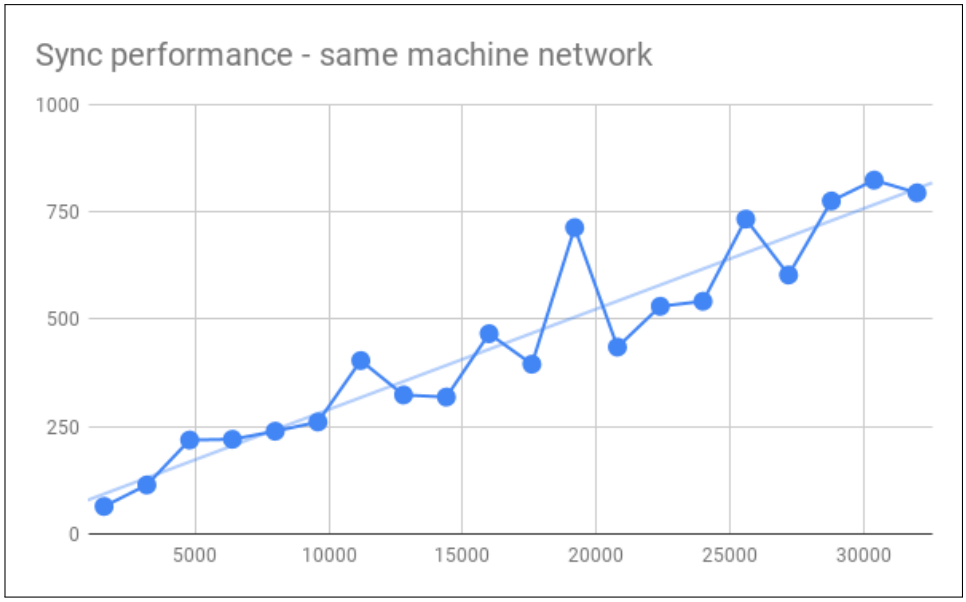
W obu przypadkach wraz ze wzrostem rozmiaru paczki wysyłanych danych można zaobserwować wzrost przepustowości oraz opóźnienia. Widać także wyższość wysyłania buforów danych—przesyłanie jest kilkadziesiąt razy szybsze przy znacznie mniejszym opóźnieniu dla pojedyńczej operacji.

2.2 Komunikacja na 1 nodzie—przez sieć)

Podczas uruchamiania testu w tej konfiguracji, można się spodziewać wyników dużo gorszych niż w przypadku pamięci współdzielonej, porównywalnych do kolejnych testów komunikacji przez sieć. Jednak otrzymywane wyniki były wręcz identyczne co przy pamięci współdzielonej, mimo wielu prób uruchomienia na różne sposoby (między innymi działająca przy implementacji w C flaga `MPIR_CVAR_CH3_NOLOCAL`). W związku z tym można wywnioskować, że `mpi4py` nie umożliwia użytkownikom uruchomienia procesów komunikujących się przez sieć kiedy znajdują się na tym samym nodzie.

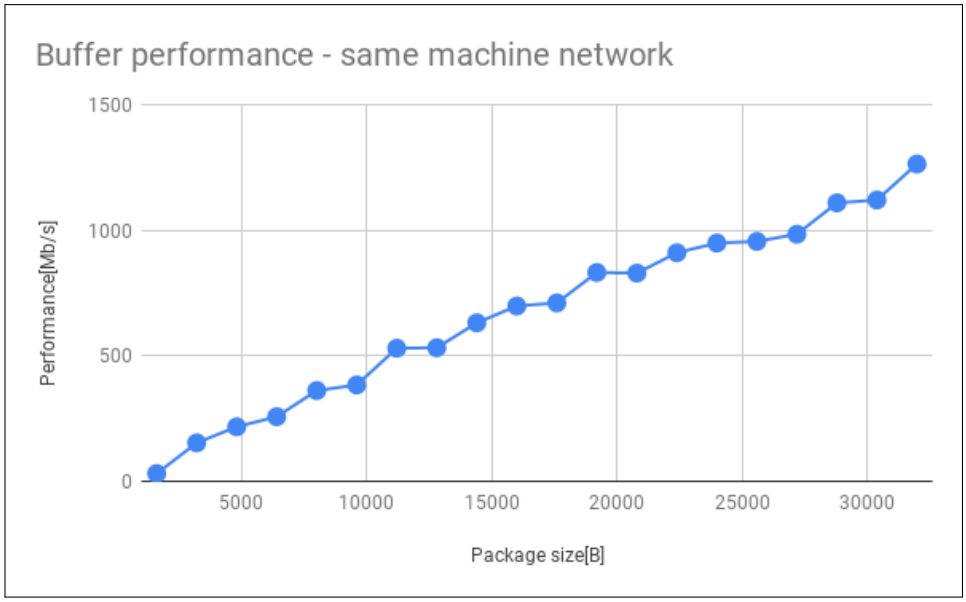
2.3 Komunikacja między 2 nodami na tym samym hoście fizycznym—przez sieć

Przy komunikacji przez sieć można zauważyć kilkukrotny spadek wydajności:

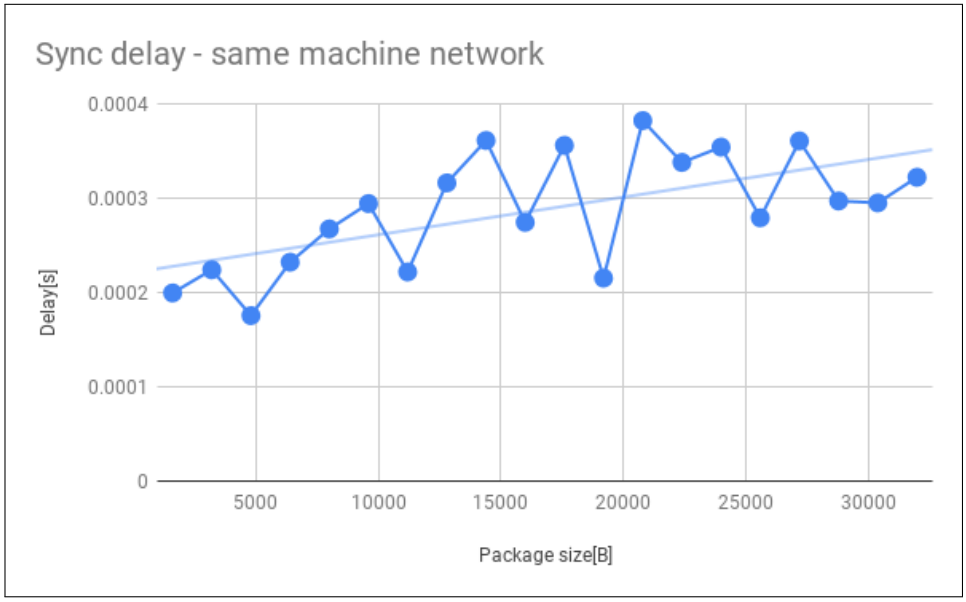


Wykres 5: Przepustowość metody sync dla tego samego hosta fizycznego

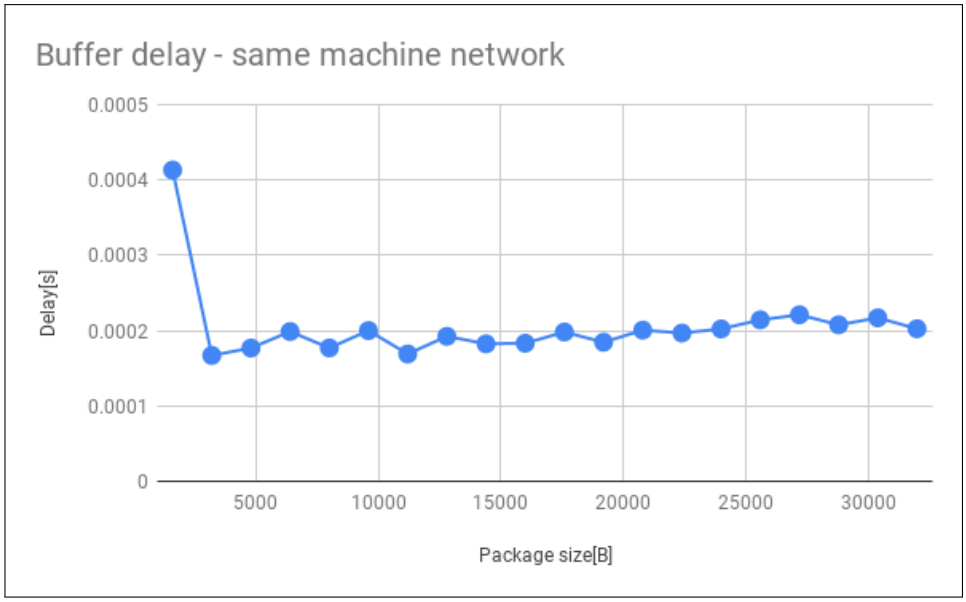
Na wykresie widać wahania, można przypuszczać że są one spowodowane narzutem na sieć AGH. Jest także zaznaczona linia trendu która jednoznacznie wskazuje na wzrost wydajności wraz ze wzrostem rozmiaru paczki.



Wykres 6: Przepustowość metody buforowania dla tego samego hosta fizycznego



Wykres 7: Opóźnienie metody sync dla tego samego hosta fizycznego



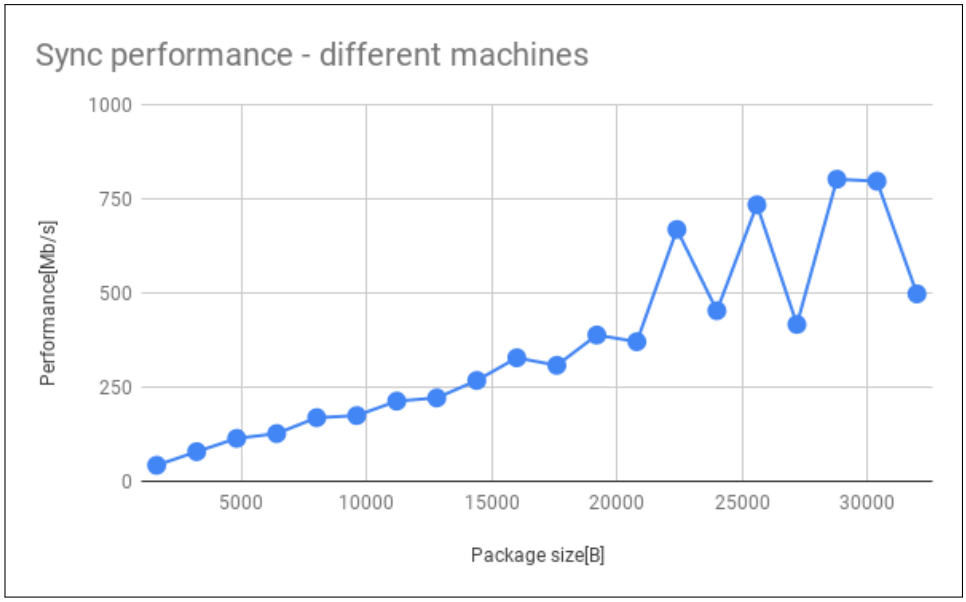
Wykres 8: Opóźnienie metody buforowania dla tego samego hosta fizycznego

Widać także zmiane w opóźnieniach—przy metodzie sync nie zmienia się ono tak gwałtownie jak w przypadku pamięci współdzielonej, a przy przetwarzaniu buforowym jest ono wręcz niezmienne (poza pierwszym pomiarem, który można uznać za błędny).

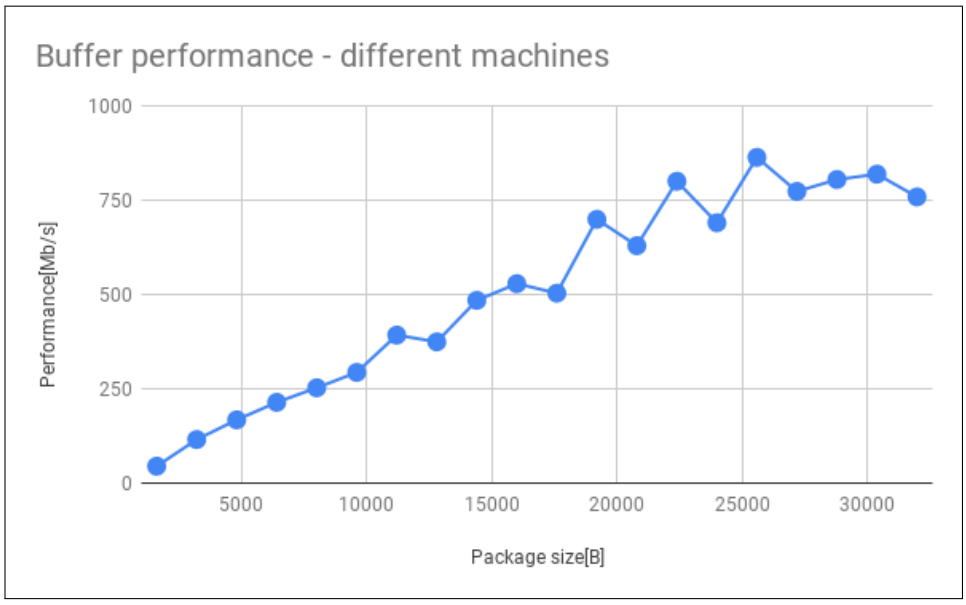
Ponadto widać także zbliżenie się metody buforowania do synchronicznej—różnica w przepustowości nie jest już tak ogromna.

2.4 Komunikacja między 2 nodami na różnych hostach fizycznych—przez sieć

W tym eksperymencie można się spodziewać jeszcze mniejszej wydajności, ze względu na dłuższą trasę, jaką muszą przebyć dane.

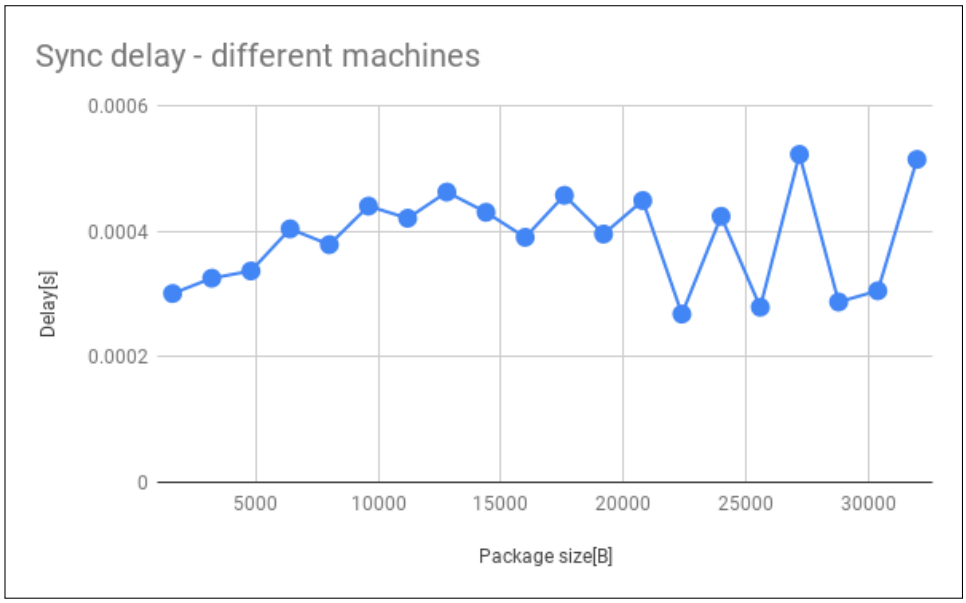


Wykres 9: Przepustowość metody sync dla różnych hostów fizycznych

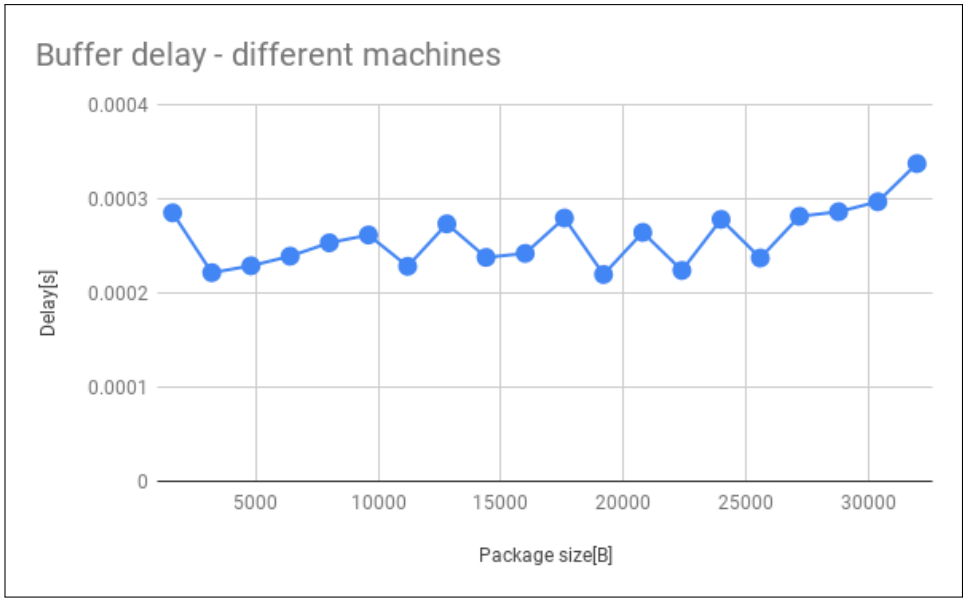


Wykres 10: Przepustowość metody buforowania dla różnych hostów fizycznych

Metoda sync ma porównywalne wyniki co na tej samej maszynie, jednak buforowanie ucierpiało jeszcze bardziej i osiąga niemal identyczne wyniki co sync. Widać także zachwianie się na wykresie metody synchronicznej— ponownie prawdopodobnym powodem jest narzut na sieć.



Wykres 11: Opóźnienie metody sync dla różnych hostów fizycznych



Wykres 12: Opóźnienie metody buforowania dla różnych hostów fizycznych

Zdecydowanie najwyższe jak dotąd opóźnienia, wyniki zgadzają się ze spodziewanymi.

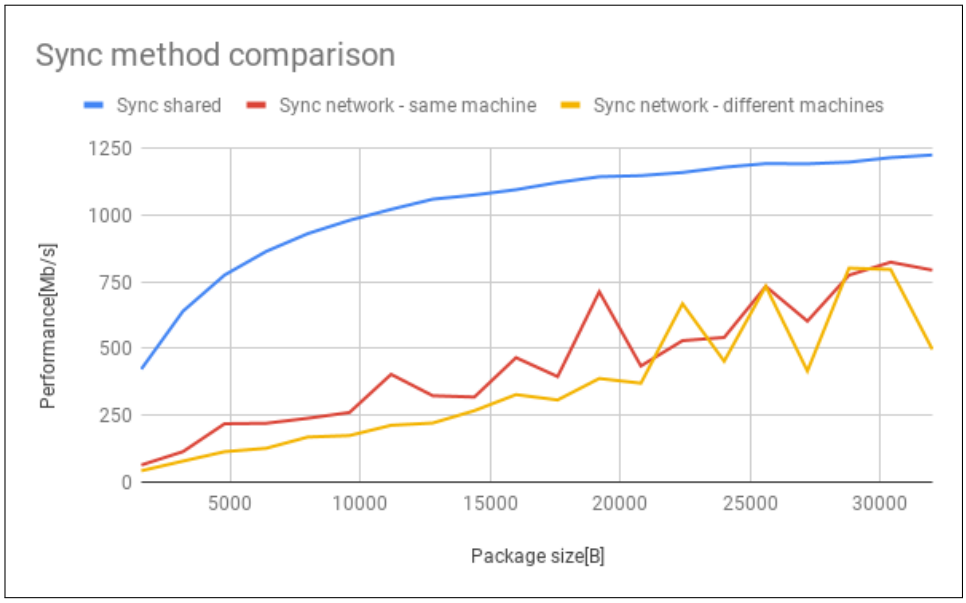
3 Wnioski

Jak widać najlepszą i najszybszą konfiguracją jest pamięć wpółdzielona. Na drugim miejscu prawdopodobnie znalazłaby się metoda komunikacji na jednym nodzie przez sieć, jednak doświadczenia nie udało się przeprowadzić.

Najwolniejszą metodą były 2 różne hosty fizyczne—pakiety danych musiały przebyć najdłuższą drogę, na współdzielonych odcinkach.

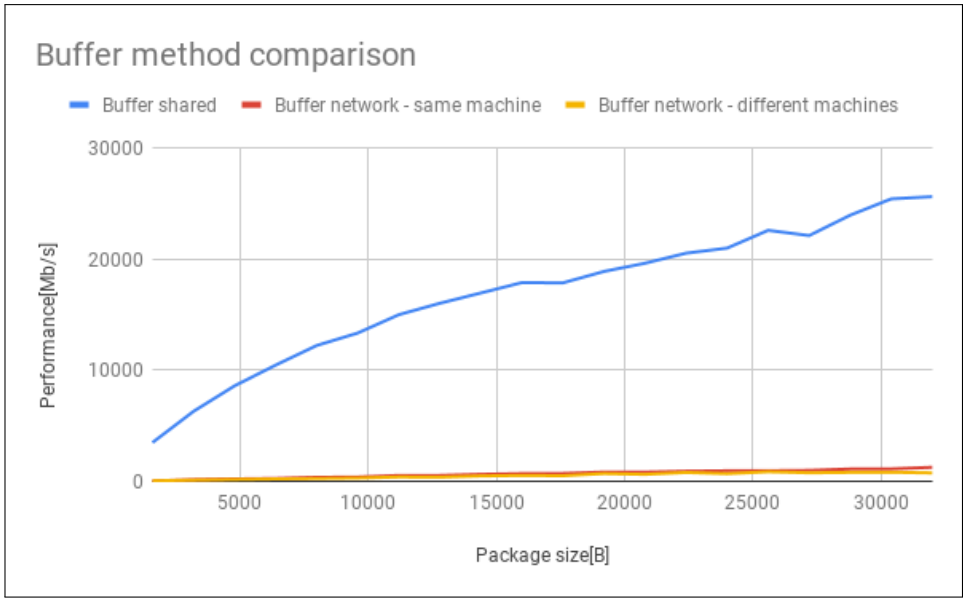
3.1 Porównania konfiguracji

3.1.1 Przepustowość



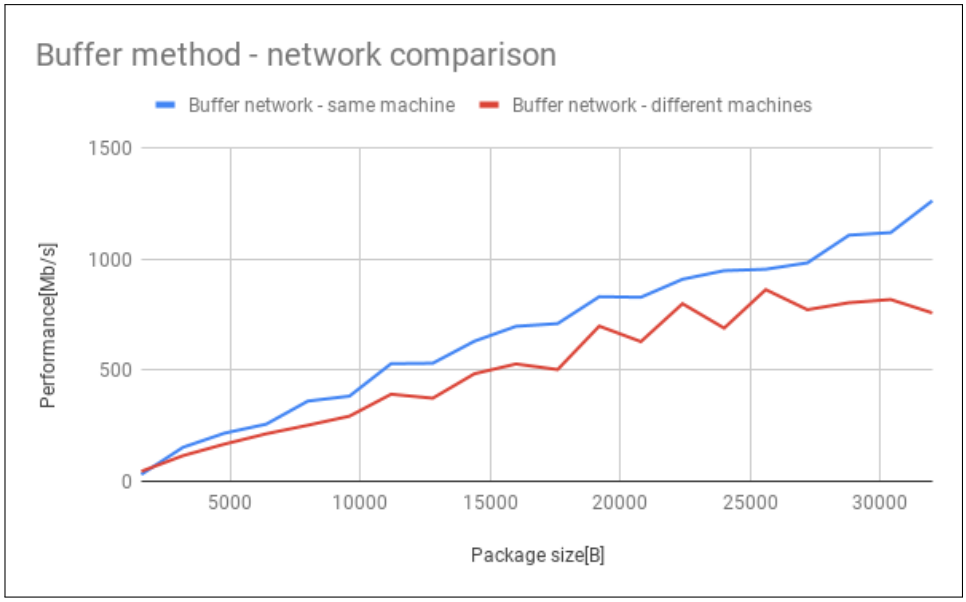
Wykres 13: Porównanie przepustowości dla metody synchronicznej

Pamięć współdzielona jest kilka razy wydajniejsza niż porozstałe, nie ma także wahań na końcu, bo duże pakiety danych nie musiały na siebie czekać.



Wykres 14: Porównanie przepustowości dla metody bufforowania

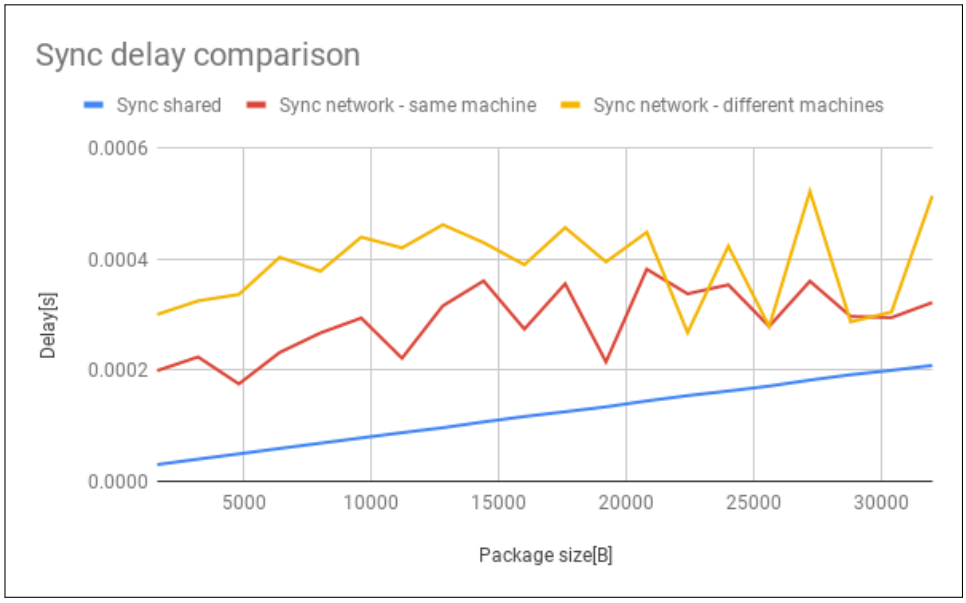
Jak widać w tej metodzie pamięć współdzielona nie pozostawia suchej nitki na konkurencji. Warto jednak przyjrzeć się samej komunikacji poprzez sieć:



Wykres 15: Porównanie przepustowości dla metody bufforowania—konfiguracje sieciowe

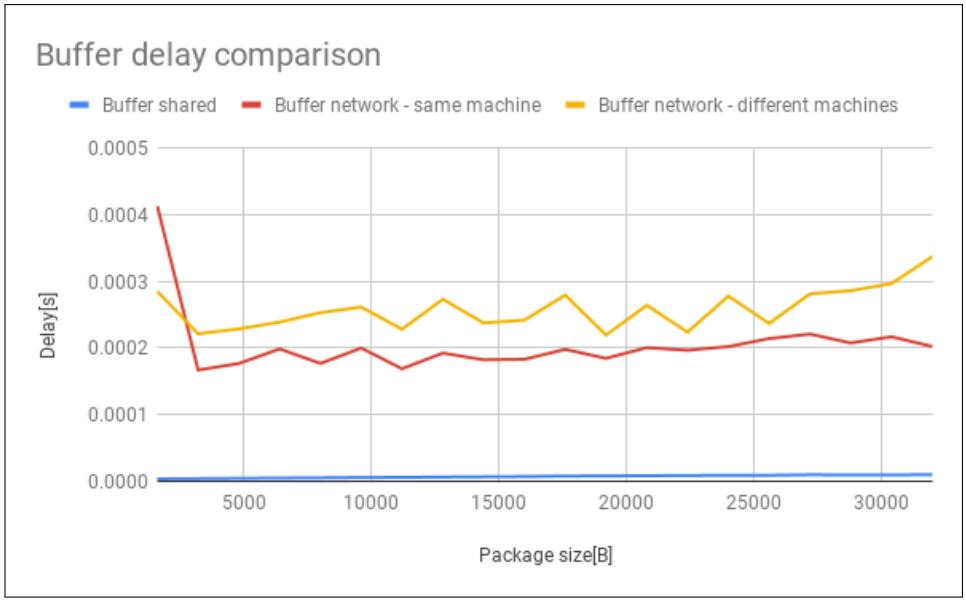
Sieć na tej samej maszynie pod koniec doświadczenia zaczęła się oddalać od konkurenta, można założyć że w miare zwiększania paczki ta różnica powiększałaby się jeszcze bardziej.

3.1.2 Opóźnienie



Wykres 16: Porównanie opóźnienia dla metody synchronizacji

Pamięć współdzielona (jak można się było spodziewać) ma najmniejsze opóźnienia. Warto także zwrócić uwagę na zaburzenia związane z narzutem na sieć, szczególnie widoczne przy dużych paczkach.



Wykres 17: Porównanie opóźnienia dla metody bufforowania

Przy metodzie buforowania widać mniejsze wahania, lecz dużo większą różnicę w opóźnieniu przy przesyłaniu przez siec i przy pamięci współdzielonej.