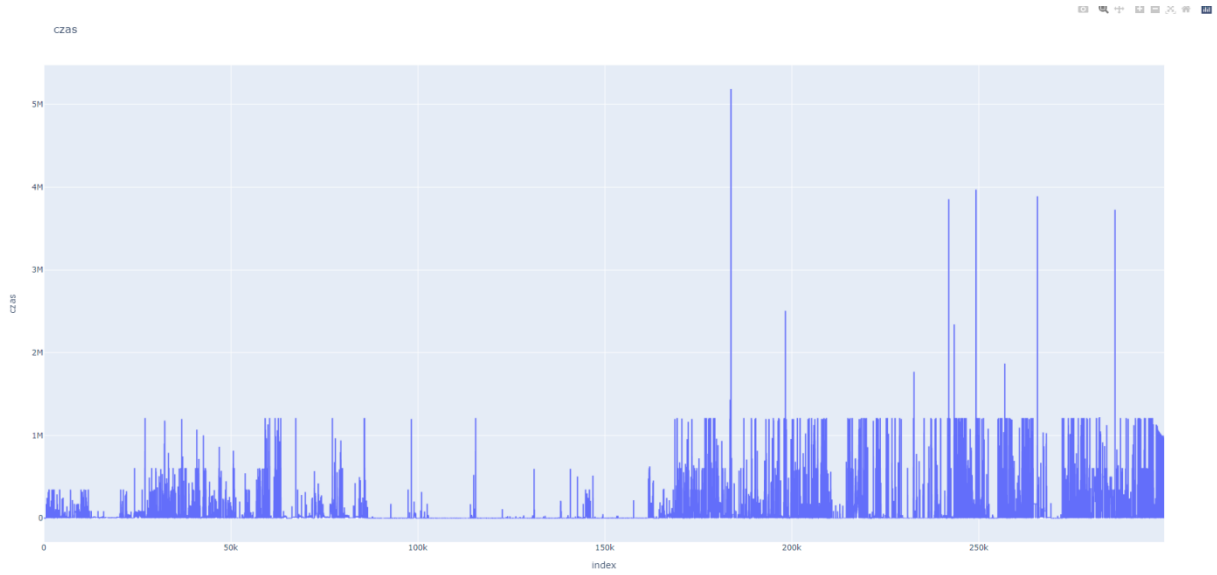
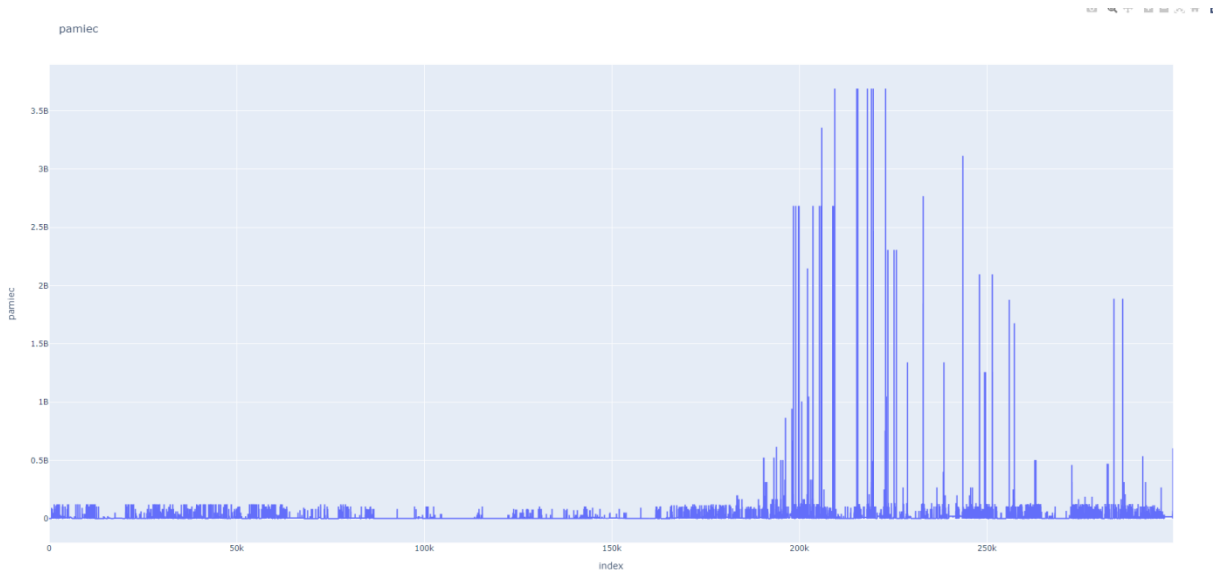
Michał Wawrzyniak

1. Na wykresie przedstawiającym rozkład czasów widzimy, że w danym okresie w omawianym klastrze były wykonywane zadania, które trwały bardzo krótko a także takie, których wykonanie zajmowało nawet ponad miesiąc. Nie da się nie zauważyć, że duża część zadań wykonuje się jednak bardzo zbliżoną ilość czasu (w okolicy: 1.2M sekund)



Jeśli chodzi o pamięć dane zadania w większości nie wymagają jej aż tak dużo choć zdażają się takie które potrzebują ponad 3.5B KB (34 GB) to większość z nich nie przekracza 9 GB.



2.

Strategia szeregowania, którą umieściłem jako ALFA to algorytm SPT (Shortest Processing Time). Algorytm ten nie różni się zbytnio od argumentu, który musieliśmy zaimplementować wcześniej.

Jedna różnica polega na tym, że w pierwszej kolejności przypisujemy algorytmy, których czas wykonywania jest najkrótszy.

Natomiast strategia BETA jest to algorytm, który przypisuje zadania w kolejności stosunku czasu wykonywania do pamięci zaczynając od najmniejszego. (czas / pamięć)

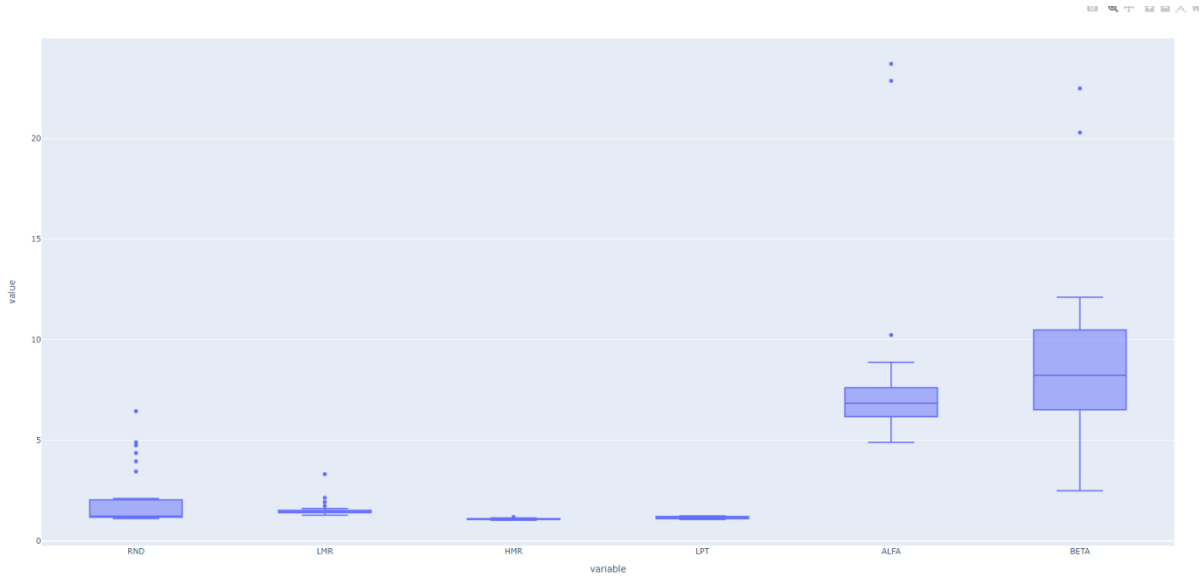
3.

$$\max\{\sum_i p_i/16, \sum_i p_i \cdot m_i\}.$$

Z oczywistych względów maksymalny czas zakończenia wszystkich zadań nie może być mniejszy niż łączny czas wykonywania wszystkich zadań podzielony przez ilość procesów (w tym przypadku rdzeni).Wynika to z tego że przydzielamy zadania tylko do wolnych procesorów a na każdym z nich wykonywać się może tylko jedno zadanie.

Problem zaczyna się z udowodnieniem drugiej części tego wzoru. W związku z tym, że jesteśmy ograniczeni ilością pamięci RAM, istnieją momenty w których musimy czekać aż zwolni się pamięć by móc przypisać następne zadanie. Jeśli dane zadanie wymaga dużo pamięci istnieje duże prawdopodobieństwo, że nie będziemy w stanie wykonać tego zadania od razu. Oznacza to przestój w naszym szeregowaniu co wydłuża czas zakończenia wszystkich zadań.

4.



Według mojego wykresu najlepszym algorytmem w tej sytuacji jest HMR myślę, że to dzięki temu że najpierw przydzielam najbardziej wymagające pod względem pamięci zadania co niweluje przestoje w późniejszej części algorytmu. Jeśli chodzi o RND jako, iż zadania są przydzielane losowo myślę, że przy odrobienie szczęścia jeśli trafi się najlepsze zadania może to być najlepszy algorytm, który może wykonać się nawet przez przestojów ale może też być najgorszy wszystko zależy od wylosowanych zadań. Wydaje mi się, że LMR jest jednym z gorszych algorytmów dlatego, że wszystkie zadania wymagające najwięcej pamięci zostają na sam koniec. Przez to pod koniec nie można „zapchać" innych rdzeni innymi zadaniami tylko przy każdym trzeba czekać koniec poprzedniego.

Jeśli chodzi o mój algorytm ALFA wydaję mi się, że coś z nim jest nie tak jako, że wszystkie inne algorytmy mimo wszystko są bardzo zbliżone do siebie podczas gdy mój SPT jest dużo dużo gorszy.