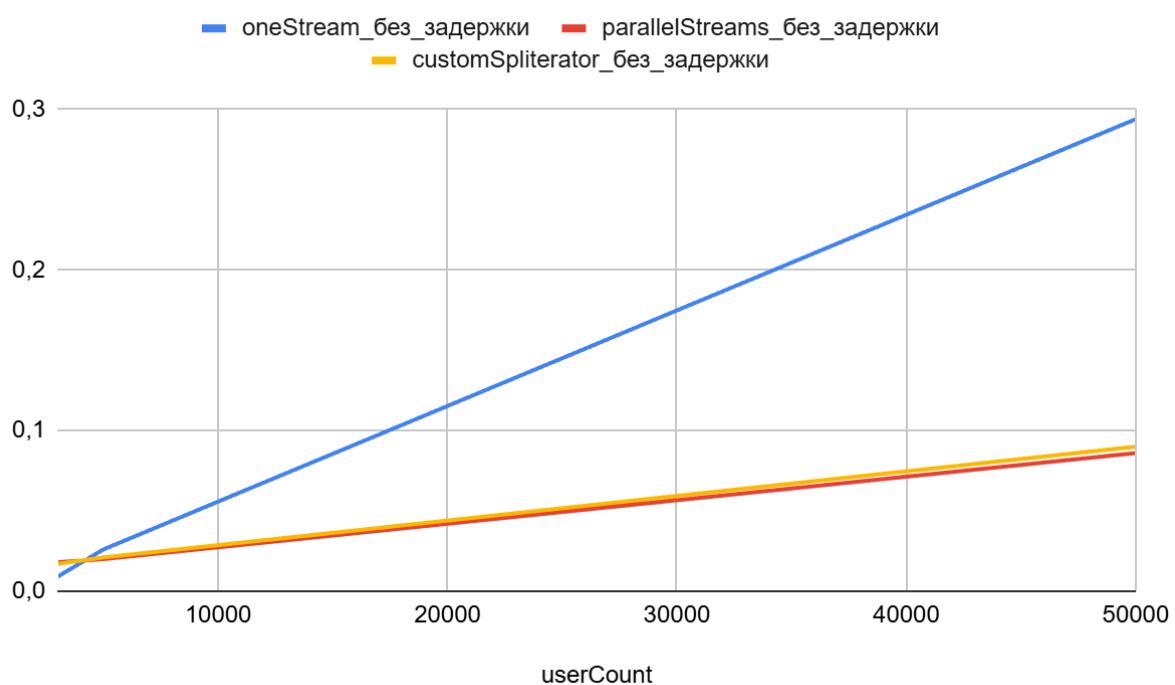
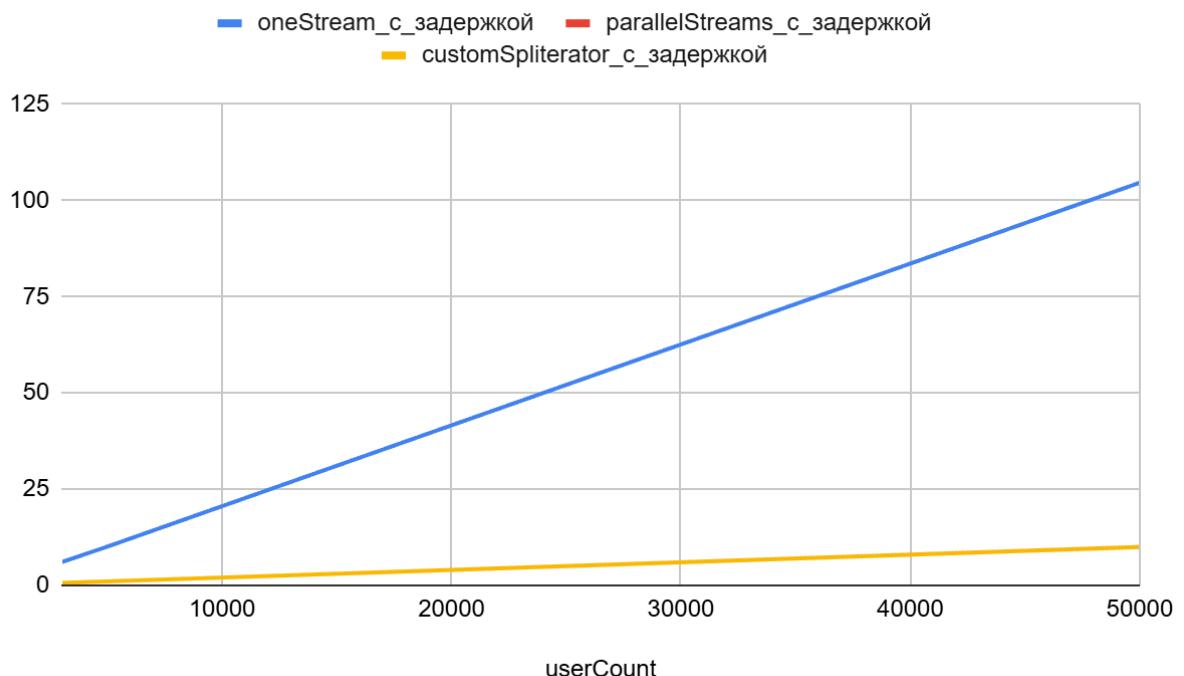


Active users Count без задержки			
userCount	oneStream_без_задержки	parallelStreams_без_задержки	customSpliterator_без_задержки
3000	0,009	0,018	0,017
5000	0,026	0,02	0,021
50000	0,294	0,086	0,09



Active users Count с задержкой			
userCount	oneStream_c_задержкой	parallelStreams_c_задержкой	customSpliterator_c_задержкой
3000	6,033	0,635	0,636
5000	10,066	1,033	1,033
50000	104,639	9,972	9,951

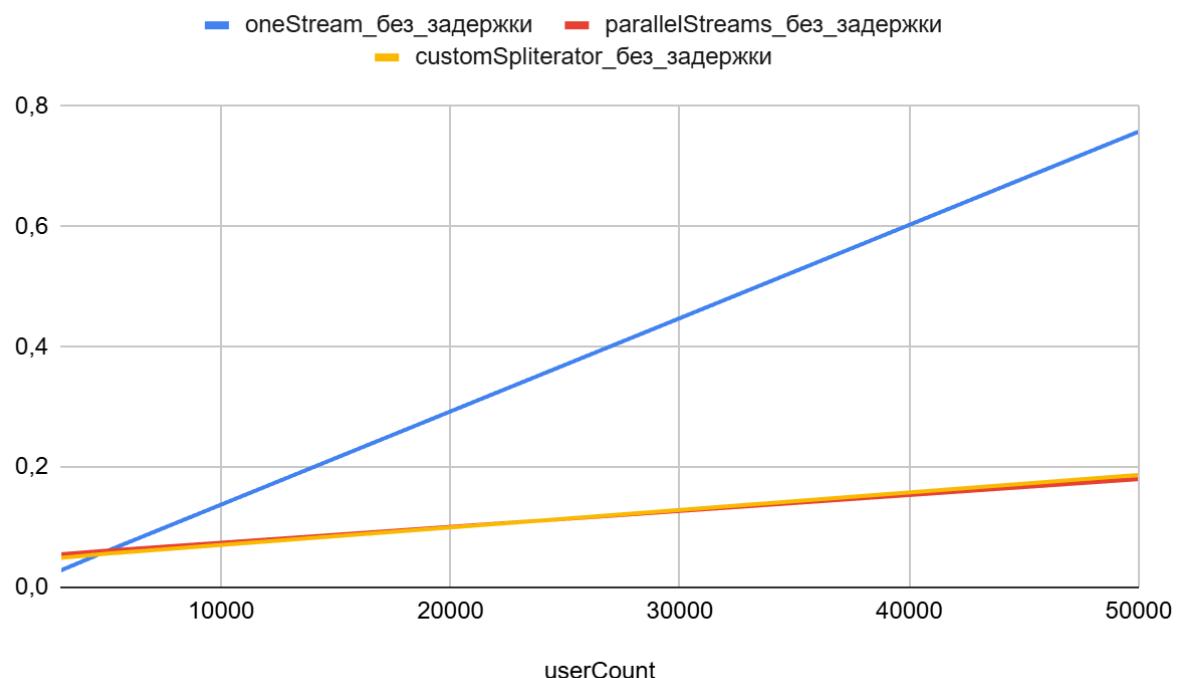


#### Результаты для вычисления счетчика количества активных пользователей

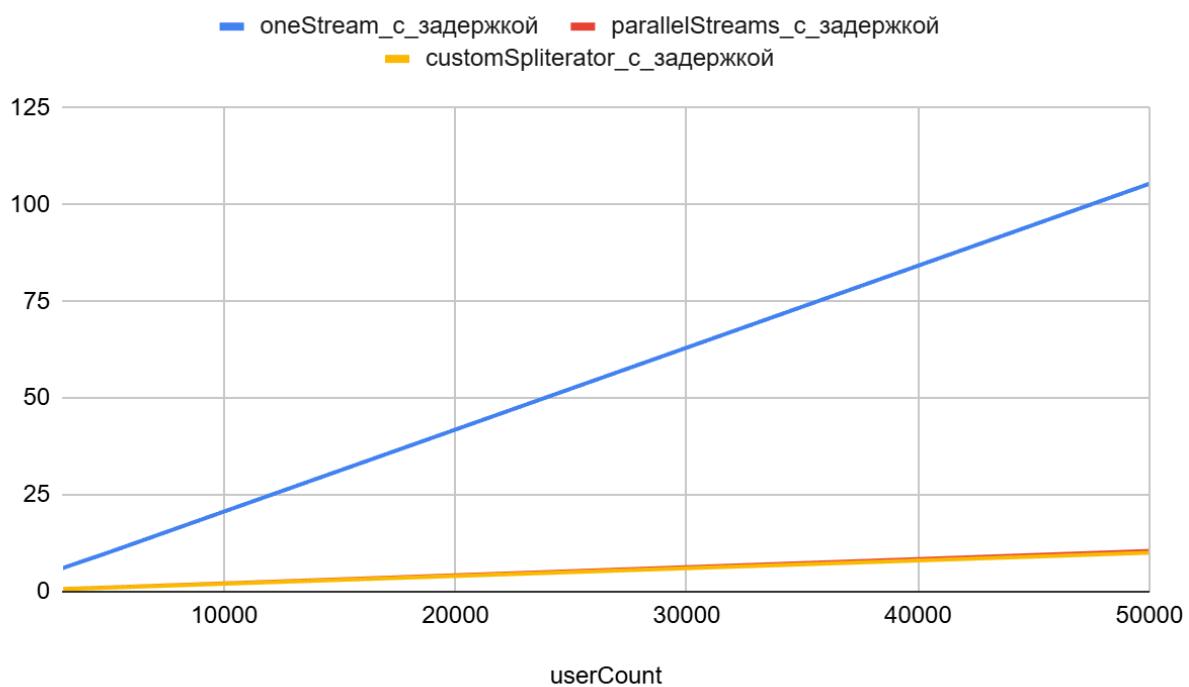
В случае наличия задержки при получении элементов parallel Stream и custom Spliterator имеет почти одинаковую производительность, обработка в одном потоке уже при малых значениях количества элементов сильно хуже, чем многопоточная обработка.

В случае отсутствия задержки custom Spliterator немного хуже, чем parallel stream, при малом количестве элементов (до 3000 тысяч) обработка в одном потоке лучше, чем многопоточная обработка из-за накладных расходов на обработку нескольких потоков).

Active users Lists без задержки				
userCount	oneStream_без_задержки	parallelStreams_без_задержки	customSpliterator_без_задержки	
3000	0,028	0,055	0,049	
5000	0,06	0,061	0,056	
50000	0,758	0,18	0,187	



Active users Lists без задержки			
userCount	oneStream_c_задержкой	parallelStreams_c_задержкой	customSpliterator_c_задержкой
3000	6,053	0,672	0,646
5000	10,128	1,074	1,038
50000	105,43	10,506	10,067



### Результаты для фильтрации массива активных пользователей

В случае наличия задержки при получении элементов parallel Stream и custom Spliterator имеет почти одинаковую производительность, обработка в одном потоке уже при малых значениях количества элементов сильно хуже, чем многопоточная обработка.

В случае отсутствия задержки custom Spliterator немного хуже, чем parallel stream, при малом количестве элементов (до 3000 тысяч) обработка в одном потоке лучше, чем многопоточная обработка из-за накладных расходов на обработку нескольких потоков).