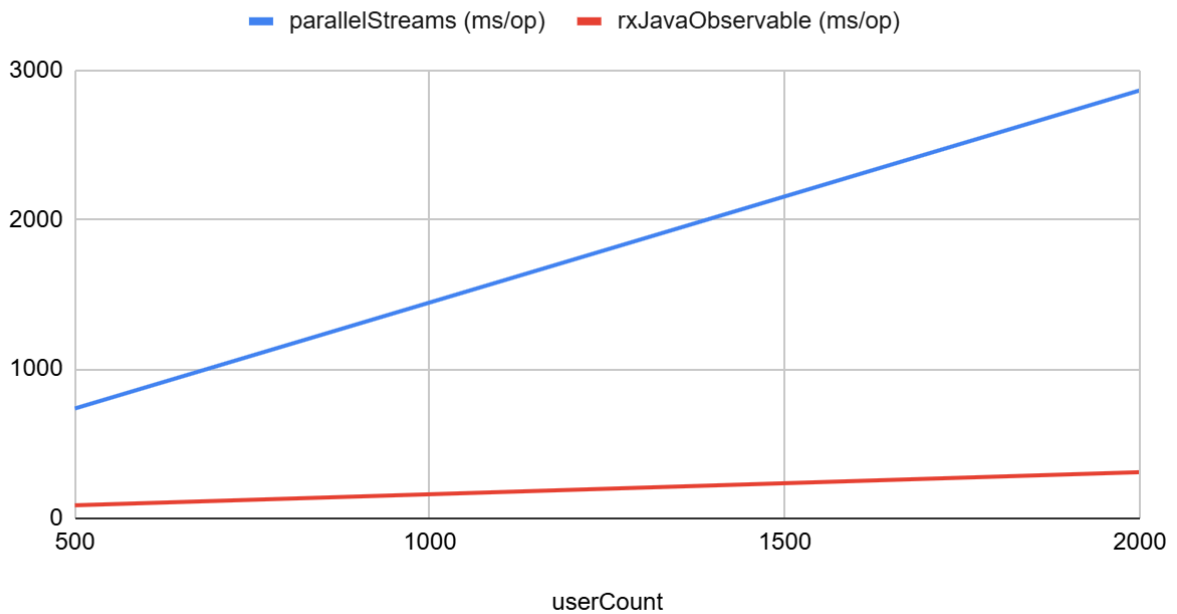


Active Users count	задержка - 50 мс		
userCount	parallelStreams (ms/op)	rxJavaObservable (ms/op)	Ускорение (parallel / rxJava)
500	735,519	87,408	8.41x
2000	2864,878	309,124	9.27x

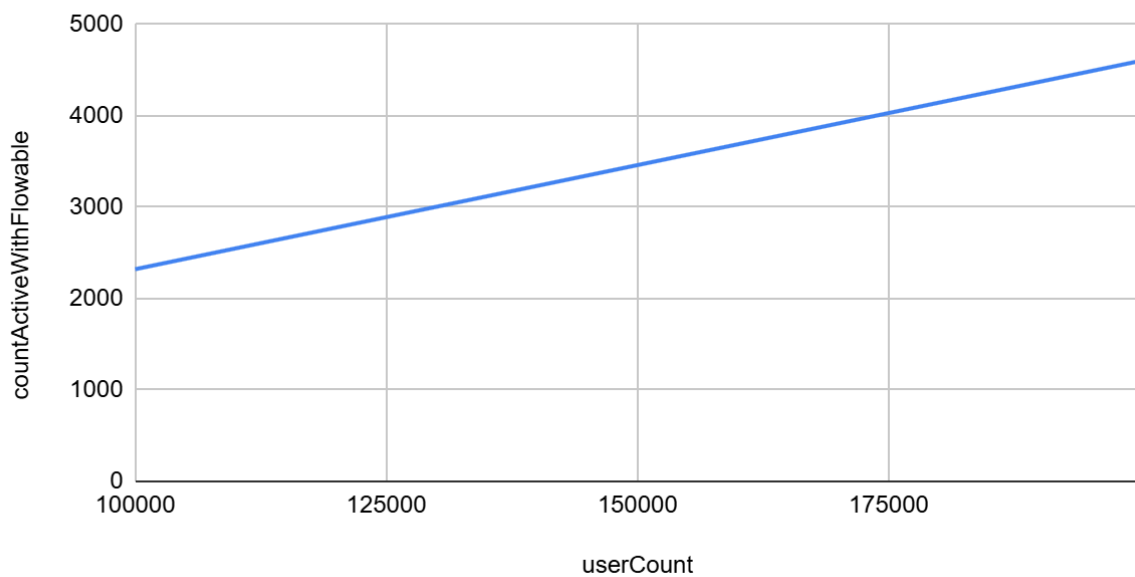
### parallelStreams (ms/op) и rxJavaObservable (ms/op)



Вывод: в I/O задачах использование асинхронного способа обработки данных против обычного синхронного (блокирующего) параллельного (многопоточного) выполнения дает прирост скорости в 8 - 9 раз, почти на порядок.

Active Users count	задержка отсутствует	
userCount	countActiveWithFlowable	
100000	2321,593	$4599,678 / 2321,593 = 1,98125941972$
200000	4599,678	

countActiveWithFlowable относительно параметра "userCount"



Время асинхронной генерации пользователей и сборка их статистики с использованием с поддержкой backpressure с помощью Flowable не зависит от их количества, рост времени выполнения идет равномерно с их количеством