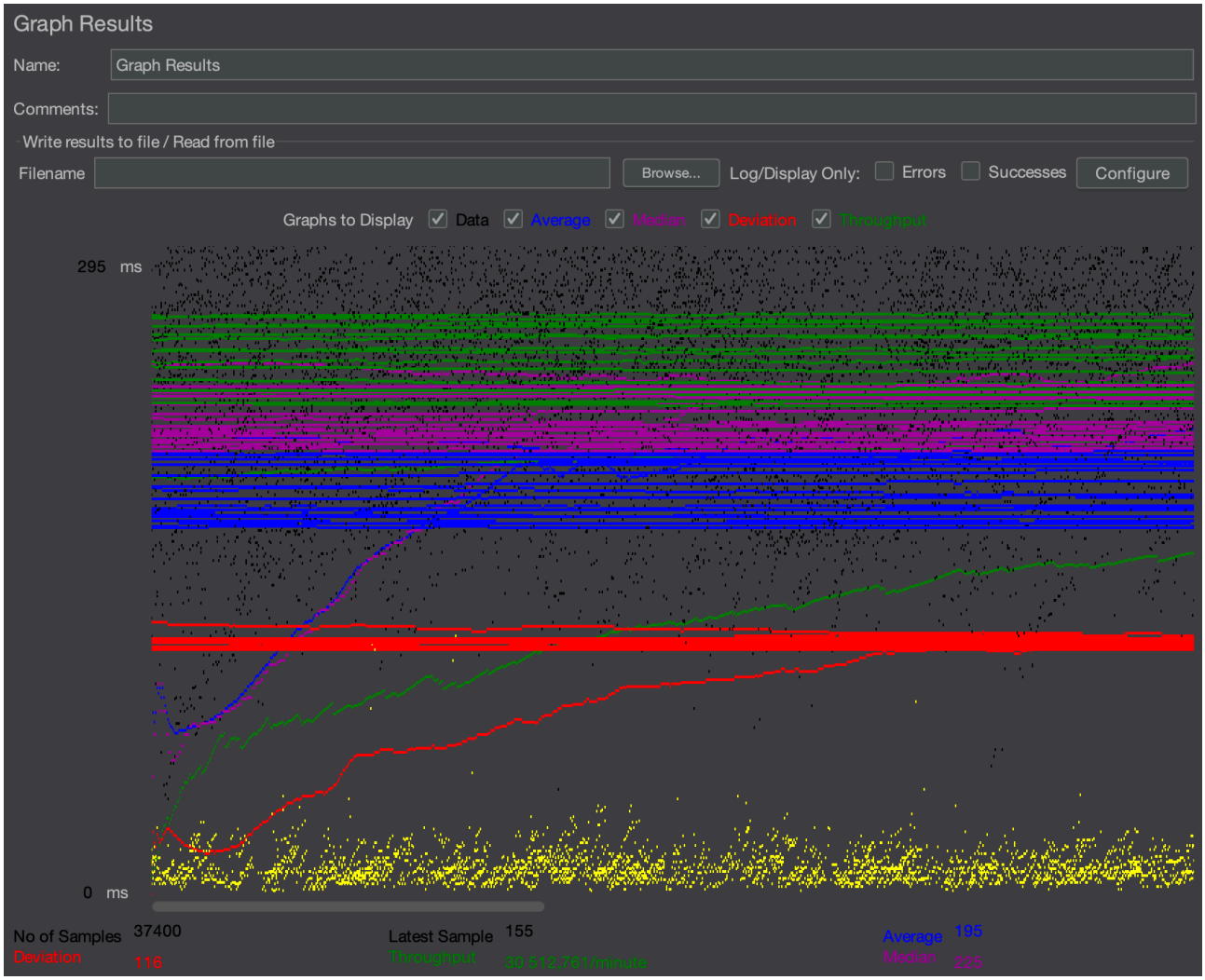
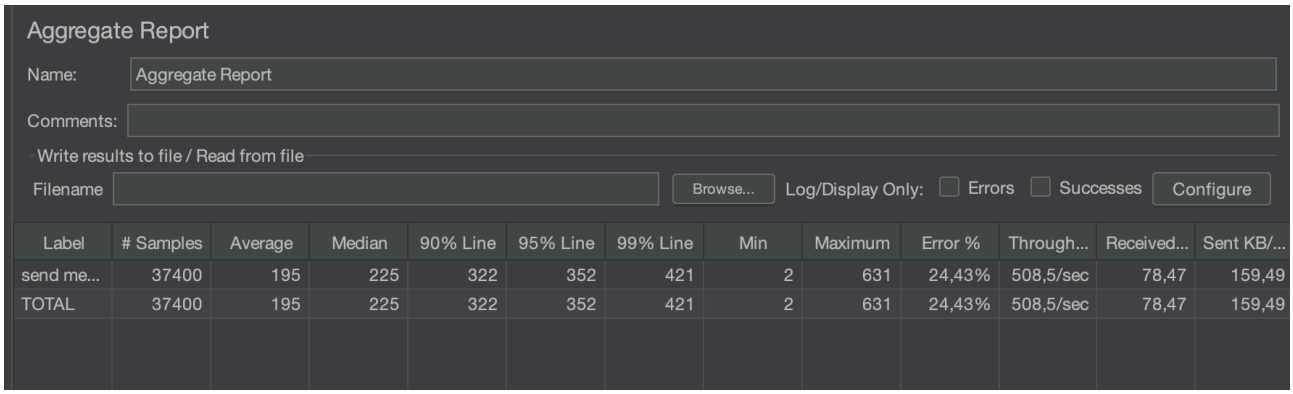
Для выполнения этой задачи был взят основной сервис без изменений (https://github.com/plohoicod/otus_highload_5.1), было убрана репликация в сервис, который работает с диалогами через Postgres (https://github.com/plohoicod/otus_highload_5.1), а так же был написан сервис, который работает с диалогами с использованием Redis(в обсуждения группы было сообщение о том что можно использовать Redis)(https://github.com/plohoicod/otus_highload_7).

Далее были проведены нагрузочные тесты, сначала для Postgres:

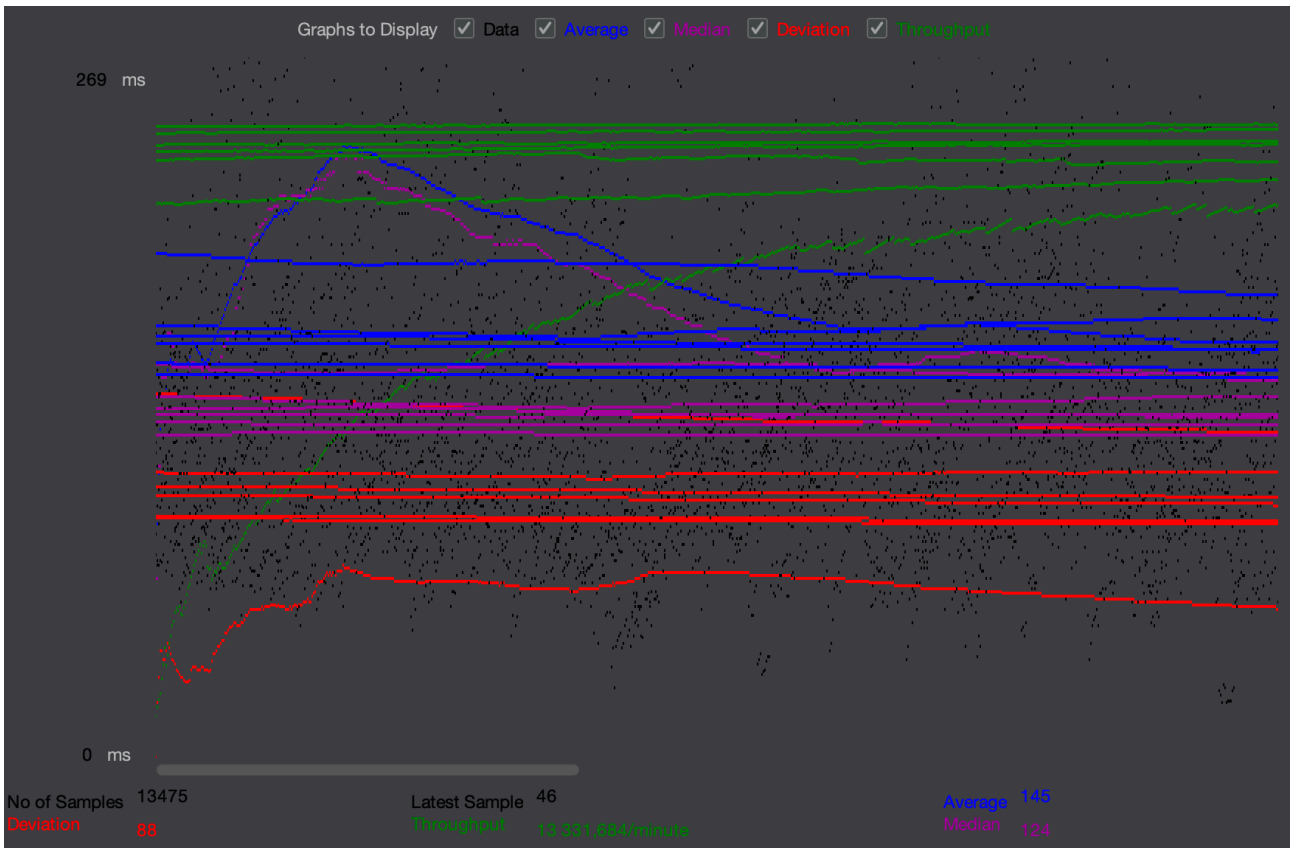
Запись сообщений:

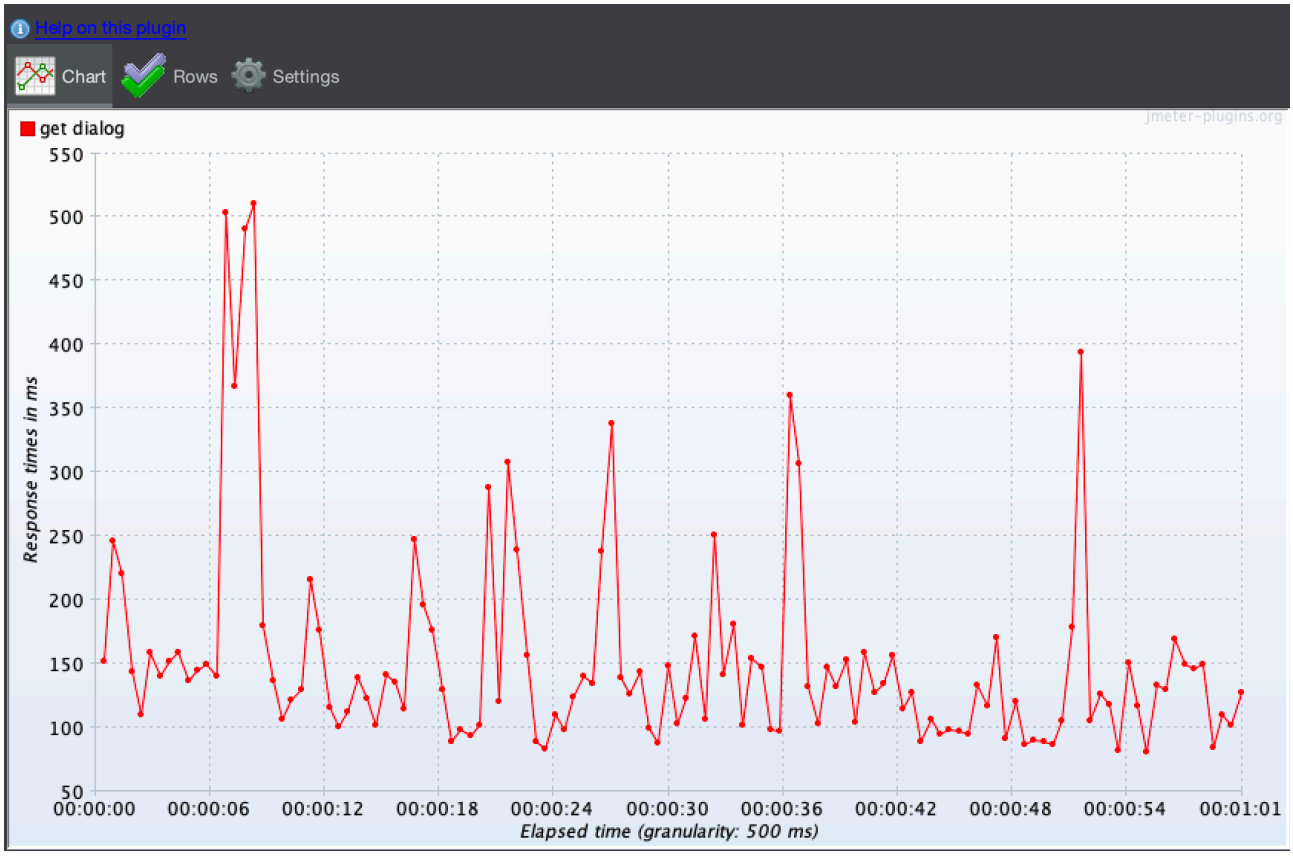




Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
send me...	37400	195	225	322	352	421	2	631	24,43%	508,5/sec	78,47	159,49
TOTAL	37400	195	225	322	352	421	2	631	24,43%	508,5/sec	78,47	159,49

Чтение сообщений:





Aggregate Report

Name: Aggregate Report

Comments:

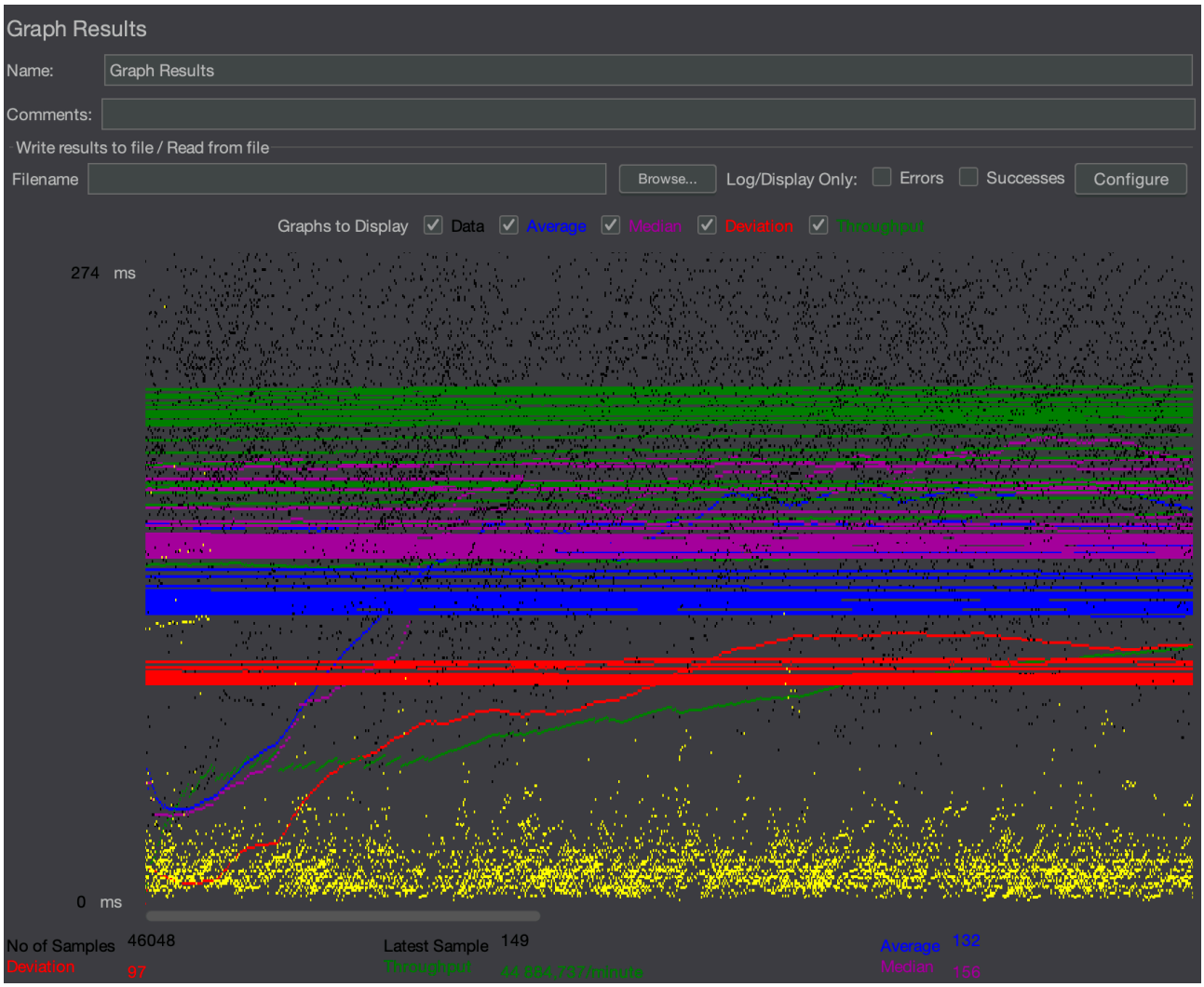
Write results to file / Read from file

Filename | Browse... | Log/Display Only: Errors Successes | Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
get dialog	13475	145	124	226	312	533	1	951	0,31%	222,2/sec	11462,12	48,02
TOTAL	13475	145	124	226	312	533	1	951	0,31%	222,2/sec	11462,12	48,02

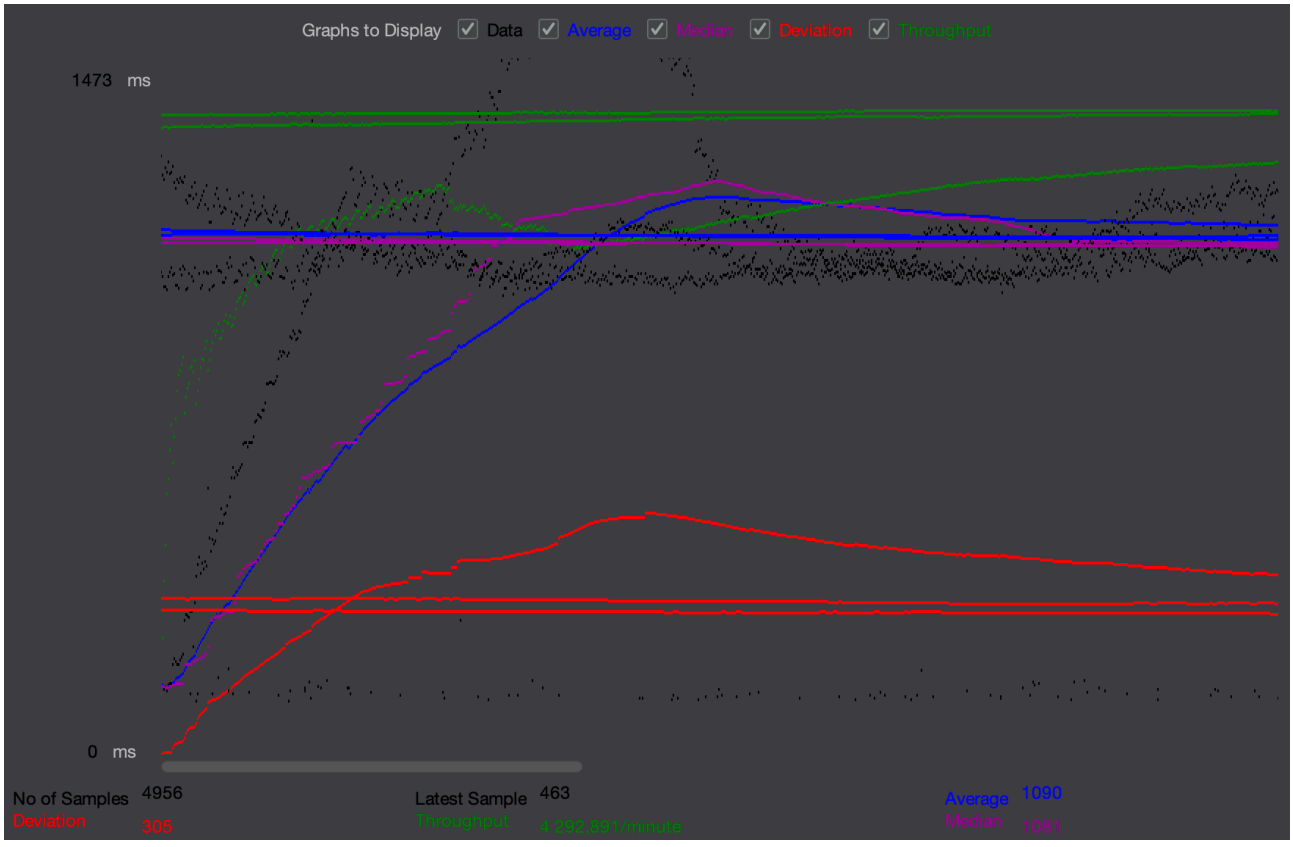
Далее были проведены нагрузочные тесты Redis:

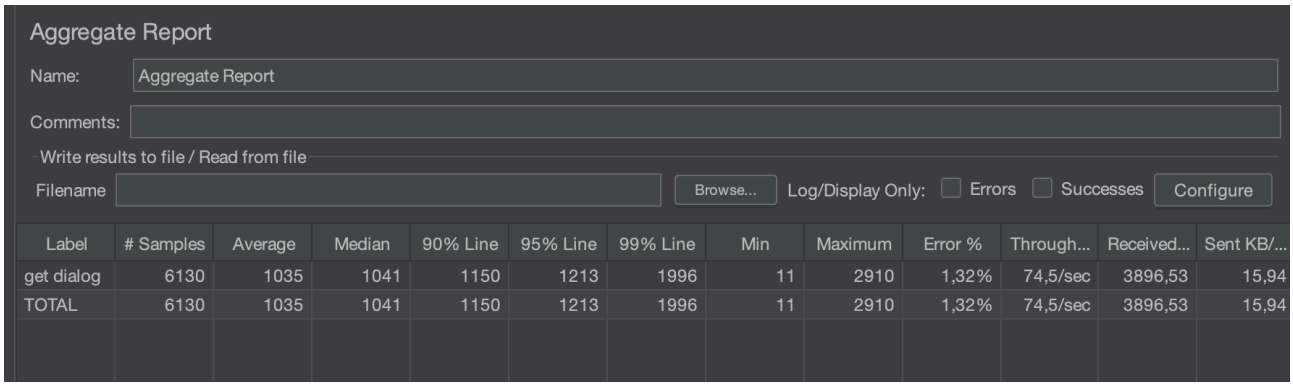
Запись сообщений:



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
send me...	46048	132	156	250	276	330	2	441	35,86%	748,1/sec	124,98	234,72
TOTAL	46048	132	156	250	276	330	2	441	35,86%	748,1/sec	124,98	234,72

Чтение сообщений:





Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Through...	Received...	Sent KB/...
get dialog	6130	1035	1041	1150	1213	1996	11	2910	1,32%	74,5/sec	3896,53	15,94
TOTAL	6130	1035	1041	1150	1213	1996	11	2910	1,32%	74,5/sec	3896,53	15,94

Все тесты были проведены с 100 параллельными пользователями. Исходя из результатов видно, что пропускная способность на запись в Redis на порядок выше чем в Postgres, а вот чтение работает медленнее, так как механизм поиска диалогов остался неизменным и происходит по отдельному id. Чтобы повысить скорость чтения, необходимо делать отдельный Redis hash для каждого диалога. Тестовый план находится в репозитории - outus7.jmx .