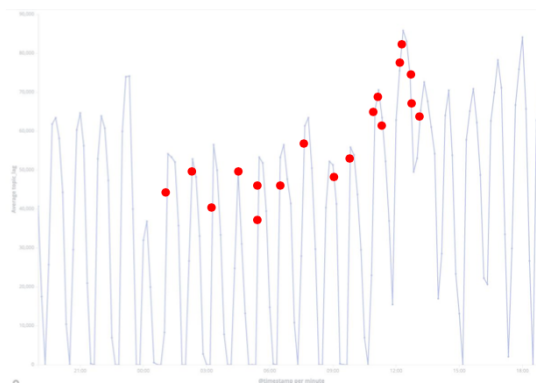


# 배포 후 kafka 이슈 해결 과정

```
[INFO ] 2021-03-26 11:01:23 [kafka_queue.py] [fetch:114] Offsets: {5: 81261079, 6: 81258162, 7: 81253393, 8: 81255753, 9: 81251980}
[INFO ] 2021-03-26 11:01:23 [kafka_queue.py] [on_revoke:85] partition revoked: [aurochs-piccoma-reco-jobs] (5, 6, 7, 8, 9)
[ERROR ] 2021-03-26 11:01:23 [kafka_queue.py] [fetch:131] msg error: KafkaError(code=_MAX_POLL_EXCEEDED,val=-147,str="Application maximum poll interval (300000ms) exceeded by 347ms")
```

이슈 : 프로세스에서 kafka max poll time을 초과해서 주기적으로 프로세스가 종료되는 문제



카프카 렉 모니터링

위의 빨간색 점들은 프로세스가 죽는 시간을 표시한 것입니다. 시간을 계산해보니 주기가 약 1시간 정도로 보였습니다. 의심 해볼 것들은 다음과 같았습니다.

- 저를 포함한 팀원의 실험 로직 문제
- 팀원들의 실험이 동시에 돌아가면서 생기는 새로운 문제

```
Every 2.0s: tail -25 1616741782_1817
gamma avg: 0.023567085234003834
theta: 9.231718301773071
theta avg: 0.014402056633031313
make_recs 117.84073662757874
logging_cache_us 117.84092855453491

load_parameter 3.337860107421875e-05
load_models 0.4305696487426758
fetched len: 5000
fetched 0.5202083587646484
gamma: 90.6003406047821
gamma avg: 0.02286155453060361
alpha: 103.08876442909241
alpha avg: 0.025304065888338834
beta: 57.57137846946716
beta avg: 0.014497954789591328
theta: 52.932575941085815
theta avg: 0.014168248378234962
make_recs 305.58776783943176
logging_cache_us 305.58802366256714

load_parameter 3.981590270996094e-05
load_models 0.6915724277496338
```

각 실험에 걸리는 시간과 각 과정에 걸리는 시간을 측정해 보았습니다. 기존에 배포되는 로직(alpha)과 각자의 실험(나머지)에 걸리는 로직은 시간이 거의 비슷하게 걸렸습니다.

그렇다면 폴 업데이트 시에 연산량이 기존의 4배로 늘어나면서 생기는 문제였습니다.



프로세스를 늘리면서 카프카 렉이 줄어드는 모습

저희는 consumer에서 한번에 가져오는 메세지 수를 5000 → 3000로 줄였고, 줄어든 처리량을 위해 프로세스를 1개 더 띄워서 이 문제를 해결하였습니다.

## 후기

- 카프카의 확장성에 대해서 느낄 수 있었습니다.
- 데이터 양에 따라서 프로세스 수만 조절하여 처리량을 늘릴 수 있다는 점을 배웠습니다.
- 협업의 가치를 배웠습니다.
  - 팀원 간의 의사소통을 통해서 문제를 더 빠르게 해결할 수 있었습니다.