Problem:

You have been given sample ***JSONs*** of three kinds of ***events*** that we capture in our product. You are also given a list of 10 random tracking Ids.

Your job is to:

A. Generate random events with randomly selected **Tracking Ids**, at a rate configured in a JSON based configuration file. **For example** 100 events per second. {***Events JSON mentioned below***}

B. Maintain an in-memory state of aggregated information from last N minutes (where N is specified in the configuration file by you), with an approximation of 10 seconds, about total events captured by 1.) type and 2.) within each type, by different tracking ids.

For example:. Data captured for any event type, would have a summary of last five minutes, something like - signup = {totalEventsCaptured: 50, eventsCapturedByTrackingIds:{trackingIdA: 10, trackingIdB: 20, trackingIdC: 20}}

C. ***Bonus assignment*** to impress us even more: Instead of using an in-memory cache, use a Memcached server - the IP address of which is stored in the configuration file. The key can be the type of event and the value will contain the sample JSON specified above.

This is a problem we have solved in our system, where we used to earlier query our datastore (Elasticsearch) for generating an aggregated summary of events captured in last N minutes for different event types. But that was putting unnecessary load on our ES server and was expensive for us hardware wise and maintenance wise. So we implemented this little piece ourselves using Memcached. This allows us to be very lean in hardware requirement and much faster in sending the latest state of last N minutes (with M seconds of error/approximation allowed). This way we compromise on accuracy for the last M seconds but we achieve greater performance, faster results to end-user and cheaper cost of our hardware requirements.

We are giving you the same problem and wish to see how you implement the time-based algorithm, and if you solve using Memcached, it will also show your learnability. Memcached is a very simple key-value store with a simple API and a great performance. We would love to see a "concurrency handled solution" with Memcached, especially under high traffic load of events being generated of each kind, and each tracking Id.

<>==========<>

Sample Events JSON of signup, review and user-event of the third type.

Use - https://jsoneditoronline.org

Signup event sample JSON

{"path":"/visitors/events/","value":{"fingerprint":"5d4697775392fc850a737fe225fbd8e9","sessionId":"0da4fc08-3f12-5890-fdce-2455fc17c394","visitorId":"be4ab37a-1de9-a0f0-f973-562b7cd4365e","trackingId":"INF-yj562hjojzbtez","userId":null,"userProfile":null,"geo":{"latitude":28.58,"longitude":77.33,"city":"Noida","country":"India","ip":"103.83.128.66"},"form":{"formId":"2bc04127-92d8-eb72-b1d3-26dc63f6ff19","email":"agency@gmail.com","anonymous":"Login"},"timestamp":"2019-12-08T06:42:57.289Z","**event**":"formsubmit","source":{"url":{"host":"app.useinfluence.co","hostname":"app.useinfluence.co","pathname":"/login","protocol":"https:"}},"referrer":""}}

User-event event sample JSON (click event)(there are 2 more event types)

{"path":"/visitors/events/","value":{"fingerprint":"5d4697775392fc850a737fe225fbd8e9","sessionId":"25ad9867-e9a5-cedf-0fc4-0c30b1c2a505","visitorId":"1bfdd0e4-2629-6ef9-21ea-9f1b1f11fe02","trackingId":"INF-yj562hjojzbtez","userId":null,"userProfile":null,"geo":{"latitude":28.58,"longitude":77.33,"city":"Noida","country":"India","ip":"103.83.128.17"},"target":{"id":"FPqR3acHqJeA3acH7MM9_0","selector":"div#FPqR2DbIqJeA2DbI7MM9_0.animated_FPqR2bI7Mf_c.slideInDown:nth-child(23)>div#FPqR3tRBqJeA3tRB7MM9_0:nth-child(1)>div:nth-child(1)>div:nth-child(1)>div:nth-child(2)>div#FPqR3dGiqJeA3dGi7MM9_0:nth-child(1)>div.FPqR2B_4qJeA2B_47MM9_0.rounded.FPqRD2zVqJeAD2zV7MM9_0:nth-child(1)>div#FPqR3acHqJeA3acH7MM9_0:nth-child(1)"},"timestamp":"2019-12-08T07:10:30.485Z","**event**":"click","source":{"url":{"host":"useinfluence.co","hostname":"useinfluence.co","pathname":"/","protocol":"https:"}},"referrer":""}}

Tracking ids

INF-yj562hjojzbtez
INF-3gbfcjjsd6vhvo
INF-ixpktk3itsk86
INF-1bi5qk0zocqcz