**Project 4 Report**

Dhruv Mahajan, UFID: 42111994

Ashvini Patel, UFID: 47949297

**What is implemented:**

Implemented the full brief.

**Part 1) Twitter Engine, which supports the following functions**

* Account Registration.
* Subscription to a user's tweets
* It allows querying tweets subscribed to.
* Tweets with specific hashtags.
* Tweets in which the user is mentioned.
* If the user is connected, we deliver the above types of tweets live (without querying)
* If user is disconnected, the server stores all the tweets for the user and delivers them when he reconnects.

**Part 2) Tester/simulator to test the engine**

* Each Client can:
  + Send tweet. Tweets can have hashtags (e.g. #COP5615isgreat) and mentions (@bestuser)
  + Re-tweets (so that your subscribers get an interesting tweet you got by other means)
  + Query for mentions, hastags or tweets subscribed to.
* We simulated periods of live connection and disconnection for random users

**Part 3) Zip-f distribution**

The program (client simulator) takes the minimum number of activities as input. Based on the input, the most subscribed to users (Users with a higher rank) set the minimum number of activities they need to do using the following:

**Zip-f version 1:**

* Top 1% of the clients do at least 20 times the minimum activities.
* Next 9% of the clients do at least 10 times the minimum activities.
* Next 50% of the clients do at least 2 times the minimum activities.
* Rest 40% of the clients do at least the minimum activities**.**

**Example:** If we simulate 1000 users with 100 minimum activities, the total requests in the system will be: 250,000

**Other considerations:**

**NOTE**: The Client simulator and server are two different processes. These can be on the same or different machines on the same local network but they must get the correct IP address from the init.getif() system call.

We assume that:

**The first address returned by the init.getif() system call is correct**. If the first IP address returned is not the address of the machine on the local network, no node can be named correctly. Hence nodes can’t connect.

**Activity:** An activity represents a single action that a client randomly selects**. Many actions can be triggered by a single activity**. For example, if a client tweets, it needs to be delivered to all the subscribers.

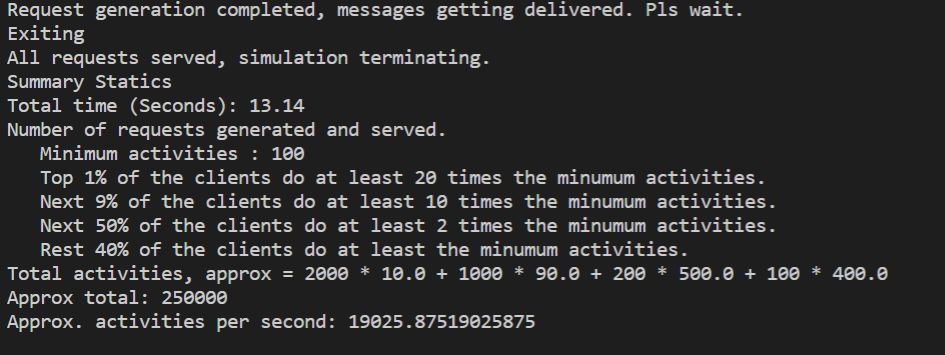
**Performance measures**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No. of Users** | **Minimum**  **Activities per user** | **Total Activities** | **Time (Seconds)** | **Activities handled per second** | **Figure** |
| Zip-f version 1, Two machines |  |  |  |  |  |
| 1,000 | 100 | 250,000 | 13.14 | ~19,025 | Figure 1 |
| 10,000 | 10 | 250,000 | 18.469 | ~13,536 | Figure 2 |
| 10,000 | 100 | 2,500,000 | 705 | ~3,542 | Figure 3 |
| 20,000 | 10 | 500,000 | 171 | ~2,909 | Figure 4 |
| Zip-f version 1, One machine |  |  |  |  |  |
| 10,000 | 100 | 2,500,000 | 86 | ~28,849 | Figure 7 |
| 100,000 | 1000 | 2,500,000 | 65 | ~38,332 | Figure 8 |
| Zip- version 2 (only for testing) |  |  |  |  |  |
| 1,000 | 10 | 33,000 | 3.3 | ~9,731 | Figure 5 |
| 10,000 | 100 | 3,300,000 | 575 | ~5,731 | Figure 6 |

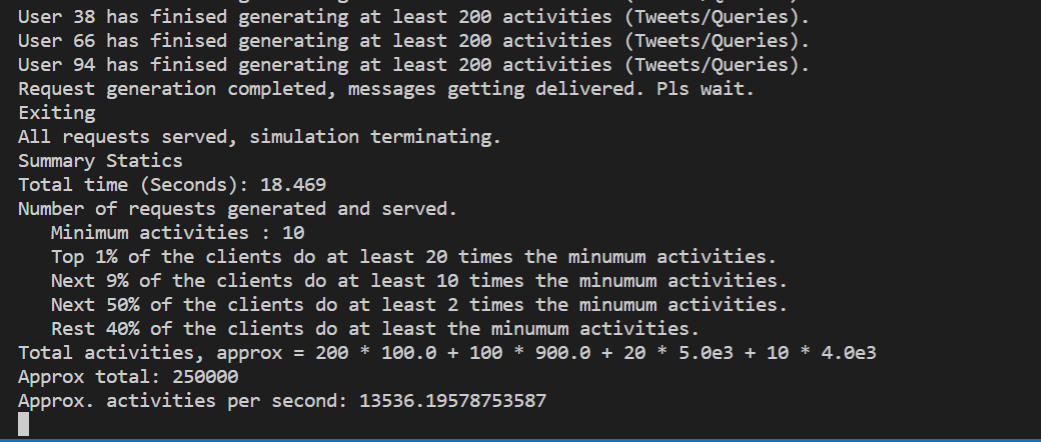
We concluded that when using two machines (like in a true client-server environment), **the network is the main bottleneck.**

**Note Zip-f version 2 is not in the final submission.** Just something we tried. It has a very big multiplier in terms of number of activities and hence was notincluded in the final submission**.**

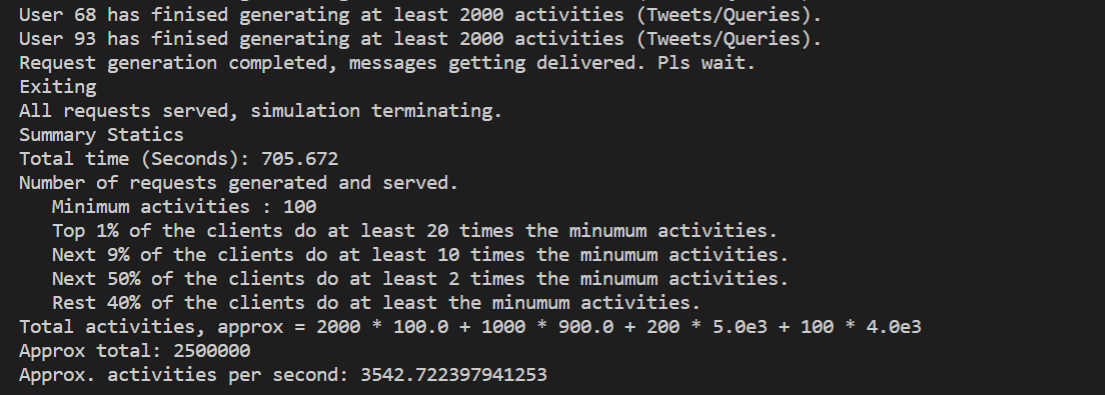
* Top 1% of the clients do at least 100 times the minimum activities.
* Next 9% of the clients do at least 10 times the minimum activities.
* Next 50% of the clients do at least 2 times the minimum activities.
* Rest 40% of the clients do at least the minimum activities**.**



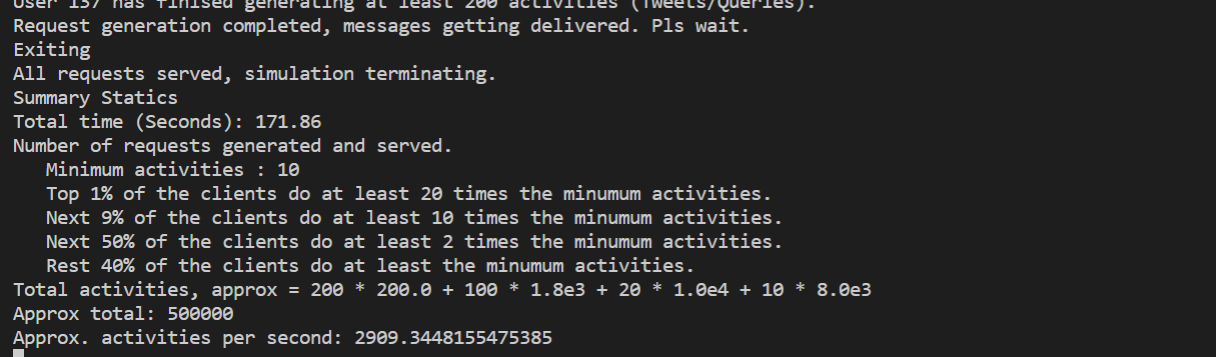
**Figure 1**



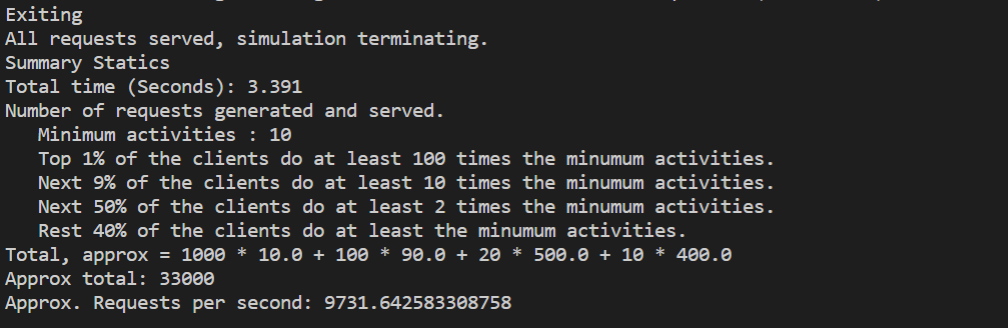
**Figure 2**



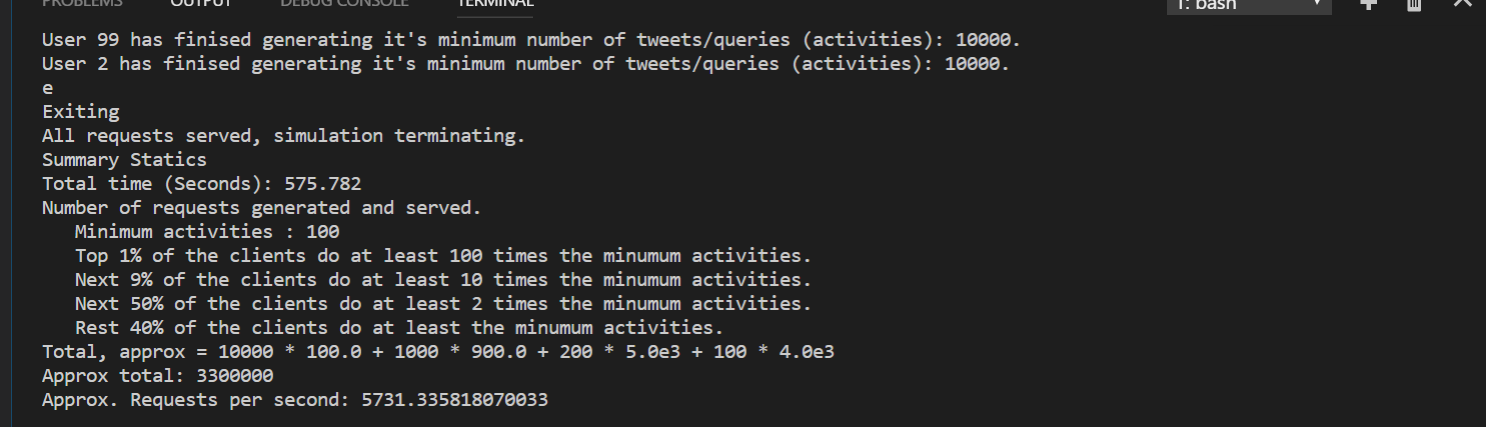
**Figure 3**



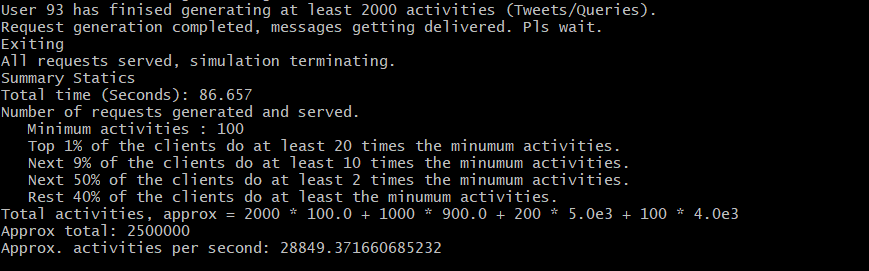
**Figure 4**



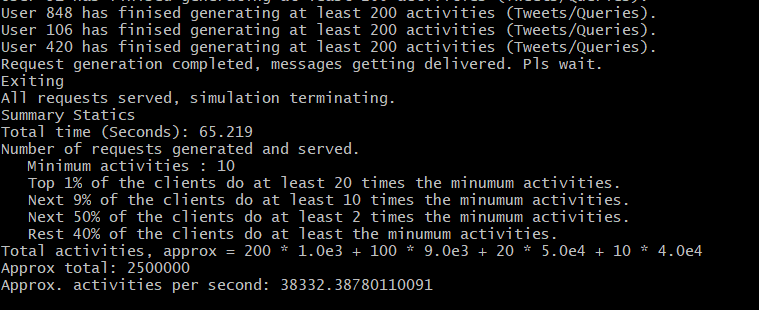
**Figure 5**



**Figure 6**



**Figure 7**



**Figure 8**

**Test setup & resource utilization:**

All testing and was done using one or two machines with Core i5 processors and 16GB memory each. In case of two, they were connected to the same local network via wireless connections. Since the server is a single process, it doesn’t use up all the on a multicore machine.

**RAM Usage:**

Server: 500 – 800 MB

Client simulator: Upto 1.5 GB