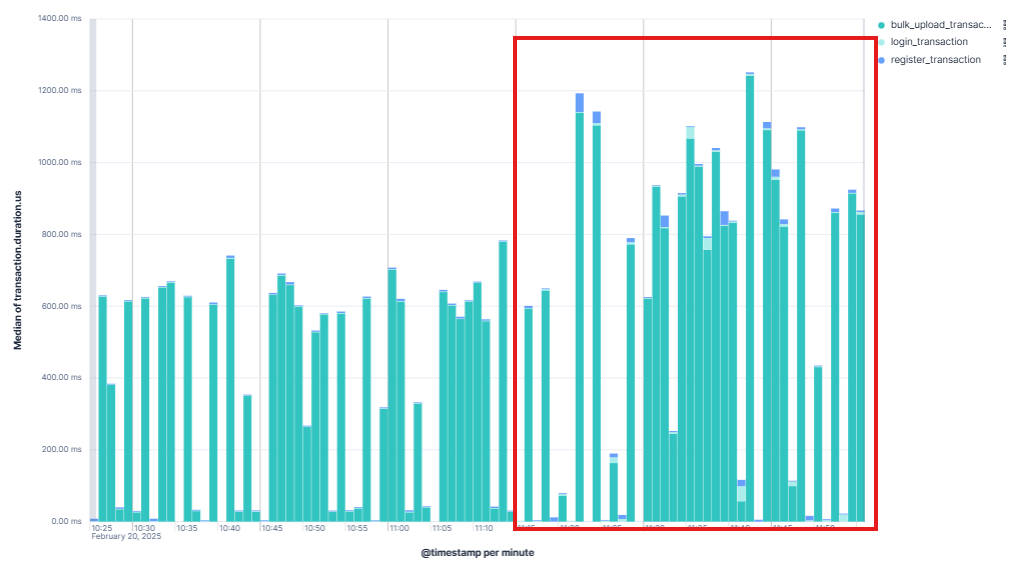
# Monithor Project - Observability Exercise

**Observation:**

**Back-end Performance Testing with Locust:**

Simulated User Behavior:  
impact of Locust load on simulated user by selenium - increase in response time of selenium transactions when loading the system with Locust (red box):

  
Median of selenium transaction duration

Locust - 100 domains check, test results:

- 100 Domains with 10 users avg time ~34 Sec.   
- Avg time is not close to PRD specifications which is 5 sec (100 domains ,10 users).  
- System crashed after 50 transactions on test of 50 users.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concurrent Users | Avg. Response  Time | 90th percentile | 95th percentile | Failure rate |
| 10 | 34240 | 46000 | 302000 | 0 |
| 50 | X | X | X | 0 |
| 100 | X | X | X | X |
| 200 | X | X | X | X |
| 300 | X | X | X | X |

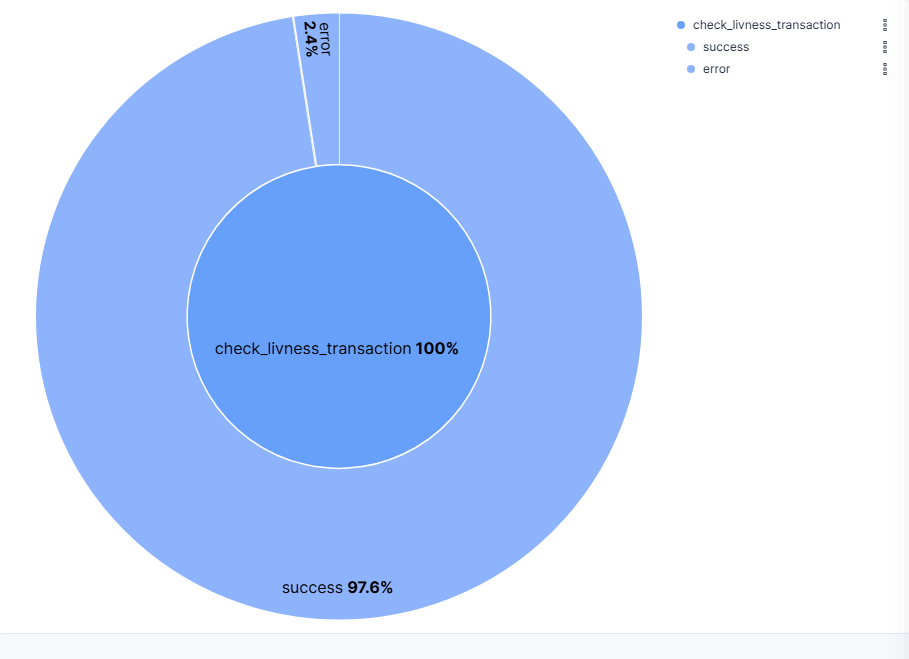
Locust - Single Domain check, test results:

- Reasonable performance up to 100 users.  
- Extreme degradation seen when ramping from 100 users to 200 users.  
- No failures seen on any load up to 300 users (for more accurate observation about failures need to check in Elastic APM)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Concurrent Users | Avg. Response  Time | 90th percentile | 95th percentile | Failure rate |
| 10 | 1026 | 1900 | 2900 | 0 |
| 50 | 1138 | 2200 | 2800 | 0 |
| 100 | 1160 | 2300 | 3000 | 0 |
| 200 | 23696 | 96000 | 100000 | 0 |
| 300 | 100720 | 181000 | 186000 | 0 |

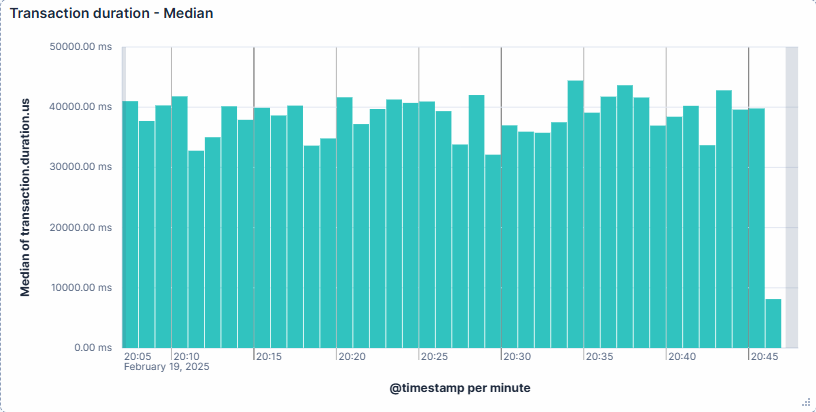
**Performance Insights:**

- Errors rate ~ 2.4 % caused by system load.



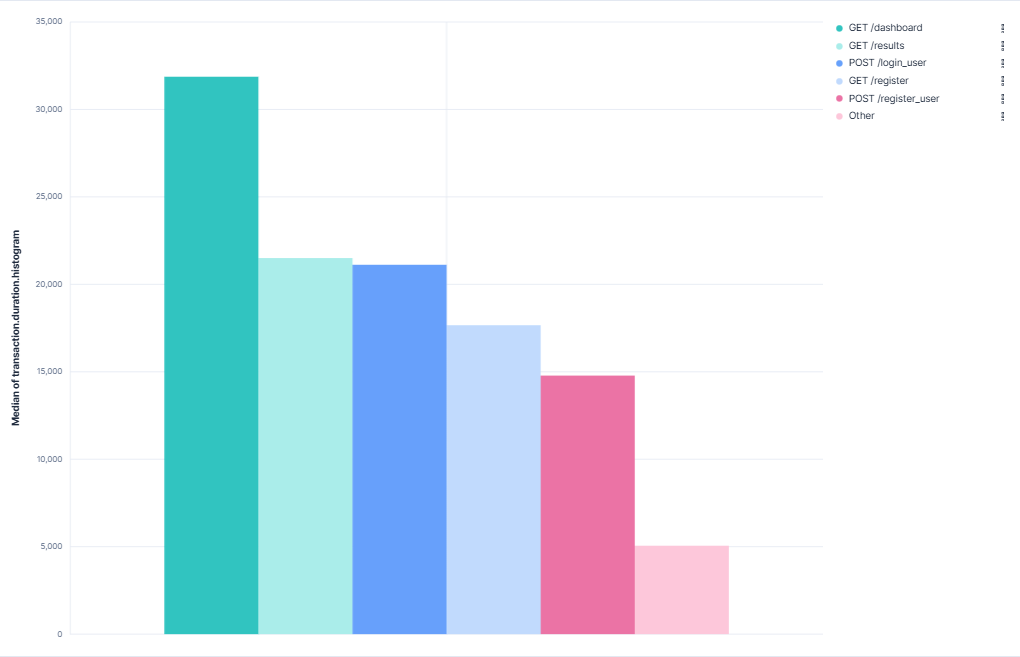
Errors rate duration during load test of 10 users

- Elastic results of transaction duration looks similar to Locust observation.

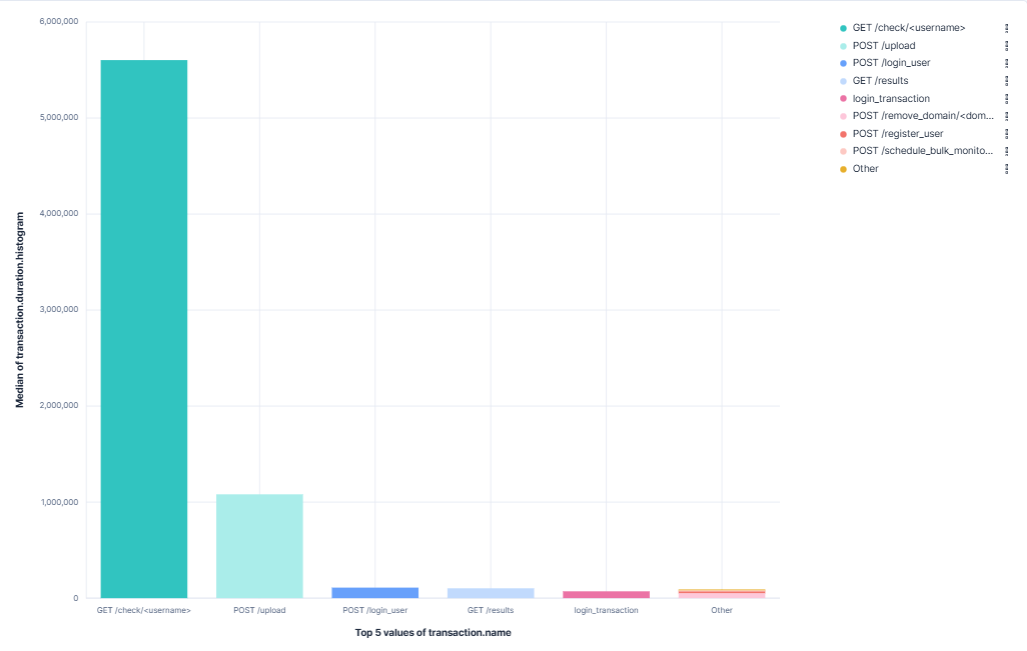


Transaction duration during load test of 10 users

**Frontend transactions**



Front-end UI transactions median

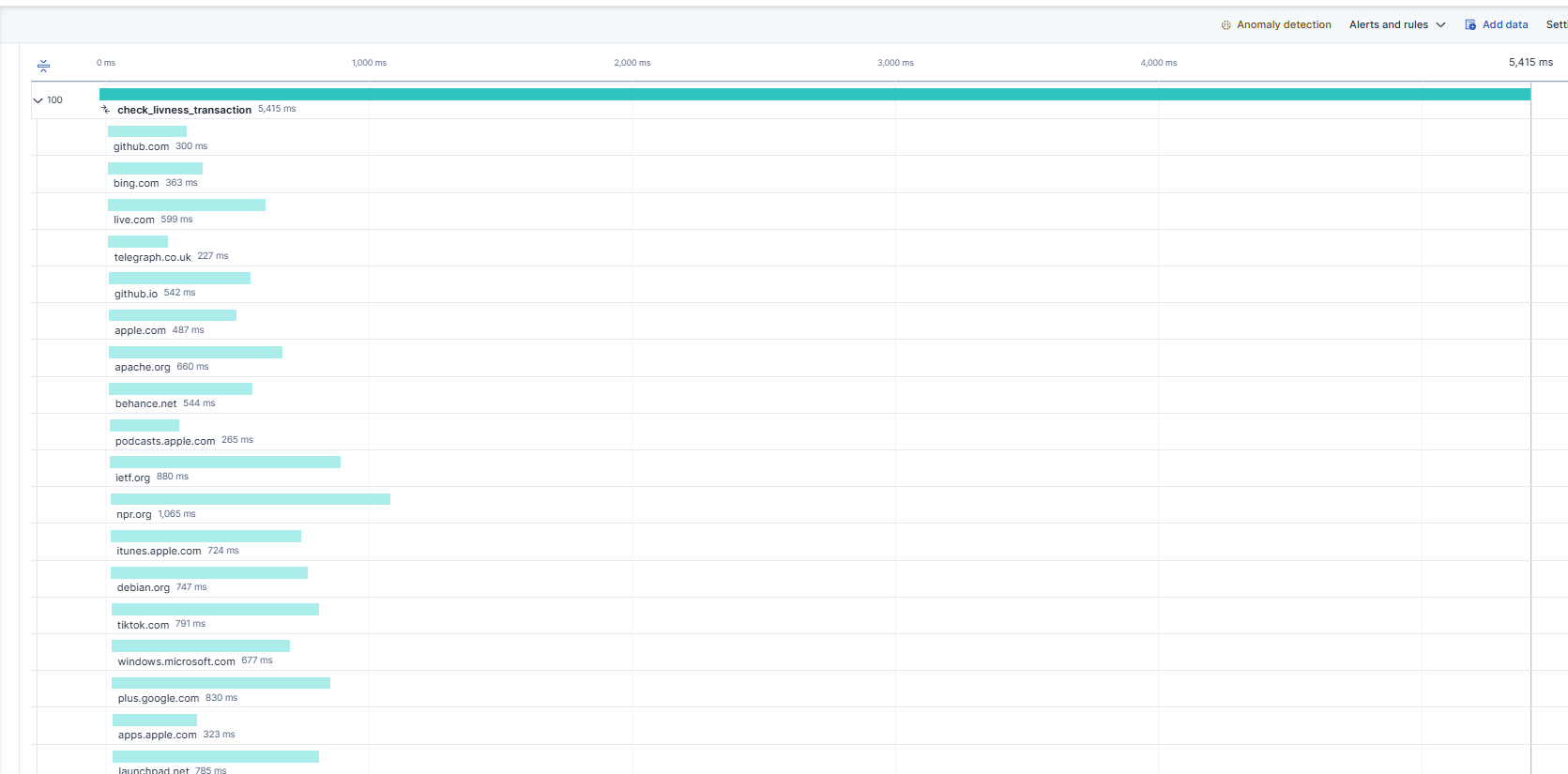


All transactions duration median

- Check, upload and login transactions are on top of duration time .

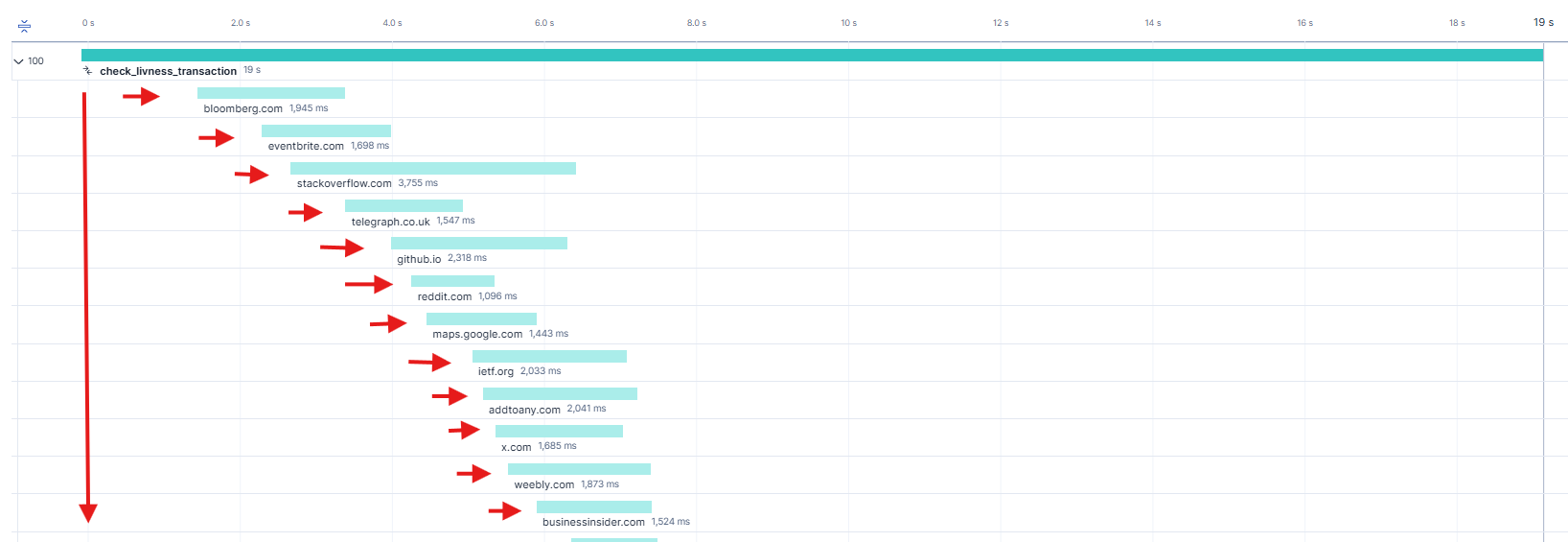
Spans

- When the system is not loaded spans (reflection of threads in this case) are starting almost at the same time , whole transaction elapsed time is determined by the slowest span.

"check\_livness\_transaction" - trace samples - **Not loaded system**

In this case - “Adobe.com”:   
  
 "check\_livness\_transaction" - trace samples - **Not loaded system, top span**

- On loaded system spans (threads in our case) are not starting at the same time probably due to lack of resources in system causing additional impact on transaction time.

"check\_livness\_transaction" - trace samples - **loaded system**