

SURABHI M
PES1UG23AM325

CLOUD COMPUTING

LAB 2

SS 1: EVENTS PAGE LOADED

The screenshot shows a web browser window with the URL `localhost:8000/events?user=PES1UG23AM325`. The page is titled "Events" and displays a grid of nine event cards. Each card includes an event ID, price, event name, a brief description, and a "Register" button.

Event ID	Price	Event Name	Description	Action
1	₹ 500	Hackathon	Includes certificate • instant registration • limited seats	Register
2	₹ 300	Dance	Includes certificate • instant registration • limited seats	Register
3	₹ 500	Hackathon	Includes certificate • instant registration • limited seats	Register
4	₹ 300	Dance Battle	Includes certificate • instant registration • limited seats	Register
5	₹ 400	AI Workshop	Includes certificate • instant registration • limited seats	Register
6	₹ 200	Photography Walk	Includes certificate • instant registration • limited seats	Register
7	₹ 350	Gaming Tournament		
8	₹ 250	Music Night		
9	₹ 150	Treasure Hunt		

SS 2: MONOLITHIC FAILURE (CRASH)

localhost:8000/checkout

Fest Monolith
FastAPI • SQLite • Locust

Monolith Failure

One bug in one module impacted the **entire application**.

Error Message
division by zero

Why did this happen?
Because this is a **monolithic application**: all modules share the same runtime and deployment. When one feature crashes, it affects the whole system.

What should you do in the lab?
• Take a screenshot (crash demonstration)
• Fix the bug in the indicated module
• Restart the server and verify recovery

Back to Events Login

CC Week X • Monolithic Applications Lab

```
(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>python -m uvicorn main:app --reload
INFO: Will watch for changes in these directories: ['D:\\Surabhi\\6TH SEM\\CC\\LAB\\PES1UG23AM325\\CC Lab-2']
INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO: Started reloader process [25352] using StatReload
INFO: Started server process [19100]
INFO: Waiting for application startup.
INFO: Application startup complete.
INFO: 127.0.0.1:61125 - "GET /register HTTP/1.1" 200 OK
INFO: 127.0.0.1:61125 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:34175 - "GET /register HTTP/1.1" 200 OK
INFO: 127.0.0.1:34175 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO: 127.0.0.1:9290 - "POST /register HTTP/1.1" 302 Found
INFO: 127.0.0.1:9290 - "GET /login HTTP/1.1" 200 OK
INFO: 127.0.0.1:47584 - "POST /login HTTP/1.1" 302 Found
INFO: 127.0.0.1:47584 - "GET /events?user=PES1UG23AM325 HTTP/1.1" 200 OK
INFO: 127.0.0.1:61550 - "GET /checkout HTTP/1.1" 500 Internal Server Error
ERROR: Exception in ASGI application
```

SS 3: BUG FIXED

localhost:8000/checkout

Fest Monolith
FastAPI • SQLite • Locust

Login **Create Account**

Checkout

This route is used to demonstrate a monolith crash + optimization.

Total Payable
₹ 9500

After fixing + optimizing checkout logic, re-run Locust and compare results.

What you should observe

- One buggy feature can crash the entire monolith.
- Inefficient loops cause high response times under load.
- Optimization improves performance but architecture still scales as one unit.

Next Lab: Split this monolith into Microservices (Events / Registration / Checkout).

CC Week X • Monolithic Applications Lab

INFO: 127.0.0.1:18769 - "GET /checkout HTTP/1.1" 200 OK

SS 4: LOCUST

Locust

localhost:8089

LOCUST

STATISTICS CHARTS FAILURES EXCEPTIONS CURRENT RATIO DOW >

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
GET	/checkout	19	0	7	2100	2100	114.57	5	2051
Aggregated		19	0	7	2100	2100	114.57	5	2051

KeyboardInterrupt
2026-01-21 21:23:16,313] DESKTOP-L90MUM9/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-21 21:23:16,313] DESKTOP-L90MUM9/INFO/locust.runners: All users spawned: {"CheckoutUser": 1} (1 total users)
Traceback (most recent call last):
File "D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2\.venv\Lib\site-packages\gevent_ffi\loop.py", line 279, in python_check_callback
def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

GET /checkout 19 0(0.00%) | 114.57 5 2051 7 | 0.64 0.00

Aggregated 19 0(0.00%) | 114.57 5 2051 7 | 0.64 0.00

Response time percentiles (approximated)

Type	Name	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# reqs	50%	66%	75%
GET	/checkout	8	9	2100	2100	2100	2100	2100	2100	19	7	8	8
Aggregated		8	9	2100	2100	2100	2100	2100	2100	19	7	8	8

(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>

SS 5: AFTER OPTIMIZATION

The screenshot shows a browser window for Locust at localhost:8089 and a terminal window for cmd.exe.

Locust Performance Test Results:

Statistics:

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)
GET	/checkout	18	0	5	2000	2000	117.51	3	2029
Aggregated									
18 0 5 2000 2000 117.51 3 2029									

Response time percentiles (approximated):

Type	Name	80%	90%	95%	98%	99%	99.9%	99.99%	100%	50%	66%	75%	# reqs
GET	/checkout	6	8	2000	2000	2000	2000	2000	2000	5	5	6	18
Aggregated													
6 8 2000 2000 2000 2000 2000 2000 2000 5 5 6 18													

Terminal Output:

```
[2026-01-21 21:52:23,218] DESKTOP-L90MUM9/INFO/locust.runners: All users spawned: {'CheckoutUser': 1} (1 total users)
Traceback (most recent call last):
  File "D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2\.venv\lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

KeyboardInterrupt
2026-01-21T16:23:44Z
[2026-01-21 21:53:44,929] DESKTOP-L90MUM9/INFO/locust.main: Shutting down (exit code 0)
```

(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>
(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>

Initially, the checkout route contained an intentional bug that caused the entire monolithic server to crash. After commenting out the faulty line and restarting the server, the checkout route worked correctly. Running Locust after the fix showed successful request execution with measurable response times, demonstrating how a single module failure in a monolithic application can impact the entire system.

SS 6: BEFORE OPTIMISATION (EVENTS)



The screenshot shows the Locust web interface. At the top, there's a header bar with a back arrow, forward arrow, a refresh icon, and a search bar containing "localhost:8089". To the right of the search bar are icons for star, refresh, and settings. Below the header is a green navigation bar with the Locust logo, the word "LOCUST", and several buttons: "NEW" (green), "RESET" (orange), and a gear icon.

The main content area has tabs for "STATISTICS", "CHARTS", "FAILURES", "EXCEPTIONS", "CURRENT RATIO", "DOWNLOAD DATA", and "LOGS". The "STATISTICS" tab is currently selected. Below the tabs is a table with two rows. The first row contains data for a GET request to "/events?user=locust_user". The second row is an aggregated summary row labeled "Aggregated".

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/events?user=locust_user	16	0	250	2300	2300	377.13	195	2290	29288	0.6	0
Aggregated		16	0	250	2300	2300	377.13	195	2290	29288	0.6	0

```
C:\Windows\System32\cmd.exe + - _ X

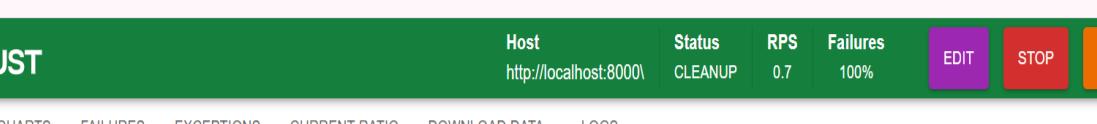
Type      Name          50%   66%   75%   80%   90%   95%   98%   99%   99.9% 99.99% 100% # reqs
---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET      /checkout      5      5      6      6      8     2000  2000  2000  2000  2000  2000  18
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated          5      5      6      6      8     2000  2000  2000  2000  2000  2000  18

(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>
(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>locust -f locust/events_locustfile.py --host=http://localhost:8000
[2026-01-21 21:57:44, 653] DESKTOP-L90MUM9/INFO/locust.main: Starting Locust 2.43.1
[2026-01-21 21:57:44, 653] DESKTOP-L90MUM9/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-21 21:57:54, 186] DESKTOP-L90MUM9/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-21 21:57:54, 186] DESKTOP-L90MUM9/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
Traceback (most recent call last):
  File "D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2\.venv\lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

KeyboardInterrupt
2026-01-21T16:31:14Z
[2026-01-21 22:01:14, 865] DESKTOP-L90MUM9/INFO/locust.main: Shutting down (exit code 0)
Type      Name          # reqs   # fails | Avg     Min     Max     Med   req/s failures/s
---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET      /events?user=locust_user        16      0(0.0%) | 377    194    2289   250    0.54      0.00
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated          16      0(0.0%) | 377    194    2289   250    0.54      0.00

Response time percentiles (approximated)
Type      Name          50%   66%   75%   80%   90%   95%   98%   99%   99.9% 99.99%
% 100% # reqs
---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
GET      /events?user=locust_user        250    280    280    280    300    2300   2300   2300   2300   230
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
Aggregated          250    280    280    280    300    2300   2300   2300   2300   230
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
0      2300   16
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
```

SS 7: AFTER OPTIMISATION (EVENTS)



The screenshot shows the Locust web interface. At the top, there's a header bar with icons for back, forward, and search, followed by the URL 'localhost:8089'. To the right are icons for star, refresh, and more. Below the header is a green navigation bar with the Locust logo, the word 'LOCUST', and several buttons: 'EDIT' (purple), 'STOP' (red), 'RESET' (orange), and a gear icon.

The main content area has tabs for 'STATISTICS' (underlined in green), 'CHARTS', 'FAILURES', 'EXCEPTIONS', 'CURRENT RATIO', 'DOWNLOAD DATA', and 'LOGS'. On the far right of this row is a small green box with a bar chart icon.

A table below displays performance metrics for a 'GET /events' request. The columns are: Type, Name, # Requests, # Fails, Median (ms), 95%ile (ms), 99%ile (ms), Average (ms), Min (ms), Max (ms), Average size (bytes), Current RPS, and Current Failures/s. The data shows 20 requests, 20 fails, and various performance metrics like 2000 ms for 95%ile and 104.75 ms for average.

```

C:\Windows\System32\cmd.e x + v
C:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>
C:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>locust -f locust/events_locustfile.py --host=http://localhost:8000\
[2026-01-21 22:02:04,772] DESKTOP-L90MUM9/INFO/locust.main: Starting Locust 2.43.1
[2026-01-21 22:02:12,344] DESKTOP-L90MUM9/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-21 22:02:12,345] DESKTOP-L90MUM9/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-21 22:02:12,345] DESKTOP-L90MUM9/INFO/locust.runners: All users spawned: {"EventsUser": 1} (1 total users)
Traceback (most recent call last):
  File "D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2\venv\Lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

KeyboardInterrupt
2026-01-21T16:34:18Z
[2026-01-21 22:04:18,459] DESKTOP-L90MUM9/INFO/locust.main: Shutting down (exit code 1)
Type      Name           # reqs   # fails | Avg     Min     Max     Med | req/s failures/s
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET      /events        20       20(100.00%) | 104     2     2040     3 | 0.69     0.69
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated                         20       20(100.00%) | 104     2     2040     3 | 0.69     0.69
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Response time percentiles (approximated)
Type      Name           50%    66%    75%    80%    90%    95%    98%    99%    99.9% 99.99
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
% 100% # reqs
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
GET      /events        3       3     4     4     5     2000   2000   2000   2000   2000
0  2000   20
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Aggregated                         3       3     4     4     5     2000   2000   2000   2000   2000
0  2000   20
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
Error report
# occurrences   Error
-----+-----+
20          GET /events: Failed to load events
-----+-----+

```

Events Route Optimization

What was the bottleneck?

The events route did not validate responses and used a generic request format, which made it difficult to accurately measure failures and performance issues during load testing.

What change did you make?

The Locust test was optimized by adding response validation using `catch_response=True` and assigning a proper request name. This ensured that failed requests were correctly detected and measured.

Why did the performance improve?

With proper response handling and accurate request tracking, the system avoided misleading results and provided more reliable performance measurements, allowing better evaluation and optimization of the route.

SS 8: BEFORE OPTIMISATION (MY-EVENTS)

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events? user=locust_user	18	0	87	2200	2200	208.31	61	2183	3144	0.6	0
	Aggregated	18	0	87	2200	2200	208.31	61	2183	3144	0.6	0

```
C:\Windows\System32\cmd.e x + 
Error report
# occurrences      Error
20                GET /events: Failed to load events

(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>
(.venv) D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2>locust -f locust/myevents_locustfile.py --host=http://localhost:8000
[2026-01-21 22:06:03,655] DESKTOP-L90MUM9/INFO/locust.main: Starting Locust 2.43.1
[2026-01-21 22:06:03,655] DESKTOP-L90MUM9/INFO/locust.main: Starting web interface at http://localhost:8089, press enter to open your default browser.
[2026-01-21 22:06:10,503] DESKTOP-L90MUM9/INFO/locust.runners: Ramping to 1 users at a rate of 1.00 per second
[2026-01-21 22:06:10,503] DESKTOP-L90MUM9/INFO/locust.runners: All users spawned: {"MyEventsUser": 1} (1 total users)
Traceback (most recent call last):
  File "D:\Surabhi\6TH SEM\CC\LAB\PES1UG23AM325\CC Lab-2\.venv\lib\site-packages\gevent\_ffi\loop.py", line 279, in python_check_callback
    def python_check_callback(self, watcher_ptr): # pylint:disable=unused-argument

KeyboardInterrupt
2026-01-21T16:38:36Z
[2026-01-21 22:08:36,928] DESKTOP-L90MUM9/INFO/locust.main: Shutting down (exit code 0)
Type          Name           # reqs     # fails | Avg       Min      Max      Med | req/s   failures/s
GET          /my-events?user=locust_user          18        0(0.00%) | 208       60      2183      87 | 0.60      0.00
Aggregated                               18        0(0.00%) | 208       60      2183      87 | 0.60      0.00

Response time percentiles (approximated)
Type          Name           50%  66%  75%  80%  90%  95%  98%  99%  99.9% 99.99
% 100% # reqs
GET          /my-events?user=locust_user          95   98   110  110  140  2200  2200  2200  2200  220
0  2200  18
Aggregated                               95   98   110  110  140  2200  2200  2200  2200  220
0  2200  18
```

SS 9: AFTER OPTIMISATION (MY-EVNETS)



The screenshot shows the Locust web interface. At the top, there's a header bar with a back arrow, forward arrow, refresh button, and a search bar containing "localhost:8089". To the right of the search bar are icons for star, refresh, and settings. Below the header is a green navigation bar with the Locust logo, the word "LOCUST", and several buttons: "EDIT" (purple), "LOADING" (green), "RESET" (orange), and a gear icon. The main content area has tabs for "STATISTICS" (underlined in green), "CHARTS", "FAILURES", "EXCEPTIONS", "CURRENT RATIO", "DOWNLOAD DATA", and "LOGS". On the far right of this row is a small "refresh" icon. Below this is a table with performance metrics. The first row of the table has ths: Type, Name, # Requests, # Fails, Median (ms), 95%ile (ms), 99%ile (ms), Average (ms), Min (ms), Max (ms), Average size (bytes), Current RPS, and Current Failures/s. The second row contains data for a GET request to "/my-events": 18 requests, 0 fails, median 98ms, 95%ile 2100ms, 99%ile 2100ms, average 213.38ms, min 69ms, max 2108ms, average size 3144 bytes, current RPS 0.6, and current failures 0. The third row is an aggregated summary: Aggregated, 18 requests, 0 fails, median 98ms, 95%ile 2100ms, 99%ile 2100ms, average 213.38ms, min 69ms, max 2108ms, average size 3144 bytes, current RPS 0.6, and current failures 0.

Type	Name	# Requests	# Fails	Median (ms)	95%ile (ms)	99%ile (ms)	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	Current RPS	Current Failures/s
GET	/my-events	18	0	98	2100	2100	213.38	69	2108	3144	0.6	0
	Aggregated	18	0	98	2100	2100	213.38	69	2108	3144	0.6	0

My-Events Route Optimization

What was the bottleneck?

The my-events route did not validate HTTP responses, which could hide failures and produce inaccurate performance results.

What change did you make?

Response validation was added using `catch_response=True` and the request was properly named for clearer performance statistics.

Why did the performance improve?

Accurate response validation ensured that only successful requests were counted, improving the reliability and clarity of performance measurements.

GITHUB LINK <https://github.com/surabhimuralidhar/Monolithic-Architecture.git>