



INFORMATIKOS FAKULTETAS

T120B162 Programų sistemų testavimas

4 laboratorinis darbas

Studentas: Gabija Šeškauskaite, IFF-0/2

Dėstytojas: lekt. Dominykas Barisas

KAUNAS 2023

Turinys

1. Įvadas	3
2. CPU testavimas	3
3. Atminties testavimas	6
4. Apkrovos testavimas	7

1. Įvadas

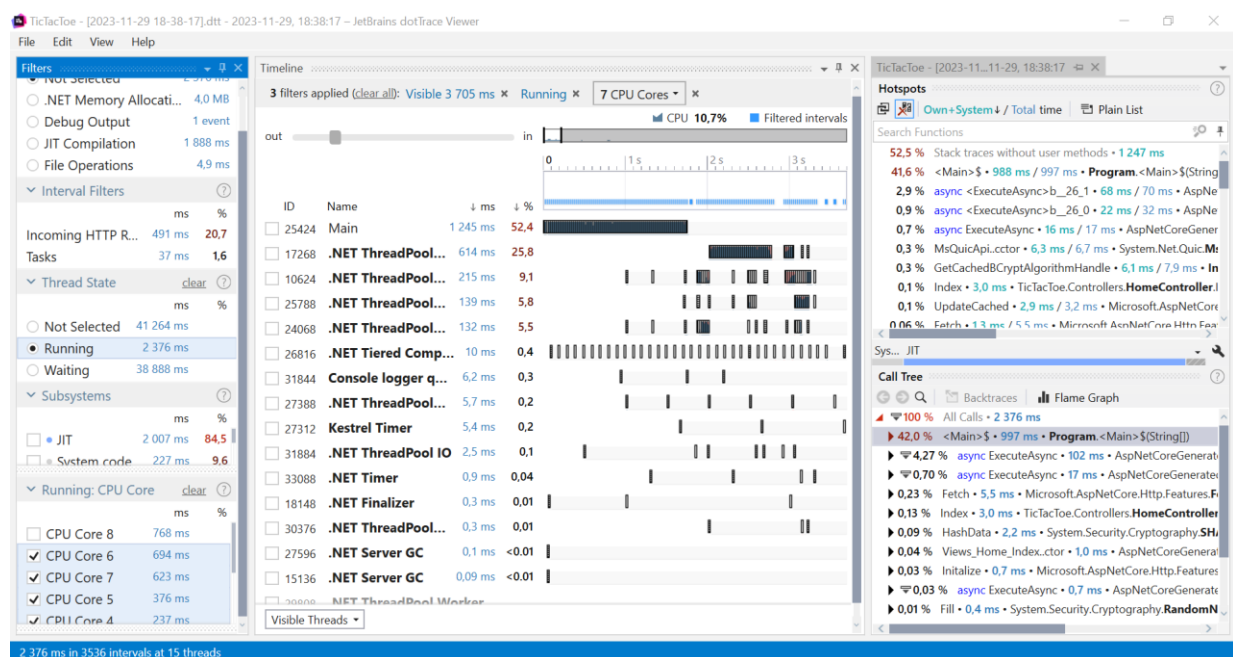
4 laboratorinio darbo tikslas – išanalizuoti programos sveikimą

Darbo uždaviniai:

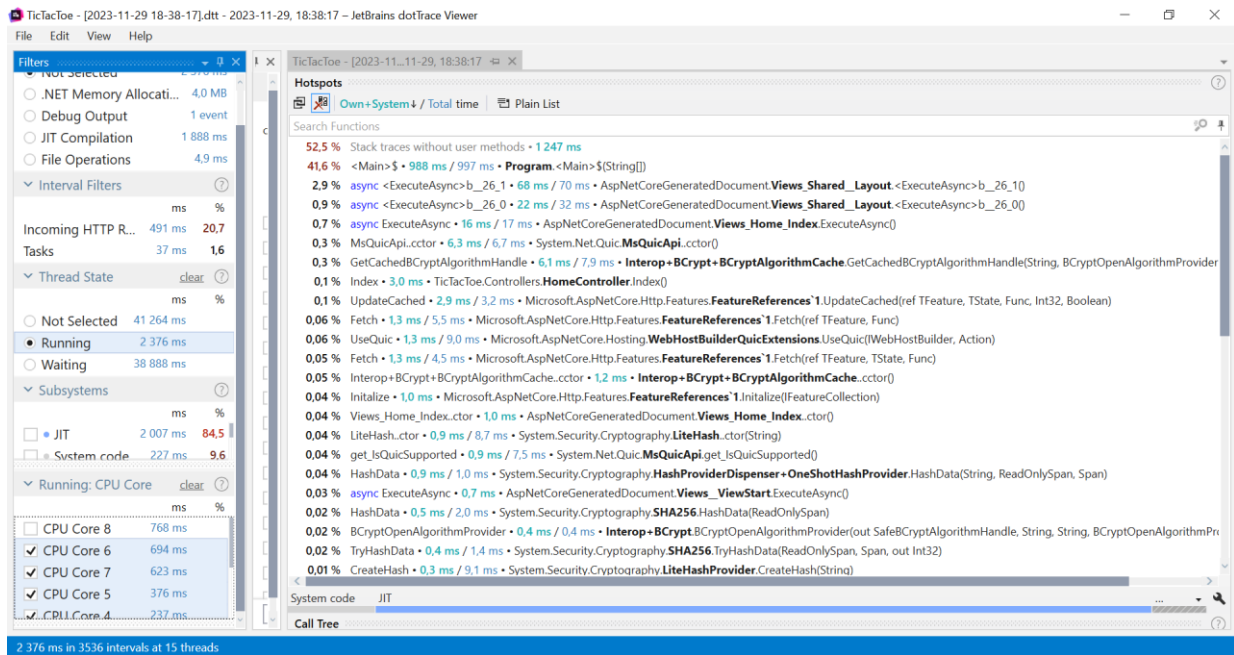
- 1) Pamatuoti CPU naudojimą;
- 2) Pamatuoti atminties naudojimą;
- 3) Apkrovos testavimas;
- 4) Rezultatus pateikti laboratorinio darbo ataskaitoje.

2. CPU testavimas

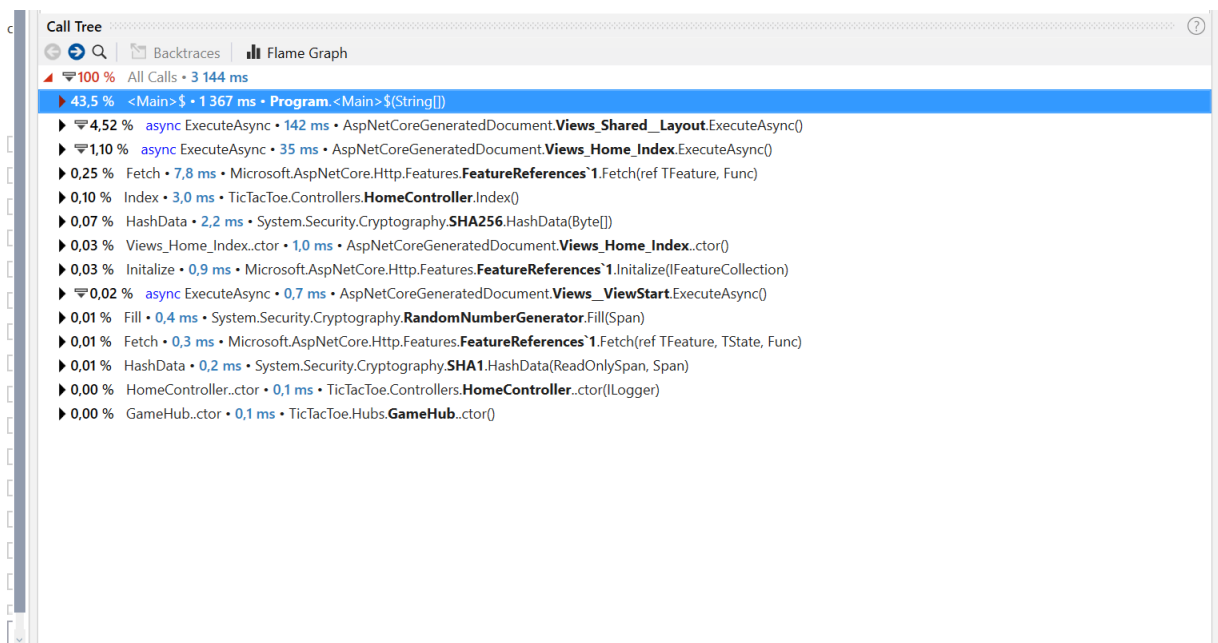
CPU matavimui buvo naudotas dotTrace Profiler.



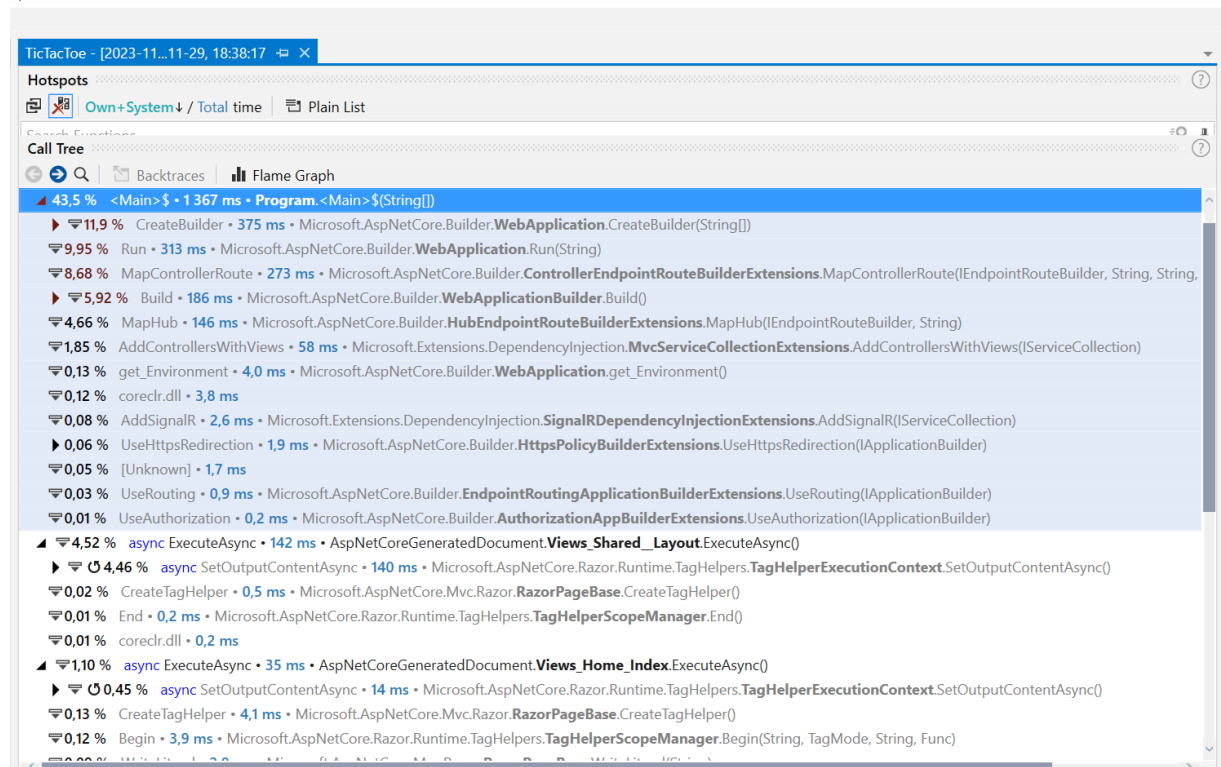
Timeline dalis rodo kada CPU buvo naudojamas (tamsios dalys) ir kada buvo tiesiog laukiama ir CPY buvo nenaudojamas (šviesios dalys).



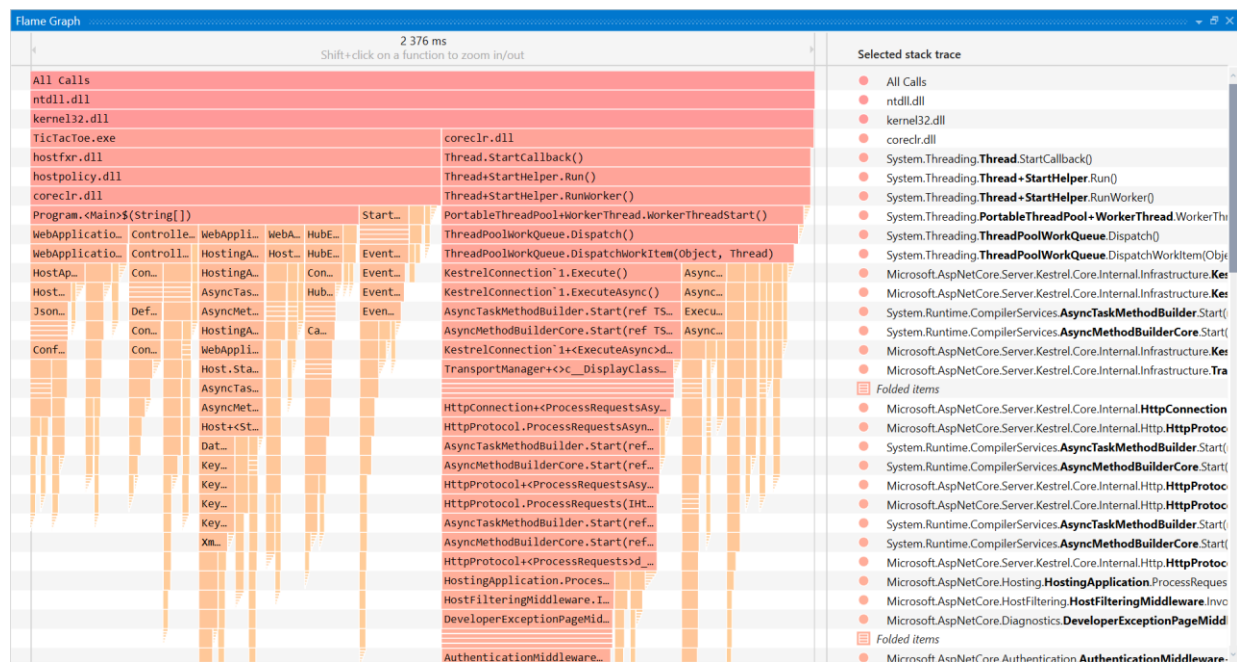
Višosios interneto prieigos taškai rodo metodų, kurie buvo vykdomi filtruotais laiko intervalais, sąrašą. Hotspots rodo kur buvo naudojama daugiausia CPU. Galima matyti, kad daugiausia CPU naudojo Main metodus. Po jo eina async metodai.



Call tree rodo metodo vykdymo kelius. Kiekvienas medžio metodo mazgas reiškia aukščiausio lygio metodą, vykdomą tam tikros gijos. Metodo mazgas pateikia metodų, kuriuos jis iškviatė, sąrašą ir pagrindinio iškviatimo pomedžio vykdymo laiką.

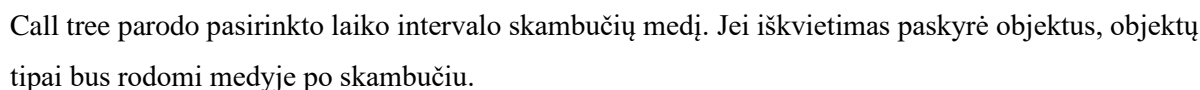
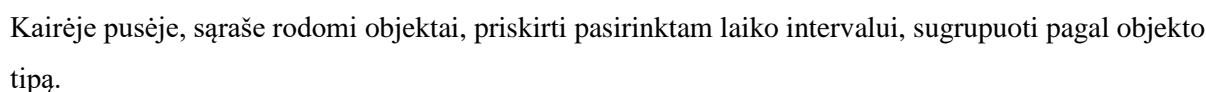


Čia galima pamatyti išskleistus Main ir async metodus.



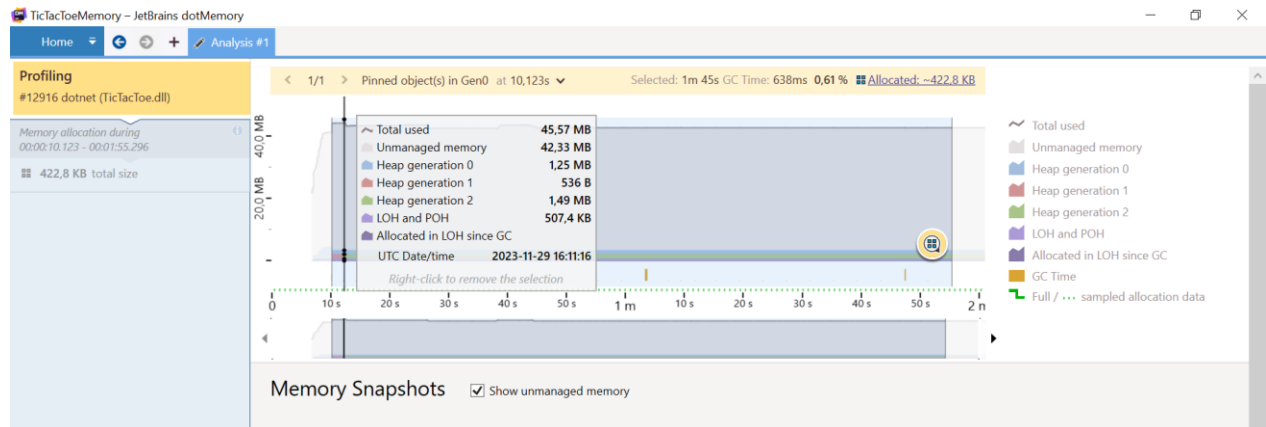
Flame graph yra grafinis Call tree vaizdas. Kiekvienas call rodomas kaip horizontali juosta, kurios ilgis priklauso nuo viso call laiko, kuris lygus pačios funkcijos laikui + visų antrinių funkcijų laikui. Kuo ilgesnis call, tuo ilgesnė juosta.

Atminties matavimui buvo naudotas dotMemory.



Stulpelyje Total / Own ... rodoma, kiek atminties (objektų) buvo skirta tam tikrame skambučių medyje (total) arba konkrečiame skambutyje (own).

Laiko juostos diagramoje parodyta, kaip programa naudojo atmintį profiliavimo proceso metu.



Heap Gen0, Gen1 ir Gen2:

Gen0: dedami naujai paskirti objektai. Kai įvyksta šiukšlių surinkimas, tai apima tik Gen0. Objektai, išgyvenantys po Gen0 šiukšlių surinkimo, pakeliami į Gen1.

Gen1: objektai, kurie išgyveno vieną ar daugiau šiukšlių surinkimo Gen0, pakeliami į Gen1. Gen1 šiukšlių surinkimas vyksta rečiau, tačiau užtrunka daugiau laiko, palyginti su Gen0.

Gen2: objektai, kurie išgyveno daugybę šiukšlių surinkimo Gen1, pakeliami į Gen2.

LOH: yra atskira valdomos heap sritis, skirta dideliems objektams, paprastai tiems, kurių dydis didesnis nei 85 000 baitų. LOH turi savo šiukšlių surinkimo procesą.

POH: pinned object heap naudojama objektams, kurie buvo aiškiai „prisegti“ atmintyje, naudojant GCHandle klasę, saugoti. Prisegimas neleidžia šiukšlių rinkėjui perkelti objekto atmintyje.

GC laikas: Tai rodo laiką, praleistą vykdant šiukšlių surinkimą vykdant programą. Tai apima laiką, praleistą Gen0, Gen1, Gen2 ir LOH.

4. Apkrovos testavimas

Apkrovos testavimui buvo naudotas K6.

```

checks.....: 100.00% ✓ 266189      × 0
data_received.....: 1.6 GB   5.2 MB/s
data_sent.....: 21 MB   71 kB/s
http_req_blocked.....: avg=5.82µs  min=0s  med=0s      max=7ms    p(90)=0s    p(95)=0s
http_req_connecting.....: avg=1.67µs  min=0s  med=0s      max=2.04ms  p(90)=0s    p(95)=0s
✓ http_req_duration.....: avg=1.67ms  min=0s  med=520.59µs max=370.45ms p(90)=1ms   p(95)=1ms
  { expected_response:true }.....: avg=1.67ms  min=0s  med=520.59µs max=370.45ms p(90)=1ms   p(95)=1ms
✓ http_req_failed.....: 0.00% ✓ 0      × 266189
http_req_receiving.....: avg=57.38µs min=0s  med=0s      max=24.53ms p(90)=0s    p(95)=532.1µs
http_req_sending.....: avg=17.22µs min=0s  med=0s      max=17.51ms p(90)=0s    p(95)=0s
http_req_tls_handshaking.....: avg=0s      min=0s  med=0s      max=0s      p(90)=0s    p(95)=0s
http_req_waiting.....: avg=1.6ms   min=0s  med=0s      max=370.45ms p(90)=1ms   p(95)=1ms
http_reqs.....: 266189  884.289395/s
iteration_duration.....: avg=1.01s   min=1s  med=1.01s   max=1.38s   p(90)=1.02s p(95)=1.04s
iterations.....: 266189  884.289395/s
vus.....: 12      min=12      max=1000
vus_max.....: 1000   min=1000    max=1000

running (5m01.0s), 0000/1000 VUs, 266189 complete and 0 interrupted iterations
default ✓ [=====] 0000/1000 VUs  5m0s
PS C:\Users\6abija\Downloads\Tic-Tac-Toe-SignalR-expand-board1\Tic-Tac-Toe-SignalR-expand-board\TicTacToe>

```

Checks: 100,00% ✓ 266189 ✗ 0: Nurodo patikrinimų rezultatus testo metu. Visi 266 189 patikrinimai buvo sėkmingi (✓), o gedimų nebuvo (✗ 0). Patikrinimas gali būti konkrečios HTTP atsakymo sąlygos patvirtinimas.

Data_received: Gauti duomenys: 1,6 GB 5,2 MB/s: bandymo metu gautų duomenų kiekis. Tai yra 1,6 GB, o vidutinis greitis yra 5,2 MB/s.

Data_sent: 21 MB 71 kB/s: bandymo metu išsiųstų duomenų kiekis. Tai yra 21 MB, o vidutinis greitis yra 71 kB/s.

http_req_blocked: avg=5.82μs min=0s med=0s max=7ms p(90)=0s p(95)=0s: matuoja laiką, kurį HTTP užklausa praleido laukiant išsiuntimo eilėje.

http_req_connecting: vid. = 1,67 μ s min = 0 s vidutin = 0 s max = 2,04 ms p(90) = 0 s p(95) = 0 s: matuoja laiką, praleistą užmezgant ryši.

http_req_duration: vid. = 1,67 ms min = 0 s vidutin = 520,59 μ s maks. = 370,45 ms p(90) = 1 ms
p(95) = 1 ms:

avg: vidutinė HTTP užklauso trukmė.

min: Minimali pastebēta trukmē.

med: Vidutinė (50 procentilių) trukmė.

max: didžiausia stebima trukmė.

p(90): 90 procentilio trukmė (90 % užklausų buvo greitesnės už šią).

p(95): 95 procentilio trukmė (95 % užklausių buvo greitesnės už šią).

http_req_failed: 0,00% ✓ 0 ✗ 266189: nepavykusių HTTP užklausų procentas. 0 % nepavyko (✓ 0), o visos 266 189 užklauskos buvo sėkmingos.

http_req_receiving: avg=57.38μs min=0s med=0s max=24.53ms p(90)=0s p(95)=532.1μs: matuojamas atsakymo gavimo laikas.

http_req_sending: avg=17.22μs min=0s med=0s max=17.51ms p(90)=0s p(95)=0s: Matuoja užklauskos siuntimo laiką.

http_req_tls_handshaking: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s: Matuoja laiką, praleistą atliekant TLS handshaking.

http_req_waiting: avg=1.6ms min=0s med=0s max=370.45ms p(90)=1ms p(95)=1ms: matuoja laiką, praleistą laukiant, kol serveris pradės siųsti duomenis.

http_reqs: 266189 884.289395/s: bendras HTTP užklausų skaičius bandymo metu ir vidutinis užklausų skaičius per sekundę.

iteration_duration: avg=1.01s min=1s med=1.01s max=1.38s p(90)=1.02s p(95)=1.04s: Matuoja vienos iteracijos trukmę.

iterations: 266189 884.289395/s: bendras per bandymą atliktų iteracijų skaičius ir vidutinis iteracijų skaičius per sekundę.

vus: 12 min = 12 max = 1000: šiuo metu aktyvių virtualių naudotojų (TPB) skaičius svyruoja nuo mažiausiai 12 iki daugiausiai 1000 bandymo metu.

vus_max: 1000 min = 1000 max = 1000: maksimalus virtualių naudotojų, sukonfigūruotų testui, skaičius, nurodantis didžiausią bandymo metu naudotojų skaičių.

Bandymo eiga:

running (5m01.0s), 0000/1000 TPB, 266189 užbaigtų ir 0 pertrauktų iteracijų: pateikia informaciją apie dabartinę testo būseną, įskaitant praėjusį laiką, atliktų iteracijų skaičių ir testo eigą.

default ✓ [=====] 0000/1000 TPB 5m0s: rodo eigos juosta, rodanti testo baigimo būseną. Jis veikė 5 minutes ir visi 1000 VU buvo užbaigti.