

Load Generator Performance

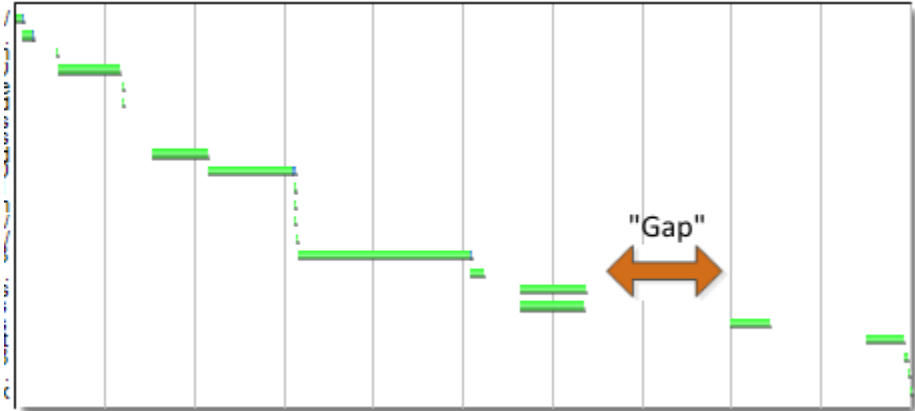
An essential element of the controller or agent is the load generator that handles request /response processing. It needs a sufficient number of .NET threads to timely send and process HTTP messages.

The system creates the necessary number of threads automatically, depending on the test complexity, the number of VUs and the test machine's capacity. The actual number of threads allocated to StresStimulus process is controlled by the operating system, and depends on the load generator's needs to issue requests.

By default, before the test begins, StresStimulus allocates the number of .NET threads equal to the maximum number of VUs expected in this test.

During test execution, one VU may consume more than one thread to handle parallel requests. If StresStimulus needs more resources, the OS will gradually increase threads to the sufficient level. On some systems, however, OS fails to ramp-up threads fast enough. This may result in delays in issuing requests after receiving a response at some moments during the test execution. You can check if such delays exist in your system by analyzing a page or transaction waterfall chart.

If you see a gap between the moment of receiving a response and issuing a subsequent request, your system may need more threads. To address the situation, you can manually adjust the initial number of threads, which is a property of a load agent.



Note: The gap between a response and a subsequent request can be caused by different reasons. For example, StresStimulus intentionally delays subsequent request to comply with the emulated browser characteristics. Each browser type has a limited number of connections-per-host and connections-per-proxy. When the number of connection exceeds this limit, StresStimulus will pause requests to realistically mimic a physical browser behavior.

If you need to ramp up threads faster, which may be necessary when running large-scale tests on high performance hardware, set the **Minimum starting threads** property to the level 120% -150% of the number of virtual users. That should boost the performance of your load generator. After that, rerun your test to check if such an increase was sufficient. You might need to try several different numbers until your load generator receives sufficient resources. This property is configured for each load agent individually. Note that setting a too high number of threads may leave insufficient resources for the OS and can slow down your computer.

The screenshot shows the LoadRunner interface with the 'Load Agents' section selected in the left-hand menu. The main window displays the configuration for the 'POWEREDGE' agent. The 'Minimum starting threads' property is highlighted in blue and set to the value '50'. Below the table, there is a section titled 'Minimum starting threads' with a descriptive note.

Agent Name	POWEREDGE
StresStimulus version	4.5.5956
Host / IP Address	10.0.2.121
Mix Weight	1
Allocate at least one thread per VU	Yes
Minimum starting threads	50

Minimum starting threads
To increase the Starting thread count, set this property to a higher number. This property will be disregarded if it is smaller than the starting thread count determined as explained in the previous property.

e
V
U
i
s
n
o
t
e
q
u
a
l
t
o
o
n
e
t
h
r
e
a
d
.
T
h
r
e
a
d
s
a
v
a
i
l
a
b
l
e
t
o
t
h
e
p
r
o
c
e
s
s
(
t
h
r
e
a
d
p
o
o
l
)
c
a
n
b
e
u
s
e
d
b
y
d
i
f
f
e
r
e
n
t
V
U
s

a
t
d
i
f
f
e
r
e
n
t
t
i
m
e
s.
E
a
c
h
V
U
c
a
n
i
s
s
u
e
m
u
l
t
i
p
l
e
c
o
n
c
u
r
r
e
n
t
r
e
q
u
e
s
t
s
(c
o
n
n
e
c
t
i
o
n
s)
,
d
e
p
e
n
d
i
n
g
o
n

y
o
u
r
t
e
s
t
c
a
s
e
a
n
d
t
h
e
c
o
n
n
e
c
t
i
o
n
l
i
m
i
t
o
f
t
h
e
s
e
l
e
c
t
e
d
b
r
o
w
s
e
r.
F
o
r
e
x
a
m
p
l
e
,
I
E
9
c
a
n
o
p

e
n
u
p
t
o
6
c
o
n
n
e
c
t
i
o
n
s.

Note: In the previous versions, therefore, read allocation algorithm is recommended. NE

The screenshot shows a software interface with a 'Main' window. The 'Workflow Tree' on the left includes options like 'Record Test Case...', 'Build Test Case', 'Test Profile', 'Other Options', 'Load Agents', 'Monitoring', 'Result Storage', 'Pass/Fail Configuration', 'Advanced Options', 'Script Editor', 'Run and Monitor Test...', and 'Analyze Results'. The 'Load Agents' option is selected. The main area displays configuration for 'Local (weight: 1, VUs: 1)'. A table shows the following settings:

Mix Weight	1
Allocate at least one thread per VU	Yes
Minimum starting threads	1
VUs	1

Below the table, there is a section titled 'Allocate at least one thread per VU' with the following text: 'Starting thread count is the number of threads created automatically when the test is launched. By default, it is equal to the number of VUs. If more threads are needed, the load engine will gradually create more threads while checking available system resources. Select No to allow .NET Framework to allocate a default (typically smaller) number of starting threads.'

T
F
r
a
m
e
w
o
r
k
's
d
e
f
a
u
l
t
a
l
g
o
r
i
t
h
m
o
f
a
l
l
o
c
a
t
i
n
g
t
h
r
e
a
d
s.
A
s
a
r
e
s
u
l
t,
t
h
e
s
t
a
r
t
i
n
g
t
h
r
e
e
a
d
c
o
u
n
t

w
a
s
t
y
p
i
c
a
l
l
y
s
m
a
l
l
e
r
t
h
a
n
t
h
e
n
u
m
b
e
r
o
f
v
i
r
t
u
a
l
u
s
e
r
s
. F
o
r
b
a
c
k
w
a
r
d
c
o
m
p
a
t
i
b
i
l
i
t
y,
t
h
e
o
p
t
i
o
n
t
o
u
s
e
t

h
e
l
e
g
a
c
y
t
h
r
e
a
d
a
l
l
o
c
a
t
i
o
n
a
l
g
o
r
i
t
h
m
e
x
i
s
t
.
T
o
u
s
e
i
t,
s
e
t
p
r
o
p
e
r
t
y
A
l
l
o
c
a
t
e
a
t
l
e
a
s
t
o
n
e
t
h
r
e
a
d
p
e
r
V
U
t
o
N
o
.

Typically the load engine is optimized for performance to handle the more virtual users on a single load generator. However, if your test has many resource requests that have a small expected response time want to change the load engine optimization for "Time Resolution". This will enable measuring response times under 0.1 second with more precision, however, will have a performance impact with more virtual users.

