

# CocoonPerformanceResults%Scenario1

As you will see below, Cocoon's response times scale linear with the number of users hitting the server (assumed that the requested pages are already in the cache). The average response times rise from 140ms (for 40 users) over 280ms (80 users) to 560ms (at 160 concurrent users). This is far beyond the limit a machine should be targeted at in a production environment, but it shows very well that there is no breakdown even under high load.

"Siege" output for 40 concurrent users (10 different URLs, 2 hour run)

```
Transactions:          2077162 hits\\
Availability:           100.00 %\\
Elapsed time:          7200.46 secs\\
Data transferred:      703126566 bytes\\
Response time:         0.14 secs\\
Transaction rate:      288.48 trans/sec\\
Throughput:            97650.23 bytes/sec\\
Concurrency:           39.93\\
Successful transactions: 2077162\\
Failed transactions:    0\\
\\
2066337      pages delivered in interval 0 - 1 s \\
4869      pages delivered in interval 1 - 2 s \\
109      pages delivered in interval 2 - 3 s \\
34      pages delivered in interval 3 - 4 s \\
5      pages delivered in interval 5 - 6 s \\
0      pages delivered in interval 6 - 10 s \\
0      pages delivered in interval 11 - 20 s \\
0      pages delivered in interval 21 - 30 s \\
0      pages delivered in interval 31 - 180 s \\
0      pages delivered > 181 s \\
2071354      pages total delivered\\
\\
~99,76 % < 1 sec.
```

"Siege" output for 80 concurrent users (10 different URLs, 2 hour run)

```
Transactions:          2060632 hits\\
Availability:           100.00 %\\
Elapsed time:          7200.42 secs\\
Data transferred:      663368765 bytes\\
Response time:         0.28 secs\\
Transaction rate:      286.18 trans/sec\\
Throughput:            92129.18 bytes/sec\\
Concurrency:           79.91\\
Successful transactions: 2060632\\
Failed transactions:    0\\
\\
1963380      pages delivered in interval 0 - 1 s \\
85413      pages delivered in interval 1 - 2 s \\
4097      pages delivered in interval 2 - 3 s \\
432      pages delivered in interval 3 - 4 s \\
99      pages delivered in interval 5 - 6 s \\
17      pages delivered in interval 6 - 10 s \\
1      pages delivered in interval 11 - 20 s \\
0      pages delivered in interval 21 - 30 s \\
0      pages delivered in interval 31 - 180 s \\
0      pages delivered > 181 s \\
2053439      pages total delivered\\
\\
~95,61 % < 1 sec.
```

"Siege" output for 160 concurrent users (10 different URLs, 2 hour run)

```

Transactions:          2061097 hits\\
Availability:          100.00 %\\
Elapsed time:          7199.85 secs\\
Data transferred:      664532442 bytes\\
Response time:         0.56 secs\\
Transaction rate:      286.27 trans/sec\\
Throughput:            92298.09 bytes/sec\\
Concurrency:           159.80\\
Successful transactions: 2061097\\
Failed transactions:    0\\
\\
1684748      pages delivered in interval 0 - 1 s \\
262224       pages delivered in interval 1 - 2 s \\
81564        pages delivered in interval 2 - 3 s \\
18337        pages delivered in interval 3 - 4 s \\
4184         pages delivered in interval 5 - 6 s \\
3043         pages delivered in interval 6 - 10 s \\
69           pages delivered in interval 11 - 20 s \\
0            pages delivered in interval 21 - 30 s \\
0            pages delivered in interval 31 - 180 s \\
0            pages delivered > 181 s \\
2054169      pages total delivered\\
\\
~82,01 & < 1 sec.

```

If you are planning to use this tool yourself you may find the attached `awk` script "siege-auswertung-awk" useful. This script generates the "... pages delivered in interval ..." messages. It is invoked in the following way:

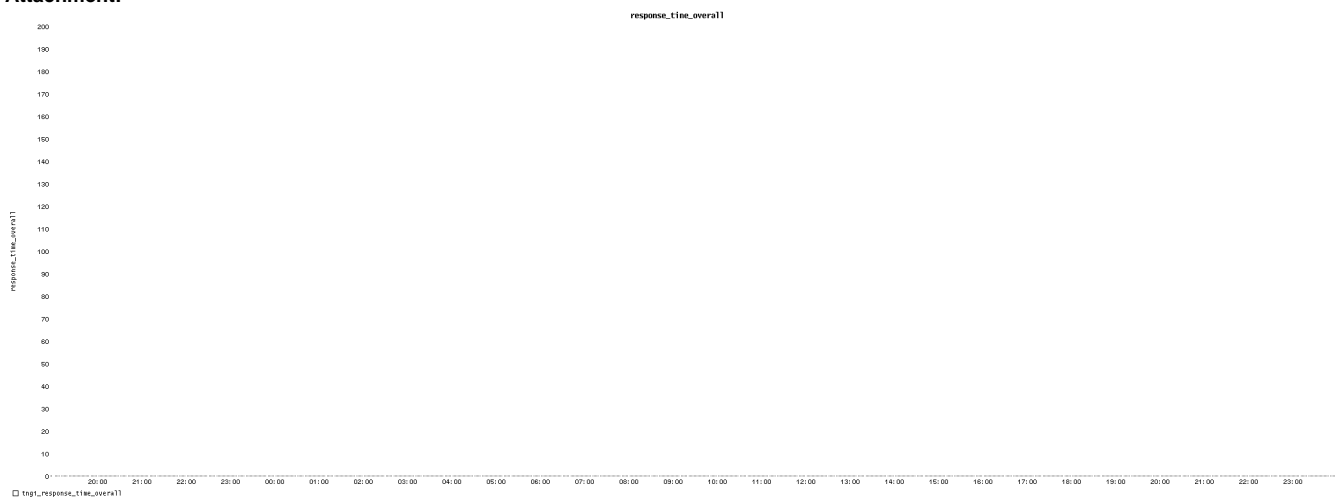
```

/usr/local/bin/siege > /tmp/siege-test.out
cat /tmp/siege-test.out |grep -v HTTP
cat /tmp/siege-test.out | grep HTTP | /usr/bin/awk -f /usr/local/etc/siege-auswertung-awk

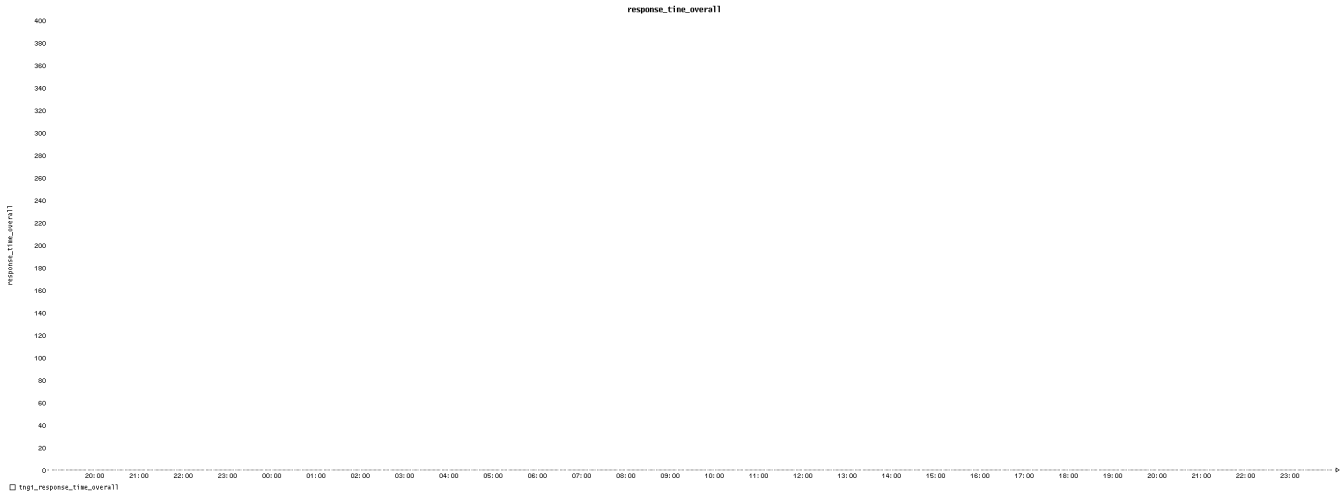
```

**Note:** The sharp bend in the curves below is not an error, it was caused because I ran out of disk space for the log files 😞

#### Attachment:



Attachment:



Attachment: [sieve-auswertung-awk](#)