Capacity Planning

[bana], [another team member] *17.05.2010*

System under test

- XAMPP
  - Apache 2.2.14
  - MySQL 5.1.41
  - PHP 5.3.1
  - Tomcat v6.0
- Server-Specifications
  - Windows 7
  - Intel Core2 Duo 2.2GHz
  - 4GB RAM
  - 510 Mflops/sec
- Client Specification
  - Windows XP
  - Intel Celeron M 1.7GHz
  - 1GB RAM
- Network
  - 54Mbps WLAN

Component under study

- Blog implemented in PHP
  - Serendipity, Open-Source Project
  - Added a view entries with lots of text and images
- Own implemented Project
  - Gets the HTML Code of a given web server
  - Made tests with fast performing sites like google.com and large sites like an entry in wikipedia

Measurement Techniques

Software Monitoring

- Microsoft Web Application Stress Test 1.1
- Own implemented program (C#) for stress test and diagrams
  - Sends a request to the server, retrieves the response and calculates the elapsed time

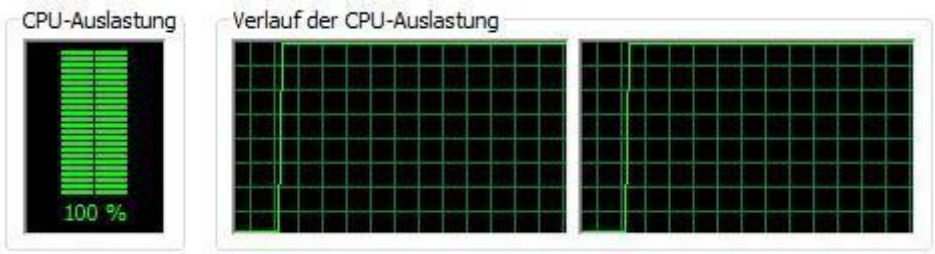
Type of workload, Quantitative Characteristics

- Real workload
  - Low amount of simultaneous connections
  - < 1 request / second
- Synthetic workload
  - High amount of requests, similar to a Denial of Service Attack
  - > 20 request / second
- Time Interval
  - 2 Minutes
  - Optional request delay of 800ms to 2000ms

Evaluation of the System

Blog Software

We used 10 Threads and 30 Sockets per Thread = 300 Connections to the Web server

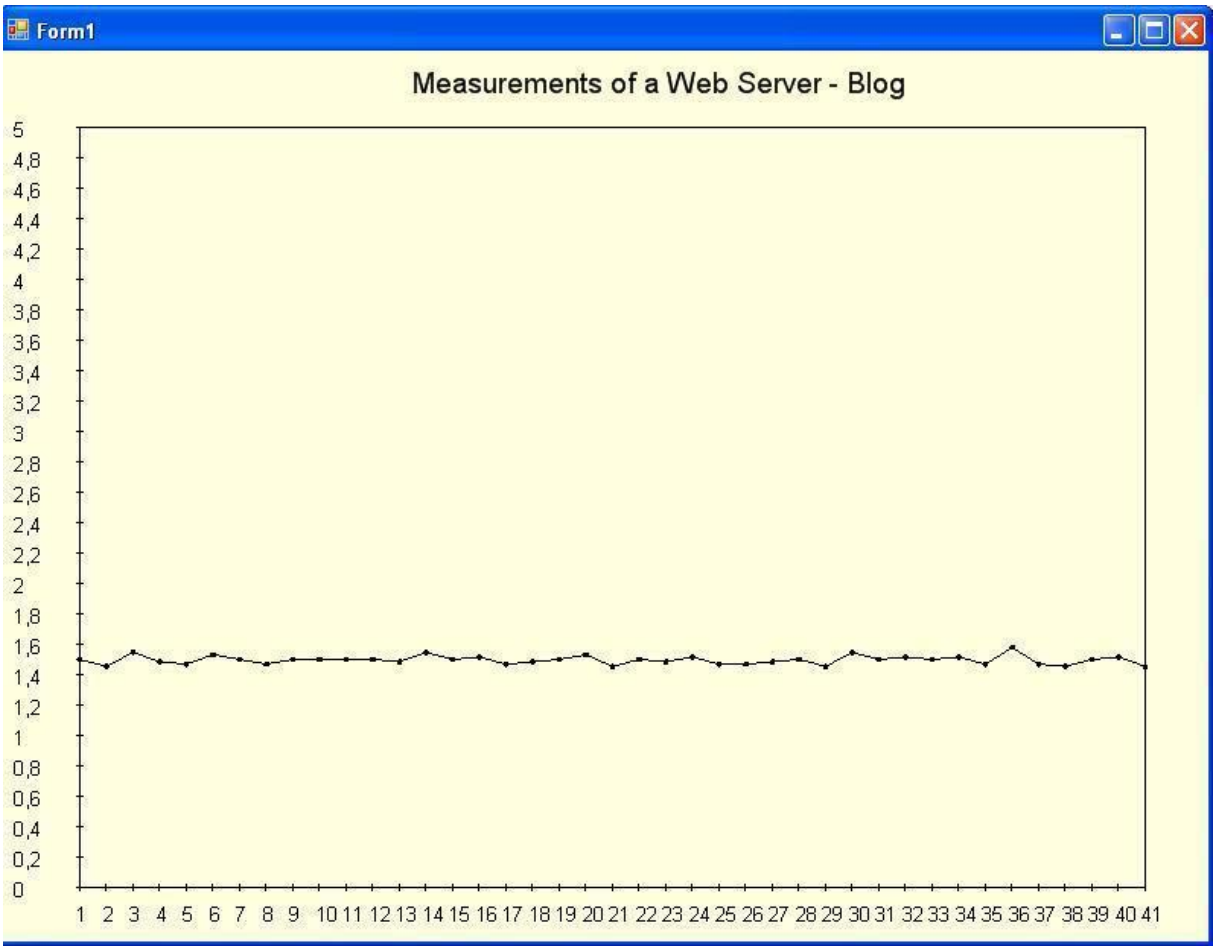


Number of Hits:	143
Requests per Second:	1.2
Avg Time to First Byte:	19081
Avg Time to Last Byte:	19589
Bytes Sent Rate :	102kB/sec
Bytes Received Rate:	362kB/sec
Total Bytes received:	43508
Throughput:	$\frac{\text{Total Requests} * \text{Avg.Size}}{\text{Period of Time}}$ = 360kB/sec

The blog contains a lot of very long entries. Therefore, the connection to the database and the hard disk must have a high throughput rate so that the system is fast.

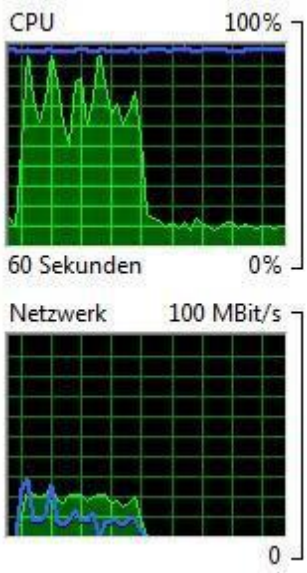
We had been testing the system until we reached a maximum of 1.2 requests per second. The CPU load was at 100% the whole time, so we defined the CPU as the limiting factor in the system. Additionally, the hard disk had a high load – so if the CPU would be replaced by a more efficient one, the HD would be the new limiting factor in the system.

The network connection can handle the requests easily. Considering the Avg TTFB and TTLB, it can be noticed that the server takes a long time to send the response. The response itself is very small and needs little time to transmit.



Java Servlet

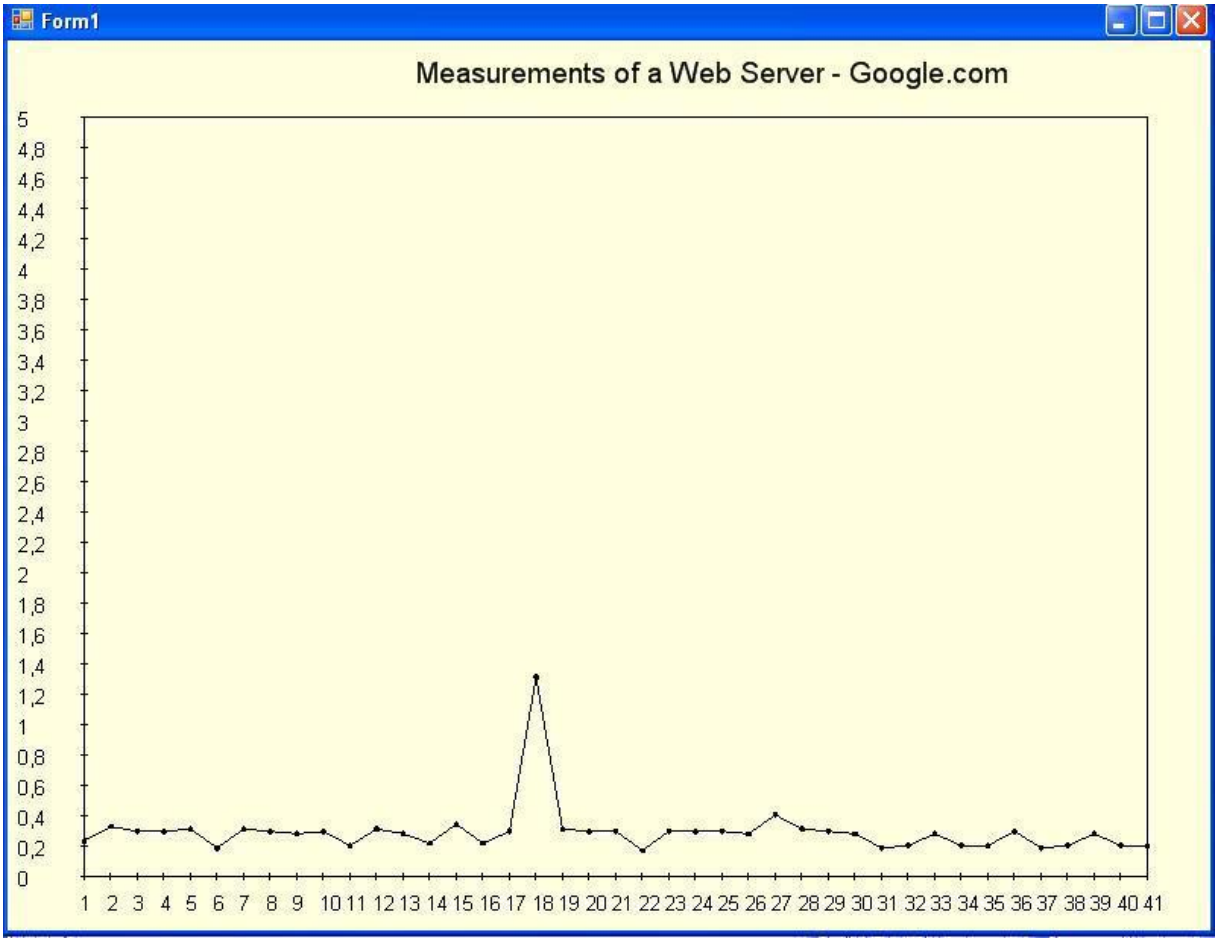
The Web server retrieves the content of the given URL; therefore we used two different URLs with different “content size”.



a. www.google.com

This one is very small and contains little text (HTML Code). We used 500 simultaneous connections.

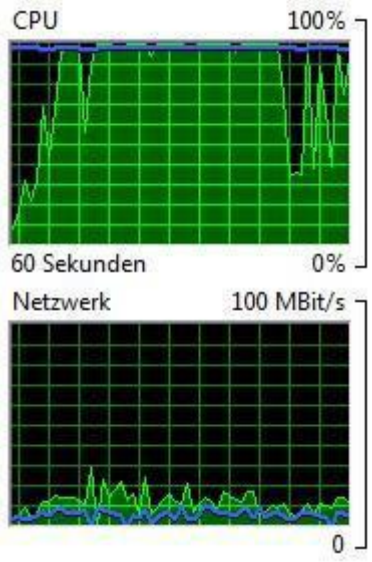
Number of Hits:	3920
Requests per Second:	64.2
Avg Time to First Byte:	612
Avg Time to Last Byte:	1210
Bytes Sent Rate :	47kB/sec
Bytes Received Rate:	632kB/sec



The CPU load never reached 100% even if we used 500 simultaneous connections. The maximum amount of requests per second was found to be 64. Considering the average TTFB and TTLB, one can see that the response from the server takes approximately the same time as the calculation on the server.

In this case study, the bandwidth of the network connection can be defined as the limiting factor for system performance.

b. de.wikipedia.org/Domain_Name_System



The Domain Name System Site contains a lot of text and should be less performing than google.com.

Number of Hits:	328
Requests per Second:	2.73

Avg Time to First Byte:	44359
Avg Time to Last Byte:	69433
Bytes Sent Rate :	3.9kB/sec
Bytes Received Rate:	306kB/sec
Total Bytes received:	36788
Throughput:	$\frac{\text{Total Requests} * \text{Avg.Size}}{\text{Period of Time}}$ = 306.6kB/sec

In this scenario, we queried the server for a long article on Wikipedia. The server needs more time to get the HTML Code and also needs more CPU power to process the request.

The CPU load is at its maximum most of the time, which lets us conclude that the CPU performance is the limiting factor in the system – closely followed by the network bandwidth. This means that an increase in CPU efficiency would only provide a few percent of higher overall performance. If an significant increase is needed, the network connection has to be improved also.

