CAPACITY PLANNING

[bana], [another team member]

Index

- System under test
- Components under study
- Measurement techniques
- Type of workload and quantitative characteristics
- Evaluation of the system
  - Blog Software
  - Java Servlet
    - Google.com
    - Wikipedia.org

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

Tools - Source Code

```
c.Title = "Measurements of a Web Server - Blog";
c.Xorigin = 0; c.ScaleX = 40; c.Xdivs = 40;
c.Yorigin = 0; c.ScaleY = 5; c.Ydivs = 25;

for (int i = 0; i <= 40; i++)
{
    hr = (HttpWebRequest)WebRequest.
        Create("http://140.78.166.172/serendipity/'
    hr.Method = "POST";
    startTime = DateTime.Now;
    hr.GetResponse();
    stopTime = DateTime.Now;
    hr.Abort();
    duration = stopTime - startTime;
    c.AddValue(i, (float)duration.TotalSeconds);
    Thread.Sleep(1);
}
pb.Image = c.Draw();
```

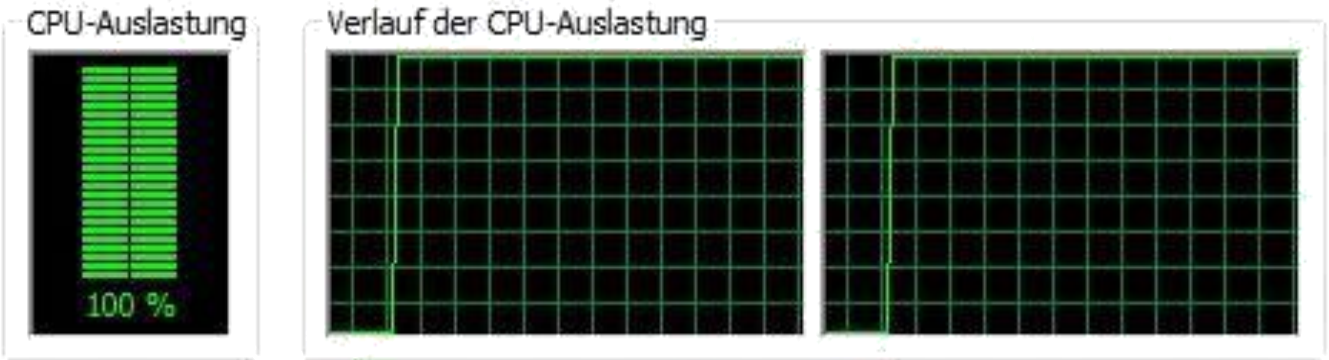
Type of workload | Quantitative Characteristics

- 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

SYSTEM EVALUATION

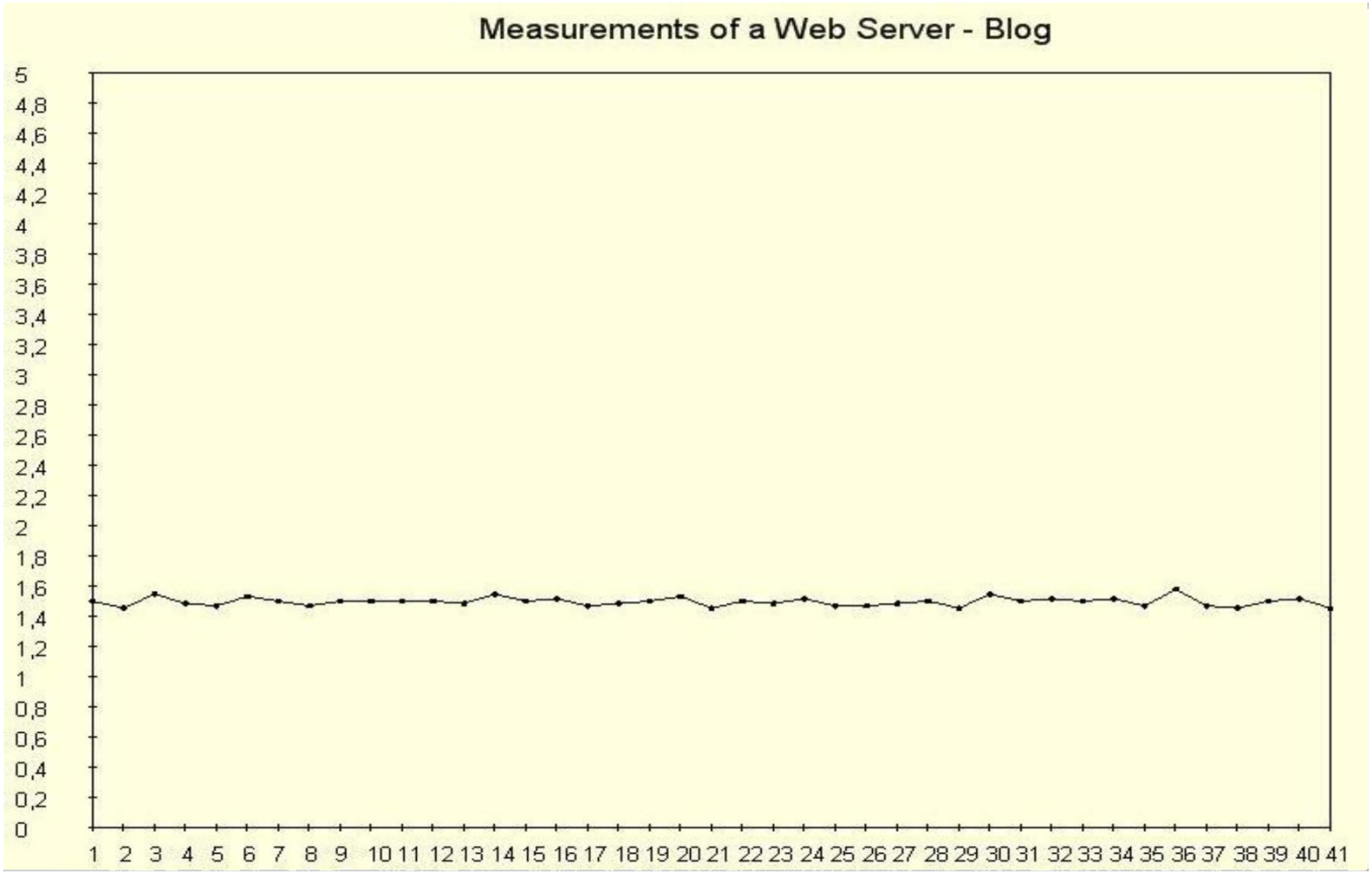
Blog Software

We used 10 Threads and 30 Sockets per Thread
= 300 simultaneous 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

Blog Software



Blog Software

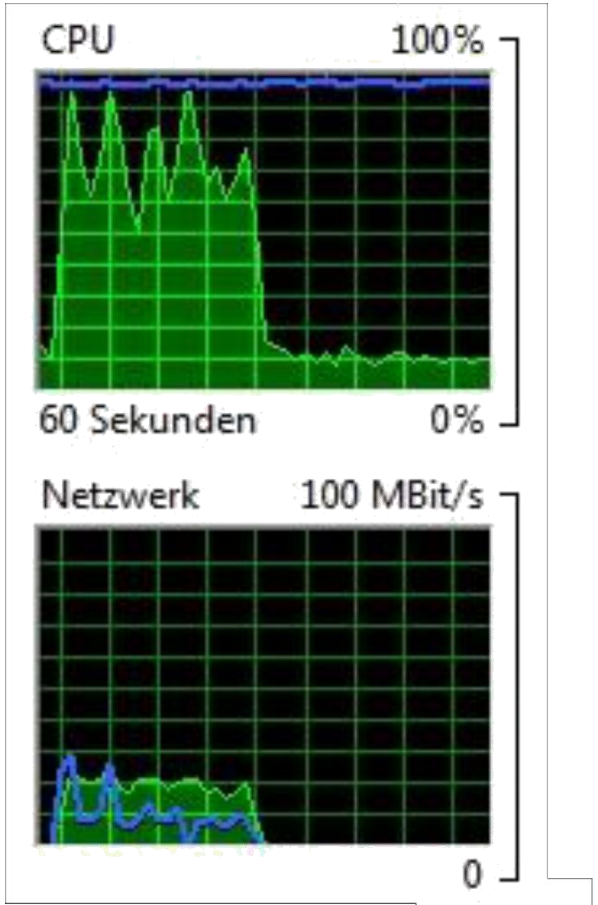
- The blog contains a lot of very long entries.
  - High demands on database connection and the hard disk
- System-Maximum was reached by 1.2 requests per second
- Limiting Factor in the System:
  - CPU
- Improvements:
  - If the CPU would be replaced by a more efficient one, the HD would be the new limiting factor in the system.
- Network connection can handle requests easily.

Java Servlet

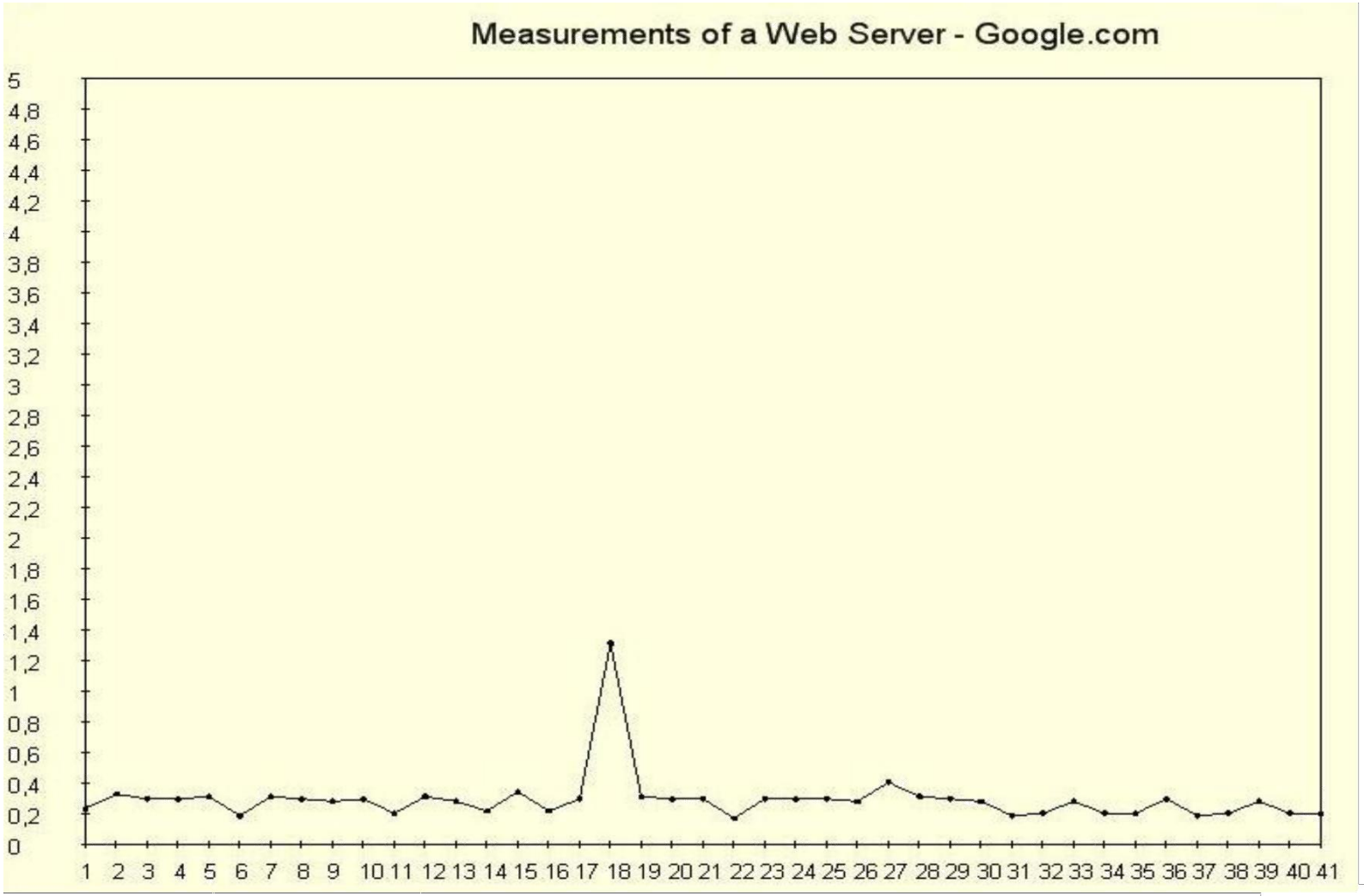
The Web server retrieves the content of the given URL; therefore we used two different URLs with different “content size”.

www.google.com

- 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



Java Servlet



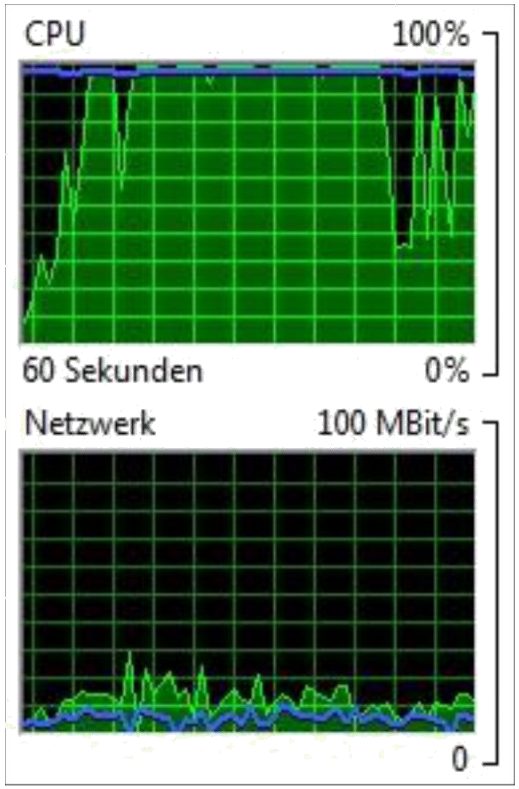
Java Servlet

- CPU load never reached 100% even if we used 500 simultaneous connections.
- System-Maximum:
  - 64 Requests per second
- Limiting Factor
  - Bandwidth of the network connection
- Improvements
  - Caching would reduce network requirements on the server side

Java Servlet

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

de.wikipedia.org/Domain_Name_System

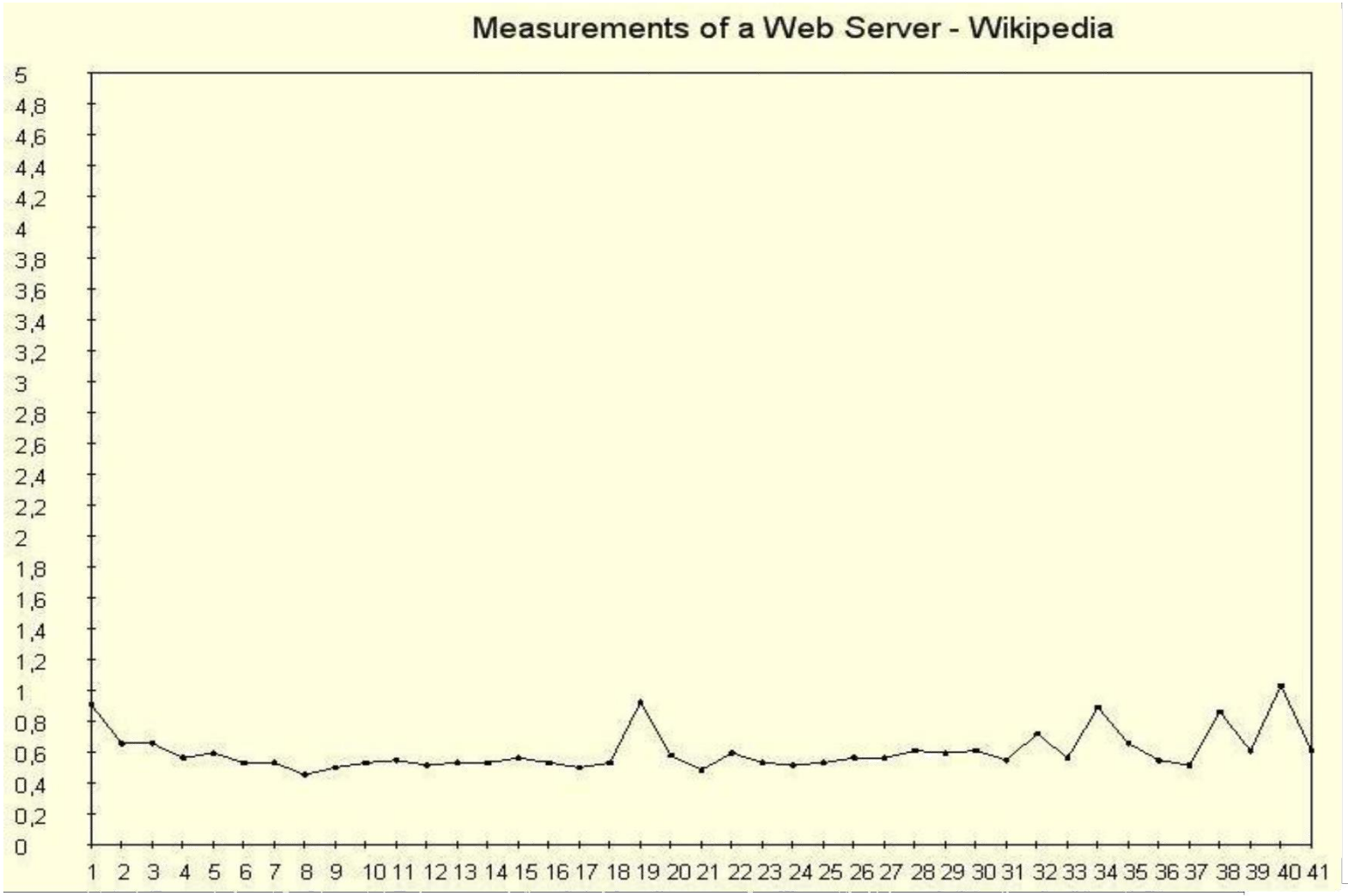


- Number of Hits: 328
- Requests per Second: 2.73
- Avg. Time to First Byte: 44359
- Avg. Time to Last Byte: 69433
- Bytes Sent Rate : 3.9kB/sec
- Bytes Received Rate: 306kB/sec
- Total Bytes received: 36788
- Throughput:

$$\frac{\text{Total Requests} * \text{Avg. Size}}{\text{Period of Time}}$$

= 306.6kB/sec

Java Servlet



Java Servlet

- Queried the server for a long article on Wikipedia.
- More demand on bandwidth and CPU
- CPU load is at its maximum most of the time
- Limiting Factor
  - CPU
  - Network bandwidth
- Improvements
  - Increasing CPU efficiency would only provide a higher overall performance if also the bandwidth gets altered

THANK YOU FOR YOUR ATTENTION

Some tips …


Google Tut uns leid...
Es tut uns leid, aber...
... Ihr Computer oder Netzwerk sendet eventuell automatische Anfragen. Um unsere Nutzer zu schützen, können wir Ihre Anfrage derzeit nicht verarbeiten.
Geben Sie die folgenden Zeichen ein, um mit der Suche fortzufahren:
Ich bin ein Mensch.
Weitere Informationen erhalten Sie in der Google-Hilfe.
© 2010 Google - Google-Startseite