7.4.2 Server Side Scripting und Browser Side Scripting

Grundsätzlich gibt es zwei Möglichkeiten wie und wo bei remote Anwendungen der Prozess abläuft. Jedes System hat so seine Vorzüge, was eventuell sogar eine Kombination lohnenswert macht.
Programme und Scripts die auf dem Server (via WebServer) ablaufen haben den Vorteil, dass alle Remote-Anwendungen auf die selben Recourcen zugreifen können. So z.B. Datenbank mit Passworten und Lagerbeständen usw.... Der Nachteil liegt aber darin, dass sich alle Remote-Anwender die CPU-Leistung des Servers aufteilen.
Beim Client-Side scripting haben wir den Vorteil, dass die Prozesse bei den Kunden laufen somit keine grosse Zentrale CPU-Leistung notwendig ist. Jedoch müssen wir den Nachteil in Kauf nehmen, dass wir keinen direkten Zugriff auf die Festplatte und die Recourcen des Servers haben.

7.4.2.2Webapplikationen mit PHP 4 Info

PHP als Open-Source-Projekt stellt für Webdesigner und Entwickler eine attraktive Basis zur dynamischen Datenanbindung dar. PHP ist Free Software — zusammen mit Linux (oder auch einer vorhandenen Windows-Plattform, z.B. IIS), dem Apache Webserver und der Datenbank MySQL ergibt sich eine komplette Web-Applikationsplattform zum Nulltarif. Nach den neuesten Statistiken wird der PHP Hypertext Preprocessor auf über zwei Millionen Domains eingesetzt.

Technisch gesehen handelt es sich bei PHP um einen Präprozessor mit integrierter Skriptsprache. PHP ist deshalb unter jedem Webserver einfach wie ein CGI-Skript einsetzbar. Für den Apache Webserver gibt es allerdings ein eigenes PHP-Modul. Dadurch läuft PHP im gleichen Prozessraum wie der Webserver und muss nicht bei jedem Aufruf neu gestartet werden. Schon heute ist PHP das am häufigsten installierte Apache-Modul.

PHP-Code ist innerhalb von Webseiten durch die Tags <?php und ?> markiert. Zur Programmierung bietet PHP eine komplett eigene Skriptsprache. PHP-Programmcode hat große Ähnlichkeit zu C: die Syntax ist — von wenigen Ausnahmen abgesehen — weitgehend identisch. Variablen werden allerdings wie bei Perl mit einem vorausgehenden Dollarzeichen markiert.

Unix- und JavaScript-Programmierer kommen mit PHP nach einer kurzen Startphase gut zurecht. PHP4 basiert auf einer komplett neu programmierten Skripting Engine namens Zend (www.zend.com), die nicht nur gegenüber PHP3 die Performance erheblich verbessert, sondern allgemein für eine bessere Abstraktion zwischen Webserver und PHP sorgt. Wie bei Cold WebEditor ist es mit PHP4 möglich, auf beliebige Java-Objekte zuzugreifen.

Ein Großteil der PHP-Funktionalität wird über Module definiert. Durch die grosse Dynamik der freien Entwickler-gemeinde existiert bereits eine ansehnliche Sammlung von PHP-Modulen für die verschiedensten Aufgabengebiete. Darunter befinden sich neben PDF- und Grafik-Generierung auch Module zum POP-, IMAPoder LDAP-Zugriff und ein Modul zum Parsen von XML-Daten.

Der Datenbankzugriff konzentriert sich aufgrund der LAMP-Architektur bei PHP stark auf die mitgelieferten MySQL-Treiber. Auch für viele andere unter Unix verbreiteten Datenbanken wie zum Beispiel Interbase, Informix, Oracle oder Sybase liefert PHP funktionierende Treiber mit. Und mit Namo WebEditor 5 und GoLive 6 können Sie Webdesign und dynamische Datenanbindungen unter einer Oberfläche vornehmen.

7.4.2.3 Einige PHP Beispiele

PHP Trace Route http://www.clinch.ch/Net-Work/internet/traceroute.htm

```
<html>
<head>
<title>Ping und Trace</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<META name="author" content="Manuel Magnin">
<META name="date" content="15.12.2001">
</head>

<body BGCOLOR="#FFFFFF">

<p><img src="http://www.clinch.ch/Images/systemcl.gif" height="50" width="550">
<font SIZE="1"> <br></font></p>

<?php
 /*_____________________________________________________________________________________________________

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%                                                                                                  %%
%%  Programm:   Tracert und Ping mit PHP                                                           %%
%%  Datum:      19.03.2002                                                                          %%
%%  Autor:      Manuel Magnin                                                                       %%
%%  Version:    1.0                                                                                 %%
%%                                                                                                  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

<?php
 //--------------------Variablen deklaration und Konfigurationsbereich ---------------------------------
 global $ip;                                          // IP-Adrasse
 $ip = getenv("REMOTE_ADDR");                         // IP-Adrasse erfragen (eigene falls keine übergeben wurde)
 global $pinge;                                       // Array mit diversen Latenzzeiten
 global $pingschnitt;                                 // Durchschnitt der Latezzeit
 $pingmax = 0;                                        // Standardvariable für Maximumwert
 $pingmin = 9000;                                     // Standardvariable für Minimumwert
 global $tracert;                                     // Variable für tracert
 $tracecount = 0;                                     // Hilfsvariable für tracert
 $swapdir = "C:/inetpub/wwwroot/PingTrace/swap/";     // SWAP MUSS IM WEB-SERVER BASISORDNER VORHANDEN SEIN
 $serverroot = "C:/inetpub/wwwroot/PingTrace/";       // ZEIGT DAS WEB-SERVER BASIS VERZEICHNIS

reset($HTTP_POST_VARS);                               // Varibeln übergabe in der URL /PingTrace/IP=127.0.0.1
$IP = (current($HTTP_POST_VARS));                     // 1. Varibele abholen
//next($HTTP_POST_VARS);
 if ($IP) {                                           // Varibele gefüllt ?
     $ip = $IP; }                                     // Var Übergabe OK
 else {
    $ip = getenv("REMOTE_ADDR");}                     // Keine übergabe nimm remote

 pingen();                                            //*** Anpingen des Clients ***

 if($pingmax != " Das ICMP Protokoll wird nicht erkannt")  //*** Route des Clients ***
   {tracert();}                                       // Ruft die Funktion tracert auf wenn der Ping erfolgreich war
 else
   {$tracert[0] = " Das ICMP Protokoll wird nicht erkannt";}

 chdir($serverroot);                                  // Wechselt Verzeichnis zum speichern
 if(file_exists($swapdir)) {
          chdir($swapdir);                            // Wenn Verzeichnis swap existiert wird in den Ordner swap gewechselt
          if(file_exists($ip))
          { speichern(); } }                          // Ruft die Funktion Speichern auf wenn der Kunde die Daten per E-Mail will

?>

<!-- &&& Weiter gehts mit HTML &&& -->

<b>Ping und Trace Testauswertung       

Host:    <? global $ip; echo $ip ?> </B><BR>
 <BR>

<B>Latenzzeit in Millisekunden aus 8 Pr&uuml;fungen</B><BR>

<table BORDER="0" CELLPADDING="0" CELLSPACEING="1">
  <tr>
    <td width="100">Durchschnitt: </td>
    <td width="100">Maximum: </td>
    <td width="100">Minimum:</td>
  </tr>
  <tr>
    <td><B><? global $pingschnitt; echo $pingschnitt ?> ms</B></td>
    <td><? global $pingmax; echo $pingmax ?> ms</td>
    <td><? global $pingmin; echo $pingmin ?> ms</td>
  </tr>
  <tr>
    <td> </td>
    <td> </td>
    <td> </td>
  </tr>
</Table>

<B>Routenverfolgung:</B>

<table BORDER="0" CELLPADDING="0" CELLSPACEING="1">
<?
// Route ausgeben
global $tracert;
```

```
$zaehler = 1;
for ($i = 1 ; $i < count($tracert); $i++) {
  echo "<tr><td width='200'> ";
  if($i < 10) { echo "0"; }
  echo "$zaehler:    $tracert[$i] </td></tr>";
  $zaehler++; }
?>
</table>

 


<?
 //*** Ping Funktion ***
 function pingen()  {
    global $ip;                                                 // IP-Adresse
    global $pingmax;                                            // Maximumwert von ping
    global $pingmin;                                            // Minimumwert von ping
    global $pingschnitt;                                        // Durchschnittswert von ping
    $pingcount = 0;                                             // Hilfsvariable für den Ping-Durchschnitt
    // exec() ruft die Kommandozeile auf mit dem Pingbefehl
    exec("ping -n 8 $ip > \\temp\\ping$ip");                    // -n = Anzahl -i TTL -w Zeitlimit in ms fuer Rueckmeldung -- Ausgabe
wird ins temp geschrieben
    $pingrueck = file("\\temp\\ping$ip");                       // Die Variable erhätlt die Ausgaben aus dem temp Verzeichnis
    exec("del \\temp\\ping$ip");                                // Verzeichnis wird gelöscht
    //Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128
    for($i = 2; $i < count($pingrueck)-5;$i++)   {
      if(isset($pingrueck[$i]))        {                        // Abfrage ob die Zeile aus der Variable was enthält
          //Antwort von 127.0.0.1: Bytes=32 Zeit<10ms TTL=128
          $ping = strtok($pingrueck[$i]," ");                   // Zerlegt ein Sting beim angegebenen Tennzeichen
          $ping = strtok(" ");                                  // Der Sting muss nur am anfang angegeben werden. (das Trennzeichen
reicht)
          $ping = strtok(" ");
          $ping = strtok(" ");
          $ping = strtok(" ");                                  //  Zeit<10ms
          $ping = ereg_replace("[^0-9]", "", $ping);            // ersetze alles ausser Zahlen
          // 10
          if(!empty($ping)) {
            $pingcount++;                                       // Hilfsvariable wird hinaufgezählt
            $pingschnitt += $ping;                              // Für den durchschnitt
            if($pingmin > $ping) {
            $pingmin = $ping; }                                 //Zuweisung
          if($pingmax < $ping) {
              $pingmax = $ping; }                               //Zuweisung
        }
      }
    }
    if($pingmin == 9000 && $pingmax == 0) {                     // Wenn kein ping ausgeführt wurde
      $pingschnitt = "Host nicht gefunden";
      $pingmax = "Host nicht gefunden";
      $pingmin = "Host nicht gefunden"; }
    else {
      $pingschnitt = $pingschnitt / $pingcount;                 // Der gesamtwert wird durch die Anzahl pings geteitl
      $pingschnitt = round($pingschnitt,2); }                   // Auf zwei Stellen runden
 }
?>


<?
// *** Tracert Funktion ***
function tracert() {
 global $tracert;                                               // Variable für tracert
 global $ip;                                                    // IP-Adresse
 global $tracecount;                                            // Hilfsvariable

 exec("tracert -d $ip > \\temp\\tracert$ip");                   // Ausgabe wird ins temp geschrieben
 $route = file("\\temp\\tracert$ip");                           // Ausgabe wird der Variable zugewiesen
 exec(" del \\temp\\tracert$ip");                               // Anschliessend wird das file gelöscht
 for ($i = 3 ; $i < count($route)-2; $i++) {
   if(isset($route[$i])) {
     if(ereg("Zeit.*", $route[$i])) {                           // Falls der String Zeit gefunden wird
          $tracert[$tracecount] = "Zeitueberschreitung! "; }    // Dann wird der Variable Zeitüberschreitung übergeben
     else {
          $tracert[$tracecount]=substr("$route[$i]",32,50);// String wird ab Position 32 bis 50 ausgegeben
        $tracert[$tracecount] = trim($tracert[$tracecount]);}
     $tracecount++; }                                           // Hilfsvariable wird hinaufgezählt
   }
 }
?>


<?php
 // *** Funktion Speichern ***
 function speichern() {
   global $ip;                                                  // IP-Adresse
   global $pingschnitt;                                         // ping Durchschnitt
   global $pingmax;                                             // ping Maximum
   global $pingmin;                                             // ping Minimum
   global $tracert;                                             // Route zum Benutzer

   if(file_exists($ip))       {                                 // Wenn die Datei existiert mache
     $fs = fopen($ip, "a");                                     // Oeffne logfile zum schreiben
     fwrite($fs, "\r\n*** Latenzzeiten ***\r\n");
     fwrite($fs, "durchschnittliche: $pingschnitt\r\n");
     fwrite($fs, "maximale: $pingmax\r\n");
     fwrite($fs, "minimale: $pingmin\r\n\r\n");
     fwrite($fs, "*** Route zum Client ***\r\n");
     for($i = 0;$i < count($tracert);$i++) {                    // Solange tracert nicht leer wird geschrieben
       fwrite($fs, "$tracert[$i]\r\n");}
     fclose ($fs);                                              // Schliessen der Beiden Dateien
   }
```

```
 }
?>


</body>
</html>
```

Und das ist der Output:

Ping und Trace Testauswertung Host: www.sbb.ch

Latenzzeit in Millisekunden aus 8 Prüfungen

Durchschnitt:	Maximum:	Minimum:
36.67 ms	40 ms	30 ms

Routenverfolgung:

01: 217.27.97.1
02: 217.27.97.5
03: 217.27.96.2
04: 217.27.96.1
05: 194.191.191.185
06: 146.228.44.1
07: 146.228.253.70
08: 146.228.43.2
09: 194.42.48.29
10: 193.192.227.55
11: 193.192.227.38
12: 195.141.225.66
13: 193.192.251.74
14: 193.192.251.17

7.4.2.3 Einige JavaScript Beispiele

```
<!doctype html public "-//IETF//DTD HTML//EN">
<HTML>

<HEAD>
<TITLE>SYSTEM-CLINCH Java Test</TITLE>
<META NAME="GENERATOR" CONTENT="Note-Pad Editor">
<META NAME="AUTHOR" CONTENT="Manuel Magnin">
</HEAD>

<BODY><BODY BGCOLOR="#FFFFFF"><BR>

<script language="JavaScript">
/*<!--  Test of Java Script   -----------*/
/* Konfiguration */
  /*double Kilometer, Miles;*/
  Miles = 0.0;
  von = 0.0;
  bis = 100.0;
  Kilometer = 0.0;

  while (Miles <= bis)  {
     Kilometer = Miles * 1.61;
     document.write("Mails= " + Miles + "    Kilometer= " + Kilometer + "<BR>");
     Miles = Miles + 10;
  }  /* End While */
</SCRIPT>

</BODY>
</HTML>
```

Und das ist der Output:

Mails= 0 Kilometer= 0
Mails= 10 Kilometer= 16.1
Mails= 20 Kilometer= 32.2

Mails= 30 Kilometer= 48.3
Mails= 40 Kilometer= 64.4
Mails= 50 Kilometer= 80.5
Mails= 60 Kilometer= 96.6
Mails= 70 Kilometer= 112.7
Mails= 80 Kilometer= 128.8
Mails= 90 Kilometer= 144.9
Mails= 100 Kilometer= 161

Oder einen Taschenrechner komplett in JavaScript: http://www.clinch.ch/Net-Work/Java/calc.htm