

Hard disk search engine

– Marian Vețeanu

Orice persoană care a navigat până acum pe Web cunoaște importanța unui motor de căutare bun. Imensitatea Web-ului ar face aproape imposibilă găsirea informațiilor fără ajutorul motoarelor de căutare. Aceste *search engines* (denumirea în limba engleză) nu sunt altceva decât niște situri Web ce oferă vizitatorilor posibilitatea de căutare de informații în baza lor de date. Această bază de date conține informații despre milioane sau miliarde de pagini web. Indexarea paginilor web și introducerea informațiilor în baza de date se face în principal automat de programe specializate numite *roboți web*. Roboții parcurg în mod continuu paginile din web în vederea menținerii bazei de date prin adăugarea de informații noi sau modificarea celor existente (anumite situri Web pot dispărea de pe Internet sau își pot schimba conținutul, și atunci este neplăcut ca un motor de căutare să întoarcă rezultate invalide).

În ultima vreme banalele PC-uri pe care și le poate permite orice utilizator vin echipate cu dispozitive de stocare (discuri hard) de dimensiuni



impresionante : 20 Gbytes, 45 Gbytes, 73 Gbytes etc. Apare astfel nevoia unui mecanism de căutare performant care să permită în timp scurt regăsirea informațiilor de pe propriul disc hard. De câte ori nu ați fost pus în situația să vă conectați iar la Internet pentru o anumită informație chiar dacă atunci când ați văzut-o prima dată ați salvat-o undeva pe discul hard ? Sunt foarte multe persoane care navighează zilnic pe Web pentru a citi diverse articole pe care le salvează de asemenea și pe calculatorul local în ideea de a le reciti mai târziu. Majoritatea acestora vor rămâne rătăcite pentru totdeauna în *memoria* calculatorului. Este clar că programe precum *Windows Find...* nu pot face față unei căutări într-un asemenea hățiș de fișiere. Chiar pare amuzant faptul că o simplă căutare pe un disc hard mai mare poate dura de câteva zeci de ori mai mult decât o căutare pe Internet într-o bază de date cu miliarde de articole.

Din fericire odată cu *Windows 2000* utilizatorii de PC-uri pot beneficia de aceleași servicii pe care până acum le puteau accesa doar prin intermediul serviciilor online de pe Internet. *Windows 2000* include în mod standard serviciul *Index Server 3.0* numit și *Indexing Service* folosit pentru indexarea fișierelor de pe discul hard sau a celor publicate pe web. Creat inițial pentru indexarea fișierelor web în vederea ajutării vizitatorilor sitului să localizeze rapid informațiile dorite, în versiunea 3, adică cea din *Windows 2000*, *Index server* are și posibilitatea de a indexa fișierele de pe discurile hard locale sau stațiile de rețea.

Index server extrage conținutul fișierelor folosind componente filtru pentru fișierele cunoscute. *Windows 2000* include astfel de filtre pentru următoarele tipuri de fișiere: documente Office, HTML, mesaje MIME și fișiere text. Acestea sunt în general suficiente pentru un utilizator obișnuit. Cei mai pretențioși își pot crea singuri propriile filtre, Microsoft punând la dispoziție toată documentația necesară pe situl <http://msdn.microsoft.com>.

În cadrul articolului de față nu voi insista pe facilitățile avansate ale lui *Index server*, ci doar voi oferi o modalitate de construire a unei aplicații care să interogheze o bază de date ce conține informații despre fișierele de pe discul hard. Scenariul concret este următorul:

- Navigând pe Internet un utilizator aduce diverse documentații, articole din reviste, etc. sub forma unor pagini web pe care le salvează pe disc într-un anumit director (de exemplu, d:\books)
- Se pune problema regăsirii rapide a acestor informații folosind o metodă de căutare specifică Web-ului.

Primul pas care trebuie făcut este verificarea funcțio-





nării corecte a lui *Index server* cât și setarea acestuia în vederea indexării documentelor dorite. Intrați în sistem ca *Administrator* și apoi efectuați clic-dreapta pe *My Computer* și din meniul context alegeți *Manage*. Mergeți la *Services and Applications* -> *Services*. Asigurați-vă că serviciul dorit este setat să pornească automat și că rulează în acel moment (vezi figura „Administrare servicii”).

După această verificare este timpul să treceți la setarea serviciului. Mergeți în *Services and Applications* -> *Indexing Service* -> *Directories* și adăugați aici directoarele în care aveți informația ce o doriți indexată. Trebuie să știți că procesul de indexare consumă un spațiu măricel pe disc, așa că nu este bine să vă indexați tot discul în cazul în care nu aveți spațiu și dacă nu aveți peste tot răspândite fișiere importante. În cazul nostru trecem: *D:\books* (vezi figura „Setarea serviciului”).

Utilizatorii ce au sistemul de fișiere NTFS mai au o posibilitate de control a programului de indexare printr-un atribut specific obiectelor ce se află pe acest sistem de fișiere. Verificați în acest caz să aveți acest atribut poziționat pentru fișierele ce le doriți a le cataloga (vezi figura „Setarea atributului *Allow Indexing Service*”).

Setați apoi proprietățile pe care doriți să le extragă *Index server* și porniți indexarea. Starea în care se află procesul de indexare precum și alte informații ca dimensiunea ocupată pe disc de fișierele index le aflați tot din consola de administrare la *Services and Applications* -> *Indexing services* (vezi figura „Servicii indexare”).

Pentru a nu încetini utilizatorul, programul ce face indexarea lucrează doar când nu a mai detectat nici o activitate a utilizatorului într-un anumit interval de timp. Din acest motiv lăsați calculatorul un timp liber pentru a permite funcționarea programului.

Odată procesul de indexare fiind terminat nu vă mai rămâne decât să construiți o simplă aplicație care să consulte baza de date și să prezinte utilizatorului rezultatele căutării. Există mai multe modalități de interogare a lui *Index server*. În acest articol vom folosi ADO pentru a trimite interogări de tipul SQL către server. Acest lucru este posibil datorită faptului că *Microsoft* oferă un *ADO provider* pentru *MS Index Server*. În ideea de a semăna cât mai mult cu căutarea de pe Web vom crea aplicația sub forma unei pagini ASP ce poate fi rulată de serverul Web *IIS* de pe mașina locală (Windows 2000 include în mod standard serverul de Web *IIS*. Verificați ca acesta să fie activ). Cei care nu doresc să pornească acest server pot crea aplicația folosind DHTML, efectul fiind similar cu pagina ASP, în cazul în care totul se rulează pe o singură mașină. Dacă intenționați să dați posibilitatea și altor utilizatori din rețea să facă căutări pe mașina dvs. atunci ASP rămâne singura soluție rapidă.

Modul în care va arăta aplicația este prezentat în figura „Aplicație”.

Deși toată aplicația este conținută într-un singur fișier ASP, acesta va conține atât părți client cât mai ales părți server. Partea client principală o constituie formularul de căutare cu cele două versiuni ale sale:

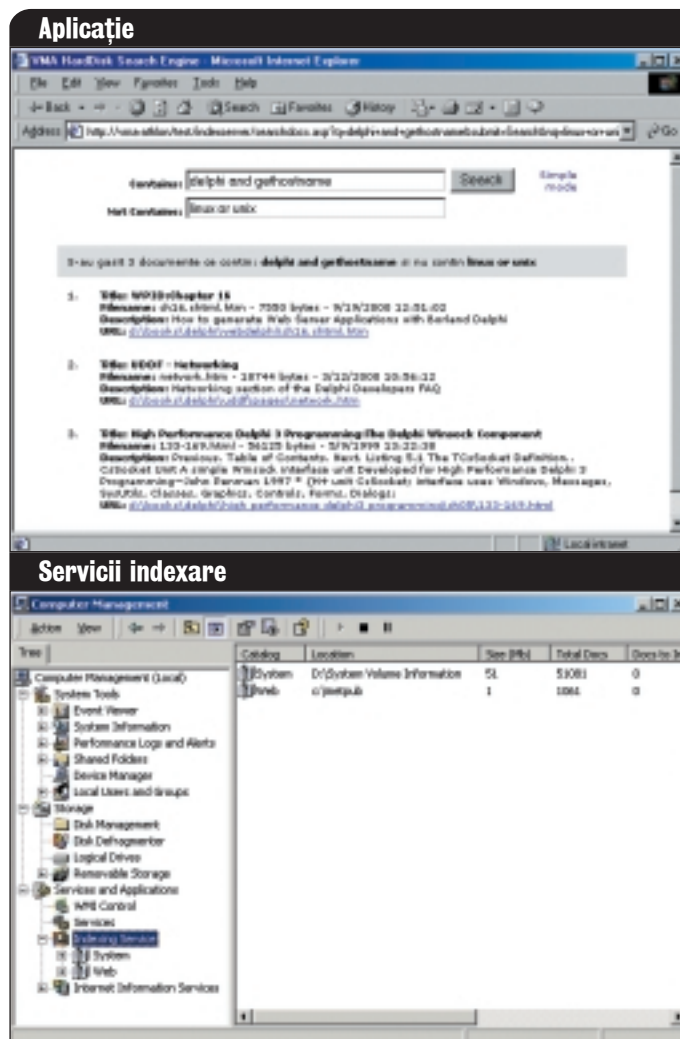
simplă și avansată. Comutarea între ele se realizează pe client cu un simplu script VBScript :

```
<script language=vbscript>

sub switchmodetext_onclick
if tablerow2.style.display="none" then
containstext.style.display=""
switchmodetext.innerHTML="Simple<br>mode"
tablerow2.style.display=""
else
containstext.style.display="none"
switchmodetext.innerHTML="Advanced<br>mode"
tablerow2.style.display="none"
end if
end sub
```

```
sub seachform_onsubmit
if tablerow2.style.display="none" then
seachform.nq.value = ""
end if
end sub
```

```
sub window_onload
if (seachform.nq.value <> "") and (tablerow2.style.display="none") then
```



```

<%@ Language=VBScript %>
<!--#include file="..\_private/advovbs.inc" -->
<%
Const RecordsPerPage = 10      ' Numarul de inregistrari de
    pe o pagina

printpageheader                ' Se afiseaza headerul
    paginii

if Request.QueryString.Count=0 then
    Response.Write "<center><font size=+2
        color=#000090><b>VMA HardDisk Search
        Engine</b></font></center>"
    printsearchform "", ""
else
    qe=Request.QueryString("q")
    nqe=Request.QueryString("nq")
    printsearchform qe,nqe
    DoSearch qe,nqe
end if

'
' Aceasta este subrutina principala ce face interogarea
    bazei
' lui Index Server si apoi afiseaza paginat rezultatele
'
sub DoSearch(q,nq)
    Set objConn = Server.CreateObject("ADODB.Connection")
    objConn.ConnectionString = "provider=msidxs"
    objConn.Open
    Set objRS = Server.CreateObject("ADODB.RecordSet")
    objRS.CursorLocation = adUseClient

    on error resume next
    objRS.Open BuildSQLQuery(q,nq), objConn,
        adOpenKeyset,adLockReadonly
    If Err.number <> 0 or (objRS.EOF and objRS.BOF) Then
        Err.Clear
        Response.Write "No records found!"
        Set objRS = Nothing
        objConn.Close
        Set objConn = Nothing
        Response.End
        Exit Sub
    End If

    objRS.PageSize = RecordsPerPage
    pag = Request.QueryString("pag")
    If pag <> "" Then
        pag = CInt(pag)
        If pag < 1 Then pag = 1
    Else
        pag = 1
    End If
    objRS.AbsolutePage = pag

    RowCount = objRS.PageSize
    RowOnPage = RecordsPerPage * (pag-1) +1
    Response.Write "<table border=0 width=90% align=center
        cellpadding=10>"

    Response.Write "<tr><td colspan=2 bgcolor=#e0e0e0>"
    Response.Write "S-au gasit " & objRS.RecordCount & "
        documente ce contin: <b>"&q"&"</b>"
    if nq<>"" then Response.Write " si nu contin
        <b>"&nq"&"</b>"
    Response.Write "</td></tr>"

    Do While Not objRS.EOF And RowCount > 0
        Response.Write "<tr>"
        Response.Write "<td valign=top align=left width=10>"
        Response.Write RowOnPage & "."
        RowOnPage = RowOnPage + 1
        with Response
            .Write "<td valign=top align=left>"
            .Write "<b>Title: " &
                Server.HtmlEncode(objRS.Fields("DocTitle").Value) &
                "</b><br>"
            .Write "<b>Filename:</b> " & objRS.Fields("File-
                name").Value & " - " & objRS.Fields("size").Value & "
                bytes - " & objRS.Fields("write").Value & "<br>"
            .Write "<b>Description:</b> " &
                Server.HtmlEncode(objRS.Fields("characterization").Valu
                e) & "<br>"
            .Write "<b>URL: </b><a href='" &
                objRS.Fields("path").Value & "'">"
            .Write objRS.Fields("path").Value
            .Write "</a><br>"&vbCrLf&vbCrLf&vbCrLf
            .Write "</td>"
            .Write "</tr>"
        end with
        RowCount = RowCount - 1
        objRS.MoveNext
    loop

    Response.Write "<tr><td colspan=2><table width=100%
        cellspacing=0 cellpadding=5 border=0 bgcolor=#e0e0e0
        align=center><tr>"

    If pag > 1 Then
        Response.Write "<td align=left><a
            href='searchdocs.asp?q="&q"&"&nq="&nq"&"&pag="&pag-
            1&">Back</a></td>"
    end if
    If RowCount = 0 Then
        Response.Write "<td align=right><a
            href='searchdocs.asp?q="&q"&"&nq="&nq"&"&pag="&pag+1&">N
            ext</a></td>"
    End If

    Response.Write "</tr></table></td></tr>"
    Response.Write "</table>"

    Set objRS = Nothing
    objConn.Close
    Set objConn = Nothing
end sub

'
' Construiesti interogarea SQL pentru Index Server
'
function BuildSQLQuery(q,nq)
    SQL = "SELECT Rank, Filename, Size, DocTitle, Path,
        Write, Characterization FROM System..Scope() "&_
        "WHERE CONTAINS(" & "'" & q & "'" & ")")
    if nq<>"" then SQL = SQL & " AND NOT CONTAINS(" & "'" &
        nq & "'" & ")")
    SQL = SQL & " ORDER BY Rank DESC"
    BuildSQLQuery = SQL
end function

'
' Tipareste headerul paginii
'
sub printpageheader%>
<head>
<title>VMA HardDisk Search Engine</title>
<!-- <DEFANGED_STYLE>
body,td
{
    font-family:verdana;
    font-size:10;
}
a.cool
{
    color:#000090;
    text-decoration:none;
}

```

```

a.cool:hover
{
    color:red;
    text-decoration:underline;
}
--> </DEFANGED_STYLE>
</head>
<%end sub

,
' Subrutina tipareste formul de search din prima pagina
' si din partea de sus a paginilor cu rezultate
,

sub printsearchform(q,nq)

if nq<>"" then
    searchformt1 = ""
    searchformt2 = "Simple<br>mode"
    searchformt3 = ""
else
    searchformt1 = " display:none;"
    searchformt2 = "Advanced<br>mode"
    searchformt3 = " style='display:none;'"
end if
%>

<DEFANGED_script language=vbscript>
sub switchmodetext_onclick
if tablerow2.style.display="none" then
    containstext.style.display=""
    switchmodetext.innerHTML="Simple<br>mode"
    tablerow2.style.display=""
else
    containstext.style.display="none"
    switchmodetext.innerHTML="Advanced<br>mode"
    tablerow2.style.display="none"
end if
end sub

sub seachform_onsubmit
if tablerow2.style.display="none" then
    seachform.nq.value = ""
end if
end sub

sub window_onload
if (seachform.nq.value <> "") and (tablerow2.style.display="none") then
    containstext.style.display=""
    switchmodetext.innerHTML="Simple<br>mode"
    tablerow2.style.display=""
end if
end sub
</script>

<table border=0 align=center cellpadding=0 cellspacing=3>
<tr><td align=center valign=center>
    <form name=seachform action=searchdocs.asp method=get>
    <table border=0 width=100% cellpadding=0 cellspacing=2 align=center>
    <tr>
    <td valign=center align=right>
        <span id=containstext style="font-weight:bold;<%=searchformt1%>">Contains:</span>
        <input name=q type=text size=40 value='<%=q%>'>
    </td>
    <td align=right>
        <input name=submit type=submit value="Search">
    </td>
    <td align=right valign=center width=60>
        <a href="#" class=cool id=switchmodetext><%=searchformt2%></a>
    </td>
    </tr>
    <tr id=tablerow2 <%=searchformt3%>>
    <td valign=center align=right>
        <span DEFANGED_STYLE="font-weight:bold;">Not Contains:</span>
        <input name=nq type=text size=40 value='<%=nq%>'>
    </td>
    <td colspan=2>&nbsp;  </td>
    </tr>
    </table>
    </form>
</td></tr>
</table>
<%end sub%>

```

```

    containstext.style.display=""
    switchmodetext.innerHTML="Simple<br>mode"
    tablerow2.style.display=""
end if
end sub
</script>

```

Partea de server primește prin `QueryString` în variabila `q` expresia ce se dorește a se găsi în interiorul fișierelor și în variabila `nq` (în modul *Advanced*) expresia ce nu se dorește a fi prezentă în interiorul fișierelor.

În continuare intră în funcțiune partea de server a paginii ASP și anume conectarea la sursa de informații, obținerea informațiilor și paginarea lor întocmai ca un *search engine* adevărat. Modalitatea de conectare prin ADO la baza întreținută de Index server este la fel de ușoară ca o conectare la o bază de date SQL Server.

```
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.ConnectionString = "provider=msidxs"
objConn.Open
```

Din informațiile primite prin `QueryString` următoarea funcție construiește o interogare SQL validă.

```
function BuildSQLQuery(q,nq)
SQL = "SELECT Rank, Filename, Size, DocTitle, Path, Write,
      Characterization FROM System..Scope() "&_
      "WHERE CONTAINS(" & "'" & q & "'" & ")"
if nq<>"" then SQL = SQL & " AND NOT CONTAINS(" & "'" & nq
& "'" & ")"
SQL = SQL & " ORDER BY Rank DESC"
BuildSQLQuery = SQL
end function
```

Această interogare SQL va fi folosită pentru construcția unui recordset ce va avea înregistrările ordonate descrescător după proprietatea *Rank*. Index server atribuie acestei proprietăți o valoare cu atât mai mare cu cât fișierul găsit se apropie mai mult de rezultatul dorit.

În continuare problema s-a redus la o simplă aplicație ASP cu ADO ce manipulează un *recordset* în vederea afișării paginate a informațiilor. Consultați sursa atașată pentru mai multe informații.

Adrese utile

MSDN – <http://msdn.microsoft.com>

ASP Today – <http://www.asptoday.com>

Marian Veteanu este ...; informații vmasoft@yahoo.com,
<http://vmasoft.hypermart.net>. ■ 45