Harvest - ein Internet-Suchsystem

Was ist Harvest

Bei der dramatisch angestiegenen Menge von Informationen im Internet, die im wesentlichen der Einführung des World Wide Web geschuldet ist, wird es immer schwieriger, relevante Informationen zu finden. Jedem wird es schon einmal passiert sein, daß er lange nach einer bestimmten Web Seite gesucht hat, nur weil vergessen wurde, diese in die Bookmark-Liste einzutragen.

Um das Chaos etwas zu entwirren, wurde eine Reihe von Lösungen vorgestellt. Diese reichen von den schon länger bekannten *Archies*, die vor allem ftp-Archive durchsuchen, bis hin zu den WWW-Robots, die sich rekursiv durch riesige Mengen von Web-Servern hangeln. All diese Systeme bergen mehr oder weniger große Nachteile in sich: entweder sind Dateinamen gesuchter Programme die einzige Information, die man erhält, Wartezeiten auf Suchergebnisse sind unerträglich groß und nicht zuletzt gehen die meisten dieser Suchsysteme nicht gerade sparsam mit Systemressourcen um und erzeugen insbesondere eine hohe Netzbelastung, da die Arbeit verschiedener Suchprogramme nicht oder nur unzureichend koordiniert wird und jeder die benötigten Informationen separat beschafft.

Zumindest einen Teil der obengenannten Probleme wollen die Harvest-Entwickler beseitigen. Harvest ist entstanden als Projekt der Internet Research Task Force Research Group on Resource Discovery (IRTF RD). Bei der Entwicklung wurde größter Wert auf den effizienten Umgang mit Ressourcen und eine hohe Konfigurierbarkeit des Systems gelegt. Dadurch eröffnet sich dem Anwender die Möglichkeit, mit geringem Aufwand eine Vielzahl bekannter Internet-Suchsysteme zu implementieren mit dem Vorteil einer hohen Wiederverwertbarkeit einmal gesammelter Informationen.

Um dies zu erreichen, wird der Prozeß der Informationssammlung von dem der Indexierung der Daten und der Suchanfrage via WWW-Formular getrennt. Der *Gatherer* sammelt Informationen, um danach in Abhängigkeit von deren Art relevante Daten zu extrahieren und diese abschließend in ein komprimiertes attribut- und streamorientiertes Format umzuwandeln. Damit entsteht für eine Vielzahl unterschiedlicher Dateiformate ein einheitliches Austauschformat (SOIF - Summary Object Interchange Format) , das äußerst effektiv weiterverarbeitet werden kann. Die eigentliche Indexierung der Daten und die Suchanfragen übernimmt der *Broker*. Ein Broker kann Daten von mehreren Gatherern und anderen Brokern erhalten. Er stellt ein WWW-Formular zur Verfügung, mit dem komplexe Suchanfragen formuliert werden können. Der Broker arbeitet dabei mit einem allgemeinen Index-Interface, so daß eine Vielzahl von Indexsoftware, wie z.B. freeWAIS, WAIS inc. commercial. und Nebula, benutzt werden kann. Standardmäßig wird *Glimpse* benutzt, das neben den „normalen“ Funktionen auch Attribute, reguläre Ausdrücke und Suchen von Wörtern mit Schreibfehlern unterstützt.

Zum Harvest-Paket gehören weiterhin ein *Replicator*, der die Verteilung von Brokern ermöglicht, um Hosts zu entlasten, die sehr viele Suchanfragen zu bewältigen haben, und der *Object Cache*, der Caching von FTP-, Gopher-, HTTP- und Newsobjekten sowie von DNS-Lookups unterstützt. Dieser Cache kann z.B. eine Alternative zum Einsatz des CERN httpd als Proxy-Cache sein. Auf die beiden letzten Komponenten soll jedoch im Rahmen dieses Artikels nicht eingegangen werden.

Vom Harvest-Team in Colorado werden eine Reihe von Brokern zu Demonstrationszwecken angeboten, darunter ein Index des eigenen WWW-Servers, ein Archiv von PC-Software und eine Sammlung von Homepages von Servern aus der ganzen Welt mit ca. 45000 Seiten. Hierbei können auch die Konfigurationsdateien der entsprechenden Gatherer eingesehen werden, so daß man sich hier eine Reihe von Anregungen für die eigene Installation holen kann. Weiterhin wird der sogenannte *Harvest Server Registry* Broker gepflegt, der Information zu sämtlichen im Internet verfügbaren Gatherern und Brokern sammelt, so daß entsprechend der Philosophie von Harvest doppelte Erfassung von Informationen vermieden werden kann.

Installation der Harvest-Software

An erster Stelle steht natürlich die Überlegung, auf welcher Maschine Harvest installiert werden soll. Wenn nur ein Gatherer installiert wird, der eine relativ kleine Datenmenge zu verwalten hat, kann sogar schon eine gutausgerüstete Linux-Maschine verwendet werden. Bei Installation eines Brokers und der dazugehörigen Indexierungssoftware steigt der Aufwand erheblich. Ein UNIX Rechner mit 64MB RAM und halbwegs schnellem Prozessor stellt dann schon ein Minimum dar, wenn eine schnelle Indexierung von Daten und Bearbeitung von Anfragen erreicht werden soll. In Abhängigkeit davon, wieviele Daten verarbeitet werden, sollten auf der Festplatte mindestens 200 MB verfügbar sein.

Die Harvest-Software kann entweder direkt beim Harvest-Team unter der Adresse **http://harvest.cs.colorado.edu/** oder von einem der Mirrorsites, wie z.B. der Akademischen Softwarekooperation in Karlsruhe (**ftp://ftp.ask.uni karlsruhe.de/pub/infosystems/harvest/**), beschafft werden. Die aktuelle Versionsnummer ist 1.4.

Für SunOS, Solaris, OSF/1 und HP UX gibt es bereits vorkompilierte Versionen. Es wird jedoch empfohlen, die Source-Distribution zu verwenden, da nur so die letzten Patches, die doch relativ häufig erscheinen, verwendet werden können. Die Binaries werden vom Harvest-Team nur alle zwei bis drei Monate hergestellt. Die Kompilierung von Harvest ist im allgemeinen unkompliziert, so daß sie auch für Nicht-UNIX-Gurus ohne Probleme durchführbar ist.

Für die Arbeit mit Harvest sollten die folgenden Programme verfügbar sein:

- Perl 4.0pl36 oder höher
- GNU gzip v1.2.4
- ein httpd-Server

Wer den Source-Code kompilieren möchte, benötigt zusätzlich:

- gcc 2.5.8 oder höher
- flex 2.4.7 oder höher
- bison 1.22 oder höher

Vom Harvest-Team selbst werden nur die drei Systeme SunOS, Solaris und OSF/1 unterstützt. Mit mehr oder weniger großem Aufwand sind jedoch erfolgreiche Installationen unter AIX, HP-UX, Linux und IRIX möglich. Gerade in der Version 1.4 wurde eine Reihe von Bugfixes eingearbeitet, die die Installation auf anderen Systemen ermöglichen. Im contrib-Verzeichnis der Harvest-Distribution befinden sich Binaries für diese nicht offiziell unterstützten Plattformen.

Zur Installation und allen anderen Fragen zu Harvest finden sich Hinweise in der Newsgroup **news:comp.infosystems.harvest** und im **User-Manual**.

Für die ersten Versuche mit Harvest sollte nach der Installation das Skript *RunHarvest* gestartet werden. Hierbei werden im Dialog ein Gatherer und ein zugehöriger Broker installiert. Damit ist eine Basis von Konfigurationsdateien erzeugt, die sich dann weiterbearbeiten läßt.

Der Gatherer

Einführung

Wie bereits erwähnt, ist der Gatherer für die eigentliche Beschaffung der Informationen verantwortlich. Dazu werden unter anderem die Protokolle HTTP, FTP, Gopher und News unterstützt, so daß Harvest auf den zu indexierenden Hosts nicht installiert sein muß. Selbstverständlich sollte nach Möglichkeit der Gatherer auf der Maschine arbeiten, auf der auch die gewünschten Informationen liegen, da dann keine aufwendigen Netzprotokolle verwendet werden müssen. Dies gilt insbesondere bei größeren Datenmengen (ab ca. 1000 Objekten). Der Inhalt der empfangenen Dateien wird dann je nach Dateityp extrahiert und in das SOIF-Format umgewandelt. Ein Beispiel für ein Objekt im SOIF-Format finden sie im Anhang des **User-manuals**.

Für die Installation eines einfachen Gatherers genügt die Angabe der URLs, die indexiert werden sollen. Diese erfolgt zusammen mit ein paar ergänzenden Informationen in einem File, das üblicherweise die Endung *.cf* trägt. Ein Beispiel finden sie in Tabelle 1.

```
# test.cf
Gatherer-Name: TEST
Top-Directory: /usr/local/harvest/gatherers/test
Gatherer-Port: 8500
Local-Mapping: http://www.hu-berlin.de /usr3

<RootNodes>
http://www.hu-berlin.de/ URL=2000,Filter
</RootNodes>

<LeafNodes>
http://www2.rz.hu-berlin.de/test/manual.ps
</LeafNodes>
```

Tabelle 1

Bei der Angabe der URLs wird zwischen Root- und Leafnodes unterschieden. Letztere sind Angaben von Einzeldokumenten, die geholt und verarbeitet werden. Rootnodes werden je nach Art des Zugriffs in mehrere Leafnodes aufgesplittet und dann weiterverarbeitet, so werden z.B. von HTML-Dokumenten alle Links weiterverfolgt, FTP holt das gesamte Verzeichnis einschließlich aller Unterverzeichnisse, News holt alle Artikel der angegebenen Newsgroup. Da diese Verfahrensweise zum Teil riesige Datenmengen erzeugen kann, wurden standardmäßig einige Einschränkungen vorgenommen, die jedoch durch verschiedenen Optionen verändert werden können. Unter anderem kann bestimmt werden, in wieviele Leafnodes eine Rootnode maximal aufgesplittet wird (Standard: 250), welche URLs ausgeschlossen werden sollen (Filterung über reguläre Ausdrücke), wieviele verschiedene Hosts angesprochen werden dürfen (Standard: 1, d.h. alle Dokumente liegen auf demselben Server) und bis zu welcher Tiefe Rootnodes verfolgt werden sollen (Standard: 0, d.h. unbegrenzte Tiefe - jeder aus Rootnodes gewonnene URL wird weiter auf darin enthaltene Links untersucht). So wird im Beispiel der Tabelle 1 die Anzahl der aus der Rootnode erzeugten URLs auf 2000 begrenzt, wobei nur solche URLs zugelassen sind, die durch einen regulären Ausdruck im File *Filter* beschrieben sind. Die Option *Local-Mapping* bildet einen URL auf das Filesystem ab, so daß die Dokumente nicht über das Netz geholt werden müssen. Weitere Informationen zur Konfiguration des Gatherers finden sich wie immer im **User-manual**.

Dateiformat	Extrahierte Daten
C, Cheader	Extract procedure names, included file names, and comments
Dvi	Invoke the Text summarizer on extracted ASCII text
FAQ, FullText, README	Extract all words in file
Framemaker	Up-convert to SGML and pass through SGML summarizer
HTML	Extract anchors, hypertext links, and selected fields (see SGML)
LaTex	Parse selected LaTex fields (author, title, etc.)
Makefile	Extract comments and target names
ManPage	Extract synopsis, author, title, etc., based on ``-man'' macros
News	Extract certain header fields
Patch	Extract patched file names
Perl	Extract procedure names and comments
PostScript	Extract text in word processor-specific fashion, and pass through Text summarizer.
RTF	Up-convert to SGML and pass through SGML summarizer
SGML	Extract fields named in extraction table (see Section~\ref{sec:sgml})
SourceDistribution	Extract full text of README file and comments from Makefile and source code files, and summarize any manual pages
Tex	Invoke the Text summarizer on extracted ASCII text
Text	Extract first 100 lines plus first sentence of each remaining paragraph
Troff	``-me'' macro packages, or extract section headers and topic sentences.
Unrecognized	Extract file name, owner, and date created.

Tabelle 2

Nachdem die entsprechenden Dateien vom Gatherer geholt worden sind, werden sie durch das *Essence*-System weiterverarbeitet. Essence ist in der Lage, anhand des Dateinamens und der Informationen, die das UNIX-Kommando *file* liefert, den Dateityp zu erkennen. In Abhängigkeit davon werden *Summarizer* aufgerufen, die bestimmte Informationen der Dateien extrahieren. Für eine Vielzahl von Dateitypen bietet Harvest schon fertige Lösungen an. Es ist jedoch kein Problem, über entsprechende Konfigurationsdateien auch andere Dateitypen zu verarbeiten. Eine Auswahl der von Harvest erkannten Typen findet sich in Tabelle 2. Ein wohl sehr interessantes Feature von Harvest ist die Möglichkeit, SGML-Dateien zu verarbeiten. Auch HTML-Dokumente, die ja im Prinzip auch nur SGML-Dokumente mit einer speziellen DTD (Document Type Definition) sind, werden zuerst nach SGML konvertiert und dann dem SGML-Summarizer übergeben. In Tabelle 3 sind die HTML-Elemente und die entsprechenden SOIF-Attribute aufgelistet.

Harvest erlaubt es, eigene Summarizer in das Essence-System zu integrieren. So wird z.B. die RTF-SGML Konvertierung durch ein Paket der Fa. Electronic Book Technologies durchgeführt und dann der SGML-Summarizer verwendet.

Auch die Arbeit von Essence kann auf vielfältige Weise über Konfigurationsdateien gesteuert werden. So können z.B. Extrahierungsstrategien für bestimmte Dateitypen verändert werden. Eine genaue Beschreibung würde jedoch den Rahmen des Artikels sprengen.

Im Ergebnis der Arbeit des Gatherers wird ein Dämon auf dem im Konfigurationsfile angegebenen Port (Standard: 8500) gestartet, der über das Kommando bin/gather abgefragt werden kann. Man erhält dann den schon erwähnten Stream von SOIF-Objekten, der vom Broker weiterverarbeitet werden kann.

Um ein regelmäßiges Update der Daten zu erhalten oder neue Dateien zu sammeln, muß der Gatherer erneut gestartet werden. Dazu kann das Skript *RunGatherer* in die crontab eingetragen und damit die Sammelhäufigkeit bestimmt werden.

Der Broker

Der Broker übernimmt die eigentliche Indexierung der vom Gatherer gesammelten Daten und bietet den Zugriff über ein WWW-Formular. Über das Skript *CreateBroker* wird ein Broker im Dialog in einem Verzeichnis installiert, das für den WWW-Server sichtbar sein muß. Weiterhin müssen die CGI-Skripte aus dem Harvest-Verzeichnis in das entsprechende Verzeichnis für Skripte des WWW-Servers kopiert werden. Für die weitere Beschreibung wird davon ausgegangen, daß Glimpse als Indexsoftware verwendet wird. WAIS und Nebula werden von Harvest direkt unterstützt. Für andere Suchprogramme müssen eigene Anpassungen an das allgemeine Such-Interface des Brokers vorgenommen werden.

Vor dem Start des Brokers sollten noch die Konfigurationsdateien *admin/broker.conf, admin/Collection.conf* und das WWW-Formular query-glimpse.html durchgesehen werden, da einige Variablen (Broker-Directory) standardmäßig falsch gesetzt sind.

Der Broker wird mit dem Skript *RunBroker* gestartet. Es können Daten von einem oder mehreren Gatherern, aber auch von anderen Brokern empfangen werden. Hierbei wird in der Konfigurationsdatei Collection.conf der Name und Port des Gatherers/ Brokers an-

HTML-ELEMENT	SOIF-ATTRIBUTES
<A>	keywords,parent
<A:HREF>	url-references
<ADDRESS>	address
<B>	keywords,parent
<BODY>	body
<CITE>	references
<CODE>	ignore
<EM>	keywords,parent
<H1-H6>	headings
<HEAD>	head
<I>	keywords,parent
<IMG:SRC>	images
<META:CONTENT>	$NAME
<STRONG>	keywords,parent
<TITLE>	title
<TT>	keywords,parent
<UL>	keywords,parent

Tabelle 3

gegeben. Beim Broker wird über das QUERY Flag eine Anfrage formuliert, und nur diese Daten werden dann empfangen. Damit besteht also die Möglichkeit, mit Views von anderen Brokern weiterzuarbeiten.
Anschließend wird der eigentliche Index erzeugt und der *glimpseserver* Dämon gestartet. Nun sollten Anfragen an den Broker über das WWW-Formular gestartet werden können. Glimpse bietet die folgenden Funktionen beim Suchen an: Unterscheidung Groß/Kleinschreibung, logische Verknüpfungen (AND/OR), Substring-Suche, Tolerieren von Schreibfehlern, reguläre Ausdrücke, Suche nach Attributen, Anzeige passender Zeilen und Limitierung der Anzahl gefundener Einträge. Zum Beispiel erhält man auf die Suchanfrage „Universität AND (url:hu-berlin)“ alle Einträge zurück, die „hu-berlin“ im URL und „Universität“ im Text enthalten.

Das Aussehen der HTML-Seite mit den gefundenen Objekten kann in der Datei *cgi bin/lib/BrokerQuery.cf* geändert werden.

Wenn bei der Suchanfrage das Flag *Display Object Description* gesetzt wird, kann man sich das entsprechende SOIF-Objekt für gefundene Dateien ansehen. Damit kann man gut kontrollieren, ob auch wirklich die gesamte Datei indexiert wurde oder z.B. aufgrund von HTML-Syntaxproblemen (siehe unten) nur Teile der Datei erfaßt wurden.

Updates der Daten werden vom Broker automatisch vorgenommen. Die Angabe des Zeitintervalls erfolgt in *broker.conf* in der Variablen *Collection-Time*.

Harvest in der Praxis

Am Rechenzentrum ist z.Z. ein Broker in Betrieb, der einen Volltextindex eines großen Teils der an der HUB betriebenen WWW-Server anbietet. Dieser Broker wird von drei Gatherern versorgt, die auf den folgenden Servern installiert sind:

1. Server der HUB (www.hu-berlin.de), dieser Gatherer erfaßt auch eine Reihe weiterer Server mit einer geringen Anzahl von Dokumenten
2. Institutsserver des Rechenzentrums mit acht verschiedenen Instituten (www2.rz.hu-berlin.de/inside/...) und
3. Server der Abteilung Pädagogik und Informatik am Erziehungswissenschaftlichen Institut (www.educat.hu-berlin.de).

Geplant ist die Installation von zwei weiteren Gatherern an der Juristischen Fakultät und am Institut für Mathematik.

Ein weiterer Broker hält einen Index von Rechenzentrumsmitteilungen deutscher Universitäten und Hochschulen. Er enthält ca. 2500 Objekte von 25 verschiedenen Servern.

Abschließend finden sie noch ein paar Hinweise zur Installation von Harvest, die sich aus der praktischen Arbeit ergeben haben:

- Die Indexierung von Umlauten wird unterstützt, allerdings sollte, um sicherzugehen, das CGI-Skript *BrokerQuery.pl.cgi* so modifiziert werden, daß bei der Ausgabe der gefundenen Dokumente HTML-Entities verwendet werden.
- Der Gatherer benötigt beim Erfassen von Seiten, die sehr viele Links enthalten (z.B. Listen von Servern zu bestimmten Themengebieten), sehr viel Zeit, obwohl die Verfolgung von Links zu anderen Hosts ausgeschlossen wurde. Solche Seiten sollten mit Hilfe eines URL-Filters nicht indexiert werden.
- Glimpse verwendet einen sogenannten Turbo-Index, der für eine Vielzahl von Suchanfragen auch bei sehr großen Datenmengen schnelle Ergebnisse liefert. Er wird verwendet bei gemeinsamer Benutzung der Optionen *Case insensitive* und *Keywords match on word boundaries*. Da er ausschließlich mit dem Index und nicht mit den Objekten arbeitet, kann er bei mehrteiligen Suchbegriffen **nicht** feststellen, ob diese Worte auch hintereinander stehen, sondern nur, ob sie gemeinsam im gesamten File vorkommen. Bei solchen Suchanfragen werden deshalb meist mehr Einträge gefunden als erwartet.
- Da HTML-Dateien erst nach SGML konvertiert und dann erst verarbeitet werden, ergeben sich hier höhere Anforderungen an die Einhaltung der Syntax. Was unter Netscape ganz hervorragend aussieht, kann eine katastrophale Syntax haben, die eine vollständige SGML-Konvertierung so gut wie unmöglich macht. Wenn also bei der Suche nach Begriffen keine Treffer zu verzeichnen sind, die indexierten Dateien diese aber auf jeden Fall enthalten, könnte es an mangelhafter HTML-Syntax liegen. Wer sich die Fehlermeldungen des SGML-Summarizers ansehen möchte, setzt in *lib/gatherer/SGML.sum* die Variable *$syntax_check*=1. Dann werden in die Datei *log.errors* zusätzlich SGML-Fehler geschrieben. So können auf einfache Weise HTML-Dokumente auf syntaktische Korrektheit überprüft werden.

Daniel Ohst