Arbeitsbericht: Projekt SWING 2001

Andreas Heuer Andreas Rann

Universität Rostock, Fachbereich Informatik,

Lehrstuhl Datenbank- und Informationssysteme

30. Dezember 2001

Abstrakt

Eine schnelle Reaktion auf Nutzeranfragen und die Bestimmung der besten Treffer sind wichtige Anforderungen an aktuelle Suchmaschinen. Im Berichtszeitraum wurden an der Suchmaschine SWING u.a. Arbeiten zur Beschleunigung einzelner Komponenten der Suchmaschine und zum Ranking durchgeführt. Dieser Bericht beschreibt diese Arbeiten und deren Ergebnisse. Anhand verschiedener Messungen wurden die erreichten Resultate überprüft.

1 Einleitung und Zielstellung

Im Anfrage- und Suchdienst des Landesinformationssystems MV-Info, in der Kooperationsplattform Business-MV und im Webauftritt des Fachbereichs Informatik an der Universität Rostock wird die durch den Lehrstuhl Datenbank- und Informationssysteme des Fachbereichs Informatik der Universität Rostock entwickelte Suchmaschine SWING (**S**uchdienst für **W**WW-basierte **I**nformationssysteme der **n**ächsten **G**eneration) eingesetzt. Diese Suchmaschine und zusätzliche Dienste wurden in 3 Projektphasen im Zeitraum 1996 bis 2001 entwickelt. Dieser Arbeitsbericht beschreibt einen Teil der Arbeiten in der 3. Projektphase[1]. Die Arbeiten wurden in 3 Arbeitsetappen ausgeführt. In der ersten Arbeitsetappe wurde untersucht, wie der aus dem gewachsenen Datenvolumen resultierenden Erhöhung der Antwortzeiten entgegengewirkt werden kann. Daher wurde geprüft, ob der Einsatz eines alternativen Indizierers innerhalb der Architektur der Suchmaschine zu einer Verbesserung der Antwortzeiten führt. Der ermittelte Indizierer wurde anschließend in die Suchmaschine integriert. In Auswertung der erreichten Laufzeitverbesserung, wurde in der zweiten Arbeitsetappe die Aufgabenverteilung innerhalb der Anfragekomponente verändert. Um eine weitere Verbesserung der Verfügbarkeit und der Antwortzeiten zu erreichen, wurden in der dritten Arbeitsetappe Untersuchungen zu den Kosten der Anfrageverarbeitung in der verteilten Architektur von SWING angestellt. In deren Ergebnis wurde eine Konzeption für einen Anfragepuffer in SWING erarbeitet.

Die Umsetzung dieser Projektaufgaben erfolgte innerhalb von 9 Personenmonaten.

Im Folgenden sind die einzelnen Realisierungsschritte beschrieben.

---

[1] Einen vollständigen Überblick liefert [3].

2 Realisierung

2.1 Erste Arbeitsetappe: Auswahl und Integration eines alternativen Indizierers

Gemäß der gegebenen Zielstellung wurde in einem ersten Schritt nach Indizierern gesucht, die für die Integration in SWING geeignet sind. In einem zweiten Schritt erfolgte die Integration von zwei geeigneten Indizierern in die Suchmaschine. Anschließend wurde in einem Laufzeitvergleich überprüft, ob die Zielstellung, die Anfrageausführung zu beschleunigen, erreicht wurde.

2.1.1 Ausgangszustand

In den folgenden Kapiteln sollen die Arbeitsweise und die Aufgaben der Bestandteile der Anfragekomponente in der Suchmaschine SWING[1], insofern sie für die Realisierung der Projektaufgabe wesentlich sind, kurz beschrieben werden. Anschließend werden resultierende Anforderungen abgeleitet, die an einen alternativen Indizierer zu stellen sind.

2.1.1.1 Arbeitsweise der Anfragekomponente

Die Anfragekomponente erfüllt im Wesentlichen zwei Aufgaben, die sich auf die Vorbereitungs-/Indizierungsphase und die Anfragephase verteilen. In der Vorbereitungsphase erfolgt die Aktualisierung der Datenbasis (Dokumentensammlung) und des Indexes sowie die Auswertung der Dokumente. In der Anfragephase erfolgt die Ausführung von Suchanfragen an die Datenbasis mit der Aufbereitung und Übergabe der Anfrageergebnisse. Als Programme kommen sowohl C- und Perl- Programme als auch Skripte zum Einsatz. In der folgenden Abbildung ist die Anfragekomponente mit ihren Bestandteilen und Schnittstellen zur Sammelkomponente und zur Web-Schnittstelle dargestellt.

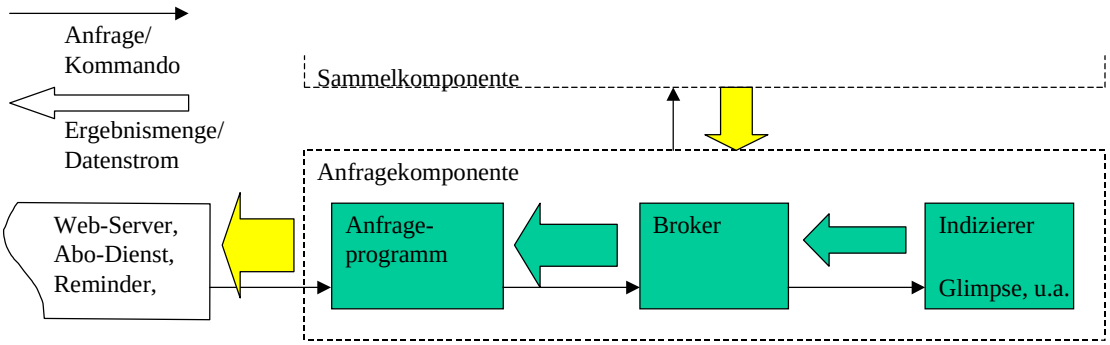

Abb.1: Architekturausschnitt der SWING: Anfragekomponente

Die Anfragekomponente besteht aus dem Anfrageprogramm[2], dem Broker und dem Indizierer.

[1] SWING ist eine Weiterentwicklung der Suchmaschine Harvest.

[2] Hierbei handelt es sich i.d.R. um das in Perl implementierte CGI-Programm *nph-search.cgi*.

In der Anfragephase kann durch den Webserver das Anfrageprogramm gestartet werden, um eine Suchanfrage zu bearbeiten. Es übermittelt den Anfragetext und weitere Parameter über eine Socketkommunikation an den als Serverdienst arbeitenden und in den Anfrageparametern festgelegten Broker. Dieser bearbeitet den Anfragestring und übergibt die Anfrage an die Schnittstelle zwischen dem Broker und dem Indizierer (Indizierer-Schnittstelle). Hier wird die Anfrage, entsprechend dem eingebundenen Indizierer aufbereitet und an diesen weitergeleitet. Der Indizierer kann als Serverdienst über eine Socketkommunikation angesprochen werden. Alternativ kann er als Programmdatei über die Kommandozeile gestartet werden. Die erste Variante kommt bei der Verwendung von Glimpse zum Einsatz. Die Einbindung von SWISH-E und MG erfolgt über Kommandozeilenaufrufe. In beiden Fällen werden dem Indizierer die Anfrage und weitere notwendige Parameter, wie z.B. Konfigurationsdateien oder der Indexname übergeben. Für eine Anfrage bestimmt der Indizierer aus dem Index die Ergebnismenge. Diese besteht wenigstens aus einer Liste von Identifikatoren für die Treffer. Diese Treffer werden an den Broker übermittelt. Dieser prüft die übergebenen Treffer und reichert die Daten pro Treffer um zusätzliche Informationen an. Diese erweiterte Ergebnismenge wird an das aufrufende Anfrageprogramm übergeben. In diesem Programm findet ein Ranking, eine Sortierung und eine Gruppierung der gefunden Dokumente statt.
In der Vorbereitungsphase muss die Datenbasis des Brokers aktualisiert und ein Index für die Daten erzeugt werden. Die Datenbasis besteht im wesentlichen aus Dateien im Standard Object Interchange Format (SOIF). Jede dieser Dateien repräsentiert eine Web-Datenquelle, z.B. eine HTML- oder PDF-Datei, eine Datenbanken oder Ähnliches. Die Dateien werden vom Indizierer durch Zugriff auf das Dateisystem indiziert. Die Dateien liegen in Unterverzeichnissen in der Verzeichnishierarchie des Brokers. Den Dateien ist über den Dateinamen ein eindeutiger Identifikator zugeordnet. Beim Start der Indizierung wird dem Indizierer das Verzeichnis der Datenquellen und ein Zielverzeichnis für den anzulegenden Index übergeben. In periodischen Abständen wird durch den Broker eine Aktualisierung des Datenbestandes und anschließend eine erneute Indizierung der Daten durchgeführt. Im Folgenden ist eine grobe Übersicht der zuvor beschriebenen Aufgaben und ihrer Verteilung zwischen den Bestandteilen der Anfragekomponenten aufgeführt.

	Anfrage-phase	**Indizierungs-phase**
Indizierer		
Erstellung und Aktualisierung des Index.		X
Ausführung von Anfragen und Rückgabe der Ergebnismenge	X	
Broker		
Anfragen entgegennehmen, umformatieren und an den Indizierer leiten.	X	
Die Ergebnismenge entgegennehmen, prüfen, um Informationen anreichern und an das Anfrageprogramm weiterleiten.	X	
Die Aktualisierung der Datenbasis durchführen.		X
Die Indizierung starten.		X

Anfrageprogramm		
Übergabe von Anfragen an den Broker.	X	
Ranking, Gruppierung, Sortierung und Formatierung der Anfrageergebnisse.	X	
Rückgabe der formatierten Anfrageergebnisse in HTML Dateien.	X	

Abb.2: Überblick über die Aufgabenverteilung in der Anfragekomponente

Aus der dargestellten Funktionsweise und Architektur werden im folgenden Kapitel notwendige Anforderungen an den Indizierer abgeleitet.

2.1.1.2 Ableitung notwendiger Anforderungen aus SWING an den Indizierer

Durch die modulare Architektur der SWING Suchmaschine, können die Aufgaben des Indizierers auf das Anlegen des Index und die Bestimmung einer zu einer Anfrage passenden Ergebnismenge reduziert werden. Dadurch ist es möglich den Indizierer relativ leicht auszutauschen. Durch den Austausch dieser Programmkomponente sind fest umrissene Funktionalitäten und Schnittstellen betroffen. Im Einzelnen ergeben sich damit folgende Anforderungen an einen neu zu integrierenden Indizierer.

- Die Indizierungs- und Anfragekomponente muss über die Kommandozeile aufrufbar sein oder über Socket-Verbindungen interagieren können[1]. Dabei müssen alle notwendigen Parameter per Kommandozeile (evtl. über Eingabeumleitung aus einer Datei) oder über den Datenstrom übermittelt werden können. Insbesondere darf keine Dialog-Schnittstelle zwingend bedient werden müssen[2].
- Es muss die Indizierung aller Dateien in einer gegebenen Verzeichnis-Hierarchie möglich sein.
- Der verwendete Index muss als Verzeichnis- oder Dateiname spezifizierbar sein.
- Die Übergabe der Ergebnismenge muss über Temporärdateien oder über Socket-Verbindungen möglich sein.
- Die Ergebnismenge muss mindestens die Namen der indizierten Dateien übergeben.
- Die zu indizierende Datenmenge muss mindestens die Größe der aktuellen Datenbasis der für MV-Info benutzten Broker[3] haben

Zusätzlich muss der Indizierer für jede Datenquelle einen Gewichtungswert liefern und die Ergebnismenge entsprechend dieses Gewichtungswertes sortieren. Durch die Verlagerung eines Teils der Berechnung der Dokumentgewichtungen in den Indizierungsprozess, soll eine Beschleunigung der Anfrageaus-

[1] Durch einem größeren Ressourceneinsatz können selbstverständlich weitere Techniken für die Einbindung von Indizierungsprogrammen genutzt werden.

[2] Dies kann als eine Minimalanforderung an einen einzubindenden Indizierer betrachtet werden.

[3] Die Broker *SwingBroker* und *uni-hro* bilden zusammen die Datenbasis der Suchmaschine von MV-Info. Im Juli 2001 waren das zusammen mehr als 82.000 Dateien mit ca. 326 MB Plattenplatz, ohne doppelte Dateien.

führung erreicht werden. Ob dieses Ziel erreicht wurde, wird in einem anschließenden Laufzeitvergleich überprüft.
Im folgenden Kapitel wird zu den abgeleiteten Anforderungen ein geeigneter Indizierer bestimmt.

2.1.2 Auswahl eines geeigneten Indizierers

Für die Auswahl eines geeigneten Indizierers wurde eine Literatur und Online-Recherche durchgeführt. Über den Literaturkatalog OPAC an der Universität Rostock, die WWW-Suchmaschine Google und die WWW-Adresse *searchtools.com* [6] wurden Informationen zur Suchmaschine Harvest und verschiedenen Indizierern gefunden. Außerdem wurden der Dokumentation und den Quellen der Installations- und Programmverzeichnisse der Harvest-Versionen 1.5.X und 1.6.X Informationen entnommen. Im folgenden Abschnitt sind die dabei gewonnenen Ergebnisse dargestellt.

2.1.2.1 Auswahlkriterien

Die in der folgenden Tabelle aufgeführten Kriterien wurden für die Auswahl eines Indizierers betrachtet. Der Schwerpunkt lag dabei auf einer höheren Geschwindigkeit bei der Anfragebearbeitung, der Menge der indizierten Daten, der freien Verfügbarkeit des Programms, dem Vorhandensein eines Ranking-Algorithmus und dem Vorhandensein von Informationen über die Integrierbarkeit in Harvest.

Funktionalität	Art der Anfragen, Stemming J/N, Felder (strukturierte Anfragen) Präsentation des Anfrageergebnisses (zusätzliche Daten, wie Titel, Trefferstellen, usw.) Art und Menge der indizierten Daten
Laufzeit	der Anfrage der Indizierung
Ranking	Qualität Formel
Support	vorhanden J/N Entwickler (Anzahl, Firma)
Verfügbarkeit	Betriebssystem Lizenzbedingungen Preis bzw. frei verfügbar (notwendige Bedingung) Quellen verfügbar
Dokumentation	Nutzer-, Installations-, Entwicklerdokumentation vorhanden J/N
Hinweise auf die Integrierbarkeit in Harvest	vorhanden J/N vorbereitete Schnittstellen

Abb.3: Kriterien für die Auswahl eines Indizierers

Im folgenden Kapitel werden Indizierer betrachtet, deren Integration in die Suchmaschine SWING bereits vorbereitet wurde.

2.1.2.2 Für die Integration vorbereitete Indizierer

Um den anfallenden Anpassungsaufwand bei der Integration in SWING möglichst gering zu halten, wurden als erstes die Indizierer betrachtet, für die Schnittstellen zur Integration in die Suchmaschine vorbereitet sind. In der folgenden Tabelle sind diese Indizierer aufgeführt. Dazu sind einige Besonderheiten erläutert. Gegebenenfalls wurde angegeben, warum der jeweilige Indizierer für eine Verwendung in SWING nicht in Frage kommt.

Indizierer	**Hinweise/Besonderheiten**
Glimpse	Glimpse ist der standardmäßig in SWING und Harvest verwendete Indizierer. Er wird in der aktuellen Harvest-Version 1.6.X (Stand März 2001) in der Version 4.12, 1999 eingesetzt. Der Indizierer wird in SWING in der Version 4.0B1, 1996 genutzt. Aufgrund der speziellen Arbeitsweise dieses Indizierers ist die Laufzeit bei der Anfragebearbeitung höher als bei vergleichbaren Indizierern. Andererseits bietet Glimpse einen großen Funktionsumfang und eignet sich damit besonders für eine "erweiterte Suche".
GRASS	Dieser Indizierer ist spezialisiert für die Indizierung geographischer Daten und scheidet somit aus der weiteren Betrachtung aus.
Nebula	Dieser Indizierer scheint nach den durchgeführten Recherchen nicht als voll funktionsfähiges Programm zu existieren.
PLWeb	Dieser Indizierer bzw. seine Quellen sind nicht frei verfügbar.
SWISH	Für diesen Indizierer existieren zwei aktuelle Nachfolgeversionen, SWISH-E und SWISH++. Obwohl diese ähnlich klingende Namen haben, handelt es sich um zwei verschiedene Programme. SWISH-E wird aktiv weiterentwickelt. Es existiert eine Mailing-Liste und durch das Entwicklerteam wird schnell auf Anfragen und Fehlermeldungen reagiert und Unterstützung bei Problemen geliefert. Da die Entwicklung und der Support von SWISH++ scheinbar nur durch einen Entwickler geleistet wird, scheidet dieser Indizierer bei der weiteren Betrachtung aus. Laut einem Ranking von *searchtools.com* [7] wird von SWISH-E und SWISH++ hauptsächlich SWISH-E eingesetzt.
WAIS	Mit freeWais-sf existiert eine Nachfolgeversion von Wais, die strukturierte Anfragen ausführen kann. Der Quelltext dieses Programmsystems wurde als relativ uneinheitlich und für eine Weiterentwicklung wenig geeignet beschrieben.

Abb.4: Informationen zu den in SWING und Harvest integrierten Indizierern

Nach den angestellten Betrachtungen und einem Vergleich mit den im Kapitel 2.1.2.1 aufgestellten Kriterien verbleibt als Alternative zum Indizierer Glimpse somit SWISH-E. Im Anhang auf Seite 46 ist ein Überblick über den durch SWISH-E gebotenen Funktionsumfang gegeben.

Im folgenden Abschnitt werden die Betrachtungen auch auf frei verfügbare Indizierer ausgeweitet, deren Integration in SWING nicht vorbereitet ist.

2.1.2.3 Frei verfügbare Indizierer

In der folgenden Tabelle sind Indizierer und Suchmaschinen aufgeführt, die frei verfügbar sind und nicht bereits im vorigen Abschnitt betrachtet wurden. Diese Übersicht wurde in erster Linie durch Auswertung der WWW-Seiten von *searchtools.com* [6] und von Hinweisen aus WWW-Dokumenten und eMail-Archiven erstellt, die über verschiedene Suchen in *Google* gefunden wurden. Die folgende Tabelle nennt die Indizierer und gibt einige Erläuterungen zu Besonderheiten und der Eignung für eine Verwendung in der Suchmaschine SWING.

Indizierer	**Hinweise/Besonderheiten**
MG	Bei *Managing Gigabytes (MG)* handelt es sich um eine Sammlung aus Sriptdateien und C-Programmen. MG kann große Datenmengen verarbeiten und bietet verschiedene Anfragemöglichkeiten und Ranking-Varianten. Da MG aus mehreren über Kommandozeilenaufrufe benutzbare Einzelprogrammen besteht, ist die Einbindung über die Broker-Schnittstelle von SWING möglich. [8], [5]
ht:Dig	Hierbei handelt es sich um eine vollständige Suchmaschine, die in Mailing-Listen als Alternative zu Harvest vorgeschlagen wurde. Informationen über den Einsatz einer Komponente von *ht:dig* als Indizierer in Harvest wurden nicht gefunden.
mg-Search	keine Informationen gefunden
Isearch	*Isearch* ist ein kleineres Suchwerkzeug. Es ist für Unix und Linux verfügbar. Aus dem ursprünglichen *Isearch* haben sich scheinbar verschiedene, zum Teil kommerzielle, Produktlinien entwickelt. Es scheint nur in geringem Maße Weiterentwicklung und Support für dieses Produkt zu geben. Hinweise auf eine Integration in Harvest als Indizierer konnten nicht gefunden werden. [9], [10]
mnoGoSearch (früher *UdmSearch*)	Hierbei handelt es sich um eine vollständige Suchmaschine. *mnoGoSearch* ist unter UNIX frei und unter Windows kostenpflichtig verfügbar. Eine Besonderheit von *mnoGoSearch* ist, dass anstelle von invertierten Indexdateien eine Datenbank benutzt wird. Mögliche Datenbanken sind zum Beispiel MySQL, POSTGRES, InterBase und Oracle. Außerdem kann die ODBC-Schnittstelle zum Datenzugriff benutzt werden. Die Verwendung einer Datenbank schließt die Verwendung des Indizierers in SWING nicht aus, allerdings erhöht sich die Komplexität der Anwendung und damit der Wartungsaufwand. Informationen zu einem Einsatz des Indizierers von *mnoGoSearch* in Harvest konnten nicht gefunden werden. [4]

Zebra	Scheinbar bezeichnen Zebra, *Isite* sowie *zquery* das gleiche Produkt. Zebra ist wahrscheinlich nicht Opensource. Die Nachfolgerversion *Z'mbol* ist kommerziell. Ab 1999 ist die volle Funktionalität zum Indizieren nur noch in *Z'mbol* enthalten.

Abb.5: frei verfügbare Indizierer und Suchmaschinen

Wie in der obigen Tabelle ersichtlich ist, konnten nicht zu allen Indizierern ausreichende Informationen zusammengetragen werden. Zum Teil war nicht klar, auf welche Version sich die Informationen beziehen. Die Informationen waren teilweise widersprüchlich und veraltet.
Als geeigneter Indizierer für den Ersatz von Glimpse in SWING ergibt sich hier *Managing Gigabytes*. Im Anhang auf Seite 46 ist ein Überblick über den durch MG gebotenen Funktionsumfang gegeben.
Der nächste Abschnitt fasst die geeigneten Indizierer zusammen und begründet die Auswahl.

2.1.2.4 Geeignete Indizierer

Nach der Auswertung der vorbereiteten und der frei verfügbaren Indizierer verbleiben als geeignete Programme für einen Ersatz von Glimpse in SWING somit SWISH-E und MG.

Indizierer	**aktuelle Version**
SWISH-E	2.05
MG	1.2.1

Abb.6: Indizierer-Alternativen für Glimpse

Aufgrund der vorbereiteten Schnittstelle, des aktiven Supports und der im Vergleich zu Glimpse zu erwartenden besseren Laufzeit wurde SWISH-E als Alternative für Glimpse gewählt.
Da bei der Integration von SWISH-E verschiedene Probleme auftraten (siehe Beschreibung von SWISH-E im Kapitel 2.1.4.1), wurde nach der Integration von SWISH-E auch *Managing Gigabytes* als Indizierer in SWING integriert.
Die unterschiedliche Funktionalität und der unterschiedliche Aufbau der Indizierer erfordern eine Anpassung der Schnittstellen und Programme in SWING. Diese Aspekte sind im Weiteren beschrieben.

2.1.3 Integration der Indizierer in SWING

Aus der Integration der ausgewählten Indizierer in die Suchmaschine SWING ergibt sich ein unterschiedlicher Änderungsbedarf. Da die Schnittstelle zum Indizierer SWISH bereits in SWING enthalten ist, sind für SWISH-E nur geringe Änderungen an der Schnittstelle zum Broker erforderlich. Im Gegensatz dazu muss für den Indizierer MG die Schnittstelle zum Broker neu geschrieben und in diesen integriert werden. Im nächsten Abschnitt werden die notwendigen Änderungen abgeleitet.

2.1.3.1 Notwendige Änderungen

Der zentrale Bestandteil der Anfragekomponente der Suchmaschine SWING ist der Broker. Die Integration eines Indizierers in die Anfragekomponente erfolgt über eine spezielle Schnittstelle des Brokers, die Indizierer-Schnittstelle. Diese Schnittstelle besteht aus 12 Routinen, die für die Initialisierung, die Indizie-

rung der Daten, die Ausführung der Anfrage und die Weiterleitung der Anfrageergebnisse an den Broker verantwortlich sind. In der folgenden Abbildung ist die Anfragekomponente von SWING mit den für die Aufgabenstellung relevanten Schnittstellen dargestellt.

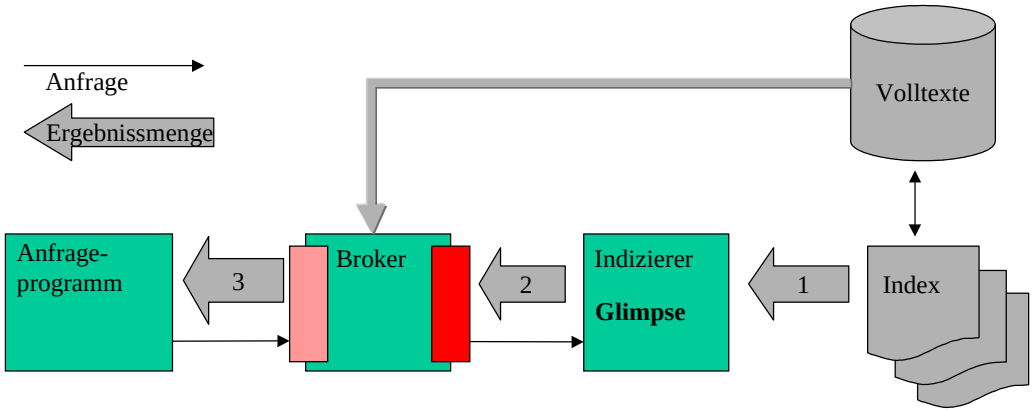

Abb.7: Anfragekomponente der Suchmaschine SWING mit Schnittstellen

Für die folgenden Indizierer Glimpse, GRASS, Nebula, PLWeb, Swish und WAIS ist diese Schnittstelle ab der Harvest Version 1.5.20 bereits vorbereitet. Bei der Verwendung eines dieser Indizierer ergibt sich kein oder nur geringer Änderungsaufwand. Die notwendigen Anpassungen ergeben sich aus den Besonderheiten von SWING bzw. aus Änderungen neuerer Versionen des Indizierers, die in der Schnittstelle noch nicht berücksichtigt sind.

Da der in Harvest standardmäßig verwendete Indizierer Glimpse zu den gefundenen Dokumenten keinen Gewichtungswert liefert, wurde in SWING u.a. ein Ranking und eine dementsprechende Aufbereitung der Anfrageergebnisse integriert. Dies erfolgte im Anfrageprogramm. Dieses Programm nimmt den vom Broker gelieferten Datenstrom der Anfrageergebnisse entgegen und erstellt daraus die HTML-Seiten zur Anzeige der Treffer. Außerdem führt es ein Ranking und eine dementsprechende Sortierung der Anfrageergebnisse durch. Ein Ziel des Austausches des Indizierers war es, einen Teil dieser Funktionalität durch den neuen Indizierer ausführen zu lassen. Da dies in dem vorkompilierten Programm i.d.R. schneller erfolgt und zum Teil bereits bei der Erstellung des Index und nicht während der Anfrageausführung erfolgen kann. Durch die Bestimmung eines Gewichtungswertes für die Treffer im Indizierer und die Übergabe dieses Wertes an das aufrufende Programm, muss zusätzlich zur Indizierer-Schnittstelle auch die Schnittstelle zwischen dem Broker und dem Anfrageprogramm (Anfrage-Schnittstelle) angepasst werden. Nach der erfolgten Änderung wird im Datenstrom zusätzlich der Gewichtungswert übergeben.

Im Folgenden wird auf die Änderungen an der Schnittstellen zwischen dem Indizierer und dem Broker eingegangen.

2.1.3.2 Anpassung der Indizierer-Schnittstelle

Durch den Austausch der Indizierer-Komponente der Suchmaschine ergibt sich die Notwendigkeit zur Anpassung der Indizierer-Schnittstelle sowie der Anfrage-Schnittstelle[1].

Die folgende Grafik zeigt die Schnittstelle zwischen dem SWING-Broker und dem Indizierer, die durch die Integration eines alternativen Indizierers betroffen ist und angepasst werden muss.

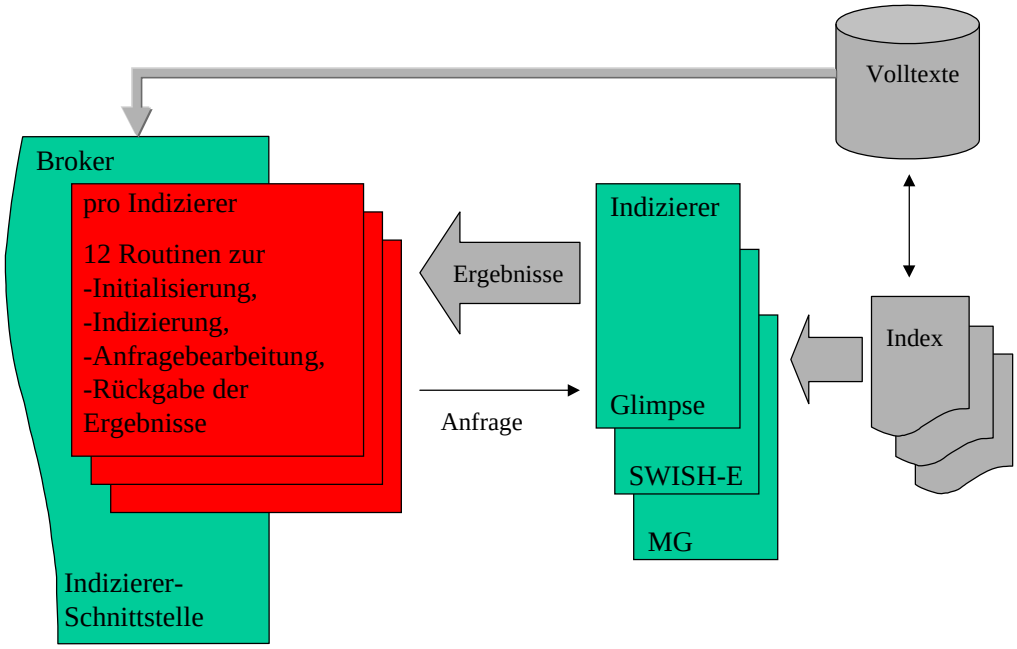

Abb.8: Indizierer-Schnittstelle

Dabei meint Anpassung vor Allem das Ausprogrammieren der Schnittstellenroutinen, sodass die im Bild und der folgenden Tabelle dargestellten funktionalen Zusammenhänge realisiert werden.

Für SWISH-E müssen die in der vorhandenen Schnittstelle befindlichen Routinen für zusätzliche Parameterübergaben[2] und die Übergabe[3] der temporären Ergebnisdatei geringfügig angepasst werden.

Für MG ist keine Ausprägung der Schnittstelle in Harvest vorhanden und ist damit für SWING zu erstellen und in den Broker zu integrieren.

Die folgende Tabelle enthält die anzupassenden Schnittstellenroutinen. Je nach Komplexität, muss nur ein Teil der Routinen ausprogrammiert werden

[1] Diese Anpassungen werden im nächster Abschnitt beschrieben.

[2] Beschränkung der Ergebnismenge, Verwendung Konfigurationsdatei

[3] Änderungen beim Auslesen, bei der Gewichtung, u.a.

Routinen[1]	**Glimpse**	**SWISH**	**MG**	**Bemerkung**
Initialisierung				
x_IND_initialize				Vorbelegung von Variablen
x_IND_config				setzen von Variablen aus Konfigurationsdatei
x_IND_Init_Flags				Variablen entsprechend der broker.conf Datei setzen
x_IND_Set_Flags				Vorbelegung von Variablen
Indizierung				
x_IND_Index_Start	X	X	X	Initialisierung für Indizierung
x_IND_Index_Flush	X	X	X	Indizierung entsprechend der gesetzten Indizierungsvariante (beendet evtl. die Indizierung)
x_IND_New_Object	X			einzelne Datei dem Index hinzufügen
x_IND_Destroy_Obj				einzelne Datei aus dem Index löschen
x_IND_Index_Full	X	X	X	Indizierung der gesamten Datenmenge
x_IND_Index_Incremental	X			Indizierung neuer bzw. aktualisierter Dateien
Anfragebearbeitung				
x_IND_do_query	X	X	X	Anfrage ausführen Ergebnismenge annehmen, formatieren, bearbeiten und an Broker übergeben
QM_user_object Call-Back-Routine zur Rückgabe der Ergebnismenge	X	X	X	Die vom Indizierer auf die Anfrage gelieferte Ergebnismenge wird mittels dieser Routine Objektweise an den Broker zurückgeliefert. Neben der ObjektID können zusätzliche Daten übergeben werden. Diese werden an das den Broker aufrufende Programm weitergeleitet.

Abb.9: Schnittstellenroutinen

Im Broker werden die Schnittstellenroutinen über ein mehrdimensionales Funktionspointerarray angesprochen. Dieses wird statisch, also zum Kompilationszeitpunkt erzeugt. Dabei kann der Broker für die gleichzeitige Benutzung von Glimpse, SWISH und Wais kompiliert werden. Grass, Nebula und PLWeb können nur einzeln und alternativ zu allen anderen Indizierern in den Broker eingebunden werden.
Die Anfrageausführung durch den Indizierer kann u.a. über einen Kommandozeilenaufruf erfolgen. In diesem Fall wird die Ergebnismenge in eine Temporärdatei gespeichert. Diese wird durch den Broker geladen und ausgewertet. Dabei werden pro gefundenem Treffer der Objekt-Identikator und evtl. zusätz-

[1] Das x im Namen der Routine steht für den jeweiligen Indizierer (MG, SWISH, Glimpse, ...).

liche Daten wie die Trefferstellen ausgelesen. Die Tabelle Abb.39 im Anhang Kapitel 4.1 enthält Beispiele für diese Aufrufe und Ausschnitte aus der übergebenen Ergebnismenge.
Im Unterschied zum Indizierer Glimpse, liefern die Indizierer SWISH-E und MG keine Trefferstellen in der Ergebnismenge. Dies wirkt sich auf die Berechnung der Gewichtung der Treffer in SWING und auf die Anzeige aus.
Ein weiterer Unterschied ist, dass SWISH-E und MG einen Gewichtungswert für den Treffer übergeben, und SWISH-E zusätzlich die Dateigröße liefert. Der Gewichtungswert in SWISH-E und MG wird aufgrund der Verteilung und der Häufigkeit des Suchbegriffs bestimmt (siehe Beschreibung der Indizierer im Kapitel 2.1.4.1 und 2.1.5.1). Zur Benutzung dieses Wertes muss die Auswertung der Übergabedatei in der Schnittstelle des Brokers angepasst werden. Der vom Indizierer ermittelte und in der Ergebnismenge übergebene Gewichtungswert wird anstelle der Trefferstelle als Nutzerdaten an den Broker übergeben[1]. Durch den Broker wird der Wert an das aufrufende Anfrageprogramm übergeben und im dortigen Ranking berücksichtigt.
Im folgenden Abschnitt sind die notwendigen Änderungen an der Anfrage-Schnittstelle aufgeführt.

2.1.3.3 Anpassung der Anfrage-Schnittstelle und des Anfrageprogramms

Durch die Übergabe eines Gewichtungswertes vom Broker an das aufrufende Programm wird die im Folgenden abgebildete Anfrage-Schnittstelle nicht direkt beeinflusst. Jedoch wirkt der Austausch des Indizierers auf das die Daten weiterverarbeitende Programm.

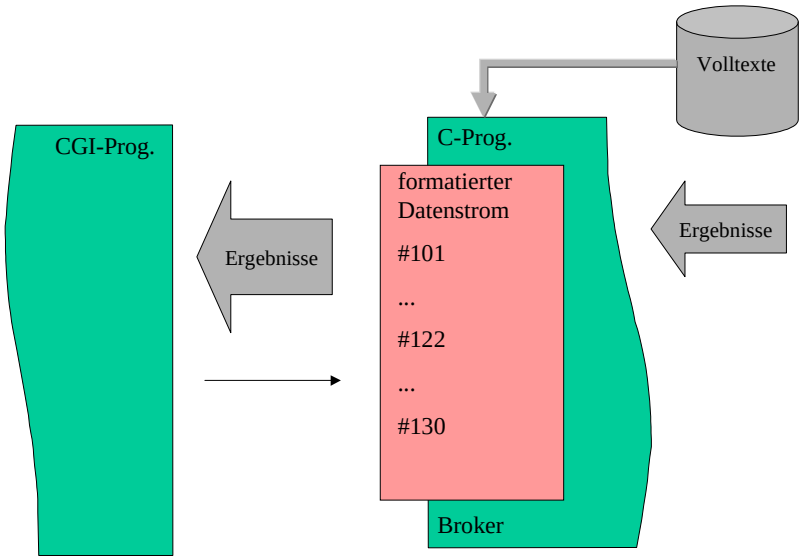

Abb.10: Anfrage-Schnittstelle

Die Übergabe der Anfrage und der Anfrageergebnisse erfolgt als formatierter Datenstrom über eine Socketkommunikation. Die Übergabe der Ergebnismenge erfolgt unter Zuhilfenahme der in der Tabelle

[1] Diese Variante wurde gewählt um den Umsetzungsaufwand möglichst gering zu halten.

Abb.40 im Anhang 4.2 aufgeführten Trennzeichen. Das Zeichen *#122* übergibt die nutzerdefinierten Daten.
Bei Nutzung von Glimpse als Indizierer werden über das Trennzeichen *#122* die Trefferstellen übergeben. Bei der Verwendung von SWISH-E und MG wird dieses Trennzeichens, da beide Indizierer keine Trefferstellen liefern, zur Übergabe des Gewichtungswert benutzt. Neben der Berücksichtigung des übergebenen Gewichtungswertes im Rankingalgorithmus des Anfrageprogramms, muss die Auswertung und Anzeige der Trefferstelle entfallen.
Für das Anfrageprogramm (*nph-search*) ergeben sich damit folgende Änderungen.

- Anpassung des Ranking-Algorithmus, aus- und einschalten der Rankingmethoden für *match_line*, *metatag_neu* und *ohp* über *swishflag=on|off*
- Das Ranking-Log wird nicht mehr geschrieben, da das Ranking im Indizierer und im Broker ausgeführt wird.
- Änderung der Algorithmen die speziell auf Glimpse abgestimmt sind.
- Änderung der Anzeige. Diese Änderung ist notwendig da z.B. Trefferstellen nicht mehr zur Verfügung stehen.

Außerdem wurde in der zweiten Bearbeitungsphase die Sortierung vollständig in den Indizierer verlegt. Die Gruppierung der Anfrageergebnisse nach dem Dokumententyp wurde aus dem Anfrageprogramm entfernt, da hierfür Sortierung und Textparsing notwendig sind.
Im folgenden Abschnitt werden die für den jeweiligen Indizierer notwendigen Änderungen und die Besonderheiten der Programme beschrieben.

2.1.4 Integration von SWISH-E

2.1.4.1 Beschreibung von SWISH-E

SWISH-E[1] ist ein unter verschiedenen Betriebssystemen verfügbares Programm, das im C-Quelltext vorliegt. Es wird über die Kommandozeile gestartet. Die Programmsteuerung erfolgt über Programmparameter und über eine Konfigurationsdatei. Die Indizierungs- und Anfragekomponenten von SWISH-E können in einem Programm oder getrennt voneinander betrieben werden. Als Index benutzt SWISH-E eine invertierte Datei. Im Gegensatz zu Glimpse, ist der gesamte Index in einer einzigen Datei abgelegt. Beim Start der Indizierung wird SWISH-E das Verzeichnis der Datenquellen und der Name der zu erzeugenden Indexdatei übergeben. Bei der Anfrageausführung benötigt SWISH-E den Namen der Indexdatei.
Bei der Indizierung erzeugt SWISH-E eine Hash-Struktur für den Index. Aufgrund der Verwendung eines Hash, können Anfragen nach ganzen Wörtern relativ schnell beantwortet werden. Anfragen mit Truncation sind in der Version 1.3.X deutlich langsamer, da die Zugriffsvorteile des Hash scheinbar nicht genutzt werden können.

[1] vgl. Anhang 4.10

Bei der Indizierung mit SWISH-E traten eine Reihe von Problemen auf, die dazu führten, dass nacheinander die Versionen 1.3.5, 2.05, 2.1-dev20 und 2.1-dev22 eingesetzt wurden. SWISH-E benötigt für die Indizierung sehr viele Ressourcen. Die Indizierung von ca. 82.000 Dateien mit ungefähr 330 MB Speicherplatz mit SWISH-E 2.05 war auf einer SUN Ultra-1 mit 255 MB verfügbarem Hauptspeicher nicht ausführbar. Auf einer SUN Ultra-1 mit 512 MB Hauptspeicher dauerte der Prozess ca. 1,5 Tage. Um die Indizierungszeiten zu verkürzen, wurde versucht eine möglichst große Stop-Wortliste und Stemming zu benutzen. Die Größe des verfügbaren Hauptspeichers hat einen wesentlichen Einfluss auf die Laufzeit der Indizierung. Die Version SWISH-E 2.1-dev22 benötigt für die Indizierung von 81466 Dateien (ca. 465 MB Speicherplatz) mit 870899 Stichworten 215 Minuten auf der gleichen SUN Ultra-1 mit 512 MB Hauptspeicher.
Die Berechnung der Gewichtungswerte erfolgt in SWISH-E in 2 Varianten.

$$rank = \frac{\log(\max(freqInFile, 5)) + 10}{freqInAllFiles} \times 1000$$

Abb 11: Rankberechnung in SWISH-E 2.05 ohne Berück sichtigung der Größe des Dokuments

bzw.

$$rank = \frac{\left(\frac{\log(\max(freqInFile, 5)) + 10}{freqInAllFiles}\right)}{numWordsInFile} \times 1000$$

Abb 12: Rankberechnung in SWISH-E 2.05 mit Berück sichtigung der Größe des Dokuments

Über eine Konfigurationsdatei kann eingestellt werden welche Formel verwendet wird. Die Bezeichner in den Formeln haben die folgende Bedeutung.

freqInFile	ist die Häufigkeit des Suchwortes im Dokument.
freqInAllFiles	ist die Häufigkeit des Suchwortes in der Dokumentensammlung.
numWordsInFile	ist die Dokumentgröße ausgedrückt in der Anzahl Wörter im Dokument.

Im Folgenden werden die für die Integration von SWISH-E notwendigen speziellen Änderungen aufgeführt.

2.1.4.2 Notwendige Änderungen

SWISH-E bezieht eine Reihe von Einstellungen aus einer eigenen Konfigurationsdatei. Der vollständige Name dieser Datei sollte bei allen Aufrufen des Indizierers übergeben werden. Als Ablageort bietet sich das *admin*-Verzeichnis des Brokers an, da dort weitere Einstellungsdateien des Brokers abgelegt sind und sich die Einstellungen des Indizierers von Broker zu Broker unterscheiden können.

Die von SWISH-E übergebene Ergebnisdatei unterscheidet sich von der Variante[1] die in der Schnittstelle für SWISH und Glimpse vorbereitet ist. Bei der Übernahme der von SWISH-E gelieferten Ergebnismenge müssen daher der Gewichtungswert und die Dateigröße berücksichtigt werden.
Der folgende Abschnitt beschreibt MG und die für die Integration von MG notwendigen Änderungen.

2.1.5 Integration von MG

2.1.5.1 Beschreibung von MG

MG[2] ist eine Sammlung aus Skript-Dateien und C-Programmen, die ebenfalls als Quellen vorliegen. Bei der Entwicklung von MG ist besonderer Wert auf die Möglichkeit zum Umgang mit großen Datenmengen gelegt worden [5]. MG bietet mehrere Möglichkeiten für Abfragen zur Bestimmung der Gewichtungswerte, und ebenfalls für die Übergabe der Abfrageergebnisse.
Die Formel zur Berechnung des Gewichtes eines Dokuments in MG leitet sich aus der Ähnlichkeitsberechnung zwischen dem Anfragevektor und dem Dokumentenvektor mittels des Kosinus ab (Vektorraummodell).

$$\text{cosine}(Q,D_d) = \frac{1}{W_d W_q} \times \sum_{t \in Q \cap D_d} w_{d,t} \times w_{q,t} = \frac{1}{W_d W_q} \times \sum_{t \in Q \cap D_d} (r_{d,t}) \times (r_{q,t} \times w_t)$$

Abb 13: Ähnlichkeitsberechnung zwischen Query- und Dokumentenvektor

Da die Gewichtung der Query W_q keinen Einfluss auf die Reihenfolge der Dokumente hat und in allen Werten als Konstante auftritt, wurde W_q bei der Berechnung des Gewichtungswertes in MG weggelassen. In der obigen Formel wird

$r_{q,t} = 1$, $w_t = \log_e\left(1 + \frac{N}{f_t}\right)$, $r_{d,t} = \left(1 + \log_e f_{d,t}\right)$ und $W_d = \sqrt{\sum_{t=1}^{n} w_{d,t}^2}$ gesetzt.

Mit Berücksichtigung der genannten Belegungen lautet die Formel zur Berechnung eines Gewichtungswertes in MG wie in der folgenden Abbildung dargestellt.

$$\text{cosine}(Q,D_d) = \frac{1}{W_d} \times \sum_{t \in Q \cap D_d} \left(1 + \log_e f_{d,t}\right) \times \log_e\left(1 + \frac{N}{f_t}\right)$$

Abb. 14: Berechnung des Gewichtungswertes in MG

[1] siehe Abbildung Abb.39 im Anhang 4.1

[2] vgl. Anhang 4.10

Die folgende Auflistung enthält die in den Formeln verwendeten Bezeichner.

D_i	i-tes Dokument in der Datenbasis (Dokumentensammlung)
t	Term
f_t	Anzahl an Dokumenten der Datenbasis die t enthalten
N	Anzahl an Dokumenten in der Datenbasis
$f_{d,t}$	Häufigkeit des Auftretens von t im Dokument D_d
$r_{d,t}$	relative Häufigkeit des Auftretens von t im Dokument D_d, (relative term frequency)
w_t	Gewicht des Terms t, (inverse Dokumentenhäufigkeit)
w_q	Gewicht der Anfrage q
w_d	Gewicht des Dokumentes d

Die Indizierung von ca. 243 MB Daten benötigte auf einer Sun Ultra-1 mit 512 MB verfügbarem Hauptspeicher knapp 5 Stunden.
Im Folgenden werden die für die Integration von MG notwendigen speziellen Änderungen aufgeführt.

2.1.5.2 Notwendige Änderungen

Beim Start der Indizierung wird dem Indizierer ein Alias für das Verzeichnis der Datenquellen sowie ein Alias für das Verzeichnis der zu erzeugenden Datenbasis und des Index übergeben. Um die Datenmenge bei der Übergabe der Treffer gering zu halten, wird MG mit der Einstellung *mode heade* [1] betrieben. Da bei dieser Übergabevariante der Ergebnismenge keine Gewichtungswerte übergeben werden, musste eine Anpassung der entsprechenden Routinen in MG erfolgen. Der Index von MG ist nach dem Prinzip der invertierten Datei aufgebaut und ist in mehreren Dateien abgelegt.
Bei der Indizierung mit MG traten regelmäßig Probleme auf, wenn eine zu indizierende Datei sehr groß war. Die in der Dokumentation beschriebene Vergrößerung des Puffers brachte keine Verbesserung, sodass die zu indizierende Datenmenge auf Dateien von maximal ca. 3 MB beschränkt werden muss. Aufgrund dieser Einschränkung wurde der Betrieb eines Brokers mit MG als Indizierer nicht in die Produktion überführt.
Das Verzeichnis der Datenquellen und das Verzeichnis für die Indexdateien wird MG über die Umgebungsvariablen MGDATA und MG_GETRC übergeben.
Der folgende Abschnitt enthält eine kurze Bewertung der Gewichtungs-Formeln der betrachteten Indizierer.

2.1.6 Wertung des Ranking der Indizierer

Die Berechnung eines Gewichtungswertes für ein Dokument sollte mindestens die drei folgenden Aspekte berücksichtigen.

- Je häufiger das Suchwort im Dokument enthalten ist, desto höher wird das Dokument gewichtet.

[1] Im *mode headers* wird der Dateiname des gefundenen Dokumentes übergeben.

- Je seltener ein Suchwort in der Dokumentensammlung auftritt, desto höher ist sein Einfluss auf die Berechnung der Gewichtung eines Dokuments.
- Da lange Dokumente tendenziell mehr Worte enthalten als kurze und so stets höhere Gewichtungswerte erhalten, wird das Gewicht in Bezug auf die Dokumentengröße normalisiert.

Alle drei Aspekte werden sowohl durch SWISH-E als auch durch MG berücksichtigt. In der Anwendung stellte sich bei SWISH-E heraus, dass die berechneten Gewichtungswerte teilweise sehr eng beieinander lagen. Zum Teil wurden für alle 200 übergebene Treffer identische Gewichtungen berechnet. Dieses Verhalten resultiert aus einer zusätzlichen Normalisierung in SWISH-E.
Das Ziel der ersten Arbeitsetappe ist eine Laufzeitverbesserung in der Anfragephase zu erreichen. Im folgenden Kapitel wird ein Laufzeitvergleich angeführt, mittels dem die erzielte Verbesserung der Anfragezeiten überprüft wurde.

2.1.7 Laufzeitvergleich

Der im folgenden Kapitel beschriebe Laufzeitvergleich wurde nach der erfolgreichen Integration von SWISH-E als Indizierer in die Suchmaschine SWING ausgeführt, um festzustellen, ob die angestrebte Laufzeitverbesserung mit diesem Indizierer erreicht werden konnte. Der Laufzeitvergleich wurde unter Testbedingungen und anschließend unter Produktionsbedingungen mit dem Gesamtsystem ausgeführt. Das Testsystem bestand aus drei Brokern, die jeder einen anderen Indizierer verwendeten. Als Indizierer kamen Glimpse im gBroker, SWISH im sBroker und SWISH-E im seBroker zum Einsatz. Für MG sind ähnliche Werte zu erwarten.

2.1.7.1 Benutzte Anfragen

Für die Anfragen wurden Suchbegriffe aus der SwingBroker -Statistik für Februar 2001 ausgewählt. In der folgenden Tabelle ist ein Ausschnitt der Statistik enthalten.

Anfrage	**Häufigkeit**
Holz	94
Verlag	43
Gesetze	9
Oberfinanzdirektion	7
Stellenangebote	15
schwerin and tourismus	9

Abb.15: Ausschnitt aus der SwingBroker-Statistik für 02/2001

Außerdem wurden Kombinationen der Suchbegriffe mit booleschen Operatoren, Truncation und Anfragen zur strukturierten Suche ergänzt. Die zusätzlichen Suchbegriffe wurden aus den Dateien der Datenquellen des Brokers abgeleitet und so gewählt, dass die zu erwartende Ergebnismenge überprüft werden konnte. Die Messung fand an der Anfrage-Schnittstelle statt.

In der folgenden Tabelle[1] sind die Anfragen und die mit den drei Brokern dazu erreichten Antwortzeiten sowie die Anzahl der Elemente in der Ergebnismenge aufgelistet.

	Broker					
	gBroker		**sBroker**		**seBroker**	
Indizierer	Glimpse		SWISH		SWISH-E Version 1.3.5	
Anfragen	**T**	**N**	**T**	**N**	**T**	**N**
holz	16	50	12	273	6(8)	275
andreas and heuer	5-9	10	7	13	5-9	13
datenbank	7	34	11(21)	484	4(8)	470
datenbank*	18	29	21(45)	897	31(44)	858
schwerin and tourismus	7	23	11(60)	1701	11(24)	1657
stellenangebote	4	50	7(18)	244	6(9)	239
gesetzte and oberfinanzdirektion	7	1	7	1	3	1
gesetzte or oberfinanzdirektion	6(12)	49	8(11)	249	4	25
author:mediawelt	8	50	8	0	4	0
title:hansenet	7	50	7	0	3	0
author:antje	8	3	8	0	4	0
holz and verlag	7	17	8	25	4	18
holz and not verlag	7	17	10	245	7(14)	256

Abb.16: Laufzeitvergleich für 3 Broker mit unterschiedlichen Indizierern

Wie in der Tabelle ersichtlich, führt der Einsatz von SWISH-E als Indizierer bei einfachen booleschen Anfragen teilweise zu einer Laufzeitverbesserung um bis zu 50%. Bei Anfragen mit Truncation, z.B.: datenbank*, ist SWISH-E langsamer als Glimpse. Der Grund für dieses bei der Version 1.3.5 beobachtete Verhalten liegt darin, dass SWISH-E als Datenstruktur für die Suche eine Hash-Tabelle verwendet. Wahrscheinlich kann SWISH-E bei Truncation nicht über die Hash-Struktur zugreifen. Strukturierte Anfragen auf Dateien im SOI-Format kann SWISH-E nicht ausführen. Der Grund dafür ist, dass SWISH-E die SOIF-Felder nicht auswerten kann. Die gesamte Datei muss somit als Plain[2]-Text indiziert w erden. SWISH-E kann als geeigneter Indizierer in SWING für einfache boolesche Anfragen bestätigt werden.
Der folgende Abschnitt beschreibt den Laufzeitvergleich um Rückschlüsse über seine Aussagekraft zu ziehen.

[1] Die T- Spalten enthalten die Laufzeit der Anfrage und die N- Spalten die Anzahl der Ergebnisse. Die Darstellung *a - b* wurde verwendet, wenn die ermittelten Werte auf einen Bereich verteilt waren. In der Darstellung c*(d)* ist c der regelmäßig gemessene Wert und d ein maximaler Ausreißer.

[2] einfacher Text ohne Struktur

2.1.7.2 Wertung des Laufzeitvergleichs

Ziel des Vergleichs ist es, die Annahmen über eine Laufzeitverbesserung und die korrekte Arbeit des Indizierers innerhalb der Suchmaschine zu prüfen. Dazu wurden drei Broker auf einem Rechner, einer Sun Ultra 4 mit 1024 MB Hauptspeicher, 4 CPU's, SunOS 5.6 und sparc CPUType installiert. Jeder der Broker arbeitete auf einer zur SwingBroker-Datenbasis äquivalenten Datenmenge. Aufgrund des relativ großen Speicherplatzbedarfs mussten die Installation und der Laufzeitvergleich für den gBroker und sBroker getrennt vom seBroker erfolgen. Dieses und der parallel weiterlaufende Betrieb anderer Anwendungen auf dem Rechner, führte zu Abweichungen in den Messergebnissen. Die Ableitung der eher prinzipiellen Aussagen, ob der Einsatz eines alternativen Indizierers in SWING möglich ist und eine Laufzeitverbesserung erreicht werden kann, wird durch die Berücksichtigung von unmittelbar vor dem Test gemessene Umgebungsparametern gewährleistet. Die Tabelle Abb.41 im Anhang Seite 41 enthält einige dieser Werte.
Der mit dem Indizierer SWISH arbeitende Broker, erzielte regelmäßig schlechtere Ergebnisse, als der mit SWISH-E arbeitende Broker. Daher bestätigte dieser Test auch die Verwendung des aktuelleren SWISH-E anstelle von SWISH.
Im folgenden Abschnitt wird der Laufzeitvergleich an der für den Nutzer relevanten Stelle des Web-Interface dargestellt.

2.1.7.3 Vergleich der Laufzeiten im Einsatz

Außer dem Indizierer wirken weitere Komponenten auf die zu erreichenden Antwortzeiten ein. Die für den Nutzer relevante Stelle an der die Laufzeit gemessen werden muss, ist sein Web-Interface der Browser. Daher wurde im Anschluss an den im letzten Kapitel geführten Laufzeitvergleich und nach Ausführung der Änderungen am Anfrageprogramm, ein weiterer Vergleich der Laufzeiten geführt. Gemessen wurden die Antwortzeiten an einem Web-Browser. Die Vergleichs-Broker liefen auf einer SUN Ultra-1 mit 512 MB Hauptspeicher, einer CPU, SunOS 5.6 und sparc CPU Typ.
Wie die Tabelle Abb.42 im Anhang zeigt, ist die Datenbasis beider Broker annähernd gleich groß, was ausschließt, dass die Antwortzeiten durch unterschiedlich große Datenbasen bestimmt wurden.
Um ermessen zu können, inwiefern die Änderung der Antwortzeiten durch den Austausch des Indizierers beeinflusst wird, wurden alle Anfragen zusätzlich direkt an den Indizierer gestellt. Die folgende Tabelle[1] enthält die erreichten Antwortzeiten und die Anzahl der Ergebnisse.

[1] Die Δ- Spalte enthält die prozentuale Laufzeitänderung. Ein negativer Wert in dieser Spalte zeigt an, dass SWISH-E bei dieser Anfrage langsamer arbeitete als Glimpse. Die Zeitangabe erfolgt in Sekunden. Da durch SWISH-E die Feldsuche auf SOIF-Dateien nicht direkt unterstützt wird, enthalten die entsprechenden Felder keinen Wert.

Fragen	**SwingBroker** (Glimpse)		**SwingBrokerS** (SWISH-E)		**Δ** (%)	**Indizierer** (direkt)	
	T	N	T	N		Glimpse	SWISH-E
holz	26	162	21	196	19,2	19	7
andreas and heuer	22	56	10	12	54,5	6	6
datenbank	22	102	25	196	- 13,6	16	8
datenbank*	20	106	92	189	- 360,0	10	91
schwerin and tourismus	26	62	38	198	- 46,2	63(75)	30
stellenangebote	26	166	22	200	15,4	- 4	5
gesetzte and oberfinanzdirektion	13	6	7	1	46,2	7	5
gesetzte or oberfinanzdirektion	14	6	8	27	42,9	5	4
author:mediawelt	21	200	9	0	-	-	-
title:hansenet	26	201	6	0	-	-	-
author:antje	15	10	5	0	-	-	-
holz and verlag	25	60	11	26	56,0	9	5
holz and not verlag	33(44)	165	30(47)	196		29/14	14

Abb.17: Laufzeitvergleich zwischen SWISH-E und Glimpse in SWING

Wie in der obigen Tabelle ersichtlich führt der Einsatz von SWISH-E gegenüber Glimpse bei einfachen Anfragen zu einer Laufzeitverbesserung. Bei der Masse, der an die Suchmaschine gestellten Anfragen, handelt es sich um Anfragen mit einem Suchbegriff[1]. Die erreichte Beschleunigung beträgt für diese Anfragen 10-20% und wird bei der Nutzung der Suchmaschine kaum empfunden. Im Gegensatz dazu liefert der direkte Vergleich der beiden Indizierer zum Teil Laufzeitverbesserungen von ca. 50%.

Die für die gesamte Suchmaschine deutlich geringer ausfallende Performancesteigerung resultiert aus dem modularen Aufbau und der Aufgabenteilung innerhalb der Anfragekomponente der Suchmaschine. Dies soll in dem folgenden Beispiel verdeutlicht werden.

Der SWING-Broker mit dem Indizierer Glimpse liefert auf die Anfrage *holz and Verlag* nach 25s das Anfrageergebnis. Der direkte Aufruf von Glimpse liefert die Ergebnismenge nach 9s. SWISH-E arbeitet fast 50% schneller als Glimpse und beantwortet die Anfrage nach 5s. Das bedeutet eine absolute Laufzeitverbesserung von 4s. Der Broker kann folglich mit SWISH-E als Indizierer die Anfrage nach 21s beantworten. Für den Broker ergibt sich damit aber nur eine relative Verbesserung der Antwortzeit um 16%. Diese Ergebnisse führen zu den folgenden Schlussfolgerungen.

2.1.8 Schlussfolgerung aus dem Laufzeitvergleich

Neben dem Einsatz eines schnelleren Indizierers muss die Aufgabenverteilung innerhalb der Anfragekomponente der Suchmaschine beeinflusst werden, um eine Beschleunigung der Anfrageausführung zu

[1] Die durchschnittliche Wortanzahl lagt von Februar bis August 2001 im Bereich von 1,17 bis 1,23 Wörtern pro Suchanfrage (siehe SWING- Statistik im Anhang).

erreichen. Für SWING bedeutet das eine Änderung der Aufgabenverteilung zwischen dem Anfrageprogramm, dem Broker und dem Indizierer. Dabei müssen Aufgaben die in der Anfragephase ausgeführt werden in den Indizierer und in die Indizierungsphase verlagert werden. Diese Änderungen wurden in der zweiten Arbeitsetappe konzipiert und umgesetzt.

2.2 Zweite Arbeitsetappe: Änderung der Aufgabenverteilung, Integration von Rankingfunktionen

2.2.1 Ausgangszustand

Die bisherige Arbeitsweise von SWING ist im Abschnitt Ausgangszustand der ersten Arbeitsetappe erläutert worden. In diesem Abschnitt werden im Wesentlichen das Ranking in SWING und die damit verbundenen Aufgaben und ihre Verteilung in der Suchmaschine betrachtet.

Ein wesentliches Problem für die Laufzeit bei der Anfragebearbeitung in SWING ist, dass die Sortierung, Gruppierung und das Ranking im Anfrageprogramm erfolgt, was in der folgenden Abbildung dargestellt wird.

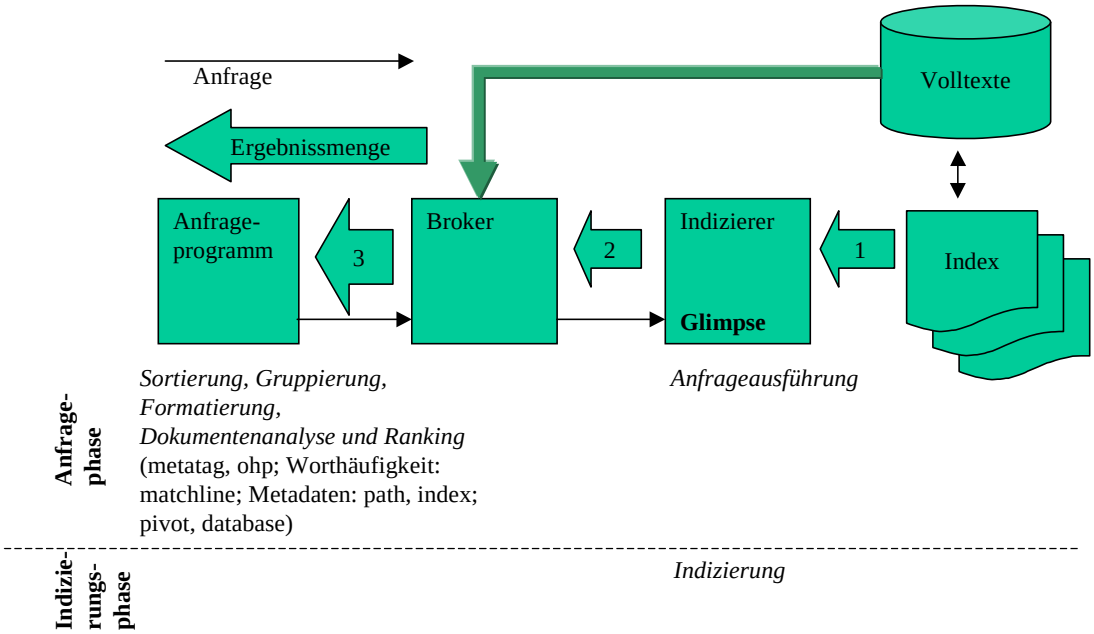

Abb.18: ursprüngliche Aufgabenverteilung in der Anfragekomponente

Diese Aufgabenverteilung resultiert vor allem aus der Verwendung von Glimpse als Indizierer. Glimpse liefert die Treffer ohne Gewichtung, und daher auch nicht entsprechend der Gewichtung sortiert.

Da das standardmäßig verwendete Anfrageprogramm ein Perl-Programm ist, und interpretierend ausgeführt wird, erfolgt die Bearbeitung hier i.d.R. langsamer als in einem vorkompilierten Programm. Aufgrund der Tatsache, dass das Ranking erst nach dem Broker stattfindet, kann von der Beschränkung der Anzahl der Anfrageergebnisse nicht Gebrauch gemacht werden. Da Treffer mit höherer Gewichtung nicht als erstes übergeben werden, würde ein Abschneiden der Ergebnisliste gegebenenfalls zum Verlust der

besten Treffer führen. Außerdem fließen in die Berechnung Zwischenergebnisse ein, die bereits in der Indizierungsphase berechnet werden könnten.

Durch den Einsatz von SWISH-E bzw. MG in der ersten Arbeitsetappe ist es zu einer Verlagerung der Aufgaben gekommen. Diese veränderte Aufgabenverteilung ist in der folgenden Abbildung dargestellt.

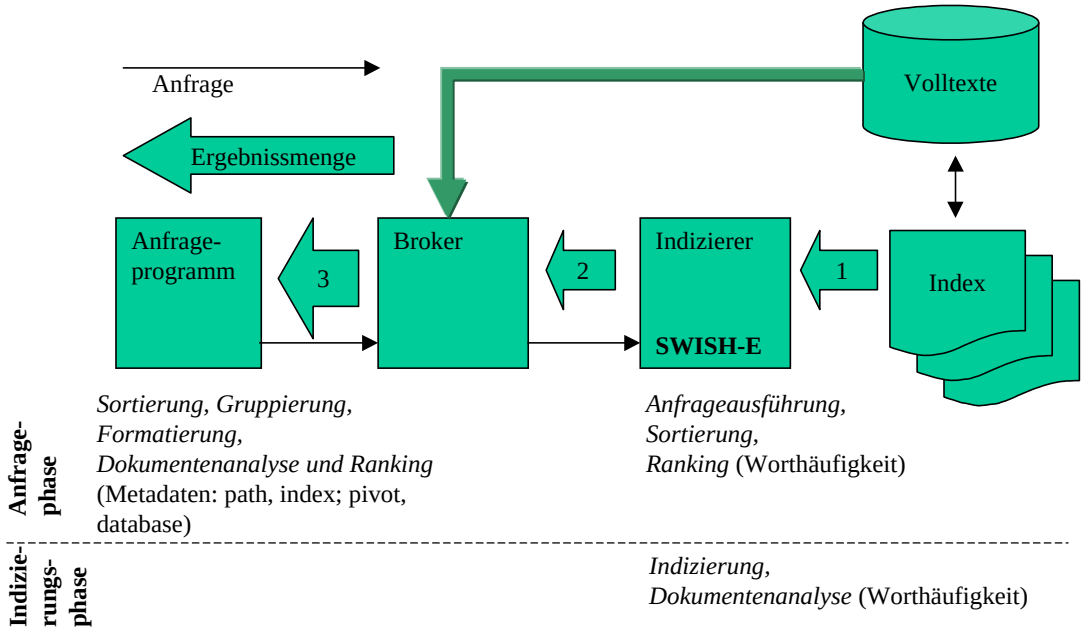

Abb.19: Aufgabenverteilung in der Anfragekomponente beim Einsatz von SWISH-E

Ein Teil des Ranking wird nach der Änderung durch den Indizierer und ein Teil durch das Anfrageprogramm ausgeführt. Insbesondere die Berechnung eines Gewichtungswertes über die Worthäufigkeit und die Bestimmung der dafür notwendigen Daten, in SWING bisher in *matchline* realisiert, erfolgt durch den Indizierer und in der Indizierungsphase. Die Berechnung eines Gewichtungswertes über das Vorkommen des Suchwortes im Titel oder in einem Metatag entfällt, da beide Algorithmen die Auswertung der Trefferstellen in der Anfragephase erfordern, SWISH-E und MG[1] aber keine Trefferstellen liefern.

Der in den vorigen Abschnitten beschriebene Laufzeitgewinn resultiert aus der schnelleren Bestimmung der Ergebnismenge und Rankingwerte durch den Indizierer, aus der Verlagerung eines Teils der Auswerteroutinen in die Indizierungsphase (Bestimmung Ranking-Daten) und aus dem Wegfall der Ranking-Routinen match_line, metatag sowie ohp im Anfrageprogramm.

Im nächsten Abschnitt wird eine Aufgabenverteilung abgeleitet, die weitere Funktionen in den Indizierer bzw. in die Indizierungsphase verlegt.

[1] mode headers

2.2.2 Ableitung einer geänderten Aufgabenverteilung

Wie man aus Tabelle Abb.20 in Zusammenhang mit der Architektur der Anfragekomponente und der Aufgabenverteilung in ihr ersehen kann, nimmt das Anfrageprogramm einen nicht unwesentlichen Teil der verbrauchten Laufzeit für sich in Anspruch.

	Web-Browser	**Broker**[1]
holz	16 (11)	≥8 (≥4)
datenbank	15 (12)	≥7 (≥4)
schwerin and tourismus	12 (10)	≥8 (≥6)
holz and verlag	12 (10)	≥8 (≥6)

Abb.20: Verteilung der Laufzeiten auf Bestandteile der Anfragekomponente: Anfrageprogramm und Broker

Durch das Anfrageprogramm wird die Zeit vor allem zur Erstellung der HTML-Seiten, zum Ranking, zur Gruppierung, zur Sortierung und zur Formatierung der einzelnen Anfrageergebnisse benötigt.
Eine weitere Verbesserung der Antwortzeiten kann erreicht werden, wenn weitere der durch das Anfrageprogramm ausgeführten Routinen in den Indizierer oder in die Indizierer-Schnittstelle verlagert werden können. Die Beschleunigung resultiert dann zum einen aus der schnelleren Abarbeitung durch ein vorkompiliertes Programm, und zum Anderen und Wesentlichen aus der Verlagerung der Bearbeitung aus der Anfragephase in die Indizierungsphase. In der folgenden Abbildung ist eine derartige geänderte Aufgabenverteilung dargestellt.

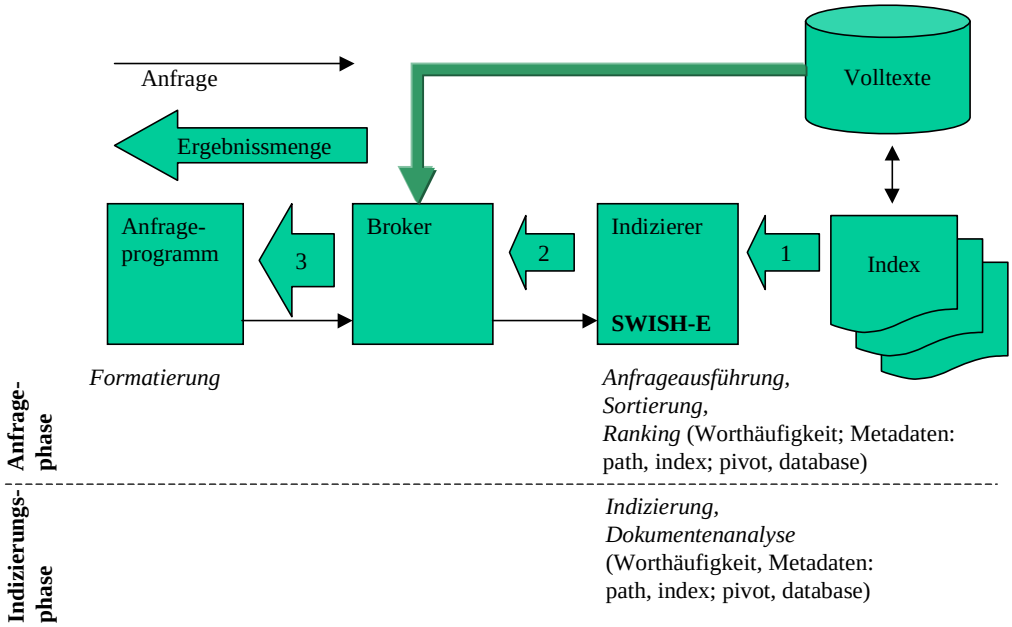

Abb.21:Aufgabenverteilung mit Schwerpunkt auf dem Indizierer

Um diese Aufgabenverteilung zu realisieren, muss der eingesetzte Indizierer vor allem um die in SWING verwendeten Ranking-Algorithmen erweitert werden. Da mit SWISH-E und MG bereits zwei zusätzliche Indizierer eingebunden wurden, und die Verlagerung des Ranking auch für die Benutzung der anderen vorbereiteten Indizierer von Interesse ist, weicht die letztlich gewählte Architektur von der in Abb.21 gezeigten Architektur ab. Die vorgestellte Architektur erfordert Änderungen pro Indizierer. Dies kann innerhalb dieses Projektes nicht realisiert werden.

Eine für SWISH-E und MG einheitliche Integration wurde realisiert, indem die SWING typischen Gewichtungswerte in der Indizierungsphase berechnet, aber nicht im Index des Indizierers abgelegt werden. Die zusätzlichen Gewichte werden nach der Indizierung geladen und permanent in einer Hash-Struktur im Hauptspeicher gehalten. In der Anfragephase wird der aus SWING resultierende Gewichtungswert mit dem Gewichtungswert des Indizierers zusammengefasst, um einen Gesamtgewichtungswert zu bilden. Die folgende Abbildung stellt die umgesetzte Aufgabenverteilung dar.

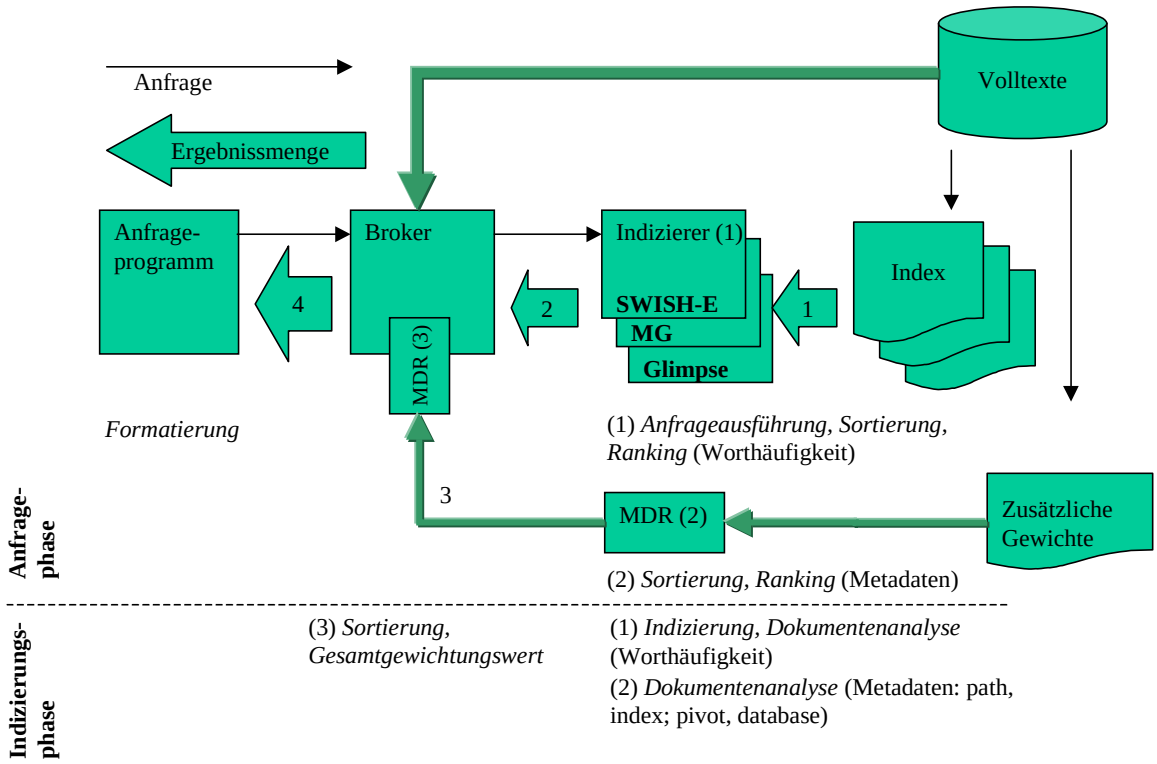

Abb.22: realisierte Aufgabenverteilung in SWING

Bei der Berechnung des endgültigen Gewichtungswertes muss beachtet werden, dass der zusätzliche Gewichtungswert nicht entsprechend der einzelnen Suchwörter das Endergebnis beeinflusst. Statt dessen wirkt die zusätzliche Gewichtung pro gefundenem Dokument.

[1] Nutzung von Glimpse als Indizierer

Die vorgestellte Variante der Integration der in SWING verwendeten erweiterten Rankingfunktionen in eine durch einen Indizierer angebotene Rankingvariante stellt zugleich einen Weg dar, wie das Ranking in bestehenden Systemen verfeinert werden kann.
Die Berechnung der Gewichtungswerte wurde analog zu der bisherigen Berechnung in SWING umgesetzt. Daher folgt im nächsten Abschnitt eine kurze Beschreibung des bisherigen Rankings in SWING.

2.2.3 Ranking in SWING

Einen nicht unwesentlichen Teil der Laufzeit der Anfrage nimmt das Ranking im Anfrageprogramm ein. Der Gewichtungswert wird in SWING bei Benutzung von Glimpse vollständig in der Anfragephase bestimmt. In die Gewichtung eines Dokuments fließen in SWING folgende Dokumenteigenschaften ein.

- Häufigkeit des Suchwortes im Dokument
- Länge des Dokumentenpfades
- Name des Dokuments
- Pivot-Wert
- Typ der Datenquelle (Datenbank oder nicht)
- Ort des Vorkommens (in einem definierten Metatag, im *official-homepage* Metatag [2], im Titel)

Die Gewichtung eines Dokuments erfolgt in SWING[1] nach folgender Formel.

$$r = \begin{cases} \text{DATABASE} & \text{URL ist als Datenbank registriert} \\ \text{PIVOT} + \text{OHP} + \text{METATAG} + \text{INDEX\_HTML} + \text{MATCHED\_LN} + \text{PATH} & \text{sonst} \end{cases}$$

Abb.23: Berechnung des Gewichtungswertes in SWING

Die Belegung jeder der aufgeführten Variablen ist von der Erfüllung bestimmter Bedingungen abhängig. Vor der Ausgabe werden die Gewichtungswerte auf 100 normiert. Die folgende Tabelle enthält die möglichen Werte der einzelnen Variablen und die Bedingung für ihre Belegung.

Variable	**mögliche Werte**	**Bedingung für Belegung**	**relativer Gewichtungsanteil (bis ca.)**
PIVOT	0; 30	URL des gefundenen Dokuments ist in der Pivot-Datei eingetragen.	30 %
OHP	0; 15	Suchwort in einer Zeile mit einem *official-homepage* Metatag (ohp-Tag)	15 %

[1] Zur Begründung der Werte und einer allgemeinen Erläuterung des Ranking in SWING siehe [1].

METATAG	0; 15	Suchwort in einer Zeile mit einem der in Metatag-Liste enthalten Begriffe	15 %
INDEX_HTML	0; 15	Der Name der Datei ist "INDEX.HTM"	15 %
MATCHED_LN	0; 4; 6; 8	je nach Anzahl der gefundenen Treffer	8 %
PATH	0; 2; 4; 6; 8	je nach Anzahl der Trennzeichen "/" in der URL des gefundenen Dokuments	8 %
DATABASE	100	Die URL des gefundenen Dokuments wurde als Datenbank registriert.	100 %

Abb.24: Variablen der Berechnung des Gewichtungswertes

Die Gewichtung eines Nicht-Datenbank Dokuments kann somit zwischen 0 und 101 liegen. Bezogen auf die volle Gewichtung[1] ergeben sich die in der Spalte relativer Gewichtungsanteil dargestellten Werte. Der Anteil eines einzelnen Gewichtungs-Summanden, z.B. von PATH, der durch die Anzahl der gefundenen "/"-Trennzeichen in der Dokumenten-URL bestimmt wird, liegt in diesem Fall bei ca. 8%. Wird kein weiteres Kriterium erfüllt, so wird der Gewichtungswert des Dokuments zu 100% durch diesen Summanden bestimmt.
Die in der Tabelle Abb.24 aufgeführten Variablen können entsprechend der für ihre Berechnung verwendeten Ausgangswerte in zwei Gruppen unterteilt werden.

Gruppe 1	Der Wert wird durch das Suchwort und den Dokumenteninhalt bestimmt.
Gruppe 2	Der Wert wird durch den Dokumentennamen und unabhängig vom Suchwort bestimmt.

Die Algorithmen zur Berechnung von MATCHED_LN, METATAG und OHP gehören zur ersten Gruppe. Zur zweiten Gruppe gehören die Algorithmen zur Berechnung von PIVOT, INDEX_HTML, PATH und DATABASE.
Die Berechnung des Gesamtgewichtungswertes, in der im Folgenden vorgestellten Variante, lehnt sich an die dargestellte Berechnungsvorschrift an.

2.2.4 MetaData Ranking - MDR

2.2.4.1 Berechnung des Ranking-Wertes

Im Gegensatz zu Glimpse wird durch SWISH-E ein Gewichtungswert r_i für jedes Dokument geliefert. Dieser Wert beruht im Wesentlichen auf der Häufigkeit und der Verteilung der Suchbegriffe in der Dokumentensammlung[2], und entspricht damit in seiner Bedeutung der Variablen MATCHED_LN aus dem vorigen Abschnitt . Dieser Gewichtungswert r_i stellt den wesentlichen Einflussfaktor auf den endgültigen

[1] Alle Gewichtungswerte gehen mit ihrem maximal Wert in das Endergebnis ein.

[2] siehe Abschnitt Beschreibung SWISH-E

Gewichtungswert *r* dar. Im Gegensatz zu MATCHED_LN dessen Wert maximal 8% der vollen Gewichtung ausmachte, stellt r_i einen Basiswert dar, der um maximal 53% durch die zusätzlichen Gewichte verändert werden kann. Die Variablen pivot, index_html, path und database, zur Bedeutung siehe voriger Abschnitt, bewirken eine relative Erhöhung des durch den Indizierer berechneten Gewichtungswertes r_i. In der folgenden Abbildung ist die Berechnungsformel dargestellt.

$$r = [\ 1 + \max(\ (\text{pivot} + \text{index\_html} + \text{path}), \text{database}\ )\ ] \times r_i$$

Abb.25: geänderte Ranking Variante in SWING

In der folgenden Tabelle sind die Variablen und ihre Belegung dargestellt.

Variable	**Wert**[1]	**Bemerkung/ Bedingung für Belegung**
r_i		durch den Indizierer bestimmter Gewichtungswert
pivot	0,3	analog zur Abb.24
index_html	0,15	
path	0,02; 0,04; 0,06; 0,08	
database	0,5	

Abb.26: neue Gewichtungsvariablen

Ähnlich dem bisher verwendeten Ranking-Algorithmus, wird der Gewichtungswert des Dokuments um 15% erhöht, wenn die URL des Dokuments den Dokumentennamen "INDEX.HTM" enthält. Gehört das gefundene Dokument zu einer registrierten Datenbank, so wird der Wert nicht auf 100 % gesetzt sondern der Gewichtungswert um 50 % erhöht.

Da durch SWISH-E keine Trefferzeilen übergeben werden, kann das OHP-Ranking und das Metatag-Ranking nicht ausgeführt werden. Eine Höhergewichtung der Treffer innerhalb von Metatags ist in SWISH-E möglich. Dies erfordert aber den Einsatz von Filterprogrammen[2]. Diese formatieren die im SOI-Format vorliegenden Datenquellen während des Indizierungsprozesses um, sodass die entsprechenden Stellen als Metatag durch den Indizierer erkannt werden. Da während der Arbeit mit dem Programm SWISH-E die Indizierungszeiten extrem lang waren, z.T. 1,5 Tage pro Indizierung, wurde diese Variante nicht realisiert. Nach der Umstellung auf die Betaversion 2.1-Dev-20, ist diese Variante möglich, konnte aber nicht mehr im Bearbeitungszeitraum realisiert werden.

MATCHED_LN wird in dieser Form nicht mehr benötigt, da beide Indizierer Gewichtungswerte für die gefundenen Dokumente liefern.

Der folgende Abschnitt behandelt Auswirkungen des Wegfalls der beiden Ranking-Teilalgorithmen.

[1] Abweichende Werte können über eine Konfigurationsdatei gesetzt werden.

[2] Bezeichnung aus SWISH-E

2.2.4.2 Auswirkung fehlenden Teilalgorithmen

Die Markierung einzelner HTML-Seiten durch den Official-Homepage Metatag führt zu einer Höhergewichtung dieser Seiten durch SWING. Leider wird von dieser Möglichkeit bisher kaum Gebrauch gemacht. So befanden sich am 04.07.2001 in der Datenbasis des SwingBrokers nur vier Dokumente die diesen Metatag enthielten. Bei drei dieser Dokumente handelte es sich um Beschreibungen zu SWING. Das vierte Dokument, das den Official-Homepage Metatag enthielt, war eine Beschreibung zu BlueView. In diesem Dokument wurde die semantische Anreicherung durch den Metatag zur Beeinflussung des Gewichtungswertes eingesetzt. Solange keine größere Akzeptanz für dieses Verfahren erreicht wird, hat der Wegfall des auswertenden Rankingalgorithmus keine weiteren Auswirkungen.
Im Gegensatz zum OHP-Tag beeinflusst der Wegfall des Rankings für den Titel und für ausgewählte Metatags den berechneten Gewichtungswert vieler Dokumente. Um diesen Nachteil zu beheben, muss eine Anpassung erfolgen, die es SWISH-E erlaubt, die SOIF-Felder der Datenbasis auszuwerten. Durch diese Anpassung kann durch den Indizierer eine Höhergewichtung bei Vorkommen des Suchwortes in einzelnen Metatags und im Titel realisiert werden.
Im nächsten Abschnitt wird das Programm zur Berechnung der zusätzlichen Gewichtungswerte vorgestellt und es werden die notwendigen Änderungen an der Indizierer-Schnittstelle beschrieben.

2.2.5 Beschreibung von MDR

MetaData Ranking (MDR) ist zugleich ein Programm, und ein in die Indizierer-Schnittstelle eingebundenes Modul. Beide wurden geschaffen um die in SWING bisher umgesetzten erweiterten Ranking-Algorithmen unter einer veränderten Aufgabenverteilung, und bei Nutzung eines alternativen Indizierers anwenden zu können. Das Programm besteht aus zwei Teilen.
Der erste Teil ist ein Programm, das in der Indizierungsphase arbeitet, und die Dokumente der Datenbasis des Brokers analysiert. Durch dieses Programm werden ausschließlich die Algorithmen der zweiten Gruppe der Rankingalgorithmen von SWING (siehe Kapitel 2.2.3) realisiert. Die Algorithmen der ersten Gruppe werden durch den Indizierer abgedeckt oder müssen in diesen integriert werden. Daher benötigt MDR zur Berechnung der Gewichtungswerte ausschließlich die Namen der Datenquellen. Diese befinden sich am Anfang der jeweils zugehörigen SOIF-Datei. Das Programm MDR liest die ersten Bytes aller Dateien unterhalb der objects-Verzeichnisstruktur[1] des Brokers ein, und bestimmt den jeweiligen Dateinamen der Datenquelle. Anschließend wird geprüft,

- ob der Dateiname in der Liste der eingetragenen Datenbanken enthalten ist,
- ob der Dateiname in der Pivot-List enthalten ist,
- ob die Zeichenfolge INDEX.HTM im Dateinamen enthalten ist und
- wieviele Trennzeichen "/" im Dateinamen enthalten sind.

[1] Der Pfad kann über eine Konfigurationsdatei festgelegt werden.

Dementsprechend werden die Werte für database, pivot, index_html und path gesetzt. Diese Analyse der einzelnen Dateien und die ermittelten Gewichtungswerte können in der Log-Datei (siehe Anhang Kapitel 4.6) nachvollzogen werden.
Die auf diese Weise für jede Datenquelle bestimmten Gewichtungswerte werden unter einem eingestellten Dateinamen abgelegt. Die Funktionsweise des Programms ist in der folgenden Abbildung dargestellt.

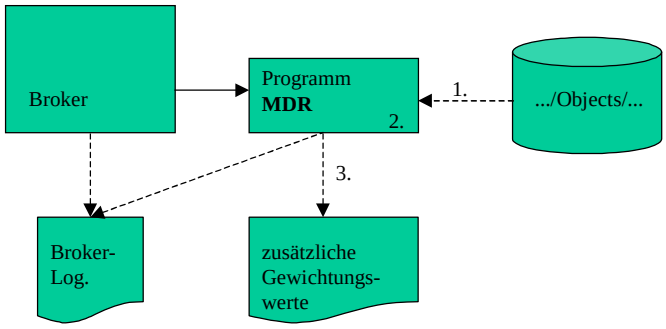

Abb.27: Arbeitsweise des Programms MDR in der Indizierungsphase

In der obigen Abbildung ist mit

1. das dateiweise Einlesen des object-Verzeichnisses, mit
2. die Analyse des Dateinamens und die Bestimmung der zusätzlichen Gewichtungswerte und mit
3. die Speicherung der Werte in einer entsprechenden Datei

dargestellt.
Im Anschluss an die Bestimmung der Gewichtungswerte wird eine Statistik über die pro Gewichtungskriterium gefundenen Dokumente ausgegeben (siehe Anhang 4.7).
Die Gesamtzeit der Vorbereitungsphase des Brokers erhöht sich entsprechend um die für die Bestimmung der zusätzlichen Gewichte durch MDR notwendige Zeit.

Der zweite Teil des Programms arbeitet hauptsächlich in der Anfragephase. Es handelt sich dabei um ein in die Indizierer-Schnittstelle integriertes Modul. Diese Modul MDR weist folgende Funktionalität auf.

I Laden der zusätzlichen Gewichtungswerte nach dem Start des Brokers bzw. nach erfolgter Indizierung.

II Ranking und Sortierung
   1. Laden der vom Indizierer übergebenen Ergebnisdatei mit Treffern und Gewichtungswerten.
   2. Pro geladenem Treffer Bestimmung der zusätzlichen Gewichtung aus dem Hash und Berechnung des Gesamtgewichtungswertes.
   3. Sortierung der Ergebnisse nach dem Gesamtgewicht,
   4. Normierung auf 100% und Übergabe der Treffer und der Gewichtungswerte an den Broker.

Die Arbeitsweise des Moduls im Zusammenhang mit dem Broker und dem Indizierer ist in der folgenden Abbildung dargestellt.

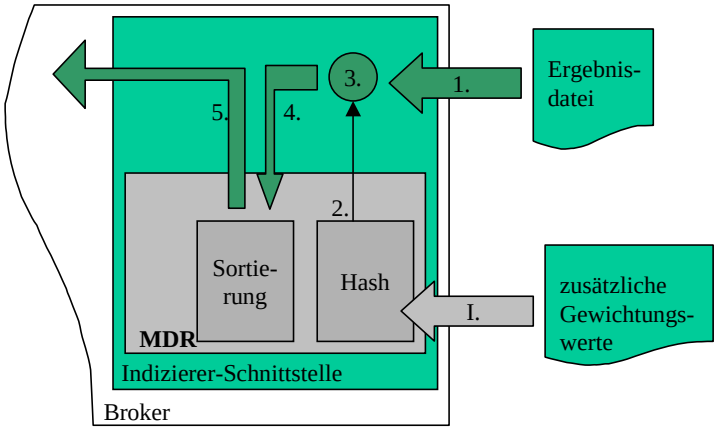

Abb.28: Funktionsweise des Moduls MDR

Die oben dargestellte Funktionsweise soll an einem Beispiel verdeutlicht werden.

- An den Broker wird eine Anfrage gestellt. Dieser bereitet die Anfrage auf und leitet sie, zusammen mit dem Namen der Ergebnisdatei, an den Indizierer weiter.
- Der Indizierer bestimmt die Ergebnismenge und schreibt sie in die Datei mit dem übergebenen Namen.
- Das Modul MDR liest diese Ergebnisdatei zeilenweise ein (1.).
  - Pro Zeile (Treffer) wird der Identifikator der Datenquelle und der Gewichtungswert bestimmt.
  - Für den Identifikator wird der zusätzliche Gewichtungswert aus dem Hash abgerufen (2.).
  - Es wird der Gesamtgewichtungswert berechnet (3.) und
  - in eine sortierte Liste eingefügt (4.).
- Nach dem die Datei abgearbeitet wurde, wird die Liste entsprechend der Sortierung nach der Gesamtgewichtung an den Broker übergeben (5.). Dabei wird der Gewichtungswert auf 100% normiert.

Das Programm und Modul MDR wurde in C geschrieben und unter SunOS 5.5.1 kompiliert (siehe Anhang 4.8).
Der im folgenden Abschnitt angeführte Laufzeitvergleich prüft ob eine Verbesserung der Antwortzeiten erreicht werden konnte.

2.2.6 Laufzeitvergleich

Durch einen weiteren Vergleich am Ende der zweiten Arbeitsetappe wurde geprüft, ob die Änderung der Arbeitsverteilung zu einer Verbesserung der Laufzeit geführt hat. Insbesondere war festzustellen ob sich an der für den Nutzer wichtigen Schnittstelle, dem Browser, eine deutliche Verkürzung der Anfragezeit ergibt.

In der folgenden Tabelle sind die wesentlichen Anfragen aus dem ersten Laufzeitvergleich, mit den in der neuen Architektur erreichten Laufzeiten, und der jeweiligen Trefferanzahl dargestellt.

Fragen	**SwingBroker** Indizierer: Glimpse				**SwingBrokerS** Indizierer: SWISH-E				**Δ**	
	T			N	T			N	abs	%
	1.	R	∅		1.	R	∅			
holz	16	15	15,5	185	15	13	14	200	1,5	9,68
andreas and heuer	17	13	15	88	13	10	11,5	200	3,5	23,33
datenbank	21	17	19	200	11	11	11	200	8	42,11
datenbank*	19	17	18	200	12	11	11,5	200	6,5	36,11
schwerin and tourismus	16	16	16	97	13	11	12	200	4	25,00
stellenangebote	12	12	12	83	9	8	8,5	122	3,5	29,17
gesetze and oberfinanzdirektion	19	9	14	3	4	2	3	1	11	78,57
gesetze or oberfinanzdirektion	14	12	13	28	15	9	12	200	1	7,69
holz and verlag	19	13	16	34	5	3	4	24	12	75,00
holz and not verlag	24	21	22,5	159	4	3	3,5	24	19	84,44

Abb.29: Laufzeitvergleich zwischen SWISH-E und Glimpse in SWING

Im arithmetischen Mittel ergibt sich an der Web-Schnittstelle eine Laufzeitverbesserung von ca. 40 % für die untersuchten Anfragen. Die im Abschnitt Wertung des Laufzeitvergleichs auf Seite 19 angesprochenen Probleme gelten analog für diesen Vergleich.

Der folgende Abschnitt beschreibt die für den Einsatz von MDR an SWING notwendig gewordenen Änderungen.

2.2.7 Änderungen am Broker und am Anfrageprogramm

Durch die Integration des MDR-Moduls in die Indizierer-Schnittstelle ergeben sich Änderungen am Broker und am Anfrageprogramm. Diese Änderungen müssen bei einer Aktualisierung der SWING zugrundeliegenden Harvest-Version beachtet und nachgepflegt werden.

Der Aufruf von MDR zur Bestimmung der zusätzlichen Gewichtungswerte wird in die *index.c* Module der jeweiligen Indizierer-Schnittstelle integriert.

Für den Zugriff auf die zusätzlichen Gewichtungswerte wird eine Hash-Struktur permanent im Speicher gehalten. Damit dieser Speicher bei Beendigung des Programms freigegeben wird, muss die Routine *Broker_Shutdown* im Modul *main.c* des Brokers entsprechend ergänzt werden. Über eine globale Variable, die bei der Initialisierung der Datenstruktur gesetzt wird, erfolgt die Freigabe. Alle weiteren Änderungen beschränken sich auf das Modul index.c im .../src/broker/swish/ Verzeichnis.

Außer am Broker erfolgen auch Änderungen am Anfrageprogramm (*nph-search*). Durch einen zusätzlichen Parameter *swishflag=on|off* werden in diesem Programm alle Ranking-Routinen ausgeschaltet.

Weiterhin bewirkt dieser Schalter eine Änderung der Überschriften, der Inhalte der Statistik, der Sortierung und der Gruppierung in den durch dieses Programm erstellten HTML-Seiten.
Im nächsten Abschnitt werden die Arbeiten und Ergebnisse der dritten Arbeitsetappe beschrieben.

2.3 Dritte Arbeitsetappe: Konzeption eines Anfrage-Cache für SWING

Über 10% der Nutzung von Suchmaschinen wird durch ungefähr 1000 wiederkehrende Suchanfragen verursacht [5]. Daher werden durch einige Suchmaschinen auf diese Anfragen vorausberechnete Ergebnismengen geliefert. Durch eine Analyse der an SWING gestellten Anfragen soll überprüft werden, ob durch ein ähnliches Vorgehen eine Lastreduzierung in der Anfragephase und eine Erhöhung der Verfügbarkeit erreicht werden kann. Gegebenenfalls soll ein Modul SWING QUERY CACHE (SQC) zur Vorausberechnung von Anfragen konzipiert und prototypisch umgesetzt werden.

2.3.1 Ausgangszustand

Im Monat Juni 2001 wurden 3604 Aufrufe für Suchanfragen an die Suchmaschine SWING[1] gestellt. Viele dieser Anfragen werden mehrfach[2] ausgeführt. In der folgenden Tabelle ist die Anzahl der häufigsten Anfragen und die zugehörige Anzahl von Aufrufen für 2 Varianten dargestellt.

	Variante 1		Variante 2	
	Anfragen	Aufrufe	Anfragen	Aufrufe
TOP 20	20	417	-	-
TOP 100	-	-	100	799
TOP 27 (ohne Ergebnis)	27	97	27	97
Σ zu cachende ...	47	514	127	896

Abb.30: Anfragen und Aufrufe (absolut)

Wie die Tabelle zeigt, wurden ca. 14% der Aufrufe durch nur 47 verschiedene Anfragen[3] ausgelöst. Betrachtet man die 100 häufigsten Anfragen, so wurden ein Viertel (896) aller Aufrufe, von nur 127 verschiedene Anfragen[4] verursacht.
In der folgenden Tabelle ist der Anteil der häufigsten erfolgreichen und nicht erfolgreichen Anfragen an der Menge der im Monat Juni ausgeführten Aufrufe aufgeführt.

[1] Einsatz der Suchmaschine im Landesinformationssystem von Mecklenburg Vorpommern, MV-Info

[2] siehe SwingBroker Statistik im Anhang

[3] Top 20 der erfolgreichen Anfragen und Top 27 der Anfragen ohne Treffer

[4] Top 100 der erfolgreichen Anfragen und Top 27 der Anfragen ohne Treffer

		Variante 1	**Variante 2**
gecachte Anfragen		47	127
gecachte Aufrufe		514	896
Erfolgreiche Anfragen:	2637	15,81 %	30,30 %
Anfragen ohne Ergebnis:	949	10,22 %	10,22 %
Anfragen insgesamt:	3604	14,26 %	24,86 %

Abb.31: Anfragen und Aufrufe (relativ)

Wie aus obiger Tabelle ersichtlich ist, kann durch eine Vorausberechnung von Anfragen eine Entlastung der Suchmaschine und damit eine höhere Verfügbarkeit erreicht werden. Die Verfügbarkeit der Suchmaschine erhöht sich, da SWING zur Zeit nicht mehrere Anfragen parallel abarbeiten kann. Bei einer kürzeren Antwortzeit pro Anfrage ist die Suchmaschine jedoch früher und somit insgesamt länger für neue Anfragen verfügbar.

Die Entlastung der Suchmaschine bezieht sich auf die Anfragephase, da durch die Vorausberechnung der Ergebnismengen der Aufwand für die Bestimmung der Ergebnismenge von der Anfragephase in die Vorbereitungsphase verlegt wurde. Durch diese Vorausberechnung von Anfragen im Anschluss an die Indizierung, kommt es zu einer Veränderung des Arbeitszyklus der Suchmaschine. Dieser Zyklus besteht bisher aus der Vorbereitungsphase (Indizierungsphase) und der Anfragephase.

In der folgenden Abbildung ist der erweiterte Arbeitszyklus der Suchmaschine dargestellt.

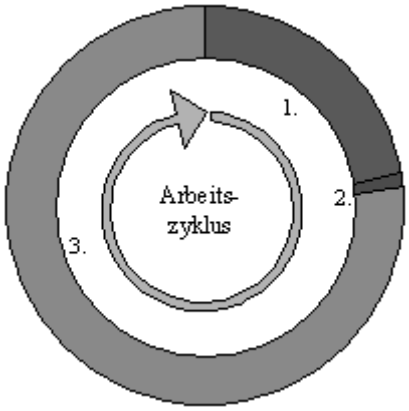

Abb.32: erweiterter Arbeitszyklus der Suchmaschine

Der veränderte Arbeitszyklus der Suchmaschine besteht aus der Vorbereitungsphase (1. und 2.) und der Anfragephase (3.). Die Vorbereitungsphase besteht aus der Indizierung (1.) und der Vorausberechnung der häufigsten Anfragen (2.).

Die absolute Belastung der Suchmaschine erhöht sich durch die Vorausberechnung, da nach jedem Indizierungsprozess die Anfragen neu berechnet werden müssen, unabhängig davon, ob die Anfragen innerhalb des Arbeitszyklus der Suchmaschine durch den Nutzer ausgeführt werden.

Im nächsten Abschnitt wird eine geeignete Schnittstelle für den SQC bestimmt.

2.3.2 Integrationsstellen für den SQC

Um eine möglichst einfache Integration des SQC in die bestehende Architektur von SWING zu erreichen, müssen die, in der Anfragekomponente von SWING vorhandenen Schnittstellen und Routinen bei der Integration berücksichtigt werden. Unter dem Aspekt der Aufwandsreduzierung bietet sich insbesondere die Anfrage-Schnittstelle an, da hier sowohl der Aufwand des Indizierers, als auch der des Brokers bei der Bearbeitung von Anfragen gepuffert würde.
Im folgenden Abschnitt wird geprüft, ob die zu cachende Datenmenge die Integration an dieser Schnittstelle zulässt.

2.3.2.1 Cache-Größe

Wesentlich für die Realisierbarkeit des Anfrage-Cache ist die Größe der vorausberechneten Ergebnisse. Daher erfolgt in diesem Abschnitt eine Größenabschätzung für die Anfrageergebnisse. Zur Abschätzung der Cachegröße wurden die im Laufzeitvergleich verwendeten Anfragen ausgeführt, und die Größe der Ergebnisdateien bestimmt. Dabei wurde die Größe der erstellten Ergebnisdateien gemessen, und in Relation zu der Anzahl der enthaltenen Treffer gesetzt. Die Ergebnisse sind in der im Anhang Seite 45 enthaltenen Tabelle Abb.47 aufgeführt.
Berücksichtigt man die Standardeinstellung von maximal 200 Ergebnissen pro Suchanfrage, so ergibt sich bei 127 vorausberechneten Anfragen ein maximaler Speicherbedarf von ca. 40 MByte, und ein durchschnittlicher Speicherbedarf von 16 MByte. Diese Werte sind für die beiden betrachteten Varianten in der folgenden Tabelle zusammengestellt.

		Treffergröße max.	**Treffergröße ∅**
Größe pro Anfrage[1]		0,3 MB	0,125 MB
Größe des Cache	Variante 1	15 MB	6 MB
	Variante 2	40 MB	16 MB

Abb.33: Größe der Ergebnisdatei und des Cache

Diese grob ermittelte Größe ist deutlich höher als der tatsächlich zu cachende Datenstrom an der Anfrage-Schnittstelle. Erstens ist die tatsächliche Trefferzahl geringer, und zweitens nimmt die durchschnittliche Größe pro Treffer mit wachsender Trefferanzahl ab. Außerdem wurde diese Messung unter Verwendung von Glimpse als Indizierer ausgeführt. Da SWISH-E und MG keine Trefferstellen in der Ergebnismenge übergeben, ist die zu cachende Gesamtmenge bei der Verwendung dieser Indizierer ebenfalls kleiner.
Der nächste Abschnitt beschäftigt sich mit der Frage, in welchen Teil der genannten Schnittstelle der SQC integriert werden sollte.

[1] bei maximal 200 Treffern pro Anfrage

2.3.2.2 Varianten für die Integration des SQC

Nachdem die Entscheidung gefällt wurde den SQC in die Schnittstelle zwischen Anfrageprogramm und Broker zu integrieren, stellt sich die Frage, in welchen Teil der in der folgenden Abbildung dargestellten Schnittstelle dies geschehen soll.

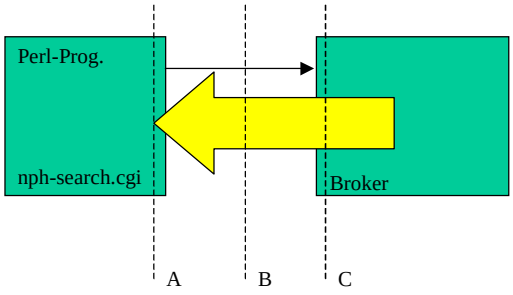

Abb.34: Mögliche Integrationsvarianten für den SQC

Der Anfragecache für SWING kann in das Anfrageprogramm (A) oder in den Broker (C) integriert werden. Als weitere Lösung ist eine Ausführung als eigenständiger Dienst zwischen Anfrageprogramm und Broker (B) möglich.
Im weiteren sollen die Vor- und Nachteile dieser Varianten aufgeführt werden.

	Realisierungsvariante	**Vor- und Nachteile**
A	Der SQC wird in das Anfrageprogramm integriert. Alle notwendigen Änderungen erfolgen im Anfrageprogramm. Insbesondere wird die Routine zur Übergabe der Anfrage an den Broker erweitert.	• schnellste Variante für eine einfache Implementierung, • höchste Laufzeit bei Anfrageausführung, • verschlechtert die Wartbarkeit des Anfrageprogramms • sicherer Betrieb, Programmabstürze des SQC haben keine Auswirkungen auf den Betrieb des Brokers • Probleme bei der vollständigen Bereitstellung der Funktionalität • benötigt weitere Programme
B	Der SQC wird als eigenständiger Dienst zwischen Anfrageprogramm und Broker betrieben.	• aufwendigste Implementierung, • gute Laufzeit • kompliziertester Betrieb • beste Wartbarkeit • geringere Stabilität des Systems bei Fehlern, es erfolgt nach Abstürzen kein automatischer Neustart, der Broker bleibt trotzdem weiter arbeitsbereit

C	Der SQC wird als Modul im Broker realisiert.	• aufwendige Implementierung, • beste Laufzeit, • komplizierte Wartung, da Update des Harvest-Systems stets ein Nachpflegen der Änderungen erfordert • evtl. Stabilitätsprobleme beim Betrieb, da Fehler im Module zu Abstürzen des Brokers führen können, d.h. nach Abstürzen des SQC muss der Broker neu gestartet werden

Abb.35: Realisierungsvarianten und ihre Vor und Nachteile

Für die Implementierung des im Rahmen dieser Arbeitsetappe geplanten Prototypen wird die Variante C gewählt. Unter dem Gesichtspunkt einer modularen Architektur, einer leicht wartbaren Implementierung und besonders im Hinblick auf den Einsatz in einer verteilten Broker-Architektur ist Variante C der Vorzug zu geben. Im Rahmen der prototypischen Implementierung in dieser Arbeitsetappe, ist diese Lösung zu aufwendig.
Der folgende Abschnitt stellt die gewählte Architektur und Funktionsweise dar.

2.3.2.3 Funktionsweise des SQC und Integration in den Broker

Der Anfrage-Cache SQC ist hauptsächlich ein Programm-Modul, das in den Broker der Anfragekomponente von SWING integriert ist. Für die Einbindung wurden die Eigenschaften der Socket-Kommunikation zwischen dem Broker und dem als Client arbeitenden Anfrageprogramm genutzt.
Jede Anfrage trifft als Zeichenkette beim Broker ein. Die eintreffende Zeichenkette wird in einer brokerinternen Datenstruktur gespeichert und bis zum Query-Manager des Brokers geleitet. Hier wird die Socket-Rückverbindung zum Client getestet und anschließend die Anfragebearbeitung[1] des Indizierers gestartet. Die vom Indizierer gelieferten Ergebnisse werden über die Routine *QM_user_object*[2] des Query-Managers an den Broker weitergeleitet, ergänzt und über die Socketverbindung an den Client übergeben.

[1] Aufruf der entsprechenden Routine des eingestellten Indizierers über *do_IND_do_query* aus *brkutil.c.*

[2] Die Routine *QM_user_object* schreibt pro übergebenen Objekt folgende Daten auf die Socketverbindung zum Client URL, Description, SOIF- Pointer und Nutzerdaten. Zusätzlich können über die Routine *QM_return_attributes* zusätzliche Daten vom Broker aus der Datenbasis geladen und an den Client geleitet werden.

Die folgende Abbildung zeigt die bisherige Aufrufstruktur im Broker.

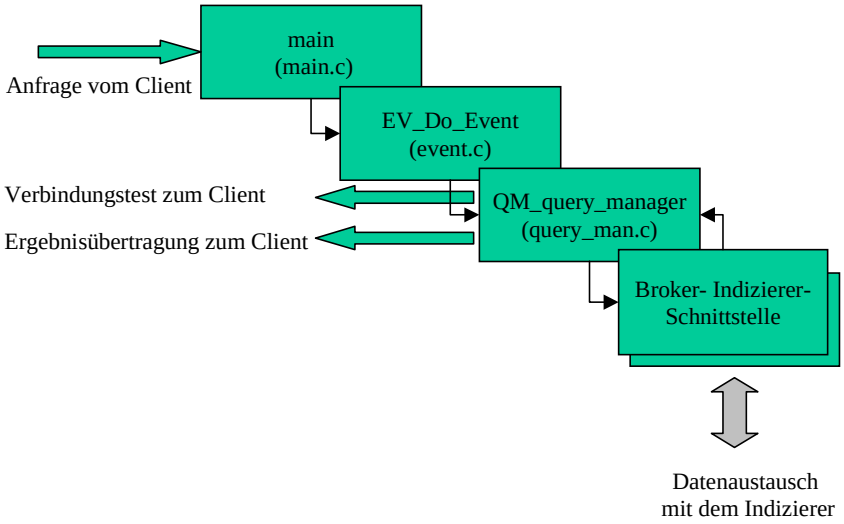

Abb.36: Aufrufstruktur im Broker (mit Socketverbindungen)

Da der SQC unabhängig vom Indizierer arbeiten soll, ist somit der Query-Manager der passende Integrationspunkt. Anstelle des Aufrufs der Anfragebearbeitung des Indizierers wird im Query-Manager nun der Aufruf an den SQC ausgeführt. Wird beim Aufruf des SQC eine zur Anfrage passende Ergebnismenge gefunden, so wird sie direkt über die Socketverbindung an den Client geleitet. Damit entfällt der Aufruf des Indizierers und die Weiterbearbeitung der Daten durch den Broker. Kann durch den SQC kein Ergebnis geliefert werden, oder tritt ein Fehler in der Bearbeitung ein, so wird mir der normalen Anfragebearbeitung durch den Broker fortgesetzt. Die folgende Abbildung stellt die Integration des SQC in den Broker, und die beschriebe Funktionalität dar.

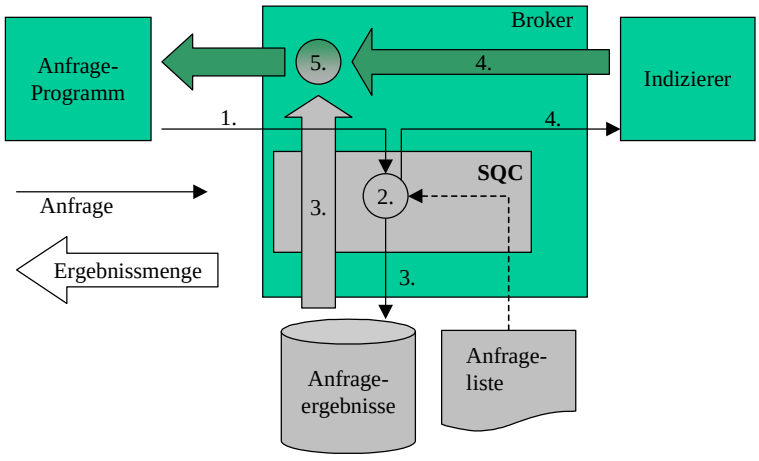

Abb.37: Funktionsweise des SQC in der Anfragephase

Das Füllen des Cache mit Anfrageergebnissen erfolgt unmittelbar nach der Indizierung der Datenbasis. Dazu wird durch den Broker ein Programm aufgerufen, das über die Schnittstelle zwischen Broker und Anfrageprogramm die zu cachenden Anfragen an den Broker stellt. Anschließend nimmt es die vom

Broker gelieferte Ergebnismenge entgegen und speichert sie im Cache. Diese Funktionsweise ist in der folgenden Abbildung dargestellt.

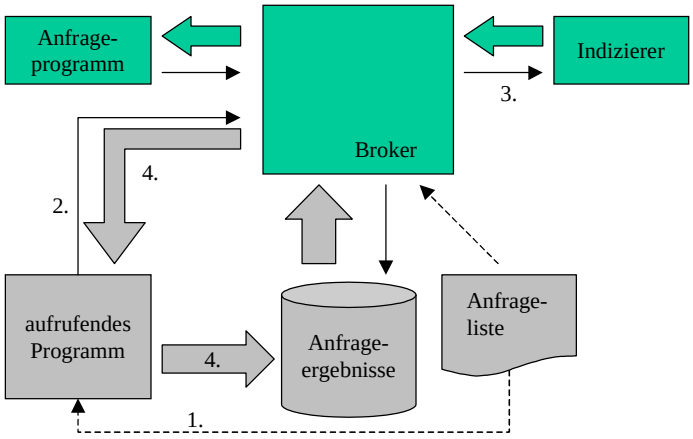

Abb.38: Funktionsweise des SQC bei Einbindung in den Broker

Die zu cachenden Ergebnisse werden als Dateien in einem Unterverzeichnis des jeweiligen Brokers abgelegt. Über eine Relation zwischen dem Anfragestring und dem Dateinamen wird die jeweilige Ergebnismenge bestimmt.
Der folgende Abschnitt enthält Hinweise zu den Besonderheiten des Prototypen.

2.3.2.4 Einschränkungen des Prototypen

Die prototypische Implementierung des SQC als statische Cache weist einige Einschränkungen auf, die im Folgenden aufgezählt werden.

- Es erfolgt keine Unterscheidung der Anfragen nach den zusätzlichen Parametern, wie der Anzahl möglicher Fehler, der Groß- und Kleinschreibung oder der Suche nach ganzen Wörtern.
- Die Zugriffe zum Laden der Ergebnismenge in den Cache werden nicht aus der Zugriffsstatistik der Suchmaschine ausgeblendet.
- Der Prototyp verfügt über keine ausreichende Fehlerbehandlung.
- Der Anfragestring wird direkt als Dateiname benutzt. Daher können im Prototypen nur Anfragen gepuffert werden, die als Dateinamen zulässig sind.
- Die Prüfung, ob eine Anfrage im Cache vorhanden ist, wird direkt als Dateioperation ausgeführt.
- Die Eingabe der zu cachenden Anfragen erfolgt manuell und in Auswertung der Statistik des Brokers.
- Die Datei mit der Anfrageliste muss in einem UNIX-typischen Format vorliegen (Zeilenendezeichen "\n").

Der folgende Abschnitt nennt die am Broker gemachten Änderungen.

2.3.3 Änderungen am Broker

Durch die Integration des SQC-Moduls in den Broker, ergeben sich Änderungen an der Suchmaschine SWING. Diese Änderungen müssen bei einer Aktualisierung der SWING zugrundeliegenden Harvest-Version beachtet und nachgepflegt werden.

Die Routine *Initialize_Broker* des Moduls *main.c* des Brokers wurde um den Initialisierungsaufruf des SQC ergänzt.
Für den Zugriff auf die im Cache vorhandenen Anfragen wird eine Hash-Struktur permanent im Speicher gehalten. Damit dieser Speicher bei Beendigung des Programms freigegeben wird, muss die Routine *Broker_Shutdown* im Modul *main.c* des Brokers ergänzt werden. In dieser Routine erfolgt der Aufruf der Routine *sqc_done* aus der Date *sqc.c* des SQC-Modul. Außerdem erfolgt in der *set_str* Routine des selben Moduls das Setzen der notwendigen Parameter für die Nutzung des SQC.
Im Modul *query_man.c* wird in die Routine *QM_query_manager* der Aufruf *sqc_write_QResult2Socket* des SQC integriert.

3 Nächste Schritte

Einige der durchgeführten Arbeiten konnten im Berichtszeitraum nicht abgeschlossen werden. Andere Arbeiten ergaben Erweiterungsmöglichkeiten. In diesem Abschnitt werden daher mögliche Erweiterungsmöglichkeiten und notwendige Arbeiten der drei Arbeitsetappen zusammengefasst.
Der Indizierer MG benutzt einen auf eine Maximalgröße eingestellten Puffer zur Bearbeitung der Dateien. Da keine maximale Größe für die WWW-Datenquellen festgelegt werden kann, muss der Gatherer die Dateien entsprechend filtern. Alternativ kann der Broker bei der Aktualisierung der Datenbasis eine maximale Dateigröße berücksichtigen.
Beim zusätzliche Ranking durch MDR kann die Aktualität[1] der Datenquelle berücksichtigt werden. Dazu muss beim Parsen der Datei das entsprechende Feld in der SOIF-Datei ausgewertet werden.
Bei der Verwendung des Brokers mit SWISH-E und MG werden keine Einträge in das Statistik-Log geschrieben. Daher entziehen sich die über diesen Broker gestellten Anfragen der späteren Auswertung.
Da die Indizierung über SWISH-E und MG deutlich mehr Zeit benötigt, als die Indizierung mit Glimpse, muss die Indizierung unabhängig von Broker erfolgen. Die durch einen zusätzlichen Prozess bestimmten Indizes werden anschließen, z.B. durch kopieren, dem aktuellen Broker zur Verfügung gestellt.
Für alle erstellten Änderungen muss im Regelbetrieb die Stabilität der Lösung geprüft werden. SWISH-E zum Beispiel reagierte während des Testbetriebs häufiger mit einem Abbruch der Indizierung.

Nach Abschluss der Arbeiten, wurden weitere wesentliche Artikel u.a. [11] und [12] zu diesem Thema gefunden, deren Berücksichtigung bei der Weiterbearbeitung als wichtig erscheint.

[1] In einem SOIF- Feld in jeder Quelldatei des Brokers enthalten

4 Anhang

Die folgenden Seiten enthalten Tabellen, Graphiken und Übersichten, die (i.d.R. durch Beispiele und Messwerte) Aussagen dieses Arbeitsberichtes untersetzen.

4.1 Ausschnitte der Ergebnismenge

Die folgende Tabelle enthält Ausschnitte aus der durch die Indizierer Glimpse, SWISH-E und MG übergebenen Ergebnismenge.

Indizierer	**Glimpse**
Aufruf	glimpse -l -H .../SwingBroker/ 'Meinke‘
Ergebnis	/users/db09/swing2/Harvest/brokers/SwingBroker/objects/68/OBJ158312968
Aufruf	glimpse -H .../SwingBroker/ 'Meinke‘
Ergebnis	/users/db09/swing2/Harvest/brokers/SwingBroker/objects/81/OBJ177154781: title{40}: Fahrschule Andreas Meinke in Ueckermünde
Indizierer	**SWISH-E**
Aufruf	swish-e -f .../SwingBrokerS/index -w Meinke
Ergebnis	1000 /users/db09/swing2/Harvest/brokers/SwingBrokerS/objects/72/OBJ461181672 "OBJ461181672" 792
Indizierer	**MG**
Aufruf	mgquery mgBobjects < Datenbank (mode headers)
Ergebnis	11 < /users/db00/zvd014/harvest/harvest-1.6.1/brokers/mgB/objects/97/OBJ411338997 > @FILE { http://wwwdb.informatik.uni-ros bzw. ---------- 11 0.001487 < /users/db00/zvd014/harvest/harvest-1.6.1/brokers/mgB/objects/97/OBJ411338997 > @FILE { http://wwwdb.informatik.uni-rostock.de/~meike/ update-time{9}: 987786334 ..... ---------- 3 0.000965

Abb.39: Kommandozeilenaufrufe und Ausschnitte aus der Ergebnismenge

4.2 Trennzeichen der Datenübergabe der Anfrageschnittstelle

Die Übergabe der Ergebnismenge über die Anfrageschnittstelle erfolgt unter Zuhilfenahme der in der folgenden Tabelle aufgeführten Trennzeichen.

#101	Message to the User
#103	Error Message to the User
#111	Error Message to the User that ends the Brokerresults
#120	URL of the Match
#122	Opaque data
#124	nbytes\nnbytes of Description
#125	URL of the SOIF object
#126	URL of the Broker's home page
#130	End of Object marker

Abb.40: Trennzeichen im Datenstrom vom Broker zum Anfrageprogramm

Bei Nutzung von Glimpse als Indizierer werden über das Trennzeichen *#122* die Trefferstellen übergeben. Bei der Verwendung von SWISH-E und MG wird dieses Trennzeichens zur Übergabe des Rankingwertes benutzt.

4.3 Umgebungsparameter 1. Laufzeittest

Die folgende Tabelle enthält einige der Umgebungsparameter vor dem Laufzeittest in Tabelle Abb.16.

	gBroker	**sbroker**	**seBroker**
Indizierer	Glimpse	SWISH	SWISH-E V 1.3.5
Größe des Brokerverzeichnis (du brokers/...)	322 MB	300 MB	443 MB
Größe des Datenbasis (du -sk objects/)	221 MB	ca. 200 MB	217 MB
Index	ca. 50 MB	ca. 50 MB	97 MB
gestartete Prozesse	160		172
CPU-Status Idle	98,4%		73,7%
freier Speicher/ freier Swap-Speicher	515 MB/ 72 MB		22 MB/ 7 MB

Abb.41: Vergleich der Umgebungsparameter

Aus den Werten lässt sich ablesen, dass der seBroker zum Zeitpunkt des Tests ähnlich belastet war, wie die beiden anderen Broker. Die mit dem seBroker gegebenenfalls erreichte bessere Laufzeit resultiert daher nicht aus einer geringeren Belastung des Rechners.

4.4 Umgebungsparameter 2. Laufzeittest

Wie die folgende Tabelle zeigt, ist die Datenbasis beider Broker annähernd gleich groß, was ausschließt, dass die Antwortzeiten durch unterschiedlich große Datenbasen bestimmt wurden.

	SwingBroker	**SwingBrokerS**
Indizierer	Glimpse	SWISH-E
Größe des Brokerverzeichnis du brokers/...	337 MB	347 MB
Größe der Datenbasis du -sk objects/	238 MB	235 MB
Indexgröße	80 MB	98 MB

Abb.42: Datenbasis die Vergleichs-Broker

4.5 Durchschnittliche Wortanzahl in Suchanfragen

Die folgende Tabelle enthält die durchschnittliche Anzahl der Suchworte in den Anfragen an die Suchmaschine SWING im Zeitraum 02-08 2001.

Monat/Jahr	durchschnittliche Wortanzahl
02/2001	1.20
03/2001	1.20
04/2001	1.19
05/2001	1.19
06/2001	1.21
07/2001	1.23
08/2001	1.17

Abb.43: durchschnittliche Wortanzahl laut SWING-Statistik

4.6 Ausschnitt aus der MDR-Logdatei

Die folgende Tabelle enthält einen Ausschnitt aus der Logdatei für ein durch MDR bewertetes Dokument bei der Einstellung VERBOSE 3.

File	/users/db09/swing2/Harvest/brokers/SwingBrokerMG/objects/99/OBJ 838413299 18047
URL	http://www.informatik.uni-rostock.de/Kennedy/WCH/index.html 59
url rank	4
index rank	15
database rank	0
pivot rank	0

Abb.44: Ausschnitt aus der Log-Datei für eine Datenquelle

4.7 MDR-Statistik

In der folgenden Tabelle ist die Statistik vom 17.07.2001 für die Ausführung von MDR für die Datenbasis vom SwingBroker und vom SwingBrokerMG enthalten.

	SwingBroker	SwingBrokerMG
Files	44325	58613
Directories	100	100
URL < 2[1]	5675	5768
URL < 3	11557	12225
URL < 4	9424	11515
URL < 5	6714	9389
URL >= 5	10955	19716
INDEX.HTM	1506	1812
DataBase	7	7
Pivot	2	2
SUM(FILES)[2]	172.773.548	234.072.297
avg. Filesize (in Byte)	3897	3993

Abb.45: Statistik für MDR aus der Log-Datei

[1] URL < 2 bedeutet, dass die URL (ohne http://) weniger als 2 Trennzeichen enthält.

[2] Die Summe der insgesamt von MDR gelesenen Bytes, wobei nur die Anfangsbytes jeder Datei gelesen werden.

Die resultierende Ergebnisdatei hatte eine Größe von ca. 10 MB. Die Laufzeit im obigen Beispiel betrug für die SwingBroker-Datenbasis ca. 30 min und für die SwingBrokerMG-Datenbasis ca. 45 min. Die Gesamtzeit[1] zur Indizierung durch den Broker erhöht sich entsprechend.

4.8 MDR-Modulgraph

Die folgende Abbildung enthält einen Modulgraphen des Programms MDR. Das Programm wurde in C geschrieben und ist unter SunOS 5.5.1 kompilierbar.

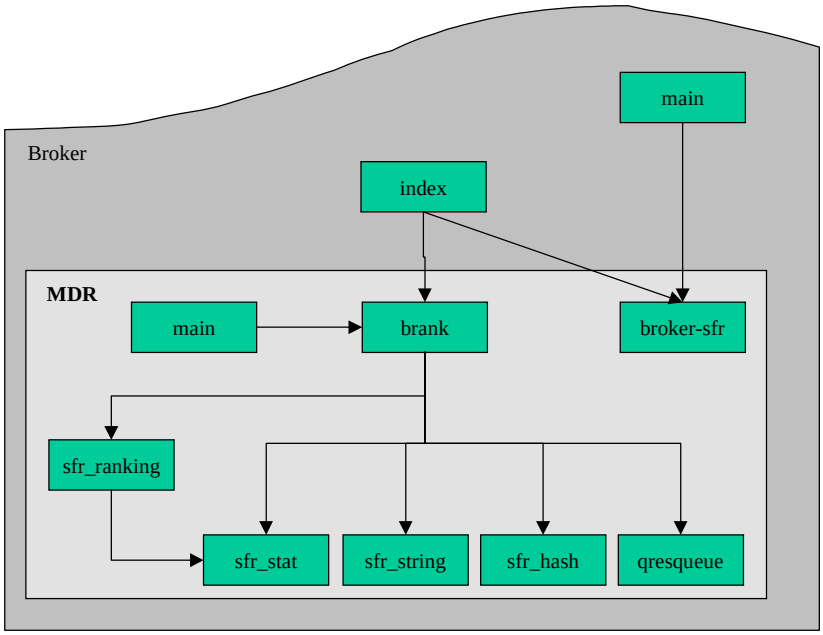

Abb.46: Graf der anwendungsspezifischen Module für
das MDR-Programm und das MDR-Modul

[1] bestehend aus der Aktualisierung der Datenbasis, der Indizierung durch Glimpse, SWISH-E oder MG sowie der zusätzlichen Indizierung durch MDR

4.9 Ergebnisgrößen

Die folgende Tabelle enthält für ausgewählte Anfragen die Größe der Ergebnisdatei und die daraus resultierende Größe pro Treffer.

Fragen	Trefferanzahl	Größe der Ergebnisdatei (in KByte)	Größe pro Treffer (in KByte)
holz	92	41	0,45
andreas and heuer	4	3	0,75
datenbank	132	52	0,39
datenbank*	132	52	0,39
schwerin and tourismus	94	40	0,43
stellenangebote	85	34	0,40
gesetze and oberfinanzdirektion	2	3	1,50
gesetze or oberfinanzdirektion	6	3	0,50
holz and verlag	29	25	0,86
holz and not verlag	97	39	0,40
max			**1,5**
∅			**0,63**

Abb.47: Größe der Ergebnisdateien und benötigter Speicherplatz pro Treffer

4.10 Vergleich der Indizierer

Der Indizierer freeWais-sf wurde für Vergleichszwecke in die Tabelle aufgenommen. Da freeWais-sf nicht eingehender betrachtet wurde, bleiben die Aussagen zu diesen Indizierer in der folgenden Tabelle unvollständig und wurden auf „nicht untersucht“ gesetzt.

	MG	**freeWais-sf** **V 2.2.14 (06/00)**	**Glimpse** **4.12 (1999)**	**SWISH-E** **V 1.3.2**
Betriebssystem	- Unix, - BSD, Linux,	- SunOS 5.6, - Linux	- Unix, - BSD, Linux, - Rhapsody (Mac OS X)	- Unix, - Linux, - Windows
Retrieval-Model	- Ranked Queries - Boolesches Retrieval	- Vectorspace-Model	- Boolesches Retrieval	- Ranked Queries
Anfragemöglich-keiten	- Truncation - Stemming	- nur and/or Queries (bei freeWais) - attributierte Anfragen[1]	- reguläre Ausdrücke - Gross-/Kleinschreibung - Suche in Wortteilen - Fehlertoleranz - attributierte Anfragen	- Truncation - Stemming - attributierte Anfragen
Ranking	- ja	- ja	- nein	- ja
Sortierung nach ...	- Relevanz	- Relevanz	- keine	- Relevanz - alphabetisch Sortiert nach einem Feld

[1] Feldsuche

Rückgabe für Treffer	- Dateiname, - Ranking, - Inhalt, - evtl. Dateigröße	- nicht untersucht	- Dateiname - Trefferstelle	- Dateiname, - Ranking, - Dateigröße
simultane Anfragen[1]	- ja	- ja	- Glimpse als Kommandozeilen-tool: ja - Glimpseserver: nein	- ja
Index	- stark komprimiert - zum Index gehören mehrere Dateien	- hoher Speicherbedarf - feinkörniger sortierter Index - verschiedene Indextypen (soundex, phonix)	- kleiner Index 10% - 50% der Datenbasis - Granularität[2] einstellbar (im Allgemeinen gröber al bei anderen Indizierern) - besteht aus mehreren Dateien - invertierte Files zerlegen Dokumentensammlung in Blöcke	- knapp 50 % Datenbasis (z.B.: Indexgröße 100 MB bei ca. 220 MB Dokumenten) - in einer Datei abgelegt
inkrementeller Update des Index	nein[3]	nicht untersucht	teilweise	teilweise (Erstellung mehrerer Indexdateien und Merging)
Stoppwortlisten	nein	nicht untersucht	evtl. über ein zusätzliches Filterprogramm	möglich

[1] Die Ausführung simultaner Anfragen wird durch den Broker nicht unterstützt. Die Fähigkeit der Indizierer kann hier nicht ausgenutzt werden.

[2] im Allgemeinen gröber als bei den anderen Indizierern

[3] evtl. besteht mit mgmerge diese Möglichkeit

Dokumentation	ausführliche Nutzer-, Installations- und Entwicklerdokumen-tation	nicht untersucht	allgemeine Beschreibung der Arbeitsweise mit Beispielen	ausführliche Nutzer-, Installations- und Entwicklerdokumentation
Bemerkung	- Der Index im Sinne einer Datenbasis enthält Inhalte der Indizierten Dokumente	- eingeschränkte Heterogenität, da keine offene Schnittstelle zw. Gatherer und Indizierer - Indizierer kann nur lokal gespeicherte Daten verarbeiten	- max. 65000 indizierbar Dateien [1] - max. Wortlänge 64 Zeichen	

4.11 SwingBroker-Statistik für 06/2001

Top 100

1.	Holz	132
2.	Maler	39
3.	haus	39
4.	Bauordnung	25
5.	e	23
6.	Landkarte	15
7.	Bau	13
8.	Bildungsfreistellungsgesetz	12
9.	Gesetze	12
10.	landkarte	12
11.	"Rügen"	11
12.	Karte	11
13.	schwerin AND elektro	11
14.	Wappen	10
15.	holz	10
16.	veranstaltungen	10
17.	er	9
18.	"wasserqualität"	8
19.	Einwohnerzahl	8
20.	"Schloß" AND Kittendorf	7
	Σ	**417**
21.	A20	7
22.	Statistik	7
23.	"Bevölkerung"	6
24.	"Güstrow"	6
25.	"Küstenschutz"	6
26.	"rügen"	6
27.	Camping	6
28.	Ehrenamt	6
29.	Schulferien	6
30.	Schwerin	6
31.	Stellenausschreibung	6
32.	Wetter	6
33.	a20	6
34.	camping	6
35.	wappen	6
36.	"Kühlungsborn"	5
37.	Barth	5
38.	Dorf AND Schwarz	5
39.	Einwohner	5
40.	Ferientermine	5
41.	Hausboot	5
42.	Immobilien	5
43.	Kommunalverfassung	5
44.	Kultusministerium	5
45.	Landeswappen	5
46.	Neukloster	5
47.	Personalvertretungsgesetz	5
48.	Rerik	5
49.	Stellenangebote	5
50.	Waren	5
51.	altentreptow	5
52.	amtsblatt	5
53.	angeln	5
54.	immobilien	5
55.	kanuverleih	5
56.	karte	5
57.	kultusministerium	5
58.	malchow	5
59.	messen	5
60.	natur AND gesetz	5
61.	stellenausschreibung AND jurist	5
62.	vereine	5
63.	verkehr AND rügen	5
64.	waren	5
65.	wetter	5
66.	"FFH-Gebiete"	4
67.	"gehörlosenschule"	4
68.	"müritz"	4
69.	Ausschreibungen	4
70.	Bauantrag	4
71.	Bundesnaturschutzgesetz	4
72.	Datenbank and Andreas	4
73.	Fahne	4
74.	Feldberg	4
75.	Ferienpark	4
76.	Gerichtskosten	4
77.	Gesundheit	4
78.	Hotels	4
79.	Jugendherberge AND "Darß"	4
80.	Justizministerium	4
81.	Kittendorf	4
82.	Landesfarben	4
83.	Landesgesetze	4
84.	Landeshauptstadt	4
85.	Landtag	4
86.	Lasertechnik	4
87.	Ludwigslust	4
88.	Malchow	4
89.	Stralsund	4
90.	Verfassung	4

91.	Wohnungen	4
92.	ausschreibung	4
93.	datenbank	4
94.	einwohner	4
95.	hausboot	4
96.	kanu	4
97.	lied	4
98.	medeocom	4
99.	mueritz AND linie	4
100.	mv	4
	Σ	**799**

Top 27 Ohne Ergebnis

1.	lehrereinstellung	8
2.	"Ministerpräsidentenkonferenz"	5
3.	chronik AND Kittendorf	5
4.	weber AND gunnar and swing	5
5.	"3.Barther" AND Metal AND "Open-Air"	4
6.	"Freizeitlärm-Richtlinie"	4
7.	"Hermannshöhe"	4
8.	ferientermine	4
9.	stellenausschreibung AND jurist	4
10.	"*ordnung"	3
11.	"Straßenausbaubeitrag"	3
12.	"Wolters-Reisen"	3
13.	"abo-dienst"	3
14.	Hintergrund AND	3
15.	IGA AND 2001	3
16.	Rockentien	3
17.	Urlauf AND auf AND dem AND Bauernhof	3
18.	abendrealschule	3
19.	bad AND sarow	3
20.	burgstargard	3
21.	hundeverordnung	3
22.	lebenspartnerschaft	3
23.	molzow	3
24.	projekt and BUSINESS-MV AND	3
25.	swingerclub	3
26.	vergaberichtlinien	3
27.	wandern AND ohne AND "gepäck"	3
	Σ	**97**

5 Literatur

[1] Heuer, Andreas; Weber, Gunnar. SWING: Eine Suchmaschine mit Datenbankanschluß. In: Saake, Gunter; Sattler, Kai-Uwe, (Hrsg.), *GI-Workshop ``Internet-Datenbanken''*, Nr. 12 (Preprint), Otto-von-Guericke Universität, Magdeburg, September 2000.

[2] A. Heuer, H. Meyer, G. Weber. SWING: Die Suchmaschine des Landesinformationssystems MV-Info, Universität Rostock, Fachbereich Informatik, Lehrstuhl Datenbank- und Informationssysteme
IUK99

[3] Bietz M., Bruder I., Heuer A., Rann A., Weber G.
Entwicklung moderner Information-Retrieval-Techniken in der SWING - Suchmaschine
Rostocker Informatikberichte 2001

[4] mnogosearch.org

[5] Witten I., Moffat A., Bell T.: Managing Gigabytes Compressing and Indexing Documents and Images, Morgan Kaufmann Publishers, Inc. San Francisco California, 1999

[6] www.searchtools.com

[7] SearchTools.com, Search Tools Survey -Popular Product Ratings, 12.07.2001
http://www.searchtools.com/surveys/survey04/ratings.html (13.09.2001)

[8] Tim Shimmin, Alistair Moffat, MG Pages, http://www.mds.rmit.edu.au/mg/, 1999

[9] http://www.etymon.com/Isearch/faq.html

[10] http://www.etymon.com/Isearch/

[11] Morgan, Eric Lease, http://www.infomotions.com/musings/opensource-indexers/, 29.05.2001

[12] http://www.searchtools.com/tools/tools-opensource.html (10/2001)