FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Leistungsmessungen bei der Volltextsuche mit Oracle Ultra Search im wissenschaftlichen Umfeld

Jochen Kreutz

FZJ-ZAM-IB-2006-08

April 2006

(letzte Änderung: 24.4.2006)

Zusammenfassung

Im Zeitalter des Internets und digitaler Medien steht dem Computerbenutzer eine enorme Fülle an Informationen zur Verfügung. Als Hilfsmittel bei der Auffindung relevanter Informationen besitzt die Volltextsuche eine große Bedeutung. Ein Software-Produkt, mit dem sich eine webbasierte Volltextsuche einrichten und administrieren lässt und das für die Volltextsuche innerhalb der Webseiten des Forschungszentrums Jülich zum Einsatz kommt, ist Oracle Ultra Search.

Neben der Treffergüte ist die Zeit für die Durchführung einer Suchabfrage ausschlaggebend für die Qualität einer Volltextsuche. Daher wurden im Rahmen von Leistungsmessungen die Antwortzeiten für verschiedene Suchabfragen mit Oracle Ultra Search im Hinblick auf Stabilität und mögliche Einflussfaktoren untersucht.

Abstract

Nowadays the internet and other digital media provide such an abundant mass of information that it becomes quite difficult for computer users to find the information they really need. In this context full-text search has become a popular method for information retrieval because it offers an easy to use interface to find relevant information. Oracle Ultra Search is a web-based search solution that enables full-text search across multiple repositories. In the Research Centre Jülich, Oracle Ultra Search is already used for the search inside the own websites.

Besides the handling and functionality the quality of a full-text search is strongly connected to the response time behaviour. Therefore, the response times for different full-text queries in Oracle Ultra Search have been investigated with regard to stability and potential factors of influence.

Fachhochschule Aachen
Abteilung Jülich

Fachbereich: Angewandte Naturwissenschaften und Technik
Studiengang: Technomathematik

Leistungsmessungen bei der Volltextsuche mit Oracle Ultra Search im wissenschaftlichen Umfeld

Diplomarbeit von Jochen Kreutz

Jülich, April 2006

Die vorliegende Diplomarbeit wurde in Zusammenarbeit mit dem Forschungszentrum Jülich GmbH, Zentralinstitut für Angewandte Mathematik (ZAM), angefertigt.

Diese Diplomarbeit wurde betreut von:

Referent: Prof. Dr. rer. nat. S. Pawelke

Korreferent: Diplom-Informatiker W. Elmenhorst

Diese Arbeit ist selbständig angefertigt und verfasst. Es sind keine anderen als die angegebenen Quellen und Hilfsmittel benutzt worden.

Jülich, 4. April 2006

Inhaltsverzeichnis

Kapitel 1

Einleitung

Im Zeitalter des Internets und digitaler Medien steht dem Computerbenutzer eine enorme Fülle an Informationen zur Verfügung. Gerade in der Wissenschaft und in modernen Wirtschaftsunternehmen besitzt die Auffindung relevanter Informationen eine große Bedeutung. Die effektive Selektion aus dem riesigen Informationsangebot spielt eine wichtige Rolle für die Produktivität der Mitarbeiter.
Besonders in einer wissenschaftlichen Umgebung wie dem Forschungszentrum Jülich ist der Austausch von Informationen und Ergebnissen zwischen den einzelnen Wissenschaftlern und Mitarbeitern für ein effizientes Arbeiten unerlässlich. Häufig werden z.B. Informationen aus verschiedenen Forschungsbereichen und Projekten als Grundlage für neue Untersuchungen verwendet und die Ergebnisse daraus können wiederum für weitere Projekte relevant sein.
Erschwert wird die Informationssuche dadurch, dass die Dokumente in den unterschiedlichsten Formaten an vielen verschiedenen Speicherorten zur Verfügung stehen z.B. abgelegt in Datenbanken, als Emails oder Webseiten. Die Informationen können strukturiert oder unstrukturiert vorliegen und statisch oder dynamisch sein. Nach Möglichkeit soll jedoch für all diese Informationsquellen ein einheitlicher globaler Zugriff genutzt werden können.
Als Mittel zur Informationsbeschaffung erfreut sich die Volltextsuche großer Beliebtheit und wird von zahlreichen Internetsuchmaschinen sowie von Programmen für die Suche nach lokal gespeicherten Dokumenten eingesetzt. Auch auf den Webseiten des Forschungszentrums wird die Möglichkeit der Volltextsuche geboten, mit der sich z.B. wissenschaftliche Publikationen zu einer bestimmten Thematik suchen lassen. Die Volltextsuche stellt ein komfortables Hilfsmittel bei der Auswahl relevanter Informationen dar, denn sie ist für den Benutzer leicht bedienbar und lässt sich meist über eine Webseite, also von vielen verschiedenen Arbeitsplätzen aus, durchführen.

In der folgenden Arbeit sollen die Funktionsweise und die Möglichkeiten der Volltextsuche am Beispiel von Oracle Ultra Search analysiert werden, da dieses Produkt für die

Volltextsuche auf den Webseiten des Forschungszentrums zum Einsatz kommt. Dabei werden technische Aspekte wie die Indexierung von Dokumenten sowie die Verwendung von Suchoperatoren und Stoppwortlisten berücksichtigt und die Handhabung von Oracle Ultra Search aus Anwender- und Administratorsicht beschrieben. Die Leistungsfähigkeit dieses Produktes wird im Bezug auf die Dauer für die Ausführung verschiedener Suchanfragen auf unterschiedlichen Software-Plattformen untersucht und dokumentiert.

Kapitel 2

Wichtige Begriffe und Methoden bei der Volltextsuche

> *Unter der Volltextsuche versteht man das Auffinden von Texten in einer Vielzahl gleicher oder verschiedenartiger Dateien auf einem Computer, einem Server und/oder im Internet.*[1]

Bei der Volltextsuche wird dem Computeranwender die Möglichkeit geboten, eine Suchanfrage zu stellen, mit deren Hilfe für ihn relevante Dokumente gefunden werden können. Die Eingabe der Suchanfrage geschieht in der Regel über ein Textfeld in einem Eingabefenster. Häufig kann diese Eingabe standortunabhängig über eine Webseite mit Hilfe eines Webbrowsers [2] erfolgen. Die Suchanfrage kann je nach verwendeter Suchmaschine [3] aus ein oder mehreren Wörtern oder Phrasen bestehen, die gegebenenfalls durch Operatoren miteinander verknüpft sein können. Als Ergebnis wird zunächst eine ungeordnete Liste mit Verweisen auf Dokumente erzeugt, die der Suchanfrage entsprechen. Jedoch ist gerade bei großen Mengen gefundener Dokumente eine weitere Einschränkung der Auswahl bzw. eine Sortierung der Ergebnisliste unerlässlich. Daher ordnen moderne Suchmaschinen die Dokumente der Ergebnislisten nach ihrer Relevanz und verwenden dafür zum Teil sehr aufwendige Bewertungssysteme. Diese nehmen z.B. unter Berücksichtigung der Häufigkeit auftauchender Suchbegriffe etc. eine Bewertung (“Scoring”) der Dokumente vor (s. auch Kapitel 3.3.5 auf Seite 33).

Bei der Volltextsuche besteht neben einer sinnvollen Sortierung der Ergebnisliste die Schwierigkeit, dass Dokumenttreffer, obwohl sie der Suchanfrage entsprechen, irrelevant

[1] vgl. [WIKIP].

[2] Webbrowser: Computerprogramm zur Darstellung von Webseiten.

[3] Suchmaschine: Ein Programm, das dem Benutzer z.B. durch eine Sichwort- oder Volltextsuche das Auffinden von Dokumenten ermöglicht, die lokal oder in einem Netzwerk gespeichert sind.

für das gesuchte Thema sein können. Dies wird z.B. durch Homonyme[4] bewirkt. Auf der anderen Seite werden unter Umständen auch Dokumente nicht gefunden, die zwar zum gesuchten Themenkomplex passen, aber z.B. aufgrund eines verwendeten Synonyms der Suchanfrage nicht entsprechen. Wenn etwa nach dem Begriff “Computer” gesucht wird, aber im Dokument der Begriff “Rechner” verwendet wird. Mit Techniken wie der Verwendung eines Thesaurus (s. auch Kapitel 2.4 auf der nächsten Seite und Kapitel 3.3.4 auf Seite 32) wird versucht, diese Probleme in den Griff zu bekommen.

Damit überhaupt in angemessener Zeit - der suchende Anwender möchte ja möglichst schnell Resultate sehen - eine Ergebnisliste erstellt werden kann, sind angesichts der riesigen Datenmengen und der unterschiedlichen Speicherorte der Dokumente spezielle Verfahren wie das Indexieren der Suchbereiche notwendig.

2.1 Indexierung

Da ein Vergleich mit den Inhalten sämtlicher Dokumente nach der Eingabe einer Suchanfrage viel zu lange dauern würde und gewöhnliche Algorithmen und Werkzeuge für die Textsuche, wie z.B. das Unix-Kommando ”grep” [5], angesichts der großen Mengen unterschiedlich gespeicherter Daten überfordert wären, müssen die Dokumente bereits **vor** der eigentlichen Suche besonders aufbereitet und gespeichert werden.

Dazu wird ein invertierter Index erzeugt, der jedem auftauchenden Wort eine Liste von Dokumenten zuordnet, die dieses Wort enthalten. Hierzu muss zunächst der eigentliche Text aus den unterschiedlich formatierten Dokumenten extrahiert und in einzelne Worte zerlegt werden. Unter Umständen wird eine weitere Zerlegung in Wortstämme und Präfixe vorgenommen, um zum Beispiel die Suche nach Wortteilen zu beschleunigen. Zudem werden meist zusätzliche Informationen wie die Position eines Wortes innerhalb des Dokuments gespeichert, um eine Phrasensuche zu ermöglichen. Damit trotz großer Datenmengen [6] ein schneller Zugriff auf den Index ermöglicht werden kann, wird dieser Index in der Regel in Datenbanksystemen abgelegt und für die spätere Erzeugung der Trefferliste zu einer Suchanfrage verwendet.

Es werden jedoch nicht die eigentlichen Dokumente im Index gespeichert (sie verbleiben an ihren ursprünglichen Speicherorten), sondern lediglich Verweise darauf. Der Suchanwender selbst hat in der Regel keinen Einfluss auf die Indexierung, die interne Speicherstruktur der Informationen bleibt ihm verborgen.

[4] Homonym: ein Wort, das für verschiedene Begriffe stehen kann, z.B. Bank (1. Sitzgelegenheit, 2. Geldinstitut)

[5] grep: Unixkommando, um in Dateien nach bestimmten Mustern zu suchen.

[6] Im für die Volltextsuche des Forschungszentrums Jülich erstellten Index, der ein Datenvolumen von knapp 9,5 Gigabyte bestitzt, sind ca. 50.000 Dokumente erfasst.

2.2 Stoppwort

Ein Stoppwort ist ein Wort, das für das Auffinden relevanter Dokumente eher hinderlich als nützlich ist, weil es eine geringe Aussagekraft besitzt. Es wird deshalb von der Indexierung ausgeschlossen. Beispiele für Stoppworte im Deutschen sind: ”ist”, ”ein”, ”als”, ”der”, ”die”, ”das”.

2.3 Abfrageoperatoren

Abfrageoperatoren ermöglichen dem Suchanwender eine präzisere Formulierung seiner Suchanfragen und tragen daher maßgeblich zur Verbesserung des Suchergebnisses bei. So können z.B. logische Verknüpfungen zwischen den Suchbegriffen hergestellt oder die Suchanfrage auf ähnliche Begriffe erweitert werden, um die Ergebnisanzahl zu erhöhen. Letzteres ist vor allem dann nützlich, wenn lediglich das Themengebiet, nicht aber der genaue Begriff bekannt ist - z.B. auf Grund von Synonymen für dieses Wort. Abfrageoperatoren können also die ursprüngliche Suchanfrage sowohl einschränken als auch erweitern. Welche Operatoren bei der Suche zur Verfügung stehen, hängt von der verwendeten Suchmaschine ab. Auf die in Oracle Ultra Search zur Verfügung stehenden Abfrageoperatoren und deren Verwendung wird im Kapitel 3.2.2 auf Seite 17 näher eingegangen. Hilfe zur Verfügbarkeit und Verwendung der Operatoren wird in der Regel von der jeweiligen Suchmaschine geboten, meistens mit Beispielen zur Verdeutlichung. Häufig lassen sich die Abfrageoperatoren auch kombinieren, jedoch kann eine allzu intensive Verschachtelung wegen der wachsenden Komplexität der Suchanfrage zu Problemen (z.B. einer starken Verlangsamung der Suche) führen. Zu berücksichtigen ist auch die Rangfolge der einzelnen Abfrageoperatoren, die bei der Durchführung der Suchabfrage eine Rolle spielt und sich ebenfalls bei verschiedenen Suchmaschinen unterscheiden kann. Sie lässt sich in der Regel durch Klammersetzung beeinflussen.

2.4 Thesaurus

Unter einem Thesaurus versteht man ein Wörterverzeichnis, das aus einer systematisch geordneten Sammlung von Begriffen besteht, die in Beziehung zueinander stehen. Diese Begriffe werden in der Regel durch Äquivalenzrelationen (z.B. Synonyme oder Abkürzungen) und hierarchische Relationen (Ober- und Unterbegriffe) vernetzt. Bei der Volltextsuche können diese Relationen zur automatischen Erweiterung der Suchanfrage verwendet werden. Hierfür stehen gewöhnlich spezielle Abfrageoperatoren zur Verfügung (s. auch Kapitel 3.3.4 auf Seite 32 zur Verwendung eines Thesaurus in Oracle Ultra Search). Meist beschreibt ein Thesaurus ein spezielles Themengebiet, wobei einzelne Themengebiete wiederum durch Relationen ihrer Begriffe miteinander verknüpft werden können.

Abgelegt wird der Thesaurus häufig in einer Datenbank.

Als Beispiel für die Struktur eines Thesaurus dient der folgende Auszug aus dem Thesaurus des ”Informationszentrum für Informationswissenschaft und -praxis”. [7]

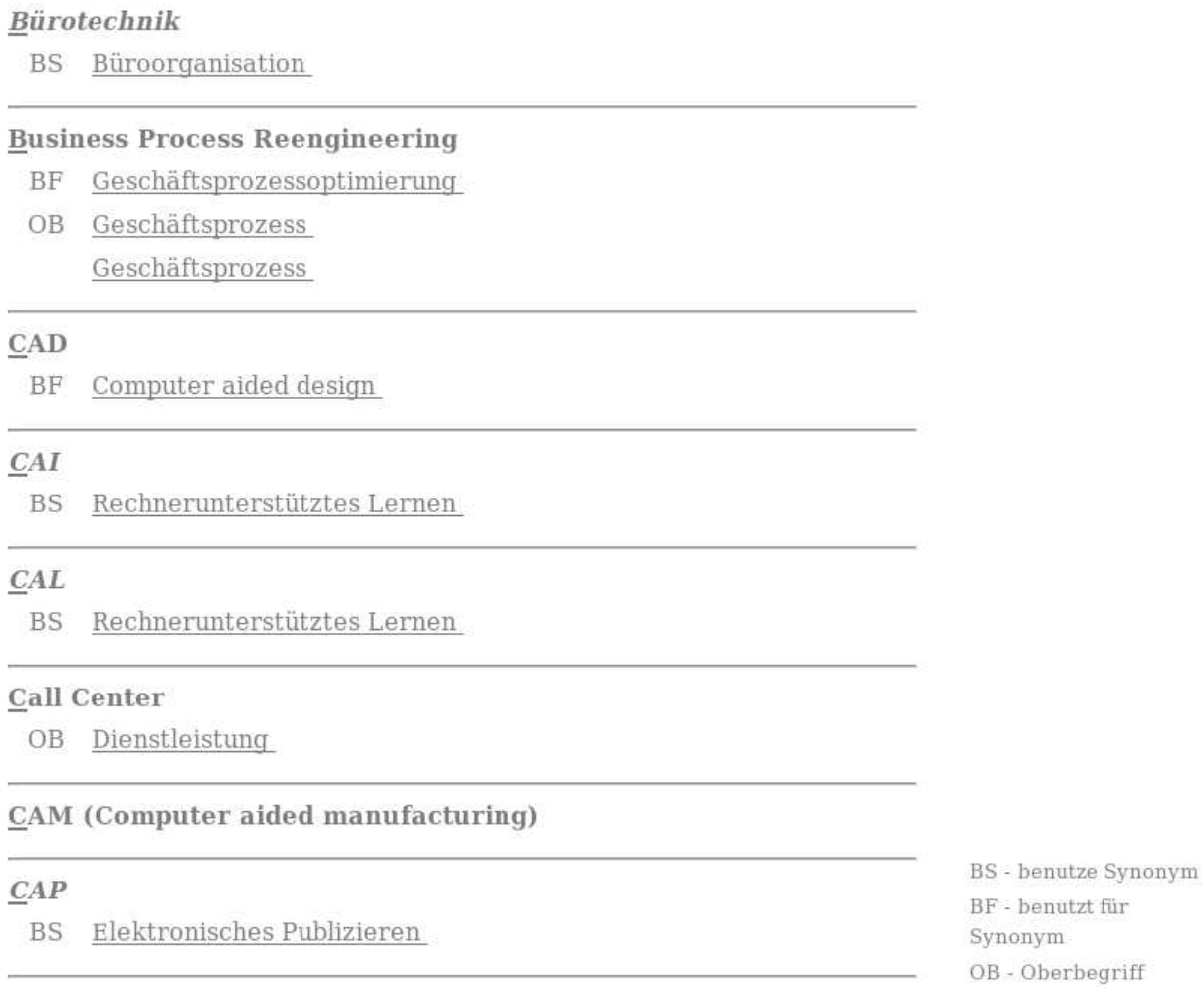

Abbildung 2.1: Beispiel für die Struktur eines Thesaurus (kurzer Auszug)

[7] Zu finden unter: ”http://www.infodata-edepot.de/thesaurus/START.HTM”.

Kapitel 3

Oracle Ultra Search

Ultra Search ist ein von Oracle entwickeltes weborientiertes Suchwerkzeug, das eine einheitliche Volltextsuche über verschiedene Datenquellen ermöglicht. Oracle Ultra Search ist in der Oracle Datenbank, der Oracle Collaboration Suite und dem Oracle Application Server bereits als Feature integriert und wird von Oracle als multifunktionale “out-of-the-box” Suchlösung[1] angepriesen. Oracle Ultra Search ist kein eigenständiges Programm, sondern eine auf die Verwendung unter den oben genannten Oracle-Produkten hin ausgerichtete Ergänzung der durch die von Oracle Text zur Verfügung gestellten Suchfunktionalität (s. Kapitel 3.3 auf Seite 28 für nähere Informationen über das Zusammenspiel von Oracle Ultra Search und Oracle Text). Ziel des Produktes ist es, die Informationsbeschaffung durch eine leicht zu bedienende Volltextsuche zu erleichtern und dadurch den Zeitaufwand zur Suche nach relevanten Dokumenten zu reduzieren.

Die Architektur von Oracle Ultra Search bestehend aus einem ”Backend”, einem ”Middle Tier”[2] und einem ”Crawler” ermöglicht eine Installation der einzelnen Komponenten auf verschiedenen Rechnern zwecks Verbesserung der Sicherheit und der Performance. Das Backend besteht unter anderem aus dem Oracle Ultra Search Repository und Oracle Text. Es ist für die Indexierung sowie die Aufbereitung der Suchergebnisse zuständig. Im Middle Tier befinden sich das Admin Tool zur Steuerung der Volltextsuche (s. auch Kapitel 3.1.4 auf Seite 13), für die Programmierschnittstellen benötigte Programmdateien und die Beispielsuchanwendung (s. auch Kapitel 3.2.4 auf Seite 23). Der Crawler wird im folgenden Kapitel näher beschrieben.
Abbildung 3.1 auf der nächsten Seite skizziert die Architektur von Oracle Ultra Search.

[1] das heißt, alle benötigten Funktionen zur Einrichtung einer Suchanwendung sind integriert und sofort einsatzbereit

[2] Aufteilung der Architektur in mehrere Schichten (engl.: tier = Schicht)

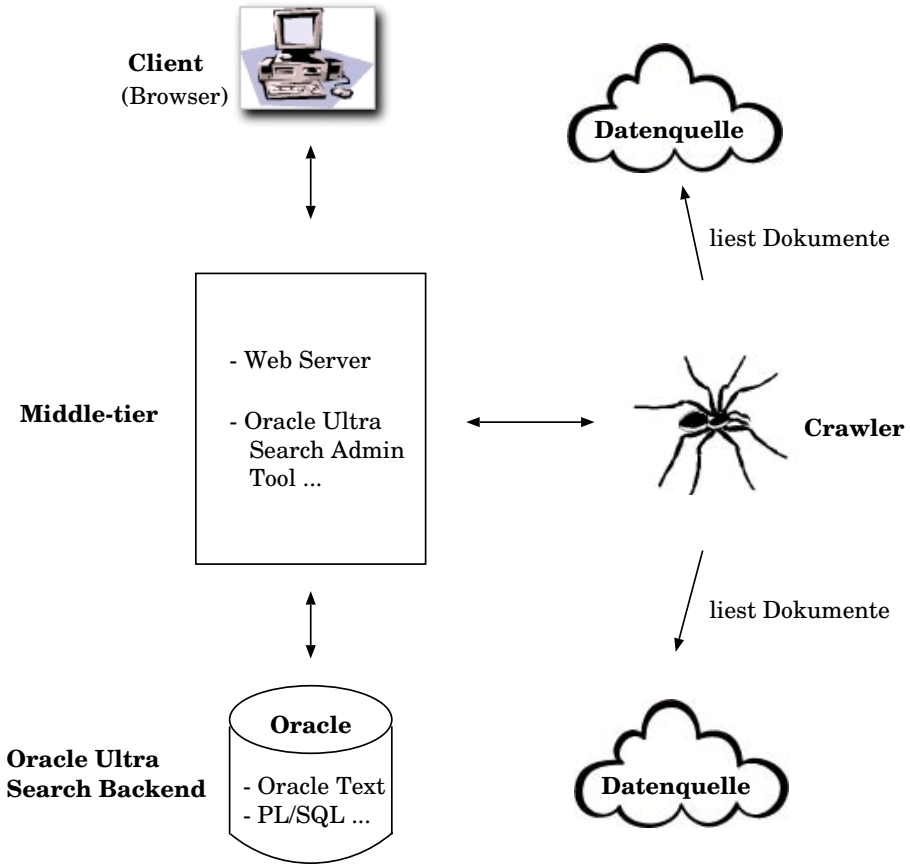

Abbildung 3.1: Oracle Ultra Search Architektur

3.1 Funktionsweise von Oracle Ultra Search

3.1.1 Crawler

Zunächst müssen die Dokumente der unterschiedlichen Datenquellen (Webseiten, Datenbanktabellen, Textdateien etc.), die in den Index aufgenommen und bei der späteren Suche berücksichtigt werden sollen, von ihren physikalischen Speicherorten gelesen werden, bevor sie weiterverarbeitet werden können.
Für diese Aufgabe verwendet Oracle Ultra Search einen sogenannten “Crawler”, der als Java-Prozess implementiert ist und gegebenenfalls als ”Remote-Crawler” (s. auch Kapitel 3.2.5 auf Seite 26) auf einem entfernten Rechner installiert sein kann. Er wird vom Oracle Server aktiviert und holt nach dem Starten die verschiedenen Dokumente aus den einzelnen Quellen, um sie zur Indexierung bereitzustellen. Dabei werden z.B. bei Webseiten die enthaltenen Links ausgewertet, um sämtliche Dokumente innerhalb der

Datenquelle zu erreichen und gleichzeitig ein “im Kreis laufen” sowie die Beschaffung unerwünschter Dokumente außerhalb der zu durchsuchenden Datenquellen (z.B. einer bestimmten Domain) zu vermeiden.

Gerade bei Webseiten ist es wichtig, die Bereiche, die vom Crawler berücksichtigt werden sollen, einzuschränken, um die Datenmenge überschaubar zu halten und nur die relevanten Informationsvorräte in die Suche einzubinden. So ist es zum Beispiel wünschenswert, die Suche innerhalb des Forschungszentrums auf die Domain *”fz-juelich.de”* zu beschränken. Um diese Begrenzung des Crawlers auf bestimmte Bereiche zu realisieren, bietet Oracle Ultra Search dem Administrator der Suchanwendung verschiedene Techniken. [3] Dem Crawler können explizite Regeln in Form von URL-Begrenzungen [4] mitgeteilt und auf diese Weise bestimmte *Hosts und* Domains (inklusive spezieller Ports) und deren Unterverzeichnisse ausdrücklich in das ”Crawlen” einbezogen oder von der Suche ausgeschlossen werden. Eine weitere Möglichkeit zur Einschränkung des Crawlerbereichs stellt das ”robots.txt”- Protokoll [5] dar, das Webmastern die Möglichkeit bietet, Crawler von ihren Webseiten fernzuhalten. Dazu wird entweder eine Datei names ”robots.txt” im Wurzelverzeichnis der Domain abgelegt, die der Crawler ausliest und daraus die Regeln für das Durchsuchen der Seiten übernimmt oder der Umgang über ein Metatag[6]geregelt. Der Oracle Ultra Search Administrator kann entscheiden, ob der Crawler diese sogenannte Robots-Exclusion berücksichtigen soll. Zudem bietet Oracle Ultra Search die Möglichkeit, die Reichweite des Crawlers über die ”Crawling Depth”, also die Verschachtelungstiefe der Webseiten zu beeinflussen und zu bestimmen, wie weit der Crawler den Links einer gegebenen Startseite folgen soll.

Da die Datenquellen in der Regel dynamisch sind, ist eine regelmäßige Aktualisierung des Indexes notwendig, um stets die aktuellsten Dokumente bei der Suche berücksichtigen zu können und ein Auftauchen veralteter oder bereits nicht mehr existierender Dokumente in der Trefferliste zu vermeiden. Dazu lassen sich mittels Oracle Ultra Search verschiedene Ausführungspläne für das Crawlen der Datenquellen erstellen, die den Crawler in durch den Benutzer festgelegten Zeitintervallen automatisch starten und so den Index aktualisieren. Beim ersten Crawler-Durchlauf werden sämtliche Dokumente einer Datenquelle berücksichtigt, bei späteren Durchläufen nur die geänderten und neuen Dokumente verarbeitet, sofern der Administrator nicht explizit angibt, bei jedem Durchlauf alle Dokumente aufzubereiten. Überprüft wird die Änderung der Dokumente durch die Auswertung von Informationen aus dem Übertragungsprotokoll und mit Hilfe von Prüfsummen.

[3] vgl. hierzu auch die Ausführungen zu dem im Forschungszentrum Jülich eingesetzten ”internen” und ”externen” Crawlen in Kapitel 3.2.3 auf Seite 22

[4] URL: Uniform Resource Locator - identifiziert Quellen über ihre Zugriffsart und ihren Speicherort

[5] vgl. [ROBOT]

[6] Meta-Tags: HTML-Elemente auf einer Webseite, die Metadaten (z.B. Autor) über die betreffende Webseite enthalten.

3.1.2 Indexierung

Zur Indexierung der zuvor vom ”Crawler” gesammelten Dokumente greift Oracle Ultra Search auf die Oracle Text Technologie zurück (s. auch Kapitel 3.3 auf Seite 28). Das Crawling und die Indexierung werden durch den Administrator gesteuert und laufen in der Regel vom Suchanwender unbemerkt im Hintergrund ab. Bei Erstellung des Indexes werden die Dokumente in mehreren aufeinander folgenden Schritten (vgl. Abbildung 3.2) bearbeitet und schließlich die enthaltenen Wörter dem Index hinzugefügt.

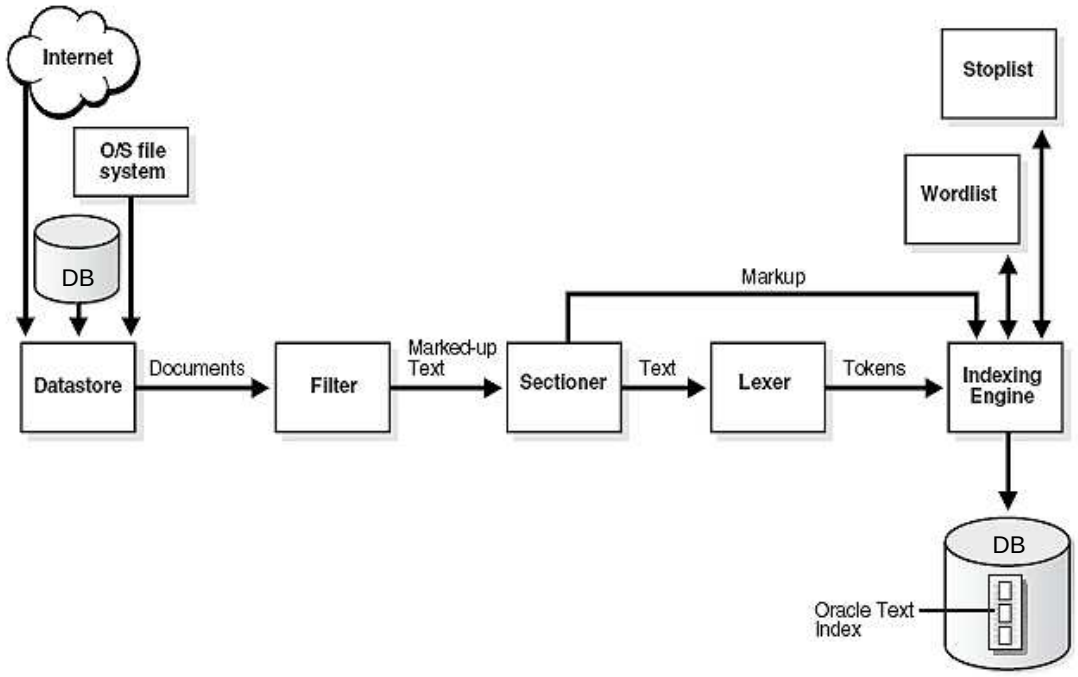

Abbildung 3.2: Indexerstellung mit Oracle Text

Zunächst werden die Dokumente aus dem jeweiligen ”**Datastore**”, ihrem Aufbewahrungsort (z.B. einem Dateisystem), eingelesen. Bei der Verwendung von Oracle Ultra Search befinden sich die zu indexierenden Dokumente in zuvor durch den Crawler erzeugten Cache-Files. Der Speicherort dieser Cache-Files kann durch den Administrator festgelegt werden.

Als nächstes wird ein ”**Filter**” angewendet, der die Aufgabe hat, die unterschiedlich formatierten Dokumente (z.B. Word- oder Pdf-Dokumente, s. auch Kapitel 3.2.1 auf Seite

17) in ein einheitliches Format zu konvertieren. Als output wird ein ”Marked-up Text” (in Form von HTML-Code) erzeugt, der den Erhalt eingebetteter Informationen wie Titel oder Überschriften erlaubt. Zudem kann der Filter den Zeichensatz eines Dokumentes an den in der Datenbank verwendeten Zeichensatz anpassen. Klartext- und HTML-Dokumente brauchen nicht gefiltert zu werden.

Nach dem Filtern wird der ”Marked-up Text” an einen sogenannten ”**Sectioner**” übergeben. Falls Abschnittsinformationen (etwa Überschriften oder Formatierungen) enthalten sind z.B. in Form von HTML-Tags [7] , werden diese durch den Sectioner vom eigentlichen Text getrennt. Die Informationen, an welcher Stelle die Abschnitte beginnen und enden, werden dann an die ”Index Engine” weitergeleitet. Der eigentliche Text wird an den ”Lexer” übergeben.

Der ”**Lexer**” zerlegt den Text in Abhängigkeit der vorliegenden Sprache in einzelne ”Token”. Diese Token entsprechen in der Regel einzelnen Wörtern. Zudem werden Sonderzeichen entfernt und die Wörter in Großbuchstaben umgewandelt, sofern der Administrator nicht explizit angibt, dass bei der Indexierung und der späteren Suche zwischen Groß- und Kleinschreibung unterschieden werden soll (s. auch Kapitel 3.3.3 auf Seite 32). Sollen spätere Themeninformationen zu den einzelnen Dokumenten verfügbar sein, stellt der Lexer diese Themeninformationen durch Analysieren des Textes bereit.

Im letzten Schritt wird durch die ”**Indexing Engine**” der invertierte Index, der die einzelnen Wörter und die Verweise auf ihre zugehörigen Dokumente enthält, erzeugt und in der Datenbank abgelegt. Hierbei werden die zu Stoppwortlisten zusammengefassten Stoppworte vom Index ausgeschlossen. Zusätzliche Informationen, zum Beispiel für die Verwendung einer Wortstamm- oder einer Fuzzy-Suche (s. auch Kapitel 3.2.2 auf Seite 17), gehen als eine sogenannte ”Wordlist” in den Index mit ein.

3.1.3 Durchführen einer Suchanfrage

Die Eingabe der Suchanfrage findet über eine Browser-basierte Eingabemaske statt. Hierbei gibt es zwei verschiedene Formen der Abfrage: eine einfache, bei der lediglich der Suchstring eingegeben wird (vgl. Abbildung 3.3 auf der nächsten Seite) und eine ausführliche, bei der neben dem Suchstring zusätzliche Einschränkungen bezüglich Metadaten und Datengruppen (s. Kapitel 3.2.4 auf Seite 22) angegeben werden können (vgl. Abbildung 3.4 auf der nächsten Seite).
Nach der Eingabe durch den Benutzer wird eine Abfrage des Indexes durchgeführt, um für die Suchanfrage relevante Dokumente zu ermitteln. Diese werden dem Benutzer als

[7] Tag - eine strukturierende Hervorhebung, mit der sich Textbereichen eine Bedeutung zuordnen lässt

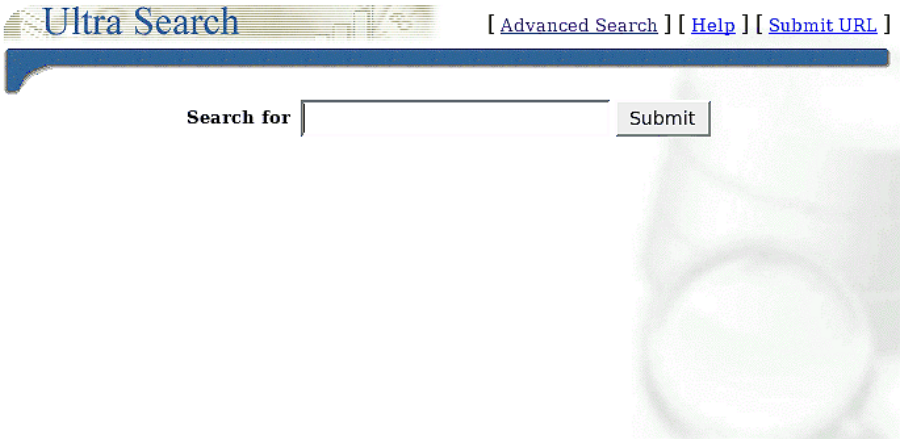

Abbildung 3.3: Einfache Eingabe einer Suchanfrage in Oracle Ultra Search

Abbildung 3.4: Ausführliche Eingabe einer Suchanfrage in Oracle Ultra Search

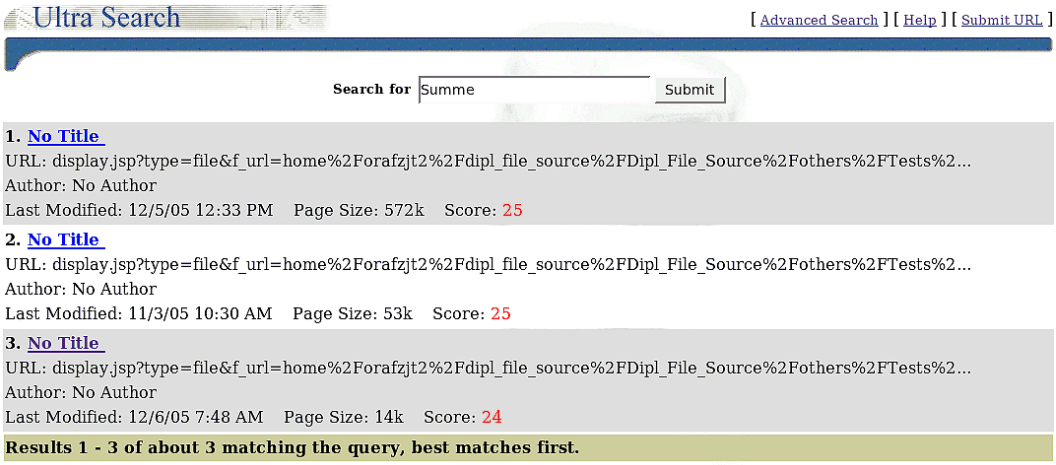

Abbildung 3.5: Trefferliste zu einer Suchanfrage in Oracle Ultra Search (ohne Dokumentauszüge der Treffer)

Trefferliste bestehend aus Verweisen auf die einzelnen Dokumente angezeigt (vgl. Abbildung 3.5).
Eine Hilfeseite, die die Verwendungen der zur Verfügung stehenden Abfrageoperatoren (s. auch Kapitel 3.2.2 auf Seite 17 für nähere Informationen zu den in Oracle Ultra Search verfügbaren Abfrageoperatoren) erklärt, ist ebenfalls vorhanden. Zudem erhält der Suchanwender, sofern der Administrator dies erlaubt, über den Link ”Submit URL” die Möglichkeit, eine eigene URL hinzuzufügen, die dann beim nächsten Crawler-Vorgang einbezogen werden soll.

3.1.4 Ultra Search Admin Tool

Zur Verwaltung von Ultra Search stellt Oracle ein Browser-basiertes Administrations-Werkzeug zur Verfügung, welches dem Oracle Ultra Search Administrator ermöglicht, die verwaltungstechnischen Aufgaben mittels Dialog-Boxen und Formularen zu bewältigen. Durch die einfache Gestaltung der Weboberfläche über Karteireiter (s. Abbildung 3.6 auf der nächsten Seite) wird dem Administrator eine intuitive Steuerung von Oracle Ultra Search ermöglicht, ohne für die Einrichtung und Überwachung der Suchanwendung spezielle Kommandos erlernen zu müssen. Zudem ist eine Online-Hilfe in verschiedenen Sprachen verfügbar. Über das Admin Tool lassen sich detaillierte Einstellungen bezüglich der Index-Eigenschaften, Crawler-Parameter und der Abfrage-Optionen vornehmen. Der Index zusammen mit den oben genannten Einstellungen wird als Oracle Ultra Search Instanz bezeichnet, nicht zu verwechseln mit einer Datenbankinstanz. Zu jeder Ultra Search Instanz gehört genau ein Datenbankbenutzer, der die Instanz administriert.

Zu den Crawler-Einstellungen gehören z.B. die Anzahl der Crawler-Threads[8], die vom Crawler für die Dokumentbeschaffung erzeugt werden, die Verwendung automatischer Spracherkennung bzw. einer Standard-Sprache, die Wartezeit für das Erreichen einer Webseite und Einstellungen für das Schreiben von Cache- und Logfiles. Weiterhin lassen sich Ausführungspläne für das Crawlen sowie die zu durchsuchenden Datenquellen verwalten, und es können Parameter für den Web-Zugang angepasst werden: es besteht z.B. die Möglichkeit, dem Crawler für bestimmte Seiten eine Kennung bzw. Formulareinträge mitzuteilen, die dann beim Crawlen automatisch für den Zugang zu der entsprechenden Webseite benutzt werden.

Das Festlegen von Dokumentattributen (s. auch Kapitel 3.2.1 auf der nächsten Seite) und das Erzeugen von themenspezifischen Datengruppen (s. Kapitel 3.2.4 auf Seite 22) für die spätere Suche sind ebenfalls über das Admin Tool möglich. Darüber hinaus bietet es dem Administrator mit Hilfe von Statistiken Informationen über den Verlauf des Crawlens und bisher getätigte Suchabfragen.

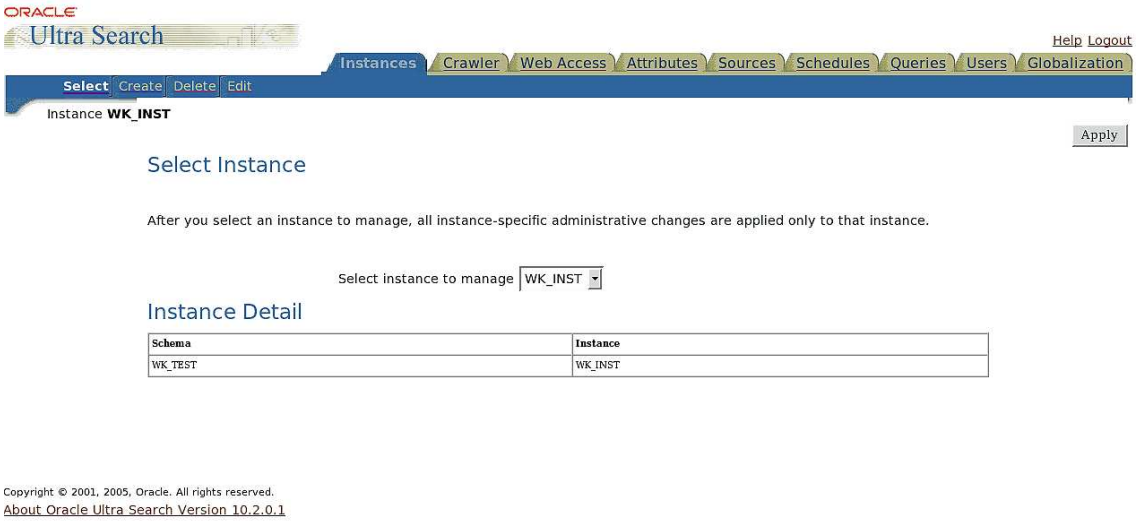

Abbildung 3.6: Oberfläche des Ultra Search Admin Tool

[8]Thread: ist in der Informatik bei der Programmausführung meist die kleinste Einheit, die verwaltet werden kann; die parallele Ausführung mehrerer Threads wird häufig zur Geschwindigkeitsverbesserung genutzt.

3.2 Eigenschaften und Fähigkeiten von Oracle Ultra Search

3.2.1 Datenquellen und Dokumenteigenschaften

Oracle Ultra Search bietet die Möglichkeit verschiedene Formen von **Datenquellen** in die Suche einzubeziehen. Durch eine Datenquelle wird eine Sammlung von Dokumenten beschrieben, die durch die Eigenschaften ihrer Speicherorte, wie z.B. eine Webdomain oder ein E-Mail-Postfach charaktarisiert wird. Oracle Ultra Search kennt folgende Typen von Datenquellen:

Web Source: Die Web Source ist die meist verwendetste Form der Datenquelle. Sie repräsentiert den Inhalt verschiedener Webseiten. Dabei kann der Bereich, der für die Suche berücksichtig werden soll, wie im vorherigen Abschnitt ”Crawler” beschrieben, explizit eingegrenzt werden. Die auf einem Webserver abgelegten Dokumente einer Web Source werden über das ”HTTP”[9] oder als gesicherte Verbindung über das ”HTTPS”[10] -Protokoll ausgelesen.

Table Source: Bei dieser Form der Datenquelle werden Dokumente berücksichtigt, die innerhalb von Datenbanktabellen abgelegt sind. Dabei muss es sich im Gegensatz zur Datenbank, in der sich der Oracle Ultra Search Index befindet, nicht zwangsläufig um eine Oracle Datenbank handeln, jedoch muss die für die Datenquelle zu verwendende Datenbank ODBC[11]-konform sein. Angesprochen wird die zur Table Source gehörende Datenbank über einen Datenbanklink. Die Dokumente können innerhalb der Tabellen z.B. im HTML- oder Plain Text-Format vorliegen.

E-Mail Source: Mit einer E-Mail Source lassen sich alle E-Mails, die an eine bestimmte Mailadresse gesendet werden, in der Suche berücksichtigen. Voraussetzung ist, dass ein IMAP-Account[12] zur Verfügung steht. Alle E-Mails, die durchsucht werden sollen, müssen sich im Posteingang dieses IMAP-Accounts befinden. Sie werden bei der Aufnahme in den Index durch den Crawler vom IMAP-Server gelöscht und in einem lokalen Verzeichnis, das bei Einrichtung der E-Mail Source angegeben werden kann, gespeichert. Von dort werden sie bei Auswahl über die Trefferliste von einem Java-basierten Email Viewer im Browser dargestellt. Die gewöhnungsbedürftige Eigenschaft, dass die indexierten E-Mails anschließend nicht mehr auf dem Mail-Server, ihrem ursprünglichen

[9] HTTP: Hypertext Transfer Protocol - ein Protokoll zur Übertragung von Daten über ein Netzwerk

[10] HTTPS: Hypertext Transfer Protocol Secure - ein Netzwerkprotokoll, das eine gesicherte HTTP-Verbindung zwischen Rechnern ermöglicht

[11] Open Data Base Connectivity - eine ursprünglich von Microsoft entwickelte standardisierte Datenbankschnittstelle, über die sich neben Oracle auch weitere gängige Datenbanksysteme wie z.B. MySQL oder PostgreSQL ansprechen lassen

[12] IMAP: Internet Message Access Protocol - erlaubt den Zugriff auf und die Verwaltung von empfangenen E-Mails

Speicherort, wiederzufinden sind und von dort nicht mehr abgerufen werden können, hat zwei Gründe. Zum einen ist dieses Vorgehen nötig, um eine effiziente Dokumentbeschaffung und Indexierung zu ermöglichen, da andernfalls der Crawler bei jedem Durchlauf alle Emails von Neuem untersuchen und feststellen müsste, welche von ihnen bereits indexiert sind. Zum anderen ist diese Art der Datenquelle nicht für individuelle E-Mail-Accounts gedacht, sondern für die Suche in Mailing-Listen entwickelt worden, bei denen wegen der großen Anzahl von E-Mails, die den Account erreichen, das Lesen jeder einzelnen Mail zu zeitaufwendig wäre und eine automatische Selektion der relevanten Mails benötigt wird.

File Source: Hierbei wird über das ”file”-Protokoll auf eine Dokumentsammlung auf dem lokalen Rechner zugegriffen. Im Gegensatz zur Table Source muss dies der Rechner sein, auf dem Oracle Ultra Search installiert ist. Diese Einschränkung lässt sich jedoch zum Beispiel durch Anbinden eines entfernten Filesystems (mounten) an diesen Rechner umgehen. Wie bei einer Web Source können die (Unter-) Ordner und Dokumente, die in die Suche eingebunden werden sollen, über Pfadregeln explizit bestimmt werden.

Oracle Source: Mit dieser Form der Datenquelle ist es möglich, bereits vorhandene Indexe zu verwenden, ohne die Dokumente der Datenquelle vorher crawlen und indexieren zu müssen. So lässt sich eine einheitliche Suche über mehrere auf unterschiedliche Weise zur Verfügung gestellte Indexe durchführen. Allerdings muss dafür vom Administrator ein eigener Suchadapter entwickelt und verwendet werden, der den Zugriff auf den externen Index realisiert. Als Oracle Source kann z.B. eine Oracle Application Server Portal [13] Source benutzt werden.

User-Defined Source: Zusätzlich zu den oben beschriebenen von Oracle bereitgestellten Datenquellen kann der Benutzer eigene Typen von Datenquellen entwickeln. Dies kann dazu verwendet werden, um Dokumentablagen und Management Systeme mit eigenen Schnittstellen und Datenstrukturen wie z.B. ”Lotus Notes” [14] oder ”Documentum”[15]in die Suche einzubinden. Auch hier ist es die Aufgabe des Oracle Ultra Search Administrators, die benötigten Java-Module zur Anpassung des Ultra Search Crawlers zu implementieren, um den Zugriff auf die Datenquellen zu ermöglichen (s. auch Kapitel 3.2.4 auf Seite 24).

Bis auf die E-Mail Source besitzen alle Datenquellen die Eigenschaft, dass die zugehörigen Dokumente an ihren ursprünglichen physikalischen Speicherorten verbleiben und nicht in die Datenbank transferiert werden müssen, dort werden lediglich Verweise auf die Dokumente gespeichert. Die einzelnen Datenquellen können unabhängig voneinan-

[13] Browser-basierte Umgebung für die Entwicklung und Bereitstellung von personalisierten E-Business Portalen

[14] s. auch: http://www.bsi.bund.de/gshb/deutsch/baust/07007.html

[15] s. auch http://www.documentum.de

der in die Ausführungspläne des Crawlers eingebunden und so in unterschiedlichen Zeitintervallen auf Aktualisierungen überprüft werden. Dies ist sinnvoll, da die Dynamik der Datenquellen sich stark unterscheiden kann (z.B. wird das Email-Postfach sich häufiger ändern als eine File Source mit Archivdaten).

Die Dokumente der einzelnen Datenquellen können - selbst innerhalb der gleichen Datenquelle - **verschiedensprachig** sein. Enthält ein Dokument keine Informationen über seine Sprache, versucht Oracle Ultra Search, die Sprache des Dokumentes selbst herauszufinden. Gelingt dies nicht, z.B. weil ein Dokument aus deutschen und englischen Abschnitten besteht, wird eine Standardsprache verwendet, die der Benutzer bestimmen kann. Für jedes Dokument werden, soweit vorhanden, sprachspezifische Lexer und Stoppwortlisten (s. auch Kapitel 2.2 auf Seite 5) verwendet. Bei der eigentlichen Suche können Übersetzungen für Dokumentattribute und Datengruppen angegeben werden, so dass dieselbe Applikation in verschiedenen Sprachen verwendet werden kann.
Oracle Ultra Search unterstützt des Weiteren **mehrere Zeichensätze** und beherrscht den Umgang mit **zahlreichen Dokumenttypen**. Die Formate ”HTML” und ”Plain Text” werden grundsätzlich bei der Dokumentbeschaffung berücksichtigt. Weitere Dokumenttypen unterschiedlicher Kategorien wie Tabellen-Formate (z.B. Excel) oder Präsentations-Formate (z.B. Power Point) können explizit über das Admin Tool ausgewählt werden (vgl. Abbildung 3.7 auf der nächsten Seite). Textinformationen aus Grafiken hingegen lassen sich nicht extrahieren und durchsuchen. Zudem gibt es zum Teil Einschränkung bezüglich der Dokumentstruktur (z.B. werden Kommentare in MS Word Dokumenten nicht vom Filter berücksichtigt).

Dokumentattribute und Metatags, die zusätzliche Informationen zu einem Dokument bereitstellen, können in die Suche einbezogen werden, um die Suchanfrage zu präzisieren und die Ergebnisqualität zu verbessern. Auf diese Weise lässt sich z.B. die Suche auf die Dokumente eines bestimmten Autors einschränken, oder die Ergebnisliste zum Suchbegriff ”Computer” auf deutsche Dokumente reduzieren. Die Angaben zur Dokumentbeschreibung werden während des Crawling- bzw. Indexierungs-Prozesses ausgelesen, einem Suchattribut zugeordnet und zusammen mit den übrigen Informationen im Index abgespeichert. Bei Webseiten z.B. werden dafür die Informationen im HTTP-Header oder in den eingebetteten HTML-Metatags genutzt. Die Suchattribute können durch den Administrator angepasst werden.

3.2.2 Abfrageoperatoren in Oracle Ultra Search

Oracle Ultra Search ermöglicht wie die meisten anderen Suchmaschinen die Suche nach einzelnen oder mehreren Wörtern sowie Textphrasen und stellt logische Abfrageoperatoren sowie eine Suche mit einem Platzhalter (Wildcard) zur Verfügung. Oracle Ultra Search greift dabei auf die Suchfunktionalität von Oracle Text zu (vgl. Kapitel 3.3 auf

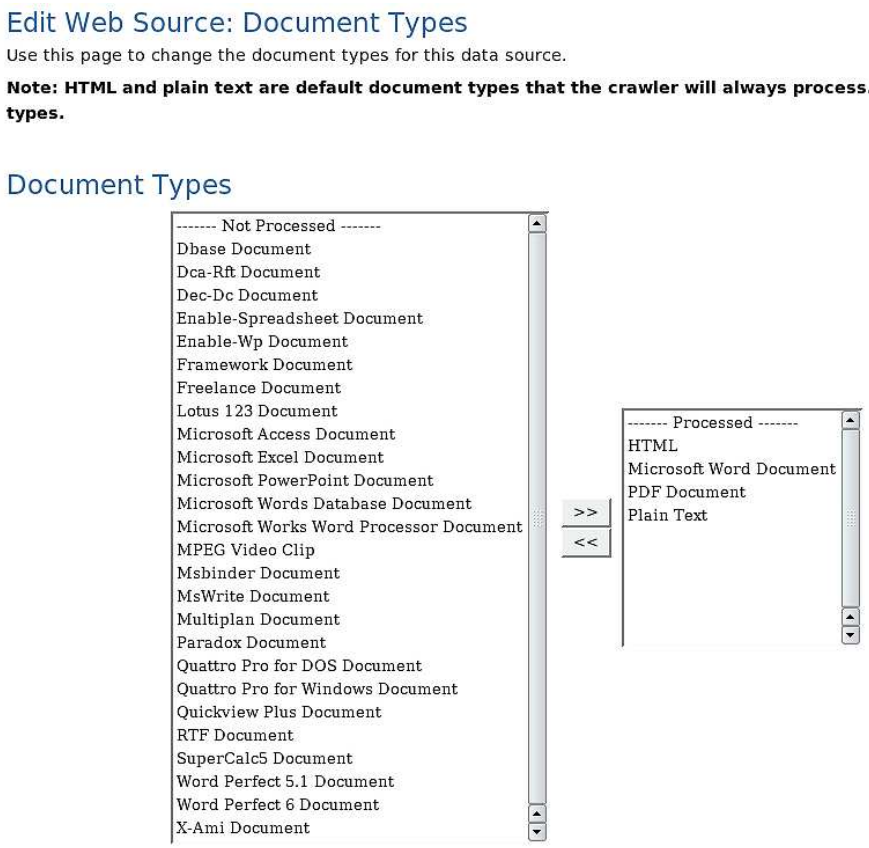

Abbildung 3.7: Auswahl der zu berücksichtigenden Dokumenttypen über das Oracle Ultra Search Admin Tool

Seite 28).

Mehrere einzelne Begriffe werden automatisch mit einer **”ODER”-Bedingung** verknüpft. Dabei werden alle Dokumente gefunden, die mindestens einen der Begriffe enthalten. Soll ein Begriff auf jeden Fall im Dokument enthalten sein, kann dies durch ein vorangestelltes **”+”** erreicht werden. So werden für die Suchanfrage ”+ Festplatte + Diskette” nur Dokumente gefunden, die BEIDE Begriffe enthalten. Diese Anfrage enstpricht einer **”UND”-Bedingung**. Mit dem Operator **”-”** lassen sich bestimmte Begriffe von der Suche ausschließen. Das heißt, dass nur Dokumente gefunden werden, die den entsprechenden Term nicht enthalten. Zum Beispiel liefert die Suchanfrage ”Oracle - Datenbank” nur solche Dokumente, die den Begriff ”Oracle”, aber nicht den Begriff ”Datenbank” enthalten.

Bei der Verwendung des **Platzhalters** (Wildcards) **”*”** wird die Suche auf Wörter erweitert, die an der Stelle des Platzhalters ein oder mehrere beliebige Zeichen enthalten

können. Für die Suchanfrage ”s*t*re” werden zum Beispiel Dokumente mit den Begriffen ”Software” oder ”sichtbare” gefunden. Die Benutzung von Wildcardzeichen kann sehr nützlich sein, wenn die genaue Schreibweise eines Wortes unbekannt ist. Jedoch ist zu beachten, dass die Verwendung von Platzhaltern einen negativen Einfluss auf die Dauer der Suche haben kann und der Platzhalter an erster Position eines Suchwortes nicht zulässig ist. Der Einsatz von Platzhaltern sollte also durchdacht sein.

Neben den oben beschriebenen elementaren Abfrageoperatoren bietet Oracle Ultra Search die Möglichkeit, weitere Operatoren zu verwenden, die allerdings zunächst durch eine Anpassung der zur Beispielanwendung gehörenden JavaServer Pages durch den Administrator verfügbar gemacht werden müssen (s. hierzu auch Kapitel 3.3.3 auf Seite 31). Viele der komplexen Abfrageoperatoren sind abhängig von der Sprache der indexierten Dokumente.

Der **Stem-Operator** sucht nach Wörtern, die dieselbe linguistische Wurzel haben. So wird etwa die Suche nach dem Begriff ”lief” auch auf die Begriffe ”laufen” und ”gelaufen” erweitert. Bei dem von Oracle verwendeten Stemmer handelt es sich um ein lizensiertes Produkt der Firma XSoft (Xerox), mit dem sich ein Stemming (Zurückführen der Wörter auf ihren Wortstamm) in verschiedenen Sprachen wie Englisch, Deutsch, Spanisch etc. durchführen lässt.
Um das Stemming bei der Informationssuche verwenden zu können, werden neben den einzelnen Wörtern der Dokumente auch deren Wortstämme im Index abgelegt. Bei der anschließenden Suchabfrage wird dann ebenfalls ein Stemming der eingegebenen Wörter durchgeführt und die Wortstämme mit denen im Index verglichen.
Eine kurze Beschreibung der Funktionsweise eines Stemmers befindet sich in Anhang B auf Seite 83.

Durch den **Soundex-Operator** wird eine Suche nach ähnlich klingenden Wörtern mit unterschiedlicher Schreibweise ermöglicht. Häufig wird der Soundex-Operator verwendet, um unterschiedlich geschriebene Eigennamen zu suchen (z.B. Meier, Meyer, Maier). Jedes Wort enthält entsprechend der vorkommenden Buchstaben einen Soundex-Code, über den der Klang der Wörter später verglichen werden kann. Der von Oracle verwendete Soundex-Code wird nach einem Algorithmus von Donald E. Knuth ermittelt. Dieser weist jedem Wort eine Kombination aus einem Buchstaben und drei Ziffern zu, die nach folgenden Regeln ermittelt wird:

1. Der erste Buchstabe des Wortes wird als Buchstabe im Soundex-Code verwendet
2. Aus den übrigen Buchstaben werden die Vokale (und Umlaute) sowie ’w’ und ’y’ entfernt und den ersten drei verbleibenden Buchstaben nach folgender Tabelle eine Ziffer zugewiesen:

Buchstaben	Ziffer
b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
l	4
m, n	5
r	6

3. Stehen zwei oder mehr Buchstaben mit der gleichen zugehörigen Ziffer nebeneinander, wird nur der erste berücksichtigt. Beispiel: Jackson → Jacon

4. Durch Hinzufügen von Nullen bzw. Entfernen von überflüssigen Ziffern (falls mehr als drei vorhanden sind) wird der Code auf die Form ”Buchstabe Ziffer Ziffer Ziffer” gebracht. Beispiel: Meyer → M600

Der Soundex-Code zu einem Wort lässt sich in einer Oracle-Datenbank mit folgendem SQL-Befehl abfragen:

```
SQL> SELECT SOUNDEX('mustermann') FROM dual;

SOUN
----
M236
```

Zur Verwendung des Soundex-Operators bei der Volltextsuche werden während der Indexierung Listen für ähnlich klingende Wörter erstellt und diese Informationen innerhalb von Tabellen im Index abgelegt, die dann bei der späteren Suchanfrage benutzt werden können, um die Anfrage auf ähnlich klingende Wörter zu erweitern.

Bei einer **Fuzzy**-Suche werden dem Suchbegriff ähnlich geschriebene Wörter in die Suche einbezogen. Dabei wird berücksichtigt, wie sehr sich die Wörter vom eigentlichen Suchwort unterscheiden und die Abweichung kann mit in die Bewertung für die Trefferliste einbezogen werden, so dass Dokumente mit Begriffen, die dem gesuchten Begriff am ähnlichsten sind, höher bewertet werden. Die Fuzzy-Suche eignet sich z.B. gut, wenn ein Wort häufig falsch geschrieben wird oder viele Buchstabendreher im Dokument vorhanden sind. Anders als beim ”Stemming” hängt die Erweiterung des Suchbegriffs bei der Fuzzy-Suche vom Bestand des Indexes ab. Es werden lediglich Erweiterungen des Begriffs auf diejenigen Wörter berücksichtigt, die sich ebenfalls im Index befinden. Die Anzahl der Treffer (vor allem der weniger relevanten) kann bei Verwendung von Fuzzy-Algorithmen unter Umständen stark in die Höhe getrieben werden.

Neben den oben beschriebenen Abfrageoperatoren gibt es noch eine Reihe weiterer wie z.B. den ”NEAR”-Operator, mit dem sich der Abstand zwischen auftauchenden Suchbegriffen in die Suche einbeziehen lässt. Zur Festlegung der Priorität bei der Ausführung sind die Abfrageoperatoren in verschiedene Klassen unterteilt. Sämtliche Operatoren und deren Verwendung werden in der Oracle eigenen Dokumentation beschrieben.[16]

3.2.3 Zugriffskontrolle

Ziel der Sicherheitsbemühungen ist es, den unauthorisierten Zugang zu Informationen und Dokumenten zu vermeiden. Dabei besteht die Schwierigkeit, dass Oracle Ultra Search den Zugriff auf viele verschiedene Datenvorräte bietet, die alle ihre eigenen unterschiedlichen Sicherheitsmodelle verwenden, um zu entscheiden, ob ein Benutzer Zugriff auf ein bestimmtes Dokument erhält oder nicht. Die sicherheitsrelevanten Informationen all dieser Speicherorte müssen berücksichtigt werden, um nicht berechtigte Zugriffe verhindern zu können.
Das Sicherheitskonzept von Oracle Ultra Search setzt in verschiedenen Bereichen an, bei der:

- Dokumentbeschaffung durch den Crawler
- Datenhaltung
- Datenabfrage

In der aktuellen Version wird das **HTTPS-Protokoll** unterstützt, um den gesicherten Transport der Dokumente durch den Crawler zu ermöglichen.
Die als Index in der Datenbank abgelegten Informationen sowie zugehörige Einstellungen und Parameter der Datenquellen sind nicht öffentlich zugänglich. Der Zugang erfolgt passwortgeschützt, wobei die Passwörter verschlüsselt abgelegt werden. Die Benutzerauthentifizierung wird durch das **Oracle Internet Directory** und **LDAP**[17] durchgeführt. Auf diese Weise wird gewährleistet, dass jeder Benutzer lediglich auf diejenigen Daten zugreifen darf, für die er eine entsprechende Berechtigung besitzt. Für die Administration der Ultra Search Instanzen können die Administratoren verschiedenen Gruppen mit unterschiedlichen Berechtigungen zugeteilt werden, um die Privilegien eines Administrators auf das Notwendigste zu beschränken.
Beim Durchsuchen der indexierten Dokumente besteht die Möglichkeit, jedes Dokument durch eine eigene ”**Access Control List (ACL)**” zu schützen. Diese wird während der Suche ausgewertet, um zu prüfen, ob der Benutzer, der die Suchanfrage gestellt hat, berechtigt ist, auf das Dokument zuzugreifen. Öffentliche Dokumente können von allen

[16] vgl. [ORATR-05], Kapitel 3.
[17] Lightweight Directory Access Protocol - ein Netzwerkprotokoll, das die Abfrage und die Modifikation von Informationen eines Verzeichnisdienstes erlaubt

Nutzern eingesehen werden.
Alternativ lässt sich ein Zugriffsschutz auch durch die Verwendung verschiedener Ultra Search Instanzen und Indexe realisieren. So wird bei der in die Webseiten des Forschungszentrums eingebundenen Volltextsuche ein Index für die interne Suche durch Mitarbeiter und ein zweiter Index für externe Benutzer gepflegt. Bei der Ausführung einer Suchabfrage wird anhand der IP-Adresse des Benutzers zunächst geprüft, welcher Gruppe (intern oder extern) der Benutzer angehört und dementsprechend der für die Suchanfrage zu verwendende Index bestimmt, so dass externe Benutzer nur die für sie bestimmten Informationen abrufen können und keinen Zugriff auf interne Seiten erhalten. Bei der Aktualisierung der beiden Indexe durch den Crawler wird dieser abwechselnd mit einer internen und einer externen IP-Adresse gestartet, um die beiden Indexe getrennt voneinander auf den aktuellen Stand zu bringen. Diese Methode hat den Vorteil, dass die Autoren der Webseiten keine besonderen Vorkehrungen beim Erstellen ihrer Seiten treffen müssen, um interne Informationen vor externen Benutzern verborgen zu halten. Dies kann über den Webserver gesteuert werden, der abhängig von der IP-Adresse, von der die Anfrage kommt (in diesem Fall der des Crawlers), den Zugriff auf bestimmte Seiten gewährt oder verbietet.

3.2.4 Anpassungen und Einbindung in eigene Anwendungen

Oracle Ultra Search bietet verschiedene Möglichkeiten der Anpassung und Integration in eigene Applikationen und verfolgt das Ziel, die gebotene Suchfunktionalität möglichst vielseitig einsetzen zu können. Diese Abstimmung auf die Bedürfnisse des Anwenders wird an vielen Stellen in unterschiedlichen Ebenen umgesetzt.

Durch das sogenannte "**Relevance Boosting**" kann der Administrator die Sortierung der Ergebnisliste beeinflussen, indem er die Bewertungszahl (Scoring) eines Dokumentes bezüglich einer bestimmten Suchanfrage explizit angibt. Auf diese Weise lassen sich Dokumente am unteren Ende der Ergebnisliste, die für den Administrator wichtig erscheinen, nach oben verschieben und umgekehrt. Die Bewertungszahl bezieht sich nur auf die spezifische Suchanfrage, wobei Groß- und Kleinschreibung sowie Leerzeichen im Suchstring exakt übereinstimmen müssen. Ist dies nicht der Fall oder wird das Dokument durch eine andere Suchanfrage ermittelt, so wird die von Oracle Text berechnete Bewertungszahl[18] verwendet. Wird für ein Dokument z.B. ein Relevance Boosting für die Suchanfrage "Ultra Search" vorgenommen, dann kommt diese manuelle Bewertung bei der Suchanfrage "ultrasearch" nicht zum Einsatz.

Zur komfortableren Gestaltung der Suche, können die verschiedenen Dokumente eines Index durch den Administrator in **"Data Groups"** eingeteilt werden. Mit Hilfe dieser

[18] s. auch Kapitel 3.3.5 auf Seite 33.

logischen Einheiten lässt sich z.B. die Suche auf spezielle Themengebiete (etwa Wissenschaftsbereiche wie Physik oder Mathematik) einschränken. Eine Data Group kann eine oder mehrere Datenquellen enthalten und umgekehrt dürfen Datenquellen auch mehreren Data Groups angehören. Beim Formulieren der Suchanfrage kann der Suchanwender dann eine oder mehrere dieser Gruppen auswählen, und so die Suche auf gewisse Bereiche eingrenzen. [19]

Oracle Ultra Search bringt eine funktionsfähige Beispielsuchanwendung mit, die "**Sample Query Application**". Sie ist als Java-Webanwendung implementiert und soll die Fähigkeiten zur Ergebnispräsentation, wie das "Highlighting" von Suchbegriffen, vorstellen und darüber hinaus als Vorlage für eigene Suchanwendungen dienen. Hierzu kann der Benutzer die JavaServer Pages[20] der Beispielanwendung entsprechend anpassen und erweitern und damit Einfluss auf die Gestaltung seiner eigenen Applikation nehmen. Die Änderungen können sich auf das Layout der Suchseiten beschränken oder auch die gesamte Struktur und den Ablauf der Anwendung umfassen, z.B. ob die Suchanfrage lediglich in einfacher Form über einen Suchstring erstellt wird oder der Benutzer die Möglichkeit erhält, die Suche zusätzlich über Metainformationen wie Autor oder Sprache zu präzisieren und welche Abfrageoperatoren dem Anwender zur Verfügung stehen sollen.

Die Gestalt der Trefferliste für die zu einer Suchanfrage gefundenen Dokumente kann entsprechend angepasst werden. Dort lassen sich nach Bedarf verschiedene Attribute wie z.B. Autor oder Größe des Dokuments anzeigen. Zusätzlich ist es möglich, für die einzelnen Dokumente zusätzlich zum jeweiligen Link für die Anzeige des Dokuments kurze Textpassagen, die die hervorgehobenen Suchbegriffe enthalten (Highlighting), auszugeben.
Die Anzeige eines in der Trefferliste aufgeführten Dokuments lässt sich für die Datenquelltypen "Table Source", "File Source" und "User-Defined Source" durch die Verwendung einer "**Display URL**" beeinflussen. Exisitiert zur Darstellung der Daten dieser Quellen bereits eine Webapplikation, so kann das Layout dieser Webanwendung unter bestimmten Voraussetzungen für die Darstellung der Daten in Oracle Ultra Search genutzt werden. So lassen sich z.B. bei einer Table Source die Tabellen strukturiert mit Rahmen und Zellen ausgeben, sofern eine Webapplikation zu Verfügung steht, die diese Darstellung übernimmt.

Oracle Ultra Search stellt mehrere Java-basierte Schnittstellen (APIs[21]) zur Verfügung, die zur Verwendung der Suchfunktionalität in eigenen Anwendungen, z.B. für die Ein-

[19] vgl. Abbildung 3.4 auf Seite 12, dort können die Datengruppen "OAI Kits" und "MixedLang" ausgewählt werden.

[20] JavaServer Pages: eine von Sun Microsystems entwickelte Technologie, die hauptsächlich zur einfachen Erzeugung dynamischer Webinhalte dient und die Einbettung von Java Code erlaubt

[21] Application Programming Interface - Programmierschnittstelle in der Informatik

bindung einer Volltextsuche in die eigenen Webseiten, dienen. Die Schnittstellen sind für unterschiedliche Bereiche der Suche (wie das Crawlen oder die Suchabfrage) implementiert. Auf diese Weise können die verschiedenen Teilbereiche unabhängig voneinander an die eigenen Anwendungen angepasst werden.

Die "**Oracle Ultra Search Query API**" ist für die Abfrage der indexierten Daten zuständig. Die verwendbaren Methoden dieser API erfassen die Resultate der Abfrage, können die Einstellungen der Suche, wie z.B. die maximale Anzahl der angezeigten Treffer, anpassen und stellen die Ergebnisse dar. Die Gestaltung der Ausgabe kann dabei von der Webanwendung übernommen werden. Bei Bedarf können die Resultate jedoch auch in HTML-Code eingebettet zurückgegeben werden. Die Query API kommt ebenfalls in der von Oracle Ultra Search zur Verfügung gestellten Sample Query Application zum Einsatz.

Die "**Crawler Agent API**" ist dazu gedacht, "User-Defined Sources"[22] in die Suche einzubinden. Mit ihr lassen sich eigene Crawler Agents implementieren, mit deren Hilfe dann die Dokumente aus den proprietären Datenquellen geholt werden (vgl. Abbildung 3.8 auf der nächsten Seite). Dazu muss der Agent unter anderem die Authentifizierung für den Zugang zur Datenquelle und den Zugriff auf die Dokumente über eine URL im HTTP-Protokoll zur Verfügung stellen.

Mit Hilfe der "**E-mail API**' ' kann die Darstellung von E-Mails in einer "E-mail Source" eingerichtet werden. So lässt sich die Funktionalität zur Anzeige von E-Mails in eigene JavaServer Pages einbauen. Die vorgefertigte Anwendung zur Darstellung von E-Mails, die ebenfalls aus JavaServer Pages besteht und die E-mail API benutzt, kann direkt abgeändert und angepasst werden. Das Crawlen und Wiedergeben von E-Mails in Oracle Ultra Search baut auf der JavaMail API und deren Umsetzung durch Sun Microsystems[23] auf.

Neben den Java-basierten Schnittstellen steht zur Administration von Ultra Search Instanzen eine "**Administration PL/SQL API** zur Verfügung. Sie stellt PL/SQL-Funktionen[24] für den Umgang mit Instanzen und Ausführungsplänen sowie der Konfiguration des Crawlers bereit.

Neben den oben erwähnten gibt es noch weitere Schnittstellen wie die "**URL Rewriter API**" und die "**Document Service API**".

[22] s. Kapitel 3.2.1 auf Seite 16.

[23] s. auch: http://java.sun.com/products/javamail/

[24] Procedural Language / Structured Query Language - eine proprietäre Datenbank-Programmiersprache der Firma Oracle, die eine prozedurale Erweiterung von SQL darstellt

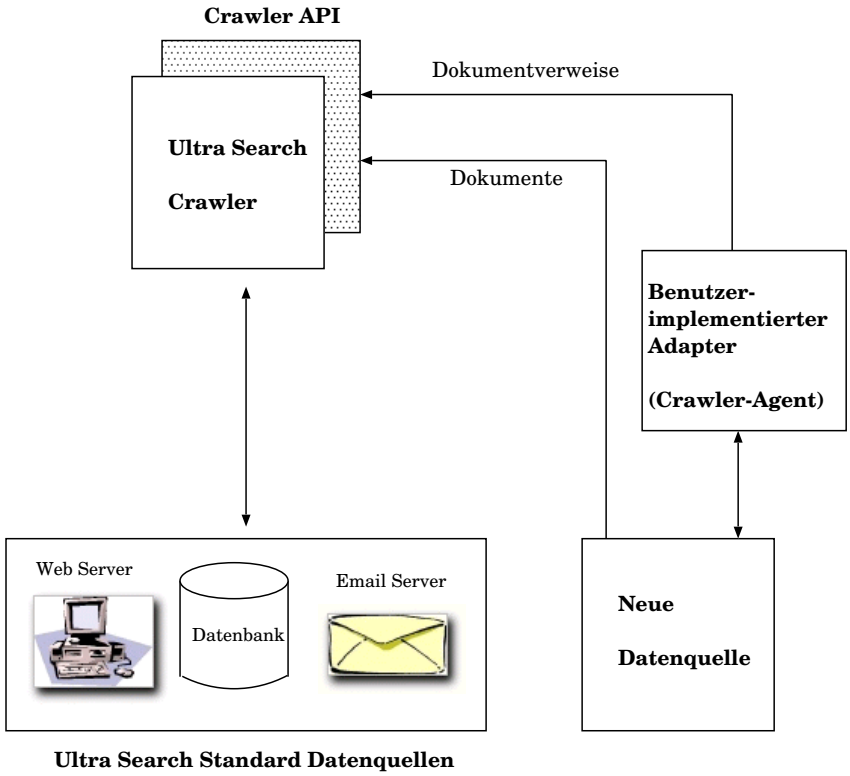

Abbildung 3.8: Prinzip der Ultra Search Crawler API

3.2.5 Optimierung und Statistiken

Um den Index und die Suche zu überwachen, kann sich der Administrator der Suchanwendung verschiedener Statistiken bedienen.
So lassen sich während und nach der Ausführung des Crawlens einer bestimmten Datenquelle Informationen zum Status des Crawlers anzeigen. Dazu gehören der Startzeitpunkt, die benötigte Zeit, die Anzahl indexierter Dokumente, aber auch die Anzahl der Dokumente, bei denen Fehler etwa bei der Konvertierung oder der Indexierung aufgetreten sind. Zudem schreibt der Crawler ein Logfile, dessen Speicherort und Ausführlichkeit der Administrator bestimmen kann. Crawler-Statistiken lassen sich auch für alle Datenquellen gemeinsam anzeigen. Sie enthalten die durchsuchten Hosts, eine Übersicht zur Schachteltiefe indexierter Dokumente und eine Auflistung bezüglich der Dateiformate und Datenquelltypen.
Ebenso wie für den Crawler lassen sich auch für bereits getätigte Abfragen Statistiken anzeigen. Diese enthalten unter anderem Aussagen über die am häufigsten getätigten sowie fehlgeschlagene Abfragen und bieten z.B. Zusammenfassungen über die Abfragetätigkeiten eines Tages.

Entspricht die Perfomance der Suchanwendung nicht den Vorstellungen, kann der Administrator diese an verschiedenen Stellen durch Änderung von Parametern oder spezielle Methoden beeinflussen.
Um die Abfragezeit zu verbessern, bietet sich die Optimierung des Indexes an. Dabei werden veraltete Daten aus dem Index entfernt und die Fragmentierung, die durch die Atkualisierung des Indexes durch den Crawler entsteht, minimiert. Der Index kann auch auf häufig auftretende Wörter und Abfragen hin optimiert werden. Da die Indexoptimierung selbst die Performance stark einschränken kann, sollte sie möglichst in Zeiten geringer Auslastung der Datenbank durchgeführt werden.
Auch die Einstellung verschiedener Datenbankparameter wie z.B. die "DB_CACHE_SIZE", die Einfluss auf die Speicherung von SQL-Anfragen in einem Cache hat, können die Abfragezeiten verbessern.
Die Beschaffung der Dokumente aus den Datenquellen durch den Crawler kann ebenfalls auf unterschiedliche Arten beschleunigt werden. Bei Webquellen auftretende "URL-Looping"-Effekte, die entstehen, wenn viele verschiedene Webseiten auf die gleiche Seite verweisen, können den Crawler verlangsamen. In diesem Fall kann das Verringern der Schachteltiefe des Crawlers Abhilfe schaffen. Zudem kann der gesamte Crawling-Prozess durch die Verwendung eines "**Remote Crawlers**", wobei der Crawler auf ein oder mehrere separate Maschinen ausgelagert wird, beschleunigt werden. Für Table Sources, die sich in einer Oracle Datenbank befinden, stellt Ultra Seach einen "Logging-Mechanismus" bereit, der das Crawlen der Tabellenquellen optimiert und auch für Table Sources eingerichtet werden kann, die sich nicht in einer Oracle Datenbank befinden.

3.2.6 Einsatz im wissenschaftlichen Umfeld

Bei der Volltextsuche in einem wissenschaftlichen Umfeld sind besondere Voraussetzungen und Anforderungen an eine Suchmaschine zu berücksichtigen:
So werden wissenschaftliche Publikationen häufig in **verschiedenen Sprachen** veröffentlicht. Zudem treten auch innerhalb eines wissenschaftlichen Dokuments nicht selten Absätze oder Begriffe in einer anderen Sprache als der im übrigen Dokument verwendeten auf (etwa die gerade im Computerbereich weit verbreiteten Anglizismen). Eine Suchmaschine in einer wissenschaftlichen Umgebung muss folglich den Umgang mit verschiedenensprachigen Dokumenten beherrschen. Die in Oracle Ultra Search verfügbaren Funktionalitäten bezüglich verschiedener Dokumentsprachen[25] erfüllen diese Anforderung.

Mit den zahlreichen verfügbaren Dokumentformaten wie PDF, HTML, Word etc. unterstützt Oracle Ultra Search die gängigsten Formate für Dokumenterstellung und -austausch. Gerade im wissenschaftlichen Bereich treten jedoch häufig **spezielle Dokumentforma-**

[25] s. auch Kapitel 3.2.1 auf Seite 17.

te auf, wie z.B. die für das Textsatzprogramm Latex verwendeten Dokumenttypen ".tex" und ".dvi" oder zum Mathematikwerkzeug Maple gehörende Maple-Worksheets (.mws).

Diese speziellen Formate werden von Ultra Search ebenso wenig unterstützt wie die Volltextsuche innerhalb von Bilddateien (dies betrifft in Grafikformaten abgelegte technische Zeichnungen etc.). Auch Quellcodedateien wie z.B. .java- oder .c-Dateien werden von Oracle Ultra Search bei der Indexerstellung standardmäßig nicht berücksichtigt, was jedoch keine große Einschränkung darstellt, da das Aufsuchen von Quellcodedateien über eine Volltextsuche wenig sinnvoll erscheint. Dagegen fällt die fehlende Unterstützung von Xml-Dateien sowie gängiger Linux-Formate, wie z.B. Sxw-Dateien der OpenOffice.org-Textverarbeitung, schon eher ins Gewicht.

Weiterhin kann es im wissenschaftlichen Bereich erforderlich sein, nicht nur reinen Text zu suchen, sondern auch nach **Formeln**, etwa in den Bereichen Mathematik, Biologie oder Chemie. Oracle Ultra Search beschränkt sich jedoch, wie bei der Volltextsuche üblich, auf Suchanfragen bestehend aus Kombinationen von Wörtern oder Wortteilen und Phrasen. Nach Integralen in einer Gleichung oder Molekülen in einer Chemischen Zusammensetzung kann z.B. nicht gesucht werden. Für diese Anforderungen werden in der Regel spezielle Programme verwendet, die eine komfortable Eingabe von Formeln etc. gewährleisten und sich auch bei der Durchführung der eigentlichen Suche von der gewöhnlichen Volltextsuche unterscheiden. Ein Index aus Worten und Dokumentverweisen reicht für die Realisierung einer derartigen Suche nicht aus.

Die Suche innerhalb von Latex-Dokumenten kann durch die Umwandlung des aus der Quelltext-Datei (.tex) erzeugten DVI-Dokuments (.dvi) in ein PDF-Dokument (dies ist in der Regel problemlos möglich) realisiert werden. Zwar kann auch dann keine Suche nach im Dokument auftauchenden Formeln durchgeführt werden, aber zumindest eine Suche nach enthaltenen Begriffen und Schlagwörtern.

Häufig arbeiten **verschiedene wissenschaftliche Einrichtungen mit unterschiedlichen Standorten** an demselben Projekt. Die Mitarbeiter der verschiedenen Institutionen sollten nach Möglichkeit den Zugriff auf sämtliche zum Projekt gehörige Informationen erhalten, nicht nur auf Informationen des eigenen Instituts.

Diese Aufgabe lässt sich in Oracle Ultra Search recht komfortabel realisieren, da sich verschiedene Datenquellen zu einem Index zusammenfassen lassen. Für den bei der Indexerstellung notwendigen Zugriff auf die Dokumente der entfernten Institutionen, die in der Regel nicht öffentlich über ein Netzwerk zugänglich sind, können dem Oracle Ultra Search Crawler Zugangsdaten mitgeteilt werden, damit diese Dokumente ebenfalls zur Indexierung bereitgestellt werden können. Eine Suchanfrage für den gemeinsamen Index kann dann über eine Weboberfläche von sämtlichen Einrichtungen aus durchgeführt werden.

Mit Hilfe der **Data Groups**[26] ist es möglich, die einzelnen Datenquellen, obwohl lediglich ein gemeinsamer Index besteht, dennoch logisch zu trennen. Im Forschungszentrum Jülich wird auf diese Weise z.B. ein gemeinsamer Ultra Search Index für die gesamten Webseiten und die darüber verfügbaren Dokumente des Forschungzentrums (also aller Institute) gepflegt. [27] Mit Hilfe von Data Groups wird dem Benutzer die Einschränkung der Suche auf die Webseiten eines bestimmten Instituts ermöglicht (s. Abbildung 3.9).

Abbildung 3.9: Über Data Groups realisierte Einschränkung der FZJ-Suche auf ein spezielles Institut

Die im Index aufgenommenen Dokumenttypen bei der Volltextsuche der Forschungszentrums-Webseiten beschränken sich derzeit auf die Formate HTML, Plain-Text und PDF, da diese den Großteil verfügbarer Dokumente abdecken. Für die Datenquellen wird ausschließlich der Typ ”Web Source” verwendet, da die in der Suche zu berücksichtigenden Dokumente über Webseiten erreichbar und somit für den Crawler zugänglich sind.

3.3 Zusammenspiel von Oracle Ultra Search und Oracle Text

Oracle Ultra Search baut auf der Architektur von Oracle Text auf. Oracle Text ist ein Textmanagement- und Suchsystem, das komplett in die Datenbank integriert ist und die notwendigen Funktionalitäten zur Textsuche und Dokumentklassifizierung bereitstellt. Ultra Search verwendet diese Technologie zur Verwaltung und Abfrage des Indexes sowie der Bewertung der gefundenen Dokumente hinsichtlich ihrer Relevanz für die Suchanfrage[28]. Oracle Ultra Search ergänzt die durch Oracle Text bereitgestellten Funktionalitäten mit seinem Crawler[29] um ein wichtiges und mächtiges Werkzeug zur Dokumentbeschaffung sowie um komfortable Oberfläche zur Handhabung und Steuerung der Suche.

[26] vgl. Kapitel 3.2.4 auf Seite 22.

[27] wegen der Unterscheidung zwischen internen und externen Benutzern (s. 3.2.3 auf Seite 22) wird ein zweiter Index parallel gepflegt

[28] s. auch Kapitel 3.3.5 auf Seite 33.

[29] vgl. Kapitel 3.1.1 auf Seite 8.

Gleichzeitig werden jedoch der Optionsumfang und die Suchfunktionalität bei Verwendung von Oracle Ultra Search gegenüber der direkten Verwendung von Oracle Text eingeschränkt. So lassen sich manche Fähigkeiten von Oracle Text in Oracle Ultra Search nicht - oder nur mit sehr großem Aufwand - nutzen. Die von Oracle Text zur Verfügung gestellte Sucharchitektur kann auch durch andere Anwendungen wie Oracle "JDeveloper" oder über SQL und PL/SQL genutzt werden.
Über das Browser-basierte Admin Tool[30] und die vorgefertigte Beispielanwendung ermöglicht Oracle Ultra Search einen einfachen Einstieg in die Volltextsuche, ohne dass der Administrator die Handhabung und Funktionsweise von Oracle Text und die Abläufe der Indexierung *im Detail* kennen muss. Die Einstellungsmöglichkeiten und Defaultparameter von Oracle Ultra Search sind zudem für viele Anwendungen bereits ausreichend, so dass eigene Suchanwendungen ohne großen Aufwand erstellt und administriert werden können.

3.3.1 Indexverwaltung

Beim Erstellen einer Oracle Ultra Search Instanz wird im Hintergrund ein neuer Index mit dem Namen "WK$DOC_PATH_IDX" für den Administrator der Instanz (bei dem es sich um einen Datenbankuser mit bestimmten Privilegien handeln muss) durch Oracle Text angelegt. Ebenso werden die notwendigen Tabellen zur Haltung des Index und der instanzspezifischen Informationen, wie verwendete Datenquellen und Dokumentattribute, automatisch angelegt. Da es sich bei dem von Oracle Ultra Search eingesetzten Index um einen regulären Oracle Text Index handelt, kann dieser ebenfalls alternativ direkt über SQL-Befehle angesprochen und abgefragt werden, ohne die Oberfläche von Oracle Ultra Search benutzen zu müssen. Ein Beispiel für die Abfrage eines mit Oracle Ultra Search erstellen Indexes über Oracle Text (SQL-Anweisung) befindet sich in Anhang A auf Seite 81.
Viele der über das Oracle Ultra Search Admin Tool ausgelösten Aktionen rufen intern eine Oracle Text Funktion auf, über deren Handhabung und Eigenschaften der Ultra Search Administrator jedoch keine Kenntnis besitzen muss.

3.3.2 Indexparameter und Preferences

Nicht alle Eigenschaften des von Oracle Ultra Search verwendeten Indexes lassen sich Oberflächen-basiert über das Oracle Ultra Search Admin Tool steuern. Dem Ultra Search Administrator wird bei der Erstellung einer Ultra Search Instanz zwar die Möglichkeit geboten, den Lexer, die Stoppwortliste und die Art der Datenspeicherung auszuwählen[31],

[30] vgl. Kapitel 3.1.4 auf Seite 13.
[31] s. Abbildung 3.10 auf der nächsten Seite.

jedoch müssen diese, sofern nicht die Standard- einstellungen für die Indexparameter verwendet werden sollen, zunächst explizit durch den Administrator in Form eines Oracle Text ”Preferences” (dies ist ein Objekt zur Voreinstellung von Indexparametern) angelegt werden.

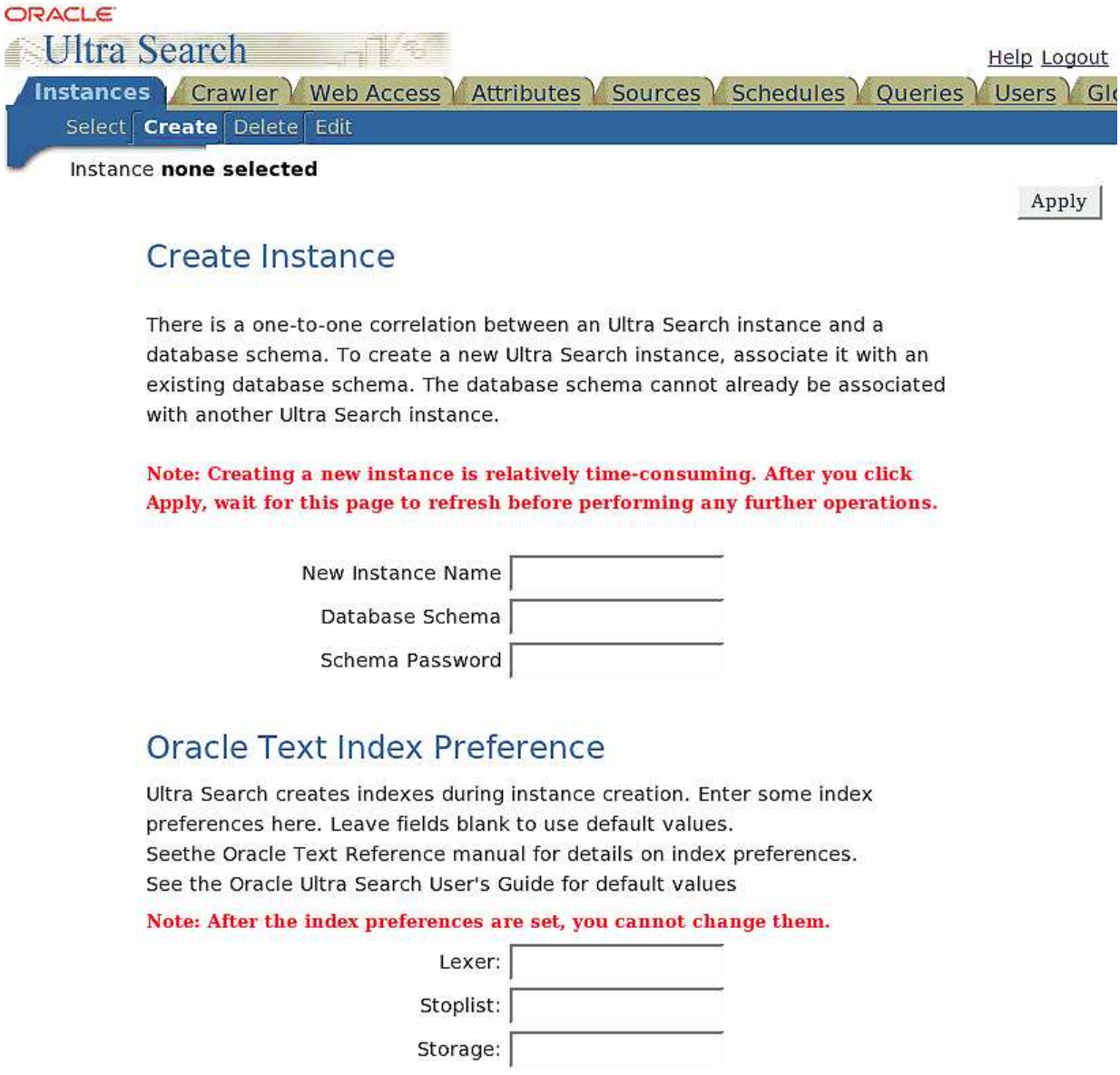

Abbildung 3.10: Wahl der Indexparameter einer neuen Ultra Search Instanz

Dies geht jedoch nicht ohne Kenntnisse der Oracle Text Architektur, da die Parameter und deren Auswirkungen für das Erstellen eines geeigneten ”Preferences” bekannt sein müssen. Soll z.B. nicht die von Ultra Search zur Verfügung gestellte Standard-Stoppwortliste (die lediglich englische Wörter enthält) verwendet werden, muss diese entweder in einem PL/SQL-Skript (”wk0pref.sql”), welches sich im Ultra Search Heimatverzeichnis befindet und beim Anlegen einer neuen Instanz ausgeführt wird, von Hand geändert werden oder mittels Oracle Text eine eigene Stoppwortliste (z.B. mit Hilfe der PL/SQL-Funktionen ”create_stoplist” und ”add_stopword”) erstellt werden, die dann im Ultra Search Admin Tool ausgewählt werden kann.
Ähnlich sieht es bei der Verwendung eigener Lexer und bei der Anpassung der Datenspeicherung aus. Auch hier müssen diese zunächst über Oracle Text als ein ”Preference” erstellt werden, bevor sie in Oracle Ultra Search verwendet werden können.
Die Standard-Einstellungen für den Lexer und die Art der Datenspeicherung sind allerdings sinnvoll gewählt, so dass eine Anpassung für einfache Suchanwendungen nicht notwendig ist. Der verwendete Lexer vom Typ ”Multilexer” beherrscht beispielsweise von Hause aus den Umgang mit verschiedenen Sprachen.

Für neu anzulegende Instanzen können die Defaulteinstellungen des Indexes über das oben erwähnte PL/SQL-Skript angepasst werden, welches beim Erstellen einer neuen Oracle Ultra Search Instanz automatisch aufgerufen wird. Die Parameter eines mit Oracle Ultra Search bereits erstellten Indexes können, wie bei gewöhnlichen Oracle Text Indexen üblich, über SQL und PL/SQL Kommandos wie z.B. den ”ALTER INDEX”-Befehl angepasst werden. Hierzu zählen unter anderem die Einstellungen, die den Einsatz von Stemming- und Fuzzy-Operatoren betreffen. Allerdings ist aus Konsistenzgründen zu beachten, dass die Anpassungen durchgeführt werden sollten, bevor der Index mit Daten gefüllt wird, also vor der ersten Ausführung des Ultra Search Crawlers.

3.3.3 Query Syntax

Die durch den Suchanwender über Oracle Ultra Search eingegebene Suchanfrage wird mit Hilfe der sogenannten ”Query Syntax Expansion” in einen Oracle Text - konformen Ausdruck umgewandelt, der dann für die eigentliche Abfrage des Indexes durch Oracle Text verwendet wird. Die Regeln dieser Syntaxerweiterung sind in der Notation der Backus-Naur-Form[32] festgehalten. Sie wird sowohl für die Suche im Dokumentinhalt als auch für das Durchsuchen der Dokumentattribute benutzt. Die durch Oracle Ultra Search bereitgestellte und als Standard verwendete ”Query Syntax Expansion” beinhaltet lediglich Basisfunktionalitäten wie die Suche nach einzelnen oder mehreren Wörtern sowie Textphrasen und ermöglicht die Verwendung der drei Operatoren ”+”, ”-” und ”*”, mit deren

[32]Kompakte formale Metasyntax (Metasprache), die benutzt wird, um kontextfreie Grammatiken wie z.B. die Syntyx höherer Programiersprachen darzustellen, vgl. [WIKIP].

Hilfe man Wörter oder Phrasen explizit in die Suche einbinden oder von der Suche ausschließen bzw. eine Platzhalter-Suche durchführen kann. Für viele Anwendungen sind diese Suchfunktionalitäten ausreichend.

Die Verwendung von komplexen Abfrageoperatoren wie Fuzzy, Stemming oder Soundex ist in der von Oracle Ultra Search bereitgestellen Basiskonfiguration der ”Query Syntax Expansion” nicht möglich. Die Fähigkeiten von Oracle Text werden also zunächst nicht ausgeschöpft. Die Transformation der Suchanfrage lässt sich jedoch durch den Administrator an die eigene Suchanwendung anpassen und somit auch die Auswahl der Abfrageoperatoren erweitern. Dies geht allerdings nicht Oberflächen-basiert über das Oracle Ultra Search Admin Tool, sondern es müssen explizit die JavaServer Pages der Oracle Ultra Search Beispielanwendung (Sample Query Application) editiert und angepasst werden: Statt der Java-Klasse ”oracle.ultrasearch.query.Contains” muss dort die Klasse ”oracle.ultrasearch.query.CtxContains” verwendet werden.
Ultra Search unterscheidet bei der Eingabe eines Suchstrings zunächst nicht zwischen Groß- und Kleinschreibung. Dies lässt sich jedoch über die Eigenschaften des Lexers ändern[33], indem das Attribut ”mixed_case” gesetzt wird. Bei der Phrasensuche ist darauf zu achten, dass nur ganze Wörter in der Textphrase enthalten sein dürfen. Taucht z.B. in einem Dokument die Phrase ”konstanter Ausdruck” auf, so wird das Dokument über die Suchanfrage ”konstanter Ausdru” NICHT gefunden.

3.3.4 Verwendung von Themensuche und Thesaurus

Wie die komplexen Abfrageoperatoren sind auch die in Oracle Text zur Verfügung stehende Themensuche und die Verwendung eines Thesaurus in Oracle Ultra Search zunächst nicht verfügbar und lassen sich ebenfalls nicht über das Oracle Ultra Search Admin Tool bereitstellen. Um eine Themensuche durchzuführen, muss eine entsprechende ”Knowledge-Base” für die jeweilige Sprache in Oracle Text zur Verfügung stehen, die ähnlich wie ein Thesaurus einzelne Begriffe themenorientiert miteinander in Beziehung setzt. Zudem müssen die Parameter des Indexes entsprechend gesetzt sein, da die Themeninformationen bereits bei der Indexierung durch den Lexer gesammelt werden. Oracle Text stellt Themeninformationen in Form einer Knowledge-Base lediglich für die Sprachen Englisch und Französisch bereit. Soll die Themensuche auch für andere Sprachen verwendet werden können, müssen weitere Knowledge-Bases durch den Administrator über Oracle Text verfügbar gemacht werden. Um diese in Oracle Ultra Search einsetzen zu können, muss zudem die Syntaxerweiterung der Suchanfragen wie oben beschrieben angepasst werden, so dass der ”about”-Operator, der in Oracle Text für die Themenabfrage zuständig ist, auch in Oracle Ultra Search verwendet werden kann.
Die Benutzung eines Thesaurus ist in Oracle Ultra Search ohne den Eingriff in die Oracle

[33] s. auch Kapitel 3.3.2.

Text Architektur und die Anpassung der Ultra Search Query Syntax Expansion zunächst nicht möglich. Oracle Text stellt eine Beispieldatei zur Einbindung eines englischsprachigen Thesaurus bereit, der in Oracle Text eingebunden werden muss (dies geschieht z.B. durch das Oracle Werkzeug "ctx_loader", mit dessen Hilfe der Thesaurus aus einer Datei in die Datenbank geladen werden kann). Für weitere Sprachen muss der Thesaurus vom Administrator selbst erstellt oder z.B. käuflich erworben werden. Damit dieser auch in Oracle Ultra Search durch den Einsatz spezieller Abfrageoperatoren , wie z.B. "SYN" zur Suche nach allen Synonymen eines Begriffs, verfügbar ist, ist hier ebenfalls eine Anpassung der Query Syntax Expansion notwendig.

3.3.5 Ranking und Scoring

Für die Bewertung und Einordnung der Dokumente hinsichtlich ihrer Relevanz (Ranking) nutzt Oracle Ultra Search wie bei der Indexierung und der Durchführung einer Suchabfrage ebenfalls die Oracle Text Funktionalität.

Im Gegensatz zu Internetsuchmaschinen kommt in einem wissenschaftlichen Umfeld wie dem Forschungszentrum Jülich oder bei einer firmeninternen Suche den kommerziellen Aspekten bei der Bewertung von Dokumenten und der Sortierung der Trefferlisten keine allzugroße Bedeutung zu. Hier steht der Austausch und das Auffinden von nüztlichen Informationen im Vordergrund. Ein Wissenschaftler wird wenig daran interessiert sein, seine eigenen wisschenschaftlichen Dokumente möglichst weit oben in der Trefferliste einer internen Suchmaschine zu platzieren. Entscheidender sind die Qualität der Treffer und die Relevanz eines Dokumentes im Bezug auf die Suchanfrage. Die Einordnung der Treffer diesbezüglich spielt eine entscheidende Rolle bei der Informationssuche. Die vermeintlich besten Treffer sollen möglichst weit vorne in der Trefferliste erscheinen. Um die Dokumente hinsichtlich ihrer Relevanz für eine bestimmte Suchanfrage bewerten zu können, verwendet Oracle Text einen Scoring-Algorithmus, der für jedes in Frage kommende Dokument eine Bewertungszahl (Score) errechnet, und nimmt mit Hilfe dieser Bewertungszahl die Sortierung der Trefferliste vor.

Der verwendete Algorithmus basiert auf einem "Inverse Document Frequency"- Algorithmus nach Gerard Salton. [34] Dabei geht man davon aus, dass Terme, die in einer Dokumentensammlung sehr häufig auftauchen, Störterme und daher gering zu bewerten sind. Für eine hohe Bewertung eines Dokuments bezüglich eines Suchterms muss der Term häufig in diesem Dokument, jedoch selten in den übrigen Dokumenten auftauchen.

[34] Gerard Salton (1927 - 1995): Professor an der Universität von Havard; wichtige Aspekte der Informationssuche und der Bewertung von Informationen werden in [SALG-83] beschrieben

Um dies bei der Bewertung zu berücksichtigen, setzt Oracle Text folgende Formel ein:

$$Bewertungszahl = 3 * f * (1 + log(\frac{N}{n}))$$

mit:

- f: Anzahl, wie oft der Term im zu bewertenden Dokument auftaucht (frequency)
- N: Gesamtanzahl der Dokumente
- n: Anzahl der Dokumente, die den Term mindestens einmal enthalten
- es wird der Logarithmus zur Basis 10 verwendet

Um den Umgang mit den auf diese Weise erhaltenen Bewertungszahlen zu erleichtern, werden diese von Oracle Text in ganzzahlige Werte zwischen 0 und 100 umgewandelt. Leicht zu erkennen ist, dass ein Dokument unabhängig von den übrigen Dokumenten in jedem Fall die Bewertungszahl 0 enthält, wenn es den Suchterm nicht enthält (f = 0).
Ist der Suchbegriff jedoch im Dokument enthalten, dann ist die Bewertungszahl sowohl von der Gesamtanzahl der Dokumente (N) als auch von der Anzahl der Dokumente, die ebenfalls den Term enthalten (n), abhängig. Die Bewertungszahl eines Dokumentes im Bezug auf einen Suchterm wird also durch das Hinzufügen weiterer Dokumente, die den Term ebenfalls enthalten, reduziert, da für $N, n \in \mathbb{N}$ gilt:

$$\frac{N+1}{n+1} \leq \frac{N}{n} \Rightarrow log(\frac{N+1}{n+1}) \leq log(\frac{N}{n})$$

Ein Sonderfall tritt auf, wenn z.B. ein Suchterm in allen Dokumenten enthalten ist (N = n). In diesem Fall fällt der Logarithmusterm weg (log(1) = 0) und die Bewertungszahl wird ausschließlich davon bestimmt, wie oft der Term im Dokument erscheint (3 * f).

Betrachtet man einen Suchterm, der nur in einem einzigen Dokument auftaucht (n = 1), lässt sich die Frage stellen, wie oft der Term abhängig von einer vorgegebenen Dokumentanzahl (N) in diesem Dokument auftauchen muss (f), damit das Dokument die Bewertungszahl 100 erhält. Dies lässt sich durch Auflösen obiger Formel nach f beantworten:

$$100 = 3 * f * (1 + log(\frac{N}{1})) \Leftrightarrow f = \frac{100}{3 * (1 + log(N))}$$

Betrachtet man diese Formel als Funktion f über die Dokumentanzahl N, ergibt sich der in Abbildung 3.11 auf der nächsten Seite dargestellte Graph für den Zusammenhang zwischen der für einen Score von 100 benötigten Häufigkeit des Suchterms im Dokument und der Gesamtanzahl an Dokumenten.
In Oracle Text ist es zusätzlich möglich, sich statt der Bewertungszahl die Häufigkeit eines auftauchenden Suchterms anzeigen zu lassen und die Trefferliste nach dieser Häufigkeit

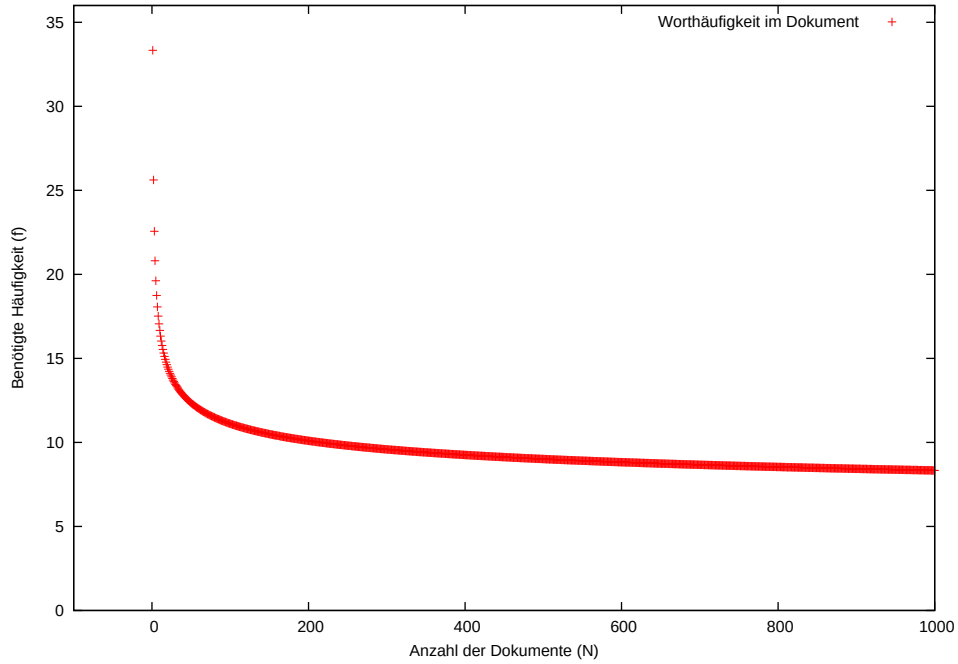

Abbildung 3.11: Benötigte Worthäufigkeit für das Erreichen der Bewertungszahl 100 in Abhängigkeit der Dokumentanzahl

zu sortieren. Eine Beispielabfrage hierzu befindet sich in Anhang A auf Seite 82.

Eine Bewertungszahl für einen konkreten Begriff oder eine Phrase zu berechnen reicht für die Bewertung der Dokumente hinsichtlich ihrer Relevanz für eine Suchanfrage jedoch meist nicht aus, da eine Suchanfrage häufig aus mehreren, durch Operatoren verknüpften Wörtern oder Phrasen besteht.
Deshalb verwendet Oracle Text zudem Bewertungsklassen (Scoring-Classes), um die Einordnung der Dokumente vorzunehmen. Besteht die Suchanfrage z.B. aus den drei Wörtern [Oracle Ultra Search] (diese Suchanfrage wird wie eine Oder-Verknüpfung behandelt), dann werden die Dokumente, die alle drei Wörter beinhalten, in eine höhere Klasse eingestuft, als diejenigen, die nur einen oder zwei der Wörter enthalten. Sind Dokumente darunter, die alle drei Wörter in entsprechender Reihenfolge, also die identische Phrase enthalten, werden diese in die höchste Bewertungsklasse eingestuft. Innerhalb der Bewertungsklassen werden die jeweiligen Dokumente dann weiter sortiert (z.B. wie oben beschrieben nach der Häufigkeit der auftauchenden Begriffe).

Bei größeren Internet-Suchanbietern (wie z.B. ”Google” oder ”Yahoo”) hingegen spielt die Güte der Bewertungsalgorithmen auch im Hinblick auf kommerzielle Aspekte eine

entscheidende Rolle. Denn gerade Wirtschaftsunternehmen besitzen ein großes Interesse, ihre Webseiten möglichst weit oben in den Trefferlisten der Suchmaschinen unterzubringen und somit einem möglichst großen Publikum zugänglich zu machen, um die Bekanntheit zu steigern und potentielle Kunden auf ihre Seiten zu locken.
Bei leicht durchschaubaren Bewertungsalgorithmen könnten Firmen ihre eigenen Internetpräsenzen durch gezielte Anpassungen in der Bewertung nach oben treiben. Die Verfahren sollten daher möglichst komplex - also nicht leicht durchschaubar - sein und bedürfen regelmäßiger Aktualisierungen, um eine Manipulation zu erschweren.

Die Häufigkeit der auftauchenden Suchbegriffe als alleiniges Kriterium ist hierfür nicht ausreichend, da diese Art von Bewertung leicht zu beeinflussen ist. So könnte z.B. durch das Wiederholen von Schlüsselwörtern - gegebenenfalls für den Betrachter der Seite unsichtbar (etwa mit weißer Schrift auf weißem Hintergrund) - die Bewertungszahl erhöht werden.
Die Grundidee für ein weiteres Kriterium besteht in der Überlegung, dass qualitativ hochwertige und wichtige Webseiten öfter durch einen Link von anderen Seiten referenziert werden, als unbedeutende Seiten. Es kann also die Anzahl der Links auf eine bestimmte Webeite als weiteres Kriterium für die Bewertung dieser Seite berücksichtigt werden. Auch diese Bewertungsidee lässt sich mit recht einfachen Mitteln überlisten, z.B. durch das Anlegen einer Vielzahl von Pseudowebseiten, die alle auf die entsprechende Webseite verweisen, damit diese in der Bewertung höher eingestuft wird. Um dies zu vermeiden, kann die Verlinkung einer Seite weiter untersucht werden, nämlich von welchen anderen Seiten aus die Referenzierung erfolgt. Die Links von wichtigen Seiten werden höher bewertet, als die Verlinkung über unwichtige Seiten.

Ein Beispielszenario für die Verwendung dieses Bewertungsansatz befindet sich in Anhang C auf Seite 85.

Kapitel 4

Testumgebung

Im Folgenden werden Leistungsmessungen bezüglich der Dauer einer Oracle Ultra Search Suchanfrage durchgeführt. Hierzu wurden verschiedene Ultra Search Instanzen angelegt, auf denen die jeweiligen Abfragen durchgeführt werden. Jede Instanz enthält genau einen Testindex.
Da ein Vergleich zwischen der Verwendung von Oracle Ultra Search auf einem Oracle Application Server (AS) und einem Datenbanksystem (RDBMS) durchgeführt werden soll, wurden die Testinstanzen und -indexe auf beiden Systemen identisch angelegt. Das heißt, es wurden - sofern der Versionsunterschied der beiden Oracle Ultra Search Installationen dies zuließ - dieselben Indexparameter sowie die gleichen Dokumente zur Indexierung verwendet.[1]

4.1 Dokumentvorrat

Für die Erstellung der Testinstanzen und die darin enthaltenen Indexe wurde ein gemeinsamer Dokumentvorrat angelegt, der sich auf einem separaten Filesystem befindet und durch Mounten für die Crawler der beiden Ultra Search Installationen erreichbar ist. Dieses Vorgehen bei der Bereitstellung des Dokumentvorrates hat gegenüber der Indexierung von Webseiten den Vorteil, dass die Dokumente statisch sind und Verfügbarkeitsprobleme weitgehend auszuschließen sind.
Bei den Dokumenten handelt es sich größtenteils um Kopien von auf dem zentralen Webserver des Forschungszentrums abgelegten Dokumenten sowie Schulungsunterlagen und Dokumentationen von Oracle, MySQL und PHP. [2] Verfasst sind die Dokumente hauptsächlich in deutscher oder englischer Sprache. Weiterhin sind unter den Dokumenten verschiedene Formate zu finden wie z.B. PDF-, Word-, und Powerpoint-Dokumente.

[1] vgl. Kapitel 4.5 auf Seite 43.

[2] PHP Hypertext Preprocessor - weitverbreitete Skriptsprache, die hauptsächlich bei der Web-Entwicklung eingesetzt wird und sich in HTML einbetten lässt.

Die meisten Dokumente sind jedoch im HTML-Format abgelegt. Insgesamt besitzt der Dokumentvorrat ein Datenvolumen von 5,8 GigaByte und enthält ca. 45.000 Dateien, wobei dazu auch Bilder und einige ausführbare Dateien und Skripte zählen.

4.2 Datenbank-Installation von Oracle Ultra Search

Für die Datenbank-Variante von Oracle Ultra Search wurde folgende Plattform verwendet:

Rechnertyp, Architektur	IBM, 7028-6E4, PowerPC_POWER4
Anzahl CPUs	1
CPU Takt	1000 Mhz
Hauptspeicher	5120 MB
Plattenspeicher	274 GB
Betriebssystem	AIX 2 5 (64-bit)
RDBMS-Version	10.2.0.1
Ultra Search Version	10.2.0.1

Bei einer Default-Installation dieser Datenbank-Version wird Oracle Ultra Search automatisch installiert. Jedoch müssen vor der Verwendung in der Regel noch einige Anpassungen vorgenommen werden, wie z.B. das Freischalten von Benutzer-Accounts. Nähere Informationen zur Einrichtung von Oracle Ultra Search befinden sich in [ORAUS-05].

4.3 Application Server-Installation von Oracle Ultra Search

Oracle Ultra Search ist ebenfalls über den Oracle Application Server verfügbar. Bei der Installation des Application Servers muss das Paket ”Portal and Wireless” ausgewählt werden, damit Oracle Ultra Search installiert wird. Wie auf dem Datenbank-System sind auch hier im Anschluss an die Installation noch einige Anpassungen notwendig, um Oracle Ultra Search einzurichten. Diese, sowie einige Besonderheiten bei der Verwendung von Oracle Ultra Search auf einem Oracle Application Server, werden ebenfalls in [ORAUS-05](Kapitel 3) beschrieben.

Rechnertyp, Architektur	GenuineIntel i686
Anzahl CPUs	2
CPU Takt	1400 Mhz
Hauptspeicher	3980 MB
Plattenspeicher	65,3 GB
Betriebssystem	SuSE SLES-8 (i386) 2.4.21 138 (32-bit)
Application Server Version	10g (9.0.4) Release 1
RDBMS-Version[3]	9.0.4
Ultra Search Version	9.0.4

Die im Application Server enthaltene Version von Oracle Ultra Search (9.0.4) enspricht nicht der aktuellsten verfügbaren Version. Die Funktionsweise und die Handhabung sind jedoch größtenteils identisch.

4.4 Anlegen der Testinstanzen

Sämtliche Testinstanzen (insgesamt 14) wurden mit Hilfe des Oracle Admin Tools erstellt. Die Anzahl der Threads für den jeweiligen Crawler der einzelnen Instanzen wurde auf 15 gesetzt. Zudem wurde - außer bei den Instanzen 11 und 12 - die automatische Erkennung der Dokumentsprache eingeschaltet. Die folgende Tabelle gibt einen Überblick über die zur jeweiligen Instanz gehörenden Indexgrößen sowie die zur Crawlerausführung benötigten Zeiten. Diese Zeiten beinhalten sowohl die Dokumentbeschaffung durch den Crawler als auch die Zeiten für die interne Indexierung der Dokumente, da diese Prozesse bei Oracle Ultra Search in einem Schritt ausgeführt werden.

Instanz Id	**Anzahl Dokumente im Index**	**Speicherplatz des Indexes in MB (RDBMS)**	**Speicherplatz des Indexes in MB (AS)**	**Crawler-Zeit RDBMS (std:min:sek)**	**Crawler-Zeit[4] AS (std:min:sek)**
1	10327	44,72	43,47	00:28:18	00:13:52
2	11020	112,34	80,84	00:39:33	00:39:41
3	12395	118,47	92,86	00:44:20	00:39:42
4	12950	120,91	97,33	00:48:33	00:48:08
5	13647	130,91	104,47	01:02:35	01:08:14
6	14308	140,97	111,36	01:07:09	01:40:06
7	15069	281,28	121,86	01:49:40	01:10:15
8	25743	289,03	233,61	02:05:15	00:24:46
9	27584	313,09	245,03	02:49:46	00:20:17
10	27847	349,16	273,52	02:26:35	00:04:13
11	5064	19,22	16,28	00:12:49	00:03:22
12	5200	20,84	15,32	00:13:29	00:03:04
13	10327	44,66	40,52	00:28:32	00:09:33
14	27847	1615,28	267,27	04:13:42	03:16:12

Tabelle 4.1: für die Crawlerausführung (Indexerstellung) benötigte Zeiten für die jeweiligen Testinstanzen

[3] Auch der Application Server verwendet zur internen Speicherung ein RDBMS-System als sogenanntes Repository

Die durchschnittliche Dokumentgröße innerhalb der einzelnen Indexe schwankt zwischen 6 und 76 KiloByte. Es befinden sich also - für Webseiten üblich - überwiegend kleine Dokumente und Dateien in den Indexen. Obwohl bei den Instanzen 10 und 14 sämtliche Unterverzeichnisse des Dokumentvorrats bei der Erstellung des Indexes berücksichtigt wurden, enthalten die Indexe dieser beiden Instanzen mit 27.847 Dokumenten nur etwas mehr als die Hälfte der sich im gesamten Dokumentvorrat befindenden Dateien (ca. 45.000). Die Ursache dafür liegt in der Einschränkung der Dokumentformate. Bei sämtlichen Instanzen wurden die Formate der zu indexierenden Dokumente auf folgende Dokumenttypen begrenzt:

- HTML
- Microsoft PowerPoint Document
- Microsoft Word Document
- PDF Document
- Plain Text

Dadurch werden sämtliche Bilder, ausführbare Dateien und sonstige durch Oracle Ultra Search nicht indexierbare Dokumenttypen vom Crawler- und Indexierungsprozess ausgeschlossen.

Stellt man die für die Ausführung des Crawlers benötigten Zeiten in Abhängigkeit der Anzahl der Dokumente, die in den jeweiligen Index aufgenommen wurden, dar, ergibt sich das Diagramm aus Abbildung 4.1 auf der nächsten Seite. Aufgrund der fehlenden Gesamtzeiten für die Instanzen 8, 9 und 10 auf dem Application Server, wurden hierbei lediglich die Crawler-Zeiten für das RDBMS-System betrachtet. Da die Indexe 1 und 13 sowie 10 und 14 dieselben Dokumente beinhalten, ergaben sich für die Anzahlen 10.327 und 27.847 jeweils zwei Zeiten, die Zeiten für Index 1 und 13 (10.327 Dokumente) sind jedoch identisch.

Obwohl die Indexe der Instanzen 10 und 14 die gleiche Anzahl von Dokumenten enthalten, dauerte die Crawler-Ausführung von Instanz 14 auf dem RDBMS mit über 4 Stunden beinahe doppelt solange wie bei Instanz 10 (knapp 2,5 Stunden). Um schwankende Auslastung des Systems bzw. Netzwerkeinflüsse als Ursache hierfür auszuschließen, wurde der Crawler von Instanz 14 ein zweites Mal ausgeführt. Jedoch nahm auch die zweite Ausführung ähnlich viel Zeit (4 Stunden und 3 Minuten) in Anspruch, wie die erste. Die

[4]Da die Crawler-Ausführung bei den Instanzen 8,9 und 10 zwischenzeitlich unterbrochen und später fortgeführt wurde, handelt es sich bei den von Oracle Ultra Search für diese Instanzen angegebenen und hier aufgeführten Zeiten nicht um die Gesamtzeit für die Crawlerausführung.

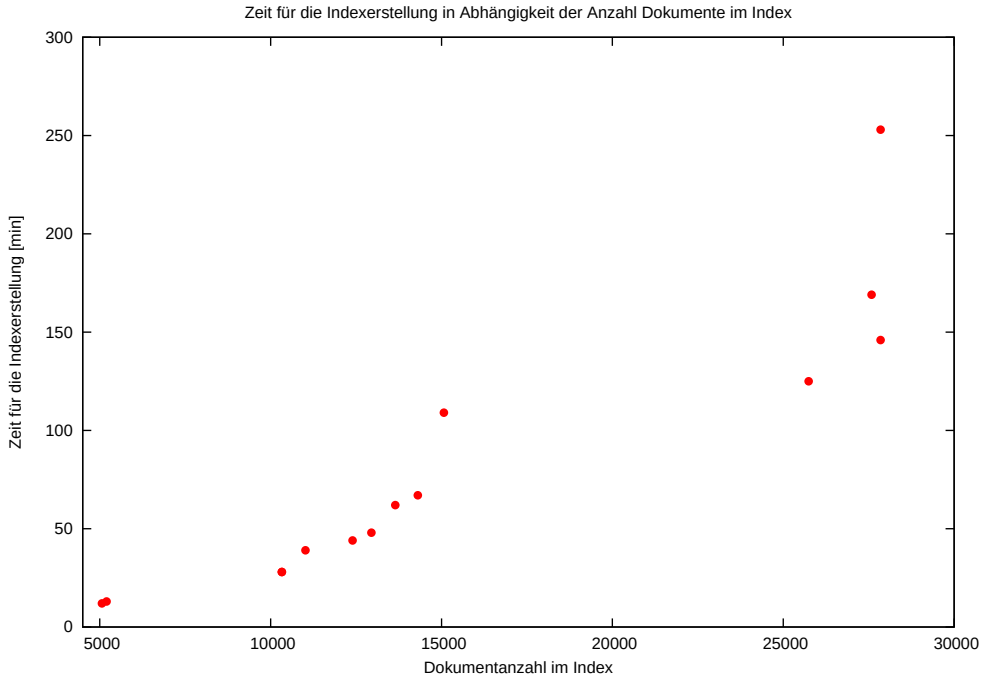

Abbildung 4.1: Zeiten für die Crawlerausführung der Testinstanzen auf dem RDBMS

Ursache für die lange Crawlerzeit scheint die Einstellung der Index-Parameter zu sein (s. hierzu Kapitel 4.5 auf Seite 45).
Es ist ein deutlicher Zusammenhang zwischen der benötigten Zeit für die Ausführung des Crawlers und der Anzahl der in den Index aufgenommenen Dokumente zu erkennen.

Die Zeiten für die Ausführung des Crawlers und die Indexerstellung sollen jedoch aus folgenden Gründen nicht näher untersucht werden:

- In der Regel werden nur bei der ersten Crawlerausführung zur Indexerstellung sämtliche Dokumente berücksichtigt. Die erste Crawlerausführung zur Erstellung eines Indexes dauert bei Dokumentmengen wie den hier verwendeten unter Umständen mehrere Stunden. Häufige Wiederholungen dieses Vorgangs, die für Zeitmessungen und begründete Schlussfolgerungen notwendig wären, sind aus Zeitgründen deshalb nicht möglich. Bei weiteren Crawler-Durchläufen werden nur noch geänderte und neue Dokumente verarbeitet, weshalb die Zeiten für weitere Crawler-Durchläufe zur Aktualisierung eines bereits vorhandenen Indexes wesentlich geringer sind als die für den ersten Durchlauf.
- Die Zeiten für die Aktualisierungen eines Indexes hängen im Wesentlichen von der Änderungsrate der Dokumente ab und können sehr unterschiedlich sein. Bei der

Suche innerhalb der Webseite des Forschungszentrums beispielsweise reicht eine wöchentliche Aktualisierung der Indexe aus, da die Änderungsrate der Dokumente nicht sonderlich hoch ist. In anderen Suchumgebungen kann sich dies jedoch anders verhalten. Daher macht eine Untersuchung der Zeiten für die Indexaktualisierung wenig Sinn, zumal der Dokumentvorrat der Testumgebung statisch ist und somit eine Aktualisierung überflüssig wäre.

- Sowohl die Indexerstellung als auch die Indexaktualisierung geschehen in der Regel vor dem Endbenutzer verborgen und können z.B. nachts durchgeführt werden, wenn sich wenige Benutzer auf dem System befinden, so dass sie keinen für den Benutzer spürbaren Einfluss auf die Dauer einer Suchanfrage haben

- Die Crawler-Zeiten werden von vielen Faktoren beeinflusst, die der Ultra Search Administrator zudem zum Teil nicht kontrollieren kann. So spielt die Netzwerkverbindung zwischen dem Host des Crawlers und den Speicherorten der zu beschaffenden Dokumente eine große Rolle: werden etwa Dokumente von einem weit entfernten Webserver geholt, dauert dies länger als die Beschaffung von Dokumenten, die sich bereits auf der Festplatte des Hosts oder im lokalen Netzwerk befinden

Für jede Testinstanz wurde ein eigener Datenbankbenutzer angelegt, wodurch eine gegenseitige Beeinflussung der verschiedenen Indexe bei den Zeitmessungen vermieden werden kann. Da der Dokumentvorrat statisch ist, also keine Änderungen an den Dokumenten vorgenommen werden, musste der Crawler- und Indexierungsprozess nur einmal durchgeführt werden. Daher sind Fragementierungen der Indexe, die bei der wiederholten Aktualisierung entstehen und sich bei der Zeitmessung einer Suchabfrage auswirken könnten, auszuschließen.

Die folgende Abbildung veranschaulicht die Datenbankauslastung des RDBMS während einer Crawler-Ausführung. Es ist ein deutlicher Anstieg der Datenbankaktivitäten zu erkennen:

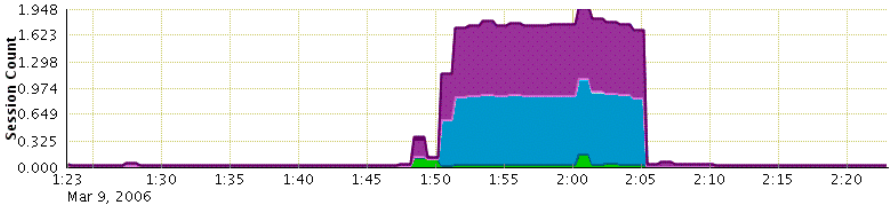

Abbildung 4.2: Datenbankauslastung während der Crawlerausführung

4.5 Anpassungen der Indexeigenschaften für die jeweilige Zeitmessung

Für einige der Zeitmessungen sind Anpassungen der Indexparameter notwendig, damit der verwendete Testindex die gewünschten Eigenschaften besitzt. Die Eigenschaften der jeweiligen Indexe werden im Folgenden beschrieben:

Index 1 ... Index 10:

- **Zweck:** Anhand dieser Testindexe soll geprüft werden, ob ein Zusammenhang zwischen der Dauer der Ausführung einer Suchanfrage und der Größe des zu durchsuchenden Indexes erkennbar ist.
- **Eigenschaften der Dokumente:** Die Dokumentsprache ist gemischt, das heißt, es wurden sowohl deutsche als auch englische Dokumente indexiert.
- **Parameter:** Bei der Indexerstellung wurden keine besonderen Einstellungen bezüglich Crawler und Indexparameter vorgenommen, sondern die in Oracle Ultra Search als Standard eingestellten Parameter verwendet.

Index 11:

- **Zweck:** Dieser Index wird zur Überprüfung der Auswirkung von Spracheigenschaften verwendet.
- **Eigenschaften der Dokumente:** Es handelt sich ausschließlich um deutschsprachige Dokumente.
- **Parameter:** Die Indexparameter wurden wie folgt angepasst:
  - **Lexer:** Statt des als Standard verwendeten Lexers vom Typ ” Multilexer” wurde ein eigener Lexer mit folgenden Eigenschaften eingrichtet:
    - composite → german
      Diese Option ermöglicht die Suche innerhalb zusammengesetzter Wörter. So werden z.B. bei der Suche nach dem Wort ”Bahnhof” auch Dokumente mit den Wörtern ”Hauptbahnhof” oder ”Westbahnhof” gefunden.
    - alternate_spelling → german
      Mit dieser Einstellung werden Umlaute und alternative Schreibweisen berücksichtigt, z.B. liefert die Suche nach dem Wort ”übung” auch Dokumente, die das Wort ”uebung” enthalten

    - new_german_spelling → yes
      Die neue deutsche Rechtschreibung wird berücksichtigt. Das heißt, die Wörter können nach alter und neuer Rechtschreibung gesucht werden. (Wegen der fortlaufenden Änderungen der deutschen Rechtschreibung in den letzten Jahren ist jedoch davon auszugehen,dass hierbei nicht die aktuellste Version der deutschen Rechtschreibung verwendet wird.) Diese Option ist erst in neueren Versionen verfügbar und wurde daher lediglich bei der Oracle Ultra Search Installation im Datenbanksystem verwendet.
  - **Wordlist:** Hier wurden Einstellungen für die Fuzzy-Suche und die Suche nach Wortstämmen vorgenommen:
    - fuzz_match → german
      Hierdurch wird die Methode der Worterweiterung auf andere Wörter festgelegt.
    - stemmer → german
      Diese Option betrifft das Vorgehen bei der Reduktion der Wörter auf ihre Wortstämme, welches ebenfalls sprachabhängig ist.
  - **Stoplist:** Da nur deutschsprachige Dokumente in den Index aufgenommen wurden, wurde eine Stoppwortliste mit deutschen Wörtern verwendet.

Index 12:

- **Zweck:** Dieser Index wird ebenfalls zur Überprüfung der Auswirkung von Spracheigenschaften verwendet.
- **Eigenschaften der Dokumente:** In diesem Index befinden sich ausschließlich Dokumente in englischer Sprache.
- **Parameter:** Die Indexparameter wurden wie folgt angepasst:
  - **Lexer:** die Optionen composite und alternate_spelling sind für die englische Sprache nicht verfügbar und wurden abgeschaltet.
  - **Wordlist:** Die Einstellungen für die Fuzzy-Suche und die Suche nach Wortstämmen wurden auf die englsiche Sprache ausgerichtet:
    - fuzz_match → english
      (Auswirkung der Option s. Index 11)
    - stemmer → english
      (Auswirkung der Option s. Index 11)
  - **Stoplist:** Es wurde die Standard-Stoppwortliste verwendet, die lediglich aus englischen Wörtern besteht.

Nicht alle Änderungen der Parameter für die Indexe 11 und 12 sind notwendig für das Funktionieren der Suche. So ist der Stem-Operator auch mit der Default-Einstellung (stemmer = englisch) bei deutschen Dokumenten verfügbar. Zudem bietet Oracle Text in neueren Versionen mit dem Lexertyp ”Worldlexer” einen Lexer an, der die Dokumentsprache automatisch über eine Unicode-Zeichenerkennung feststellt. Allerdings ist eine manuelle explizite Einstellung sprachspezifischer Parameter durch den Administrator in der Regel vorzuziehen, um möglichst gute Suchergebnisse zu erzielen. Automatisch eingestellte Parameter können unter Umständen schlecht gewählt sein und so zu Qualitätseinbußen bei der Suche führen. Die Fähigkeiten der zur Verfügungen stehenden Algorithmen und Funktionen werden in diesem Fall häufig nicht ausgeschöpft.
Eine Auswirkung auf die benötigte Zeit zur Indexerstellungen bei den Instanzen 11 und 12 konnte nicht festgestellt werden, was jedoch auch an der sehr geringen Datenmenge von ca. 5.000 indexierten Dokumenten liegen kann.

Index 13:

- **Zweck:** Mit Hilfe diese Indexes soll die zeitliche Auswirkung der Trefferlistendarstellung überprüft werden.
- **Eigenschaften der Dokumente:** Es befinden sich deutsch- und englischsprachige Dokumente im Index.
- **Parameter:** Es wurde die Default-Parameter verwendet. Allerdings dürfen zur Verwendung einer anderen Trefferdarstellung die vom Crawler erzeugten ”Cache-Files” nach der Indexierung nicht gelöscht werden. Daher wurde bei Instanz 13 die Crawler-Option ”Delete Cache after Indexing” auf ”false” gesetzt.

Index 14:

- **Zweck:** Dieser Index dient zur Leistungsmessung bei der Platzhaltersuche.
- **Eigenschaften der Dokumente:** Der Index besteht aus deutsch- und englischsprachigen Dokumenten.
- **Parameter:** Anstelle der Standard-Wordlist wurde eine eigene Wordlist verwendet und das Attribut ”prefix_index” auf ”true” gesetzt, was dazu führt, dass auch Teilwörter in den Index aufgenommen werden und deshalb die Suche mit Platzhaltern beschleunigt werden soll. Taucht z.B. in einem Dokument das Wort ”Haus” auf, wird bei gewöhnlicher Indexierung folgender Eintrag in die Indextabelle geschrieben:

Token	Type	Information
HAUS	0	DOCID 1 POS 1

Wird die Option prefix_index verwendet, so werden zusätzliche Einträge für einzelne Wortteile vorgenommen. Dabei kann die minimale Länge der Wortteile eingestellt werden (vgl. [ORATR-05] Kapitel 2 Seite 57). Hier wurde der Standartwert '1' verwendet, das heißt, es werden auch einzelne Buchstaben in den Index aufgebnommen:

Token	Type	Information
HAUS	0	DOCID 1 POS 1
H	6	DOCID 1 POS 1
HA	6	DOCID 1 POS 1
HAU	6	DOCID 1 POS 1

Allerdings nahmen auf Grund der Erweiterung der Indexierung die Zeit für die Erstellung des Indexes sowie auf dem Datenbanksystem auch dessen Speicherplatzbedarf im Vergleich zu Index 10 trotz gleicher Dokumentanzahl stark zu (1.615 MB für Index 14 gegenüber 349 MB für Index 10).

Die Anpassungen der Indexe 11, 12 und 14 wurden über die von Oracle Text bereitgestellten PL/SQL-Pakete und -Funktionen durchgeführt (ctx_ddl.create_preference, ctx_ddl.set_-Attribute, ctx_ddl.create_stoplist).[5]

4.6 Einrichtung der Zeitmessung

Bei den folgenden Leistungsmessungen sollen die Zeiten für die Dauer der Duchführung einer Suchanfrage gemessen werden. Da bei der Durchführung der Suche mehrere Ebenen durchlaufen werden (vgl. Abbildung 4.3 auf der nächsten Seite), gibt es prinzipiell verschiedene Ansätze für die Zeitmessung.
Die Zeiten könnten z.B. Client-seitig gemessen werden. Das heißt, es würde die Zeit ermittelt, die nach dem Abschicken der Suchanfrage aus dem Browser des Endbenutzers vergeht, bis die Trefferliste zu dieser Suchanfrage im Browserfenster dargestellt wird. Dabei besteht jedoch der große Nachteil, dass die Übertragung zum Browser und damit die Netzwerkgeschwindigkeit und die Eigenschaften des Clients (Browsers) die Messzeiten beeinflussen und verzerren. Die Zeiten für die eigentliche Durchführung der Suchabfrage können daher wegen der schwankenden Übertragungszeiten der Ergebnisse nicht korrekt ermittelt und bewertet werden.

Um die Übertragungszeiten und Netzwerkeinflüsse von der Messung auszuschließen, könnte die Zeitmessung z.B. in die JavaServer Pages von Oracle Ultra Search eingebaut werden. Ebenfalls möglich wäre eine Zeitmessung der intern durchgeführten Oracle

[5] Die Einstellung von Indexparametern wird in [ORATR-05] ausführlich beschrieben.

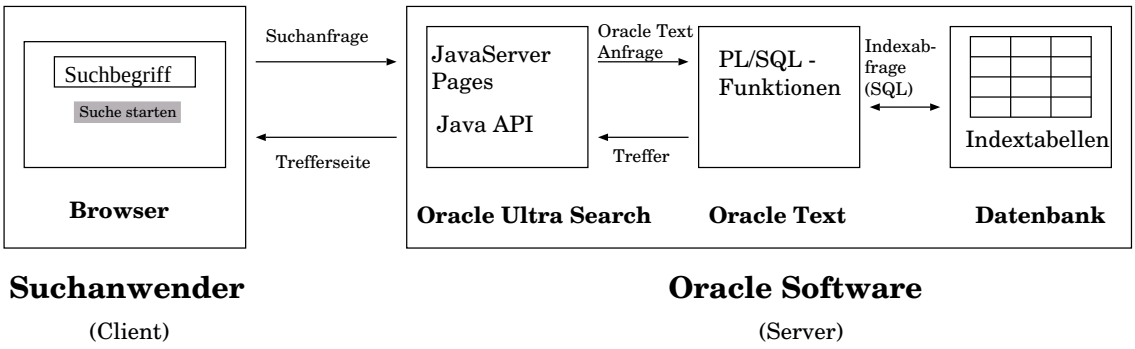

Abbildung 4.3: Bearbeitung einer Suchanfrage in Oracle Ultra Search

Text Suchabfrage bzw. der dazu notwendigen Datenbank-Abfragen der Index-Tabellen (als SQL-Statements). Für die Zeitmessungen im Rahmen dieser Arbeit wurde aus folgenden Gründen die Implementierung innerhalb der JavaServer Pages gewählt:

- Die gesamte Architektur von Oracle Ultra Search ist auf die Verwendung von JavaServer Pages und der zur Verfügung stehenden Java-Schnittstellen ausgerichtet. So basieren sowohl das Ultra Search Admin Tool als auch die Beispiel-Suchanwendung (Sample Query Application) auf dieser Technik.
- Eine Überprüfung der Zeiten für eine Oracle Text-Abfrage bzw. der intern durchgeführten Datenbank-Abfragen würde zwar einen Aufschluss über die Leistungsfähigkeit der durch Oracle Text zur Verfügung gestellten Suchfunktionalität geben, aber nicht die Performance von Oracle Ultra Search wiederspiegeln.
- Eine Messung der internen SQL-Abfragen könnte eventuell vorhandene Zeitunterschiede, die auf den Architekturen der beiden Installationsvarianten (Datenbanksystem und Application Server) beruhen, nur schlecht wiedergeben, sondern würde lediglich die Leistungsfähigkeit der verwendeten Datenbankversionen aufzeigen.
- Da die JavaServer Pages serverseitig ausgeführt werden, geht die Übertrangungszeit der Ergebnisse zum Client (Browser), die auf Grund von Netzwerkeinflüssen zur Verfälschung der Ergebnisse führen könnte, nicht in die Messung mit ein. Dennoch geben die auf diese Weise gemessenen Zeiten den besten Aufschluss über die vom Oracle Ultra Search Suchanwender zu erwartende Dauer der Abfrage.

Die Zeitmessung wurde in die von Oracle Ultra Search bereitgestellte Beispiel - Suchanwendung eingebaut. Diese liegt in zwei Varianten vor: zum einen in einfacher Form als eine JavaServer Page namens ”usearch.jsp” und zum anderen in modularisierter Form bestehend aus mehreren JavaServer Pages. Es wurde aus Gründen der Übersichtlichkeit die einfache Form gewählt.

Die entsprechende JavaServer Page ”usearch.jsp” enthält breits zwei auskommentierte Anweisungen zur Performance-Messung, mit deren Hilfe einmal am Anfang und einmal am Ende der Seite die Systemzeit in Millisekunden gemessen wird. Als Startpunkt wird dabei die Übergabe der Suchanfrage beim Aufruf der Seite gewählt. Die zweite Messung erfolgt nach der Erzeugung der Trefferliste.

Neben der Entfernung der Kommentierung dieser Anweisungen wurde eine zusätzliche Ausgabe der Dauer für die Suche (als Differenz aus diesen beiden Zeiten) in Sekunden sowie der Trefferanzahl vorgenommen (vgl. Abbildung 4.4). Die Anzahl der angezeigten Treffer pro Seite beträgt 10. Werden mehr Treffer gefunden, kann durch Links zu den einzelnen Trefferseiten verzweigt werden. Bei den einzelnen Abfragen wird jeweils die Zeit für die Erzeugung der ersten Trefferlistenseite (Treffer 1 - 10) gemessen.

Abbildung 4.4: Für die Zeitmessung angepasste Ausgabe der Trefferliste

Da viele verschiedene Suchanfragen für die Messungen erforderlich sind und zudem auch Wiederholungen derselben Anfragen zur Berechnung von Durchschnittswerten durchge-

führt werden sollen, wurden PHP-Skripte erstellt, die den Aufruf der JavaServer Page realisieren und die ausgegebenen Ergebnisse (Tefferanzahl und Abfragezeit) auswerten. Dies hat neben der Automatisierung der Durchführung und Auswertung der Messungen den Vorteil, dass Einflüsse durch Browser-Caching (Zwischenspeichern von aufgerufenen Seiten) ausgeschlossen werden können.
Die folgende Abbildung skizziert den Aufbau der Testumgebung bestehend aus einem Dokumentvorrat und den beiden Oracle Ultra Search - Installationen und deren Instanzen sowie die Ausführung der Suchabfragen durch ein PHP-Skript:

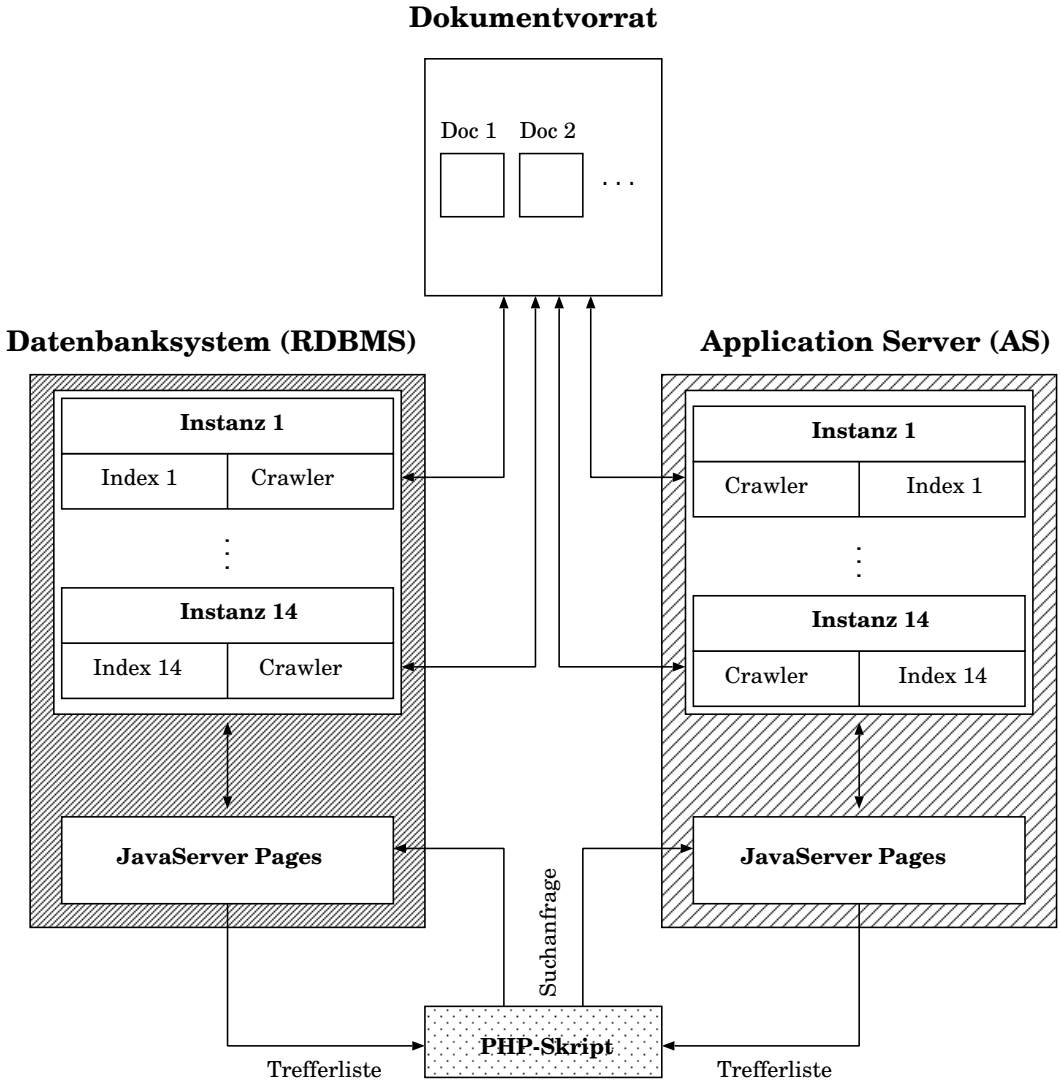

Abbildung 4.5: Aufbau der Testumgebung

Kapitel 5

Leistungsmessung

Die Qualität einer Volltextsuche wird neben dem Funktionsumfang (wie z.B. dem Suchanwender zur Verfügung stehende Abfrageoperatoren und Umgang mit verschiedenen Dokumentformaten etc.), der in den vorangegangenen Kapiteln ausführlich beschrieben wurde, im Wesentlichen von der Dauer für die Verarbeitung einer Suchanfrage bestimmt. Eine Volltextsuche, bei der die Suchanwender minutenlang auf Ergebnisse warten müssen, ist wenig praxistauglich.
Im Folgenden wird die Leistungsfähigkeit von Oracle Ultra Search daher bezüglich der Dauer für die Durchführung verschiedener Suchanfragen untersucht. Dabei wird geprüft, wie stabil die Abfragezeiten sind und welche Faktoren einen Einfluss auf die Dauer der Abfrage haben.

5.1 Stabilität der Abfragezeiten

Zunächst muss geprüft werden, wie stark die Zeiten für die Durchführung einer Suche schwanken. Dies ist unerlässlich, um Fehlschlüsse bei den weiteren Messungen im Hinblick auf Zusammenhänge zwischen den Abfragezeiten und einzelnen Faktoren zu vermeiden. Weiterhin ist von Interesse, ob z.B. eine kontinuierliche Verkürzung der Abfragezeiten bei Wiederholung derselben Abfrage durch internes Zwischenspeichern der Abfrage und der zugehörigen Ergebnisse zu erkennen ist.
Hierzu wird die Suche nach dem Begriff ”Funktionsparameter” auf der Testinstanz 10 der beiden Testsysteme mehrfach durchgeführt: mit 10, 20, 50, 100, 200, 300, 400 und 500 Wiederholungen. Dabei werden jeweils die minimalen, maximalen und durchschnittlichen Abfragezeiten ermittelt. Die folgende Abbildung stellt die Abfragezeiten bei 500 Wiederholungen (ausgeführt auf dem Application Server) dar:

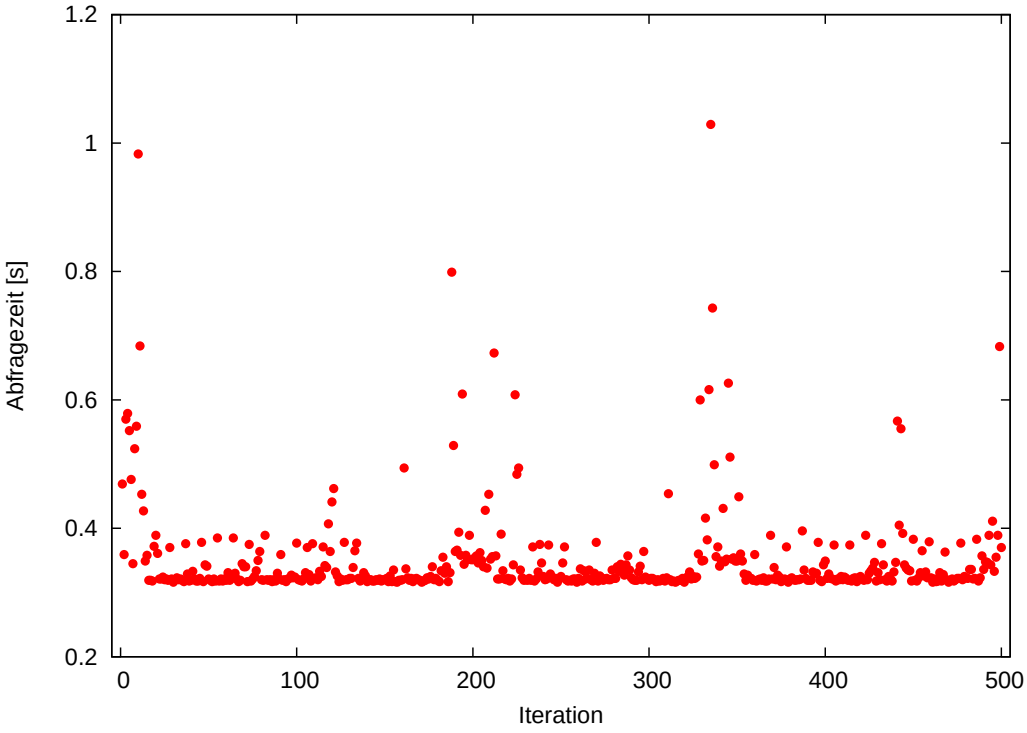

Abbildung 5.1: Abfragezeiten bei **500 Wiederholungen** der Suche nach dem Begriff **”Funktionsparamter”** auf dem **Application Server**

Wie zu erkennen ist, streuen die Abfragezeiten vereinzelt stark nach oben. Eine fortlaufende Verkürzung der Abfragezeit durch internes Zwischenspeichern ist nicht zu erkennen. Mittelwert, Median, Varianz und Standardabweichung aus den 500 Wiederholungen der obigen Abfrage betragen:

$$\overline{x} \approx 0,35 \text{ Sekunden}$$
$$\tilde{x} \approx 0,323 \text{ Sekunden}$$
$$s^2 \approx 0,025 \text{ Sekunden}^2$$
$$s \approx 0,162 \text{ Sekunden}$$

Obwohl die meisten der Zeiten sich weniger als 0.1 Sekunden unterscheiden (ein Großteil der Zeiten befindet sich im Intervall [0.3, 0.4]), gibt es dennoch einige Ausreißer nach oben. Von diesen unregelmäßig auftretenden Spitzen wird wegen deren Größe auch die Durchschnittszeit (in diesem Fall 0.35 Sekunden) stark beeinflusst. Die Standardabweichung beträgt mit 0,162 Sekunden ca. 50 % des Mittelwertes.
Das heißt, die durchschnittliche Zeit für die Durchführung einer Abfrage ist nur wenig aussagekräftig, da die meisten der Werte von der Durchschnittszeit relativ weit entfernt sind.

Dieser Zusammenhang gilt ebenfalls für die Datenbankinstallation von Oracle Ultra Search. Hier traten sogar *bei gleicher Wiederholungsanzahl* für *dieselbe Suchanfrage* stark unterschiedliche Durchschnittszeiten auf:
So wurden unter anderem zweimal 100 Wiederholungen der obigen Abfrage mit dem Suchbegriff ”Funktionsparameter” durchgeführt. Für die ersten 100 Durchläufe ergab sich eine Durchschnittszeit von 0,358 Sekunden. Für die zweiten 100 Durchläufe betrug die durchschnittliche Abfragezeit jedoch nur 0,113 Sekunden. Der erste Durchschnittswert ist somit mehr als dreimal so groß wie der zweite. Die folgende Abbildung zeigt die durchschnittlichen Abfragezeiten in Abhängigkeit der Anzahl durchgeführter Wiederholungen auf dem Datenbanksystem (für die Anzahlen 100 und 200 wurden jeweils zwei Messungen durchgeführt):

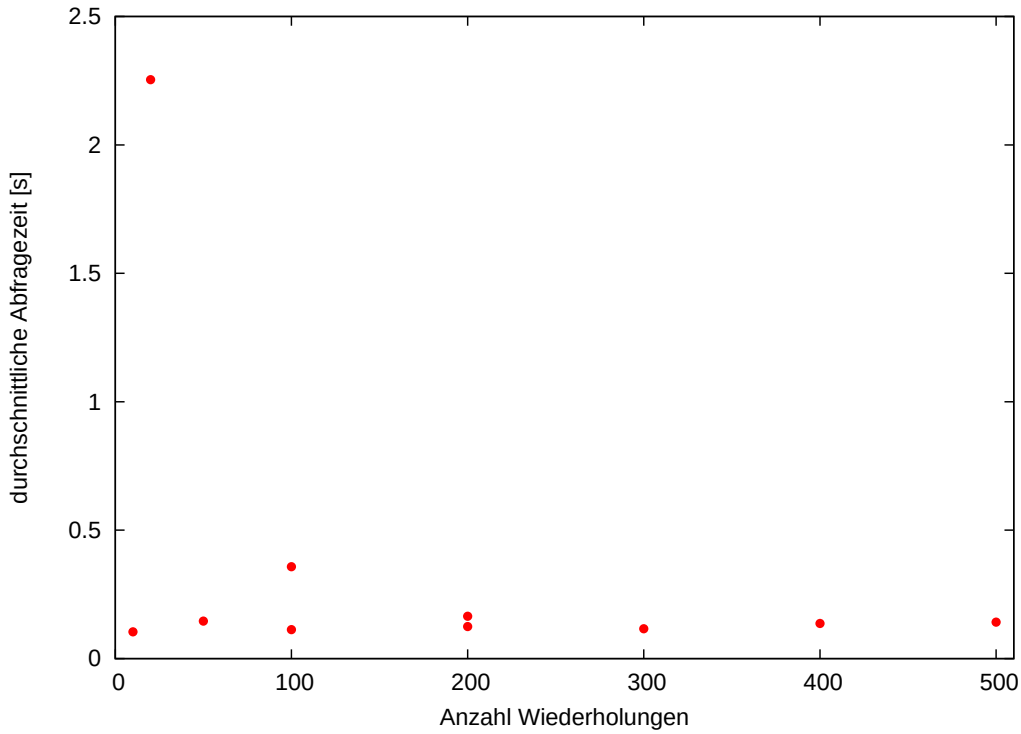

Abbildung 5.2: **durchschnittliche Abfragezeiten** in Abhängigkeit der Anzahl durchgeführter Wiederholungen für die Suche nach **”Funktionsparameter”** auf dem **Datenbanksystem**

Auffällig ist der extrem hohe Durchschnittswert bei 20 Wiederholungen. Dieser macht deutlich, dass Durchschnittswerte, die aus geringen Anzahlen von Wiederholungen resultieren, kaum Aussagekraft besitzen. Daher sind Schlussfolgerungen aus den druchschnittlichen Abfragezeiten nur bedingt (nämlich erst ab sehr großen Anzahlen von Wiederho-

lungen) zulässig.

Das Schwanken der Durchschnittswerte hängt maßgeblich mit den in Abbildung 5.1 auf Seite 52 zu erkennenden Ausreißern zusammen. Deren Zeiten betragen ein Vielfaches der durchschnittlichen Abfragezeit und können Werte von mehreren Sekunden annehmen. Bei 20 Wiederholungen der Abfrage auf dem Datenbanksystem ergab sich z.B. als maximale Zeit ein Wert von 5 Sekunden. Auch bei weiteren Tests mit unterschiedlichen Abfragen und Wiederholungsanzahlen auf beiden Systemen traten solche Ausreißer immer wieder auf. Das Streuen der Abfragezeiten ist also weder von der verwendeten Installationsvariante noch vom verwendeten Suchbegriff abhängig.

Der Grund hierfür scheinen Prozesse und Vorgänge innerhalb der Datenbank zu sein, die in unregelmäßigen Abständen ausgeführt werden. So werden etwa zu bestimmten Zeiten mehrere Jobs unter dem Benutzer-Account ”WKSYS”, welcher für die Verwaltung von Oracle Ultra Search innerhalb der Datenbank verwendet wird, ausgeführt.

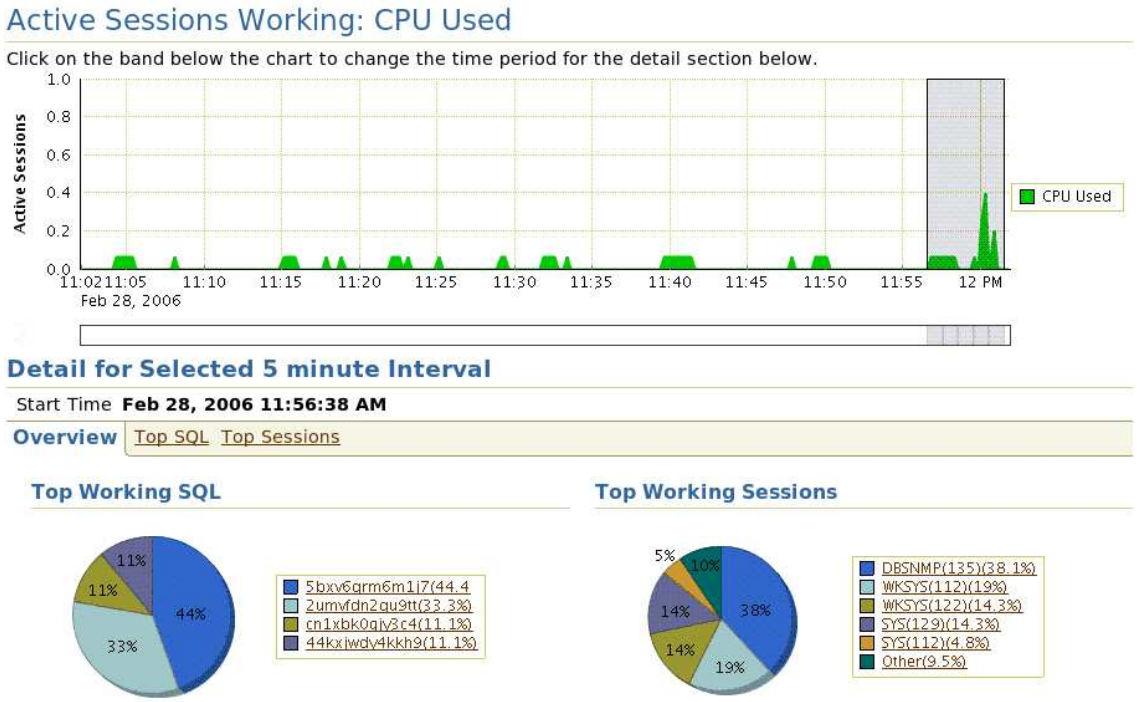

Abbildung 5.3: **CPU-Auslastung** und zugehörige Datenbank-Accounts zwischen den Messungen auf dem **RDBMS**

Zudem werden auch Jobs zur allgemeinen Datenbankverwaltung im Hintergrund ausgeführt. Schaut man sich die Performance-Statistiken der beiden verwendeten Testsyste-

me an, stellt man fest, dass auch zwischen den Zeitmessungen immer wieder kurzzeitige Belastungen der Datenbanken zu erkennen sind (vgl. Abbildung 5.3 auf der vorherigen Seite), obwohl sich keine weiteren Benutzer auf dem System befinden.
Dieses kurzzeitige Anssteigen der Datenbankauslastung wird, wie in der Grafik zu erkennen, von den zur Verwaltung der Datenbank und Oracle Ultra Search verwendeten System-Accounts ”SYS”, ”DBSNMP” und ”WKSYS” verursacht.

Weiterhin werden beide Testsysteme von einem zentralen Administrations-Werkzeug aus überwacht, mit dessen Hilfe die obige Abbildung erstellt wurde. In regelmäßigen Abständen werden Auslastung und weitere Eckdaten der Datenbanken ermittelt und an das Administrations-Werkzeug gesendet. Dieser Vorgang kann ebenfalls eine kurzzeitige Belastung der Datenbanken darstellen und somit einen Einfluss auf die Zeitmessung haben.
Auch die Architektur von Oracle Ultra Search kann durch die Ausführung der Suche über JavaServer Pages und Servelets einen Einfluss auf die Streuung der Abfragezeiten haben.

Um einen verlässlichen Durchschnittswert für die Abfragezeiten zu erhalten müsste auf Grund der oben beschriebenen Ausreißer eine sehr große Anzahl Wiederholungen zu jeder einzelnen Abfrage durchgeführt werden. Dies ist jedoch bei der Vielzahl von im Rahmen der Zeitmessungen durchgeführten Abfragen aus Zeitgründen nicht möglich. Um auch bei wenigen Wiederholungen einen aussagekräftigen Durchschnittswert zu erhalten, müssten bei der Messung diejenigen Werte, die stark von den übrigen Werten abweichen, (die sogenannten Ausreißer) entfernt werden. Eine weitere Möglichkeit, eine repräsentative Zeit bei einer geringen Anzahl Wiederholungen für eine Abfrage zu erhalten, ist das Ermitteln der minimalen Abfragezeit.

Betrachtet man die Abfragezeiten der 500 Durchläufe aus der Abbildung 5.1 auf Seite 52 genauer, ist zu erkennen, dass die Abfragezeiten zwar stark nach oben streuen, allerdings nach unten beschränkt zu sein scheinen. Weiterhin befindet sich die Mehrheit der Zeiten in unmittelbarer Nähe zur minimalen Abfragezeit (416 der 500 Werte weichen weniger als 0,05 Sekunden vom Minimum 0,316 ab). Dieses Verhalten wurde auf beiden System bei verschiedenen Abfragen festgestellt.
Setzen wir die minimalen Abfragezeiten in Abhängigkeit der durchgeführten Wiederholungen, sehen wir, dass die Schwankungen für die minimalen Abfragezeiten sich lediglich im Millisekundenbereich bewegen, also auf beiden Systemen nahezu unabhängig von der Anzahl der Wiederholungen sind (vgl. Abbildung 5.4 auf der nächsten Seite).
Um Vergleiche zwischen verschiedenen Suchabfragen sowie die Auswirkung verschiedener Faktoren auf die Abfragezeit zu untersuchen, ist die minimalen Abfragezeit aus mehreren Wiederholungen einer Abfrage somit gut geeignet. Daher werden bei den weiteren Messungen die minimalen Abfragezeiten aus 50 Wiederholungen für die jeweilige Suchanfrage untersucht.

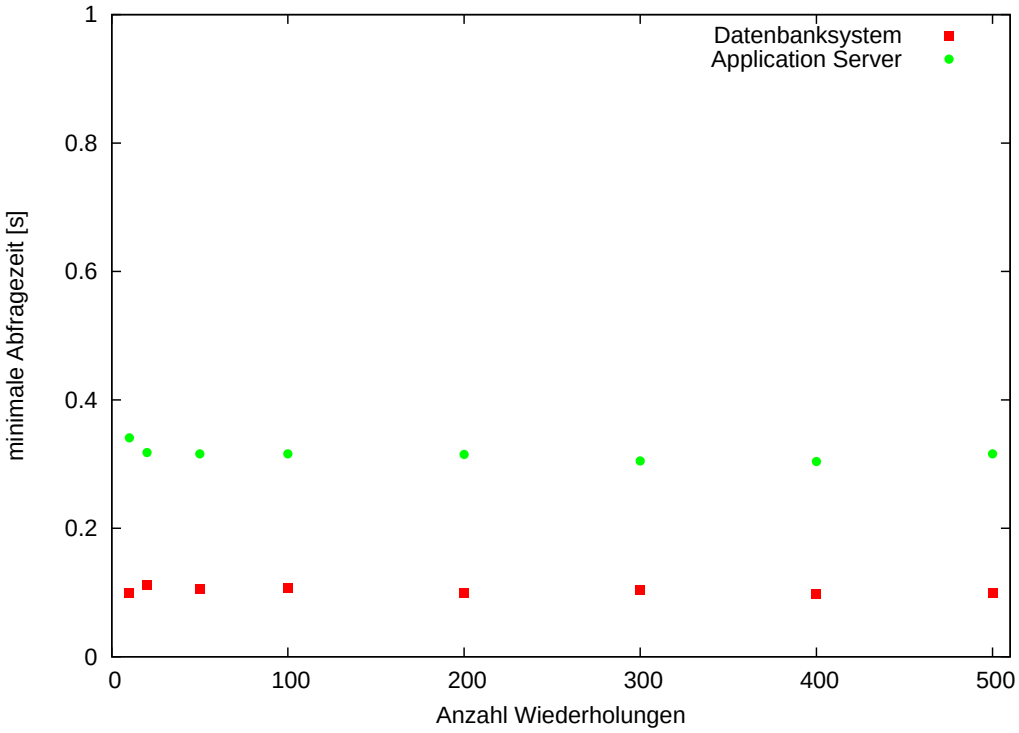

Abbildung 5.4: **minimale Abfragezeiten** in Abhängigkeit der Wiederholungen für die Suche nach dem Begriff **”Funktionsparameter”** auf dem **RRDBMS und** dem **AS**

Die folgende Tabelle 5.1 auf der nächsten Seite fasst die Abfragezeiten für die zur Stabilitätsuntersuchung durchgeführte Suche nach dem Begriff ”Funktionsparameter” auf beiden Systemen (Datenbanksystem und Application Server) noch einmal zusammen.
Es fällt auf, dass bei den durchgeführten Testmessungen für den Suchbegriff ”Funktionsparameter” die Oracle Ultra Search Installation auf dem Datenbanksystem sowohl bei den minimalen als auch bei den durchschnittlichen Abfragezeiten deutlich schneller war als die des Application Servers. Um diesen Eindruck zu bestätigen oder zu widerlegen, ist auch bei den weiteren Messungen ein Vergleich zwischen den Installationsvarianten anzustellen.

System	**Wiederholungen**	x_{min} **[s]**	x_{max} **[s]**	$\overline{x}$ **[s]**
AS	10	0,341	1,861	0,402
AS	20	0,318	0,503	0,348
AS	50	0,316	0,958	0,379
AS	100	0,316	0,658	0,336
AS	200	0,315	0,912	0,35
AS	300	0,305	2,173	0,346
AS	400	0,304	0,982	0,348
AS	500	0,316	1,029	0,350
RDBMS	10	0,100	0,311	0,104
RDBMS	20	0,112	5,003	2,254
RDBMS	50	0,106	0,530	0,146
RDBMS	100	0,107	5,400	0,358
RDBMS	200	0,100	0,581	0,113
RDBMS	300	0,104	4,00	0,165
RDBMS	400	0,097	0,766	0,116
RDBMS	500	0,099	5,002	0,137

Tabelle 5.1: Abfragezeiten für die Suche nach dem Begriff **”Funktionsparameter”** auf **beiden Testsystemen** (verwendete **Instanz: 10**)

5.2 Test: Auswirkung der Indexgröße auf die Abfragezeit

Es liegt die Vermutung Nahe, dass die Dauer für die Durchführung einer Suchabfrage von der Größe des Indexes, der durchsucht wird, und somit von der Anzahl indexierter Dokumente abhängt - also eine Suche in einem großen Index entsprechend länger dauert. Diese Vermutung soll anhand mehrerer Testanfragen überprüft werden.

5.2.1 Durchführung der Messungen

- Auf **beiden Testsystemen** werden folgende drei Abfragen durchgeführt:

Abfrage Id	Suchbegriff
1	”Wissenschaft”
2	”Oracle -Datenbank”
3	”Datenv*n”

  Bei Suchbegriff 2 werden Dokumente gesucht, die das Wort ”Oracle”, aber nicht das Wort ”Datenbank” enthalten.

  Der dritte Suchbegriff enthält den Platzhalter ’*’ der durch beliebig viele Buchstaben ersetzt werden kann. So werdem z.B. Dokumente gefunden, die den Begriff ”Datenvolumen” enthalten.

- Die Anfragen werden jeweils auf den **Testindexen 1 bis 10** durchgeführt

5.2.2 Auswertung der Messungen

Die drei folgenden Diagramme zeigen die Abfragezeiten für die drei Suchanfragen jeweils auf den Indexen 1 bis 10 beider Testsysteme, wobei Index 1 derjenige mit der gringsten Dokumentanzahl ist, und Index 10 die meisten Dokumente besitzt.

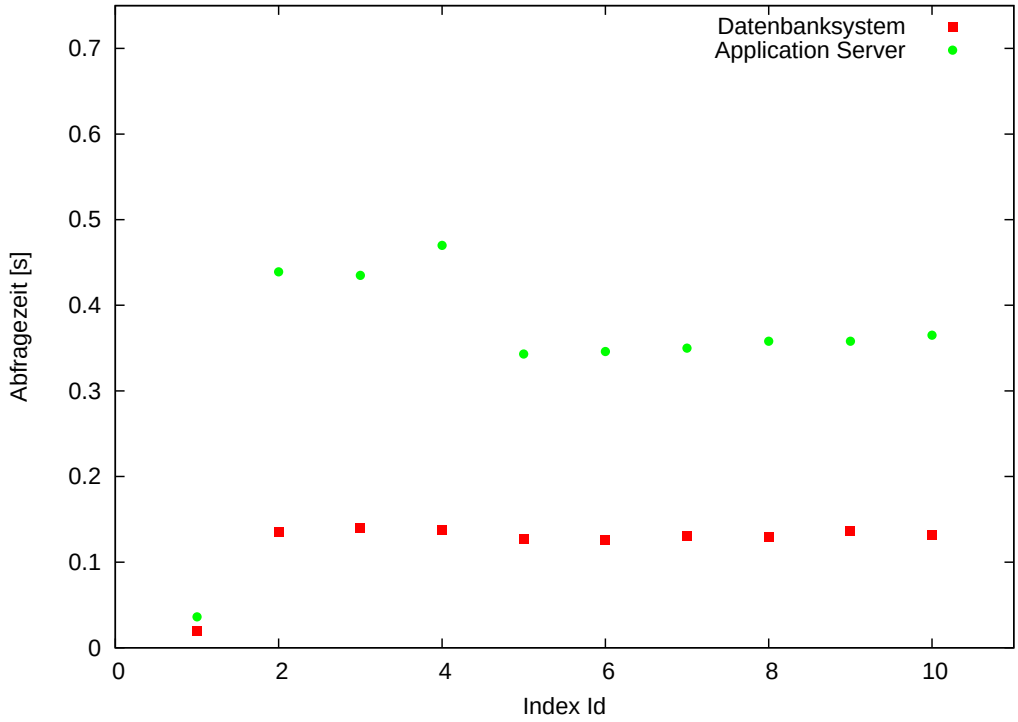

Abbildung 5.5: Abfragezeiten für die Suche nach dem Begriff **”Wissenschaft”** auf den **Indexen 1 bis 10 beider Testsysteme**

Bei der Suche nach dem Begriff ”Wissenschaft” hebt sich der Index 1, der die geringste Anzahl an Dokumenten besitzt, auf beiden Systemen deutlich von den übrigen Indexen ab und besitzt die mit Abstand kleinste Abfragezeit.
Auf dem Datenbanksystem unterscheiden sich die Zeiten für die weiteren Indexe nur wenig. Eine Tendenz ist nicht zu erkennen.
Auf dem Application Server liegen die Zeiten für die Indexe 2 bis 4 deutlich über denen der übrigen Indexe. Bei den Indexen 5 bis 10 ist eine leichte Steigerung der Abfragezeiten zu erkennen.

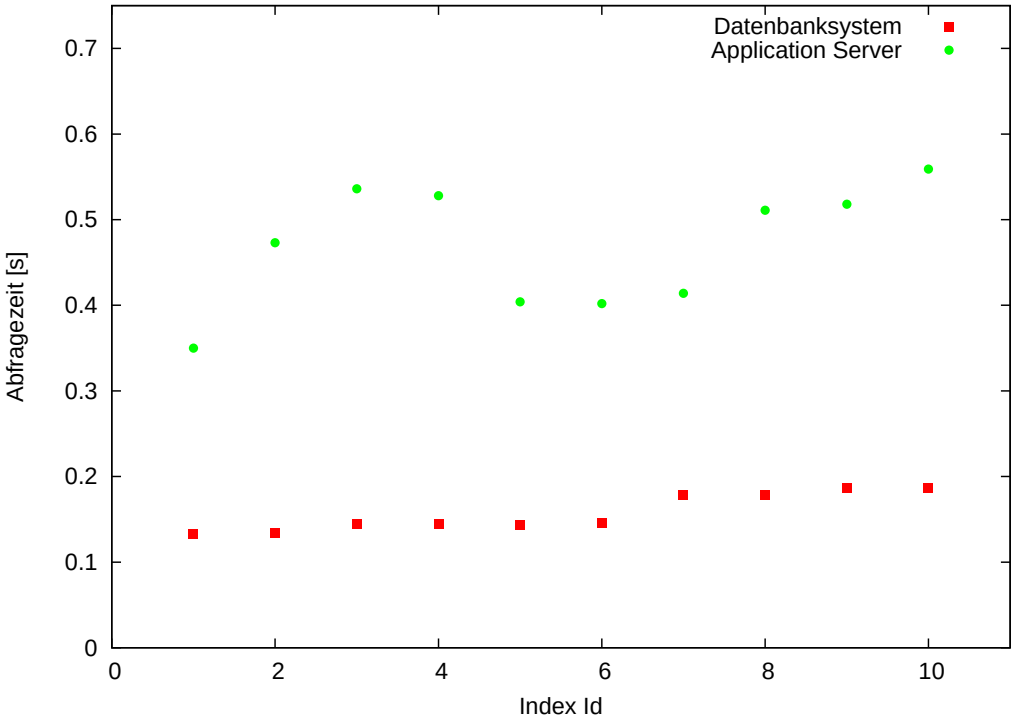

Abbildung 5.6: Abfragezeiten für die Suche nach **”Oracle -Datenbank”** auf den **Indexen 1 bis 10 beider Testsysteme**

Bei den Zeiten für die zweite Abfrage ist auf dem Datenbanksystem eine leichte Zunahme der Abfragezeiten mit steigender Indexgröße im Index zu erkennen. Jedoch beträgt die Differenz zwischen der Abfragezeit auf Index 1 (0,133 Sekunden) und Index 10 (1,87 Sekunden) lediglich 0,054 Sekunden, der Zeitunterschied ist also für den Suchanwender kaum wahrnehmbar.
Auf dem Application Server schwanken die Werte stärker. Wie bei der ersten Abfrage fallen auch hier die Zeiten für die Indexe 2 bis 4 auf, da sie deutlich größer als die der nachfolgenden Indexe 5 bis 7 sind. Die Zeiten für die Indexe 8 bis 10 weisen eine leicht steigende Tendenz auf.

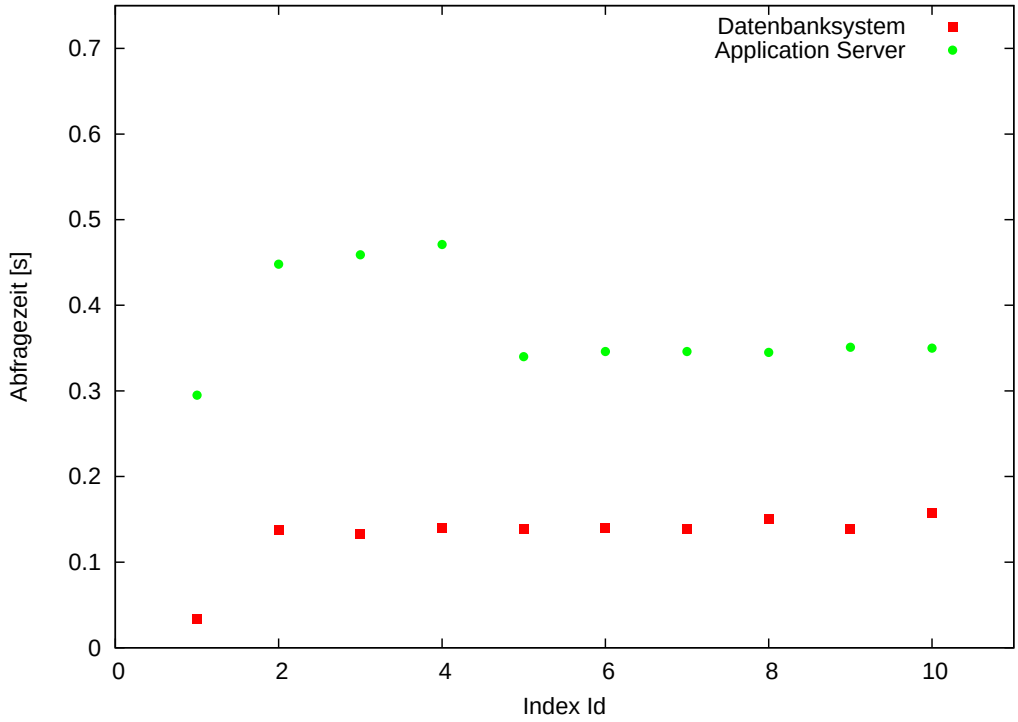

Abbildung 5.7: Abfragezeiten für die Suche nach **”Datenv*n”** auf den **Indexen 1 bis 10 beider Testsysteme**

Bei der dritten Abfrage, ’Datenv*n’, zeigt sich ein ähnliches Bild wie bei der ersten. Auf dem Application Server liegen die Abfragezeiten für die Indexe 2 bis 4 deutlich über denen der anderen Indexe. Zudem hebt sich Index 1 nach unten ab. Letzteres gilt auch für das Datenbanksystem, bei dem die übrigen Werte sich jedoch kaum voneinander unterscheiden und allenfalls einen minimalen Anstieg aufweisen.

5.2.3 Bewertung der Ergebnisse

Zusammenfassend lässt sich sagen, dass besonders auf Grund der schwankenden Abfragezeiten auf dem Application Server für die getesteten Indexgrößen kein eindeutiger Zusammenhang zwischen der Indexgröße und der benötigten Zeit für eine Suchabfrage zu erkennen ist.

Die Zeiten auf dem Datenbanksystem sind wie bei der vorangegangen Stabilitätsprüfung deutlich geringer als die des Application Servers. Zudem streuen sie weniger.

5.3 Test: Auswirkung der Trefferanzahl auf die Abfragezeit

Es soll geprüft werden, ob ein Zusammenhang zwischen der Anzahl zu einer Suchanfrage gefundener Treffer und der für die Durchführung dieser Abfrage benötigten Zeit festzustellen ist.

5.3.1 Durchführung der Messungen

- Für diese Untersuchung werden **65 Abfragen mit unterschiedlichen Trefferanzahlen** durchgeführt
- Bei den Suchanfragen handelt es sich um **einzelne Wörter**, um einen Einfluss auf die Abfragezeiten durch komplexe Suchanfragen (die Verwendung von Abfrageoperatoren) auszuschließen
- Auf **beiden Testystemen** wird jeweils der **Index 10** verwendet, da dieser die meisten Dokumente enthält
- Eine Verlängerung der Abfragezeit auf Grund der erhöhten Ausgabe bei vielen Treffern zu einer Suchanfrage wird dadurch ausgeschlossen, dass zu jeder Anfrage lediglich die ersten **10 Treffer ausgegeben** werden. Die Auswirkung verschiedener Trefferlistendarstellungen (z.B. die Erhöhung der Anzahl pro Seite angezeigter Treffer) wird im nächsten Test untersucht.

5.3.2 Auswertung der Messungen

Die folgende Abbildung zeigt die ermittelten Abfragezeiten für die verschiedenen Suchanfragen in Abhängigkeit der Trefferanzahl für die jeweilige Abfrage. Die Abfragezeiten der beiden Testsysteme werden gegenübergestellt.

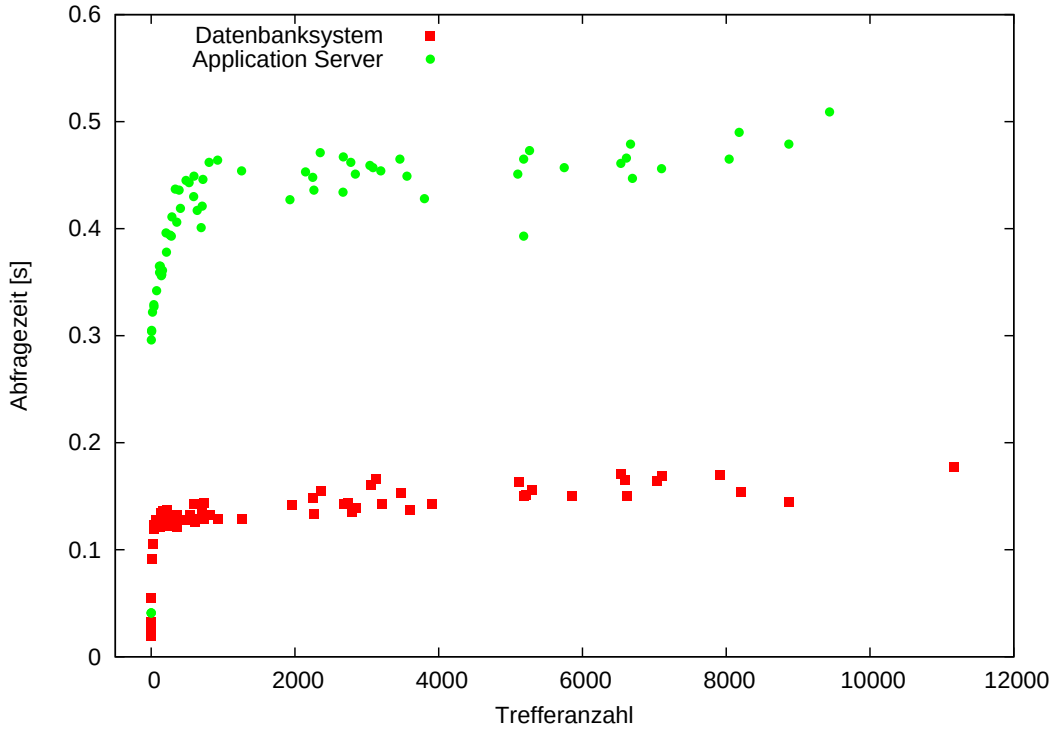

Abbildung 5.8: Abfragezeiten in **Abhängigkeit der Trefferanzahl** für 65 Suchbegriffe auf dem **Index 10 beider Testsysteme**

Auf beiden Systemen ist ein ähnliches Verhalten der Abfragezeiten zu erkennen. Für kleine Trefferanzahlen sind die Zeiten deutlich geringer als für höhere Trefferanzahlen. In diesem Bereich ist ein sehr starker Anstieg der Zeiten zu erkennen. Bei größeren Trefferanzahlen ist dagegen nur noch ein leichter Anstieg der Abfragezeiten zu erkennen.
Besonders die Abfragezeiten auf dem Application Server weisen Ähnlichkeit zu einer logarithmischen Funktion auf. Daher wird für diese Messwerte eine Ausgleichsfunktion mit folgenden Parametern ermittelt:

$$f(x) = a * log(x) + b$$

Die Abfragezeit für die Trefferanzahl 0 wird bei der Berechnung der Ausgleichsfunktion nicht berücksichtigt, da die Logarithmusfunktion nur für Werte größer 0 definiert ist. Die folgende Abbildung zeigt die Messwerte zusammen mit der ermittelten Ausgleichsfunktion:

$$f(x) = 0,023075 * log(x) + 0,268309$$

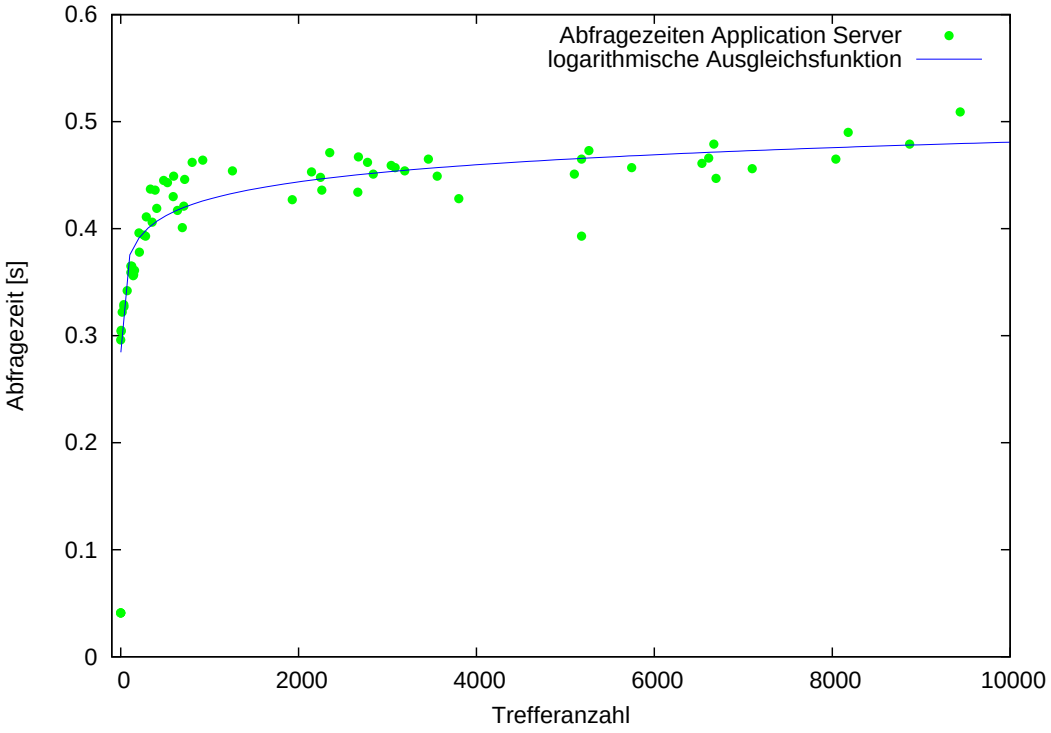

Abbildung 5.9: Abfragezeiten für 65 Suchbegriffe in **Abhängigkeit der Trefferanzahl** auf dem **Application Server** zusammen mit einer **logarithmischen Ausgleichsfunktion**

5.3.3 Bewertung der Ergebnisse

Auf beiden Systemen ist eine deutliche Abhängigkeit der Abfragezeiten von der Trefferanzahl zu erkennen, obwohl die Ausgabe der Treffer bei allen Abfragen auf 10 limitiert ist. Dieser Zusammenhang zwischen Abfragezeit und Trefferanzahl ist jedoch nicht linear, sondern gleicht einer logarithmischen Funktion. Bei wenigen Treffern steigt die Abfragezeit wesentlich stärker mit der Trefferanzahl als bei großen Trefferanzahlen.
Die Abfragezeiten auf dem Application Server sind - außer für die Abfrage mit 0 Treffern - wie in den vorherigen Tests deutlich höher als die des Datenbanksystems.
Zudem unterscheiden sich die Trefferanzahlen der Abfragen auf den beiden Testsystemen teilweise deutlich. So beträgt die maximale Trefferanzahl auf dem Datenbanksystem 11.174, während auf dem Application Server maximal 9.438 Treffer für eine Abfrage erzielt werden. Dies ist auf die unterschiedlichen Ultra Search Versionen zurückzuführen. Neben dem Funktionsumfang scheinen sich zum Teil auch die Algorithmen für die Ausführung einer Suchabfrage geändert zu haben, so dass je nach Version mehr oder weniger Treffer zu einem Suchbegriff gefunden werden.

5.4 Test: Darstellung der Trefferliste

In Oracle Ultra Search besteht die Möglichkeit, die Darstellung der Trefferliste zu einer Suchanfrage über die verwendeten JavaServer Pages zu verändern. Neben optischen Gestaltungsmöglichkeiten, wie dem Einstellen von Schriftart und Hintergrundfarben, lassen sich auch inhaltliche Anpassungen, wie die Anzahl der Treffer pro Seite und die zu den Treffern angezeigten Informationen, vornehmen.
Bei der folgenden Zeitmessung soll die Auswirkung der Ausgabe zusätzlicher Dokumentinformationen zu jedem Treffer sowie der Erhöhung der Anzahl ausgegebener Treffer (standardmäßig werden unabhähgig von der Anzahl gefundener Dokumente nur 10 Treffer pro Seite ausgegeben) untersucht werden.

5.4.1 Durchführung der Messungen

- Folgende 10 Abfragen werden auf beiden Testsystemen ausgeführt:

Abfrage Id	Suchbegriff	Abfrage Id	Suchbegriff
1	function	6	daten
2	parameter	7	+sql +php
3	number	8	sql -my*
4	program	9	php mathe mysql der
5	tabelle	10	datenbank

- Die Abfragen werden auf dem **Index 13 der beiden Testsysteme** mit Hilfe verschiedener JavaServer Pages durchgeführt, so dass **unterschiedliche Trefferausgaben** verwendet werden können:
  - **Ausgabe 1**: Standardausgabe der Trefferliste mit 10 Treffern pro Seite (vgl. Abbildung 4.4 auf Seite 48)
  - **Ausgabe 2**: zusätzliche Ausgabe eines Auszugs (200 Zeichen) aus dem Dokument, der den Suchbegriff enthält (”Keyword in context”). Da die Möglichkeit, einen Auszug aus dem Dokument mit dem jeweiligen Suchbegriff darzustellen erst in neueren Versionen von Oracle Ultra Search zur Verfügung steht, wurde diese Funktion nur auf dem Datenbanksystem getestet. Auf dem Application Server wurden stattdessen die ersten 100 Zeichen des Dokuments angezeigt.
  - **Ausgabe 3**: wie bei der 2. Ausgabe werden zusätzliche Trefferinformationen angezeigt, zudem wird die Anzahl dargestellter Treffer von 10 auf 200 pro Seite erhöht

5.4.2 Auswertung der Messungen

In den folgenden beiden Diagrammen werden die Zeiten der auf den beiden Testsystemen durchgeführten Abfragen dargestellt. Dabei werden jeweils die Abfragezeiten aller drei Trefferausgaben parallel in einem Diagramm angezeigt.

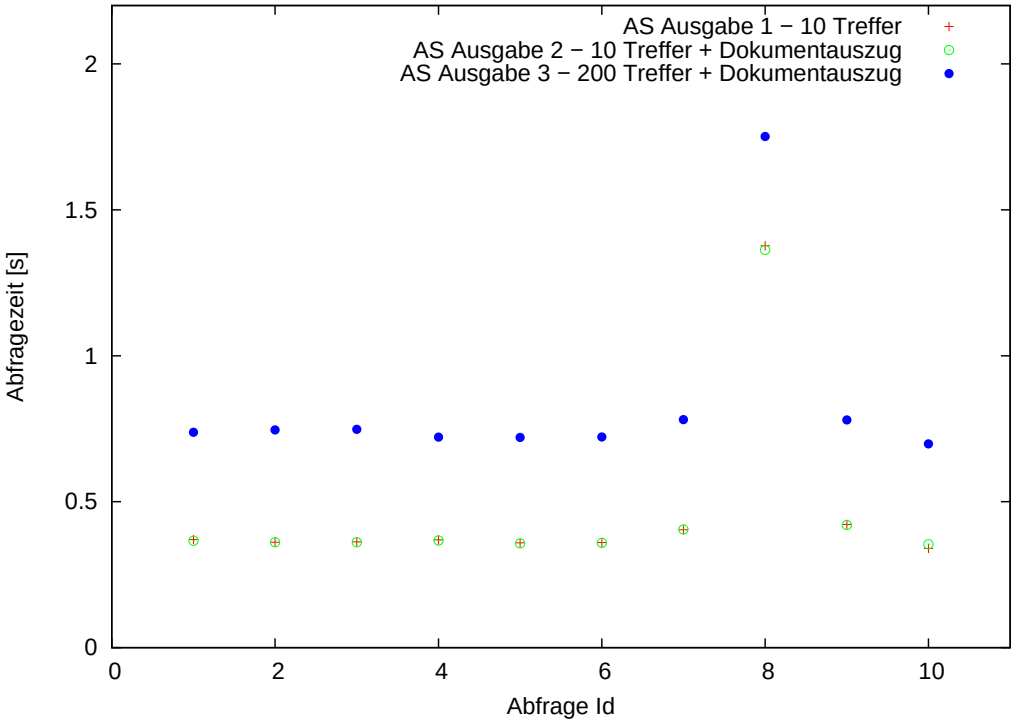

Abbildung 5.10: Abfragezeiten für **10 Beispielabfragen** mit **verschiedenen Trefferausgaben** auf dem **Application Server**

Während die Abfragezeiten für die Standardausgabe der Trefferliste (Ausgabe 1) und die um eine Dokumentbeschreibung ergänzte Trefferlistenausgabe (Ausgabe 2) nahezu identisch sind, sind die Zeiten für die 3. Ausgabe (200 Treffern pro Seite) erheblich größer (ca. 0.4 Sekunden, also bis auf Abfrage 8 beinahe doppelt so groß). Die Differenzen zwischen Ausgabe 1 und Ausgabe 3 sind nahzu konstant, so dass es sich bei der Messkurve für die Ausgabe 3 um ein nach oben verschobenes Abbild der Messkurve zur Ausgabe 1 handelt.

Auffällig ist die Abfragezeit für die Suchanfrage 8, die für alle Trefferdarstellungen deutlich über denen der anderen Abfragen liegt. Ein Grund hierfür könnte sein, dass diese Suchanfrage als einzige den Platzhalter '*' enthält. Ob die Verwendung von Platzhaltern generell die Abfragezeiten erhöhen, wird in einem späteren Test untersucht.

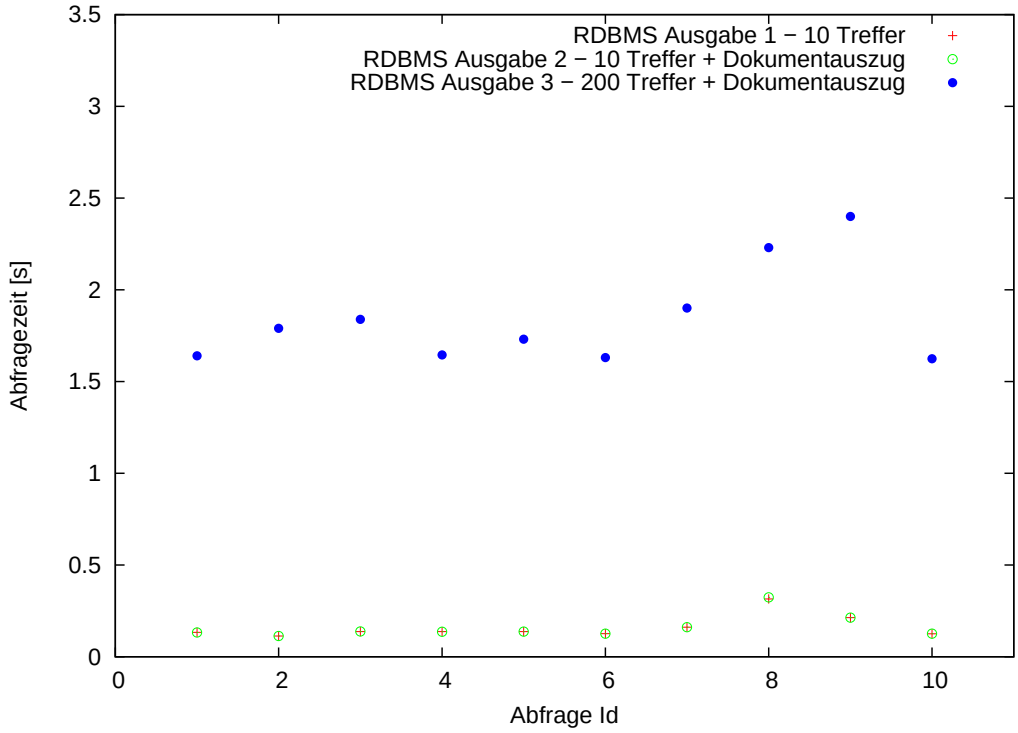

Abbildung 5.11: Abfragezeiten für **10 Beispielabfragen** mit **verschiedenen Trefferausgaben** auf dem **Datenbanksystem**

Auch auf dem Datenbanksystem (vgl. Abbildung 5.11) ist kein Unterschied zwischen den Abfragezeiten für die Standardausgabe (Ausgabe 1) und die um einen Dokumentauszug erweiterte Ausgabe der Trefferliste (Ausgabe 2) zu erkennen. Allerdings liegen auf diesem System die Zeiten für die Trefferausgabe mit 200 Treffern pro Seite ebenfalls erheblich darüber. Diese Zeiten schwanken zudem stark, so dass keine einheitliche Verschiebung der Werte zu erkennen ist. Die Differenzen für die Zeiten liegen ungefähr zwischen 1,5 Sekunden und 2,2 Sekunden (was z.B. für Abfrage 9 dem Faktor 10 entspricht), sind also mehr als dreimal so groß wie die auf dem Application Server.

Die Zeit für Suchanfrage 8 hebt sich trotz des verwendeten Platzhalters im Gegensatz zum Application Server nicht von den übrigen Zeiten ab.

5.4.3 Bewertung der Ergebnisse

Während das Anzeigen zusätzlicher Dokumentinformationen, wie z.B. eines kurzen Dokumentauszuges, keine Auswirkung auf die Abfragezeit besitzt, verschlechtern sich die Abfragezeiten durch die Erhöhung der Anzahl angezeigter Treffer pro Seite auf beiden Testystemen erheblich, weil mehr Treffer für die Anzeige in der Trefferliste aufbereitet werden müssen. Bei 200 angezeigten Treffern pro Seite (Ausgabe 3) liegen die Abfragezeiten auf dem Datenbanksystem erstmals über denen des Application Servers.

5.5 Test: Suche mit Platzhaltern

Es soll geprüft werden, wie sich die Verwendung der in Oracle Ultra Search standardmäßig zur Verfügung gestellten Abfrageoperatoren '+', '-' und '*' auf die Abfragezeiten auswirkt. Dabei wird ein besonderes Augenmerk auf die Platzhaltersuche und das hierfür eingesetzte Prefix-Indexing gelegt, wobei auch Wortteile bis hin zu einzelnen Buchstaben in den Index aufgenommen werden. (vgl. Kapitel 4.5 auf Seite 45).

5.5.1 Durchführung der Messungen

- Die folgenden 15 Suchanfragen, die **zum Teil mehrere Platzhalter** enthalten, werden auf beiden Testsystemen ausgeführt:

Abfrage Id	Suchbegriff	Abfrage Id	Suchbegriff
1	Benutzer	6	Benutzer*
2	computer	7	computer*
3	data	8	data*
4	index	9	index*
5	mathe	10	mathe*

Abfrage Id	Suchbegriff
11	si*m*n (z.B. Simulation)
12	F*t*e*r (z.B. Filter)
13	ob*er*n* (z.B. observieren)
14	+si*m*n +Da*t*n (z.B. +Simulation +Daten)
15	+o*er*n* +so*t*e -pass* (z.B. +observieren +Software -Passwort)
16	B*er* (z.B. Benutzer)

- Die Abfragen werden auf den Indexen 10 und 14 beider Testsysteme, also einmal **ohne und** einmal **mit Prefix-Indexing**, durchgeführt.

5.5.2 Auswertung der Messungen

Die folgenden Diagramme zeigen die Abfragezeiten auf den jeweiligen Indexen der beiden Testsysteme. Dabei wird die Suche mit (Index 14) und ohne Prefix-Indexing (Index 10) jeweils in einem Diagramm gegenübergestellt.

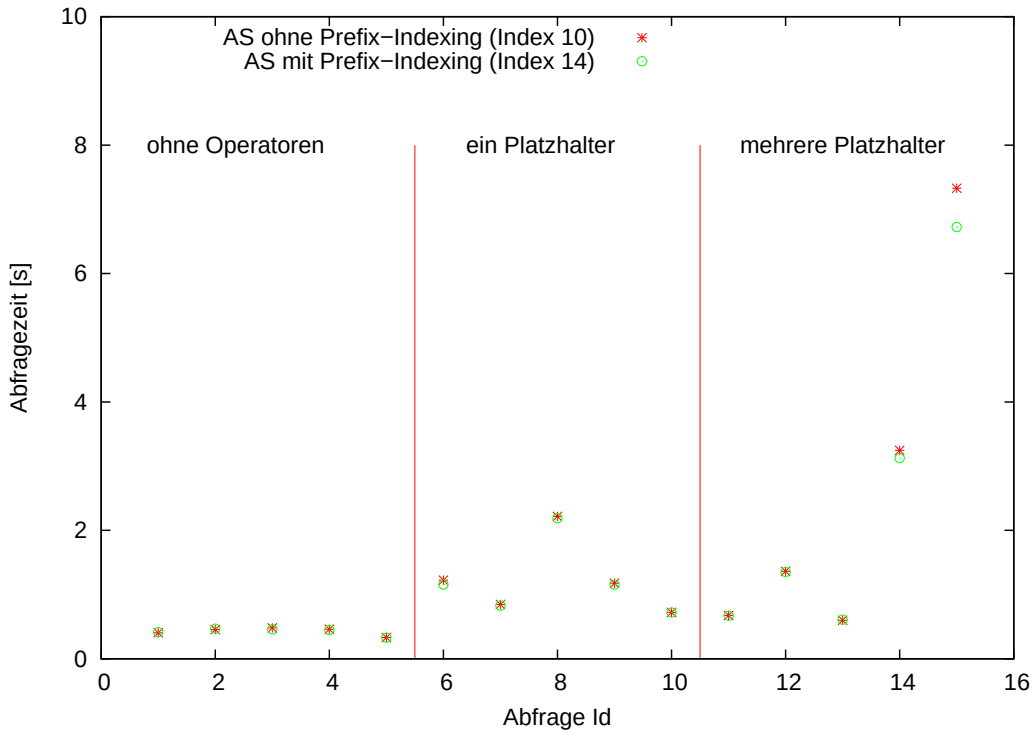

Abbildung 5.12: Abfragezeiten für **15 Abfragen** mit **verschiedenen Abfrageoperatoren** auf dem **Application Server**

Die Suchanfragen 1 bis 5 (ohne Platzhalter) benötigen auf beiden Indexen des Application Servers (also mit und ohne Prefix-Indexing) die geringsten Abfragezeiten und unterscheiden sich nur wenig voneinander. Die Zeiten für die übrigen Abfragen streuen stärker und liegen mit Werten von bis zu mehreren Sekunden deutlich höher. Die längste Zeit wurde für Abfrage 16 benötigt und beträgt mehr als 20 Sekunden (wegen der Übersichtlichkeit wurde diese Zeit nicht ins Diagramm aufgenommen). Die Abfragezeiten sind auf beiden Indexen nahezu identisch. Der Größte Zeitunterschied ist bei Abfrage 15 festzustellen. Dort dauert die Abfrage für Index 10 (ohne Prefix-Indexing) mit 7,328 Sekunden ca. 0.6 Sekunde länger als für Index 14 (6,725 Sekunden, mit Prefix-Indexing).
Obwohl Suchanfrage 15 (”+o*er*n* +so*t*e -pass*”) insgesamt sechs Platzhalter und zudem mehrere Verküpfungsoperatoren enthält, ist die für diese Abfrage benötigte Zeit

deutlich geringer als für Suchanfrage 16 (”B*er*”), die lediglich zwei Platzhalter enthält. Dies hängt mit der Menge der Wörter, auf die sich ein Ausdruck mit Platzhaltern erweitern lässt, zusammen. Gibt es viele mögliche Erweiterungen, dauert die Abfrage länger, da entsprechend mehr Zeilen des Indexes abgefragt werden müssen.

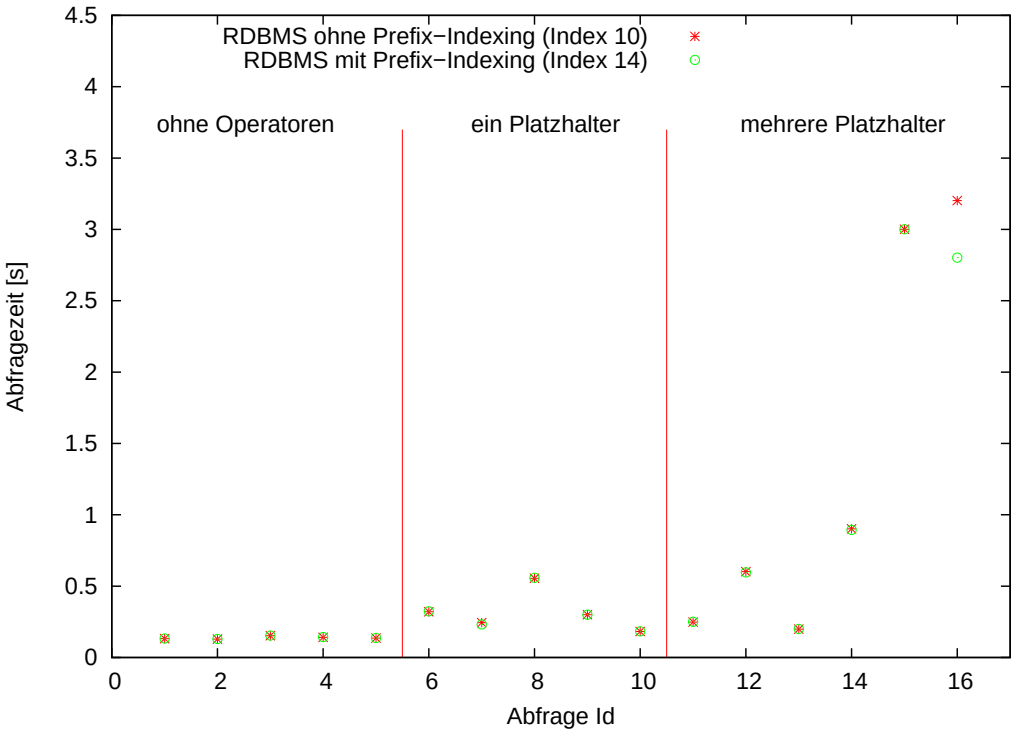

Abbildung 5.13: Abfragezeiten für **15 Abfragen** mit **verschiedenen Abfrageoperatoren** auf dem **Datenbanksystem**

Auf dem Datenbanksystem zeigt sich ein ähnliches Bild wie auf dem Application Server: die ersten 5 Suchanfragen benötigen die kürzesten Abfragezeiten und unterscheiden sich nicht wesentlich voneinander. Jedoch hebt sich Suchanfrage 16 hier nicht von den übrigen Anfragen ab, sondern besitzt eine ähnliche Zeit wie Abfrage 15. Die Abfragezeiten für beide Indexe (mit und ohne Prefix-Indexing) sind auf dem Datenbanksystem bei den ersten 15 Abfragen nahezu indentisch. Lediglich bei Abfrage 16 liegt die Zeit für Index 10 (3.202 Sekunden, ohne Prefix-Indexing) 0.4 Sekunden über der für Index 14 (2.802 Sekunden, mit Prefix-Indexing).

5.5.3 Bewertung der Ergebisse:

Auf beiden Systemen sind die Abfragezeiten bei der Verwendung von Platzhaltern höher als bei der Suche ohne Platzhalter. Die Abfragezeiten nehmen bei der Platzhaltersuche zum Teil sogar erheblich zu. Allerdings hat die Verwendung mehrerer Platzhalter nicht zwangsläufig eine hohe Abfragezeit zur Folge. Vielmehr spielen die Position der Platzhalter und die Anzahl der Wörter, auf die sich der Suchbegriff erweitern lässt, bei der Platzhaltersuche eine wichtige Rolle.
Das Indexieren von Wortteilen mit Hilfe der Index-Option 'prefix_index' scheint keinen Vorteil zu bringen, da fast alle Abfragezeiten auf beiden Systemen mit und ohne Prefix-Indexing identisch sind. Die minimalen Vorteile des Prefix-Indexing (bei den Abfragen 14, 15 und 16 auf dem Application Server sowie Abfrage 16 auf dem Datenbanksystem) rechtfertigen also keinesfalls den Mehraufwand bei der Indexerstellung bezüglich Indexierungszeit und Speicherplatzbedarf.
Auffällig ist auch in diesem Test, dass die Zeiten für die Abfragen auf dem Application Server deutlich höher sind - für Abfrage 16 sogar mehr als sechsmal so groß - als auf dem Datenbanksystem.

5.6 Test: Verschiedensprachige Dokumente

Einige Algorithmen bei der Volltextsuche wie z.B. die Fuzzy-Suche oder die Verwendung von Wortstämmen unterscheiden sich je nach Sprache der in den Index aufgenommenen Dokumente und Wörter. Auch einige der Indexeigenschaften sind abhängig von der Dokumentsprache. Ob die Sprache der indexierten Dokumente einen Einfluss auf die Abfragezeit hat, sollen die folgenden Zeitmessungen klären.

5.6.1 Durchführung der Messungen

- Auf beiden Testsystemen werden insgesamt 15 Abfragen druchgeführt. Dabei handelt es sich um 5 Begriffe, die einmal **ohne Abfrageoperatoren** (Abfragen 1 ... 5) und zusätzlich für eine **Wortstamm-Suche** (Abfragen 6 ... 10, durch '$' gekennzeichnet) bzw. für eine **Fuzzy-Suche** (Abfragen 11 ... 15) verwendet werden:

Id	Suchbegriff	Id	Suchbegriff	Id	Suchbegriff
1	information	6	$information	11	fuzzy(information)
2	computer	7	$computer	12	fuzzy(computer)
3	matrix	8	$matrix	13	fuzzy(matrix)
4	navigation	9	$navigation	14	fuzzy(navigation)
5	java	10	$java	15	fuzzy(java)

- Diese insgesamt 15 Suchanfragen werden jeweils auf einem Index mit **deutschen (Index 11) und** einem aus **englischsprachigen Dokumenten** aufgebauten Index **(Index 12)** auf beiden Testsystemen ausgeführt.
- zudem werden die Abfragen 2, 7 und 12 auch auf Index 10 beider Testsysteme ausgeführt, um eine Auswirkung der Operatoren bei einer größeren Anzahl Dokumente im Index zu prüfen

5.6.2 Auswertung der Messungen

Im folgenden Diagramm werden die Zeiten für die Suche nach obigen Begriffen betrachtet. Dabei werden die Abfragezeiten auf beiden Indexen (Index 11, Index 12) und beiden Systemen gegenübergestellt.

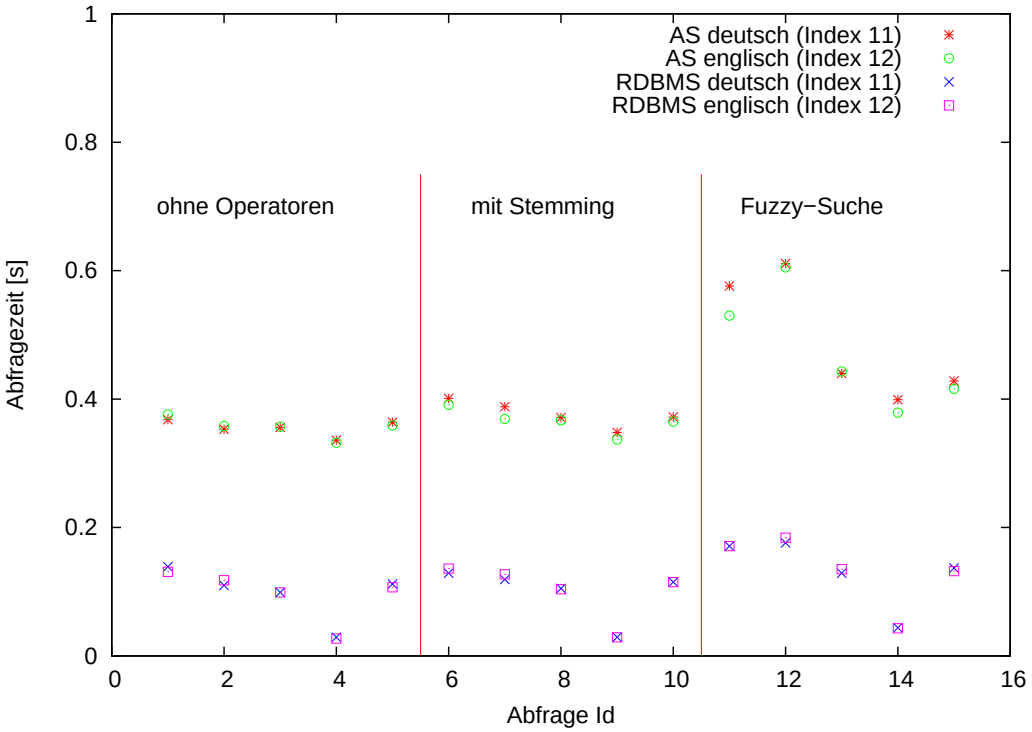

Abbildung 5.14: Abfragezeiten für **15 Suchbegriffe** auf einem **englischsprachigen und einem deutschsprachigen Index** beider Testsysteme

Auf beiden Systemen sind die Abfragezeiten für die 5 Suchbegriffe ohne Abfrageoperatoren auf dem deutsch- und dem englischsprachigen Index nahezu identisch. Zudem sind die Zeiten für den Application Server erneut größer als die des Datenbanksystems.

Auch bei der Verwendung des Stem-Operators (zur Wortstamm-Suche) ist weder auf dem Appliation Server noch auf dem Datenbanksystem ein wesentlicher Unterschied der Abfragezeiten zwischen dem deutsch- und dem englischsprachigen Index festzustellen.
Bei der Ähnlichkeitssuche mit Hilfe der Fuzzy-Funktion liegen auf dem Application Server die Zeiten für die Abfragen 11 und 14 auf dem deutschsprachigen Index leicht über denen des englischsprachigen Indexes. Allerdings beträgt der Unterschied weniger als 0,05 Sekunden.
Im Vergleich zu den Zeiten für die Wortstamm-Suche und die Suche ohne Abfrageoperatoren dauerte die Fuzzy-Suche auf beiden Systemen ein wenig länger. Bei der Suche auf einem größeren Index (Index 10), ist eine noch deutlichere Auswirkung der Fuzzy-Suche zu erkennen:

	computer	**$computer**	**fuzzy(computer)**
RDBMS	0,121 s	0,144 s	0,67 s
AS	0,457 s	0,447 s	3,635 s

Auf dem Application Server nimmt die Abfragezeit für eine Fuzzy-Suche gegenüber der Suche ohne Abfrageoperatoren und der Wortstamm-Suche um ca. 3 Sekunden zu. Auf dem Datenbanksystem ist der Unterschied nicht ganz so groß, aber mit ca. 0,5 Sekunden immer noch deutlich zu erkennen. Bezogen auf die Abfragezeiten ohne Operatoren entspricht das für das Datenbanksystem einer Verlängerung um den Faktor 5, auf dem Application Server dauert die Fuzzy-Suche sogar 8-Mal solange.

5.6.3 Bewertung der Ergebnisse

Die Sprache der indexierten Dokumente hatte bei den durchgeführten Testabfragen keinen nennenswerten Einfluss auf die Abfragezeit. Allerdings ist zu berücksichtigen, dass die Anzahl indexierter Dokumente für die verwendeten Testindexe sehr gering war. Zudem wurden lediglich die Sprachen ’Deutsch’ und ’Englisch’ verglichen. Es ist also nicht auszuschließen, dass bei größeren Indexen und anderen Dokumentsprachen dennoch sprachspezifische Auswirkungen auf die Abfragezeit festzustellen sind.

Die Fuzzy-Suche wirkt sich selbst bei kleinen Indexgrößen leicht negativ auf die Abfragezeiten aus. Bei größeren Dokumentanzahlen ist bereits ein deutlicher Unterschied zwischen der Fuzzy-Suche und der Wortstamm-Suche bzw. der Suche ohne Operatoren zu erkennen. Dies liegt daran, dass die Anzahl der Begriffserweiterungen bei der Fuzzy-Suche im Gegensatz zur Wortstamm-Suche von der Indexgröße abhängig sind. Bei sehr

großen Indexen ist damit zu rechnen, dass die Abfragezeit für eine Fuzzy-Suche extrem lange dauert und bei zu großen Dokumentanzahlen unter Umständen sogar zu Timeouts oder Fehlern führen kann.

Kapitel 6

Zusammenfassung

6.1 Eigenschaften von Oracle Ultra Search in Abgrenzung zu Oracle Text

Oracle Ultra Search ermöglicht durch seine oberflächenorientierte und Browser-basierte Struktur einen schnellen, unkomplizierten Einstieg in die Volltextsuche. Mit relativ wenig Aufwand lassen sich erste Suchapplikationen erstellen, die den Ansprüchen vieler Anwendungsgebiete bereits genügen. In der Regel ist jedoch eine Anpassung der Trefferausgabe an die eigenen Bedürfnisse vorzunehmen, um den Informationsgehalt der Trefferliste zu optimieren.
Mit dem zur Dokumentbeschaffung verwendeten und über Ausführungspläne automatisch steuerbaren Crawler ist Oracle Ultra Search sehr gut für die Suche verteilt gespeicherter Dokumente, wie z.B. Webseiten, geeignet und verwendet mit der Oracle Text Architektur ein solides und vielseitiges Werkzeug zur Indexerstellung und Indexabfrage. Auf Grund der größtenteils intuitiven Administration von Oracle Ultra Search ist Spezialwissen [1] nur in sehr geringem Maße erforderlich. Durch den geringen Aufwand zur Wartung einer Oracle Ultra Search-Suchanwendung und die Tatsache, dass bei einer Oracle Enterprise Edition Lizenz keine zusätzlichen Kosten für die Verwendung von Oracle Ultra Search entstehen, bietet dieses Produkt eine günstige Alternative zu anderen kostenpflichtigen oder aufwendig zu wartenden Suchlösungen.

Da Oracle Ultra Search hauptsächlich auf die Suche in Firmen-Netzwerken (Intranet) und Teilbereichen des Internets ausgerichtet ist und eine leichte Bedienbarkeit gewährleistet werden soll, ist der Funktionsumfang gegenüber einem vollständigen Textmanagementsystem wie Oracle Text eingeschränkt. Die Einstellungsmöglichkeiten und die Funktionsvielfalt von Oracle Ultra Search sind gegenüber der direkten Verwendung von Oracle

[1] zumindest für den Einstieg in die Volltextsuche

Text zum Teil reduziert. Zum Beispiel sind in Oracle Ultra Search nicht alle der 150 Dokumenttypen, die Oracle Text beherrscht, verfügbar. Für einfache Suchanwendungen wie die Volltextsuche innerhalb einer Webseite spielen diese Einschränkungen keine große Rolle. Sind die Anforderungen an die Suchfunktionalität jedoch komplexer und spezieller, sollte der Administrator der Suchapplikation über genaue Kenntnisse der Oracle Text Architektur und deren Handhabung verfügen. Für komplexe Suchanwendungen und Textfunktionalitäten, die über eine Volltextsuche hinausgehen und die z.B. in großen Verlagen benötigt werden, wird häufig eine direkte Verwendung von Oracle Text ohne Oracle Ultra Search, z.B. durch Einbettung in eigene mittels verschiedener Skriptsprachen erzeugte Oberflächen, bevorzugt. Auf diese Weise lässt sich eine Anpassung an die eigenen Anforderungen besser gewährleisten. Die Dokumentbeschaffung wird in diesem Fall durch eigene Implementierungen realisiert, da Oracle Text über keinen eigenen Crawler verfügt. Hinzu kommen etwa in Verlagshäusern auch juristische Fragen bezüglich der Dokumentbeschaffung: das Crawlen der Webseiten anderer Verlage zwecks Beschaffung von Textmaterial und Artikeln ist z.B. in diesem Kontext als rechtlich problematisch zu betrachten. Weiterhin verursachen komplexere Such- und Textfunktionen in der Regel wesentlich mehr Kosten und Aufwand zur Bereitstellung und Wartung.

6.2 Ergebnisse der Leistungsmessung

Die Zeitmessungen haben gezeigt, dass die Abfragezeiten bei der Volltextsuche mit Oracle Ultra Search Schwankungen unterliegen. Zwar unterscheiden sich ca. 80% der Werte zu einer Abfrage nur ganz gering, jedoch treten vereinzelt immer wieder Ausreißer nach oben auf, die sich von den übrigen Werten stark unterscheiden. Die Zeitspanne zwischen der minimalen und der maximalen Abfragezeit für denselben Suchbegriff kann daher einige Sekunden betragen. Dementsprechend sind die durchschnittlichen Abfragezeiten erst für sehr große Anzahlen von Wiederholungen (mindestens 500) aussagekräftig.
Grund für die immer wieder auftretenden Spitzen bei den Abfragezeiten sind interne Prozesse zur Datenbankverwaltung und zur Überwachung und Steuerung von Oracle Ultra Search.

Im Gegensatz zur Crawlerausführung (Indexerstellung) war bei der Suchabfrage für die hier getesteten Dokumentanzahlen (maximal ca. 30.000 pro Index) kein direkter Zusammenhang zwischen der Indexgröße und der Abfragezeit zu erkennen.
Die Anzahl gefundener Treffer hat dagegen - unabhängig von der Trefferausgabe - einen Einfluss auf die Abfragezeiten. Hier ist besonders bei geringen Trefferanzahlen ein starker Anstieg der Abfragezeiten zu erkennen. Bei großen Trefferanzahlen nimmt der Anstieg der Zeiten ab, so dass der Zusammenhang zwischen Abfragezeit und Trefferanzahl einer logarithmischen Funktion ähnelt. Dieses verhalten ist vermutlich auf die interne Speicherstruktur des Indexes zurückzuführen. Die Speicherung der Informationen zu den im Index

abgelegten Wörtern (wie z.B. die Verweise auf zugehörige Dokumente) wird mit Hilfe von Binary Large Objects [2] realisiert. Der Zugriff auf diese Objekte zum Ermitteln der Trefferdokumente könnte für das aufgetretene Verhalten der Abfragezeit verantwortlich sein. Ein Einfluss durch Caching (internes Zwischenspeichern) der Indexinformationen ist ebenfalls denkbar.

Der Umfang der zu einem Treffer angezeigten Informationen, wie z.B. das Anzeigen eines kurzen Dokumentauszugs, hatte in den durchgeführten Zeitmessungen keine Auswirkung auf die Abfragezeit.
Dagegen führt das Erhöhen der Anzahl pro Seite ausgegebener Treffer zu einer deutlichen Verschlechterung der Abfragezeiten. Dies hängt damit zusammen, dass sich für die Ausgaben von mehr Treffern pro Seite der Aufwand zur Aufbereitung der Trefferliste erhöht. Die Anzahl dargestellter Treffer pro Seite sollte also im Hinblick auf die Performance und den Informationsgehalt sinnvoll gewählt werden.

Auch wenn die Verwendung von Platzhaltern nicht zwangsläufig zu einer schlechten Abfragezeit führt, kann die Platzhaltersuche besonders in Kombination mit den Abfrageoperatoren '+' und '-' die Abfragezeiten stark erhöhen. Ausschlaggebend für die Abfragezeit ist die Anzahl der Wörter, auf die der eingegebene Suchbegriff durch seine Platzhalter erweitert werden kann. Hierbei spielt die Positionen der Platzhalter und die für den Suchbegriff verwendeten Teilwörter bzw. Buchstaben eine wichtige Rolle. So ist die Suche nach Begriffen, die lediglich aus einem Buchstaben und einem Platzhalter bestehen (z.B. "B*") in der Regel sehr aufwendig und kann zu extrem hohen Abfragezeiten oder sogar Fehlern wie Timeouts führen. Der Einsatz von Platzhaltern sollte also wohl überlegt sein.

Bei kleinen Indexen (mit ca. 5.000 indexierten Dokumenten), konnte keine Auswirkung der in den Dokumenten verwendeten Sprache (hier wurden die Sprachen Deutsch und Englisch untersucht) festgestellt werden. Die Zeiten für einen deutschsprachigen und einen englischsprachigen Index unterschieden sich auch bei Verwendung sprachspezifischer Operatoren wie Fuzzy-Suche und Stemming nicht wesentlich voneinander. Allerdings wird vermutet, dass für größere Indexe und die Verwendung anderer Sprachen (z.B. Japanisch) durchaus ein Einfluss der Sprache auf die Abfragezeiten festzustellen ist.
Die Ähnlichkeitssuche mittels Fuzzy-Algorithmus verschlechterte im Gegensatz zur Wortstamm-Suche die Abfragezeiten bei kleinen Indexen leicht, auf einem größeren Index war bereits eine deutliche Zunahme der Abfragezeiten zu erkennen. Es ist davon auszugehen, dass bei sehr großen Dokumentanzahlen eine Fuzzy-Suche die Abfragezeiten stark verschlechtert und eventuell sogar zu Problemen führt, da der Aufwand für eine Fuzzy-Erweiterung eines Suchbegriffs von der Indexgröße abhängt. Daher ist abzuwägen, ob der Nutzen der Bereitstellung einer Fuzzy-Suche eventuelle Performance-Verluste recht-

[2] kurz: BLOB, verfügbarer Spaltentyp in einer Oracle Datenbank

fertigt.

Bei fast allen im Rahmen der Arbeit durchgeführten Zeitmessungen schnitt die Oracle Ultra Search - Installation auf dem Application Server gegenüber der Datenbank-Version - zum Teil sogar deutlich - schlechter ab. Dies ist angesichts der Tatsache, dass vor allem für größere Oracle Ultra Search-Anwendungen in der Regel der Application Server seitens Oracle empfohlen wird, zunächst verwunderlich. Da zum Zeitpunkt der Leistungsmessungen die neueste Version des Application Servers noch nicht zur Verfügung stand, wurden die Zeitmessungen auf einer älteren Version des Application Servers durchgeführt. Erste Zeitmessungen auf einem mittlerweile verfügbaren Application Server neuerer Version (vgl. hierzu Anhang D auf Seite 89) haben jedoch gezeigt, dass die schlechteren Abfragezeiten des im Rahmen der Arbeit eingesetzten Application Servers mit der verwendeten Version und nicht mit der Software und der Architektur des Application Servers zusammenhängen. Ausschlaggebend für die Performance von Oracle Ultra Search sind sowohl die Version der verwendeten Datenbank, als auch die verwendete Oracle Ultra Search Version. Die zu Grunde liegende Software-Plattform (Datenbanksystem oder Application Server) spielt eine untergeordnete Rolle.
Weiterhin ist zu beachten, dass alle Abfragen im Rahmen der Zeitmessung nacheinander ausgeführt wurden. Da die Anzahl der Abfragen bei der Suche innerhalb der Webseiten des Forschungszentrums eher gering ausfällt (es wurden ca. 200 Abfragen pro Tag registriert, was weniger als einer Abfrage pro Minute entspricht), spielt die Abfragezeit für viele gleichzeitig gestellte Abfragen hier keine große Rolle. Der Application Server ist jedoch vor allem für den Einsatz im Umfeld mit vielen Clients und einer Vielzahl zeitlich parallel gestellter Anfragen konzipiert. In anderen Szenarien kann daher durchaus der Einsatz von Oracle Ultra Search auf einem Application Server von Vorteil sein.

Insgesamt ergaben sich folgende Erkenntnisse aus den durchgeführten Leistungsmessungen:

- Wesentlich für die Performance von Suchanfragen mit Oracle Ultra Search sind die Version der zur Indexverwaltung verwendeten Datenbank sowie die verwendete Oracle Ultra Search Version. Die Software-Plattform (Datenbanksystem oder Application Server) spielt in kleineren Suchumgebungen wie der des Forschungszentrums eine untergeordnete Rolle.

- Auch der erhöhte Funktionsumfang, wie z.B. das Hervorheben eines Suchbegriffs innerhalb eines Dokumentauszugs in der Trefferliste, spricht für die Verwendung der aktuellsten Oracle Ultra Search Version. Die Ausgabe eines Dokumentauszugs wirkt sich nicht negativ auf die Abfragezeiten aus. Dagegen verschlechtert eine Erhöhung der Anzahl angezeigter Treffer pro Seite die Abfragezeiten deutlich. Bezüglich der Trefferanzahl sollte daher eine sinnvolle Einstellung der Trefferausgabe vorgenommen werden.

- Vor allem die Fuzzy-Suche sowie die Verwendung von Platzhaltern können gerade bei großen Indexen zu erheblicher Verschlechterung der Abfragezeiten führen und sollten daher nicht unüberlegt eingesetzt werden.

- Ein Unterschied bezüglich der Abfragezeiten auf einem deutschen und einem englischen Index konnte bei den untersuchten Indexgrößen nicht festgestellt werden. Bei weiteren Sprachen und größeren Indexen ist jedoch eine Auswirkung der Sprache, vor allem bei Verwendung sprachspezifischer Abfrageoperatoren, zu erwarten.

6.3 Ausblick

Ein ausführlicher Vergleich mit der aktuellsten Version des Application Servers könnte genaueren Aufschluss darüber geben, in wieweit die Software-Plattform im Bezug auf verschiedene Trefferlistendarstellungen und die Verwendung verschiedener Abfrageoperatoren eine Auswirkung auf die Abfragezeiten hat. Eine Bestätigung der aus den ersten, auf dem aktuellen Application Server durchgeführten Messungen, aufgestellten These, dass nicht die Software-Plattform, sondern die verwendeten Versionen von Oracle Ultra Search und der zu Grunde liegenden Datenbank ausschlaggebend sind, wäre hierbei von Interesse.

Bei den durchgeführten Messungen wurde jeweils mit der Basiskonfiguration und den empfohlenen Standard-Werten beider Installationen gearbeitet. Die Auswirkung verschiedener Parameter-Einstellung und Optimierungsmaßnahmen der beiden Insallationsvarianten wurde bisher nicht geprüft und verglichen.
Im Rahmen der Zeitmessungen wurden reine Volltextsuchen ohne Einbeziehung von Metadaten durchgeführt. Oracle Ultra Search bietet die Möglichkeit, zusätzlich eine Einschränkung der Suche über Metadaten aus strukturierten Texten (z.B. HTML- oder PDF-Dokumenten) durchzuführen. Die Metadaten werden durch den Crawler gesammelt und in Form von Datenbanktabellen abgelegt, so dass sie später in die Suche einbezogen werden können. Auf diese Weise kann eine Suche nach Dokumenten eines bestimmten Autors oder in einer bestimmten Sprache durch die Kombination der Volltextsuche mit einer Metadatensuche durchgeführt werden. Ob die zusätzliche Verwendung von Metadaten bei der Volltextsuche sich auf die Abfragezeit auswirkt, könnte in weiteren Zeitmessungen festgestellt werden.

Weiterhin wäre ein Vergleich zwischen den Abfragezeiten über die Oracle Ultra Search-Oberfläche und den Zeiten für eine direkte Oracle-Text Abfrage mittels SQL-Statements von Interesse. Jedoch ist zu berücksichtigen, dass die Durchführung einer Volltextsuche mittels SQL-Statements im Praxiseinsatz nicht realistisch erscheint und diese Verwendung der Oracle Text - Funktionen ein anderes Einsatzgebiet beschreibt. Soll eine Voll-

textsuche vielen Anwendern zur Verfügung stehen, ist eine komfortable Benutzeroberfläche unumgänglich.
Es könnte geprüft werden, ob die Ausführung der Suche in Oracle Ultra Search über JavaServer Pages und Java Servelets sich negativ auf die Abfragezeiten auswirkt. Ein Vergleich mit der Einbettung von Oracle Text - Funktionalität in eigene Oberflächen, die sich z.B. in Skriptsprachen wie PHP, TCL, PL/SQL oder PERL implementieren lassen, wäre von Interesse.

Ein genauer Vergleich zwischen der Verwendung von Oracle Ultra Search und Oracle Text, bei dem auch die verschiedenen Einsatzgebiete beider Produkte genauer beleuchtet werden, könnte Thema einer weiteren Arbeit sein.

Anhang A

Beispielabfragen mit SQL-Statements

Das folgende Beispiel zeigt die Abfrage eines mit Oralce Ultra Search ertellten Indexes durch ein SQL-Statement:

```
SQL> SELECT SCORE(1), d.URL_ID, u.URL
  2  FROM WK$DOC d, WK$URL u
  3  WHERE CONTAINS (d.cache_file_path, 'Funktionsparameter', 1) > 0
  4  AND d.URL_ID = u.URL_ID
  5* ORDER BY SCORE(1) DESC

SCORE(1) URL_ID URL
-------- ------ ----------------------------------------------------------------
      40   7984 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/functions.arguments.html

      27   7861 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/functions.html

      13  35297 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/zamdoc/vortraege/perl/perlkurs_part1.4up.pdf

      13  14158 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/function.pdf-get-value.html

      13  33905 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/nic-series/volume28/nic-series-band28.pdf

      13  34455 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/zamdoc/bhb/bhb-0140.pdf

      13   8097 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/language.oop5.reflection.html

      13   7986 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/functions.returning-values.html

      13  14152 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                e/others/PhpManual/german/html/function.pdf-get-parameter.html


9 rows selected.
```

Im folgenden Abfrage-Beispiel wird die Bewertung der Dokumente auf die Worthäufigkeit des Suchbegriffs beschränkt:

```
SQL> SELECT SCORE(1), d.URL_ID, u.URL
  2  FROM WK$DOC d, WK$URL u
  3  WHERE CONTAINS (d.cache_file_path,'
  4  <query>
  5      <textquery grammar="CONTEXT"> funktionsparameter </textquery>
  6      <score datatype="integer" algorithm="COUNT"/>
  7  </query>', 1) > 0
  8  AND
  9  d.URL_ID = u.URL_ID
 10  ORDER BY SCORE(1) DESC

SCORE(1)   URL_ID URL
---------- ------ --------------------------------------------------------------
         3   3088 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/functions.arguments.html

         2   2098 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/functions.html

         1  18481 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/zamdoc/vortraege/perl/perlkurs_part1.4up.pdf

         1   4565 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/nic-series/volume28/nic-series-band28.pdf

         1  21042 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/function.pdf-get-parameter.html

         1  21050 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/function.pdf-get-value.html

         1   3625 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/language.oop5.reflection.html

         1   3106 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/others/PhpManual/german/html/functions.returning-values.html

         1   3507 file://localhost/home/orafzjt2/dipl_file_source/Dipl_File_Sourc
                  e/zamdoc/bhb/bhb-0140.pdf


9 rows selected.
```

Anhang B

Porter-Stemming

Ein weit verbreiteter und frei verfügbarer Stemmer für die englische Sprache ist der ”Porter Stemming Algorithm” [1], dessen Ansätze zum Verständnis der Funktionsweise eines Stemmers hier kurz beschrieben werden:
Das Ziel des Stemmers ist die Verbesserung bei der Auffindung von Informationen (Information Retrieval). Die Grundidee besteht darin, dass Terme mit gleichem Wortstamm für gewöhnlich auch eine ähnliche Bedeutung besitzen und das Zusammenfassen dieser Terme zu einem einzigen Term (Wortstamm) die Größe und Komplexität der Informationssuche verringern kann. Das beste Kriterium für das Zurückführen zweier Wörter auf denselben Wortstamm ist laut Porter folgendes:

”Perhaps the best criterion for removing suffixes from two words W1 and W2 to produce a single stem S, is to say that we do so if there appears to be no difference between the two statements ’a document is about W1’ and ’a document is about W2.”

Da dies jedoch in der Praxis schwierig zu beurteilen und diese Entscheidung schwer zu automatisieren ist, wird beim Stemming ein sogenanntes ”Suffix-Stripping” durchgeführt. Das heißt, dass die einzelnen Wortendungen nach bestimmten Regeln entfernt werden, um den Stamm eines Wortes zu erhalten. Dazu wird eine explizite Liste mit Wortendungen verwendet, wobei zu jeder Wortendung (Suffix) ein Kriterium gehört (z.B. ob der Wortstamm einen Vokal enthält), nach dem entschieden wird, ob das Suffix entfernt werden soll. Problematisch bei dieser Vorgehensweise sind z.B. die Fälle, bei denen ein Suffix die Bedeutung des Wortes stark verändert, wie etwa bei den englischen Wörtern ”probe” (Sonde, Fühler, Untersuchung) und ”probate” (Testamentseröffnung). Hierbei kann es passieren, dass Wörter mit stark unterschiedlicher Bedeutung auf denselben Wortstamm reduziert werden. Allerdings treten diese Besonderheiten laut Porter recht selten auf, so dass sie in einem Stemming-Algorithmus nicht unbedingt berücksichtigt werden müssen.

[1] Porter Stemmer: entworfen von Martin Porter und beschrieben in [PORMF-80], s. auch: http://www.tartarus.org/ martin/PorterStemmer/

Der Porter-Stemmer besteht im Wesentlichen aus fünf Stufen, in der jeweils bestimmte Fälle und Wortformen betrachtet werden. Stufe 1 beschäftigt sich z.B. mit den Pluralen und dem Past Participle. Als Beispiel für die Vorgehensweise soll das Wort ”Generalizations” dienen:

Beispiel: Stemming des Wortes ”generalizations” mit dem Porter-Stemmer:

generalizations	→	generalization	(Stufe 1)
generalization	→	generalize	(Stufe 2)
generalize	→	general	(Stufe 3)
general	→	gener	(Stufe 4)
gener	bleibt	gener	(Stufe 5)

Anhang C

Beispiel für PageRank

Der Ansatz, die Referenzierung einer Seite für deren Bewertung zu berücksichtigen, wird z.B. vom derzeit wohl bekanntesten Suchmaschinen-Anbieter ”Google” verwendet und nach seinem Erfinder Lawrence Page als ”**PageRank**” bezeichnet. Die Berechnung des PageRank für eine bestimmte Webseite wird wie folgt beschrieben:

”*... We assume page A has pages T1...Tn which point to it (i.e., are citations). The parameter d is a damping factor which can be set between 0 and 1. We usually set d to 0.85. There are more details about d in the next section. Also C(A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows:*

$$PR(A) = (1-d) + d\ (PR(T1)/C(T1) + ... + PR(Tn)/C(Tn))$$

Note that the PageRanks form a probability distribution over web pages, so the sum of all web pages’ PageRanks will be one.

PageRank or PR(A) can be calculated using a simple iterative algorithm, and corresponds to the principal eigenvector of the normalized link matrix of the web ...”[1]

Der PageRank kann als Modell für das Benutzerverhalten eines Webseitenbesuchers verstanden werden. Dabei geht man davon aus, dass ein Websurfer bei einer zufälligen Startseite beginnt und sich dann willkührlich über Links zu anderen Seiten vorarbeitet ohne den ”Zurück-Button” zu verwenden. Der PageRank gibt in diesem Fall die Wahrscheinlichkeit einer Seite dafür an, dass der Websurfer bei seinem Rundgang auf diese Seite trifft. Für gewöhnlich stoppt der Webseitenbesucher hin und wieder seine ”Reise” über die Links der Webseiten und startet zufällig bei einer völlig anderen Seite. Dieses Verhalten wird durch den oben erwähnten Dämpfungsfaktor *d* umgesetzt. Als einfaches Beispiel

[1] vgl. [PAGEL-98] Kapitel 2.1.1

für den PageRank nach obiger Formel soll folgendes Szenario dienen:

Es seien zehn Webseiten (Seite 1 ... Seite 10), wie in obiger Tabelle dargestellt, miteinander verlinkt (Tabelle[i][j] = x bedeutet, dass Seite i auf Seite j verweist):

Webseite	1	2	3	4	5	6	7	8	9	10
1	x		x	x	x			x		x
2		x		x	x	x	x	x		x
3			x	x					x	
4	x	x	x	x	x	x		x		x
5	x	x		x	x		x	x		x
6	x			x	x	x				
7		x	x	x		x	x	x	x	x
8	x			x	x	x		x	x	
9	x			x	x			x	x	x
10	x	x	x	x		x		x	x	x

Tabelle C.1: Beispiel zur Verlinkung von 10 Webseiten

Somit ergibt sich folgende Situation bezüglich ausgehender (eine Seite verweist auf eine andere Seite oder sich selbst) und eingehender (auf die entsprechende Seite wird von einer anderen Seite oder durch die Seite selbst verwiesen) Links:

Webseite	ausgehende Links	eingehende Links
Seite 1	6	7
Seite 2	7	5
Seite 3	3	5
Seite 4	8	10
Seite 5	7	7
Seite 6	4	6
Seite 7	8	3
Seite 8	6	8
Seite 9	6	5
Seite 10	8	7

Tabelle C.2: Anzahl eingehender und ausgehender Links der verlinkten Webseiten

Die Berechnung der Bewertungszahl (pagerank) für die einzelnen Webseiten wurde nach dem oben beschriebenen Algorithmus mittels folgender Formel durchgeführt:

$$R_i = \frac{1-d}{n} + d * \sum_{ji} \frac{R_j}{C_j}$$

Der Dämpfungsfaktor d beträgt 0.85, die Anzahl n der Webseiten ist 10. In der Summe am Ende der Formel werden alle Seiten j berücksichtigt, die auf die Seite i verweisen. Nach 21 Iterationen wurde die Berechnung abgebrochen [2] und es ergaben sich folgende Bewertungenzahlen:

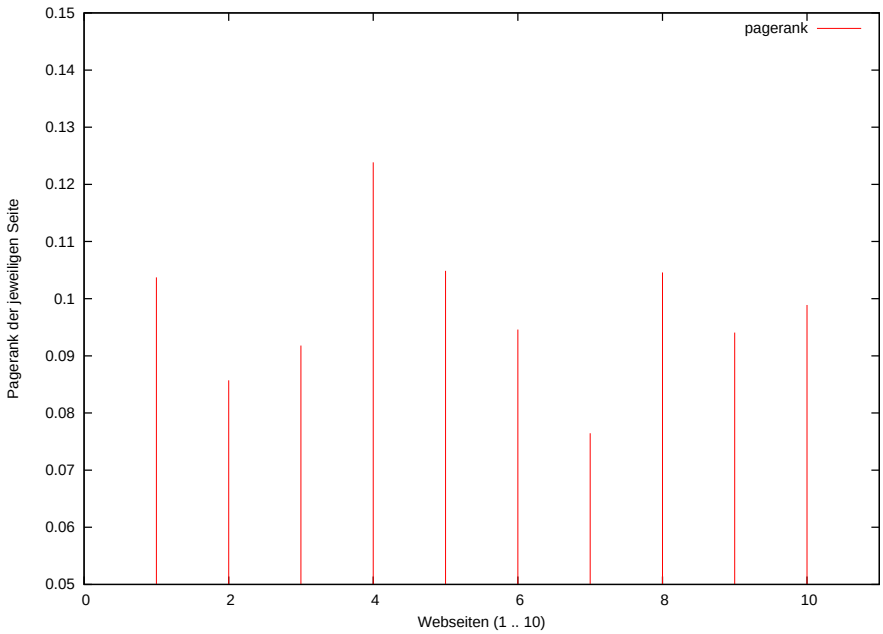

Abbildung C.1: Pageranking für 10 verlinkte Seiten

Deutlich zu erkennen ist die Abhängigkeit des Pagerank von der Anzahl eingehender Links. So erhält die Seite mit den wenigsten eingehenden Links (Seite 7) den niedrigsten pagerank und die Seite mit den meisten eingehenden Links (Seite 4) die höchste Bewertung.

Der in Wirklichkeit von Google verwendete Algorithmus zur Seitenbewertung ist natürlich wesentlich komplizierter "... und so geheim wie das Rezept von Coca-Cola" [3].
Tatsächlich ist die Seitenbewertung und deren Geheimhaltung bei großen Internet-Suchmaschinen von enormer Bedeutung: wird der Algorithmus durchschaut, kann die Qualität der

[2] die Berechnung wurde durch ein C-Programm unter Verwendung des Datentyps 'double' realisiert; die Iteration wurde abgebrochen, nachdem alle Werte im Vergleich zu ihren Vorgängern keine Änderung mehr aufwiesen

[3] vgl. [WELT-05]

Trefferlisten auf Grund von Manipulationen der Seitenbewertung nicht mehr gewährleistet werden und die Suchmaschine den Zweck der Erleichterung beim Auffinden nüztlicher Informationen nicht mehr erfüllen. Ein regelrechter Wettbewerb hat sich darin entwickelt, auf die Schliche des Google-Algorithmus zu kommen. Sogar eine neue Berufsbezeichnung ist im Zuge der großen Popularität von Internetsuchmaschinen entstanden: der ”Suchmaschinen-Optimierer”. [4]

[4] vgl. [WIRTW-05]

Anhang D

Application Server Release 2

Die folgenden Abbildung zeigt die Abfragezeiten für 65 Beispielabfragen auf drei Testsystemen:

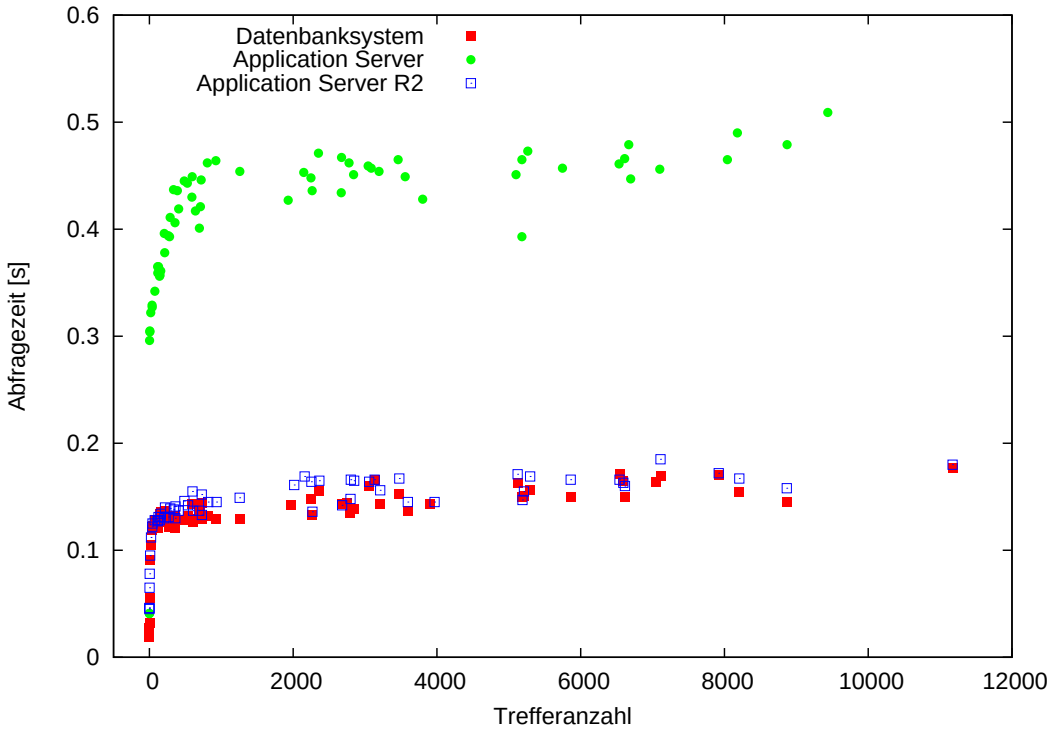

Abbildung D.1: Abfragezeiten für 65 Beispielabfragen in Abhängigkeit der Trefferanzahl auf **drei Testsystemen**

Neben den bereits aus Kapitel 5 bekannten Testsystemen wurde zusätzlich eine Zeitmessung mit der neuesten Version des Application Servers (Oracle Application Server 10g

Release 2) durchgeführt. Die intern verwendete Datenbank sowie Oracle Ultra Search liegen hierbei in der Version 10.1.0.4.2 vor. Als Beispielabfragen wurden die 65 Suchanfragen aus der Untersuchung der Trefferanzahl (vgl. Kapitel 5.3) verwendet.

Es ist deutlich zu erkennen, dass sich die Zeiten auf dem Application Server 10g Release 2 im Gegensatz zur Vorgängerversion (Release 1) kaum vom Datenbanksystem unterscheiden.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[DIPLN-02] Jose Matas Nobis (2002)
Entwicklung einer Suchmaschine unter Verwendung von Oracle 9iAS Portal und Oracle interMedia Text
Diplomarbeit, Fachhochschule Köln, Fachhochschule Dortmund

[DIPLW-03] Jörg Waitelonis (2003)
Google - PageRank
Studienarbeit, Institut für Informatik Universität Jena

[HOPMA-05] Manfred Hoppe (2005)
Dokumentensuche und vieles mehr mit Oracle Text
Vortrag im Rahmen der ORACLE-Anwenderkonferenz

[KAEHE-04] Heiko Käppler (2004)
Oracle Text
Vorlesung zum Thema Multimedia-Datenbanken an der TU Cottbus
http://dbis.informatik.tu-cottbus.de/data/pub/skript/mmdb_ws0405/MMDB_oracle_text.pdf

[ORATA-05] Oracle (Juni 2005)
Oracle Text Application Developer's Guide 10g Release 2 (10.2)
http://tahiti.oracle.com/ (Part No. B14217-01)

[ORATR-05] Oracle (Juni 2005)
Oracle Text Reference 10g Release 2 (10.2)
http://tahiti.oracle.com/ (Part No. B14218-01)

[ORAUS-05] Oracle (Juni 2005)
Oracle Ultra Search Administrator's Guide 10g Release 2 (10.2)
http://tahiti.oracle.com/ (Part No. B14222-01)

[PAGEL-98] Sergey Brin and Lawrence Page (1998)
The Anatomy of a Large-Scale Hypertextual Web Search Engine
http://www-db.stanford.edu/ backrub/google.html

[PORMF-80] C.J. van Rijsbergen, S.E. Robertson and M.F. Porter (1980)
New models in probabilistic information retrieval
London: British Library. (British Library Research and Development Report, no. 5587).

[ROBOT] Robots Exclusion *http://www.robotstxt.org/wc/exclulsion.html*

[SALG-83] G. Salton, M. J. MacGill (1983)
Introduction to modern information retrieval
Hamburg : MacGraw Hill , 1983 (ISBN: 3-89028-051-X)

[WELT-05] ”Die Welt”, Artikel vom 24.11.2005, S. 31
Wie Google die richtigen Treffer findet

[WIKIP] Wikipedia
http://wikipedia.org

[WIRTW-05] ”Wirtschaftswoche”, Artikel vom 20.10.2005, S. 177
Suchmaschinen-Optimierer machen Web-Seiten im Internet auffindbar. Das Jobprofil eins neuen, gefragten Berufes.