Konzept für die Implementierung einer Online-Suche auf einem großen Datenbestand

Inhaltsverzeichnis

A Aufgabenstellung & Ziel

Ziel ist es, eine **fehlertolerante**, **performante und synchronisierte Online-Suche** auf einem derzeitigen **Gesamtbestand von mehreren Millionen Adressen** bereitzustellen, wobei die erforderliche Hard- und Software den Erfordernissen angepasst werden kann und derzeit nicht vorgegeben ist.

Fehlertolerant heißt in diesem Zusammenhang, dass auch bei geringfügig fehlerhafte Dateneingaben in allen Feldern dennoch die korrespondierenden Datensätze im zugrundeliegenden Gesamtbestand gefunden werden sollen. Neben der Zielvorgabe, alle gesuchten Datensätze zu finden, soll darüber hinaus jeglicher Overkill vermieden werden, d. h. es sollen möglichst keine Datensätze gefunden werden, die nicht den Sucheingaben entsprechen.

Performant heißt, dass die durchschnittliche Suchdauer 1 Sekunde nicht wesentlich überschreiten sollte.

Synchron heißt hier, dass eventuelle Änderungen, Löschungen und Neuzugänge im Gesamtbestand zeitnah in der Online-Suche berücksichtigt werden müssen, d. h. wenn z. B. eine neue Adresse im Gesamtbestand eingetragen wurde, muss eine anschließende Suche nach dieser Adresse diese auffinden.

B Allgemeine Hard- und Softwareanforderungen

B.1 RDBMS (Datenbanksystem)

Aufgrund der sehr hohen Datenmenge des Gesamtbestands und der Anforderung einer funktionierenden, zeitnahen Synchronisation empfehlen wir dringendst den Einsatz einer Datenbank, idealerweise Oracle 9.x. Der Einsatz einer Datenbank ist keine Voraussetzung für den später diskutierten Einsatz der AS Software AS Match(ing)Box, sollte allerdings aus protokollarischen Sicherheitsgründen erfolgen.

B.2 Hardware/Betriebssystem

Zu empfehlen ist hier vernünftigerweise ein leistungsstarker Mehrprozessorrechner mit entsprechender Ausstattung hinsichtlich RAM-Speicher, performanter Festplatten etc. Allerdings wäre auch eine entsprechend gut ausgestattete Windows oder Linux Systemumgebung denkbar. Ein **ausreichend großes RAM-Speichervolumen** ist zu empfehlen.

C Technische Umsetzung

C.1 Grundgedanke

Das Softwareprodukt AS Match(ing)Box beinhaltet alle notwendigen Funktionalitäten, die für die Implementierung einer performanten Online-Suche mit fehlertoleranten Mechanismen notwendig sind. Neben den Werkzeugen zum Aufbau einer Suchdatei/Suchbestandes (vgl. Produktbeschreibung zur AS Match(ing)Box) werden vorgefertigte Suchfunktionen bereitgestellt, die diesen Anforderungen gerecht werden und durch frei definierbare Parametrisierungen den speziellen Kundenwünschen

angepasst und entsprechend optimiert werden können. So können bspw. Häufigkeiten berücksichtigt werden, mit denen bestimmte Begriffe oder Namen im Adressdatenbestand auftauchen. (Es macht z.B. wenig Sinn nach dem Namen „Schmidt“ im gesamten Verzeichnis zu suchen – der Name „Kleinbrahm“ könnte hingegen schon ausreichend identifizierend sein!)

Bevor eine Online-Suche stattfinden kann, muss ein entsprechender Suchbestand – auch Suchdatei genannt – generiert werden. Diese wird die aus den analysierten Originaldaten des Gesamtbestandes generiert und außerhalb der Datenbank auf Betriebssystemebene abgelegt. Im Falle einer Oracle Implementierung geschieht dies sozusagen „auf Knopfdruck“.

Die Vorgehensweise der Verwendung einer Suchdatei außerhalb jeder Datenbank hat ganz entscheidende Vorteile, u. a. die folgenden:

1. Die Originaldaten werden nicht modifiziert oder erweitert, da nur ein Abbild derselben erstellt und auf diesem gesucht wird.
2. Die Suchdatei enthält nur die standardisierten, analysierten und komprimierten Informationen, auf denen letztendlich gesucht werden soll. Dadurch ist die Suchdatei um ein Vielfaches kleiner als die Originaldatei und die anschließende Suche ist enorm performant.
3. Die Suchdatei ist völlig unabhängig von der eigentlichen Quelle der Originaldaten, es ist also unerheblich, ob die Originaldaten aus einer Datenbank oder einer oder mehrerer Dateien kommen.
4. Die Suchdatei wird auf Betriebssystemebene durchsucht, d. h. es wird möglichst viel Datenbank-Overhead vermieden.

Die sogenannt Suchdatei wird normalerweise initial einmal zu Beginn des Produktionseinsatzes erzeugt, geeignet indiziert und dann durch die zugehörigen Synchronisationsfunktionen up-to-date gehalten. Es werden auf einer **1000 Mhz. Maschine ca. 2.000 Datensätze/Sekunde analysiert und interpretiert.** Unter der Voraussetzung, dass der Gesamtbestand in zehn etwa gleichgroße Teile aufgesplittet wird, dauert der **Aufbau jeder Suchdatei (~ 4.000.000 Adressen) ca. 0,5 Stunden**. Es soll noch einmal erwähnt werden, dass der Aufbau der Suchdatei nur einmal am Anfang erfolgt und dann durch Synchronisationsfunktionen aktualisiert wird. Es ist jedoch aus Performancegründen sinnvoll, in bestimmten Zeitabständen den gesamten Bestand noch einmal neu zu analysieren und die Suchdatei neu aufzubauen (z. B. alle 3 oder 6 Monate). Dieser Aufbau der Suchdatei muss selbstverständlich in einem Zeitfenster erfolgen, in dem die Anwendung nicht zur Verfügung steht (also z. B. am Wochenende oder an Feiertagen).

Der Platzbedarf einer Suchdatei entspricht ca. 400 MByte für 4.000.000 Adressen, also ca. 1 MByte für 1.000.000 Adressen.

Bei einem Bestand von 40.000.000 (oder mehr) Adressen empfiehlt sich aus Performancegründen eine 10-fach parallele Vorgehensweise, also ein paralleler Aufbau von 10 Suchdateien. Diese Parallelität wird weiter unten genauer beschrieben und erklärt.

C.2 Online-Suche auf Einprozessormaschinen

Die AS Match(ing)Box ist so organisiert, dass eine Suchanfrage trotz der Verwendung fehlertoleranter Verfahren (phonetische Suche) selbst in großen Datenbeständen kurze Antwortzeiten bietet, die im Durchschnitt nicht wesentlich mehr als etwa 1 Sek. betragen sollte. Natürlich sind zur Einhaltung dieser Antwortzeiten gewisse Grenzen vorgegeben, die z.B. durch das zu durchsuchende Datenvolumen oder den Grad der Fehlertoleranz und nicht zuletzt durch die Systemumgebung bestimmt werden. Erfahrungswerte mit großen Datenmengen in der Vergangenheit haben gezeigt, dass das oben definierte Ziel eine durchschnittlichen Antwortzeit von 1 Sek. für Datenmengen bis zu etwa 5-8 Millionen Records mit einer vernünftigen Fehlertoleranz auf Einprozessorsystemen erreichbar ist. Selbstverständlich ist auch die Suche in größeren Datenvolumina auf

Einprozessormaschinen möglich, jedoch ist zu bedenken, dass dies die Antwortzeit erhöhen und die Treffsicherheit beeinträchtigen kann.

Wird die AS Match(ing)Box – wie oben empfohlen – in einem Datenbankumfeld eingesetzt, so wird als Ergebnis einer Suchanfrage zunächst der/die Schlüssel (z.B. eindeutige Kundennummer) und der Grad der Übereinstimmung (0-100%) der Daten zurückgeliefert, die als mögliche Treffer identifiziert wurden. Über diesen Schlüssel können dann die gewünschten Originaldaten aus den hinterlegten Tabellen selektiert und z.B. zur Anzeige gebracht werden.

Die Suche selbst wird von einem AS Match(ing)Box Online Server durchgeführt (im folgenden Search-Server genannt), bei dem es sich um ein ausführbares Programm handelt, dass einerseits die Werkzeuge enthält, um auf die Suchdateien zugreifen zu können, andererseits die Verbindung zur (Oracle) Datenbank herstellt und mit dieser kommuniziert.

C.3 Online-Suche auf Mehrprozessormaschinen

Spätestens wenn die Datenmenge ein Volumen von ca. fünf Millionen Records übersteigt, ist die Implementierung einer parallelisierten Suche anzuraten. Hierbei wird eine Suchanfrage in mehrere Prozesse aufgesplittet, die jeweils disjunkte (sich nicht überschneidende Datenmengen) zu verarbeiten haben. Die Antwortzeit für eine Suchanfrage steigt hierbei entsprechend mit jedem Prozess, der für die Recherche verwendet wird. Da die Parallelverarbeitung organisiert und verwaltet werden muss, steigt die Verarbeitungszeit nicht im gleichen Maße bei jedem zusätzlichen Prozess. Trotzdem kann man davon ausgehen, dass zehn parallele Prozesse die Antwort bis zu neun mal schneller liefern als ein einziger Prozess.

Ausgehend von einer Datenmenge von mehreren Millionen Records bietet sich die Aufteilung des Bestandes in 10 Einzelbereiche an, um somit unterhalb der oben angegebenen Menge zu liegen, die eine optimale, fehlertolerante Suche gewährleistet. Die Praxis hat gezeigt, das die erste Ziffer der PLZ ein geeignetes Mittel darstellt, eine entsprechende Unterteilung vorzunehmen. Wird unterstellt, dass der Datenbestand auch Datensätze enthält, die keine (gültige) PLZ enthält, sollte zusätzlich ein elfter Bestand generiert werden, in den diese Datensätze eingeordnet werden.

Es ergibt sich somit eine Konstellation für die Suchdateien der AS Match(ing)Box, die wie folgt dargestellt werden kann:

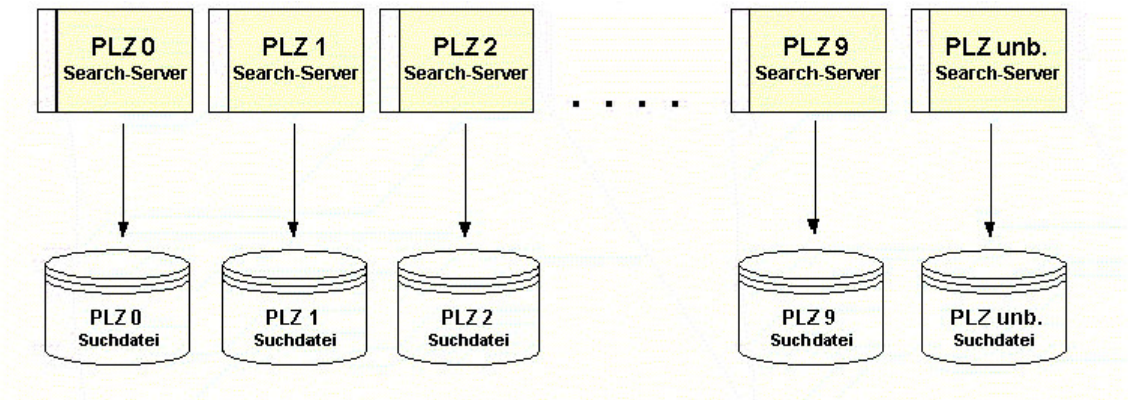

Bild 1 (PLZ-Bereiche mit Search-Server)

Jeder einzelne PLZ-Bereich kann hierbei an einen dedizierten Search-Server geknüpft werden, der betriebssystemseitig wiederum als Prozess auf einem Prozessor abläuft. Dadurch wird erreicht, dass bspw. nach einem Namen in allen PLZ-Bereichen gleichzeitig gesucht werden kann, wenn jegliche Angabe wie Ort, PLZ und/oder Straße, die eine Bestimmung des PLZ-Bereiches ermöglicht, fehlt.

Die somit gewonnene Möglichkeit der parallelen Suche erfordert ein zentrales Modul, das einerseits die Weiterleitung einer Suchanfrage and die Einzelprozesse je PLZ-Bereich weiterleitet und ebenso die von diesen zurückgelieferten Resultate wieder bündelt und als Gesamtergebnis zur Verfügung stellt. Diese Aufgabe übernimmt das Modul „AS Parallel Search Server“ (AS PSS).

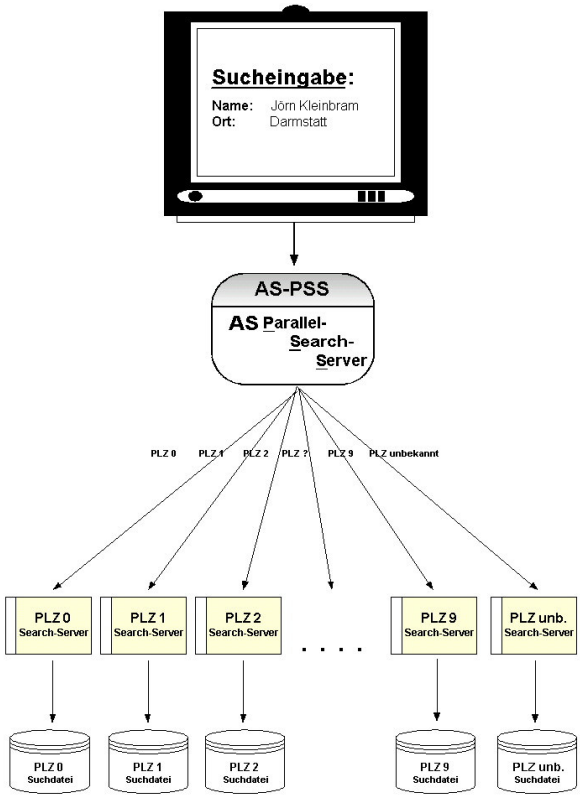

Bild 2 (Suchanfrage an alle PLZ-Bereiche)

Die Suche in jedem PLZ-Bereich liefert 0-n Resultate, die nach dem Grad der Übereinstimmung mit den Eingabedaten (Score) in absteigender Reihenfolge sortiert werden.

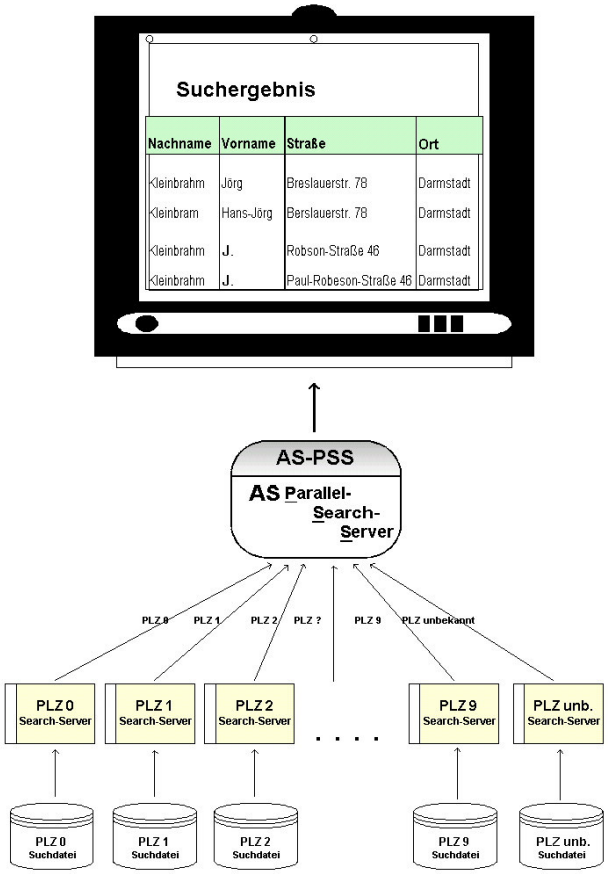

Bild 3 (Rückgabe der Treffer aus allen PLZ-Bereichen und Bündelung der Ergebnisse im AS PSS; die Gesamtergebnismenge wird sortiert nach absteigendem Score sortiert)

Im produktiven Betrieb findet aber eine Suche nach einem Namen allein ohne zusätzliche Information über den Ort o.ä. relativ selten statt. Im Regelfall liefert eine Suche nur nach einem Namen in großen Adressbeständen ohnehin zu viele Ergebnisse, um sofort den gewünschten Datensatz zu identifizieren. Üblicherweise wird daher z.B. nach einem Namen in einem bestimmten Ort gesucht, (z.B. „Herr Schröder in Hannover“). Seltener kommt es vor, dass PLZ oder Straße und Hausnummer bekannt sind. Abhängig davon kann aber die gewünschte Suche vor dem Verteilen an die einzelnen Search-Server je PLZ-Bereich sinnvoll eingeschränkt werden:

- Wird die PLZ oder wenigstens die erste Ziffer der PLZ bei der Suchanfrage übergeben, kann die Suche auf den entsprechenden PLZ-Bereich eingegrenzt werden.

- Ist der Ort bekannt und wird bei der Suchanfrage übergeben, so ermittelt der AS PSS zunächst die möglichen PLZ-Bereiche, in denen dieser Ortsname phonetisch auftritt, so dass die Suche auf diese PLZ-Bereiche eingegrenzt werden kann.

C.4 Gleichzeitige Suche in einem PLZ-Bereich

Bei einer entsprechend hohen Anzahl von Suchanfragen - wie dies in vorliegendem Projekt zu erwarten ist - muss davon ausgegangen werden, dass einer oder mehrere PLZ-Bereiche gleichzeitig durchsucht werden sollen. Um dieser Anforderung gerecht zu werden, können für jeden PLZ-Bereich mehrere Search-Server gestartet werden, die sich gegenseitig nicht behindern, obwohl sie auf den gleichen Suchdateien arbeiten. Dies kann deshalb gewährleistet werden, weil jeder Search Server nur lesend auf die Suchdatei zugreift und daher keine Deadlocks o.ä. auftreten können. Darüber hinaus benötigt ein wartender Search Server keinerlei Prozessorleistung, so dass bedenkenlos mehrere Server pro PLZ-Bereich gestartet werden können, auch wenn die Anzahl der laufenden Server Prozesse die Anzahl der verfügbaren Prozessoren deutlich überschreitet.

C.5 Online Synchronisation der Suchdateien

Insbesondere dann, wenn mehrere Personen mit dem gleichen Datenbestand arbeiten, ist eine zeitnahe Synchronisation der Suchdateien mit dem Originalbestand notwendig. Wird bspw. ein neuer Adressdatensatz eingefügt oder ein vorhandener modifiziert, dann muss dieser Datensatz im Datenbestand, auf dem eine Suche stattfindet (AS Suchdatei) ebenfalls aktualisiert werden. Hierfür wird der AS Synchronisations Server (AS Sync-Server) verwendet. Der Sync-Server wird über Änderungen im Originalbestand per Datenbank-Trigger informiert und führt die gewünschte Änderung in der Suchdatei durch. Da dieser Mechanismus sehr schnell reagiert, und auch nicht zu erwarten ist, dass tausende Änderungen im Originalbestand innerhalb weniger Sekunden auftreten, kann davon ausgegangen werden, dass der Suchbestand immer zeitgleich auf dem aktuellen Stand ist.

Einzige Restriktion in diesem Zusammenhang ist, dass nur ein Sync-Server je PLZ-Bereich gestartet werden kann, was aus technischer und fachlicher Sicht aber auch sinnvoll ist, da immer nur ein Prozess eine Suchdatei ändern und diese Änderungen abspeichern darf.

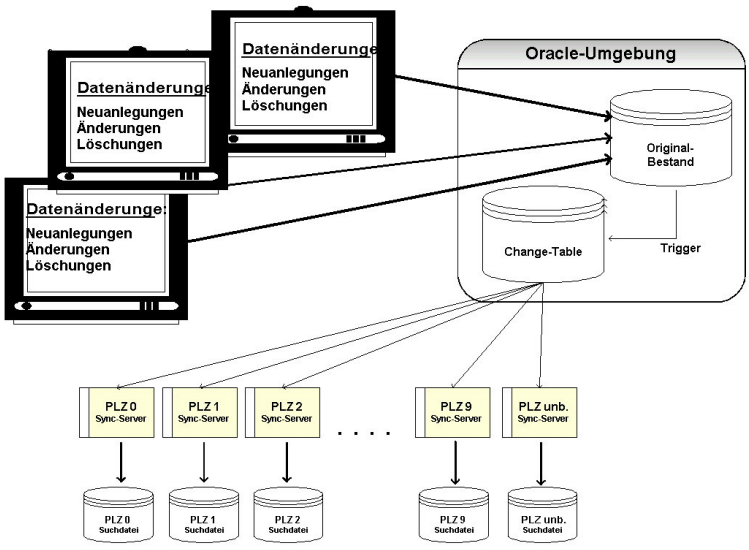

Bild 4 (Synchronisation)

D Fazit und Schlussbemerkungen

Mit den Software Werkzeugen AS Match(ing)Box und den zugehörigen Oracle Server Komponenten ist die gewünschte Aufgabenstellung einer performanten und fehlertoleranten Online-Suche im Rahmen einer Call-Center-Lösung problemlos möglich. Weitgehend vorgefertigte Systemfunktionen, die über Parametrisierungen den individuellen Wünschen angepasst werden können, liefern darüber hinaus die notwendige Flexibilität, die in Projekten derartiger Größenordnungen geboten sind. Hierzu zählt z.B. auch die Möglichkeit der Parallelisierung, die letztlich wiederum auch die bestmögliche Kombination von Qualität (Treffsicherheit) und Performance bedeutet. Der in Abschnitt B aufgezeigte Weg der Parallelisierung über die erste Ziffer der PLZ ist dabei nur eine von vielen Möglichkeiten, die aber nach unseren Erfahrungen oft die einfachste und praktikabelste Lösung ist. Denkbar sind z.B. auch Aufteilungen entsprechend der Anfangsbuchstaben des Ortes oder des Namens. Welcher Weg hier gewählt wird hängt natürlich in starkem Maße von den äußeren Rahmenbedingen ab, z.B. mit welchen Informationen normalerweise ein Datensatz gesucht wird etc.

Die AS Match(ing)Box ermöglicht neben der Suche in reinen Adressdaten (wie Name, Ort, Straße, etc.) auch die Recherche nach Telefonnummer, eMail-Adresse, Kontoverbindung, Geburtsdatum oder auch jeder anderen frei definierbaren Information. Sofern sinnvoll, können hierfür auch fehlertolerante Mechanismen angewendet und in den Suchvorgang einbezogen werden

Es kann durchaus vernünftig sein, in bestimmten Situationen ohne fehlertolerante Verfahren zu arbeiten. Dies ist z.B. dann gegeben, wenn der Name einer zu suchenden Person durch Angabe von Vor- und Nachname bereits ausreichend eindeutig ist. Der Name „Xaver Schmidt“ ohne Angabe weiterer Informationen stellt ein typisches Beispiel für eine derartige Situation dar. Wird hier mit fehlertoleranten Mechanismen gearbeitet, werden unter Umständen sehr viele Datenbank- und Vergleichsoperationen durchgeführt, die möglicherweise zu einer unnötig langen Laufzeit führen. Ursache ist in diesem Fall, dass nicht nur alle Personen mit dem Nachnamen „Schmidt“ im Gesamtbestand gesucht werden, sondern auch diejenigen, die den gleichen phonetischen Code besitzen („Schmitt“, „Schmied“, „Schmid“, ...). Die AS Match(ing)Box bietet auch Werkzeuge, mit denen Operationen durchgeführt werden können, die weniger fehlertolerant arbeiten und in diesen Fällen möglicherweise eingesetzt werden sollten. Führen diese dann nicht zum gewünschten Ergebnis, kann als nachfolgender Schritt dann immer noch die fehlertolerantere Suche gestartet werden.