Produktöversikt Boolware

SOFTWARE CORPORATION
http://www.softbool.com

Innehåll

1 Nyttan med Boolware

Boolware är ett sökverktyg specialiserat på att ge snabb återsökning, analys (textmining) samt beräkning i både strukturerad, ostrukturerad information – med direktuppdatering av index.

Boolware indexerar exempelvis textdata, numerisk data, RTF dokument, PDF dokument, Markup Language data m.fl. som lagrats i filsystemet, ”recordfiler” eller i relationsdatabaser (RDBMS) och används för att på ett effektivt sätt söka fram relevant information ur heterogena datakällor.

Installera och tala om för Boolware vilka datakällor som ska indexeras så sköter Boolware resten själv. Om du behöver, kan du finjustera genom att säga till Boolware vilka tabeller / kolumner som ska indexeras samt hur dessa ska indexeras.

Boolware kan mycket enkelt integreras i befintliga och nya tillämpningar med hjälp av våra standardiserade gränssnitt och ger betydligt kortare utvecklingstider genom att mycket funktionalitet redan finns färdigt. Genom enkla inställningsmöjligheter anpassar man Boolware till tillämpningen, och inte tvärt om.

Boolware kan med fördel användas av företag i såväl interna som externa tillämpningar, där man har något eller några av följande krav i sin tillämpning:

- krav på boolesk sökning (FIND/AND/OR/NOT/XOR)
- krav på utökad sökfunktionalitet (närord, fonetisk, synonym, fuzzy, tesaurus, rankning, viktning, mönsterigenkänning/mönstermatchning, trunkering etc.)
- krav på direktuppdatering
- krav på snabba prestanda
- krav på skalbarhet (kostnadseffektiv informationshantering)
- krav på att kunna klassificera informationen (automatisk kategorisering baserat på mönster)

Fördelar är bland annat:

- Direktuppdaterande index – alltid synkroniserat med datakällan!
- Mycket snabb indexering och sökning
- Relevans och precision genom utökad sökfunktionalitet och kombinationer av dessa:
    - Ord, sträng, numerisk, fritext (med eller utan ”joker”-tecken)
    - Närord, fonetisk (låter/liknar som), fuzzy, case, inom sökning, SET mm.
    - Grundformning (Stemming)
    - Synonymer, tesaurus, stopp-ord
    - Samsökning i flera databastabeller (join)
    - Mönsterigenkänning, mönstermatchning
    - Mycket snabb sortering (stigande/fallande)
    - Rankning (förekomst, frekvens samt poängsättning - ”scoring”)
    - Dynamisk rankning (bl.a. baserat på angivna söktermer och ordningsföljden på dessa)
    - Viktning (vid sökning och poängsättning – ”scoring”)
    - Statistik, beräkning och enklare multidimensionell analys
    - Visning och navigering av indextermer och indexsträngar
    - Duplikathantering vid sökning
- Skalbart
- API’er (funktionsbaserat, C, C++, C#, XML, JSON, .NET, Java, COM, PHP)

2 Exempel på tillämpningsområden.

Exempel 1 (Elektronisk handel): Ett exempel där systemet passar mycket bra är i en e-handelslösning där man kan använda systemets förmåga att hitta information, trots felstavningar och omkastade bokstäver, med hjälp av specifika inbyggda sökfunktioner som hanterar synonymer, tesaurus och undantag vid sökning.

Exempel 2 (Katalogsökning): Ett annat lämpligt användningsområde är att använda systemet för sökning i kataloginformation som exempelvis gula/vita sidorna, produktinformation etc. där systemets inbyggda funktioner för att ranka och sortera framsökta resultat på ett snabbt och effektivt gör helhetsupplevelsen för en slutanvändare mycket positiv.

Exempel 3 (Matchning): Ett ytterligare exempel på lämpligt användningsområde är att använda systemet, för att på ett snabbt och effektivt sätt, matcha, ”tvätta”, organisations/personnummersätta data med hjälp av systemets inbyggda och effektiva funktioner för framsökning, rankning och poängsättning (scoring).

Exempel 4 (Finansiell företags- och adressinformation):
Tänk dig att blixtsnabbt kunna söka fram ett företag inom en viss region, storlek, omsättning etc. och därefter på ett par sekunder kunna analysera (mönstermatcha) framsökt företags bolagsordning/verksamhetsbeskrivning mot hundratusentals- eller miljontals andra företag i systemet och få ett resultat presenterat i likhetsordning. Man kan även utföra statistik samt egna beräkningar på numerisk information i databasen.

Exempel 5 (Läkemedelsinformation, biverkningsrapporter etc.):
Att från en framsökt biverkningsrapport inom en viss kategori på ett par sekunder kunna analysera (mönstermatcha) denna biverkningsrapport mot hundratusentals- eller miljontals andra i databasen och få resultatet sorterat i likhetsordning.

Exempel 6 (Tidningsartiklar/Pressreleaser etc.): Ett annat exempel är att man, genom systemets mönsterigenkännings- och mönstermatchningsmekanismer, hittar liknande artiklar i en tidningsdatabas och/eller att man använder systemet för att kategorisera nya artiklar som läggs in i databasen.

Exempel 7 (Brottsinformation): Ett ytterligare exempel då systemets mönsterigenkännings- och mönstermatchningsmekanismer kan utnyttjas är, då man gjort en personprofil av ett kriminellt beteende och vill kunna matcha denna mot en databas med lagrade uppgifter om kriminella personer för att få fram de profiler som mest liknar denna.

3 Uppbyggnad Boolware

Boolware består av tre delar; Boolware Manager, Boolware Client(s), Boolware Index Server.

Boolware Index Server kopplas till den befintliga datakällan med hjälp av ett administrativt program; Boolware Manager, där samtliga i systemet ingående datakällor (recordfiler, filsystem och databaser) kopplas ihop med Boolware Index Server för indexering.

Posterna i de berörda datakällorna delas in i tabeller och kolumner för att passa sökning och presentation. Boolware Index Server tillvaratar befintliga datamodeller, och man har sedan möjlighet att ”fininställa” sökningen på kolumnnivå (fält) med hjälp av Boolware Manager.

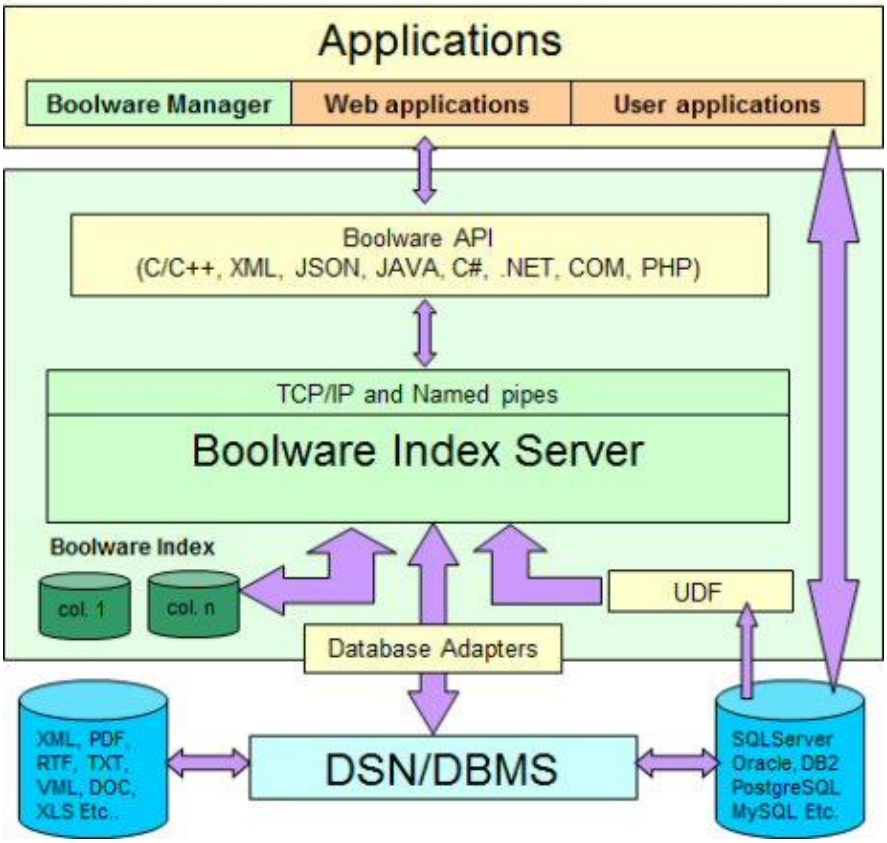
För att erhålla en effektiv sökning, skapas Boolware Index, vilka innehåller samtliga söktermer och referenser till alla de i respektive datakälla lagrade posterna. Innehållet i Boolware Index återspeglar därmed vid varje ögonblick aktuell information, lagrad i respektive datakälla.

Det normala förfarandet är, att datakällan används för att utföra uppdateringar; tillägg, ändringar och borttag, medan Boolware Index Server används för sökningar. Alla uppdateringar som görs i datakällan kommer automatiskt och momentant att uppdatera Boolware Index. Detta innebär, att uppdaterade poster omedelbart är sökbara på den nya informationen.

Boolware är ett verktyg, vilket levereras i form av flertal API: er vilka användaren kan använda för att skriva egna applikationer eller integrera funktionaliteten i befintliga applikationer.

För att testa systemet tillhandahåller Boolware en demonstrationsapplikation, vilken visar Boolwares funktionalitet samt också kan användas för vissa prestanda tester.

Nedan visas en schematisk skiss av systemet:



4 Funktionalitetslista

1. ***Hantering av datakällor via Boolware Manager och Boolware Index Server***

1.1 Koppling till datakällor (RDBMS, ”recordfiler”, filsystem)
1.2 Laddning av index samt möjlighet att i Boolware lagra kolumndata
1.3 Fininställning av sökning per Boolware Index (dvs. per kolumn/fält)
1.4 Optimering av Boolware Index (Reorganisation mm.)
1.5 Validering av Boolware Index
1.6 Uppdatering av Boolware Index
1.7 Automatkategorisering (klassificering av data) baserat på mönster i textdata
1.8 Språk/teckenhantering (på kolumn/fält nivå) samt stöd för UNICODE
1.9 Inställning för duplikathantering (på kolumn/fält nivå)

2. ***Sökning***

2.1 Boolesk sökning (FIND/AND/OR/NOT/XOR)
2.2 Viktad sökning (söktermer åsätts olika vikter vid sökning)
2.3 Samsökning mellan kolumner/tabeller samt RELATE sökning (JOIN via tabellrelationer)
2.4 Fonetisk sökning (flera olika algoritmer finns att välja på kolumnnivå)
2.5 Fuzzy sökning (strängdistanshantering)
2.6 Synonym sökning
2.7 Tesaurus sökning
2.8 Stoppords sökning
2.9 Grundforms sökning (stemming)
2.10 Numerisk sökning
2.11 Intervallsökning
2.12 Strukturerad sökning
2.13 Fritext sökning
2.14 Trunkeringar (höger, vänster samt “mitt i”, ? används för ett tecken, * för flera tecken i rad)
2.15 Närords sökning
2.16 Likhetssökning
2.16.1 Analys och sökning utförs baserat på mönster i textdata
2.16.2 Analys och sökning utförs baserat på mönster i numeriskt data
2.17 SET sökning
2.18 Duplikat sökning
2.19 Geo-sökning (på koordinater)
2.19.1 Polygon
2.19.2 Cirkulär
2.19.3 Rektangulär

3. *Sortering/rankning (sökresultat)*

3.1 Sortering avseende innehåll i kolumner (stigande/fallande)
3.2 Rankning avseende antal förekommande söktermer
3.3 Rankning avseende frekvens av antal förekommande söktermer
3.4 Viktad rankning avseende antal förekommande viktade söktermer samt
3.5 Viktad rankning avseende frekvens av antal förekommande viktade söktermer
3.6 Poängsättning (scoring) av framsökta poster
3.7 Dynamisk rankning (exempelvis baserat på angivna söktermer och dess angivna ordningsföljd i sökresultatet)

4. *Statistik & beräkning*

4.1 Statistik på numeriskt innehåll (summa, genomsnitt, median, varians, avvikelse, kvartiler, kvintiler etc.)
4.2 Beräkning på och mellan kolumndata med numeriskt innehåll
4.3 Rapportmallar (multidimensionell analys)

5. *Presentation av resultat*

5.1 Angivet antal tecken från specificerade kolumner
5.2 Hela raden/raderna (dvs. hela posten/posterna i sökresultatet)
5.3 Unikt identifikationsbegrepp (primärnyckeln för varje post)
5.4 Beräknade resultat
5.5 Statistik resultat
5.6 Rapportmallars resultat

5 Flöden och rankning

Flöden

För avancerad sökning använder sig Boolware av något som kallas för flöden.
Flöden är sökstrategier som utformas enligt de behov applikationen/tjänsten har.

Exempelvis kan en applikation som skall söka fram företagsnamn anropa ett flöde som utför sökningen mot datakällan enligt ett fördefinierat scenario.
Poängen med att använda sig av flödestekniken är att det låter utvecklaren definiera och strukturera en sökstrategi och hur träffarna i datakällan och resultatet skall presenteras.

I ett enkelt scenario, kan träffarna exempelvis genom definitionen att sökargument som ger exakt träff på företagsnamnet (sträng), skall visas först och därefter skall träffar där företagsnamnet innehåller orden från sökargumenten presenteras.

I ett mer komplicerat scenario kan flödet analysera en inkommande sträng med ord och göra olika sökningar i olika databaser/tabeller för att avgöra vad den inkommande söksträngen innehåller och genom kombination av olika sökningar presentera ett relevant sökresultat.

Flöden kan också användas till mer komplexa uppgifter så som att matcha register mot varandra eller att analysera inkommande sökargument för att ta reda på om användaren exempelvis har fyllt i namn, telefonnummer, ort etc. och därefter utföra sökningar i relevant källdata.

Flöden skapas med hjälp av ett scriptspråk och till hjälp finns en utvecklingsmiljö där flöden skrivs och kan testköras. Scriptspråket är kraftfullt och innehåller en mycket stor mängd funktionalitet för att exempelvis normalisera och jämföra strängar, trimma ord, backa i en sökstrategi vid 0 träff sätta rankscore på individuella träffar etc.

Flödets funktionsbibliotek kombinerat med Boolwares olika indexeringsmöjligheter gör att det går att åstadkomma mycket avancerade sökstrategier, som dels snabbt genererar relevanta svar till tjänsten/applikationen och dels avlastar och därmed snabbar upp applikationsutvecklingen.

Rankning

Rankning går att genomföra på ett flertal olika sätt, exempelvis:

1) Statisk rankning/sortering som sorterar/rankar och försorterar läst källdata för att presentera data i angiven ordning.

2) Rankning avseende antal förekommande söktermer samt frekvens av antal förekommande söktermer eller viktad förekomst av söktermer respektive viktad frekvens av förekommande söktermer.

3) Rankning via sökstrategi (se flöden ovan) och poängsättning (scoring) av framsökta poster.

4) Dynamisk rankning som används för att presentera träfflistor på ett för användaren så relevant sätt som möjligt. Exempelvis baserat på de ord (sökargument) som användaren skrivit in och dess angivna ordning. Det vill säga; ”man får de bästa träffarna först”.

Traditionellt så visas träfflistor i olika sorteringsordning baserad på olika fält, exempelvis företagsnamn, omsättning etc. Dessa träfflistor är därför statiska och ger en begränsad användarupplevelse eftersom det namn man sökt efter inte syns i början av träfflistan.

Men eftersom dynamisk rankning inte använder sig av statisk sortering av olika fält utan i realtid sorterar träfflistan baserat på de sökargument som angetts ger detta en mycket flexibelt och effektivt verktyg för att presentera relevanta träfflistor för användaren.

Tanken är att en träfflista med exempelvis företagsnamn skall presenteras med de använda sökargumenten först i träfflistan så att användaren snabbt hittar det han/hon sökt efter. Söker man efter ”Saab AB” skall detta visas före ”Ericsson SAAB AB”, detta låter sig inte göras med en traditionell sortering på företagsnamn men åstadkoms enkelt med dynamisk rankning.

Använder man den dynamiska rankningen på ett fält med personnamn och söker på ”Per Erik” så kommer alla namn som börjar på ”Per Erik” att visas först och namn som exempelvis ”Erik Per Anton”, ”Viktor Per Erik” etc. kommer längre ner i träfflistan.

Det finns många olika inställningar för att styra den dynamiska rankningen och den kan användas på ett eller flera fält.

6 Plattformar

Aktuella

Operativsystem (Boolware Index Server):

- Windows (2012 server eller senare)
- Linux (RHEL 6.7 eller senare)

Datakällor:

- MS SQL Server 2008 eller senare
- DB2 UDB 9 eller senare
- Oracle 10 eller senare
- Sybase ASE 15 eller senare
- MySQL 5 eller senare
- PostgreSQL 9 eller senare

- Recordfiler (en fil i CSV-format per tabell)
- Filsystemsfiler (dvs. innehållet i olika filer i, det för systemet synliga, filsystemet. Ex. Word, Excel, PDF etc.)

Stöd för andra operativsystem/datakällor kan utvecklas på beställning i projekt.