

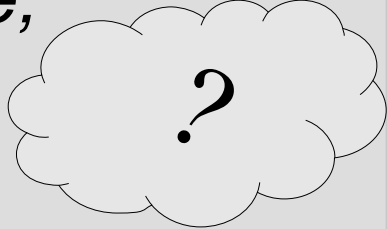


Lucene

USIT/SAUS seksjonsseminar
2005

Oversikt

- Introduksjon – *Hva er Lucene ? Hvorfor Lucene ?*
- Indekser og tekstsøk – *Grunnbegreper, terminologi*
- Bruksområder – *Hva omfattes av Lucene ?*
- Webapplikasjoner – *Innholdssøk med Lucene*
- API – *kodeeksempel*
- Eksisterende prosjekter som bruker Lucene
- Lucene i Vortex – *status quo, eksempel på tenkt funksjonalitet, egne erfaringer, teknisk*
- Del 2 (dag 2) - *Mer om Lucene+Vortex, Tavle, fremtidig bruk, tenkt funksjonalitet*
- Oppsummering og spørsmål



?

Introduksjon

- Hva er Lucene ?

- Motor for å bygge indekser over tekstlig innhold og utføre effektiv (og avansert) søking i disse indeksene
- Åpen kildekode (Apache prosjekt)
- Implementert i ren Java, distribueres som et javabibliotek (JAR)
- Finnes også implementasjoner i andre språk, som bruker samme type indeks (plattformuavhengig format)
- Håndterer kjernen av problemområdet tekst-indeksering/søking på en solid måte og begrenser seg til dette.
- Noe Lucene ikke er: en fullblods web-søkemotor med indeksering/datainnhenting(crawling) og presentasjon.
- Men: kan brukes som en sentral komponent i en slik applikasjon.

Introduksjon (2)

- Hvorfor Lucene ? Noen fordeler:
 - God ytelse på søking i- og oppdatering av indekser
 - Rikt API for oppbygning av komplekse programmatisk søk
 - Støtte for avanserte søk, f.eks. fuzzy queries (søk på ordlikhet), rangering/vekting, ordstemming, filtrering ..
 - Inkluderer parser for tekstsøk
 - Skalerbart
 - Relativt lett å ta i bruk i eksisterende applikasjoner (men ikke direkte Plug'n'Play)
 - Utvidbart API, lett å legge til egen spesialisert funksjonalitet ved behov
 - Kan indeksere alt som kan gjøres om til tekst
 - Ingen lisenskostnader og åpen kildekode

Viktige fakta



Doug Cutting – *mannen som har laget Lucene*

... og bak enhver
fremgangsrik mann
står det en kvinne .. (?)



.. I dette tilfellet er det hvertfall
opphavsmannens kone, med mellomnavn
«Lucene».

Grunnbegreper

- Indeks

- Datastruktur for effektiv fremhenting av dokumentpekere basert på dokumentenes innhold (invertert indeks)
- Typisk scenario: «Gi meg pekere til alle dokumenter som inneholder ordene A og B, sortert på feltet tittel»
- Men også: «Gi meg dokumentet med den eksakte tittelen 'Foo Bar'».
- Effektiv sortering av resultat

- Forekomster i en indeks

- Dokumentet (typisk mappet til en instans i den virkelige verden)
 - Består av felter (*tittel, forfatter, URL, innhold*)
 - Feltene beskriver semantisk dokumentets deler
 - Bestemmer selv hva slags felter man ønsker
 - Feltene i et dokument består av en eller flere termer

Grunnbegreper (2)

- Term

- Minste indekserbare enhet, består av {felt:tekststreng}
- Typisk blir dokumentinnhold splittet opp i termer/ord på mellomrom (tokenizing/analyzing) før indeksering
- Søk blant alle termer innenfor «contents» feltet:
«contents:foo OR contents:bar»
 - Vil gi alle dokumenter som har forekomster av ordene «foo» eller «bar» innefor «contents»-feltet.

Grunnbegreper (3)

- Indeksering

- Datahenting (http://, file://, DBMS, ...)*
- Parsing av dokumenttype (.doc, .xml, .html, .pdf, osv.)*
 - Gir deg det tekstlige innholdet i feltene du ønsker å indeksere (tittel, innhold osv.)
 - Kan ha forskjellige typer dokumenter i samme indeks: mappe dokumentfelter til et felles sett med Lucene indeks felter.
- Send dataene til Lucene, bygger/oppdaterer indeksen
- Lagre indeksen fysisk (typisk til disk, eller også database)
- Inkrementell oppdatering

*Ikke Lucene-funksjonalitet

Grunnbegreper (4)

- Spørring

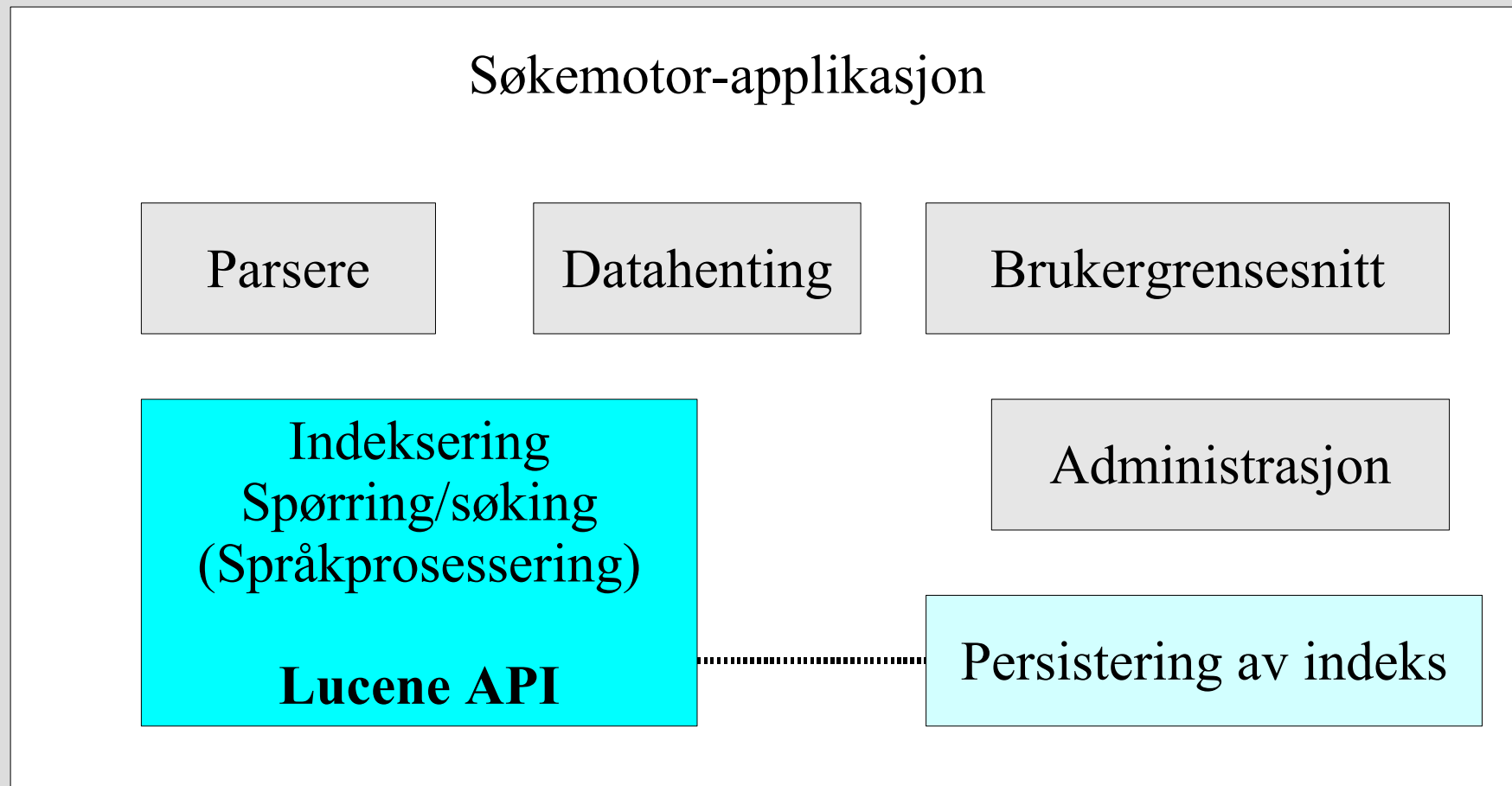
- Parsere søkestrengen og bygge opp klausulene:
 - Eks: «title:'A Document' AND 'red'»
- Bruke indeks for å finne dokumentforekomster som tilfredsstiller alle kravene
- Returnere treffliste (rangert, beste treff først)
- Programmatisk søk

- Ranging

- Rekkefølgen på trefflisten
- Hva havner på toppen ? Default: dokumentet med flest forekomster av en gitt term
- Vekting av dokumentfelter (boosting)
- Kan implementere egne klasser for evaluering av score for et gitt dokument i en spørring

Hva omfatter Lucene ?

- I forhold til fullstendig søkemotor:



Språkprosessering

- Bearbeiding av dokumentinnhold
 - Skjer før indeksering
 - Splitting/tokenizing av innhold til ord/termer
- Stoppord
 - Fjerne ord som ikke er interessante for søking i tekst
 - Eks: «en», «ei», «hvor», «hvem», «hva», «det»
 - Unødvendig å indeksere, gir liten verdi
 - Finnes standardimplementasjon i Lucene, men ikke spesifikk støtte for norske stoppord.
 - Men: API tillater å lage egne Analyser-implementasjoner.
 - Finnes for bl.a. tysk og russisk
- Standard funksjonalitet
 - Gjøre om alle ord til lowercase, fjerne komma, apostrofer, fjerne engelske stoppord

Språkprosessering(2)

- Stemming
 - Finne grunnstammen til ord
 - Eks: «Bilene» -> stem-filter -> «bil»
 - Utføres både ved indeksering og søking (om ønsket)
 - Porter-stem filter (stemming av engelske ord)
 - Finnes også støtte for tysk og russisk
 - Øke treffkvaliteten

Webapplikasjoner – integrere Lucene ?

- Enkelt fulltekstsøk i eksisterende web-applikasjon, hva må kodes (grovt sett) ?
 - Innhenting av data for dokumentene det skal søkes i, f.eks. henting av publiserte dokumenter over http:// + HTML parsing.
 - Typisk lage kode som henter ut HTML dokumenter og lager Lucene Documents med de ønskede feltene.
 - Kode som bruker Lucene for å lage en indeks og holde den oppdatert (om nødvendig)
 - Kode som henter søkeinput fra bruker, utfører søk med Lucene på indeksen, og presenterer resultatet, typisk i form av en treffliste med dokument URL'er
 - Evt. kode som utfører predefinerte programmatisk dokumenter søk, f.eks. for generering av menylinker, eller et spesialisert «forfatter-søk».

Lucene API

- Enkelt kodeeksempel
- Viser hvordan man indekserer dokumenter, for deretter å utføre et søk på den genererte indeksen
- .. også demo-kode i Lucene-distribusjonen.

Dokumentasjon

- Javadoc, FAQs, mailingliste-arkiver
- Kodeeksempler
- Diverse artikler på internett (Google)
- Bok: <http://www.lucenebook.com/>

Lucene / Vortex

- Hva finnes i dag ?
 - Prototype-implementasjonen av Tavle-prosjektet bruker Lucene (blir vist i morgen)
- Eksempel på tenkt funksjonalitet:
 - Erstatte batch-genererte mapping filer i Vortex (inngangsveier) med programmatisk oppslag/spørringer i Lucene-indeks
 - Indeks holdes automatisk oppdatert i tråd med endringer som skjer i Vortex-dokumenter
 - Integrert del av Vortex, fjerne avhengighet til eksterne script
 - Støtte et fast sett med metadata som indekseres for alle dokumenter
 - Utvide med flere indekser for å understøtte andre behov
 - Også mer om dette i morgen

Lucene / Vortex (2)

- Egne erfaringer med bruk av Lucene
 - Greit API å forholde seg til
 - Men endel ting å passe på som Lucene ikke håndterer 100% grasiøst:
 - Ingen funksjoner for å oppdatere et enkelt dokument i en indeks, må slette gammelt dokument fra indeks manuelt, for deretter å legge til ny oppdatert utgave
 - Multithreading og skriveoperasjoner på indeks, kontroll av tilgang
 - Batche oppdateringer på indeks sammen for å oppnå effektivitet: må sortere slettinger+addisjoner hver for seg, da disse to operasjonene er mutually exclusive
 - Alternativt: full reindeksering ved jevne mellomrom (ikke effektivt, men man unngår da mye av problemene over)

Lucene i den virkelige verden

- <http://www.theserverside.com/>
- <http://lucene.apache.org/nutch/>
 - Open source web søkemotor
- <http://www.searchblox.com/>
 - Kommersiell innholdssøkemotor
- <http://www.usajobs.org/>
- Lucene's egne skryteside:
 - <http://wiki.apache.org/jakarta-lucene/PoweredBy>

SLUTT

- Diverse ressurser
 - <http://lucene.apache.org/>
 - <http://wiki.apache.org/jakarta-lucene/LuceneFAQ>
 - <http://www.javaranch.com/newsletter/200404/Lucene.htm>
 - <http://vortex.uio.no/brukere/oyviste/lucene-presentasjon.pdf>
- Spørre-minutter (hvis det er noen igjen) ...