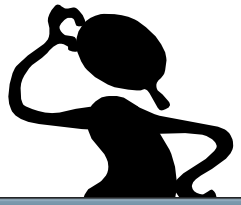
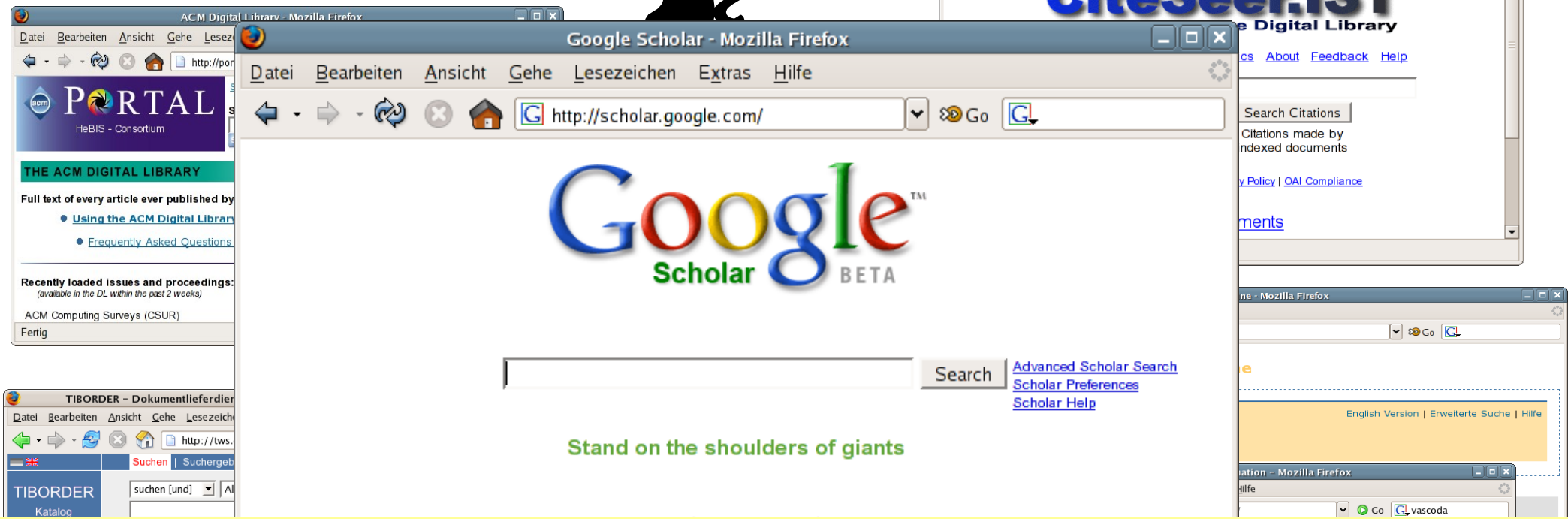
Federated Search: Integration von FAST DataSearch und Lucene

Christian Kohlschütter
L3S Research Center

BSZ/KOBV-Workshop, Stuttgart
24. Januar 2006

Motivation

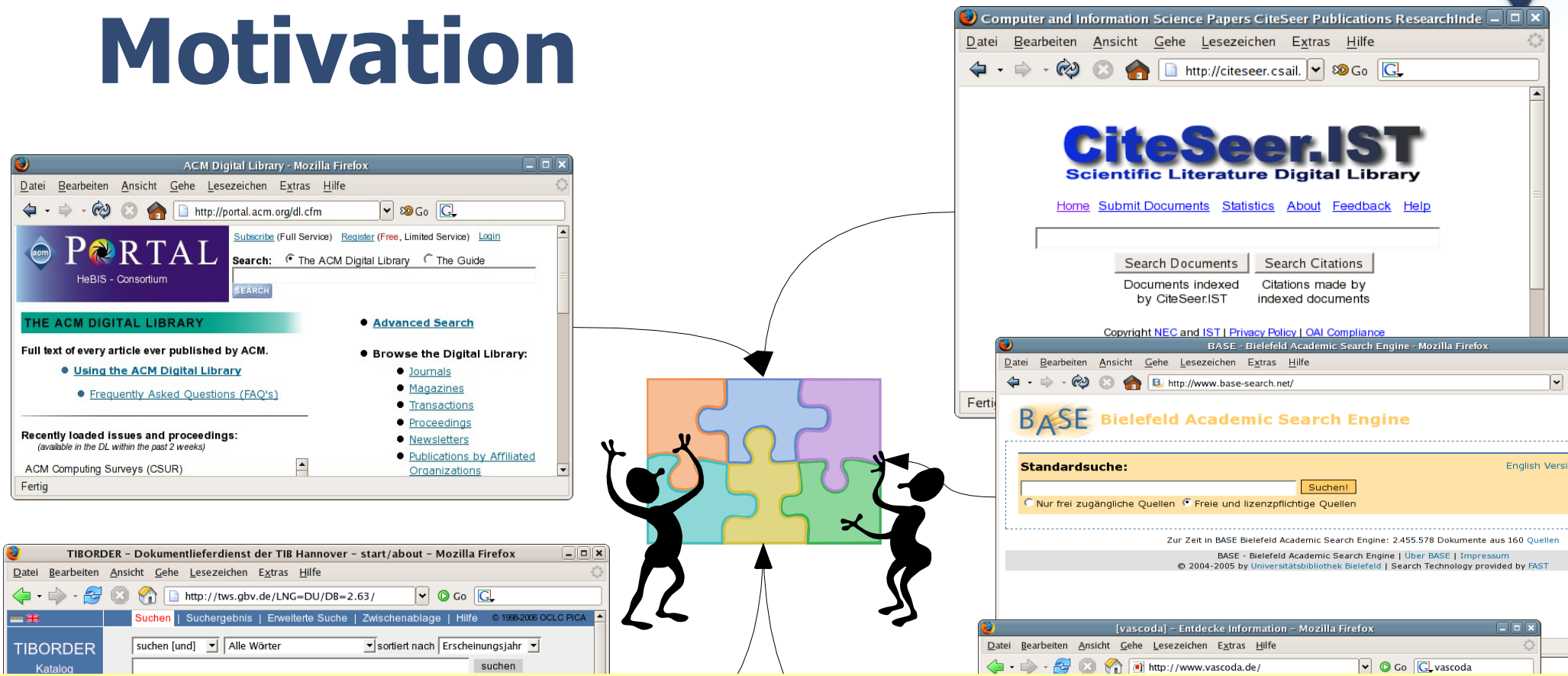




Idee: Gemeinsamer Index

Eine Datenbasis, ein Suchsystem
Suche über einheitliches Interface

Motivation



Idee: Verteilte Suche (Meta-Search; Federated Search)

Eigenständige Datenbasen und Suchsysteme
Suche über zusätzliche Komponente

Eingesetzte Such-Systeme

Gemeinsamkeiten

- Zugriff auf Dokumente per Schlüsselwortsuche
  - Ergebnisliste, üblicherweise nach Relevanz sortiert
- Definition „Dokument"
  - Volltext und/oder Abstrakt in Textform
  - Metadaten (Autor, Erscheinungsdatum, ...)

Eingesetzte Such-Systeme

Unterschiede

- Schemata
  - Dublin Core, Vascoda Profile, ...
- Text-Modell
  - Boolesches Modell, Bag of Words, Vektorraum-Modell, ...
- Ranking
  - Keines (Sortierung), TF/IDF, PageRank, ...
- ...

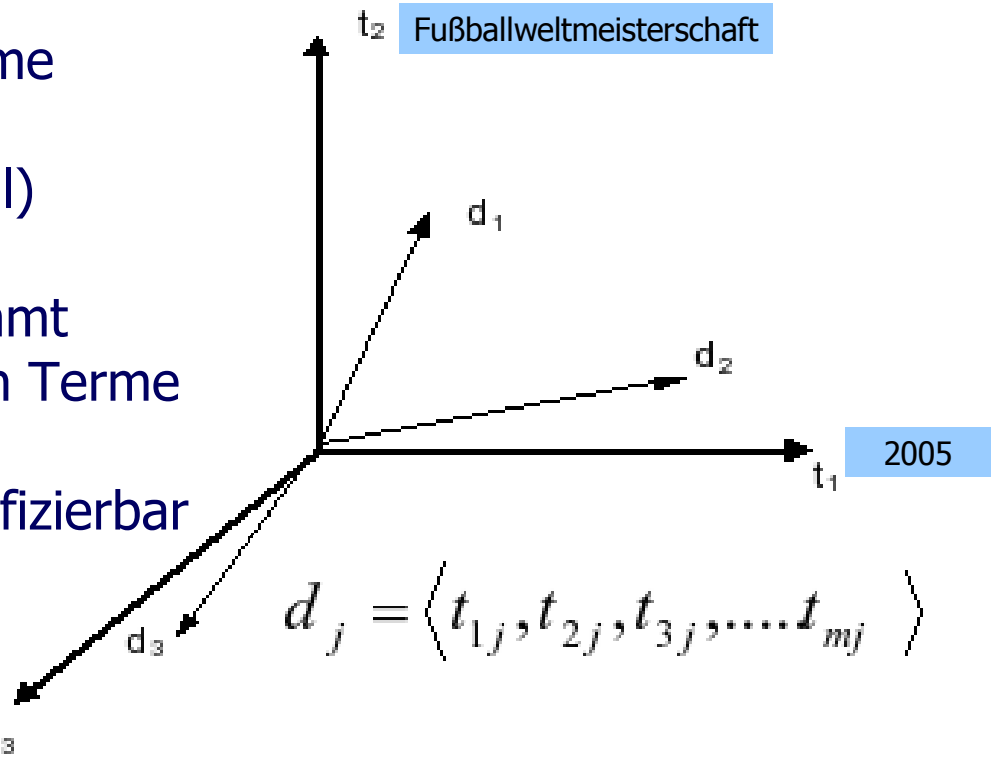
Meta- vs. Federated Search

- Meta-Suche
  - Einfache Kombination aller Ergebnisse
  - Ggf. Rank-Optimierung mittels Heuristiken
  - Kein konsistentes Ranking möglich
  - Durch Analyse der Ergebnis-Listen relativ langsam
- Föderierte Suche
  - Gemeinsames Schema
  - Gemeinsames Text-Modell

Ziel:
Föderierte Suche auf Basis des Vektorraum-Textmodells
Gemeinsames Ranking mittels TFxIDF Maß

Vektorraum-Modell (VSM)

- Indexierung von Dokumenten in Terme
- Eine Dimension pro Term (orthogonal)
- Koordinaten eines Dokuments bestimmt durch die Gewichtunge der jeweiligen Terme
- Ähnlichkeit zweier Dokumente quantifizierbar mittels Distanz im Vektor-Raum, z.B. mit Cosine Measure
- Anfrage = Dokument



$$d_j = \langle t_{1j}, t_{2j}, t_{3j}, \ldots . t_{mj} \rangle$$

$$\rho(q,d) = \frac{\sum_{t \in q \cap d} w_q(t) * w_d(t)}{\|q\| * \|d\|}$$

TFxIDF

- „Term Frequency“, „Document Frequency“
  - $TF_{t,d}$ Wie oft kommt ein Term t im Dokument d vor?
  - DF_t Wie oft kommt ein Term t in allen Dokumenten vor?
  - N Anzahl aller vorhandenen Dokumente
  - $TFxIDF_{d,t}$ Gewichtung eines Dokuments d in Bezug auf den Term t

    $TF_{t,d} \cdot \log (N / DF_t)$
- Verwendung als $\omega_d(t)$ im VSM Relevanz-Maß

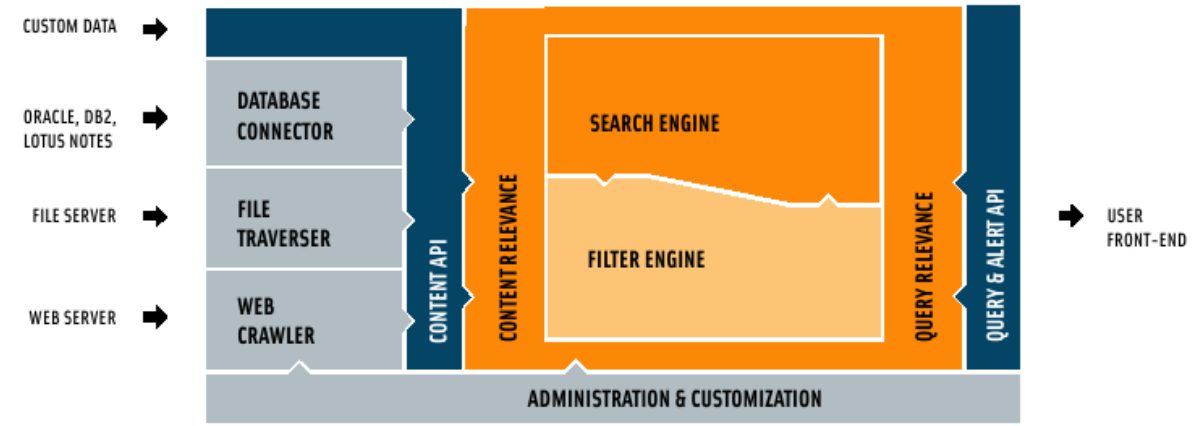
TFxIDF: Verteilte Suche

- Verteilte Suche = Suche über mehrere Kollektionen
- IDF ist abhängig vom Datenbestand!
  Beschreibungskraft eines Terms variabel
- TFxIDF-Maß nicht direkt zwischen zwei Datenbeständen vergleichbar
- Lösung: $TFxIDF' = TF_{t,d} \cdot \log (\sum N / \sum DF_t)$
- Konsequenz:

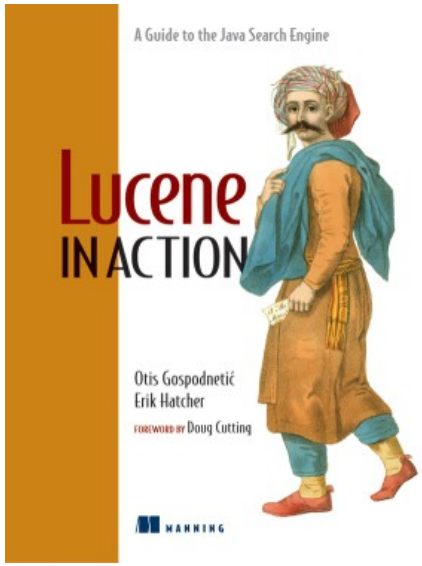
  Austausch von N und DF_t zwischen allen beteiligten Systemen notwendig

FAST Data Search

- Kommerzielle Suchsystem
  (Crawler, Extractor, Indexer, Query Interface)
- Eingesetzt u.a. bei AllTheWeb, Vascoda, Scirus (Elsevier),
  BASE Bielefeld Academic Search Engine,
- Proprietäre, nicht vollständig offen gelegte Programm-Schnittstellen
- Verteilte Suche in anderen FAST-Installation möglich (laut Hersteller)



Lucene



- Open-Source Programmbilbiothek zur Entwicklung von Suchsystemen, keine vollständige Suchmaschine!
- Vollständig offene Programmschnittstellen zur Speicherung von Dokumenten mittels VSM/TFxIDF
- Verteilte Suche bereits implementiert
- Verschiedenste Implementationen von Suchmaschinen oder ähnlichen Produkten auf Basis von Lucene (z.B. Nutch, TIBorder, Eclipse Online-Hilfe, Fuzzy Gazetteer, ...)

A + B = ?

- Einfache Kombination zweier Suchsysteme selbst bei VSM/TFxIDF-basiertem Textmodell nicht ohne weiteres möglich, denn:
  - Bestimmung von „Termen“ ggf. Unterschiedlich:
    - Lemmatisierung/Stemming
    - Synonymwort-Suche
    - Stop-Worte
    - Dokument-Felder
  - Kompatible Schnittstellen zum Austausch von Statistiken erforderlich (N, DF_t)
  - Dies gilt auch für die Verbindung zweier Lucene-basierter Suchmaschinen!

A + B = A∪B

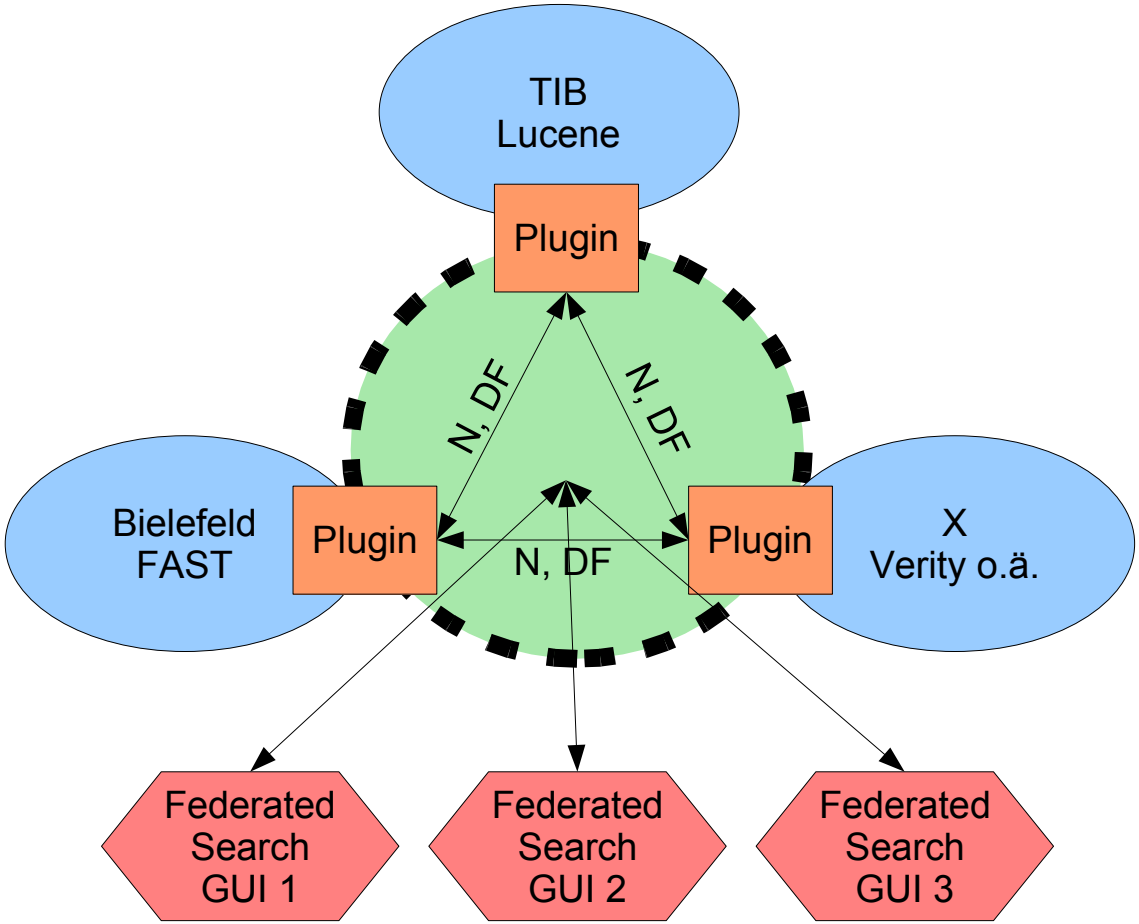
- Standards helfen
  - Z39.50-2003
  - ZING („Z39.50-International: Next Generation“)
  - STARTS/SDARTS
    STAnford protocol proposal for Internet ReTrieval and Search

    SDARTS = STARTS + SDLIP
    Simple Digital Library Interoperability Protocol
- Implementation?
  - FAST: ?
  - Lucene: SDARTS?

Suchmaschinen-Plugin

- Erweiterung bestehender Suchsysteme um eine einheitliche Schnittstelle
- Gewährleistet homogenes Ranking
- Suchanfragen an die Föderation über Plugin
  - Parallele Anfrage an alle Teilnehmer
  - Durch TFxIDF ' automatisch gemeinsames Ranking
- Verwendung von (Quasi-)Standards
  (z.B. STARTS/SDARTS oder Lucene Searchable)

Suchmaschinen-Plugin (2)



Suchmaschinen-Plugin (3)

- Re-indexierung bestehender Korpora (vollautomatisch, schnell)
- Einheitliche Lemmatisierung/Stemming, Stop-Wort-Listen usw.
- Einheitliches Dokument-Schema
- Einheitliche Abfragesprache
- Kann parallel zu bestehenden Systemen betrieben werden
- Übernimmt nur die Suchfunktion – Crawling und Präprozessierung werden weiterhin mit der bestehenden Systemumgebung durchgeführt
- Kosteneffizienz

Prototyp

- Lucene-Implementation für FAST Data-Search
  - Importiert FAST's internes Dokument-Format FIXML.
  - Verteilte Suche mittels Lucenes RemoteSearchable und RMI.
  - Parallele Suche mittels Lucenes ParallelMultiSearcher
- Test-Installation in Hannover (L3S) und Bielefeld (UB Bielefeld)

Demo

Next Steps

- Beta-Version des Federated Search Systems
  - Große Datenbestände
  - SDLIP-Implementation
  - Besseres GUI
-

Vielen Dank!

Christian Kohlschütter
kohlschuetter@l3s.de