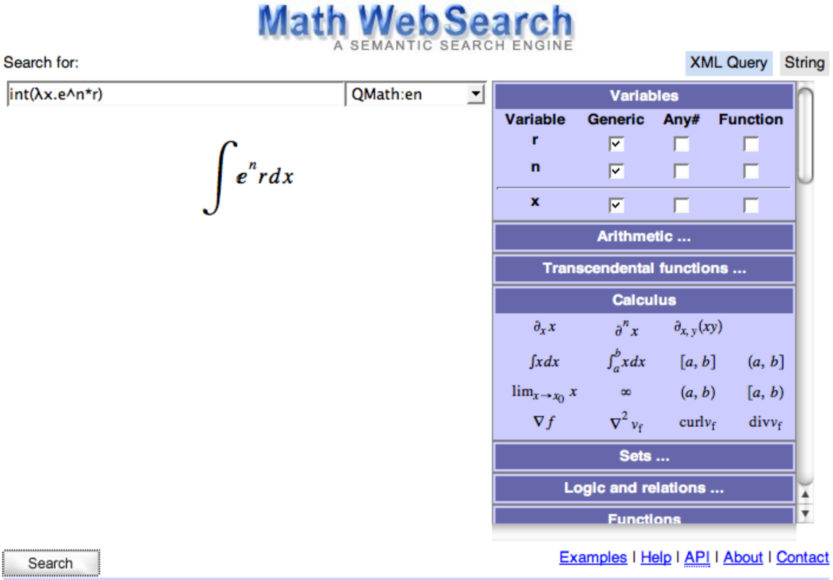
MathWebSearch 0.5: Scaling an Open Formula Search Engine

Michael Kohlhase, Bogdan A. Matican, Corneliu C. Prodescu

http://kwarc.info/kohlhase
Center for Advanced Systems Engineering
Jacobs University Bremen, Germany

July 13, 2012

Instead of a Demo: Searching for Signal Power



Instead of a Demo: Search Results

Other integrals (5 formulas) (Source)

Other integrals (5 formulas)

Matched term:

$$\int \frac{e^{3z/4}}{\left(-2+e^{3z/4}\right)\sqrt{-2+e^{3z/4}+e^{3z/2}}}\,dz = \frac{2}{3}\left(\log\left(-2+e^{3z/4}\right)-\log\left(4\sqrt{-2+e^{3z/4}+e^{3z/2}}+5e^{3z/4}-2\right)\right)$$

Rank: 100%

XML Source

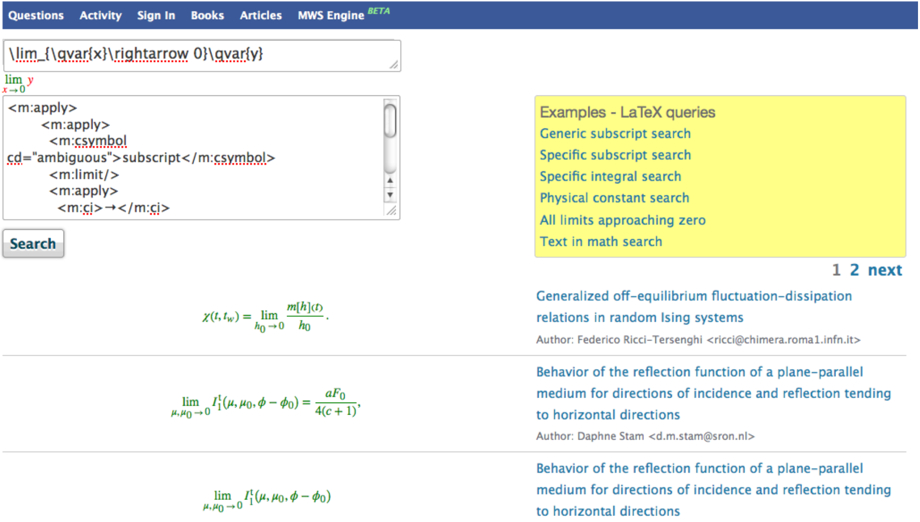
Used substitution:

$\mathbf{n} \rightarrow 3z4^{-1}$

$\mathbf{r} \rightarrow \left(\left((-2)+e^{3z4^{-1}}\right)\left((-2)+e^{3z4^{-1}}+e^{3z2^{-1}}\right)^{1/2}\right)^{-1}$

$\mathbf{x} \rightarrow z$

Instead of a Demo: LaTeX-based Search on the arXiv



Instead of a Demo: Appliccable Theorem Search in Mizar

```
definition
 let k, n be Ordinal;
 pred k divides n means :Def3: :: MTEST1:def 3
 ex a being Ordinal st n = k *^ a;
 reflexivity
 proof
  let n be Ordinal; :: thesis:
  thus ex a being Ordinal st n = n *^ a ;

                    ATP Proof not found

                      status: Timeout
             Suggest hints, Unification query,

                     Suggested hints

               t73_card_2, t39_ordinal2,

          Try SPASS, Export problem to SystemOnTPTP

  :: thesis:
 end;
end;
```

MathWebSearch: Search Math. Formulae on the Web

- Idea 1: Crawl the Web for math. formulae (in OpenMath or CMathML)
- Idea 2: Math. formulae can be represented as first order terms (see below)
- Idea 3: Index them in a substitution tree index (for efficient retrieval)
- Problem: Find a query language that is intuitive to learn
- Idea 4: Reuse the XML syntax of OpenMath and CMathML, add variables

History of MWS

- 2005 Initial implementation/first prototype for content search [KŞ06]
- Problem: There was almost nothing to index (crawler found 13 new content MathML pages in 3 months)
- Starting to convert the `arXiv.org` with LaTeXML (500.000 papers)
- 2006/7 work on user interfaces (Sentido [GP06])
- 2009 combination with text search (Stefan Anca [Anc07])
- 2010 complete re-implementation of core (Corneliu Prodescu [PK11])
  - RESTful Web Service Infrastructure (mwsd)
  - Content MathML as an interface language throughout (MWS harvests)
- 2011: ?LaTeX as a query language (via the LaTeXML daemon [GSK11])
- 2011: Applicable Theorem Search for Mizar ([IKRU11])
- 2012: Distributing MathWebSearch ([KMP12])
- 2012: Indexing Induced Statements ([KI12])

Instantiation Queries

- Application: Find partially remembered formulae
- **Example 1** An engineer might face the problem remembering the energy of a given signal $f(x)$
  - Problem: hmmmm, have to square it and integrate
  - Query Term: $\int_{\boxed{min}}^{\boxed{max}} \boxed{f}(x)^2 dx$ ($\boxed{i}$ are search variables)
  - One Hit: Parseval's Theorem $\frac{1}{T}\int^{T_0} s^2(t)dt = \sum_{k=-\infty}^{\infty} \|c_k\|^2$ (nice, I can compute it)
- This works out of the box (has ween working in `MathWebSearch` for some time)
- Another Application: Underspecified Conjectures/Theorem Proving
  - during theory exploration we often have some freedom
  - express that using metavariables in conjectures
  - instantiate the conjecture metavariables as the proof as the proof dictates

applied e.g. in Alan Bundy's "middle-out reasoning" in proof planing

Generalization Queries

- Application: Find (possibly) appliccable theorems
- **Example 2** A researcher wants to estimate $\int_{\mathbb{R}^2} |\sin(t)\cos(t)| dt$ from above
  - Problem: Find inequation such that $\int_{\mathbb{R}^2} |\sin(t)\cos(t)| dt$ matches left hand side.
  - e.g. Hölder's Inequality: ($\boxed{i}$ are universal variables)

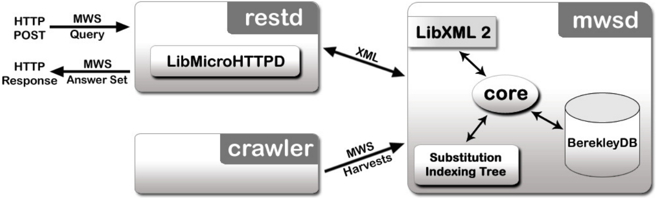
$$\int_{\boxed{D}} \left|\boxed{f}(x)\boxed{g}(x)\right| dx \leq \left(\int_{\boxed{D}} \left|\boxed{f}(x)\right|^p dx\right)^{\frac{1}{p}} \left(\int_{\boxed{D}} \left|\boxed{g}(x)\right|^q dx\right)^{\frac{1}{q}}$$

  - Solution: Take the instance

$$\int_{\mathbb{R}^2} |\sin(x)\cos(x)| dx \leq \left(\int_{\mathbb{R}^2} |\sin(x)|^p dx\right)^{\frac{1}{p}} \left(\int_{\mathbb{R}^2} |\cos(x)|^q dx\right)^{\frac{1}{q}}$$

Problem: Where do the index formulae come from in particular the universal variables (we'll come back to that later)

System Architecture

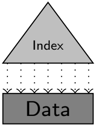


- crawlers for MathML, `OpenMath`, and OAI repositories. (convert your's?)
- multiple search servers based substitution tree indexing (formula search)
- a RESTful server that acts as a front-end for multiple search servers.
- various front ends tailored to specific applications (search appliances)
  - a Google-like web front end for human users (`search.mathweb.org`)
  - a LaTeX-based front-end for the arXiv (`http://arxivdemo.mathweb.org`)
  - special integrations for theorem prover libraries (MizarWiki, TPTP)

Term-Indexing

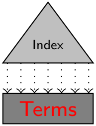
- Motivation: Automated theorem proving (efficient systems)
- Problem: Decreasing inference rate (basic operations linear in # of formulae)
- Idea: Make use of structural equality between terms (term indexing)

  database systems (Algorithms: select, meet, join)



  - Data: `PERSON(hans, manager, 32)`
  - Query: "find all 40-year old persons"

  automated theorem proving (Algorithm: Unification)

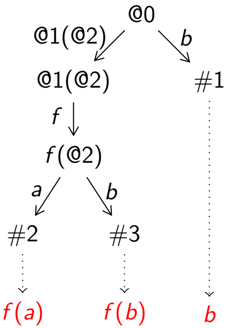


  - Data: $P(f(x,g(a,b)))$
  - Queries: "find all literals that are unifiable with $P(f(c,y))$"

An (additional) index data structure can make the retrieval logarithmic

Term Indexing in MathWebSearch

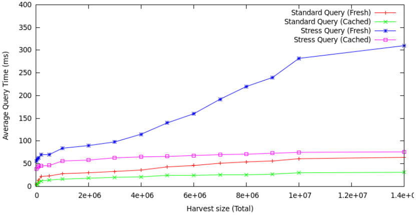
- in-memory index
- leaf nodes linked to database
- depth-first substitution tree
- collapse redundant subterms
  - $f(a,b,b) \rightarrow f(a,b,[3])$
  - $g(a,f(a),f(a)) \rightarrow g(a,f([2]),[3])$
- encode tokens: $\mathit{token} : \mathit{string} \rightarrow \mathit{id} : \mathit{int}32$



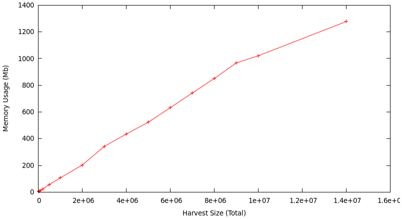
Index statistics

- Experiment: Indexing the arXiv (700k documents, $\sim 10^8$ non-trivial formulae)
- Results: indexing up to 15 M formulae on a standard laptop

Query Times



Memory Footprint



  - query time is constant ($\sim$ 50 ms) (as expected; goes by depth $\times$ symbols)
  - memory footprint seems linear ($\sim 100 \frac{B}{\text{formula}}$) (expected more duplicates)
- So we need ca. 200 *GB* RAM for indexing the whole arXiv.
- Can index all published Math ($\hat{=}$ 5 $\times$ arXiv) on a large server (1 *TB* RAM). (ZBL $\hat{=}$ 3M art.)

Coping with Memory Problems

- Intel has announced motherboard that can take 1 *TB* of RAM. (Q2 2012)
- Our new server only has 128 *GB*, ...
- ...but we have (access to) a cluster of 4 *GB*-RAM machines.
- Idea: Make `MathWebSearch` a distributed system (solves other load problems as well)
- Problem: Need to distribute the index data structure (non-standard in distribution)
- Design Goals:
  - efficient tree distribution,
  - persistency, migration, load balancing,
  - tree space optimizations.
- top-level hashing not enough (trees very unbalanced)

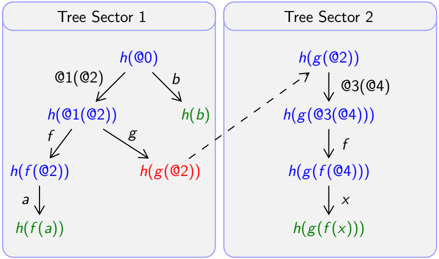
Dividing Memory into Sectors (for distribution, persistency, migration)

- Idea: Organize the memory needed for the index into chunks that can be moved between machines
- **Definition 3** memory sectors are continuous RAM chunks of fixed size
- implement as mmapped file (using POSIX mmap) (yields persistency, migration)
- no serialization (not necessary in homogenous clusters)
- bound size to 2^{31} (pointer size reduction in trees)

Tree Sectors in Memory Sectors

- Idea: Need to split index tree into parts that fit into memory sectors

Example 4 (Tree Sectors)



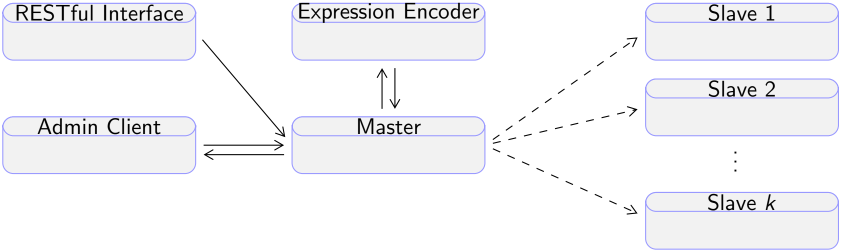
Internal nodes * Leaf nodes * Remote nodes *

- Supported Operations
  - insert / update
  - query
  - split
- Split goals
  - even distribution
  - minimized remote nodes

- Tree Sector Splitting: DFTraverse monitoring sizes of explored part and fringe when a threshold is reached redistribute nodes (60% size; fringe minimal)
  - explored nodes ⇝ old sector
  - unexplored nodes ⇝ new sector
  - fringe ⇝ old sector (**) and new (sector*)

Distributed Architecture

- Master/Slave Architecture:
  - Master manages slaves, distributes actions, and keeps metadata maps (slim)
  - Slaves update/query, pass metadata to master (keep multiple tree memory sectors)



- Distributed Update: Master finds slave with index root sector, forwards request, slave
  - updates term db (if it hits a leaf note)
  - forwards to remote slave (if it hits a remote node)
- Distributed Query: Similar, but all paths must be checked
  - master reserves a unique ID for query, monitors result bound
  - slaves report hits to master, abort search, when master stops them.

Evaluation of Distribution

- Implementation ca. 3 months for two (very strong) undergrads
- query time punishment $\leq 3\times$ worst case, $\leq 1.5\times$ avg. case
- memory footprint reduction by 35% (pointer size reduction)
- What is missing?: working on next (when Prode is back from Facebook)
  - more experiments, large Installations (waiting for LaTeXML improvements)
  - load balancing and index-distribution strategies (fine-tuning efficiency)
  - fault tolerance (what happens if a slave runs away?)
- Alternatives: We would like to compare to disk-based alternatives:
  - just let it swap (possible baseline; scary)
  - keep selected parts of the index on disk (needs query prediction)
  - competitive parallelism of partial indexes (how to integrate hits for prolific queries)
- But most importantly. . . : We did it!

Conclusions and Recap

- Recap: (what should you remember?)
  - Need Math Search Engines for unlocking the scientific Web
  - Presentation-based search is not enough (symbolic computation)
  - 4 simple ideas (Crawl, FOFormulae, Index, GUI) are enough
  - we can now deal with very large indexes (needs tuning)
  - Implementation running at `http://arxivdemo.mathweb.org/index.php?p=/article/MWS` (1k papers)
- Remaining Problems (what are we be working on?)
  - Query tools (input formula editor, firefox plugin,...)
  - (almost) no content Math on the Web (arXiv trafo, parallel markup,...)
- Opportunities (Why are we so excited?)
  - Theorem prover libraries (and finally interoperability)
  - indexing time series (approximate by polynomials, index those)
  - just like Gooogle drives the commercial web, `MathWebSearch` could drive science

Ştefan Anca.
MaTeSearch a combined math and text search engine.
Bachelor's thesis, Jacobs University Bremen, 2007.

Alberto González Palomo.
Sentido: an authoring environment for OMDoc.
In OMDoc – *An open markup format for mathematical documents [Version 1.2]*, number 4180 in LNAI, chapter 26.3. Springer Verlag, August 2006.

Deyan Ginev, Heinrich Stamerjohanns, and Michael Kohlhase.
The LaTeXML daemon: Editable math on the collaborative web.
In James Davenport, William Farmer, Florian Rabe, and Josef Urban, editors, *Intelligent Computer Mathematics*, number 6824 in LNAI, pages 292–294. Springer Verlag, 2011.

Mihnea Iancu, Michael Kohlhase, Florian Rabe, and Josef Urban.
The mizar mathematical library in omdoc: Translation and applications.
submitted to JAR, 2011.

Michael Kohlhase and Mihnea Iancu.
Searching the space of mathematical knowledge.
MIR Symposium, 2012.

Michael Kohlhase, Bogdan A. Matican, and Corneliu C. Prodescu.

Mathwebsearch 0.5 – scaling an open formula search engine.
In Johan Jeuring, John A. Campbell, Jacques Carette, Gabriel Dos Reis, Petr Sojka, Makarius Wenzel, and Volker Sorge, editors, *Intelligent Computer Mathematics*, number 7362 in LNAI. Springer Verlag, 2012.

Michael Kohlhase and Ioan Şucan.
A search engine for mathematical formulae.
In Tetsuo Ida, Jacques Calmet, and Dongming Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation, AISC'2006*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

Corneliu C. Prodescu and Michael Kohlhase.
Mathwebsearch 0.5 - open formula search engine.
In *Wissens- und Erfahrungsmanagement LWA (Lernen, Wissensentdeckung und Adaptivität) Conference Proceedings*, sep 2011.