code{4}lib

LucidWorks™

SoIr Update

code4lib conference, 13 Feburary '13, Chicago

presented by
Erik Hatcher

Search | Discover | Analyze

Abstract

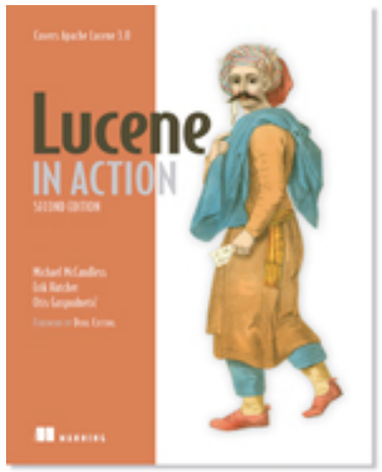
Solr is continually improving. Solr 4 was recently released, bringing dramatic changes in the underlying Lucene library and Solr-level features. It's tough for us all to keep up with the various versions and capabilities.

This talk will blaze through the highlights of new features and improvements in Solr 4 (and up). Topics will include: SolrCloud, direct spell checking, surround query parser, and many other features. We will focus on the features library coders really need to know about.

LucidWorks™

About: È ℛ Ỉ Ķ Ḥ Ằ Ţ Ç Ḣ Ḗ Ŕ

- Co-author “Lucene in Action”
- Lucene/Solr Committer and PMC, ASF Member
- Senior Solutions Architect and co-founder, LucidWorks
  - (formerly Lucid Imagination)
- Library Cred:
  - developer for Rossetti Archive and NINES
  - originator/namer of Blacklight



blacklight

LucidWorks™

LucidWorks™

Lucene 4 Highlights

- Flexible index formats
- Pluggable scoring
- String -> BytesRef
- DWPT (Document Writer Per Thread)
  - faster, more consistent indexing speed
- NRT (Near Real-Time)
  - per-segment loading of FieldCache, soft commits
- Spatial overhaul
- FST/FSA
  - FuzzyQuery over 100x faster
  - also reduces memory footprint for Terms index
- And much much more!
  - See http://lucene.apache.org/core/4_1_0/changes/Changes.html

Flexible index formats

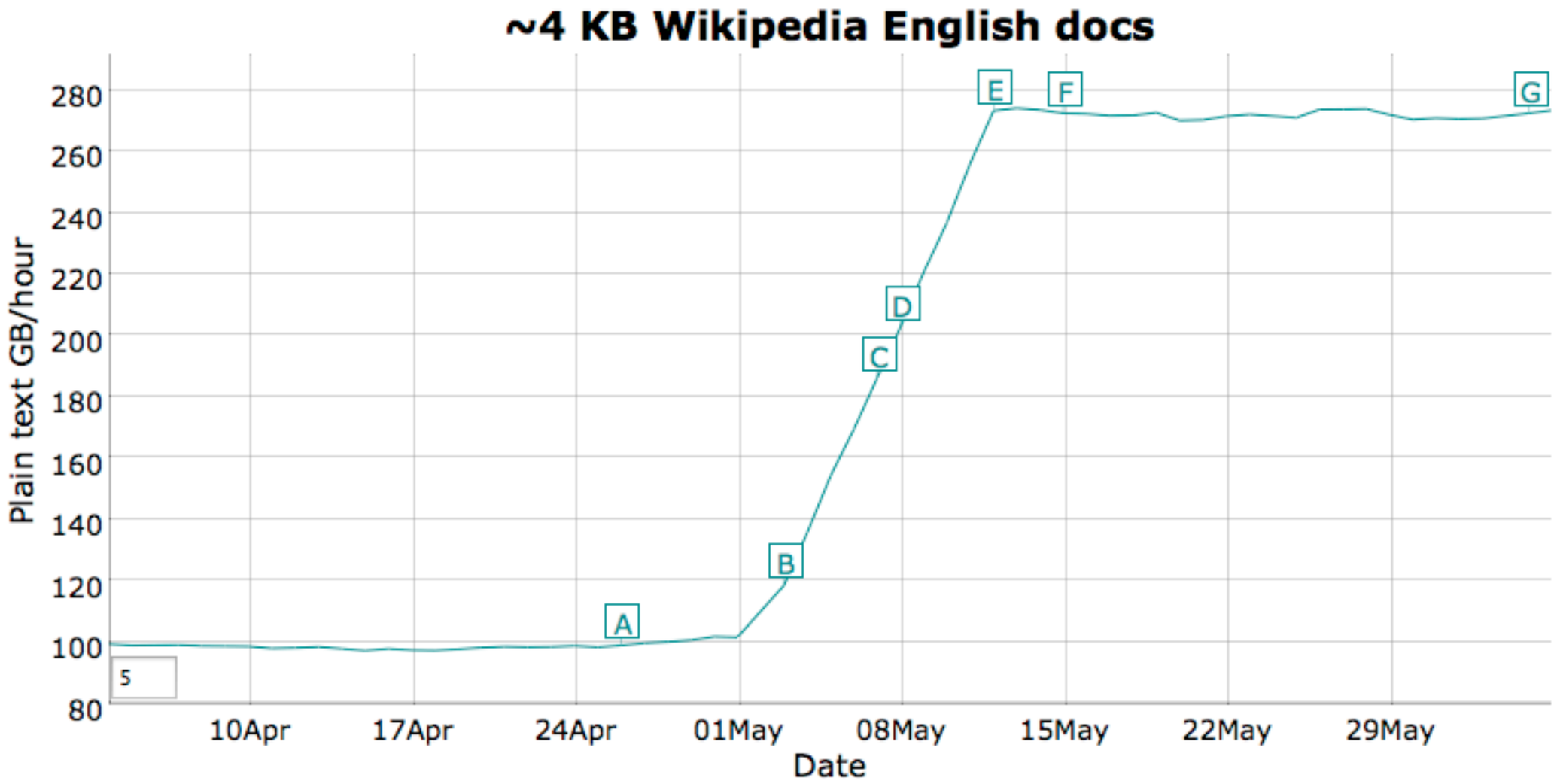
- For terms, postings lists, stored fields, term vectors, etc
- Several new posting list codecs
  - Pulsing (inlines low doc freq)
  - Block (packed int blocks)
  - SimpleText (debugging, transparency)
  - Bloom (experimental, also inlines low doc freq)
  - Appending (for append-only filesystems such as HDFS)
  - Memory (terms as FST)
- Compressed stored fields

Pluggable scoring

- Decoupled from traditional vector space (TF/IDF)
- Additional index statistics
  - number of tokens for a term or field
  - number of postings for a field
  - number of documents with a posting for a field
- Several built-in alternatives:
  - BM25
  - DFR – divergence from randomness
  - Information-based models

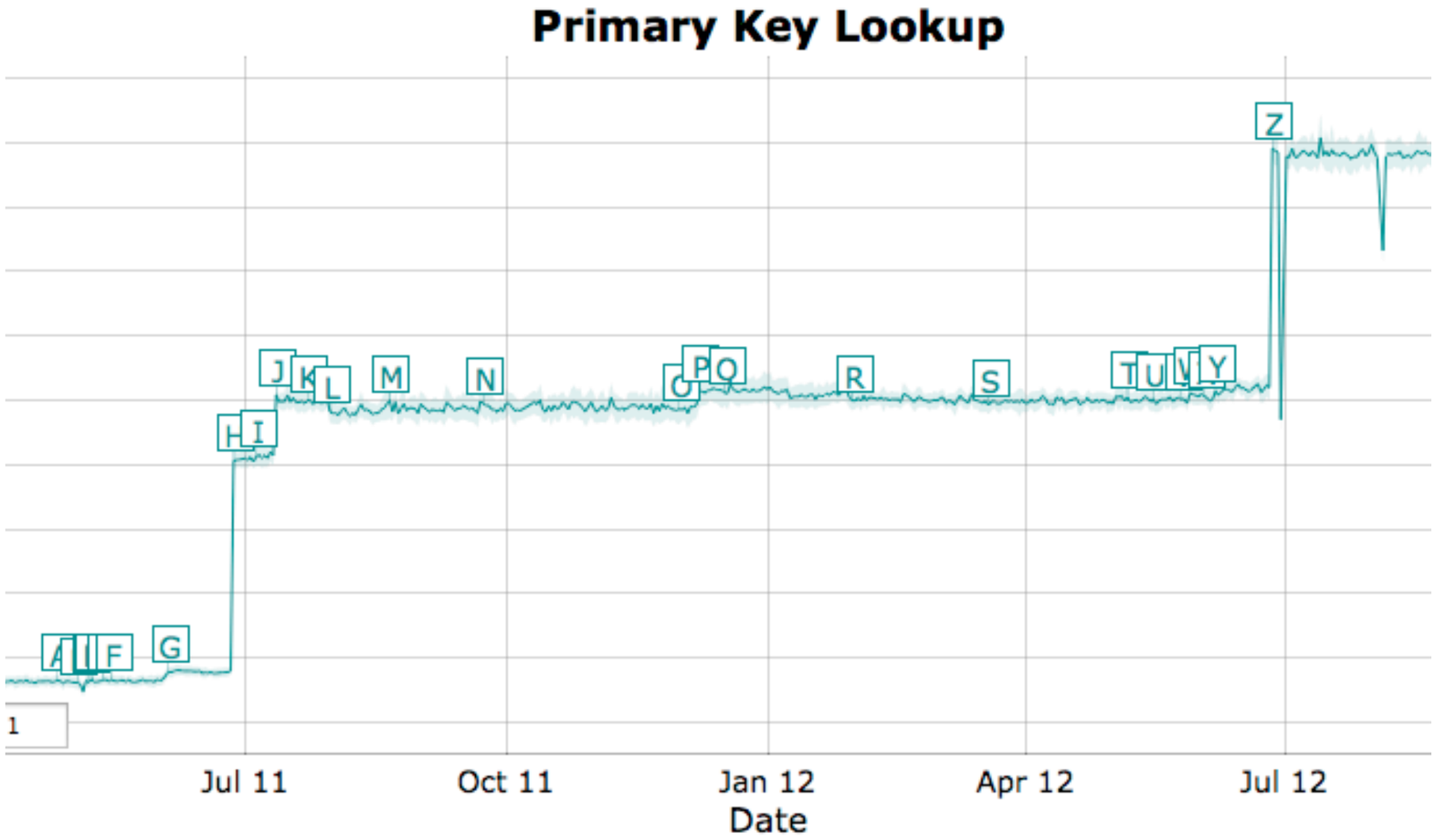
Indexing performance

- http://people.apache.org/~mikemccand/lucenebench/indexing.html



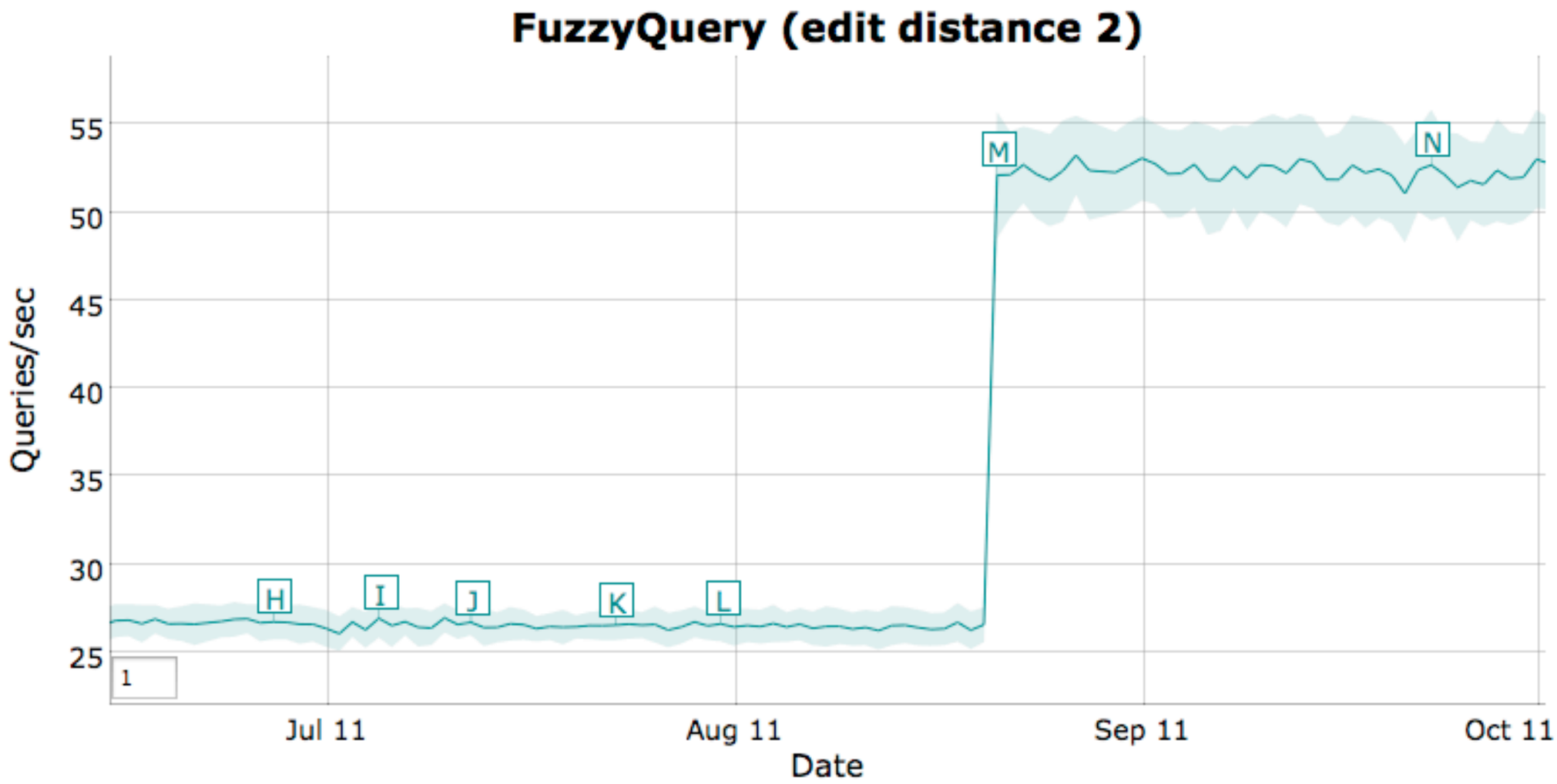
QPS (primary key lookup)

- http://people.apache.org/~mikemccand/lucenebench/PKLookup.html



FuzzyQuery

- http://people.apache.org/~mikemccand/lucenebench/Fuzzy2.html



LucidWorks™

Solr 4 Highlights

- Requires Java 1.6+
- Pivot facets
  - http://wiki.apache.org/solr/SimpleFacetParameters#facet.pivot
- DirectSpellChecker support
  - http://wiki.apache.org/solr/SpellCheckComponent
- Improved document response
  - DocTransformer: [shard], [explain], [value], [docid]
  - Function query results
  - http://wiki.apache.org/solr/DocTransformers
- Pseudo-join
  - http://wiki.apache.org/solr/Join
- Surround query parser

More Solr 4 Highlights

- Transaction log
- Several new update processors, including a “script” one
  - http://wiki.apache.org/solr/ScriptUpdateProcessor
- Spatial overhaul
  - http://wiki.apache.org/solr/SpatialSearch
- Content-type savvy /update handler
- SolrCloud
  - http://wiki.apache.org/solr/SolrCloud
- And more!
  - See http://lucene.apache.org/solr/4_1_0/changes/Changes.html

Solr 4.1

- Enhanced document routing (custom sharding)
- Compressed stored fields
- MoreLikeThis distributed capability
- AnalyzingSuggester
  - http://blog.mikemccandless.com/2012/09/lucenes-new-analyzing-suggester.html
  - via lookupImpl = org.apache.solr.spelling.suggest.fst.AnalyzingLookupFactory
  - and FuzzyLookupFactory
- Many SolrCloud fixes and improvements
- Stanford! - _query_ no longer needed to specify nested query parsers

LucidWorks™

Looks Good!

Apache Solr

- Dashboard
- Logging
- Core Admin
- Java Properties
- Thread Dump
- collection1

Instance

Start	5 minutes ago
Host	192.168.1.102
CWD	/opt/code/lusolr/solr/example
Instance	/opt/code/lusolr/solr/example/solr/collection1
Data	/opt/code/lusolr/solr/example/solr/collection1/data
Index	/opt/code/lusolr/solr/example/solr/collection1/data/index

Versions

solr-spec	5.0.0.2012.08.15.13.17.06
solr-impl	5.0-SNAPSHOT 1373442M - yonik - 2012-08-15 13:17:06
lucene-spec	5.0-SNAPSHOT
lucene-impl	5.0-SNAPSHOT 1373442 - yonik - 2012-08-15 13:15:15

JVM

Runtime	Java HotSpot(TM) 64-Bit Server VM (20.8-b03-424)
Processors	4

System

Physical Memory 73.0%

5.84 GB

8.00 GB

Swap Space 72.4%

2.17 GB

3.00 GB

File Descriptor Count 1.5%

153

10240

JVM-Memory 14.7%

18.24 MB

81.06 MB

123.94 MB

Documentation | Issue Tracker | IRC Channel | Community forum | Solr Query Syntax

LucidWorks™

Pivot Faceting

- Finds the top N constraints for field1, then for each of those, finds the top N constraints for field2, etc
- Syntax: facet.pivot=field1,field2,field3,…

```
facet.pivot=cat,inStock
```

	#docs	#docs w/ inStock:true	#docs w/ instock:false
cat:electronics	14	10	4
cat:memory	3	3	0
cat:connector	2	0	2
cat:graphics card	2	0	2
cat:hard drive	2	2	0

DirectSpellChecker

- Automaton-based
- Candidates are presented directly from the term dictionary, based on Levenshtein distance.
- A practical benefit of this spellchecker is that it requires no additional datastructures (neither in RAM nor on disk) to do its work.
  - http://lucene.apache.org/core/4_1_0/suggest/org/apache/lucene/search/spell/DirectSpellChecker.html

LucidWorks™

Improved document response

- Returns other info along with document stored fields
- Function queries
  - fl=name,location,geodist(),add(myfield,10)
- Fieldname globs
  - fl=id,attr_*
- Multiple “fl” (field list) values
  - &fl=id,attr_*
  - &fl=geodist()
  - &fl=termfreq(text,’solr’)
- Aliasing
  - fl=id,location:loc,_dist_:geodist()
- fl=id,[explain],[shard]

Improved document response example

```
$ curl http://localhost:8983/solr/query?
    q=solr
    &fl=id,apache_mentions:termfreq(text,'apache')
    &fl=my_constant:"this is cool!"
    &fl=inStock, not(inStock)
    &fl=other_query_score:query($qq)
    &qq=text:search

{ "response":{"numFound":1,"start":0,"docs":[
      {
        "id":"SOLR1000",
        "apache_mentions":1,
        "my_constant":"this is cool!",
        "inStock":true,
        "not(inStock)":false,
        "other_query_score":0.84178084
    }]}
```

Query Parsing

- _query_ no longer needed for nested queries
  - https://issues.apache.org/jira/browse/SOLR-4093
- "surround" query parser
  - enables the use of Lucene's SpanQuery family, sophisticated proximity matching
  - Examples:
    - 5n(dog cat)
    - dog 5w cat
  - http://wiki.apache.org/solr/SurroundQueryParse

New Spatial Support

- wiki.apache.org/solr/SpatialSearch
- Multiple values per field
- Index shapes other than points (circles, polygons, etc)
- Indexing:
  - "geo”:”43.17614,-90.57341”
  - “geo”:”Circle(4.56,1.23 d=0.0710)”
  - “geo”:”POLYGON((-10 30, -40 40, -10 -20, 40 20, 0 0, -10 30))”
- Searching:
  - fq=geo:"Intersects(-74.093 41.042 -69.347 44.558)"
  - fq=geo:"Intersects(POLYGON((-10 30, -40 40, -10 -20, 40 20, 0 0, -10 30)))"

Add and Retrieve document

```
$ curl http://localhost:8983/solr/update -H 'Content-type:application/
json' -d '
[
 { "id" : "book1",
   "title" : "Infinite Jest",
   "author" : "David Foster Wallace"
 }
]'
```

```
$ curl http://localhost:8983/solr/get?id=book1
{
  "doc": {
    "id" : "book1",
    "author": "David Foster Wallace",
    "title" : "Infinite Jest",
    "_version_": 1410390803582287872
  }
}
```


LucidWorks™

Atomic Updates

```
$ curl http://localhost:8983/solr/update
       -H 'Content-type:application/json' -d '
[
 {"id"         : "book1",
  "pubyear_i" : { "add" : 2006 },
  "ISBN_s"    : { "add" : "0-380-97365-1"}
 }
]'
```

```
$ curl http://localhost:8983/solr/update
       -H 'Content-type:application/json' -d '
[
 {"id"         : "book1",
  "copies_i"  : { "inc" : 1},
  "cat"       : { "add" : "fiction"},
  "ISBN_s"    : { "set" : "0-316-92004-5"}
  "remove_s"  : { "set" : null } }
]'
```


LucidWorks™

Pseudo-Join

id: **blog1**
name: Blog 1
owner: c4l
Started: 2007-10-26

id: **blog2**
name: Blog 2
owner: zoia
started: 2005-1-31

id: post1
blog_id: blog1
author: John Doe
title: Pseudo-join can be handy!
body: Here's how to use {!join....}

id: post2
blog_id: blog1
author: John Doe
title: Solr Update
body: Live streaming today!

id: post3
blog_id: **blog2**
author: Jane Doe
title: What's New at code4lib

Restrict to blogs mentioning netflix:

fq={!**join** from=**blog_id** to=**id**}body:code4lib

- How it works:
  - Finds all documents matching “code4lib”
  - Maps to different docs by following **blog_id** to **id**

Pseudo-Join Examples

- Only show posts from blogs started after 2010

  &fq={!join from=id to=blog_id}started:[2010 TO *]

- If any post in a blog mentions “Chicago”, then search all posts in that blog for “conference” (self-join)

  q=conference

  &fq={!join from=blog_id to=blog_id}Chicago

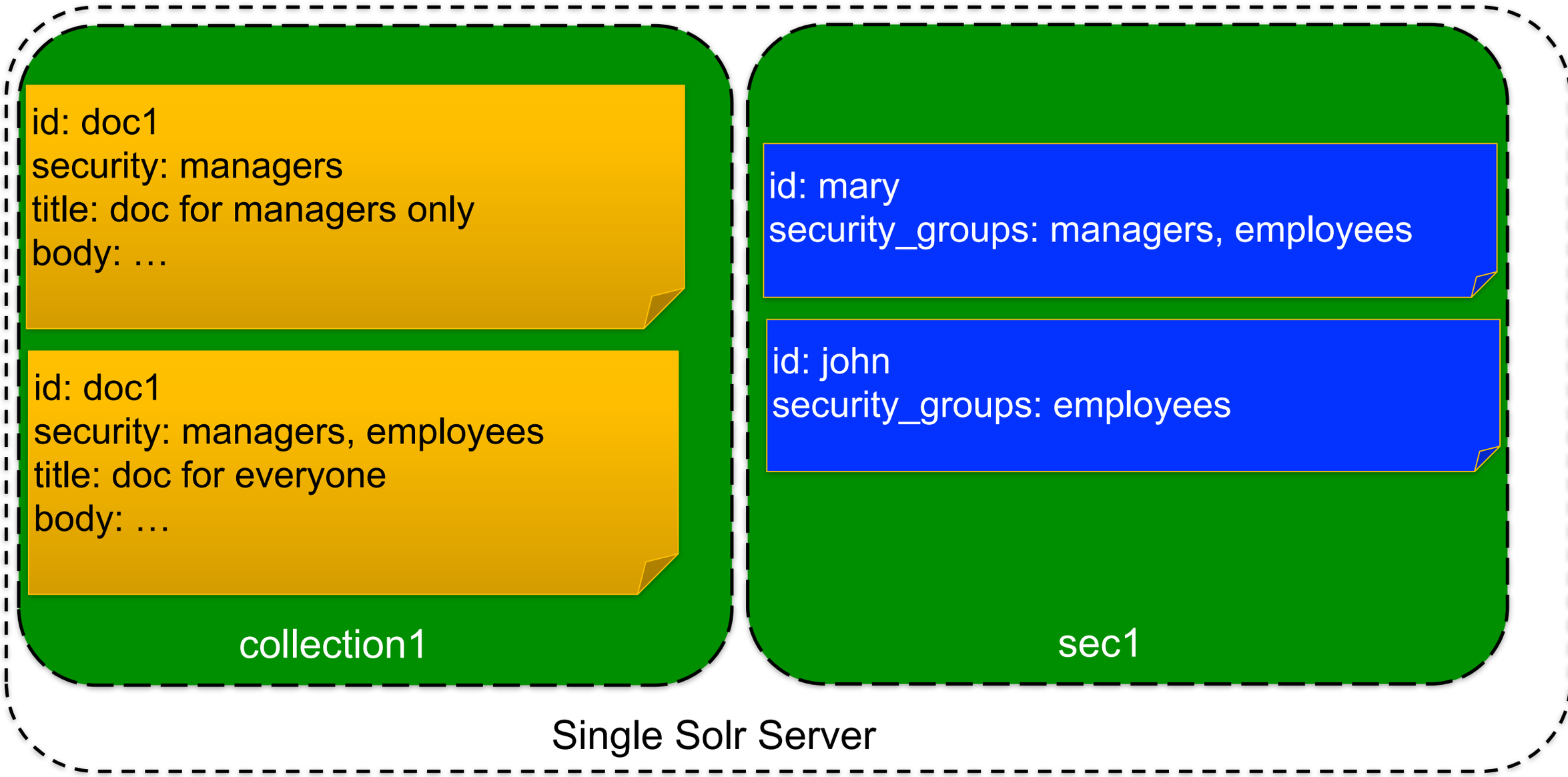
- If any blog post mentions “Chicago”, then search all emails with the same blog owner for “conference”

  q=email_body:conference

  &fq={!join from=owner_email_user to=email_user}{!join from=blog_id to=id}Chicago

Cross-Core Join

http://localhost:8983/solr/collection1/select?q=*:*
&fq={!join **fromIndex**=sec1 from=security_groups to=security}user:john



LucidWorks™

New UpdateProcessor's

- FieldMutatingUpdateProcessor family:
  - ConcatField, CountField, FieldLength, HTMLStripField, IgnoreField, RegexReplace, RemoveBlankField, TrimField, TruncateField
- ScriptUpdateProcessor
  - enables update processing code to be written in a scripting language. The script can be written in any scripting language supported by your JVM (such as JavaScript), and executed dynamically so no pre-compilation is necessary.
  - http://wiki.apache.org/solr/ScriptUpdateProcessor

SolrCloud: Solr 4's scalability

- Sharded leaders and replicas
- ZooKeeper used for cluster management
- Distributed indexing
  - Automatically distributes updates to appropriate shard
  - Facilitates Near Real-Time (NRT) searching
- Distributed search
  - Automatically distributes to nodes of each shard
- Robust, automatic update recovery
- Real-time /get
  - Leverages transaction log
- No single point of failure
- Large scale NRT using soft commits
- Transaction log uses:
  - Durability for updates that have not yet been committed
  - Peer syncing in SolrCloud
  - Real-time get

SolrCloud Visualization

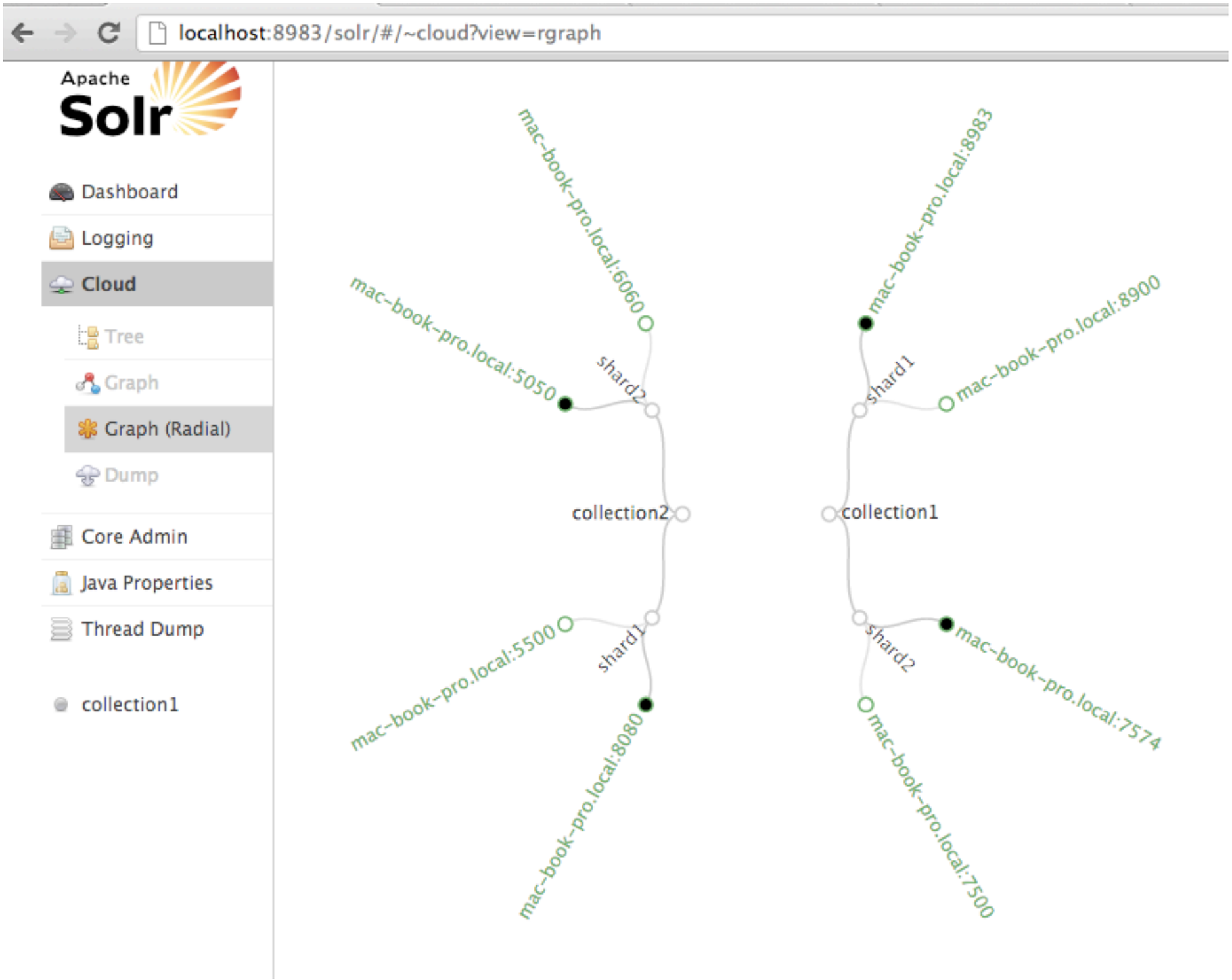


Image from http://bit.ly/X4E5H9

LucidWorks™

Near Real Time (NRT) softCommit

- softCommit opens a new view of the index without flushing + fsyncing files to disk
  - Decouples update visibility from update durability
- commitWithin now implies a soft commit
- Current autoCommit defaults from solrconfig.xml:

```
<autoCommit>
  <maxTime>15000</maxTime>
  <openSearcher>false</openSearcher>
</autoCommit>

<!--
  <autoSoftCommit>
    <maxTime>1000</maxTime>
  </autoSoftCommit>
-->
```


LucidWorks™

Solr is NoSQL

- Update durability
  - A transaction log ensures that even uncommitted documents are never lost.
- Real-time Get
  - The ability to quickly retrieve the latest version of a document, without the need to commit or open a new searcher
- Versioning and Optimistic Locking
  - combined with real-time get, this allows read-update-write functionality that ensures no conflicting changes were made concurrently by other clients.
- Atomic updates
  - the ability to add, remove, change, and increment fields of an existing document without having to send in the complete document again.
- Real-time /get combined with SolrCloud make a very powerful key/value pair database

ṀÊŢĂḌÂṬÄ

Future

- JSON Query Parser
  - https://issues.apache.org/jira/browse/SOLR-4351
- Shard splitting
  - https://issues.apache.org/jira/browse/SOLR-3755

Credits

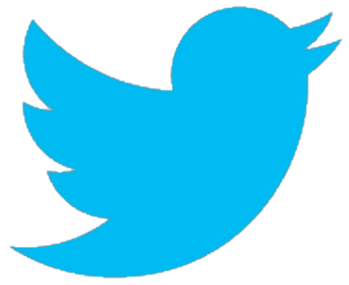
- LucidWorks
  - lucidworks.com
- Manning Publications
  - manning.com/lucene
- Apache Software Foundation
  - apache.org
- Apache Lucene
  - lucene.apache.org

LucidWorks™

Contact Info

- IRC: erikhatcher
- **erik** *dot* **hatcher @ lucidworks** *dot* **com**
- @ErikHatcher
- http://searchhub.org/author/erik/
- http://erikhatcher.tumblr.com/

Get at me...



@ErikHatcher

LucidWorks™