



Lucene/Solr 4: A Revolution in Enterprise Search Technology

2013 March 27

Webinar presented by Erik Hatcher

Search | Discover | Analyze



Talk Description

- Lucene/Solr 4 is a ground breaking shift from previous releases. Solr 4.0 dramatically improves scalability, performance, reliability, and flexibility. Lucene 4 has been extensively upgraded. It now supports near real-time (NRT) capabilities that allow indexed documents to be rapidly visible and searchable. Additional Lucene improvements include pluggable scoring, much faster fuzzy and wildcard querying, and vastly improved memory usage.
- The improvements in Lucene have automatically made Solr 4 substantially better. But Solr has also been considerably improved and magnifies these advances with a suite of new “SolrCloud” features that radically improve scalability and reliability.

Takeaways

- What are the Key Feature Enhancements of Lucene/Solr 4, including the new distributed capabilities of SolrCloud
- How to Use the Improved Administrative User Interface
- How Sharding has been improved
- What are the improvements to GeoSpatial Searches, Highlighting, Advanced Query Parsers, Distributed search support, Dynamic core management, Performance statistics, and searches for rare values, such as Primary Key

- What are the Key Feature Enhancements of Lucene/Solr 4, including the new distributed capabilities of SolrCloud?
 - as of Lucene/Solr 4.2(.1)

Key Feature Enhancements of Lucene 4

- Flexible Index Formats

- New posting list codecs: Block, Simple Text, Append (HDFS..), etc
- Pulsing codec: improves performance of primary key searches, inlining docs, positions, and payloads, saves disk seeks

- Pluggable Scoring

- Decoupled from TF/IDF
- Built in alternatives include BM25 & DFR
 - » http://en.wikipedia.org/wiki/Okapi_BM25
 - » http://terrier.org/docs/v3.5/dfr_description.html

- Faster fuzzy & wildcard query processing

- Higher performance searching

- String -> BytesRef

- Much improved data structure
- ... means more less memory and less garbage collection effort

Key Feature Enhancements of Lucene, continued

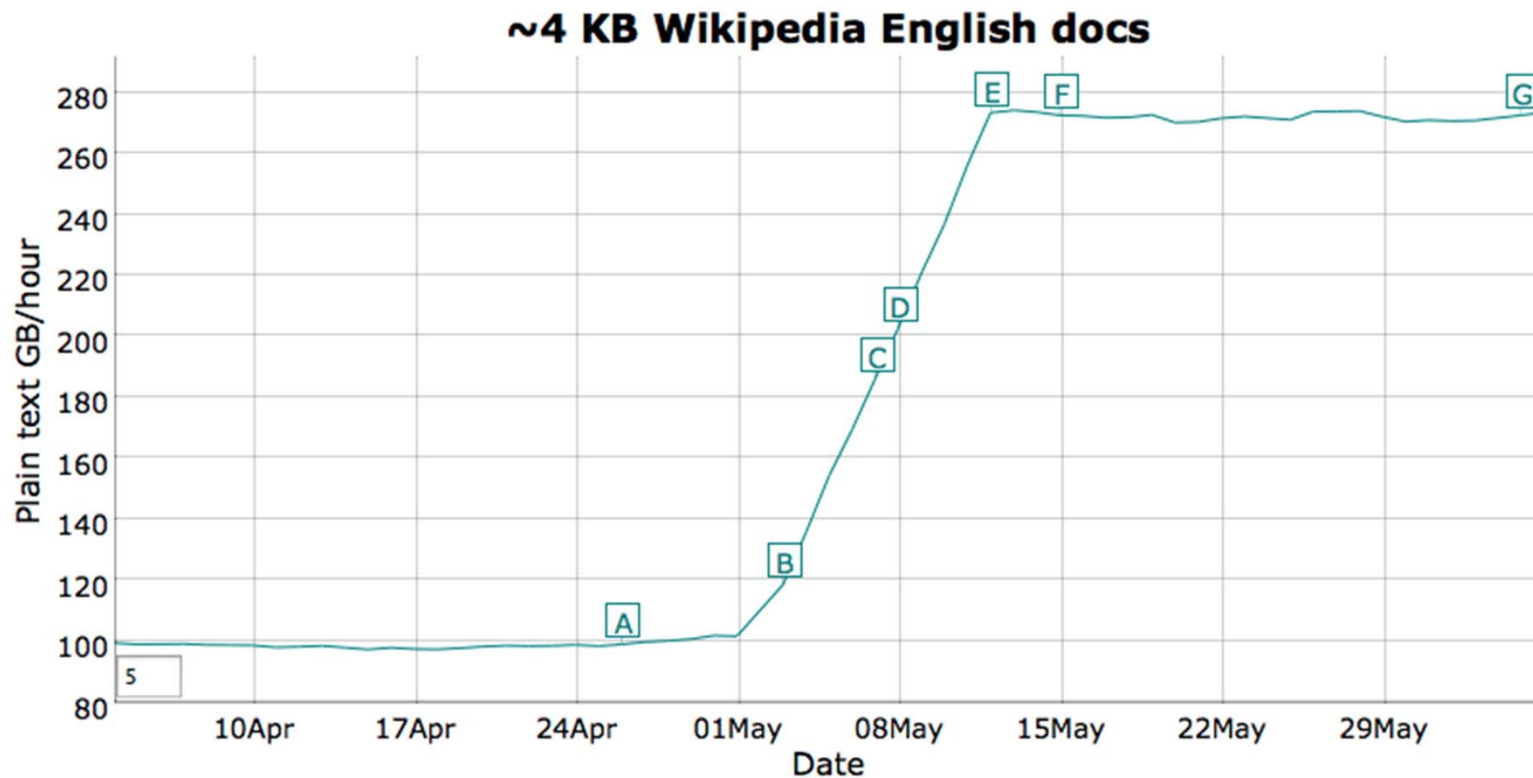
- NRT
 - Per segment
 - » FieldCache can be controlled to only load new segments
 - » Soft commit: faster without fsync, allows quicker update visibility
- DWPT (Document Writer per Thread)
 - Faster more consistent index speed
- DocValues: aka column-stride fields
- DirectSpellChecker
 - Uses main search index directly
- Geospatial Overhaul (covered later)
 - Search in polygonal areas
 - Great for location proximity searching

BytesRef memory management improvements

- On a Wikipedia index (11M documents)
 - Time to perform the first query with sorting (no warmup queries)
Solr 3x: 13 seconds, Solr 4: 6 seconds.
 - Memory consumption Solr 3x: 1,040M, Solr 4: 366M. Yes, almost a 2/3 reduction in memory use. And that's the entire program size, not counting memory used to just start Solr and Jetty running.
 - Number of objects on the heap. Solr 3x: 19.4M, Solr 4: 80K. No, that's not a typo. There are over two orders of magnitude fewer objects on the heap in trunk!
 - (data from an Erick Erickson blog entry)

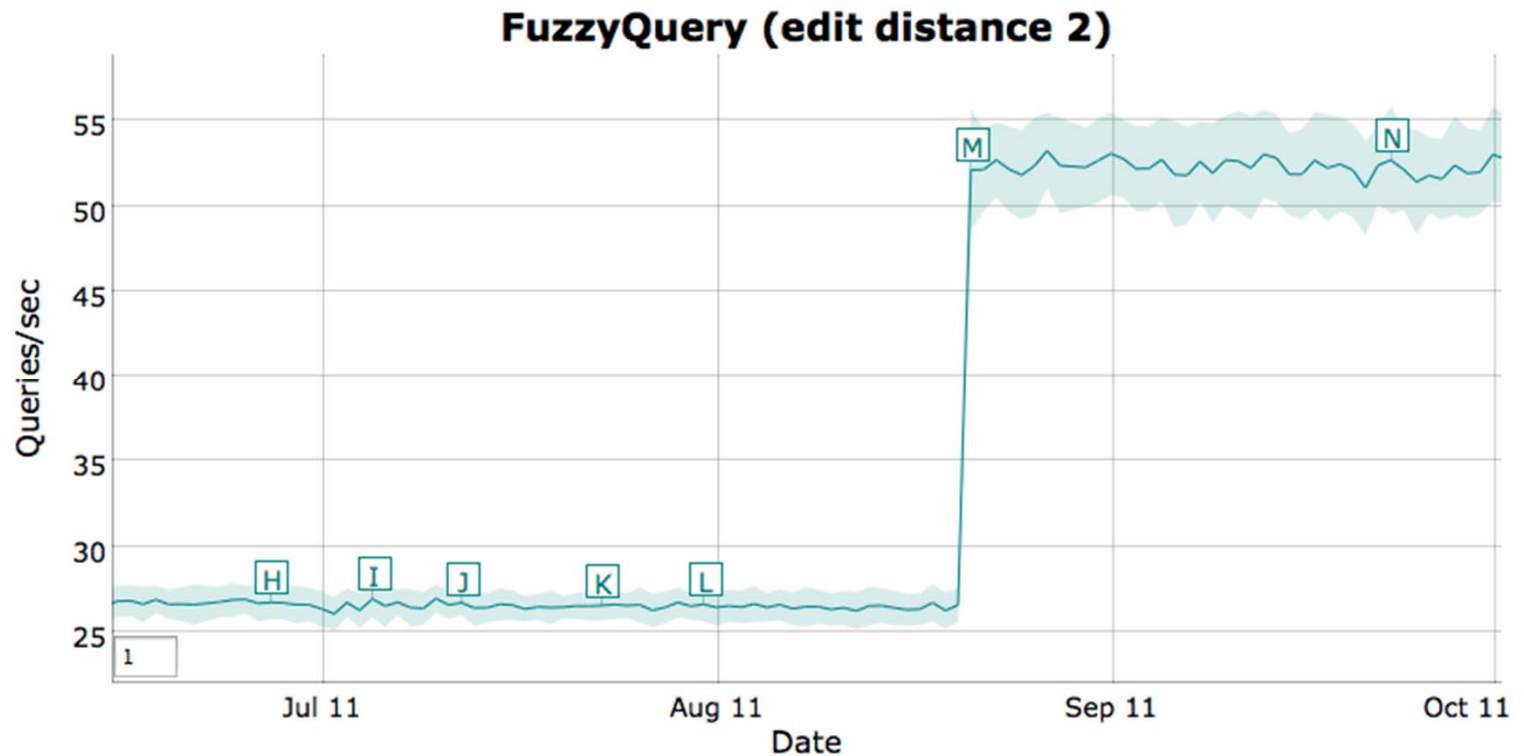
Indexing performance (Wikipedia 4KB docs)

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



FuzzyQuery

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



Key Feature Enhancements of Solr 4

- SolrCloud: large, manageable, scaling
- Several new update processors, including a “script” one
- SolrJ streaming response
- Content-type savvy /update handler
- Improved document response: DocTransformer, function calculations
- Pivot faceting
- New relevancy functions
- DirectSpellChecker support
- Pseudo-join
- Improved geospatial capabilities (more detail later)
- NRT/transaction log
- Improved Admin UI
 - including SolrCloud cluster visualizations

SolrCloud

- Distributed/sharded indexing & search
 - Auto distributes updates and queries to appropriate shards
 - Near Real Time (NRT) indexing capable
- Dynamically scalable
 - New SolrCloud instances add indexing and query capacity
- Reliable
 - No single point of failure
 - Transactions logged
 - Robust, automatic recover

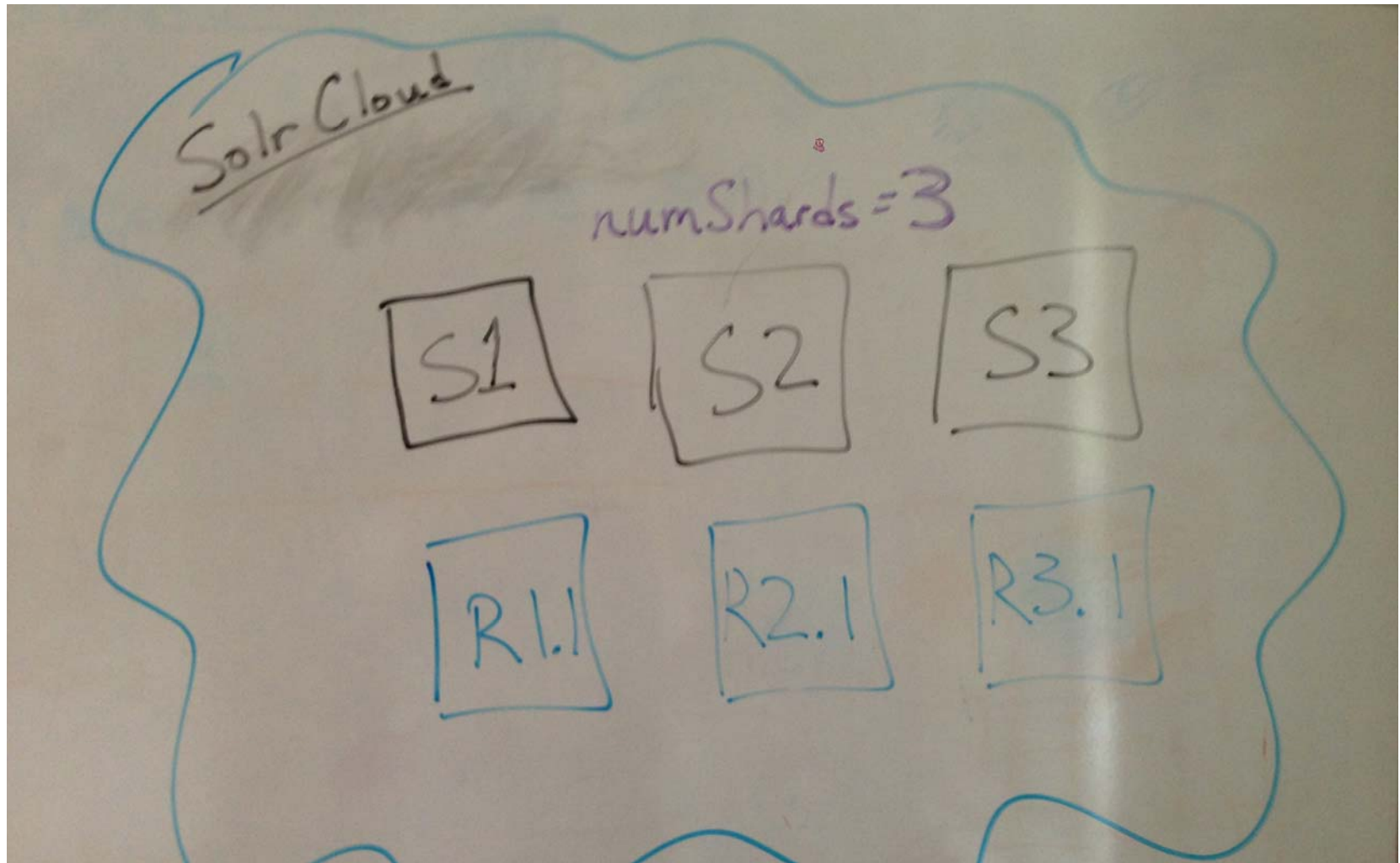
SolrCloud's capabilities

- Transaction log
 - All updates are added to the transaction log. The tlog provides support for: durability for updates that have not yet been committed, peer syncing, real-time get (retrieve documents by unique id) always up to date because it checks the tlog first, does not require opening a new searcher to see changes
- Near Real Time (NRT) indexing
 - Soft commits make updates visible
 - Hard commits make updates durable
- Durability
 - Updates to Solr may be in several different states: buffered in memory, flushed, but not committed or viewable, soft committed (flushed and viewable), committed (durable)
 - The transaction log ensures data is not lost in any of these states if Solr crashes.
- Recovery
 - Solr uses the transaction log for recovery; on startup Solr checks to see if the tlog is in a committed state, if not updates since the last commit are applied
- Optimistic locking
 - Solr maintains a document version (`_version_` field); updates can now specify `_version_`; updates to incorrect version will fail

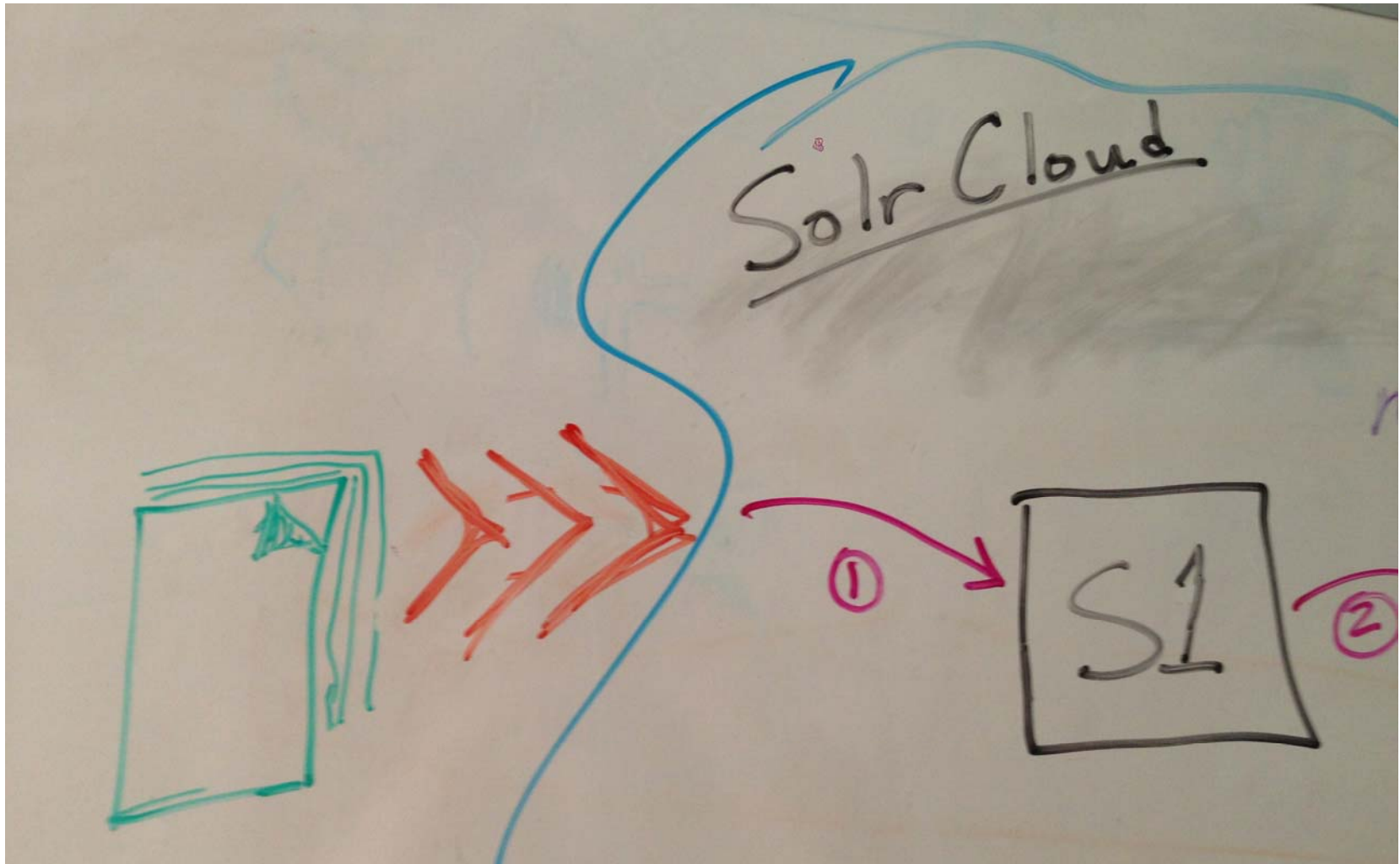
SolrCloud details

- “Leaders” and “replicas”
 - Leaders are automatically elected
- Leaders are just a replica with some coordination responsibilities for the associated replicas
- If a leader goes down, one of the associated replicas is elected as the new leader
- New nodes are automatically assigned a shard and role, and replicate/recover as needed
- SolrJ’s CloudSolrServer
- Replication in Solr 4
 - Used for new and recovering replicas
 - Or for traditional master/slave configuration

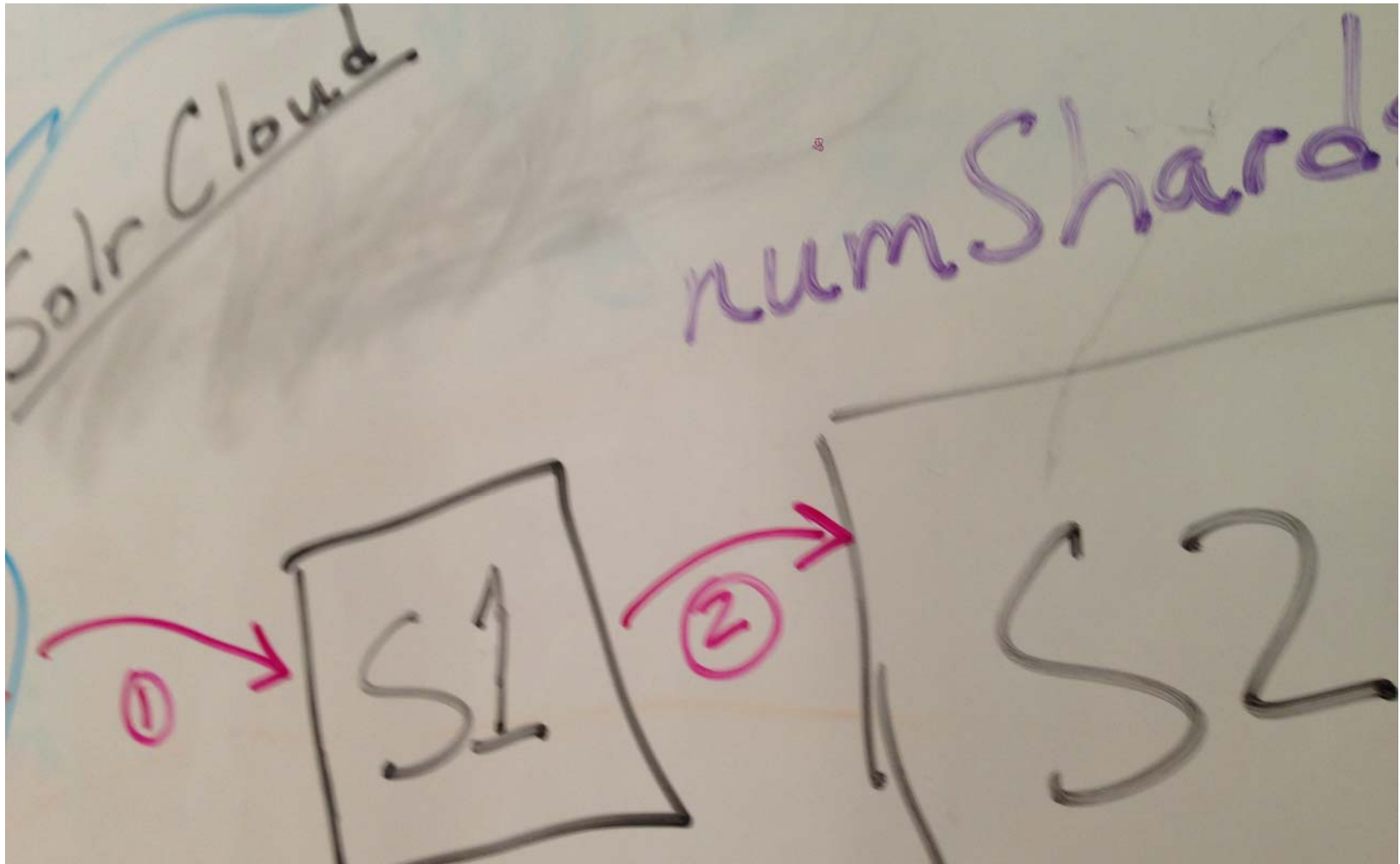
SolrCloud



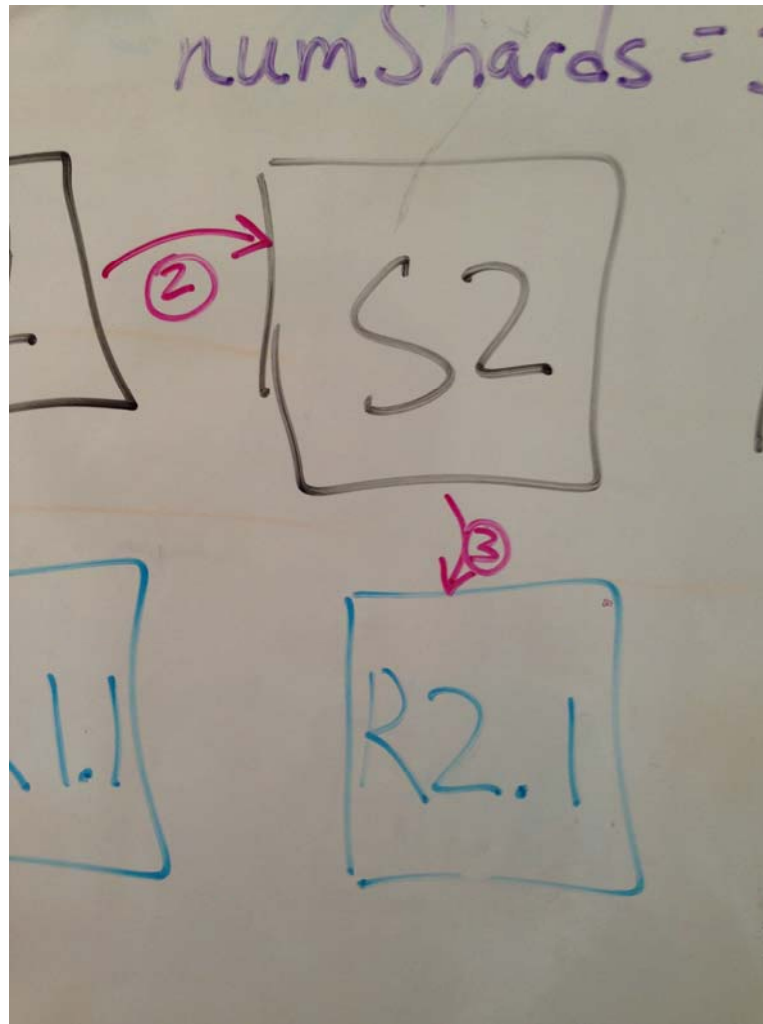
Document indexing



Routing to appropriate shard



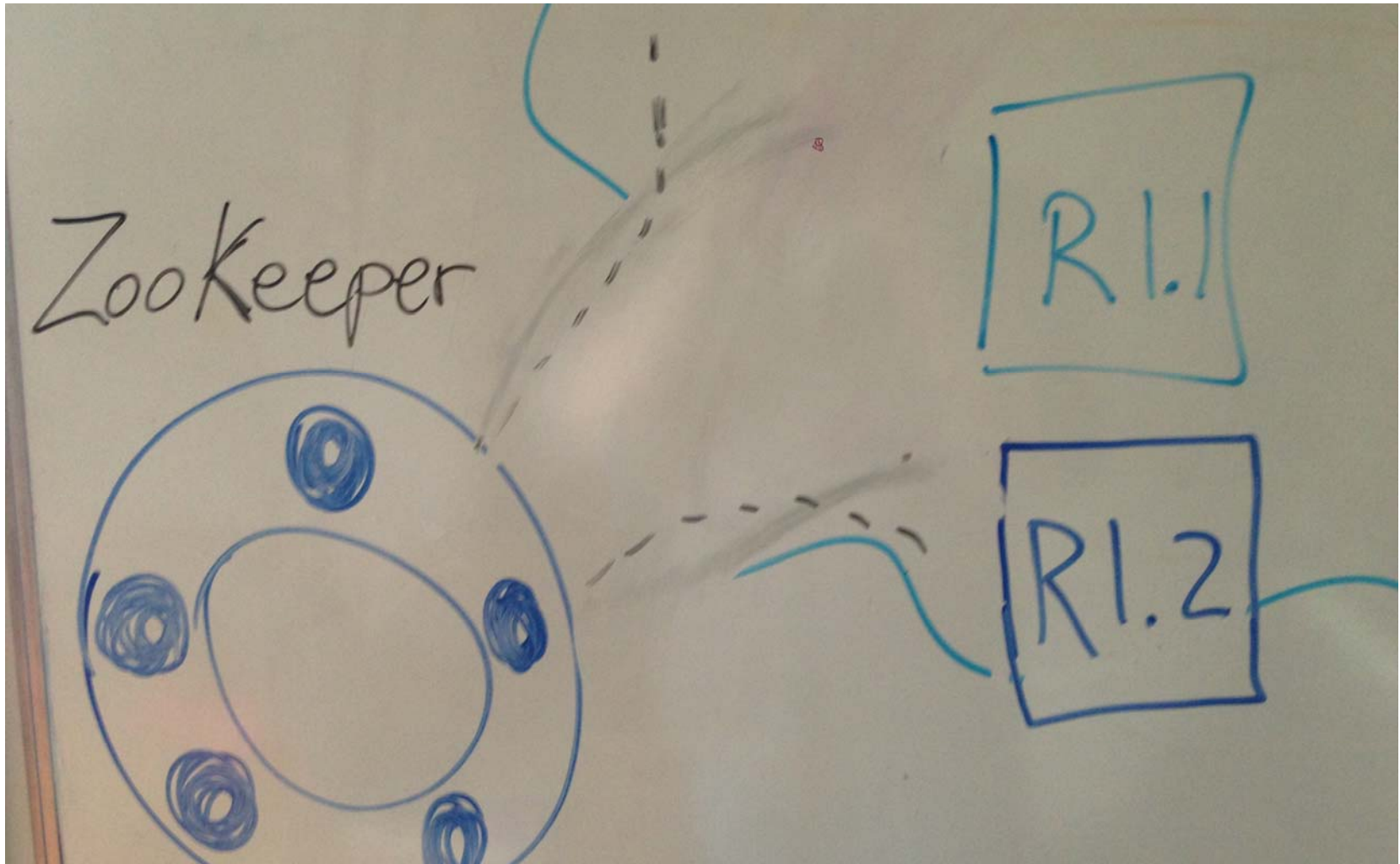
Replicating during indexing



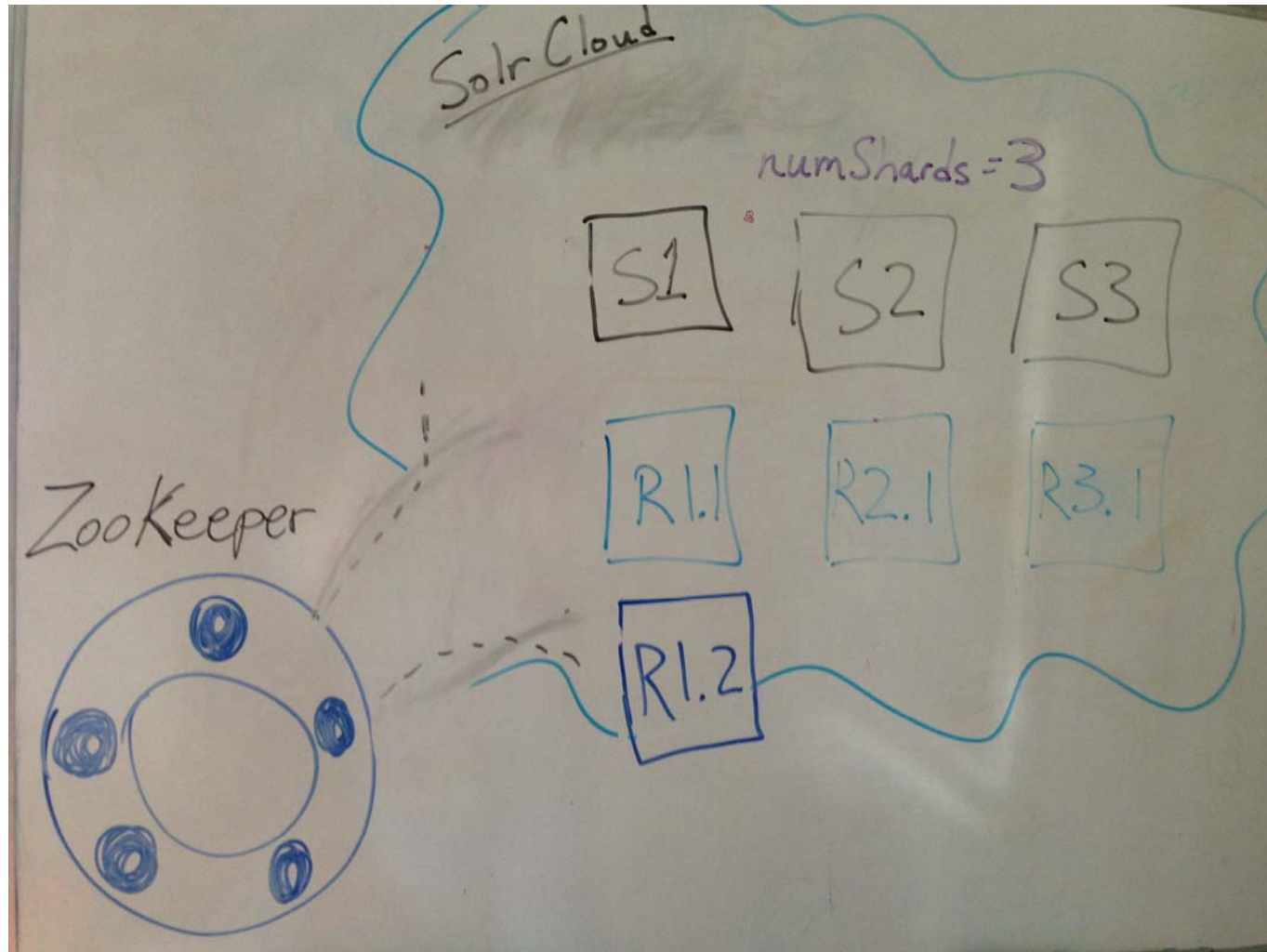
Increasing capacity or recovering



ZooKeeper



The Big (Data) Picture of SolrCloud

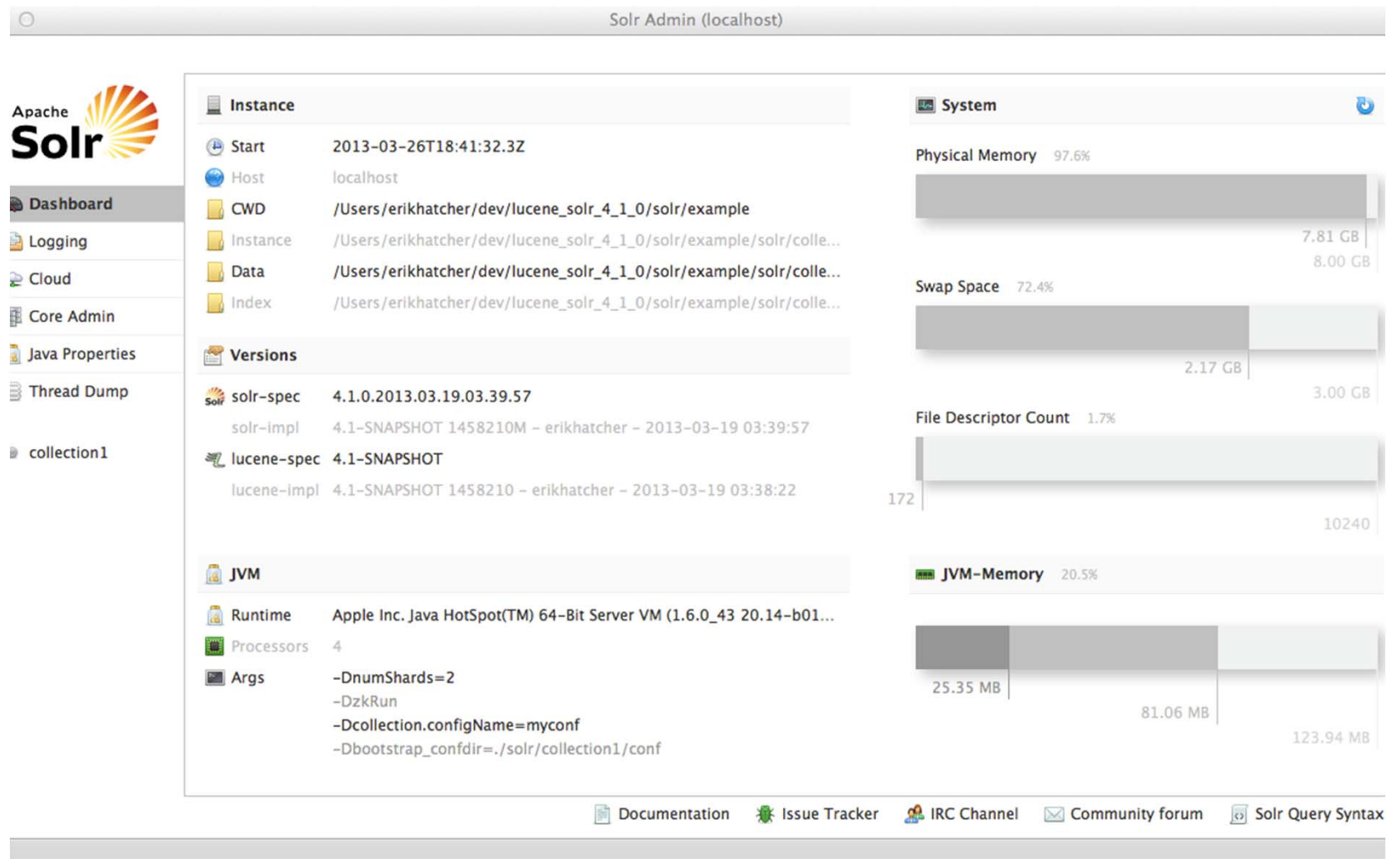


- How to Use the Improved Administrative User Interface


Solr Console

- View SolrCloud cluster state
- View collection schema and configuration
- Querying and analysis tools

/solr/#/



Core Admin



[Dashboard](#)
[Logging](#)
[Cloud](#)
[Core Admin](#)
[Java Properties](#)
[Thread Dump](#)

[collection1](#)

[+ Add Core](#)

[✖ Unload](#) [📄 Rename](#) [🔄 Swap](#) [🔄 Reload](#) [🔧 Optimize](#)

collection1

Core

startTime:

about 2 hours ago

instanceDir:

solr/collection1/

dataDir:

solr/collection1/data/

Index

lastModified:

about 2 hours ago

version:

7

numDocs:

20

maxDoc:

20

deletedDocs:

-

optimized:

✓

current:

✓

directory:

org.apache.lucene.store.NRTCachingDirectory:NRTCachingDirectory(org.apache.lucene.store.NIOFSDirectory@/Users/erikhatcher/dev/lucene_solr_4_1_0/solr/example/solr/collection1/data/index
lockFactory=org.apache.lucene.store.NativeFSLockFactory@20a83c2a; maxCacheMB=48.0
maxMergeSizeMB=4.0)

[Documentation](#)


[Issue Tracker](#)

[IRC Channel](#)

[Community forum](#)

[Solr Query Syntax](#)

Collection View



Dashboard

Logging

Cloud

Core Admin

Java Properties

Thread Dump

collection1

Ping

Query

Schema

Config

Replication

Analysis

Schema Browser

Plugins / Stats

Dataimport

Statistics

Last Modified: about 2 hours ago


Num Docs: 20


Max Doc: 20

Deleted Docs: 0

Version: 7

Segment Count: 3

Optimized: 

Current: 

Replication (Master)

	Version	Gen	Size
Master:	1364323501762	5	21.73 KB

Healthcheck


Ping request handler is not configured with a healthcheck file.


Admin Extra


Dataimport


Dataimport is not configured

Documentation


 Issue Tracker

 IRC Channel

 Community forum

 Solr Query Syntax

Analysis



- Dashboard
- Logging
- Cloud
- Core Admin
- Java Properties
- Thread Dump
- collection1**
 - Ping
 - Query
 - Schema
 - Config
 - Replication
 - Analysis**
 - Schema Browser
 - Plugins / Stats
 - Dataimport

Solr Admin (localhost)

Field Value (Index)

Lucene/Solr Revolution

Field Value (Query)

Analyse Fieldname / FieldType:

text_en

?


☒ Verbose Output

Analyse Values

ST	text	Lucene	Solr	Revolution
	raw_bytes	[4c 75 63 65 6e 65]	[53 6f 6c 72]	[52 65 76 6f 6c 75 74 69 6f 6e]
	start	0	7	12
	end	6	11	22
	type	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>
	position	1	2	3
SF	text	Lucene	Solr	Revolution
	raw_bytes	[4c 75 63 65 6e 65]	[53 6f 6c 72]	[52 65 76 6f 6c 75 74 69 6f 6e]
	position	1	2	3
	start	0	7	12
	end	6	11	22
	type	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>
LCF	text	lucene	solr	revolution
	raw_bytes	[6c 75 63 65 6e 65]	[73 6f 6c 72]	[72 65 76 6f 6c 75 74 69 6f 6e]
	position	1	2	3
	start	0	7	12
	end	6	11	22
	type	<ALPHANUM>	<ALPHANUM>	<ALPHANUM>
EPF	text	lucene	solr	revolution

Schema Browser

Solr Admin (localhost)



- Dashboard
- Logging
- Cloud
- Core Admin
- Java Properties
- Thread Dump
- collection1**
 - Ping
 - Query
 - Schema
 - Config
 - Replication
 - Analysis
 - Schema Browser**
 - Plugins / Stats
 - Dataimport

text

Field
text
Copied from
author
cat
content
content_type
description
features
includes
keywords
manu
name
name
resourcename
title
url
Type
text_general


Unique Key Field
id

Default Search Field

Field: text

Field-Type: org.apache.solr.schema.TextField
Properties: Indexed, Tokenized, Multivalued
Schema: Indexed, Tokenized, Multivalued
Index: (unstored field)
PI Gap: 100
Docs: 16

? Index Analyzer: org.apache.solr.analysis.TokenizerChain ✓
? Query Analyzer: org.apache.solr.analysis.TokenizerChain ✓

 Load Term Info

☐ Autoload

10 / -1 Top-Terms: ?

7 book electronics

6 inc

4 the memory

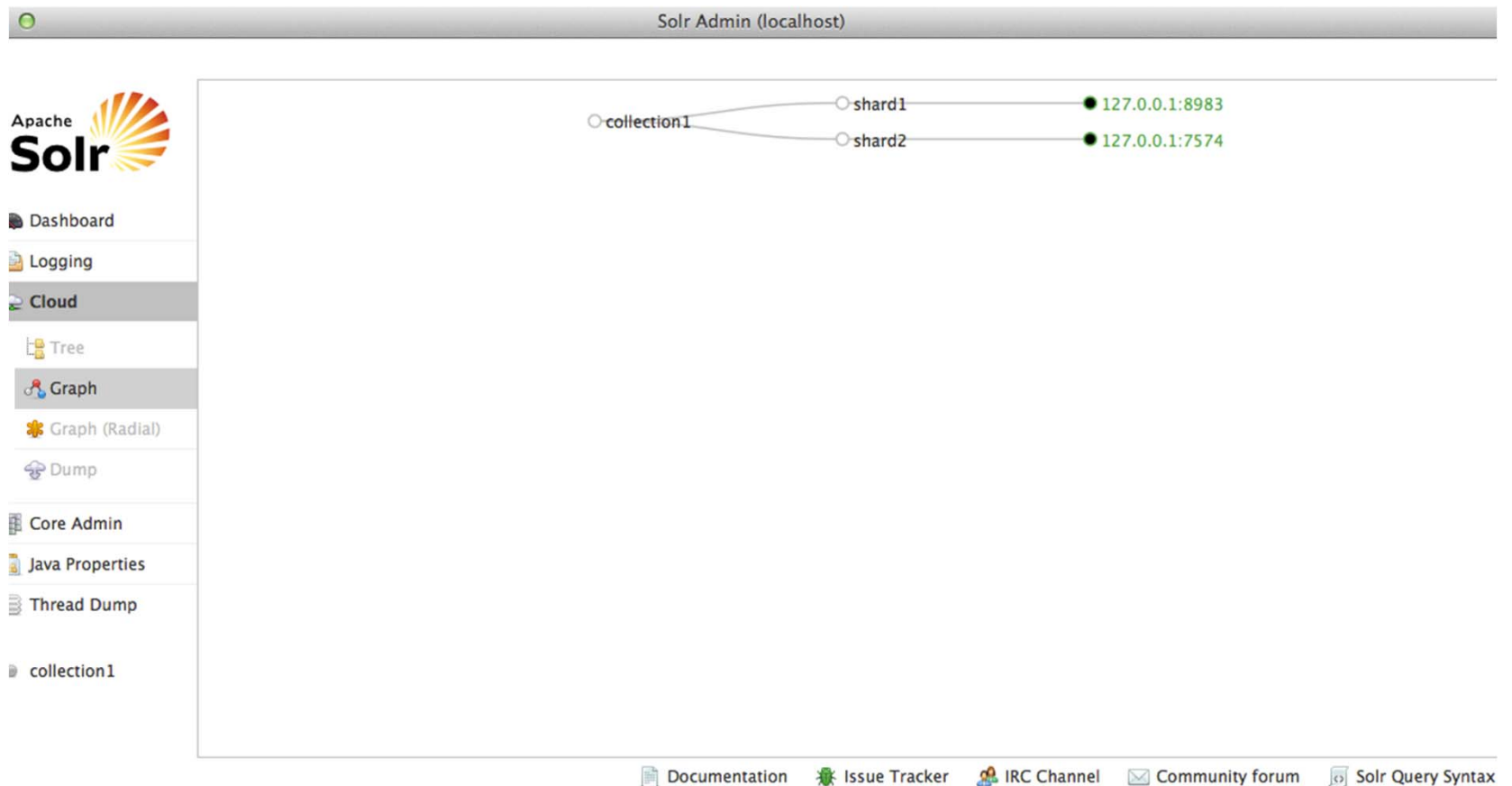
3 1gb 184 ddr 400 2.0

Histogram:

SolrCloud: tree view

The screenshot displays the Solr Admin (localhost) interface. On the left is a sidebar with the Apache Solr logo and a menu of navigation options: Dashboard, Logging, Cloud (selected), Tree (selected), Graph, Graph (Radial), Dump, Core Admin, Java Properties, Thread Dump, and collection1. The main content area shows a hierarchical tree view of the SolrCloud configuration. The root is '/', which contains several sub-items: /clusterstate.json, /collections (which contains collection1), /configs (which contains myconf), /live_nodes (which contains two nodes: 127.0.0.1:7574_solr and 127.0.0.1:8983_solr), /overseer, /overseer_elect, and /zookeeper. At the bottom of the interface is a footer with links to Documentation, Issue Tracker, IRC Channel, Community forum, and Solr Query Syntax.

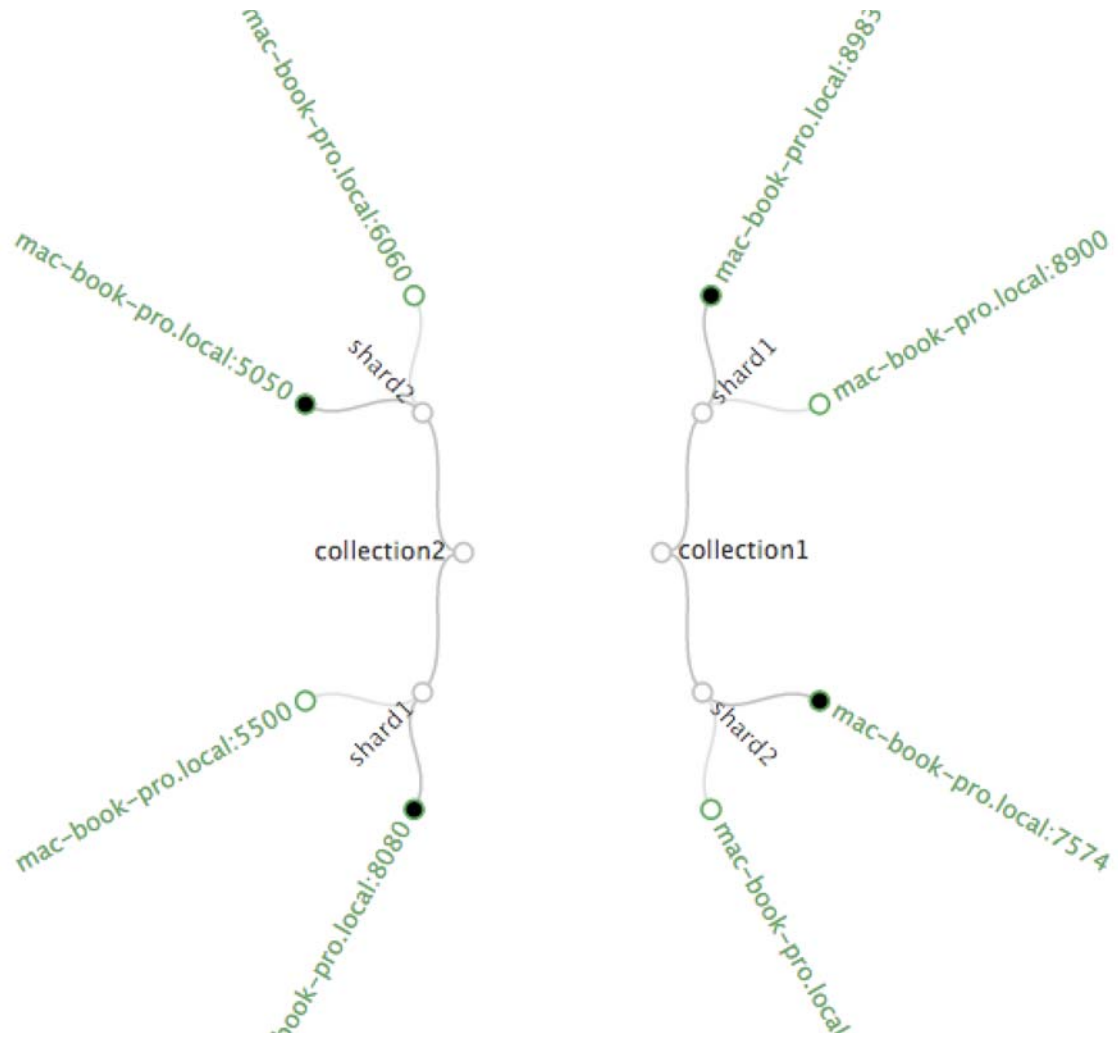
SolrCloud: graph view



SolrCloud: radial

SOLR

- Dashboard
- Logging
- Cloud**
- Tree
- Graph
- Graph (Radial)**
- Dump
- Core Admin
- Java Properties
- Thread Dump
- collection1



- How Sharding has been improved

Routing

- Allows you to route documents and queries to a subset of shards
- Provides efficient multi-tenancy
- Indexing:
 - A shard key can be prepended to the unique document id: `shard_key!unique_id`
 - Documents with the same `shard_key` will reside on the same shard.
- Querying: `shard.keys=shard_key1!...`
 - Much more efficient than searching the entire collection.

What are the improvements to ... ?

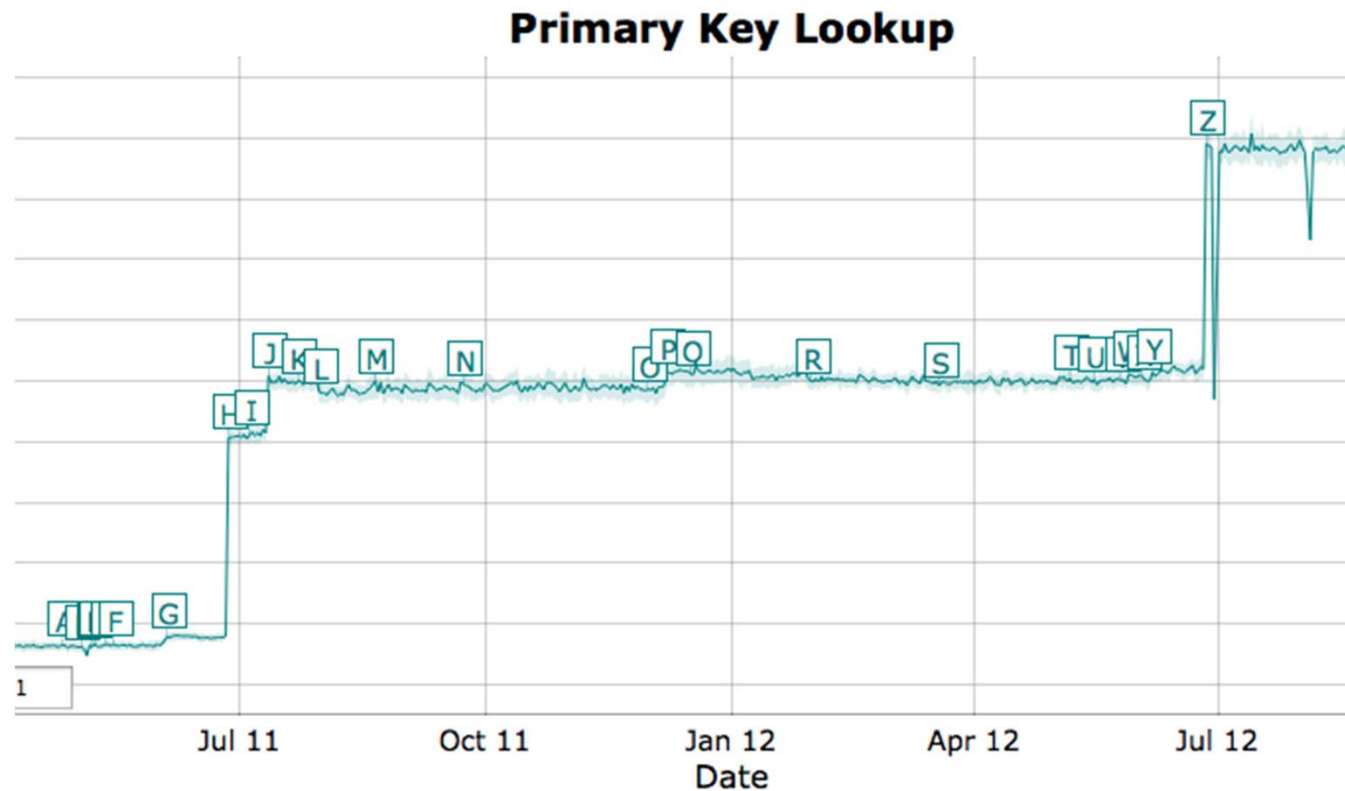
- Geospatial searches
- Highlighting
- Advanced query parsers
- Distributed search support
- Dynamic core management
- Performance statistics
- and searches for rare values, such as Primary Key

Geospatial improvements

- 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)))"

QPS (primary key lookup)

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



Solr as NoSQL

- Characteristics

- Non-traditional data stores
- Not designed for SQL type queries
- Distributed fault tolerant architecture
- Document oriented, data format agnostic(JSON, XML, CSV, binary)

- Updated durability via transaction log

- Real-time /get fetches latest version w/o hard commit

- Versioning and optimistic locking

- w/ Real Time GET, allows read/write/update w/o conflicts

- Atomic updates

- Can add/remove/change and increment a field in existing doc w/o re-indexing

Distributed Key / Value Pair Database

- Real-time Get combined with Solr Cloud make a very powerful key/value pair database
 - Durable (tlog)
 - Isolated (Optimistic locking)
 - Redundant (Solr Cloud Replicas)
 - Distributed & scalable (billions of keys, Solr Cloud Sharding)
 - Efficient Multi-tenant (Solr Cloud document routing, Solr 4.1)
 - Fast (milli-second response time, Pulsing Codec)
 - Real-time (tlog)

Looking ahead

- A couple of notable features on the horizon:
 - Automatic shard splitting
 - Query parsing: rich query tree control via JSON/XML
- Continually improving performance, scalability, and robustness

About LucidWorks

- LucidWorks Search
 - Lucene/Solr 4 powered
 - Rich connector framework for SharePoint, web crawling, etc
 - Built-in security support
- LucidWorks Big Data
 - Scalable classification, machine learning, analytics
- Lucene/Solr commercial support
- Consulting
- Training
- <http://www.lucidworks.com>

Thank You! / Q & A

- Learn More
 - www.SearchHub.org
 - www.LuceneRevolution.org
- Provide Feedback
 - <http://svy.mk/10MOIjg>
- Join the conversations
 - @LucidWorks
 - @LuceneSolrRev
 - @ErikHatcher