

NYC Apache Lucene/Solr Meetup



Agenda

- ▼ Welcome
- ▼ "Faster. Better. Solr! What to look for in Solr 1.4"
 - ▼ Yonik Seeley, Lucid Imagination
- ▼ How fast is it? Assessing Performance in Lucene and Solr
 - ▼ Mark Miller, Lucid Imagination
- ▼ Finding more than music: how MTV Networks drives Viacom entertainment brands with Solr search
 - ▼ Michael Rosencrantz, MTV Networks
- ▼ Lightning Talks



What's New In Solr 1.4

Yonik Seeley

Thinking Lucene ▼ Think Lucid.

Apache

Solr



Performance! Scalability/Concurrency!

▼ FastLRUCache – ConcurrentHashMap based

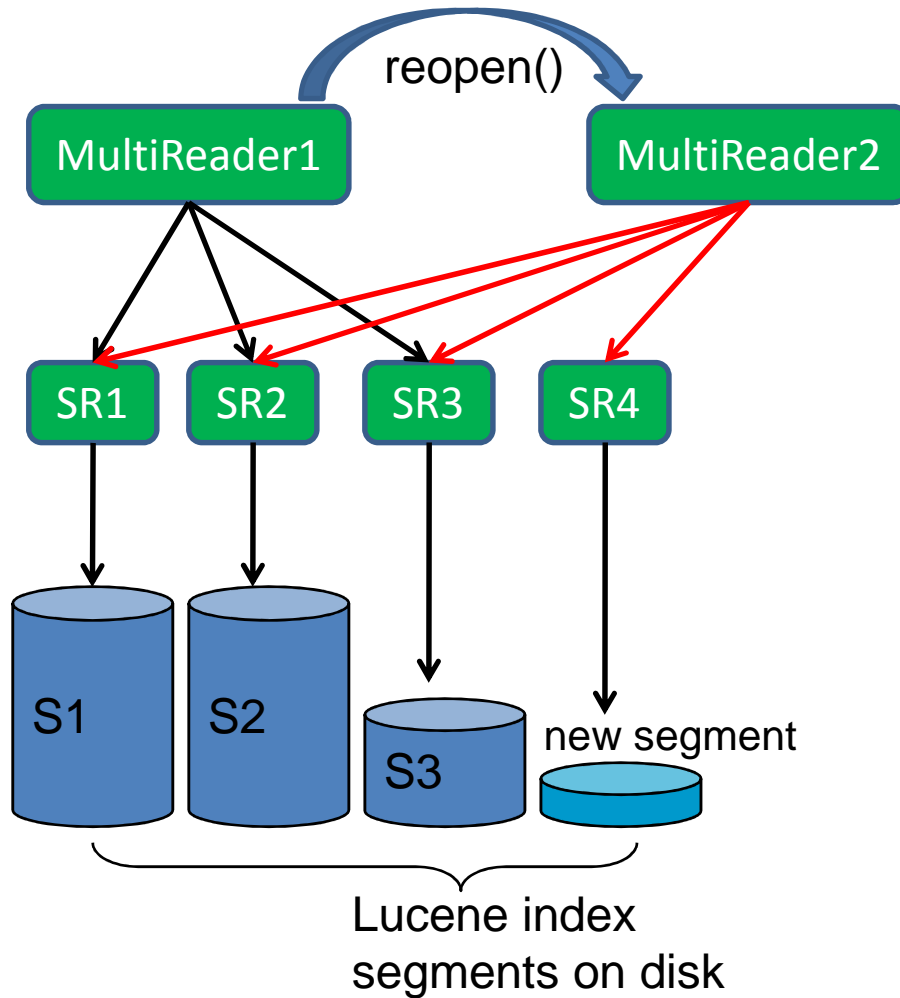
- ▼ Reads are lockless, writes are partitioned
- ▼ Can be slower if hit rate is low with few cores
- ▼ filterCache, queryCache, documentCache

▼ NIOFSDirectory!

- ▼ `sync{ seek(pos), read(nBytes) } => pread(pos, nBytes)`
- ▼ Windows still defaults to synchronized (JVM bug)

▼ <http://yonik.wordpress.com/2008/12/01/solr-scalability-improvements/>

Performance! IndexReader.reopen()



Field Cache
Un-inverted RAM resident

SR1 popularity	SR2 popularity	SR3 popularity
5	8	7
3	9	7
9	8	6
2	3	6
1	7	
8	5	
9	7	
6	4	

Performance! Faceting!

▼ New UnInvertedField (FieldCache-like method)

- ▼ Good for many unique terms, but relatively few values per doc
- ▼ Builds a doc-id => values mapping, for multi-valued fields
- ▼ Lots of tricks to reduce memory footprint
- ▼ Hybrid approach: filters used for “big” terms (>5% of index)
- ▼ Default for multi-valued fields
- ▼ **facet.method=enum** switches back to old behavior
- ▼ How big is it? Check out [admin/stats.jsp](#), go to **fieldValueCache**
- ▼ Result: up to 50x faster, 5x smaller (100K unique values, 1-5/doc)

Performance! TrieRangeQuery

▼ Trie* fields index multiple precisions

▼ Works for numerics & dates... renamed NumericField in Lucene

▼ 175 is indexed as hundreds:1 tens:17 ones:175

▼ TrieRange:[154 TO 183] is executed as

tens:[16 TO 18] **OR** ones:[154 TO 159] **OR** ones:[181 TO 183]

▼ Result: up to 40x faster than standard range queries

▼ Configurable precision step

▼ Only for single valued fields!

▼ Not completely integrated into Solr yet (no faceting)

Performance!

Binary format for updates (no XML parsing)

- Use SolrJ, it's the default transfer syntax

SolrJ's StreamingUpdateSolrServer

- Streams multiple documents over multiple connections
- Simple test went from 231 docs/sec to 25000 docs/sec!

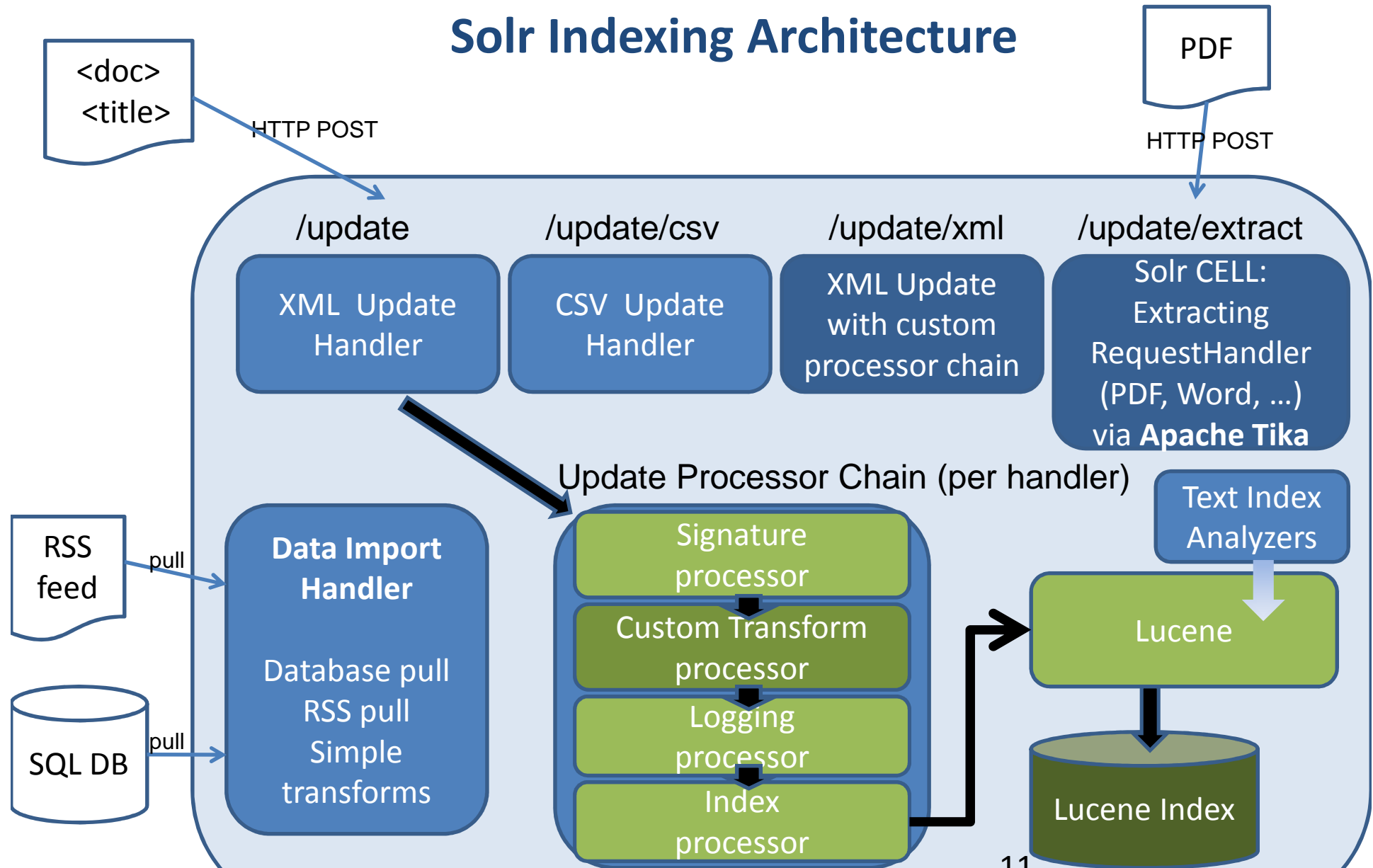
omitTermFreqAndPositions

- Omits number of terms in that specific field & list of positions
- Saves time and index space for non-text fields

Performance!

- ▼ **avoid scoring when generating docsets/filters**
 - ▼ Enabled by new Collector classes in Lucene
- ▼ **Filters now apply before main query**
 - ▼ [300% faster in some cases](#)
- ▼ **new small set filter implementation**
 - ▼ Used when cardinality < maxDoc/64
 - ▼ 40% smaller, good news for the filterCache
 - ▼ 60% faster at calculating intersections (facet.method=enum)

Solr Indexing Architecture



New update components

▼ Solr Cell (Content Extraction Library)

- ▼ Allow apps to send in Office, PDF, etc. and index it
- ▼ Integrates Apache Tika (v0.4) into Solr
- ▼ <http://wiki.apache.org/solr/ExtractingRequestHandler>

▼ SignatureUpdateProcessor

- ▼ Detect duplicates during indexing and handle them
- ▼ Adds a signature field to the document (could be uniqueKey)
- ▼ Exact (hash on certain fields) or Fuzzy duplicate detection
- ▼ <http://wiki.apache.org/solr/Deduplication>

Replication

▼ Old:

- ▼ UNIX only
- ▼ Difficult/Annoying to setup

▼ New

- ▼ See <http://wiki.apache.org/solr/SolrReplication>
- ▼ Java-based, self contained
- ▼ Replication of configuration files!
- ▼ Simple configuration

▼ Master:

```
<requestHandler name="/replication" class="solr.ReplicationHandler" >
  <lst name="master">
    <str name="replicateAfter">commit</str>
    <str name="confFiles">schema.xml,stopwords.txt</str>
  </lst>
</requestHandler>
```

▼ Worker:

```
<requestHandler name="/replication" class="solr.ReplicationHandler">
  <lst name="slave">
    <str name="masterUrl">http://localhost:8983/solr/replication</str>
    <str name="pollInterval">00:00:60</str>
  </lst>
</requestHandler>
```

Multi-select support

The screenshot shows a search interface with two facets: PROJECT and SOURCE. The PROJECT facet is selected with checkboxes for Lucene (3040) and Solr (1850). The SOURCE facet is selected with checkboxes for Lucid (4), Wiki (16), Apache Lucene Web (3), and Email (4685). The Email facet is further broken down into user (3882) and dev (803). The search results on the right show a search for 'index replication' with 4,887 results, including a [WIKI] Solr Replication entry and a [WIKI] Collection entry.

Very generic support

- Ability to tag filters
- Ability to exclude certain filters when faceting, by tag

`q=index replication&facet=true
&fq={!tag=proj}project:(lucene
OR solr)`

`&facet.field={!ex=proj}project
&facet.field={!ex=src}source`

<http://search.lucidimagination.com>

New Request Handler Components

▼ ClusteringComponent

- ▼ Uses Carrot2 to dynamically cluster the top N search results
- ▼ Like dynamically discovers facets

▼ Terms Component

- ▼ Return indexed terms+docfreq in a field, use for auto-suggest, etc

▼ TermVector Component

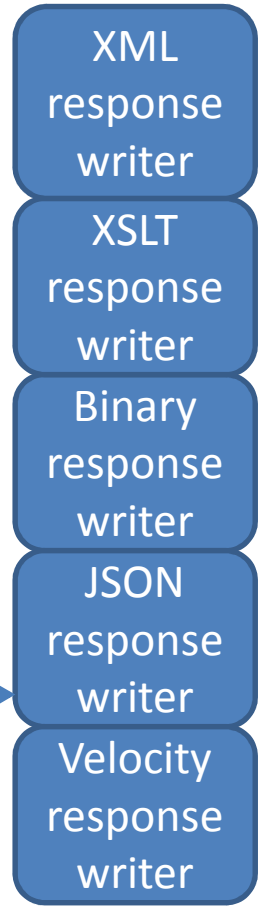
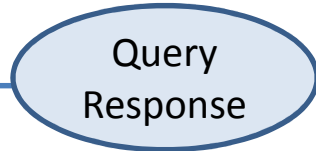
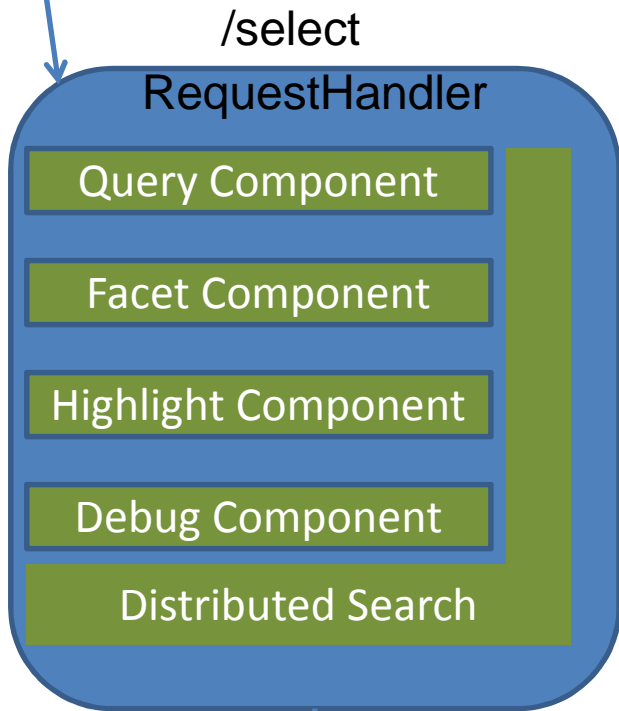
- ▼ Returns term info per document (tf, positions)

▼ Stats Component

- ▼ min, max, sum, sumOfSquares, count, missing, mean, stddev

Solr Request Plugins

http://.../select?q=cheese&wt=json



```
{ "response" = {
  "docs" = {
```

Additional plug-n-play search components



Tons more new features!

- ▶ Ranges over arbitrary functions: `{!frange l=1 u=2}sqrt(sum(a,b))`
- ▶ Nested queries, for function queries too
- ▶ solrjs – javascript client library
- ▶ `commitWithin` – doc must be committed within x seconds
- ▶ Binary field type
- ▶ Merge one index into another
- ▶ SolrJ client for load balancing and failover
- ▶ Field globbing for some params: `hl.fl=*_text`
- ▶ Doublemetaphone, Arabic stemmer, etc
- ▶ `VelocityResponseWriter` – template responses using Velocity



Now it is much easier to find my plans to get Bugs Bunny with Solr. I am a super genius to use Solr!