Cassandra SF 2011

Solandra
Scaling Solr with Cassandra

http://github.com/tjake/Solandra

T Jake Luciani
@tjake

DataStax

Cassandra was Built for Search

- Facebook Inbox Search
- SuperColumn per user
  - User (key)
    - Term (SuperColumn)
      - » {Msg1, Freq}
      - » {Msg2, Freq}
      - » {Msg3, Freq}
- Pre-fetch the user index when click on the search box
- Individual index limited by memory :(
- Supports millions of *small* indexes :)

What is Lucene?

- core search library
  - Document analysis
  - Inverted Index Add/Delete
  - Query and Search (+title:apac* +date:[20100101 TO 20110101])
  - Extremely fast and optimized

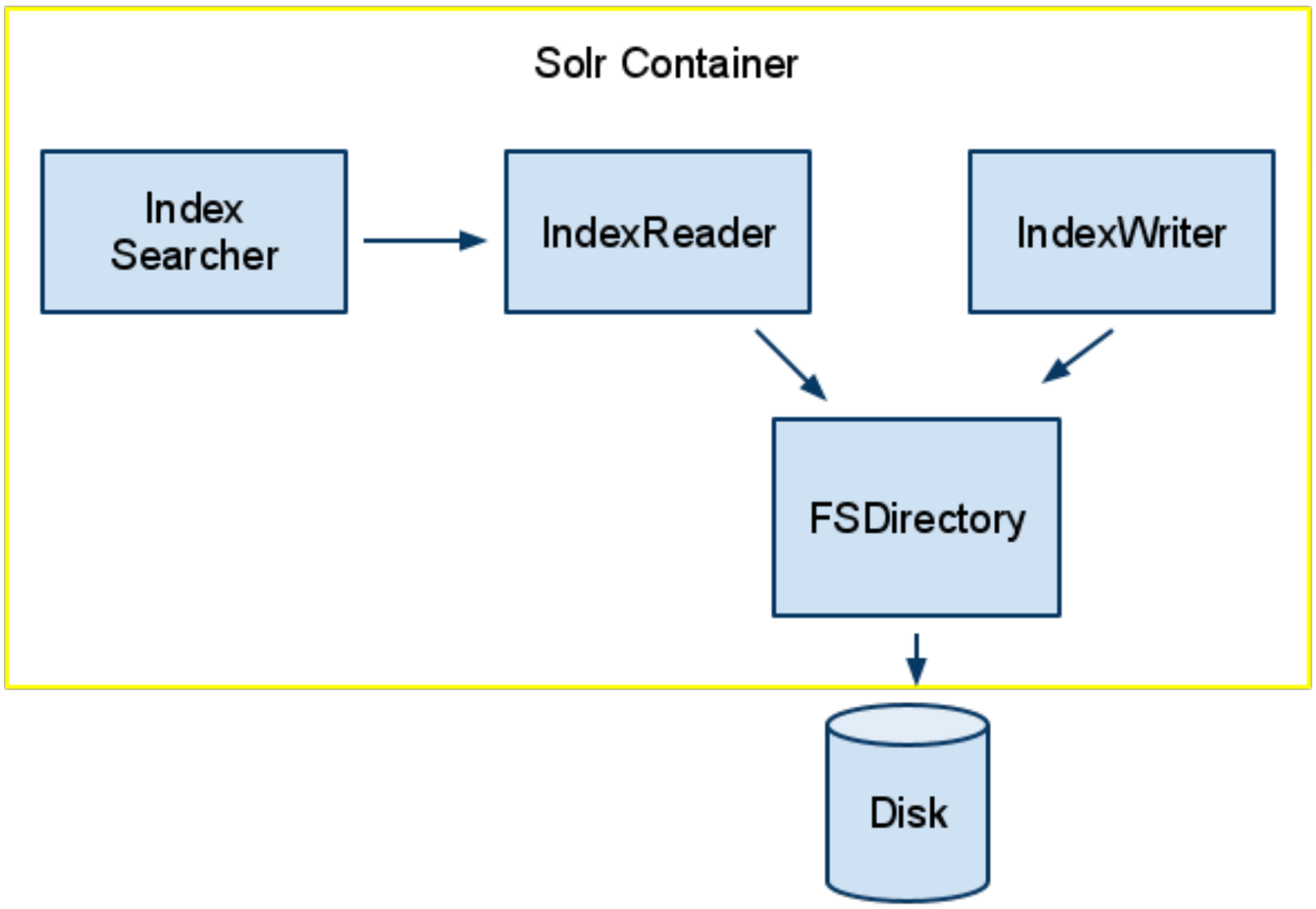
```
<doc id=”a”>
   <field name=”title”>apache talk</field>
   <field name=”date”>20101103</field>
</doc>
```

field	term	freq	position
title	apache	{a:1}	{a:0}
title	talk	{a:1}	{a:1}
date	20101103	{a:1}	{a:0}

What is Solr?

- HTTP service layer for lucene
  - Adds xml schema definitions
  - Advanced tokenizers and search features
    - Geo, Facets, Numeric types
  - Caching
  - Replication
  - Sharding

Solr Components

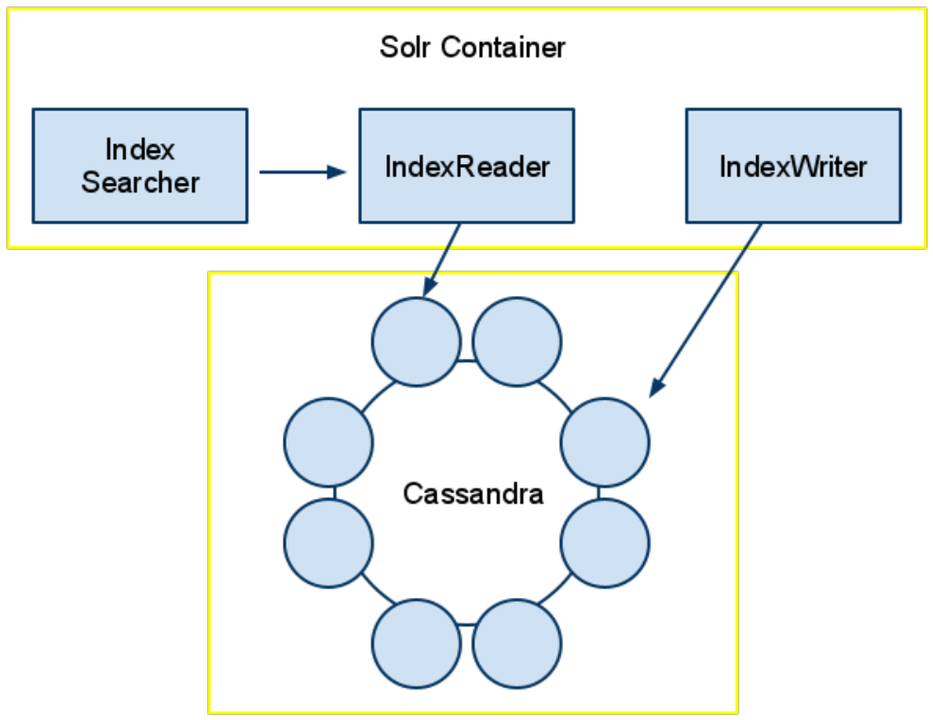


Solr Problems

- Not at all real-time
- All writes go through single master
- optimize() sucks
- Replication and Sharding is manual and requires careful planning for scaling and failover (think mysql)

Lucandra (2010)

- A Lucene IndexReader and Writer that communicates with Cassandra via Thrift
- Replaces Lucene index file format with Cassandra's data model.
- Multi-master
- Replication
- Real-time (no commit)
- Can manage millions of small indexes
- No more optimize



Lucandra Limitations :(

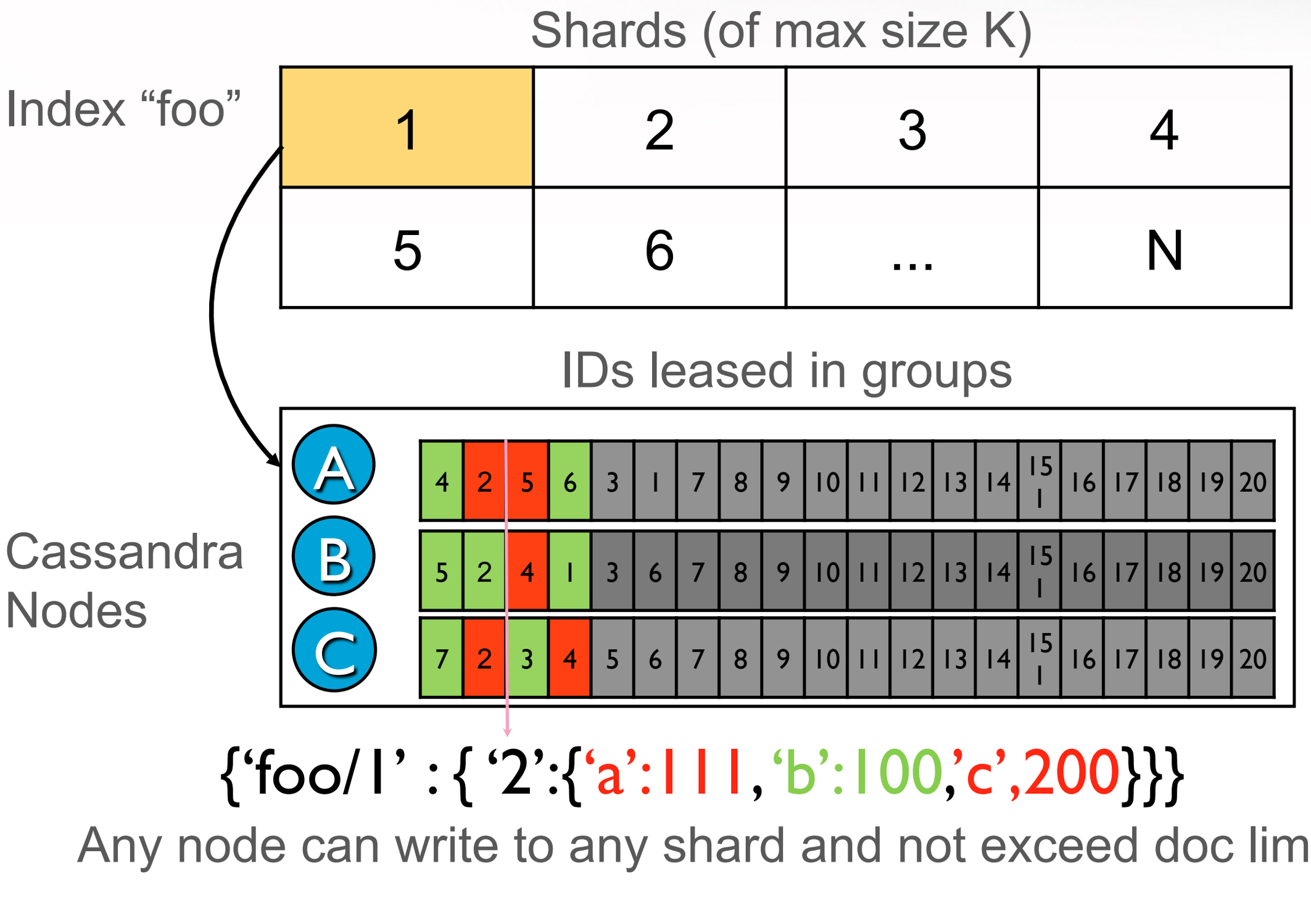
- Lucene and Cassandra communicate via Thrift. For large indexes with common terms this too much data! Queries with > 100k hits
- Lucene represents doc ids as absolute numbers (1-N). Lucandra refers to them by UUID. Mapping between them affects read performance
- Row scans require OrderPreservingPartitioner. Hard to balance data across nodes
- Solr doesn’t like schemas added on the fly

Try try again...

Solandra (2011)

- Embed Solr in Cassandra node
  - No RPC layer
  - Same JVM, in memory reads/writes (Shared Cache)
  - Solr becomes aware of Cassandra ring (Locality)
  - Manage N Solr Cores via Cassandra (REST API)
- IndexManager caps the number of docs in a given index. Indexes > cap are auto sharded.
- Keep all data for a sub-index on one node
  - Uses modified RandomPartitioner!
- Use Solr+C* ring to shuttle queries to nodes containing sub-indexes (using distributed solr api)
- Scale indexes with millions of docs. Profit!

Solandra Index Mgr



Any node can write to any shard and not exceed doc limit!

Solandra's Random Paritioner

- Extends RandomPartitioner but is aware of Solandra's composite row key structure:
  - SHA1(indexName~shard)/rowKey
- When encountered *ONLY* the SHA1 is used as token.
  - All docs of a index shard are on the same node.
- The bad news: Currently row scans broken because key to token lookup is not 1:1
- The good news: CASSANDRA-1600

Solandra ColumnFamilies

Doc (document info)

	field1	field2	meta
sha1(foo~1)/doc1	“like a boss”	“oh yea”	-

TermInfo (inverted terms)

	doc1
sha1(foo~1)/field1/boss	{freq:1, pos:2}

TermList (term range scans)

	field1/a	field1/boss	field1/like
sha1(foo~1)/t	-	-	-

FieldCache (first term per field: efficient sorts)

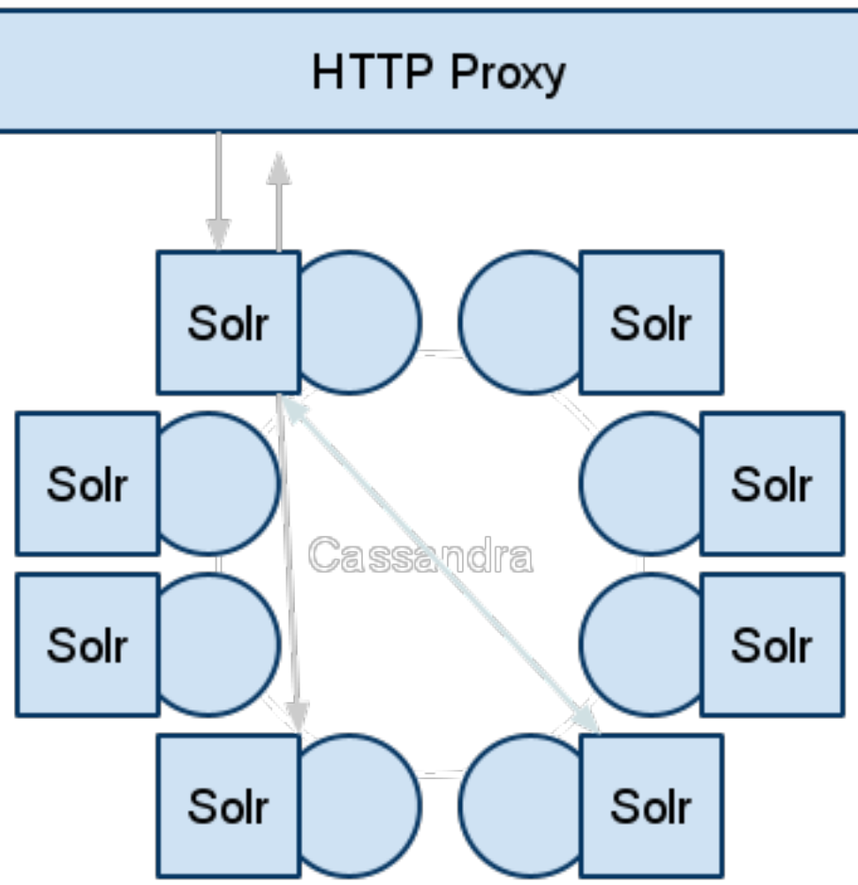
	doc1
sha1(foo~1)/field1	like

Solandra cores

I can't color without removing hyperlink

- Admin of a Solr core
  - POST http://host/solandra/schema/core/resource
  - GET http://host/solandra/schema/core
- Using a Solr core
  - (Write) POST http://host/solandra/core/update
  - (Read) GET http://host/solandra/core/select
- Solandra also has "Virtual" Cores /core.vcore/
  - Indexes that share the same core but internally a different index (think /inbox.jake/ and /inbox.matt/)

Solandra Deployed



Does it Scale?

- First large production deployment imminent
  - ~2 Billion documents
  - 400k Indexes
  - 10 Nodes
  - 5 TB of text

Reuters Demo


DataStax