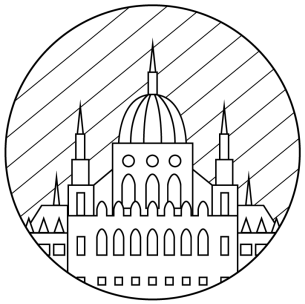
Flexible search in Apache Jackrabbit Oak

Tommaso Teofili



APACHECON EUROPE

CORINTHIA HOTEL
BUDAPEST, HUNGARY
NOVEMBER 17-21, 2014

Agenda

- Oak overview
- Indexing
- Querying
- Lucene, Solr, …

Apache Jackrabbit Oak

- Scalable content repository
- JCR 2.0
- Designed for concurrent access (MVCC)
- Pluggable components (storage, indexes)
- Powering AEM 6.0

Oak Architecture

- Oak-JCR
- Oak-Core
- Oak-MK

Oak – the Query Engine

- Search over the content repository
- Query languages
  - XPATH
  - SQL-2
- Looks for available indexes and selects the one(s) supposed to perform better
  - Search is demanded to the underlying indexes
  - No index? The repository is traversed

Indexing – the IndexEditor API

- NodeState before = builder.getNodeState();
- builder.child(”a").setProperty(”foo", ”bar");
- NodeState after = builder.getNodeState();
- NodeState indexed = editorHook.processCommit(before, after, …); // who said MVCC?

Searching – the QueryIndex API

- Filter filter = … ; // "select * from [nt:folder]"
- filter.restrictPath("/somenode", Filter.PathRestriction.DIRECT_CHILDREN);
- Cursor cursor = queryIndex.query(filter, nodeState); // search on a specific revision
- IndexRow row = cursor.next(); // result rows

Searching – Filters

- Full text expressions
- Property restrictions
- Path restrictions
  - Exact
  - Parent
  - Child
  - Descendant
- Node type restrictions

Configuring indexes

- Indexes are declared by adding “query index configuration” nodes in the repository
  - Type
  - Asynchronous
  - Reindex
  - Index specific properties

In repository indexes

- Property index
- Ordered property index
- Node type index
- Reference index

Lucene index

- Full text and (sorted) property restrictions
- Stored in repository
- DocValues for sorted property restrictions
- Tika for indexing binaries
- Configurable indexing rules (boost), codec, analyzers

Solr index

- Full text, property, path restrictions
- Embedded or remote Solr(Cloud)
- Configurable
  - Mapping restriction / fields
  - Page size
  - Commit policy
- Most is configured on the Solr side

Index selection

- Like in DBMSs the fastest (supposed) is selected
  - Cost
  - Index plan
- Different indexes may be selected for different parts of the same query

Problems

- Hard to express complex queries
- Cannot leverage underlying indexes advanced capabilities

Native language support

- Leverage underlying index capabilities
  - Multiple query languages/parsers
  - Advanced index capabilities (e.g. MLT)
- More accurate full text queries (and results)

Adding more indexes

- Create an IndexEditor
  - Turn diff into an “indexable”
- Create a QueryIndex
  - Turn a Filter into an index query

Looking forward

- Results aggregation features (e.g. facets)
- More configuration options (Lucene, Solr)


Q&A