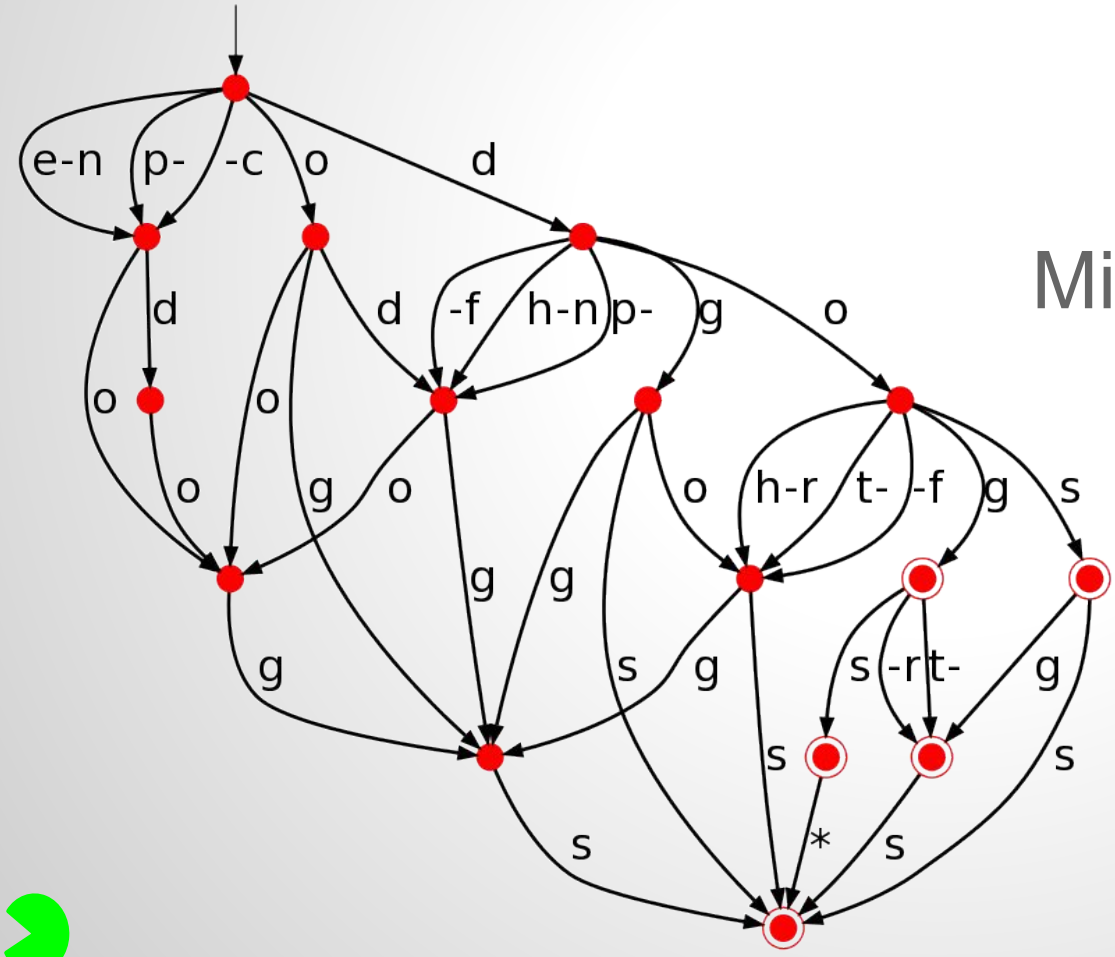
Automata Invasion



Michael McCandless
Robert Muir

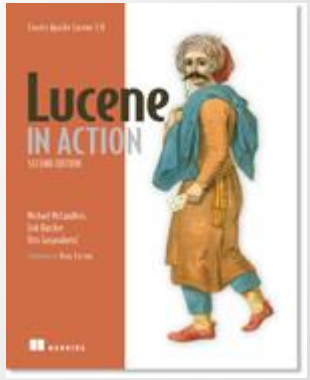
Lucene Revolution
May 9, 2012

Agenda

- Who we are
- Overview of finite state automata/transducers
- Three Lucene implementations
  - Automaton
  - FST
  - TokenStream
- Applications in Lucene

Who we are

- Lucene/Solr committer, PMC member
  - 6 years = old hat!
- Tika committer, PMC member
- Co-author Lucene in Action, 2nd ed.
- http://blog.mikemccandless.com
- Sponsored by IBM (thank you!)



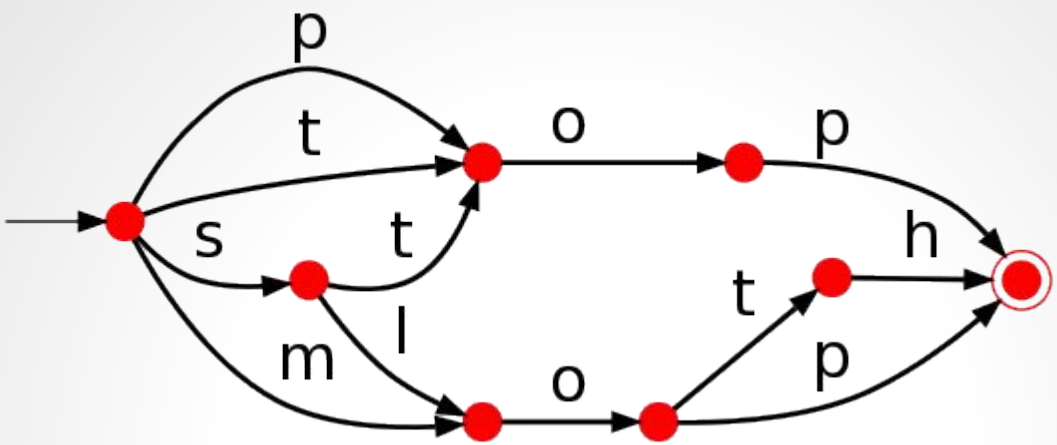
Who we are

- Lucene/Solr committer, PMC member
  - 3 years = young hat, yet: less hair!
- Lucid Imagination employee
- Background in internationalization
- Tangled up in finite state
  - NLP tasks
  - One of my first contributions to Lucene
- http://twitter.com/rcmuir

Agenda

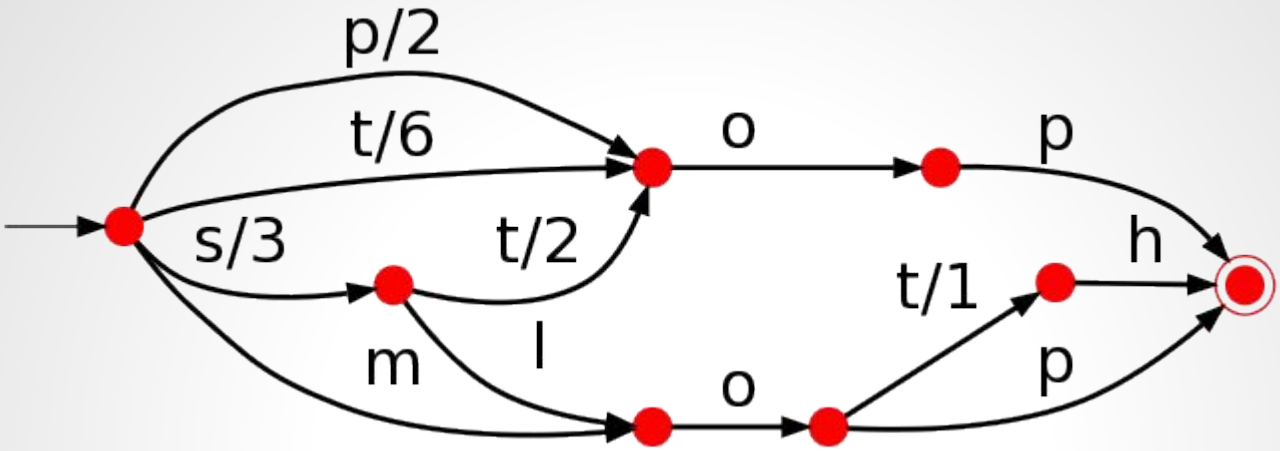
- Who we are
- Overview of finite state automata/transducers
- Three Lucene implementations
  - Automaton
  - FST
  - TokenStream
- Applications in Lucene

Finite State Automata (FSA)



- Set<int[]>:
  - mop, moth, pop, slop, sloth, stop, top
- Start node, final node
- Append arc labels as you traverse
- Traverse all paths to get all strings
- Determinize, minimize, union, intersect, ...

Finite State Transducer (FST)



- Adds optional output to each arc
  - Accumulate output as you traverse
- Map<int[],T>
  - mop: 0, moth: 1, pop: 2, slop: 3, sloth: 4, stop: 5, top: 6
- T is pluggable (defined by output algebra)
- Deep theory, many algorithms...

Agenda

- Who we are
- Overview of finite state automata/transducers
- Three Lucene implementations
  - Automaton
  - FST
  - TokenStream
- Applications in Lucene

Lucene's FSA Implementation

- org.apache.lucene.util.automaton.*
  - Poached from Anders Møller's Brics http://www.brics.dk/automaton
- Build up any FSA one node/arc at a time
  - Automaton, Transition, State
  - Transition has min/max label
- Many standard algorithms
  - Minimize
  - Determinize
  - Intersect
  - Union
  - Regexp -> Automaton
- RunAutomaton for stepping

Lucene's FST Implementation

- org.apache.lucene.util.fst.*
- FST encoded as a byte[]
- Write-once API
  - From Mihov & Maurel paper
  - Build minimal, acyclic FST from pre-sorted inputs
  - Fast (linear time with input size), low memory
  - Optional two-pass packing can shrink by ~25%
- Traverse FST one arc at a time
  - Decode byte[] during lookup
- SortedMap<int[],T>: arcs are sorted by label
  - getByOutput also possible if outputs are sorted
- http://s.apache.org/LuceneFSTs

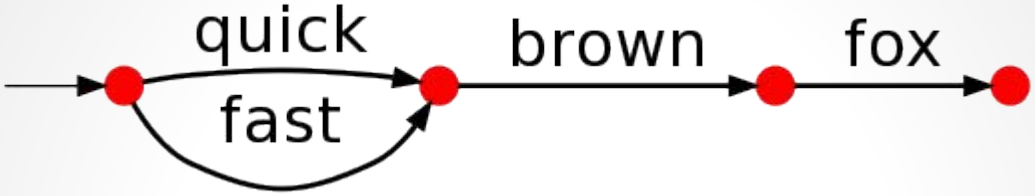
Lucene's FST Implementation

Downside: Generics Policeman does **not** approve!

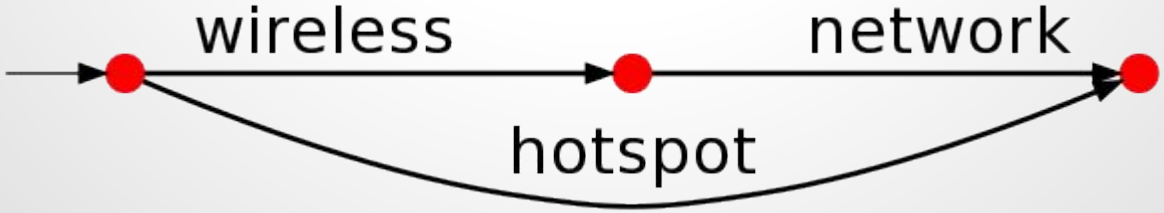
@UweSays "*I looked at the code, it was ununderstandable why this thing was generified*"

Lucene's TokenStreams are FSAs!

- Streaming FSA, one arc at a time



- New PositionLengthAttribute in 3.6.0
  - How many positions does the token "span"
  - TokenStreamToAutomaton
  - Indexer ignores it (sausage!)
  - http://s.apache.org/TokenGraphs



Lucene's TokenStreams are FSAs!

- Fixing all analyzers to behave with graphs?

Agenda

- Who we are
- Overview of finite state automata/transducers
- Three Lucene implementations
  - Automaton
  - FST
  - TokenStream
- Applications in Lucene

AutomatonQuery

- Automaton specifies which terms match
- New Terms.intersect(Automaton) method
  - Visits all matching terms & docs
- Example
  - RegexpQuery
  - WildcardQuery
  - FuzzyQuery
- Other possible uses
  - Stemming at query time via expansion

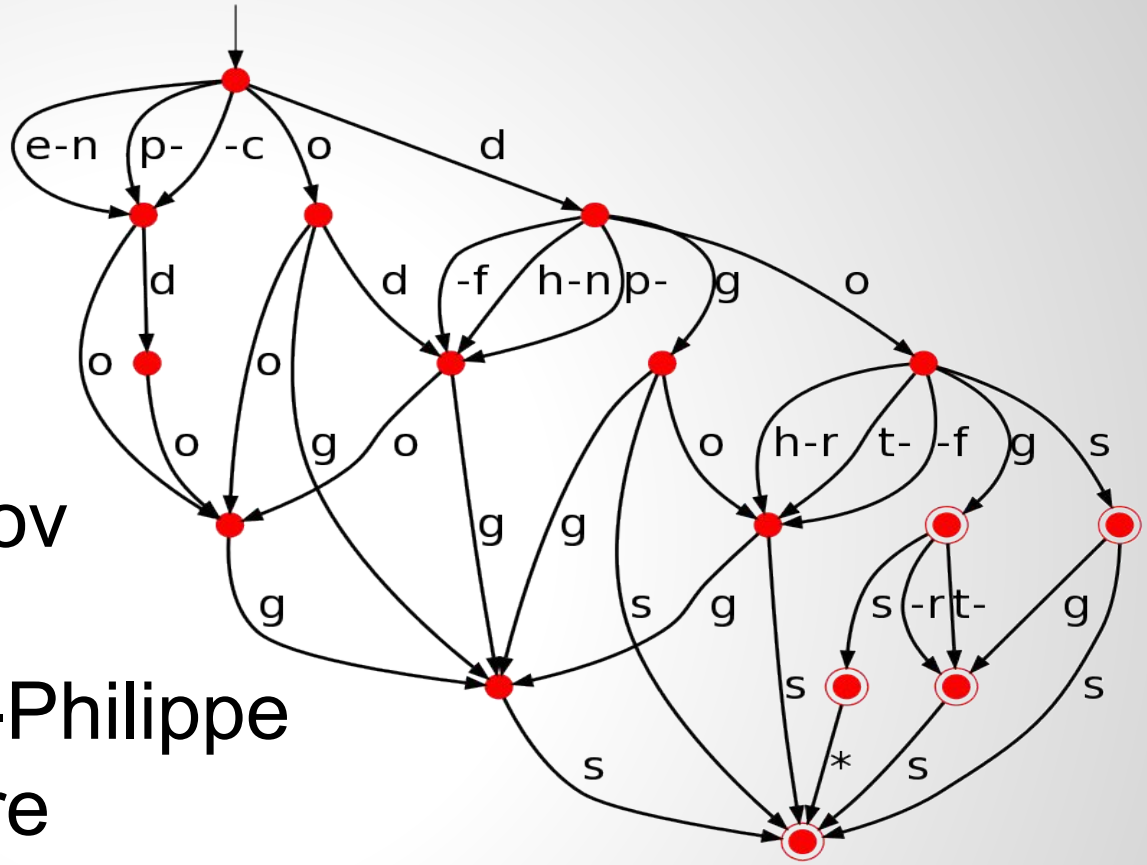
How to implement algorithm from 67-page paper

Hands-On

First…

- Fetch some beer!
- Tell your girlfriend that you will not come to bed!
- Forget about Eclipse & Co! We need a command line and our source code…

FuzzyQuery



- Lev1(dogs)
- 100X faster!
- Schulz and Mihov algo is hairy
- Help from Jean-Philippe Barrette-LaPierre
- Includes transpositions
- UTF32toUTF8 conversion
- http://s.apache.org/FuzzyQuery

DirectSpellChecker

- Use LevT automata to find respellings
- Decent performance
  - ~47 QPS on Wikipedia, single thread
- No side-car index required!

"I'm confident that in three weeks, I'll be done."

JapaneseTokenizer (Kuromoji)

- Donated by Atilika Inc. (アティリカ株式会社)
- Hard problem! (no whitespace, mixed Kanji, Hiragana, Katakana, compounds)
- Dictionaries are stored as FST
  - System dictionary and user dictionaries
  - 11.8X smaller than double array trie
- Viterbi search finds least-cost segmentation
- Token graph for compound words
  - ショッピングセンター (shopping center),
  - ショッピング(shopping), センター (center)

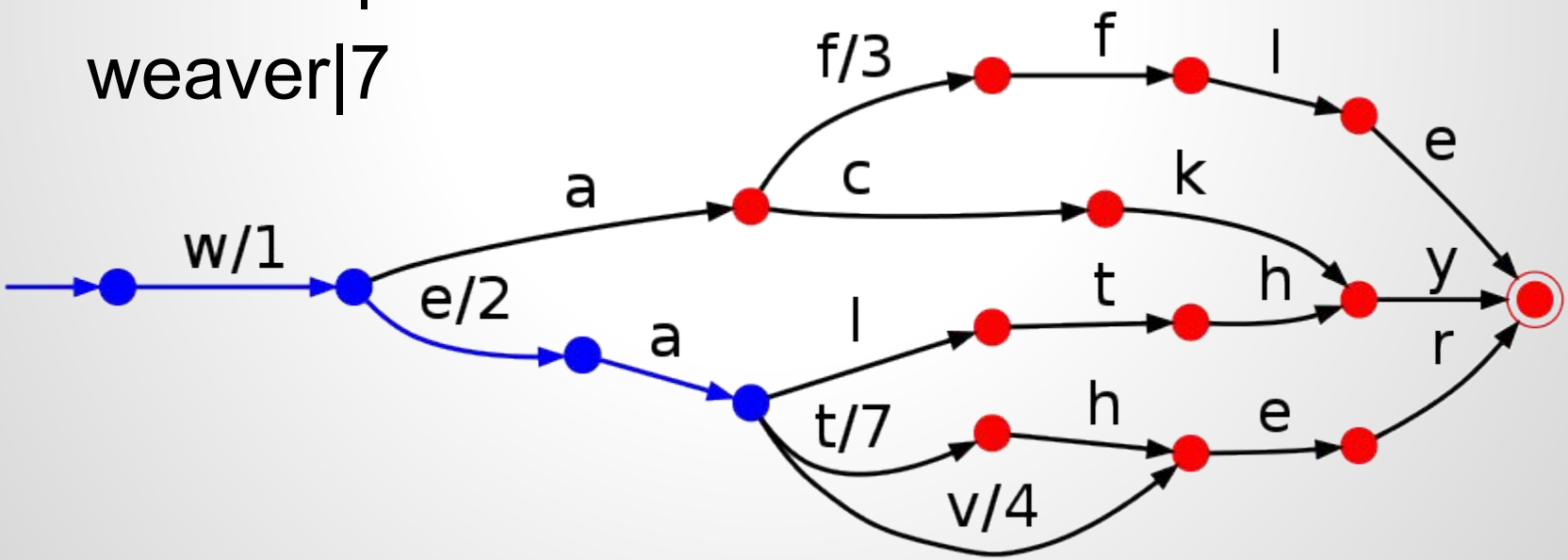
Query Suggesters

- E.g: **`w e a`** suggests weather
- Compile all suggestible queries + weights into FST(s)
- Two FST based suggesters
  - FSTSuggester quantizes weights into buckets
  - WFSTSuggester doesn't
- Find all paths after user's prefix
- Fast: ~240K lookups/sec
- Active work in progress
  - Fuzzy, analyzing, ngram

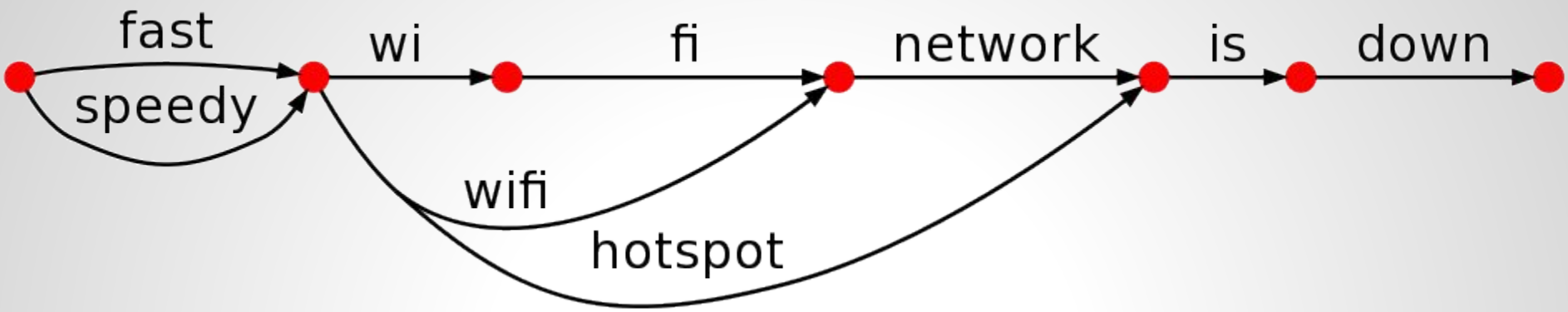
WFSTSuggester example

wacky|1
waffle|4
wealthy|3
weather|10
weaver|7

"I don't think this will work but I can't provide a counterexample right now"

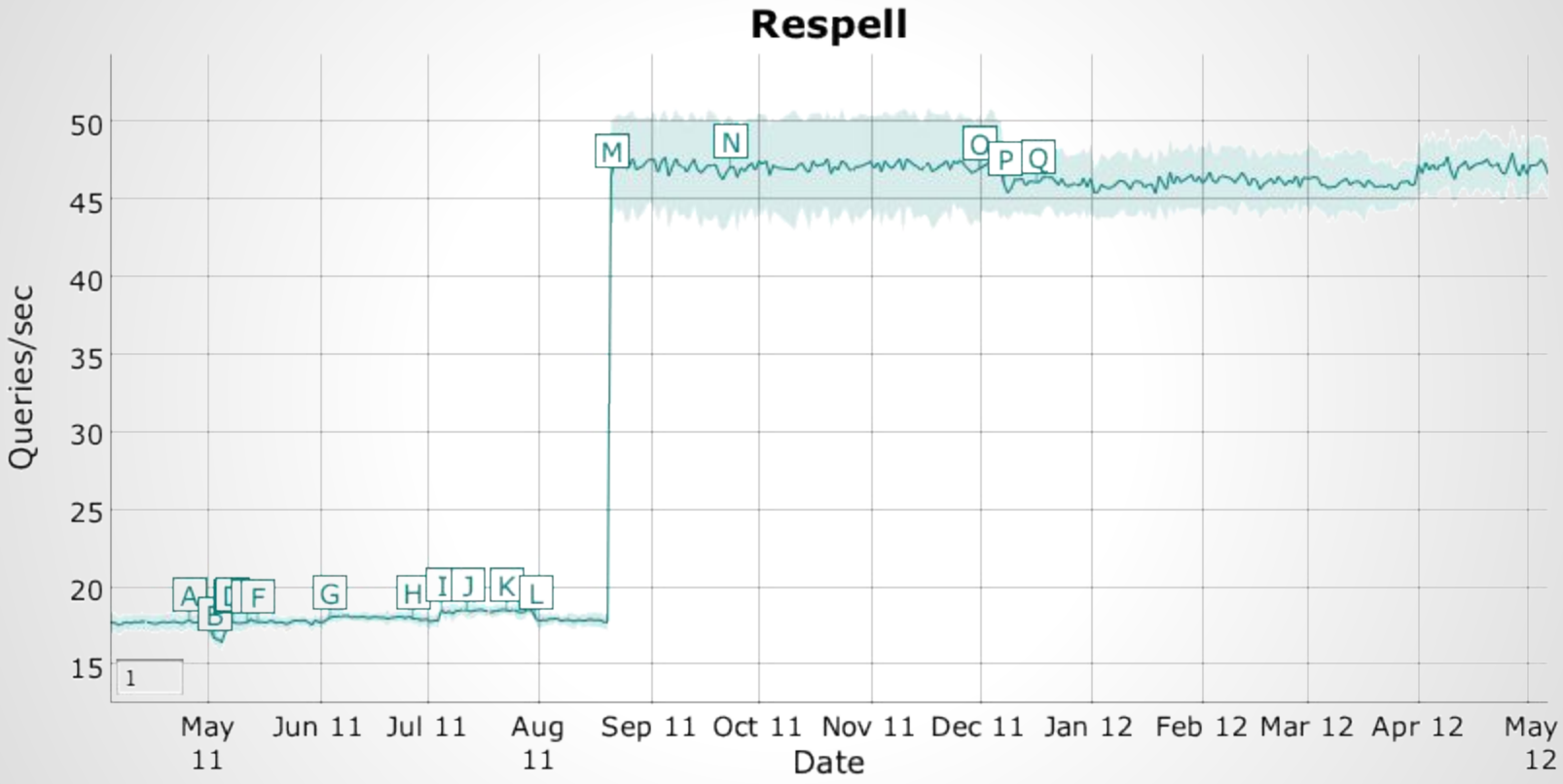


SynonymFilter



- Apply at index time or query time or both
- First version used recursive maps
- New version (in 3.4.0) uses FST
  - 5X faster filter time
  - 14X faster build time
  - 59X less RAM
- Multi-token synonyms mess up graph
- Cannot consume token graph
- http://s.apache.org/TokenGraphs

BlockTree Terms Dictionary



http://s.apache.org/LuceneRespellPerf

BlockTree Terms Dictionary

- Terms dict maps terms to metadata/postings
  - SortedMap<Term,Postings>
  - Pluggable (per codec)
- Term blocks on disk; FST index in memory
  - Variable number of terms per block (vs 3.x)
- Variant of a burst trie (Heinz, Zobel, Williams)
  - Terms assigned to blocks by shared prefix, e.g. http://www.*
- Fast intersect(Automaton)
- Fast "term can't exist"

BlockTree Example

able
above
apple
perfect
preface
prefecture
prefix
previous
profit
programmer
project
zoo

BlockTree Example

perfect

preface

prefecture

prefix

previous

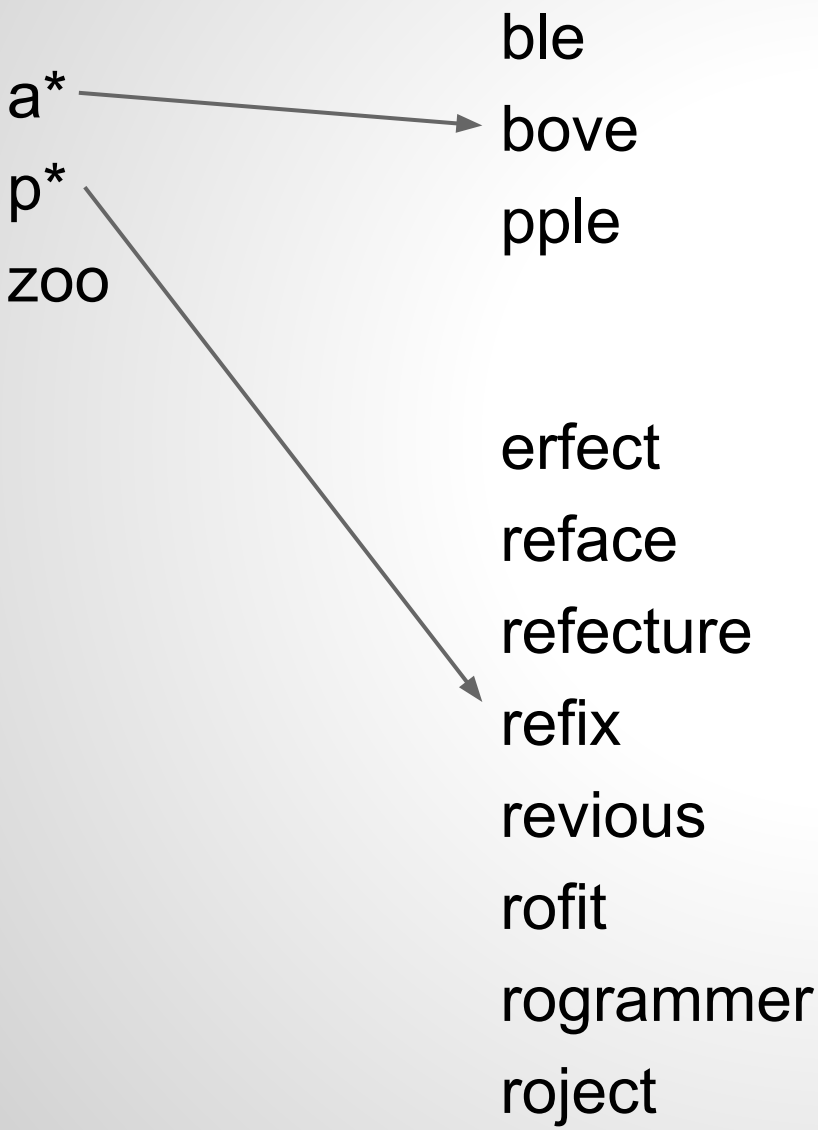
profit

programmer

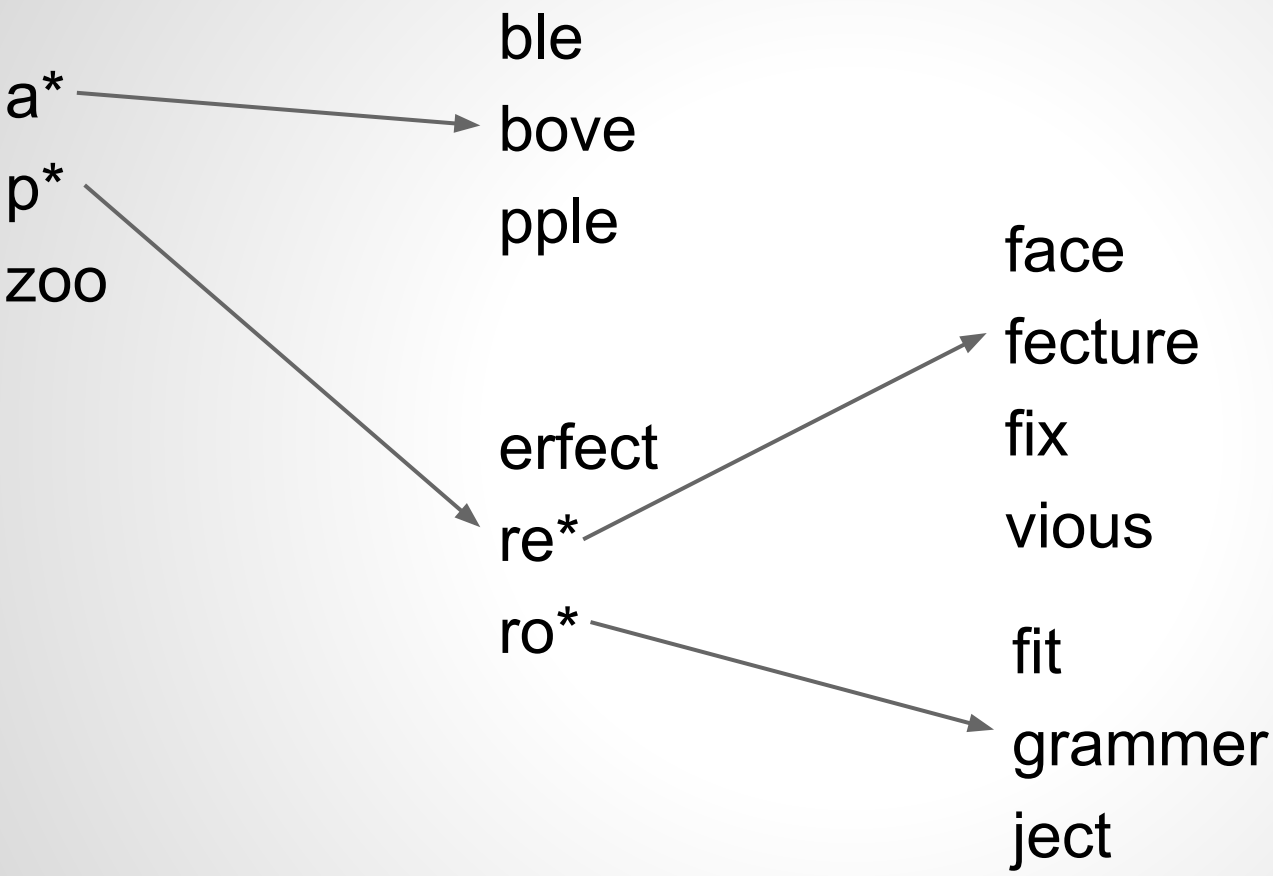
project

zoo

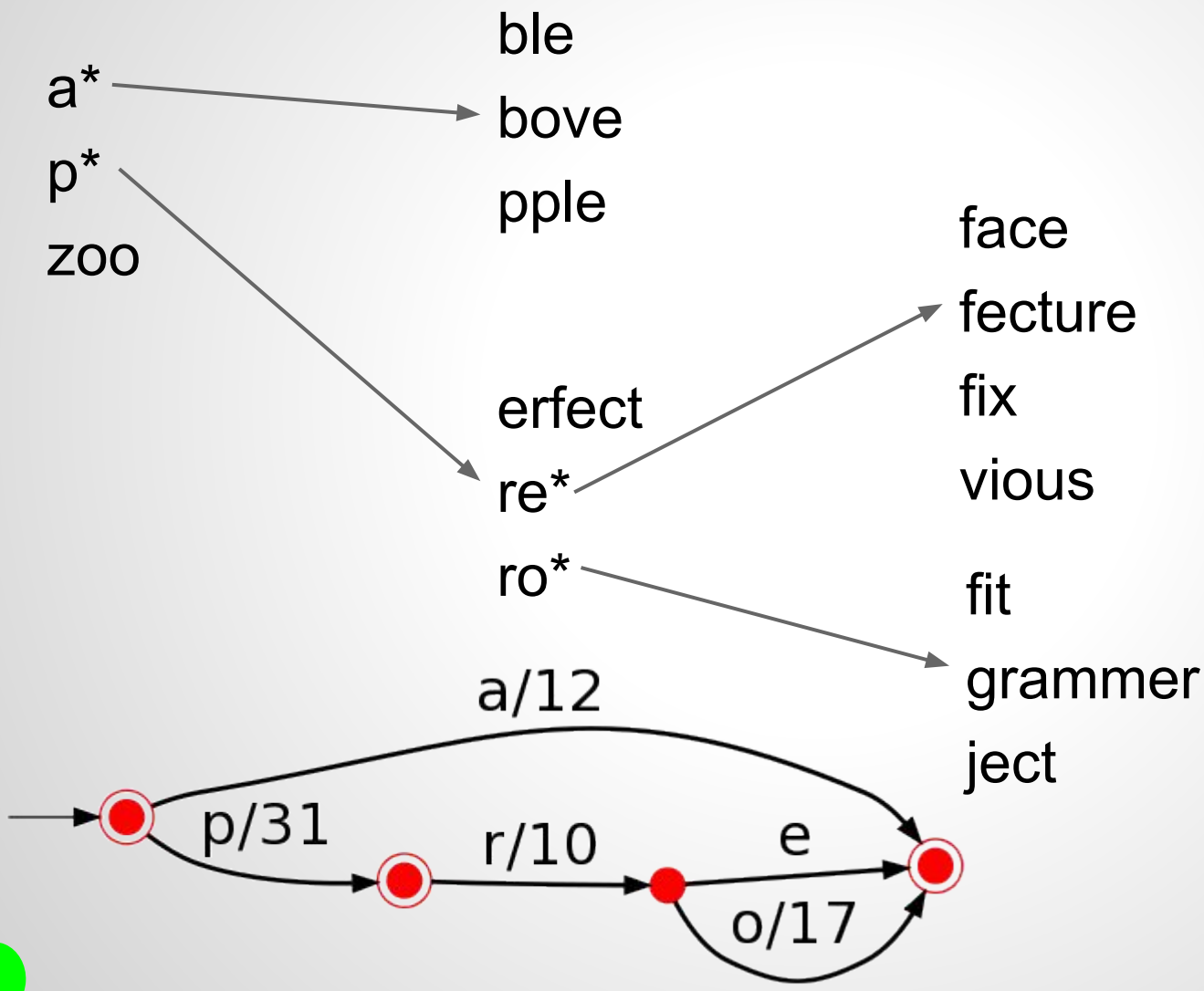
BlockTree Example



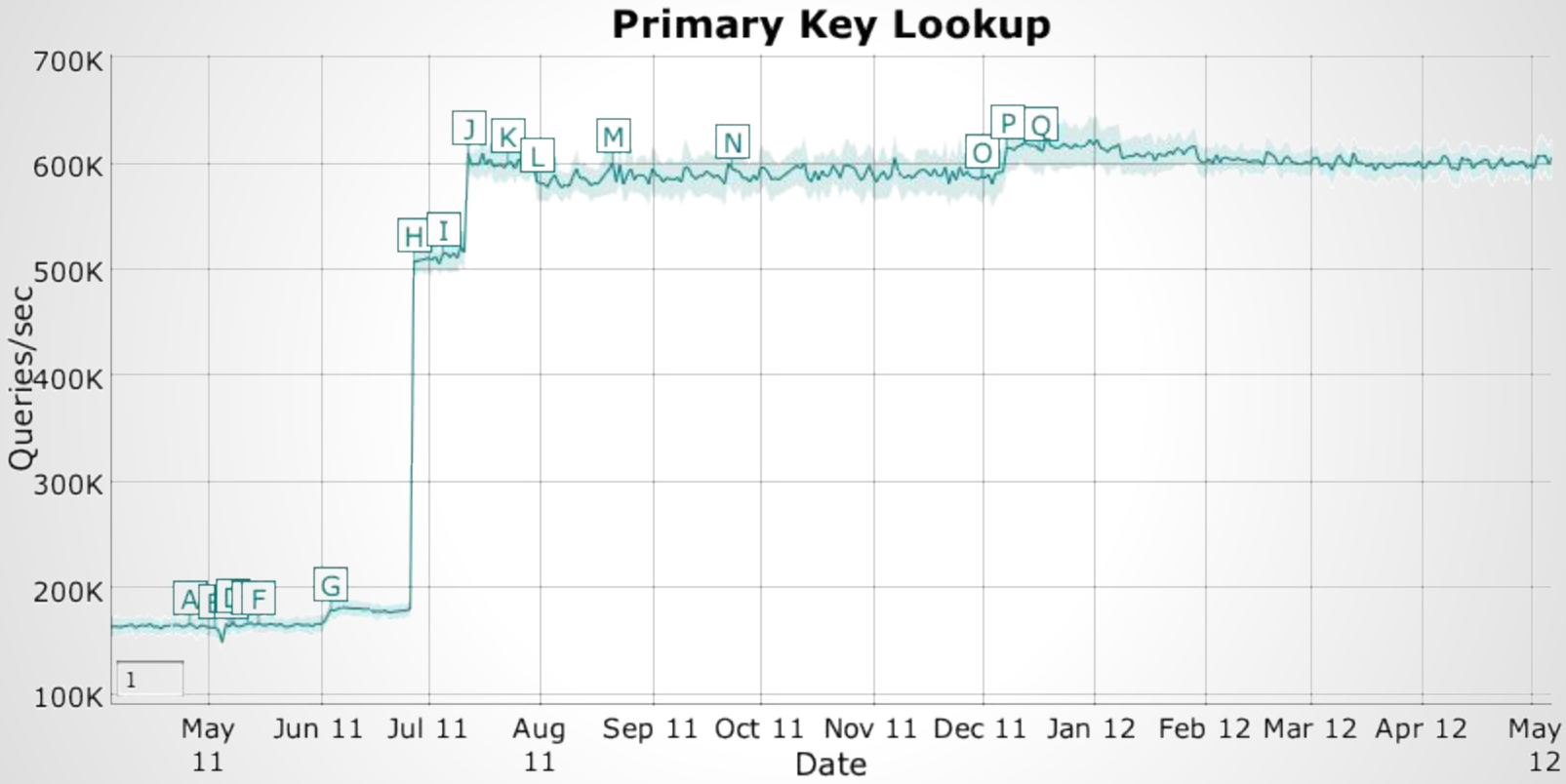
BlockTree Example



BlockTree Example



MemoryPostingsFormat



http://s.apache.org/LucenePKPerf

MemoryPostingsFormat

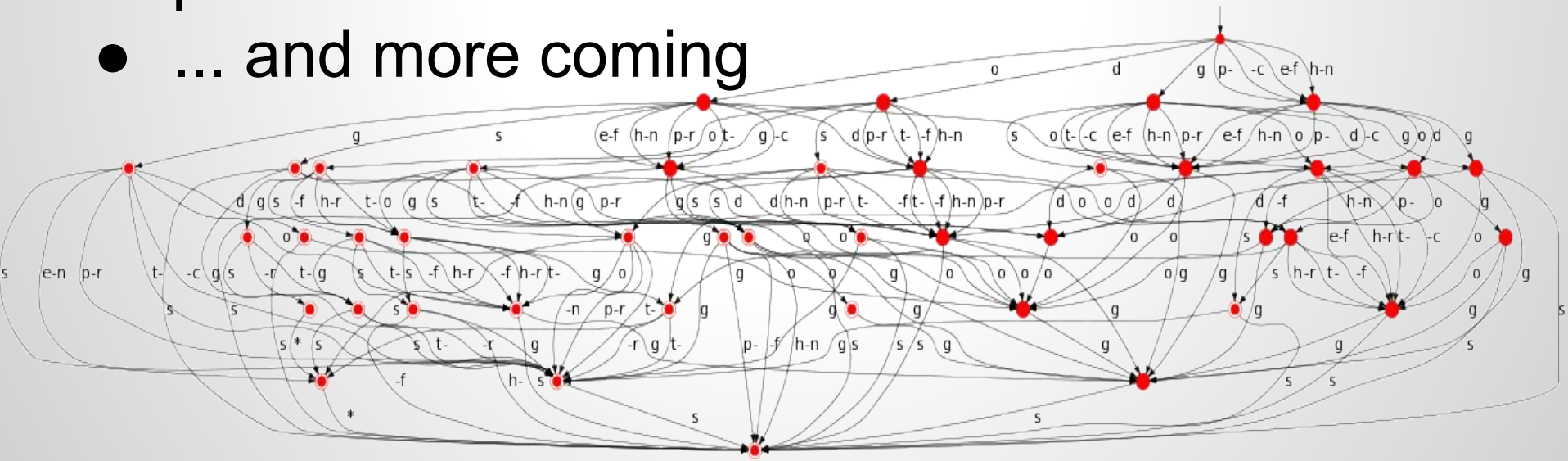
- Postings format is pluggable per-field
- Store all terms + postings in an FST
  - FST is saved to disk
- Great match for primary key fields, date filters
  - 2.8X faster PK lookup
- http://s.apache.org/MemPostingFormat

Future FSA/T Usages

- MappingCharFilter (in progress...)
- FieldCache/DocValues (prototype patch)
- FSTQuery?
- More suggesters
- Top-N most frequent terms for approx distributed IDF
- .... patches welcome!

Conclusions

- FSA/Ts are a good match for Lucene
- Lucene has three wildly different implementations
- FSA/Ts are now used in a number of places...
- ... and more coming



More Information

- Finite State Automata in Lucene
  - Dawid Weiss: Lucene Revolution 2011
  - http://slidesha.re/vKtpVA
- FST API code samples
  - http://s.apache.org/pR