LUCENE REVOLUTION

Presented by lucid IMAGINATION

Spatial Search with Geohashes

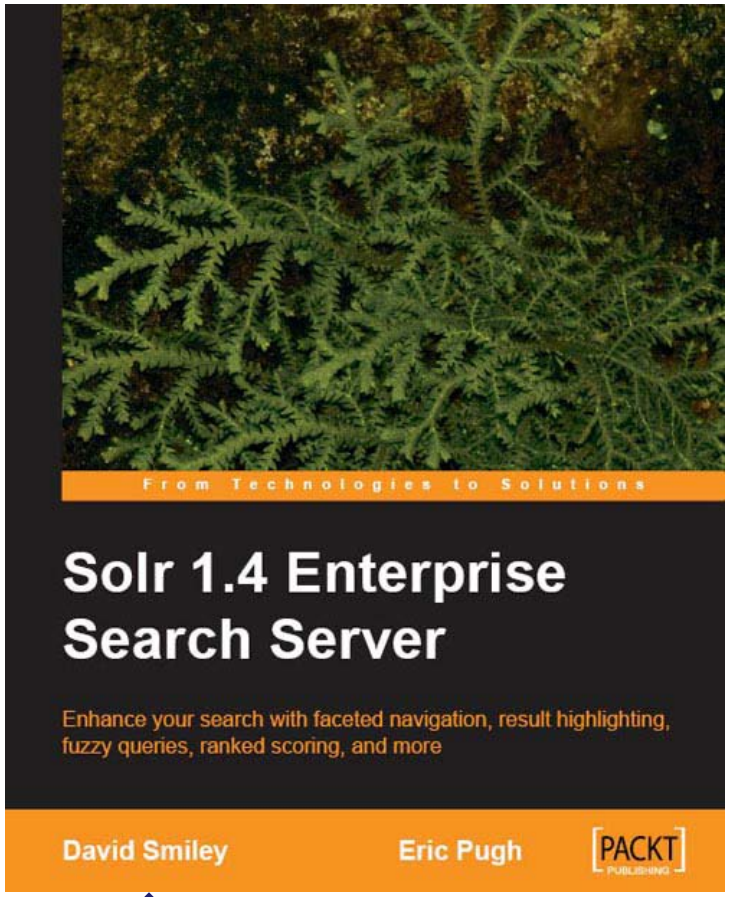
David Smiley, MITRE, dsmiley@mitre.org, October 2010

lucid IMAGINATION

What I Will Cover

- My requirements
- What is a geohash?
- Geohash prefix boundaries
- My Solr geohash prefix indexing strategy
- Analysis, conclusion, and the future

My Background



- **Solr book author**
- **Solr instructor**
  - (at MITRE)
- **Working at MITRE for 10+ years**
  - Supporting internal apps and US DOD sponsors

LUCENE REVOLUTION

lucid IMAGINATION

My Geospatial Requirements

- **Documents have multiple points**
- **Filter search results**
  - No relevancy (i.e. ranking, sort) needed
    - Using proprietary MetaCarta plugin separately for geo-relevancy ranking
- **Lat-Lon bounding box**

- **Environment Considerations:**
  - Large distributed Solr index
  - Low point cardinality
  - Point detail is course – a few miles

lucid
IMAGINATION

Spatial Search in Solr

- ***Nothing officially available yet***
- **SOLR-773, SOLR-1568**

  "Incorporate Local Lucene/Solr"
  - Only point-radius, no bounding lat-lon box
  - Geohash to/from lat-lon (but that's it)
- **JTeam Spatial Solr (fork of SOLR-773)**
- **MetaCarta GeoSearch Toolkit for Solr (alpha)**
  - Steep RAM requirements when only geo filtering
- **Brad Giaccio (attached to SOLR-773)**
  - A good start; could be much faster

Geohashes

- **What is a Geohash?**
  - A lat/lon geocode system
  - Has a hierarchical spatial structure
  - Gradual precision degradation
  - In the public domain

  http://en.wikipedia.org/wiki/Geohash

- **Example: (Boston) DRT2Y**

Geohash Decoding

DRT2Y

- **Translate base-32 dictionary to binary**

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 32	0	1	2	3	4	5	6	7	8	9	b	c	d	e	f	g
Decimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Base 32	h	j	k	m	n	p	q	r	s	t	u	v	w	x	y	z

01100-10111-11001-00010-11110

5 bits needed for each original geohash character

- **Separate even (longitude) & odd (latitude) bits**

0100110101110 (13 bits, longitude)

101111000011 (12 bits, latitude)

Depending on geohash length, the latitude bit count is either one less or equal to that of longitude

lucid IMAGINATION

Geohash Decoding (cont.d)

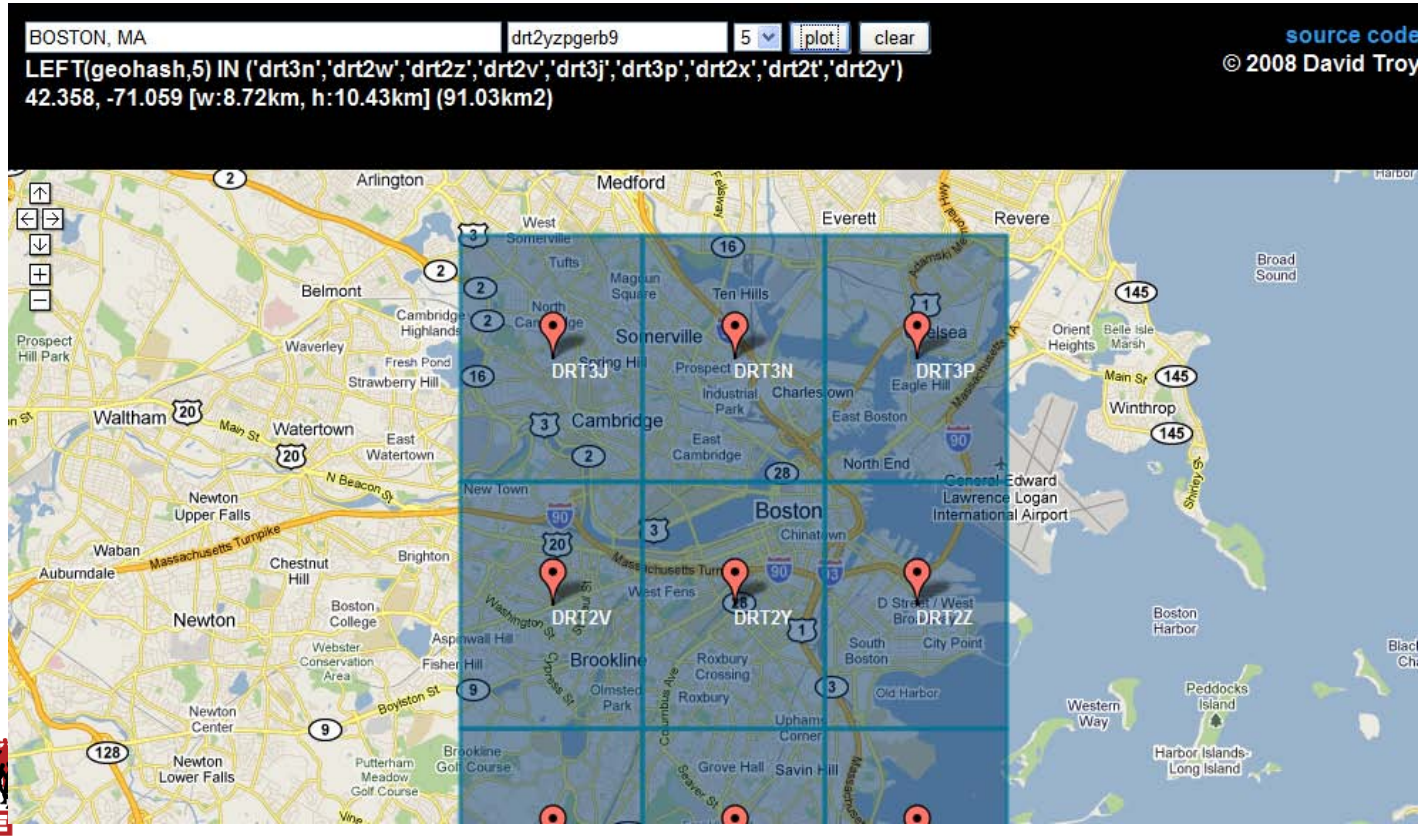
DRT2Y

Latitude example

bit	min	mid	max	val	err
1	-90.000	0.000	90.000	45.000	45.000
0	0.000	45.000	90.000	22.500	22.500
1	0.000	22.500	45.000	33.750	11.250
1	22.500	33.750	45.000	39.375	5.625
1	33.750	39.375	45.000	42.188	2.813
1	39.375	42.188	45.000	43.594	1.406
0	42.188	43.594	45.000	42.891	0.703
0	42.188	42.891	43.594	42.539	0.352
1	42.188	42.539	42.891	42.363	0.176
0	42.188	42.363	42.539	42.275	0.088
0	42.188	42.275	42.363	42.319	0.044
1	42.275	42.319	42.363	42.341	0.022

Demo

http://openlocation.org/geohash/geohash-js/



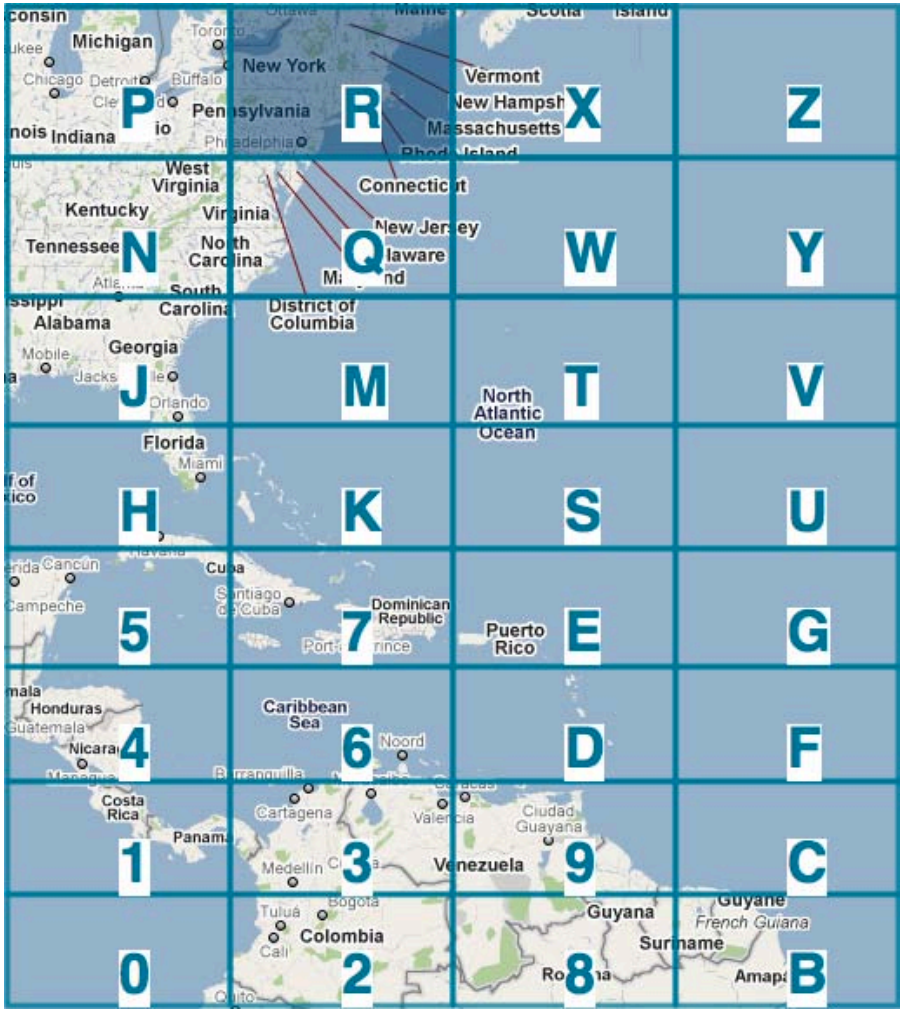
LUCENE REVOLUTION

lucid IMAGINATION

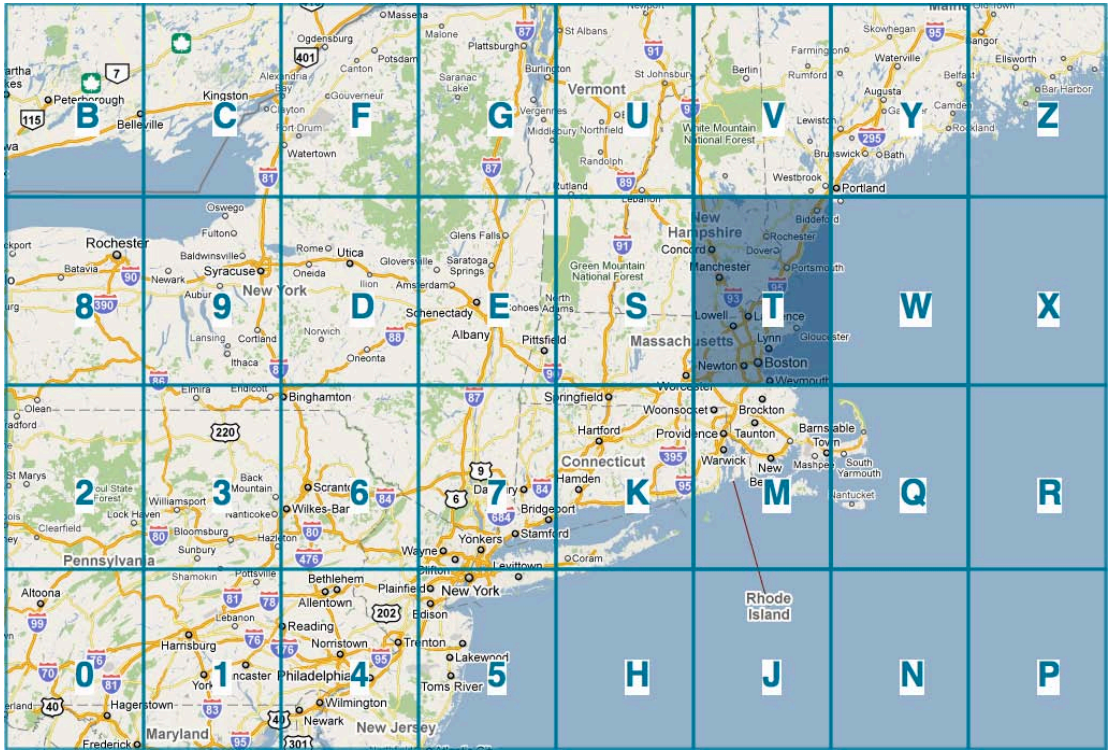
Zooming In: D


B
C
F
G
U
V
Y
Asia
Europe
8
9
D
E
S
T
W
South America
Indian
2
3
6
7
K
M
Q
0
1
4
5
H
J
N
Antarctica

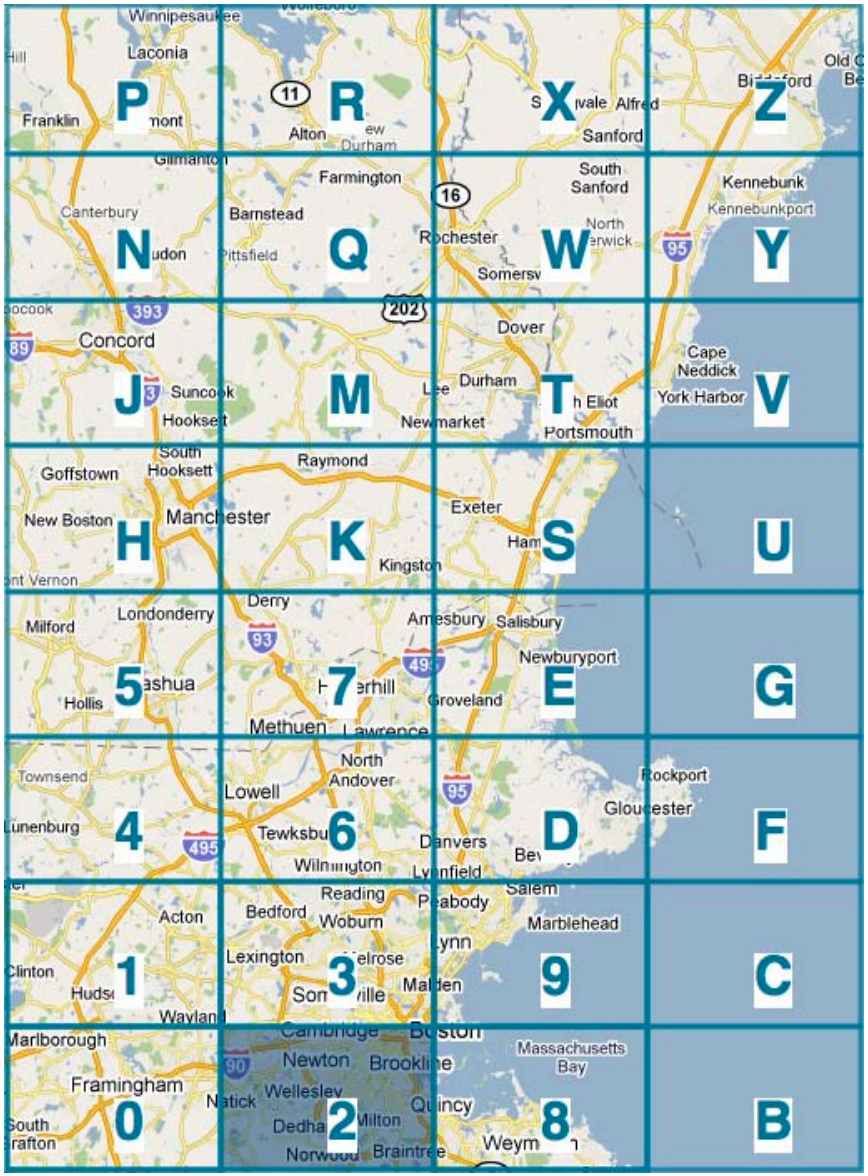
Zooming In: DR



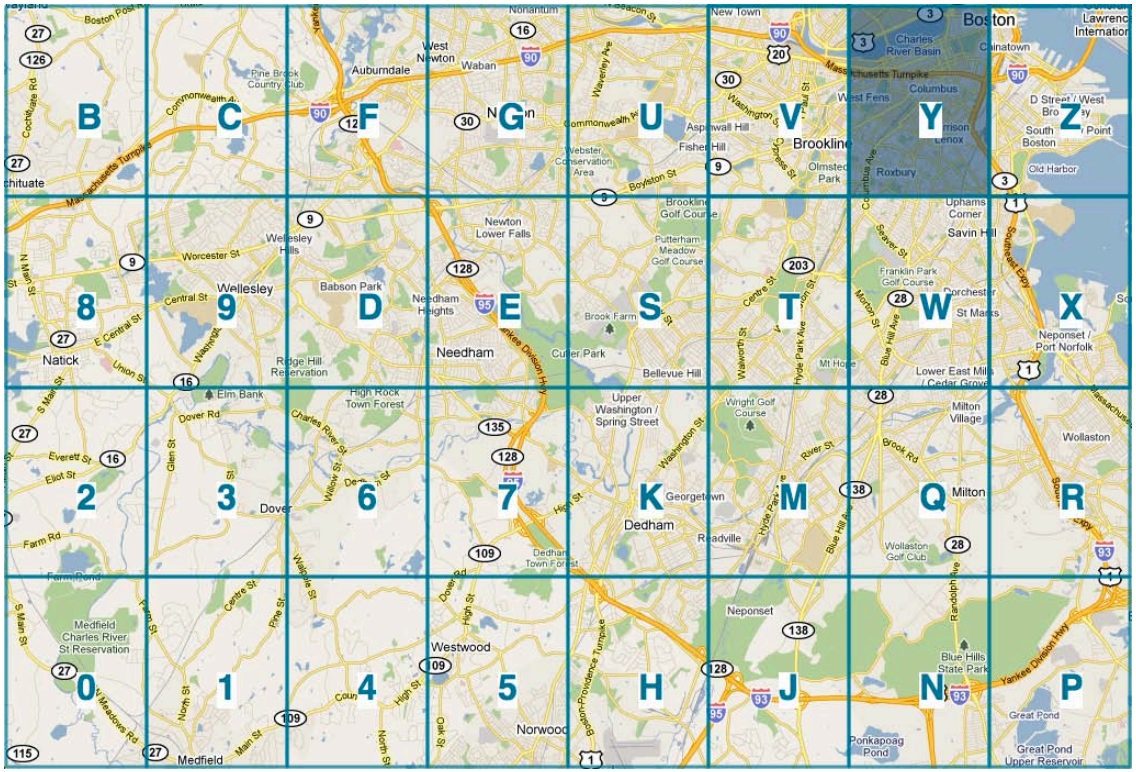
Zooming In: DRT



Zooming In: DRT2



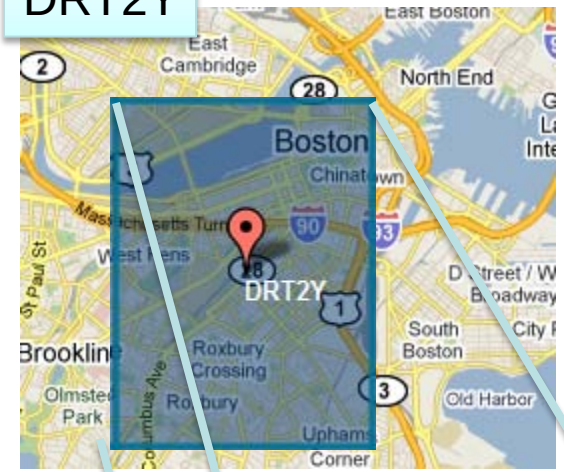
Zooming In: DRT2Y



Geohash Grids

Internal coordinates of an **odd** length geohash…

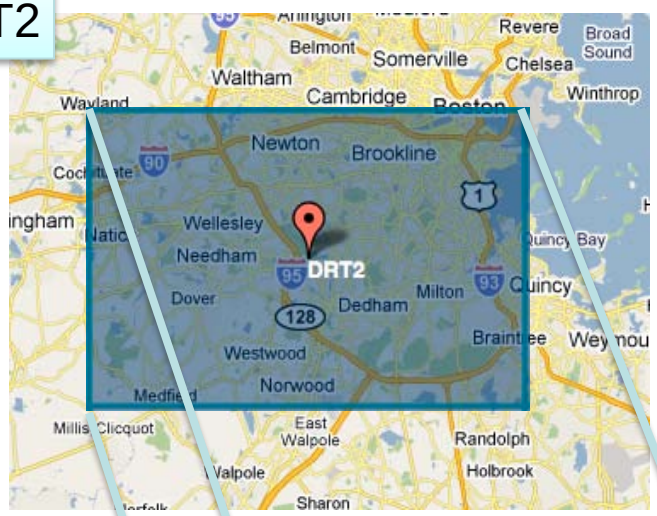
DRT2Y



P	R	X	Z
N	Q	W	Y
J	M	T	V
H	K	S	U
5	7	E	G
4	6	D	F
1	3	9	C
0	2	8	B

…and an **even** length geohash

DRT2



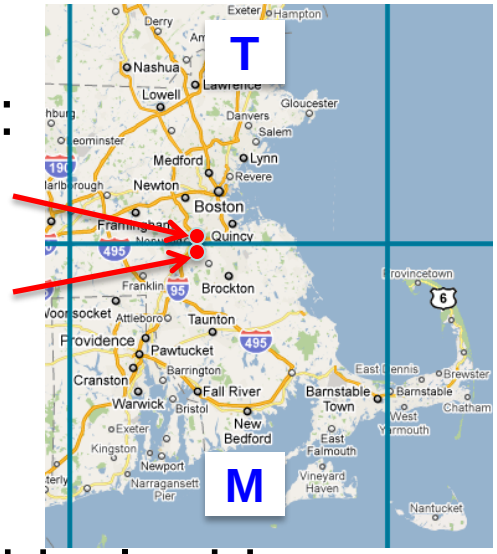
B	C	F	G	U	V	Y	Z
8	9	D	E	S	T	W	X
2	3	6	7	K	M	Q	R
0	1	4	5	H	J	N	P

LUCENE REVOLUTION

lucid IMAGINATION

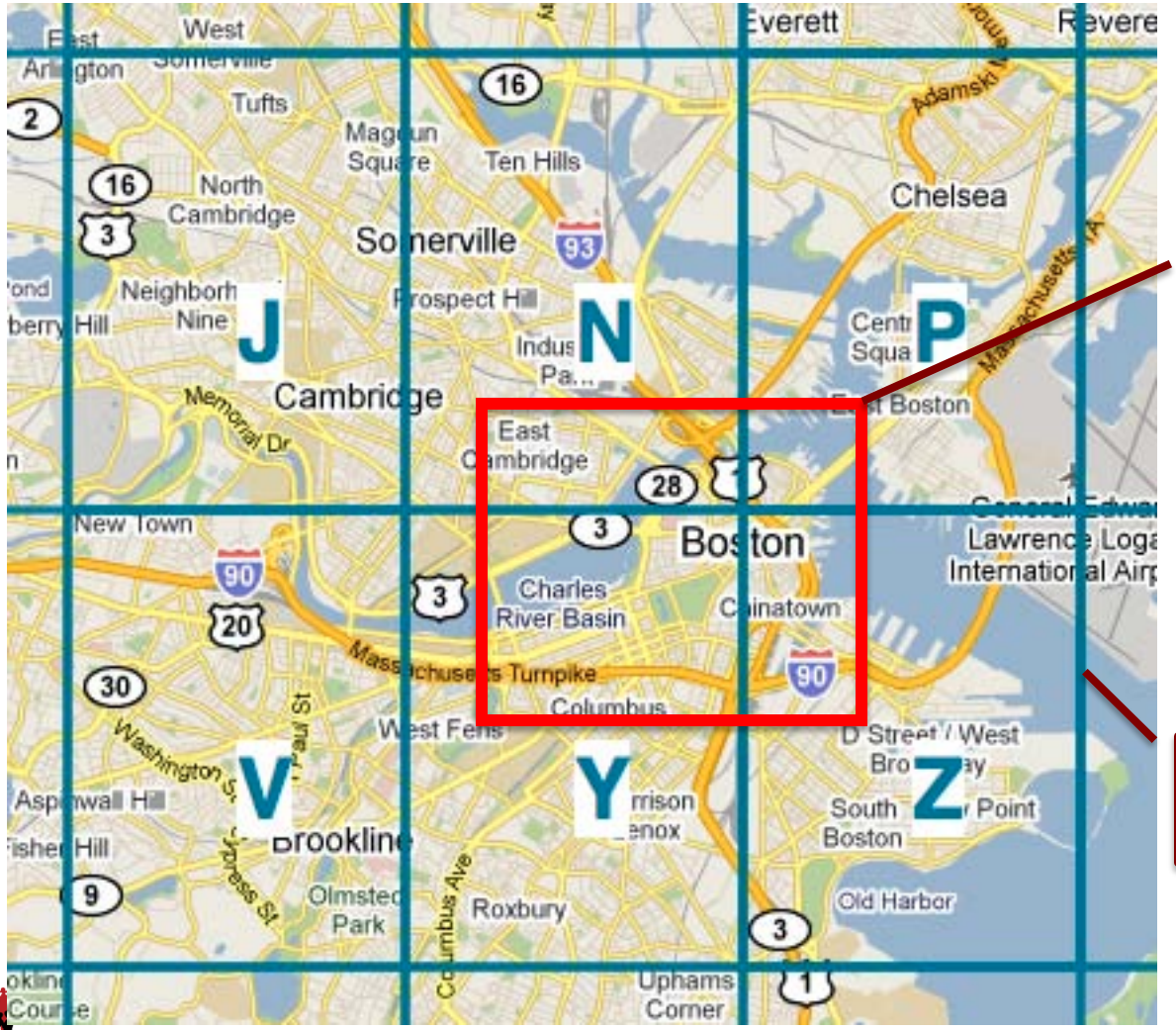
Proximity with Geohashes

- **Points with a long common prefix are near each other**
  - But the converse is not always true!
    - i.e.: ~~Points near each other share a long common prefix~~
    - Edges cases exist at every level:
      - D R **T**....
      - D R **M**...
      - (only share 2 letters)



  - However geohashes form a hierarchical grid

Filtering by Lat-Lon Box



User Query

Geohash Resolution: 5

Indexing Strategy Overview

- **Index geohashes**
- **Develop query support to specify lat-lon box**
- **Choose a good geohash resolution for the box**
- **Lookup set of overlapping geohashes**
- **Develop a Lucene Filter:**
  - Seek to each overlapping geohash in termsEnum
  - Skip term (geohash) if not in desired box
  - Accumulate matching documents

Indexing: Index Geohashes

- **Provided by SOLR-1586**
- **solr.GeoField**
  - Input/output is latitude-comma-longitude, indexed internally as a geohash
- **See also: GeoHashUtils.java**
  - Only bidirectional lat-lon conversion routines
  - Needed code to navigate to adjacent geohashes. (35 L.O.C.)

Indexing: Query Support

- **Modified SOLR-1586:**

  {!sfilt fl=body
  ll=42.358,-71.059 ur=43.000,-70.000}

  (~15 L.O.C.)

Indexing: Determine Resolution

double[] lookupDegreesSizeForHashLen(int hashLen)

(14 L.O.C.)

int estimateGoodPrefixLen(lat or lon, degrees)

(9 L.O.C.)

lucid
IMAGINATION

Indexing: Lookup Overlapping Geohashes

String[] calcPrefixHashesToFilter(double north, south, east, west)

(45 L.O.C.)

Indexing: Develop Lucene Filter

- **Implement Lucene Filter getDocIdSet():**
  - Call calcGeohashPrefixesToFilter()
  - Seek to each overlapping geohash in termsEnum
  - Skip term (geohash) if not in desired box
  - Accumulate matching documents

  (45 L.O.C.)
- **Alter solr.GeoHashField createSpatialQuery() to use the filter**

  (5 L.O.C.)

Future Improvements

- **Lookup table based geohash adjacency**
- **Avoid isInBox() calculation for geohashes known to be completely within the query**
- **Index multi-precision geohashes**
  - DRT2Y: DRT2Y, DRT2, DRT, DR, D
  - Reduces / avoids term enumerating, isInBox()
- **Use different geohash-like encoding**
  - See javageomodel project

References

Pre-Existing Solr Spatial Search

- **SOLR-773, SOLR-1568**
- **JTeam Spatial Solr Plugin**

  http://www.jteam.nl/news/spatialsolr.html
- **Brad Giaccio, of ManTech**

  https://issues.apache.org/jira/browse/SOLR-773 (solrGeoQuery.tar)
- **MetaCarta GeoSearch Toolkit for Solr**

  http://berlinbuzzwords.wdfiles.com/local--files/links-to-slides/goodwin_bbuzz2010.pdf

References

Geohash

- **Geohash**
  http://en.wikipedia.org/wiki/Geohash
- **David Troy's Demo**
  http://openlocation.org/geohash/geohash-js/
- **javageomodel (similar to geohash)**
  http://code.google.com/p/javageomodel/
  – Geospatial queries in Google App Engine