SpatialForTimeDurations

If you need to index multi-value time durations, then Solr 5.0 has a new field type that supports it: DateRangeField. For information on how to use DateRangeField, see the Solr Reference Guide: https://cwiki.apache.org/confluence/display/solr/Working+with+Dates

If you have multi-value number ranges that are not times, you're probably best off still using DateRangeField, but encoding your data into a date since that's all that DateRangeField accepts. DateRangeField is based on spatial technology, but it's optimized for dates. If you do abuse DateRangeField for non-date data, then, if you can, try to keep the data as a number of seconds instead of milliseconds (use 000 milliseconds). This limits the numeric space you have to work with, but it will likely perform much better if used this way. It will also perform better if the dates are after the "Gregorian change date" – October 15th, 1582. At some point it's likely a NumberRangeField might be developed but that has yet to occur.

If you are not yet using Solr 5, then one of Solr 4's spatial field types can be used for non-spatial means like this... or instead, think of this problem as being turned into a spatial problem. Usually, "spatial" is nearly synonymous with "geospatial" but it can be used for other purposes like this too. Read on for more...

First, read Chris Hostetter (aka Hossman)'s illustrated slides from a Solr meetup:
Spatial Search Tricks for People Who Don't Have Spatial Data.

Configuration

However, don't use the field configuration as given in that presentation, not to mention there are some tweaks to be done to the queries to avert edge cases.

Here is an example Solr fieldType configuration that may only require some small changes for your data:

```
<fieldType name="days_of_year"
           class="solr.SpatialRecursivePrefixTreeFieldType"
           geo="false"
           worldBounds="0 0 366 366"
           distErrPct="0"
           maxDistErr="1"
           units="degrees"
        />
```

Some explanation:

- worldBounds: The left two numbers are the starting epoch values (usually 0) and the end two are the maximum values. "minX minY maxX maxY" order. The min and maximums should be set to the same in the X & Y dimensions. I'm tempted to suggest using a power of two large enough for your max value, particularly if you experience any small precision problems. But I'm unaware of problems so you don't have to do this. The syntax here will eventually be unsupported in Solr 5, where you'll have to use the ENVELOPE WKT style syntax (supported in Solr 4.7 and beyond).
- distErrPct: This use of spatial generally implies precise results, not approximate results. Setting distErrPct to 0 means no approximation.
- maxDistErr: This is basically the smallest indivisible value. This use of spatial should generally use the integer range to keep things simpler, so use "1".

Indexing

Use "x y" (x space y) order for the points:

```
<doc>
  ...
  <field name="shift">1 3</field>
  ...
```

Search

Now for queries, look at the examples on Hossman's slides. **However** to avoid edge cases, you should slightly buffer the query shapes – the edges other than the minimum or maximum. In addition, the syntax used is deprecated; use the rectangle range query style instead. One example query given was `Intersects(0 9 8 365)`. In rectangle range query format, this is `["0 9" TO "8 365"]`. **But we need to buffer it:** `["0 8.5" TO "8.5 365"]` Math: 9 - 0.5, 8 + 0.5

Limitations

It's not realistic to configure the max values in worldBounds to be a huge number (say Long.MAX_VALUE 2^{64}. Maybe as high as perhaps 2^{50})?

Credit

This durration modeling as coordinates idea originated in a solr-user@lucene thread. See David's initial response to Geert-Jan's question and the subsequent followup about using different rectangle intersections.