SpanQuery

I've been experimenting with using SpanQuery to perform what is essentially a limited type of database 'join'. Each document in the index contains 1 or more 'rows' of meta data from another 'table'. The meta data are simple tokens representing a column name/value pair (e.g. color$red or location$123). Each row is represented by a span with a maximum token length equal to the maximum number of meta data columns. If a column has multiple values, they are all indexed at the same position (e.g. color$red, color$blue). All rows are added to a single field. The spans are 'separated' from each other by introducing a position gap between them via 'Analyzer.getPositionIncrementGap'. This gap should be greater than the number of columns in each span.

At query time, a SpanNearQuery is constructed to represent the meta data to join. The 'slop' value is set to the maximum number of meta data columns (minus 1). Using a simple Antlr parser, boolean span queries with AND, OR, NOT can be constructed fairly easily. The SpanQuery is And'd to the main query to build the final query.

This approach is flexible and pretty efficient because no stored fields or external data are accessed at query time. Span queries are more expensive than other queries, though. We measure performance via throughput (as opposed to the response time for a single query), and the addition of a SpanQuery reduced throughput by 5X for ordered spans and 10X for unordered spans. Still, this may be acceptable for some applications, especially if spans are not used on every query.