SummerOfCode2011

Lucene Google Summer of Code 2011

Google Summer of Code 2011 has started! Now it's time to get some exciting projects underway for this year's GSoC:

Project List

LUCENE-1768: NumericRange support for new Query Parser

Apache Lucene supports indexing and searching for numeric types. This allows Lucene to support faster range queries, since building the field cache is much faster than using text-only numbers. One of the big limits today is the lack of support for numeric range queries in Lucene contrib query parser, which still only supports text range queries. This project proposes to implement numeric support in contrib query parser.

LUCENE-2308: Separately specify a fields type

Goal of this project is to refactor the Field Lucene API by introducing new FieldType class to separate Fields values from their properties and open way for easier Field extensions. This will result in more understandable instantiation of similar fields across documents. Field class, as part of core API, is very sensitive to shallow design or implementation which can cause drastic performance degradation due to its massive usage all over Lucene and Solr project, making this a challenging task.

LUCENE-2959: Implementing State of the Art Ranking for Lucene

Lucene employs the Vector Space Model (VSM) to rank documents, which compares unfavorably to state of the art algorithms, such as BM25. Moreover, the architecture is tailored specifically to VSM, which makes the addition of new ranking functions a non-trivial task. This project aims to bring state of the art ranking methods to Lucene and to implement a query architecture with pluggable ranking functions.

LUCENE-2793, LUCENE-2795 : Enable Lucene to take advantage of low-level IO options (direct IO) and generalize it’s Directory implementation

Aims to generalize the current Lucene Directory implementation by making it a UnixDirectory. This would be done by adding IOContext to the lower level API. These are two existing Lucene tasks (LUCENE-2793 and LUCENE-2795).

LUCENE-2979: Simplify configuration API of contrib Query Parser

Lucene contrib query parser has a configuration API that was inherited from token stream API, which uses AttributeSource and Attributes to share token information across token filters. However, the use of this Attribute API in contrib query parser makes configuration much more complex than it needs to be. This project proposes to simplify this API to something much simpler, using a map data structure instead of the complex Attribute API.

Helpful Hints

Open source development here at the Apache Software Foundation happens almost exclusively in the public and I encourage you to follow this. Don't mail folks privately; please use the mailing list to get the best possible visibility and attract interested community members and push your idea forward. As always, it's the idea that counts not the person!

That said, please do not underestimate the complexity of even small "GSoC - Projects". Don't try to rewrite Lucene or Solr! A project usually gains more from a smaller, well discussed and carefully crafted & tested feature than from a half baked monster change that's too large to work with.

Once your proposal has been accepted and you begin work, you should give the community the opportunity to iterate with you. We prefer "progress over perfection" so don't hesitate to describe your overall vision, but when the rubber meets the road let's take it in small steps. A code patch of 20 KB is likely to be reviewed very quickly so get fast feedback, while a patch even 60kb in size can take very long. So try to break up your vision and the community will work with you to get things done!