

An ExtendJ frontend for PQL/Java

Christoph Reichenbach, christoph.reichenbach@cs.lth.se

December 20, 2017

Wanted: One or two compiler frontend hackers who have taken the Compilers course and want to write a Java language extension in ExtendJ that auto-parallelises certain kinds of computations. This is an M.Sc. thesis project supervised by Christoph Reichenbach in the SDE group.

PQL/Java is a language extension to Java that makes it easy to write SQL-like queries over Java datastructures and execute them in parallel. For instance, you can write

```
query(Set.contains(x)): s1.contains(x)
    && s2.contains(x)
    && s3.contains(x);
```

to compute the intersection of three sets, or

```
reduce(sum) z: s.contains(x) && x > 0;
```

to sum up all elements of the set `s` that are greater than zero. PQL/Java can automatically parallelises its queries, too, and it outperforms Java streams in our benchmarks. However, PQL/Java's current frontend is a modified version of the Java 6 compiler, so it cannot take advantage of modern conveniences like Java 8's lambdas or integrate nicely with streams.

In this project, you will implement a new frontend for the PQL/Java system in ExtendJ, as a general language extension to Java. We have a prototype from an earlier compiler project that you can build on, or you can start from scratch. Optionally, you can also influence the design of the PQL/Java language or add new optimisations to it.

This thesis project entails:

- Learning about PQL/Java.
- Implementing syntax and static analyses for PQL/Java in ExtendJ.
- Generating PQL intermediate code and invoking the PQL optimiser and runtime.
- Testing and benchmarking.
- *Optionally:* evolutions to the PQL/Java language.
- *Optionally:* new PQL/Java optimisations.
- Writing the thesis.

To learn more about the language and see the current implementation, have a look at the project home page: <http://creichen.net/pql/index.html>.

If you are interested, feel free to contact me at christoph.reichenbach@cs.lth.se.