Streams à la carte: Extensible Pipelines with Object Algebras (Artifact)

Aggelos Biboudis[1], Nick Palladinos[2], George Fourtounis[1], and Yannis Smaragdakis[1]

1 **Department of Informatics and Telecommunications**
University of Athens, Greece
{biboudis,gfour,smaragd}@di.uoa.gr

2 **Nessos Information Technologies S.A.**
Athens, Greece
npal@nessos.gr

Abstract

In Streams à la carte we address extensibility shortcomings in libraries for lazy-streaming queries with a new design. The architecture underlying this design borrows heavily from Oliveira and Cook's object algebra solution to the expression problem, extended with a design that exposes the push/pull character of the iteration, and an encoding of higher-kinded polymorphism.

In this library we apply our design to Java and show that the addition of full extensibility is accompanied by high performance, matching or exceeding that of the original, highly-optimized Java streams library.

In this artifact we present a fundamental set of sequential operators `map`, `filter`, `reduce`, `count`, `take/limit` and `iterate`.

Additionally we present the behaviors that are discussed in the paper: `push`, `pull`, `fused pull`, `logging`, `id` (for blocking terminal operators), `future` (for non-blocking terminal operators).

1998 ACM Subject Classification D.2.2. Software libraries, D.1.5 Object-oriented Programming, D.3.3 Language Constructs and Features

Keywords and phrases object algebras, streams, extensibility, domain-specific languages, expression problem, library design

Digital Object Identifier 10.4230/DARTS.1.1.9

Related Article Aggelos Biboudis, Nick Palladinos, George Fourtounis, and Yannis Smaragdakis, "Streams à la carte: Extensible Pipelines with Object Algebras", in Proceedings of the 29th European Conference on Object-Oriented Programming (ECOOP 2015), LIPIcs, Vol. 37, pp. 591–613, 2015.
http://dx.doi.org/10.4230/LIPIcs.ECOOP.2015.591

Related Conference 29th European Conference on Object-Oriented Programming (ECOOP 2015), July 5–10, 2015, Prague, Czech Republic

1 Scope

The artifact is designed to support repeatability of all the experiments of the companion paper, allowing users to test the library on a variety of benchmarks.

It includes the implementation of all operators and semantics discussed in the paper and the corresponding unit tests. We also include three basic categories of benchmarks: basic pipelines with various combinations of both `Pull` and `Push` algebras, fused pipelines to exercise `map` and `filter` fusion, `iterator`-based pipelines to demonstrate differences between the `Pull` algebra and by obtaining of an `iterator` in Java 8 Streams and `take` pipelines (the take operator is the same as the `limit` operator in Java 8 Streams).

2 Content

The artifact package includes:

- *object algebras* and factories.
- benchmarks.
- unit tests.
- any higher kinded types used.
- the GADT encoding.
- the Fluent APIs of Scala and C#.

To simplify repeatabiliy of our experiments, we provide a `maven` script. The command `mvn test` runs all the unit tests and the script `sh run_benchmarks.sh` runs all the benchmarks. The `run_benchmarks.sh` script simply builds the JMH benchmarks über-jar and then uses the command line interface of JMH to pass the arguments of the experiments. The user can run the benchmark script as is or by passing a regular-expression for filter to select only some of them. The micro-benchmark suite can be passed the number of elements `N` for `map`, `count`, operations with large number of elements, `N_small` that is used for `cart` and `limit/take` examples. `N_limit` is used as the parameter for `limit` and `F` is used for benchmarks with fused pipelines.

For more information on JMH the user can run it directly, e.g., to get the help dialog `java -jar target/microbenchmarks.jar -h`.

We omitted baseline tests from the paper (although we included them in the repo) as the focus of the paper is not on comparing hand-optimized tight loops with streaming pipelines. We have investigated this in previous work (`http://arxiv.org/abs/1406.6631`) and it is something that we would like to investigate in the OpenJDK specifically in the immediate future.

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). The latest version of our code is available on github: `http://biboudis.github.io/streamalg/`.

4 Tested platforms

The artifact is known to work on Fedora Linux x64 operating system (version 3.17.4-200.fc20) that run natively on an Intel Core i5-3360M vPro 2.8GHz CPU (2 physical x 2 logical cores). The total memory of the system was 4GB. We used version 1.8.0.25-4.b18 of the Open JDK.

5 License

MIT (`http://www.apache.org/licenses/LICENSE-2.0.html`)

6 MD5 sum of the artifact

7e5646ff37150e9f7ea7bf109b78478a

7 Size of the artifact

100.5 KB