StreamIt 2.1.1 Release

January 29, 2007

This is for the January 29, 2007 release of the StreamIt compiler and runtime system, version 2.1.1.
`$Id: readme.tex.in,v 1.26 2006/09/11 20:04:10 thies Exp $`

1 What is this?

StreamIt is a programming language designed for efficient implementation of streaming systems, particularly digital signal processing applications. This is a public release of the StreamIt compiler. While the compiler is still under active development, we believe this release should be useful for developing StreamIt applications. See "caveats", below, and report bugs and feature requests to `streamit@cag.csail.mit.edu`. This compiler should produce functional code for uniprocessor targets, multicore architectures, clusters of workstations, and the MIT Raw processor. There is also a runtime library for testing programs through a Java compiler.

There are two versions of this release. These are:

streamit-src-2.1.1.tar.gz "Source release." This contains the full sources of the StreamIt compiler and runtime libraries, but no binaries. Build instructions are included in the INSTALL file.

streamit-2.1.1.tar.gz "Binary release." This contains no sources, but does contain a Java jar file with the full compiler.

The StreamIt compiler is built on top of version 1.5B of the Kopi Java compiler from DMS Decision Management Systems GmbH. The independent components of the system, including the scheduler, Java runtime library, C/C++ libraries, the front end, and the StreamIt .str files, are released subject to the terms of the MIT license agreement. Other components of the system are released under the terms of the GNU General Public License; see the COPYING file for details. The binary release also include class files from the ANTLR LL(k) parser generator.

Several documents on StreamIt are in the docs directory. You can find more information on our Web page, at `http://cag.csail.mit.edu/streamit/`.

2 Dependencies

The StreamIt makefile infrastructure uses the GNU tools, which are available from `http://www.gnu.org`. In particular, GNU Make is required to build; this is the default ¡tt¿make¡/tt¿ on Linux machines, and may be available as ¡tt¿gmake¡/tt¿ on other Unix-like machines. The source release depends on Sun's Java compiler (javac). Both the source and binary releases require a Java virtual machine for Java version 1.5 or later; see `http://java.sun.com/` for possibilities.

The front end depends on the ANTLR LL(k) parser generator; see `http://www.antlr.org/`. If you are using the source release, the ANTLR classes must be in your Java CLASSPATH when compiling and running the StreamIt compiler. If you are using the binary release, the required ANTLR classes are already included in the StreamIt jar file.

The linear optimizations (not turned on by default) depend on the FFTW library; see `http://www.fftw.org/`. The StreamIt runtime system currently only supports fftw-2.1.x, not the newer fftw-3.0 branch. FFTW should be installed somewhere in your default compiler search path. You need to build single-precision versions of the library, such that you have include files sfftw.h, srfftw.h and library files libsfftw.a and libsrfftw.a. The linear optimizations are turned on only if `configure` can find all four files. If you are building FFTW from source, pass the following options to configure: `--enable-shared --enable-type-prefix --enable-threads --enable-float`.

While this is not a a dependency, the `graphviz` package from `http://www.graphviz.org` contains a viewer, `dotty`, that will display the dot files created by the StreamIt compiler.

3 Caveats

This is a *snapshot* release of the StreamIt compiler. This is a research compiler; as such, it has several known shortcomings:

- The compiler works by converting StreamIt syntax to a Java intermediate form, and then using a Java compiler. Of note, compiling 'foo.str' will clobber a file named 'foo.java' in the same directory.
- The dimensions of a filter field cannot be declared in terms of another filter field. That is, code of the following form will fail:

```
float->float filter Foo {
  int N = 10;
  float[N] coeff;
  ...
}
```

In such cases, just declare the array to be `float[10]`.

- In splitjoins and feedbackloops, splitters and joiners cannot be specified from within conditional code. For example, the following code causes problems:

  ```
  float->float splitjoin Bar {
    if (...) {
      split duplicate;
    } else {
      split roundrobin;
    }
  }
  ```

  If you really need such a conditional, then define two separate splitjoins (Bar1 and Bar2) and do the test when adding them to a parent stream:

  ```
  float->float pipeline Parent {
    if (...) {
      add Bar1(); // uses duplicate splitter
    } else {
      add Bar2(); // uses roundrobin splitter
    }
  }
  ```

- The compiler does not support passing structures or complex numbers as stream parameters. This will result in an error such as

  ```
  at.dms.util.InconsistencyException: Expected constant arguments to
  init, but found non-constant VarExp:p in parent SIRPipeline
  name=pipe_3
  ```

  Structures and complex should work as local variables and as the input or output types of filters.

- Arrays cannot be returned from helper functions.

- In most back ends, arrays cannot be used as the input or output types of streams. This is largely a limitation in the uniprocessor back end; you can get equivalent code by using streams of the base type of the array, and constructing the array inside the work function if necessary.

- The `static blocks` are limited (except in the Java library back end) in that they act as specified only if every declared variable in the static block is assigned to exactly once. Furthermore, there can currently be at most one static block in a program.

4 For more info

The StreamIt home page is at `http://cag.csail.mit.edu/streamit/`. There are a number of places you can send electronic mail to:

streamit@cag.csail.mit.edu General information on StreamIt, to report bugs in the compiler or request new language features, or to be added to streamit-users or streamit-dev

streamit-users@cag.csail.mit.edu Discussion list for application developers and others using the StreamIt compiler

streamit-dev@cag.csail.mit.edu Discussion list for people working on the internals of the StreamIt compiler