Fortress (Installation notes and Stream source code) 2008-02-17

Main Resources

http://projectfortress.sun.com/Projects/Community/wiki`

http://research.sun.com/projects/plrg/

Setup:

1. Download and install latest JDK from Sun www.
2. Download and uncompress apache-ant http://ant.apache.org/
3. Download and uncompress junit http://www.junit.org/
4. Set JAVA_HOME and ANT_HOME and junit.tar in the CLASSPATH
5. Check out Fortess svn checkout https://projectfortress.sun.com/svn/Community/trunk PFC
6. cd PFC/ProjectFortress
7. ./ant clean
8. ./ant compile
9. ./ant test (optional)
10. Set FORTRESS_HOME to you PFC dir
11. Set PATH ~PFC/bin
12. Test hello.fss

```
<dmz02>$ fortress hello.fss
Parsing hello.fss: 12 milliseconds
Hello, World!
Program execution: 2764 milliseconds
```

```
hello.fss

component hello
export Executable

run(args:String...) = println "Hello, World!"

end
```

Stream example (with 64-bit floats)

4 separate implementations for each Copy, Scale, Triad and Add tests since timers to instrument code are currently not available.

```
triad[\RR64,nat N\]() = do

  scalar: RR64 = 3

  a = array[\RR64\](N)
  a.fill(1)

  b = array[\RR64\](N)
  b.fill(2)

  c = array[\RR64\](N)
  c.fill(0)

(* Triad *)

  for i <- 0#N do
     a[i] := b[i] + scalar c[i]
  end

end

run(args:String...) :() = do

  triad[\RR64,200\]()

end
```

Key concepts in Stream implementation

- Integers ZZ32
- Double RR64
- For loop constructs
- Multiplication
- For loop construct (default parallel)
- Many alternate implementations possible

```
<dmz02>$ fortress stream_triad.fss
Parsing files: stream_triad.fss
Parsing stream_triad.fss: 383 milliseconds
Program execution: 53987 milliseconds
```

Generic Implementation (for different data types: ZZ, RR, etc.)

```
triad[\Elt extends Number, nat N
    \](): () = do

  s: Elt = 3

  a = array[\Elt\](N)
  a.fill(1)

  b = array[\Elt\](N)
  b.fill(2)

  c = array[\Elt\](N)
  c.fill(0)

(* Triad *)

  for i <- 0#N do
     a[i] := b[i] + s c[i]
  end

end
```

To do

- High-level performance comparison for alternate implementations and output from stream.c