difxspeed

A new tool, difxspeed (attached to this wiki page) has been developed that facilitates benchmarking and parameter optimization of DiFX. The program is provided a file describing the tests to perform that looks like the following:

```
datastreams = node-2,node-2,node-2,node-2,node-2
cores = node-4,node-5
antennas = BR,FD,HN,KP

nAnt = 4
nCore = 2
nThread = 3
tInt = 1
specRes = 0.125
fftSpecRes = 0.125
xmacLength = 16,32,64,128
strideLength = 8,16,32
numBufferedFFTs = 1,2,4,8

vex = dq224.vex
```

This file must end with extension `.difxspeed` . This file can be run with command:

```
difxspeed dq224.difxspeed
```

This file must end with extenThis program takes a given `.vex` file, creates a series of `.v2d` files, one for each potential configuration as described by the parameter options, creates DiFX file sets, and executes them. For the above example, DiFX will be invoked 49 times. The first time is a *dummy* run, used to make sure all of the equipment is “warmed up” (and this can make a difference), and then there is a run followed by each of the 48 combinations of `xmacLength`, `strideLength`, and `numBufferedFFTs` suggested by the file. By default, correlation is always run in *fake* mode where no data is actually read, rather randomly generated data is pushed through the system. See below for running with real data. Most of the parameters in this file match those of a `.v2d` file.

Running this set of benchmarks will produce an output file (ending in `.out`) which may resemble:

```
# Cluster definition:
# datastreams = node-2 node-2 node-2 node-2 node-2
# cores = node-4 node-5
# antennas = BR FD HN KP

# Fixed parameters:
# tInt = 1
# nCore = 2
# nThread = 3
# vex = dq224.vex
# fftSpecRes = 0.125
# nAnt = 4
```

```
# specRes = 0.125

# Table columns:
# 1  xmacLength
# 2  strideLength
# 3  numBufferedFFTs
# 4  Average execute time (seconds)
# 5  RMS of execute times (seconds)
# 6...  Individual execute times (seconds)
16 8 1  173.316333 11.533935  167.477000 170.374000 198.990000 168.518000
167.000000 167.539000  # dq2-dummy
16 8 1  168.994833 1.587027  168.248000 169.449000 166.009000 170.317000
169.032000 170.914000  # dq2-001
32 8 1  153.121833 1.754966  152.973000 150.302000 153.099000 155.327000
151.882000 155.148000  # dq2-002
64 8 1  144.864333 0.820851  145.432000 144.675000 145.414000 143.498000
144.241000 145.926000  # dq2-003
128 8 1  140.450833 1.595531  141.689000 138.555000 142.672000 138.522000
141.478000 139.789000  # dq2-004
16 16 1  161.438333 1.277240  162.031000 159.841000 162.904000 162.690000
159.659000 161.505000  # dq2-005
32 16 1  147.192833 1.641706  147.618000 147.069000 149.917000 145.065000
148.090000 145.398000  # dq2-006
64 16 1  139.531167 0.702910  138.658000 139.798000 140.628000 138.639000
139.572000 139.892000  # dq2-007
128 16 1  135.296833 2.163581  139.098000 132.751000 136.408000 135.570000
135.064000 132.890000  # dq2-008
16 32 1  162.137833 2.020091  165.909000 160.574000 160.380000 160.250000
163.111000 162.603000  # dq2-009
32 32 1  146.173000 0.869731  145.583000 146.854000 146.342000 144.651000
147.345000 146.263000  # dq2-010
64 32 1  138.041000 1.054704  138.733000 137.355000 137.407000 140.074000
137.020000 137.657000  # dq2-011
128 32 1  135.441333 1.175489  136.056000 135.924000 134.921000 134.397000
133.913000 137.437000  # dq2-012
16 8 2  162.960333 1.872134  160.249000 165.464000 164.852000 161.855000
163.776000 161.566000  # dq2-013
32 8 2  147.695500 1.031172  148.084000 147.786000 149.308000 146.001000
148.076000 146.918000  # dq2-014
64 8 2  140.857667 1.283726  139.261000 140.713000 142.267000 142.081000
139.101000 141.723000  # dq2-015
128 8 2  137.169333 2.135411  138.614000 135.088000 141.125000 135.113000
136.904000 136.172000  # dq2-016
16 16 2  155.763167 0.947386  155.550000 156.038000 155.843000 154.792000
157.587000 154.769000  # dq2-017
32 16 2  141.416167 0.743320  142.142000 140.400000 142.456000 140.687000
141.673000 141.139000  # dq2-018
64 16 2  134.592333 1.162151  135.721000 135.289000 132.636000 133.523000
134.621000 135.764000  # dq2-019
128 16 2  129.878000 1.714953  131.755000 127.032000 131.530000 130.647000
130.060000 128.244000  # dq2-020
```

```
16 32 2  154.156167 0.701789  153.809000 153.519000 154.926000 154.784000
153.126000 154.773000  # dq2-021
32 32 2  142.544500 0.924184  142.413000 141.210000 141.808000 143.850000
142.411000 143.575000  # dq2-022
64 32 2  134.336167 1.294484  132.663000 132.550000 134.623000 135.440000
135.929000 134.812000  # dq2-023
128 32 2  129.712667 1.545513  129.996000 129.518000 131.904000 126.714000
130.356000 129.788000  # dq2-024
16 8 4  159.267000 1.079145  157.869000 158.827000 158.232000 159.909000
159.724000 161.041000  # dq2-025
32 8 4  145.291667 0.853220  145.022000 146.220000 145.935000 145.178000
145.770000 143.625000  # dq2-026
64 8 4  137.506167 1.635699  135.677000 138.278000 135.558000 138.338000
140.215000 136.971000  # dq2-027
128 8 4  132.648500 1.874152  136.052000 132.965000 131.729000 130.824000
130.621000 133.700000  # dq2-028
16 16 4  152.580833 1.330866  151.177000 153.808000 154.512000 152.321000
150.776000 152.891000  # dq2-029
32 16 4  142.534833 5.768529  138.042000 154.916000 139.289000 141.397000
142.808000 138.757000  # dq2-030
64 16 4  131.260500 1.768340  133.423000 133.130000 131.366000 131.395000
129.972000 128.277000  # dq2-031
128 16 4  127.708833 1.688246  124.451000 127.947000 126.891000 129.732000
128.723000 128.509000  # dq2-032
16 32 4  153.663333 0.864594  153.347000 154.816000 152.001000 153.753000
154.115000 153.948000  # dq2-033
32 32 4  139.584833 1.800902  137.971000 141.756000 137.390000 140.782000
138.103000 141.507000  # dq2-034
64 32 4  131.773500 1.885981  132.505000 130.985000 132.531000 131.846000
128.280000 134.494000  # dq2-035
128 32 4  127.477500 1.145064  129.405000 127.688000 126.900000 125.946000
126.618000 128.308000  # dq2-036
16 8 8  157.735500 0.889988  157.684000 156.673000 158.945000 158.361000
156.507000 158.243000  # dq2-037
32 8 8  143.670333 1.359781  145.601000 142.081000 141.747000 143.766000
144.531000 144.296000  # dq2-038
64 8 8  137.096833 1.405056  135.903000 139.766000 136.591000 135.516000
137.823000 136.982000  # dq2-039
128 8 8  131.898000 0.909325  132.055000 132.183000 130.984000 132.989000
130.440000 132.737000  # dq2-040
16 16 8  152.148000 0.789127  152.105000 153.515000 151.217000 151.503000
151.749000 152.799000  # dq2-041
32 16 8  138.139167 1.617144  139.700000 136.463000 139.164000 140.204000
137.247000 136.057000  # dq2-042
64 16 8  131.431000 1.262607  130.196000 132.366000 130.705000 133.447000
130.441000  # dq2-043
128 16 8  126.720167 1.586434  124.873000 127.076000 124.452000 127.404000
127.477000 129.039000  # dq2-044
16 32 8  151.147500 1.748858  152.798000 152.331000 149.240000 149.135000
153.415000 149.966000  # dq2-045
32 32 8  138.268167 0.945924  138.183000 138.132000 137.222000 139.771000
```

```
137.160000 139.141000  # dq2-046
64 32 8  130.166500 1.286886  128.630000 131.164000 130.814000 131.897000
130.106000 128.388000  # dq2-047
128 32 8  125.817333 1.249397  124.964000 125.225000 125.255000 124.508000
128.090000 126.862000  # dq2-048
```

The above file is actually the result of running the same benchmarking script 6 times. The run times are extracted from the `.difxlog` files which are not erased after each run, this system can be used to construct a history of benchmarking results. The first set of columns in this file are the three parameters that are allowed to vary in this case. The next two are the average and RMS run times, and finally the individual trial run times are printed. A comment at the end of each line indicates the DiFX file set associated with each run.

Running with non-fake data

If performance of the entire correlation process, not just the number-crunching, is desired, two changes are needed to the file. First, fake mode needs to be disabled through inclusion of `fake=false` in the `.difxspeed` file. The second is including the antenna and datastream information, which is done much as its is in a regular correlation. The ANTENNA sections (and DATASTREAM sections, if used) can be explicitly transferred from the `.difxspeed` file. To do this, include a line in the `.difxspeed` file consisting entirely of `--` (two hyphens). All lines of text below this are copied verbatim to the bottom of the `.v2d` file.

An shortened example demonstrating non-fake data correlation follows:

```
datastreams = node-2,node-3
cores = node-4,node-5
antennas = BR,FD

nAnt = 4
nCore = 2
nThread = 3
tInt = 1
specRes = 0.125
fftSpecRes = 0.125
xmacLength = 16,32,64,128
strideLength = 8,16,32
numBufferedFFTs = 1,2,4,8

vex = dq224.vex

fake = false

--

ANTENNA BR { vsn=NRAO+234 }
ANTENNA FD { file=dq224_fd_01.mark5b }
```

Some notes of interest

- Once in a while (e.g., dq2-043 in the above output) the full runtime seems not to be captured. This is being investigated. The problem appears to be in the capture of the multicast messages by the `difxlog` program.
- The files produced invoke `singleScan=false`, so a multiple scan `.vex` file can be used.
- Things will work poorly (if at all) if more than 1 setup is used within the `.vex` file.