Correlator Use Case Scenarios for Mark 6

RJ Cappallo
2013.11.22

There are many different ways that one might imagine using mk6 data in conjunction with a difx software correlator, depending on available software modules (some of which are Not Yet Implemented), the characteristics of the data, and the hardware resource constraints at the correlator. The principal factors to be considered are:

data format

- mk5b
- vdif

In the mk6 system, mk5b data are converted on the fly into vdif records. This involves splicing together 2 packets that are the two halves of a mk5b record, converting the sample format encoding for all samples, and embedding the resulting mk5b record within a vdif wrapper. Furthermore, each mk5b stream is assigned a thread number, from 0 through 3 in the case of 4 input streams; these threaded records are scattered throughout the mk6 fileset.

Currently, as of 2013.11.22, difx is unable to handle multithreaded vdif when there is more than 1 channel per thread. Until this capability is developed, it is necessary to break the mk6 file down into 4 files, each containing 1 stream, an operation that is accomplished as an option of the dqa program. The necessary operations would be no different if the data originated as multichannel, multithreaded vdif.

mk6 file mode

- scatter-gather
- RAID

If the mk6 was operated in RAID mode, then it will be possible to mount the disk module(s) as a RAID filesystem, perhaps then using NFS, and to access it as linux-based files. If instead, the data were scattered over multiple disks, a data-assembly step is necessary.

available options for gathering scattered data

- gather program
- FUSEmk6 filesystem (NYI)
- native-mode difx reader for mk6 (NYI)

When the mk6 is operated is sg mode, the scattered files must be reassembled into a single file. This can be done prior to correlation with the stand-alone gator script (which invokes the gather program, perhaps multiple times for multiple scans). The natural output target for gator is a RAID array, either external to the mk6, or perhaps as the other module in case the input group consists of a single module. A FUSEmk6 filesystem is under development to do this reassembly more transparently as part of the access of the filesystem.

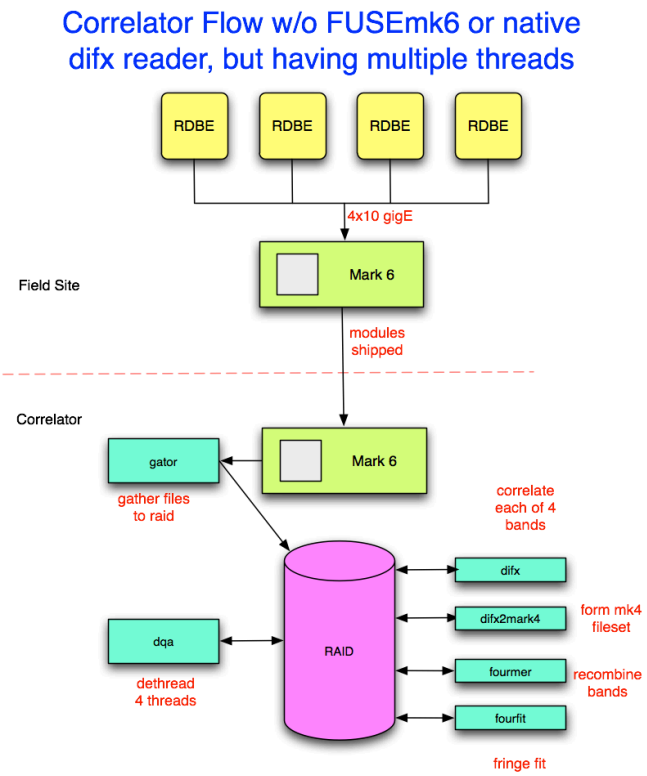
Ultimately, it would be nice to have a native difx capability for mk6, wherein a difx datastream object would be able to handle mk6 modules, of either RAID or sg format.

local storage for data

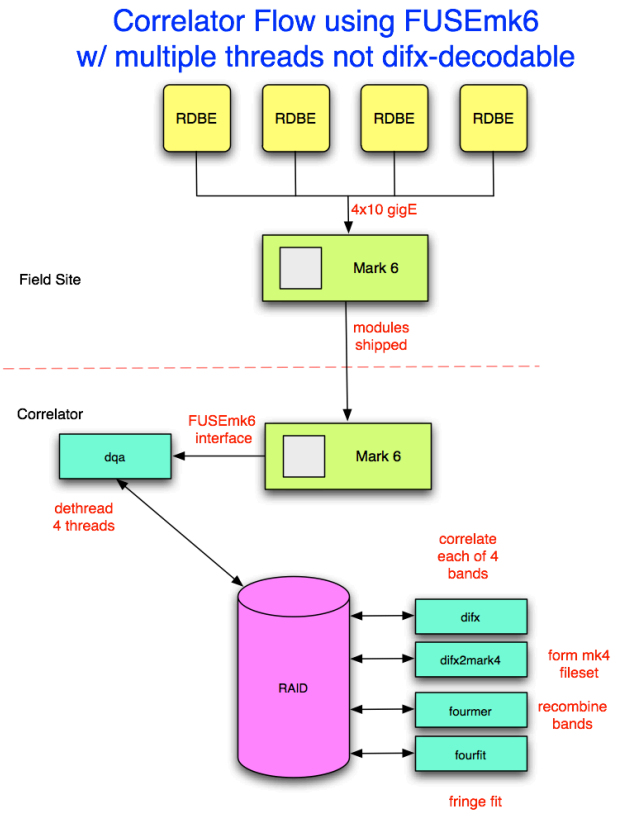
- separate RAID system
- mk6 modules

Using a mk6 module as an output RAID may make sense when there is only 1 input sg module. It would be necessary to mount the module as RAID, and that currently is not handled automatically by the gator program, though it is a capability likely to be added in the near future.

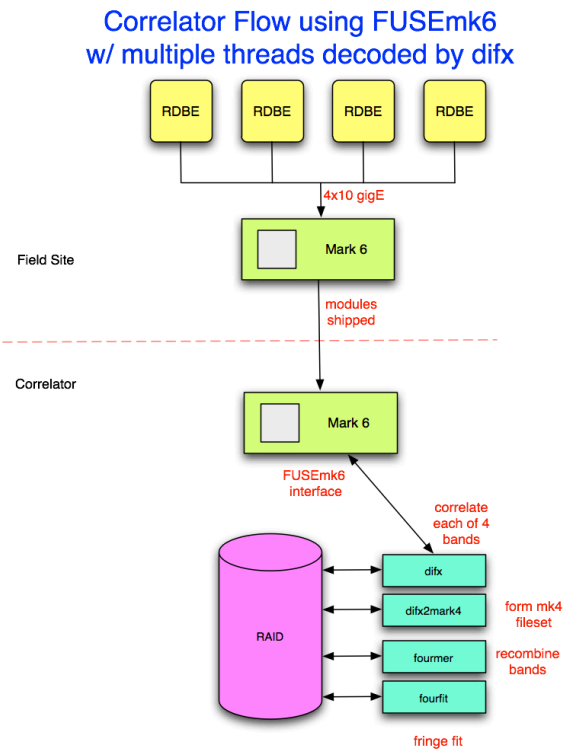
There are 4 Use Cases depicted below, which comprise a subset of the possible combinations derivable from the above factors. The intent is to shed some light on the use of the tools that are (or will be) available. UC1 is the principal near-term pathway for VGOS data, in that all parts of the pipeline are complete and known to work. However this is just an interim scheme, as it requires more total I/O than is strictly necessary, by a factor of a few.



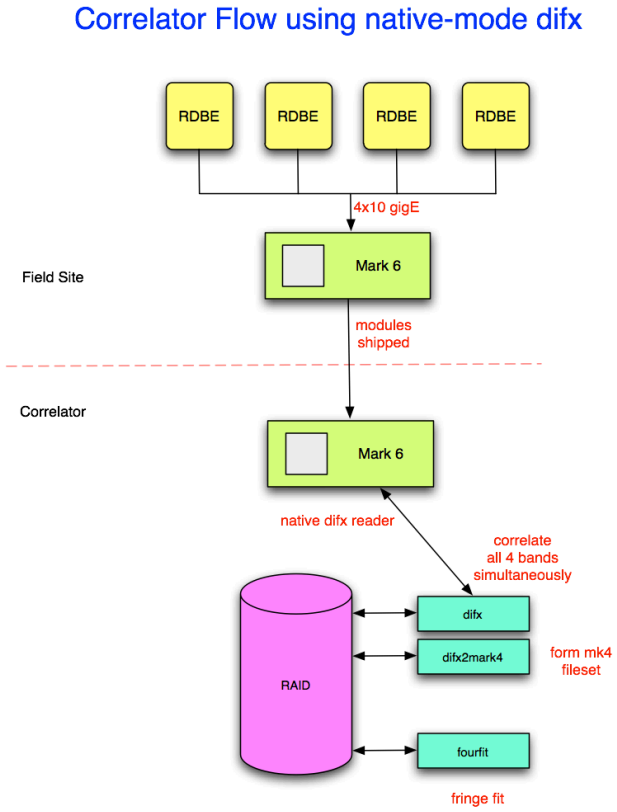
UC1: scatter-gather (sg) files are reassembled by gator running on a mk6 and written to NFS-mounted RAID. Files are dethreaded by dqa and written back to RAID. difx correlates each band (= thread) of 16 channels separately, and data are merged by fourmer into single filesets to be fringe fit with fourfit.



UC2: sg files are reassembled by FUSEmk6 file-system running on a mk6, read into dqa, where the threads are separated into separate files and written to NFS-mounted RAID. difx correlates each band (= thread) of 16 channels separately, and data are merged by fourmer into single filesets to be fringe fit with fourfit.



UC3: sg files are reassembled by FUSEmk6 file-system running on a mk6, and read directly by difx. This requires difx to have the capability to correlate multiple (4) vdif threads, each having multiple (16) channels. The data then no longer need to be merged by fourmer, and the output of difx2mark4 consists of single filesets with 64 channels of correlated data. These filesets are then fringe-fitted with fourfit.



UC4: sg files are reassembled upon reading by difx. This requires code for a native-mode difx datastream handler. It also requires difx to have the capability to correlate multiple (4) vdif threads, each having multiple (16) channels. The data then no longer need to be merged by fourmer, and the output of difx2mark4 consists of single filesets with 64 channels of correlated data. These filesets are then fringe-fitted with fourfit.