DiFX Performance Testing

Introduction

This document outlines some performance testing of the DiFX software correlator. Tests were run on the Curtin CUPPA cluster. This is a 20 node Beowulf cluster of quad core, dual processor Intel server (2.66 GHz Xeon X5355 processors). Testing was done with a fake 6 station LBA array with data nominally "recorded" at 512 Mbps, specifically eight 16 MHz subbands with 2 bit sampling. Cross polarisation products were **not** computed. The data correlated was generated on the fly and fed into DiFX as an eVLBI data stream, using the loopback network interface. Simple tests show data can be generated at around 8 Gbps per antenna so the data generation overhead is a negligible overhead for performance. If disk based data had been used then the effect of disk i/o would have a much larger impact on the results. Data from each telescope was run on a dedicated i/o node separate from the compute nodes.

The tests were run in an automated fashion using a simple Bourne shell script and the output of DiFX and the fake VLBI data generator programs logged. The reported rates are the playback rate from each telescope, calculated as the median transfer rate reported by the data generation program every couple of seconds (this is to overcome some cache filling issues at the start and end of correlation). Tests were run with 1 to 12 compute nodes, each with 1-8 compute threads. Two minutes of data were correlated for each test (two minutes of data, not 2 minutes wall clock).

Scaling with nodes and threads

Figure 1. shows the scaling of DiFX with number of threads. Each line shows the thread scaling with different number of compute nodes (from 1-12). With up to 10 compute nodes we see the speed increases when adding compute threads up to 6 threads when the performance levels off or slightly decreases. 6 threads are roughly 4 times faster than a single thread.

Figure 2 shows the same data, with the axes transposed to highlight the affect of scaling with compute nodes. The scaling is quite linear and very close to a 1-1 ratio of number of nodes to speed increase, until it gets to the maximum number of nodes/threads. It seems that the i/o nodes saturate at an output data rate of around 850 Mbps. This is slightly below the maximum value expected on gigabit Ethernet (950 Mbps) and indicates either MPI overheads or, more likely, inefficiencies (e.g. deadtime) in the data distribution approach. From these results, with 6 stations to process there is no point using more than 10 compute nodes.

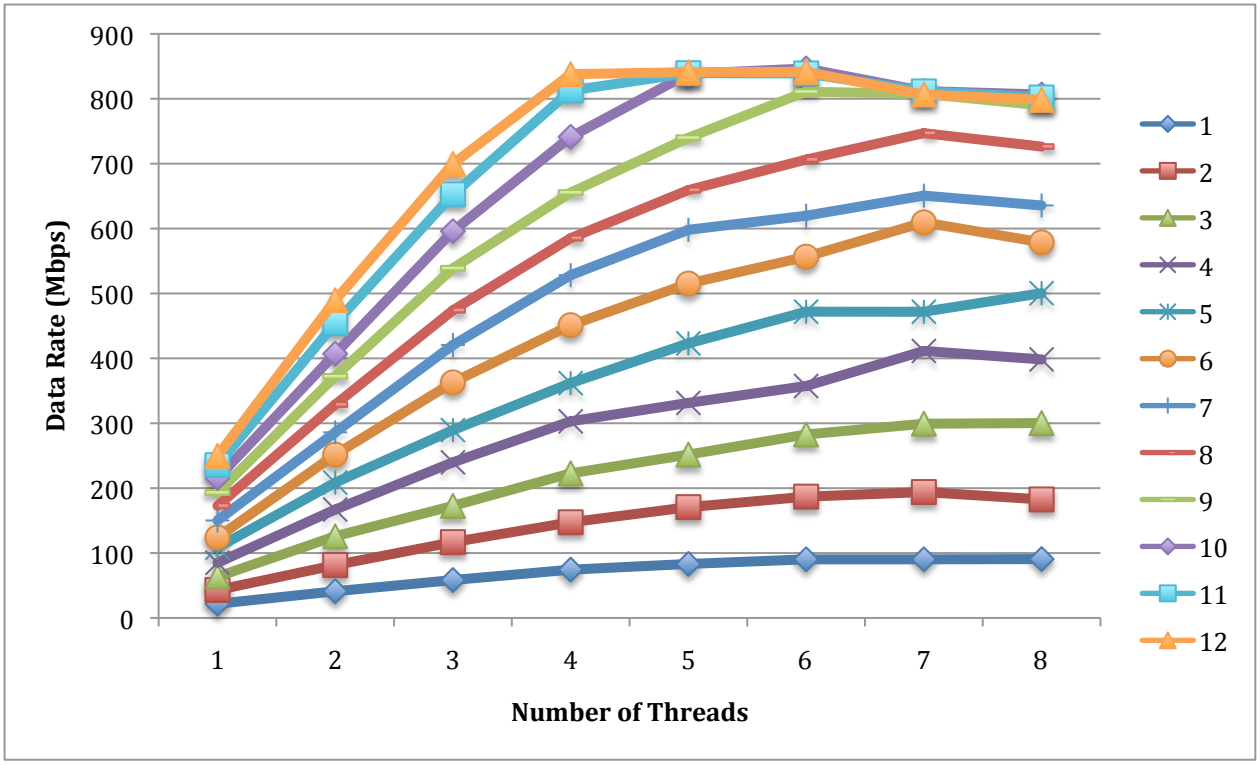

Figure 1: Scaling with threads with 1-12 compute nodes

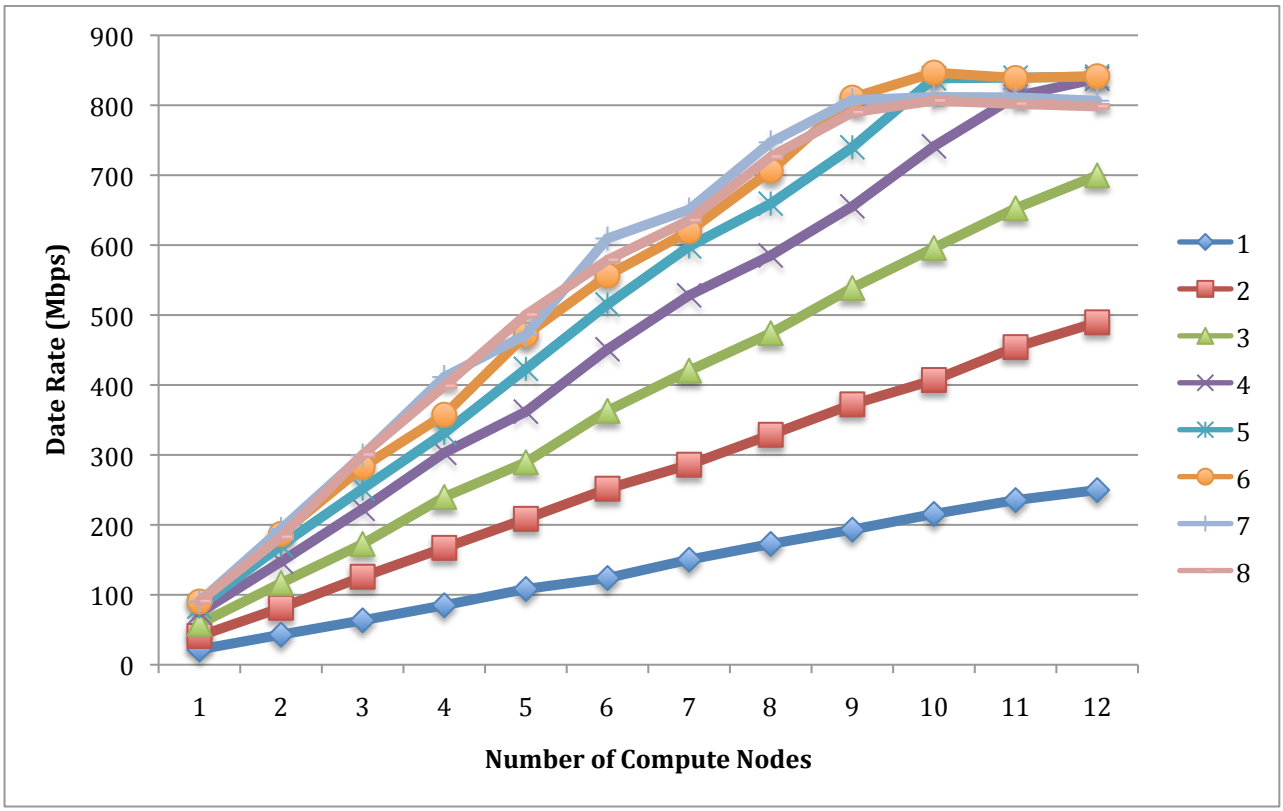

Figure 2: Scaling with compute nodes, with 1-8 threads

Figure 3. shows the same data as Figure 2, but the data rates are normalised by the number of compute nodes – ie it shows the data rate per compute node. Data for 7 and 8 threads has been omitted for clarity. This shows rates of up to 600 Mbps can be processed on a single node.

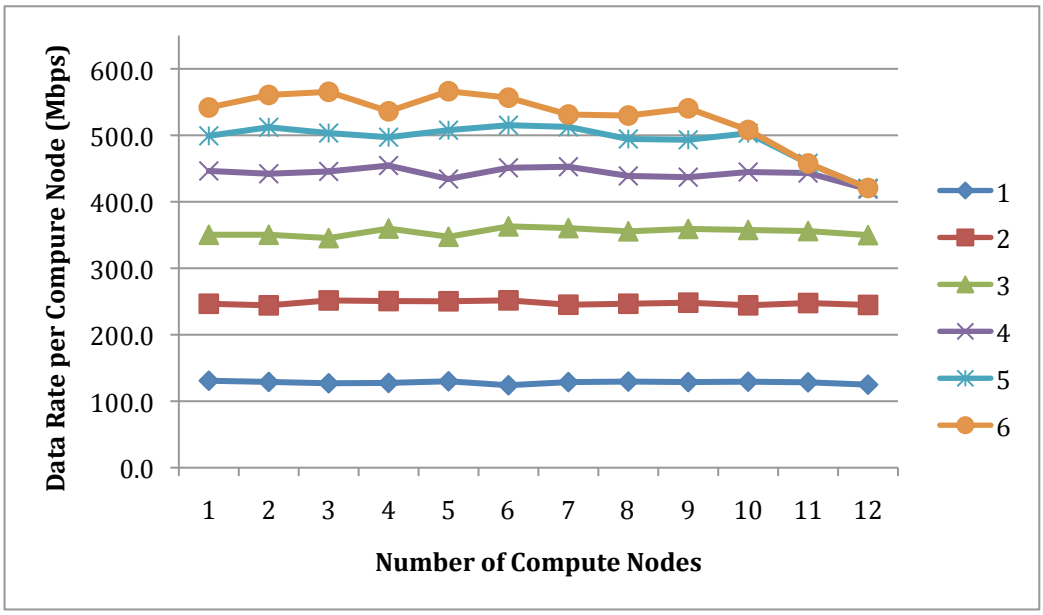

Figure 3: Data Rates per compute node

Scaling with spectral points

To test scaling with number of spectral points per channel (ie FFT size), Tests with 8 computer nodes and 6 threads were run. 16 to 2048 spectral points per channel were used for the test in powers of 2. A Figure 4 shows, peak efficiency is between 64 and 256 spectral points and drops off quite quickly outside that range. This is presumably dependent of CPU cache size and IPP implementation details.

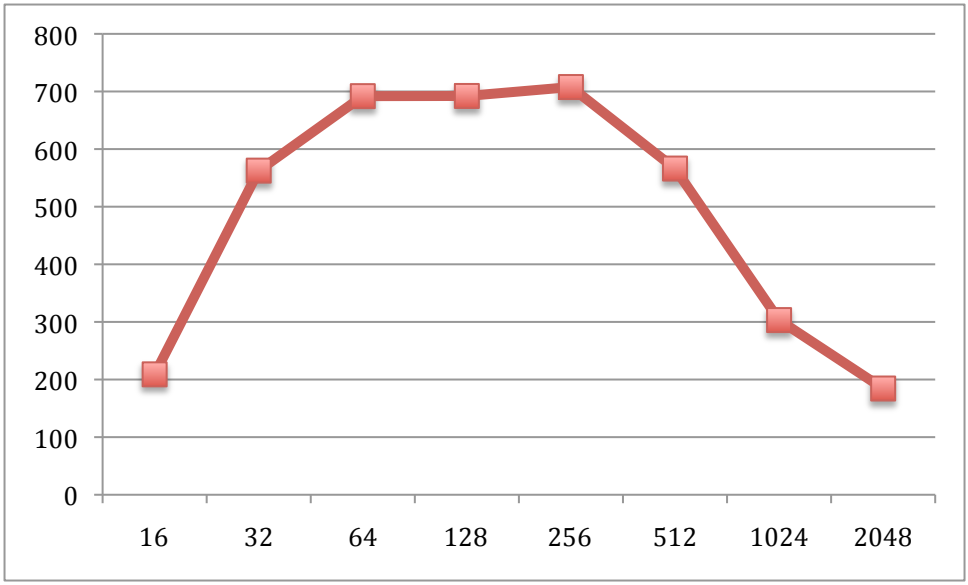

Figure 4: Scaling with number of spectral points (FFT size)

Conclusions

Best efficiency is achieved with 6 compute threads per compute node. Thread scaling is very poor – though this is presumably an issue with the architecture rather than DiFX. With Gigabit Ethernet interconnects, data rate from i/o nodes saturates at around 850 Mbps. Scaling past this will require faster interconnects (e,.g multiple 1 Gbps connections or 10 Gbps Ethernet). Dual processor quad core compute nodes can process data up to about 600 Mbps. This implies the next generation of CPUs (Nehalem) will break the 1 Gbps/node and standard gigabit Ethernet interconnects will become a bottleneck.