Benchmarking Difx

John Morgan

Istituto di Radioastronomia, Bologna, Italy

3rd DiFX Meeting
Perth
Sept/Oct 2009

Aims

1. To look at bottlenecks limiting DiFX
   → Usually reading from media but not always
2. How to diagnose them
3. How to be sure you get the most out of the cluster

How?

1. Run (1st, 2nd and) 3rd party software to characterise the cluster (and mark5s/network etc)
2. Know what kind of difx performance you can expect
3. Know what you should be choosing for `BLOCKS PER SEND` etc.

   → Preferred send size could even be specified in v2d

Factors which could affect DiFX performance (in principal)

1. Reading data from media/network
2. Transfering data datastream->core
3. CPU power
4. Writing to disk

Reading Data

Symptoms:-

⇒ Disk read at top speed

→ ganglia
→ nagios

When could this be a limiting factor?

⇒ Most of the time

Transferring data datastream->core

Symptoms

⇒ Maxed out network

→ top
→ ganglia
→ nagios

When could this be a limiting factor?

⇒ vlbi_fake (Chris Phillips)

⇒ ???

CPU power

Symptoms:-

⇒ CPUs maxed out

→ top
→ ganglia
→ nagios
→ difxmessage

When could this be a limiting factor?

⇒ Huge number of channels??

⇒ Vlbi_fake (Chris Phillips)

Writing out to disk

Symptoms:-

⇒ Visbuffer fills up then correlator grinds to a halt

When could this be a limiting factor?

⇒ When you're writing out to a stupid place.

Reading from media/network

We need to know:-

- ⇒ Maximum speed
- ⇒ Optimum read size

Testing

- ⇒ dd
- ⇒ Bonnie++
  - → Read/write speed
  - → Latency
  - → Run multiple times to get best disk read size

bonnie++

```
./bonnie++ -d /fs0/difx
```

Sequential Output			Sequential Input			Random Seeks		
Speed	latency	CPU	Speed	latency	CPU	Speed	latency	CPU
92002	3412ms	19	94209	347ms	13	584.9	218ms	13

Transferring data datastream->core

We need to know:-

- ⇒ Network speed
- ⇒ Network latency
- ⇒ Optimum mpi packet size

Testing

- ⇒ ttcp
- ⇒ netPIPE

```
wn01$  ttcp -t -s -fg -n640000 -u wn02
ttcp-t: buflen=8192, nbuf=640000, align=16384/0, port=5001  udp  -> wn02
ttcp-t: socket
ttcp-t: 5242880000 bytes in 43.76 real seconds = 0.89 Gbit/sec +++
ttcp-t: 640006 I/O calls, msec/call = 0.07, calls/sec = 14624.35
ttcp-t: 0.2user 12.5sys 0:43real 29% 0i+0d 0maxrss 0+4pf 87559+30csw
```

```
wn01$ mpirun -np 2 --machinefile wn.machine NPmpi -I
Performance measured without cache effects
0: wn01
Performance measured without cache effects
1: wn02
Now starting the main loop
  0:       1 bytes   1407 times -->      0.12 Mbps in      62.57 usec
  1:       2 bytes   1598 times -->      0.24 Mbps in      62.69 usec
  2:       3 bytes   1595 times -->      0.37 Mbps in      62.45 usec
  3:       4 bytes   1067 times -->      0.47 Mbps in      64.77 usec
  4:       6 bytes   1157 times -->      0.73 Mbps in      62.42 usec
...
123: 8388611 bytes      3 times -->   1684.40 Mbps in   37995.85 usec
```

CPU speed

We need to know:-

⇒ how good the cpu is at vector crunching

Testing

⇒ lapack

⇒ non-difx (Chris)

Summary

Do tests so we know the maximum our correlator could handle if:-

- ⇒ Disk limited
- ⇒ Network limited
- ⇒ CPU limited

Know exactly how to optimise our parameter files.

Discussion

1. How do we get all this into automated script?
2. How do we build a database of all this?

References

1 http://www.coker.com.au/bonnie++/experimental/