Mark6 playback performance

Real correlation test (2023)

Test carried out during the 15th DiFX Users and Developers Meeting in 2023.

Setup (/Exps/TESTS/Helge/mk6speed):

- based on e22c20 (b4)
- two stations PV, SZ
- no zoom/outputbands
- modules in mark6-01 and mark6-02 (4 modules each, grouped)
- total scan duration: 120s

1. run: 38 nodes 19 threads

mk6mon peaks **at 16Gbps**

wallclock time: 223s

average processing rate: **8.5 Gbps**

2. run: 58 nodes 19 threads

wallclock time: 192s

average processing rate 10 Gbps

3. run 58 nodes 19,threads, FFTSpecRes=0.5

wallclock time: 188s

average processing rate 10 Gbps

4. run 58 nodes 19,threads, FFTSpecRes=0.5, neuteredmpifxcorr

wallclock time: 217s

average processing rate 10 Gbps

4. run 58 nodes 19,threads, FFTSpecRes=0.5, double read size

EHT e22c20 correlation test

Testing also carried out at Bonn with e22c20 b4 scans 1026, 1033, 1070 etc under DiFX 2.8.1 with actual correlation. The number of nodes was kept constant. Only v2d SETUP numBufferedFFTs and subintNS were altered. The DiFX directory was /Exps/e22c20/v1/b4_outputbands/perftest

Scans 1026, 1033 - only ALMA with 2 datastreams.
Scan 1019 - 5 stations x 2 datastreams.
Scan 1070 - 4 stations total, Ax 1 datastream, Gl 2, Nn 4, Pv 2 datastreams.

1. run: 1.6 ms subint, 20 buffered FFTs

e22c20-1-b4_1019 : 748.983 sec, 12.6x slowdown, MpiDone
e22c20-1-b4_1026 : 165.502 sec, 2.6x slowdown, MpiDone
e22c20-1-b4_1033 : 147.769 sec, 2.5x slowdown, MpiDone
e22c20-1-b4_1070 : 2267.63 sec, 7.6x slowdown, MpiDone

2. run: 3.2 ms subint (EHT default), 20 buffered FFTs

e22c20-1-b4_1019 : 507.172 sec, 8.5x slowdown, MpiDone
e22c20-1-b4_1026 : 120.792 sec, 1.9x slowdown, MpiDone
e22c20-1-b4_1033 : 100.509 sec, 1.7x slowdown, MpiDone
e22c20-1-b4_1070 : 1522.75 sec, 5.1x slowdown, MpiDone

3. run: 8.0 ms subint, 20 buffered FFTs

e22c20-1-b4_1019 : 362.91 sec, 6.1x slowdown, MpiDone
e22c20-1-b4_1026 : 89.3295 sec, 1.4x slowdown, MpiDone
e22c20-1-b4_1033 : 78.328 sec, 1.3x slowdown, MpiDone
e22c20-1-b4_1070 : 1048.75 sec, 3.5x slowdown, MpiDone

4. run: 40.0 ms subint, 20 buffered FFTs

e22c20-1-b4_1019 : 264.533 sec, 4.4x slowdown, MpiDone <-- approx 2x faster than default EHT (2. run)
e22c20-1-b4_1026 : 74.2592 sec, 1.1x slowdown, MpiDone
e22c20-1-b4_1033 : 67.7457 sec, 1.1x slowdown, MpiDone
e22c20-1-b4_1070 : 780.338 sec, 2.6x slowdown, MpiDone

5. run: 40.0 ms subint, 4 buffered FFTs

e22c20-1-b4_1019 : 268.615 sec, 4.5x slowdown, MpiDone
e22c20-1-b4_1026 : 70.0171 sec, 1.1x slowdown, MpiDone
e22c20-1-b4_1033 : 62.7848 sec, 1.1x slowdown, MpiDone

6. run: 40.0 ms subint, 10 buffered FFTs, dataBufferFactor 48, visBufferLength 20

e22c20-1-b4_1026 : 73.3848 sec, 1.1x slowdown, MpiDone

7. run: 40.0 ms subint, 100 buffered FFTs, default dataBufferFactor and visBufferLength

e22c20-1-b4_1026 : 72.5849 sec, 1.1x slowdown, MpiDone

RDMA Testing (2020)

The pure file-to-Infiniband connectivity performance can be tested with various RDMA based file transfer utilities. One compact transfer utility is https://github.com/JeffersonLab/hdrdmacp. The build needs CentOS package *rdma-core-devel*.

```
sudo yum install rdma-core-devel
git clone https://github.com/JeffersonLab/hdrdmacp
cd hdrdmacp; g++ -I . -g -std=c++11  -o hdrdmacp *.cc -libverbs -lz
```

The server that receives files can be started e.g. on fxmanager.
Need to specify a buffer set of e.g. 4 buffers (-n 4) each sized 4MB (-m 4):

```
./hdrdmacp -s -n 4 -m 4
```

Transfer speed from a FUSE-mounted 2 x 8-disk Mark6 module pair (EHT 2018, RCP slot 1&2) can be tested with e.g.

```
ssh oper@mark6-04
cd ~/jwagner/hdrdmacp/
fuseMk6 -r '/mnt/disks/[12]/*/band1/' /`hostname -s`_fuse/b1/12
vdifuse -a /tmp/label.cache -xm6sg -xrate=125000 -v /mark6-04_fuse/
vdifuse_12/   /mnt/disks/[12]/*/band1/
./hdrdmacp -n 4 -m 4 /mark6-04_fuse/b1/12/e18g27_Sw_117-0737.vdif
fxmanager:/dev/null
./hdrdmacp -n 4 -m 4 /mark6-04_fuse/vdifuse_12/sequences/e18g27/Sw/
117-0737.vdif fxmanager:/dev/null
dd if=/mark6-04_fuse/b1/12/e18g27_Sw_117-0737.vdif bs=4M of=/dev/null
mk6gather -o - "/mnt/disks/[12]/*/band1/e18g27_Sw_117-0737.vdif" | pv >
/dev/null
```

Performance of RDMA from a local FUSE based file into remote /dev/null vs local /dev/null:

Client	Server (dest.)	Rate (fuseMk6->rdmacp->dest)	Rate (vdifuse->rdma
mark6-04:/fuse	fxmanager:/dev/null	Transferred 308 GB in 198.8 sec (**12.41 Gbps**)	Transferred 308 GB in 267.4
mark6-04:/fuse	mark6-04:/dev/null	Transferred 308 GB in 207.4 sec **(11.89 Gbps)**	Transferred 308 GB in 283.1 **Gbps)**

Performance of RDMA from a non-FUSE file into remote /dev/null:

Source	Server (dest.)	Rate (file->rdmacp/dd->dest)
mark6-04:/data (**beegfs**)	fxmanager:/dev/null	rdmacp: Transferred 308 GB in 190.90 sec **(12.92 Gbps)**
io11:/data11/ (**hw RAID**)	fxmanager:/dev/null	rdmacp: Transferred 49 GB in 35.40 sec (**11.10 Gbps**)

Performance of RDMA from FUSE into "remote" beegfs:

Source	Server (dest.)	Rate (file->rdmacp/dd->dest)
mark6-04:/fuse (fuseMk6)	fxmanager:/data/rdma.vdif (beegfs)	rdmacp: Transferred 308.29 GB in 323.24 sec (**7.63 Gbps**)

Plain non-RDMA performance into local /dev/null:

Source	Method	Rate (file -> /dev/null)
mark6-04:/fuse (fuseMk6)	local *dd* copy to /dev/null	dd: 308 GB copied, 206.139 s, 1.5 GB/s (12 Gbps)
mark6-04 (mk6gather)	*mk6gather* via *pv* to /dev/null	mk6gather|pv: 1.50 GB/s (12 Gbps)
io11:/data11/ (hw RAID)	local *dd* copy to /dev/null	dd: 49 GB copied, 29.631 s, 1.7 GB/s (13.6 Gbps)

Test: Swaping modules

In order to determine if the different playback speeds are due to differences in the mark6 units or tie to the data recorded on the modules two sets of modules (PV, AZ) were swapped.:

mark6-02	AZ: 1272 Mbps
mark6-03	PV: 3669 Mbps

Playback performance seems to be tied to the data on the module. Need to repeat the playback speed measurements with recently recorded data (e.g. from the DBBC3 recordings in the lab).

Comparison: Fuse/Gather

Mark6 files were gathered on the fly and piped trough dd:

```
./jwagner/kvnvdiftools/gather-stdout/gather /mnt/disks/[1234]/*/data/
bf114a_Lm_142-0628.vdif - | dd of=/dev/null bs=1M count=100000
```

Results:

```
90000+10000 records in
90000+10000 records out
99921920768 bytes (100 GB) copied, 43.7195 s, 2.3 GB/s
```

Gathering yields much higher performance (=18 Gbps) than vdifuse (=.1.4 Gbps)

Using fuseMk6 instead of vdifuse:

```
fuseMk6 -r "/mnt/disks/[12]/*/data/" /home/oper/ftmp/
Found 258 scans, and 258 entries in JSON

dd if=/home/oper/ftmp/c22gl_Cr_081-0000.vdif of=/dev/null bs=1M count=1000
1048576000 Bytes (1,0 GB) kopiert, 0,480969 s, 2,2 GB/s

dd if=/home/oper/ftmp/w27us_Cr_086-1830.vdif of=/dev/null bs=1M count=1000
1048576000 Bytes (1,0 GB) kopiert, 0,464167 s, 2,3 GB/s
```

```
dd if=/home/oper/ftmp/w27us_Cr_086-1821.vdif of=/dev/null bs=1M
count=15000
15728640000 Bytes (16 GB) kopiert, 5,56799 s, 2,8 GB/s
```

iostat

on mark6-01 iostat finds the following:

Device:	tps	kB_read/s	kB_wrtn/s	kB_read	kB_wrtn
sda	12.12	1767.14	0.01	486307174	4096
sdb	12.12	1767.17	0.01	486315780	4096
sdc	12.12	1767.22	0.01	486328745	4096
sdd	12.12	1767.18	0.01	486316915	4096
sde	12.12	1767.13	0.01	486305259	4096
sdf	12.12	1767.13	0.01	486304968	4096
sdg	11.94	1768.72	0.01	486741075	4096
sdh	12.12	1767.09	0.01	486292692	4096
sdi	12.12	1767.13	0.01	486304984	4096
sdj	11.93	1768.20	0.01	486597699	4096
sdk	11.93	1767.90	0.01	486516615	4096
sdl	11.95	1770.25	0.01	487163049	4096
sdm	11.93	1767.95	0.01	486530620	4096
sdn	11.93	1767.94	0.01	486527604	4096
sdo	11.93	1767.94	0.01	486526314	4096
sdp	11.93	1767.86	0.01	486506020	4096
sdr	0.00	0.02	0.01	6119	4096
sds	11.94	1767.81	0.01	486490497	4096
sdt	11.94	1767.80	0.01	486487765	4096
sdu	11.95	1769.34	0.01	486911815	4096
sdv	0.00	0.02	0.01	6117	4096
sdw	0.00	0.02	0.01	6121	4096
sdx	0.00	0.02	0.01	6119	4096
sdy	0.00	0.02	0.01	6290	4096
sdz	0.00	0.02	0.01	6116	4096
sdaa	12.07	1767.16	0.01	486313319	4096
sdab	0.00	0.02	0.01	6119	4096
sdac	0.00	0.02	0.01	6117	4096
sdad	12.06	1767.11	0.01	486298721	4096
sdae	12.06	1767.13	0.01	486304411	4096
sdaf	12.06	1767.11	0.01	486300109	4096

The io performance of some disks is much lower than expected. The following mount logic applies (red are slow devices):

Module 1: g j k l m n o p

Module2: y aa ab ac ad ae af ag

Module 3: r s t u v w x z

Module 4: a b c d e f g h i

Repeat speed measurements on Mark6 lab machines

Mark6 machines in the correlator cluster have a redhat based OS installation. In order to check whether the differences in playback speed reported by Haystack and measured in Bonn are due to OS specific differences the speed tests were repeated on Mark6 machines running the original Debian installation.

Results: Playback speed < 1Gbbps

so the OS does not seem to be the reason for the slow playback speeds.

General IO tuning

Take a look at: http://cromwell-intl.com/linux/perfo...ing/disks.html

```
IO scheduler should probably be set to noop on all mark6 machines
```

Tested setting io scheduler to NOOP on mark6-05. No measurable difference in read performance

Hyperthreading

Repeated tests with Hyperthreading enabled & disabled. no significant difference in results.