



Reliability of jive5ab + m5copy

last “<transfer>?” status should always be available



Keeping time inside jive5ab

- sample rates + clocks are *rationals*
- deal correctly with > 4Gbps data rates

jive5ab

status update

Current stable release still 2.7.1

2.7.4-dev beta release to select group in July 2016

- holiday season, illness, other EU projects cause delay
- fallout from feedback [one (1) user]
 - details follow; also delaying release



Reliability of jive5ab + m5copy

last “<transfer>?” status should always be available



Keeping time inside jive5ab

- sample rates + clocks are *rationals*
- deal correctly with > 4Gbps data rates

High time resolution / data rates

Simple proof:

```
mode = vdif_8000-262144-1-1 ;  
!mode = 0;
```

That is: a **256Gbps** stream – 1 channel of 128GHz sampled at 1 bit / sample,
i.e. a **track bit rate of 256Gbps**

```
mode? ;  
!mode? 0 : vdif_8000-262144-1-1 : VDI : 1 : 262144.000 : 8000 ;
```

In favor of bean counting you get ...

Transient control connections

(necessary but not sufficient for the reliability fork)

- if transient connection closes:
 - transfers started from it are terminated
 - resources freed
 - sockets released
- `m5copy` updated to already make use of this feature

In favor of bean counting you get ...

```
scan_set = <label> ;
```

autodetects Mark6 or FlexBuff format

- i.e. don't have to tell jive5ab what recording format to expect anymore before `scan_check`? on either Mark6/FlexBuff

In favor of bean counting you get ...

Mark5 non-bank support
not just Mark5C

- bundle two disk packs for
 - 4Gbps recording on Mark5C
 - more capacity w/o switching disk packs
- not entirely implemented to MIT Haystack docs
 - IMHO that's impossible:
 - Haystack mental model of this mode \neq Conduant's reality
- coordination between Ed H., Jon Q. and yours truly
 - ensure Field System and `jive5ab` are on same page

In favor of bean counting you get ...

Mark5 non-bank support
not just Mark5C

- bundle two disk packs for
 - 4Gbps recording on Mark5C
 - more capacity w/o switching disk packs
- not entirely implemented to MIT Haystack docs
 - IMHO that's impossible:
 - Haystack mental model of this mode \neq Conduant's reality
- coordination between Ed H., Jon Q. and yours truly
 - ensure Field System and `jive5ab` are on same page

In favor of bean counting you get ...

Proper bank switch / module switch handling

bank/module switch did not reset current scan:
scan_check? w/o scan_set= would likely error

- new disk mounting:
 - reset current scan to nothing
 - select scan #1 if possible

In favor of bean counting you get ...

Significantly improved UDT performance on high BW x delay links

“32” changed into “375”

a two-character edit which pays off really well □

- UDT performance from .NL □ .NZ was limited to 800Mbps
 - 10Gbps path all the way with jumbo frames
 - eliminated HDD, network, CPU bottlenecks as candidate
- UDT socket buffer size in `jive5ab` increased
 - disk-to-disk performance jumped to 3.2Gbps!
 - expect better UDT performance for all

In favor of bean counting you get ...

Recording multiple FiLa10G streams is now a breeze!

Added a new 'protocol' `udpsnor` :

“udp with sequence number – no reordering”

- FiLa10G outputs VTP: VDIF with 64 bit sequence number
 - excellent for e-VLBI (packet loss/reordering)
 - multiple sequence number streams mess up reordering
- `udpsnor` is for local recording
 - expect no loss/reordering on local network
 - records packet payload straight through in order received:
 - sequence number read and accounted (exact loss)
 - bookkeeping done per unique sender

In favor of bean counting you get ...

Recording multiple FiLa10G streams is now a breeze!

- configure all FiLa10G streams to send to port *XXX* on FlexBuff
- set destination IP addresses to those of the FlexBuff/Mark6

```
fila10g = destination 0 aa.bb.cc.dd:XXX
```

```
fila10g = destination 1 aa.bb.cc.ee:XXX
```

...

Assume aa.bb.cc.{dd,ee,..} are FlexBuff IPv4 eth addresses

In favor of bean counting you get ...

Recording multiple FiLa10G streams is now a breeze!

- configure jive5ab to record `udpsnor` from port `XXX`

```
mode = VDIF_8192-16384-32-2
net_port = XXX
net_protocol = udpsnor : 768M : 128M
mtu = 9000
record = nthread : : 4
```

In favor of bean counting you get ...

16Gbps (2 x 8Gbps) lossless recording on On's BigBuff

- quite beefy machine (target = 32Gbps)
- O/S + app need tuning or else not lossless at all

In favor of bean counting you get ...

16Gbps (2 x 8Gbps) lossless recording on On's BigBuff

```
!tstat? 0 : 1131.53s : vbsrecord : UdpsNorRead 16.4638Gbps : F 0.0% ;
```

```
!evlbi? 0 : total : 498732431 : loss : 0 ( 0.00%) : out-of-order : 0 ( 0.00%) : extent : 0seqnr/pkt ;
```


In favor of bean counting you get ...

16Gbps (2 x 8Gbps) lossless recording on On's BigBuff

```
!tstat? 0 : 1131.53s : vbsrecord : UdpsNorRead 16.4638Gbps : F 0.0% ;
```

```
!evlbi? 0 : total : 498732431 : loss : 0 ( 0.00%) : out-of-order : 0 ( 0.00%) : extent : 0seqnr/pkt ;
```

So what's
keeping
the release?!

So what's keeping the release?!

32- vs 64-bit UserDirectory cross-system portability issue

- Had ***nothing*** to do with changes for the new release
 - likely present since release of 64-bit `jive5ab` (Jan 2015)
 - found because of scrutinizing the new release ...
- triggered urgent release of `2.7.1-ack-udfix` (09 Sep 2016)

So what's keeping the release?!

non-bank mode still needs to be verified

- coordination between Ed H., Jon Q. and yours truly
 - ensure Field System and `jive5ab` are on same page
 - ensure Field System procedures work transparently in both bank and non-bank mode
- Sep 13th 4Gbps tests reveals issue:
 - diagnosed (and confirmed): **Conduant changed SDK9.4!!**
 - breaks `protect = off; <command>;`
 - Not sure if safe to ignore if `protect = off;` fails ...

So what's keeping the release?!

scan_check? – the bane of our existence

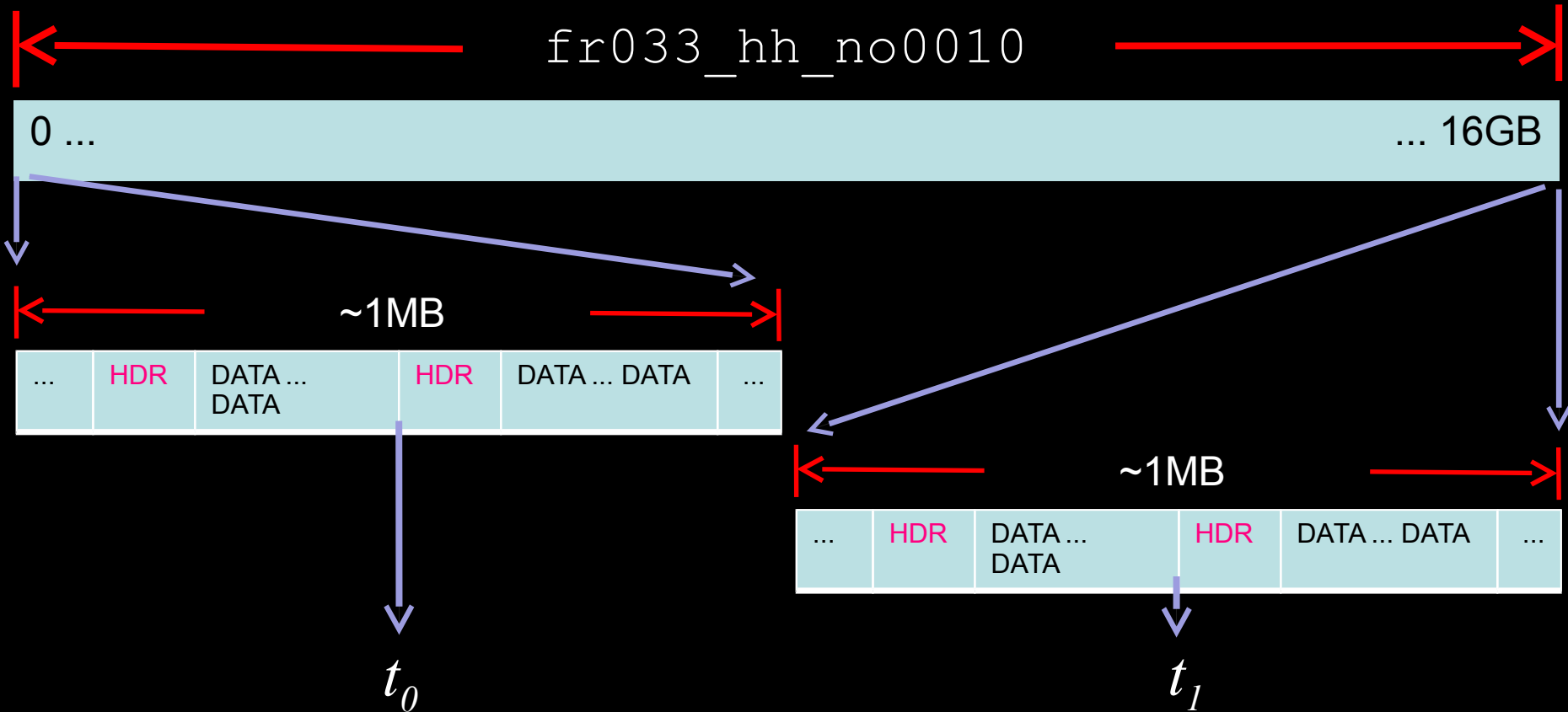
- regression w.r.t. 2.7.1 fails on VDIF
 - intermittent problems:
 - wrong time stamp
 - some scans outrageous scan duration
 - need to verify if limited to VDIF
 - possibly related to internal time format changes (fixable)
 - last Friday found two issues: generic and a VDIF spec.
 - but there's more ...

scan_check?

Purpose in life is simple:

- given any chunk of binary data, without external knowledge,
 - recover the data format
 - recover the recorded data rate
 - if possible recover the number of recorded tracks
 - compute missing bytes
- independent verification recording is what was intended

scan_check?



$$\text{data rate} = \delta t / \delta \text{bytes} = (t_1 - t_0) / \text{recording length}$$

scan_check?

The formats we pretend to know about:

- MarkIV / VLBA
- Mark5B (Mark5B)
- Mark5B (FiLa10G, RDBE)
- VDIF
 - legacy / vdif
 - single thread / multi thread

scan_check?

Two categories,

- MarkIV / VLBA
- Mark5B (Mark5B)
- Mark5B (FiLa10G, RDBE)
- VDIF
 - legacy / vdif
 - single thread / multi thread

scan_check?

Two categories, **the good ones**

- **MarkIV / VLBA**
- **Mark5B (Mark5B)**
- Mark5B (FiLa10G, RDBE)
- VDIF
 - legacy / vdif
 - single thread / multi thread

scan_check?

Two categories, **the good ones** and **the bad ones**

- **MarkIV / VLBA**
- **Mark5B (Mark5B)**
- **Mark5B (FiLa10G, RDBE)**
- **VDIF**
 - **legacy / vdif**
 - **single thread / multi thread**

scan_check?

Two categories, **the good ones** and the bad ones

- **MarkIV / VLBA**

0xFFFFFFFF	YDDDDHHMMSS.sss	DATA DATA
------------	-----------------	------------	------------

- **Mark5B (Mark5B)**

0xABADDEE D	UT second	frame#	ssss	DATA DATA
----------------	-----------	--------	------	------------	------------

scan_check?

Two categories, **the good ones** and the bad ones

- **MarkIV / VLBA**

0xFFFFFFFF	YDDHHMMSS.sss	DATA DATA
------------	---------------	------------	------------

- **Mark5B (Mark5B)**

0xABADDEE	UT second	frame#	sss	DATA DATA
-----------	-----------	--------	-----	------------	------------

They have syncwords – can look for these

scan_check?

Two categories, **the good ones** and the bad ones

- **MarkIV / VLBA**

0xFFFFFFFF	YDDDHHMMSS.sss	DATA DATA
------------	----------------	------------	------------

- **Mark5B (Mark5B)**

0xABADDEE D	UT second	frame#	ssss	DATA DATA
----------------	-----------	--------	------	------------	------------

They have syncwords – can look for these
They contain sub-second (sss) wall clock time

scan_check?

Two categories, the good ones and **the bad ones**

- **Mark5B (FiLa10G, RDBE)**

0xABADDEE D	UT second	frame#	0000	DATA DATA
----------------	-----------	--------	------	------------	------------

- **VDIF (all flavours)**

UT second	frame#	threadID	DATA DATA
-----------	--------	----------	------------	------------

scan_check?

Two categories, the good ones and **the bad ones**

- **Mark5B (FiLa10G, RDBE)**

0xABADDEE D	UT second	frame#	0000	DATA DATA
----------------	-----------	--------	------	------------	------------

- **VDIF (all flavours)**

UT second	frame#	threadID	DATA DATA
-----------	--------	----------	------------	------------

VDIF has no standardized syncword – cannot search

scan_check?

Two categories, the good ones and **the bad ones**

- **Mark5B (FiLa10G, RDBE)**

0xABADDEE D	UT second	frame#	0000	DATA DATA
----------------	-----------	--------	------	------------	------------

- **VDIF (all flavours)**

UT second	frame#	hreadID	DATA DATA
-----------	--------	---------	------------	------------

VDIF has no standardized syncword – cannot search
No relation to wall-clock – must find max frame#!

scan_check?

Two categories, the good ones and **the bad ones**

- **Mark5B (FiLa10G, RDBE)**

0xABADDEE D	UT second	frame#	0000	DATA DATA
----------------	-----------	--------	------	------------	------------

- **VDIF (all flavours)**

UT second	frame#	threadID	DATA DATA
-----------	--------	----------	------------	------------

VDIF has no syncword – cannot search

No relation to wall-clock – must find max frame#!

VDIF degenerate in track bit rate and number of threads

scan_check?

Long story short: scan_check? implementation needs rewrite ...

need scanning algorithm (binary search?)

- recover maximum frame number
 - frame rate x frame size = data rate, frame number = time
- find all thread id's
 - recording pattern may depend heavily on recording application, format, configuration, phase of the moon ...
 - expect heuristics ... to prevent grovelling zillions of bytes
 - 32Gbps \Rightarrow 4GB of data for 1s!

scan_check?

Proposal:

- fix showstoppers now, accept `scan_check?` may not give 100% accurate answers
- start rewrite and postpone to next official release

Thanks
for
your
attention!