ALMA

ATACAMA LARGE MILLIMETER ARRAY

Atacama
Large
Millimeter /
submillimeter
Array

APP Mark6 Recorder Module Test Report

ALMA-05.11.53.03-0001-A-REP

2014-11-21

Prepared by:	Organization Role	Date and Signature
G. Crew	MIT	
Approved by:	Organization Role	Date and Signature
Authorized by:	Organization Role	Date and Signature

Change Record

Version	Date	Affected Section(s)	Author	Reason/Comments
A	2014-11-21	All	G. Crew	First Issue

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Purpose

This document reports on tests performed on recorder modules to meet the requirements of the APP.

1.2 Scope

This document reports on module testing performed at Haystack (PAI) at the end of 2013, and at the OSF (PAI/PAS) in the first quarter of 2014.

1.3 Reference Documents

The following documents contain additional information, are referenced in this document, and should be consulted for further, more detailed information.

Table 1.1: *Reference Documents*

Reference	Document Title	Document ID
[RD1]	APP Update to Corr/Control Design	ALMA-05.11.61.01-0001-A-DSN
[RD2]	APP Mark6 Recorder Test Procedures	ALMA-05.11.50.02-0001-A-PRO

1.4 Acronyms

ALMA Atacama Large Millimeter/submillimeter Array

APP ALMA Phasing Project

MIT Massachusetts Institute of Technology

PIC Phasing Interface Card

VLBI Very Long Baseline Interferometry

Chapter 2

Module Testing at Haystack

These tests were carried out as outlined in [RD2], Section 7.2.1.

2.1 Module Assembly

The modules were assembled at Haystack. Since these are the first modules with higher capacity disks, it was desirable to test disks from at least two vendors.

As a side effect of the module tests, this was an opportunity to test the reliability of the Mark6 in a number of operational scenarios. A developmental version of the Mark6 software was in use, so some issues with the command set were uncovered and fixed (in later releases).

Modules were assembled in two flavors; 24TB modules built of 3TB Seagate disks, and 32TB modules built of 4TB Western Digial (WD) disks. These are "standard" 3.5 inch hard drives (with the form factor used by PC computers and servers for several decades).

The full module designation is of the form `LABEL/CAPACITY/SPEED`. In our case, these modules are all rated at 4 Gbps, and the modules were built of like disks, so the capacities were 24000 and 32000. The modules were numbered serially in the `MHO` series, and for completeness, here is the set:

```
MHO%0001/24000/4 ok - full test    - Mark6-4007
MHO%0002/24000/4 ok - full test    - Mark6-4007
MHO%0003/24000/4 ok - repaired     - shelved (Mark6-4006)
MHO%0004/24000/4 ok - full test    - shelved (Mark6-4006)
MHO%0005/32000/4 ok - full test    - Mark6-4007
MHO%0006/32000/4 ok - full test    - shelved (Mark6-4006)
MHO%0007/32000/4 ok - full test    - shelved (Mark6-4006)
MHO%0008/32000/4 ok - full test    - Mark6-4007
MHO%0009/24000/4 ok - full test    - Mark6-4005
MHO%0010/24000/4 ok - full test    - Mark6-4005
MHO%0011/24000/4 ok - full test    - Mark6-4008
MHO%0012/24000/4 ok - full test    - Mark6-4008
MHO%0013/32000/4 ok - full test    - Mark6-4005
MHO%0014/32000/4 ok - full test    - Mark6-4005
MHO%0015/32000/4 ok - full test    - Mark6-4008
MHO%0016/32000/4 ok - full test    - Mark6-4008
MHO%0017/24000/4 ok - partial test - Mark6-4004
MHO%0018/24000/4 ok - partial test - Mark6-4004
MHO%0019/24000/4 ok - partial test - Mark6-4004
MHO%0020/24000/4 ok - partial test - Mark6-4004
```

`MHO%0003` had a bad backplane in the module chassis. This exhibited symptoms that were consistent with an intermittently bad disk or cable or controller card, and it was only after a fair amount of part swapping that the problem was finally identified, the part replaced, and a fully vetted module produced. Most of the tests were conducted with a substitute module (exchanged from another project) loaded with the disks originaly assigned to `MHO%0003`.

(The errant module (backplane) was replaced for use with the other project.)

Modules `MHO%0017` through `MHO%0020` are in Charlottesville supporting PIC/ASUMMER development and testing. Cursory tests were performed to establish that they modules worked fine, but they were not subjected to extensive testing.

The last column in the table above indicates where the modules where when the recorders were packed up.

2.2 Setup

These tests were executed between Nov 22, 2013 and Jan 7, 2014.

Recorders 1 and 4 (Mark6–4005 and Mark6–4008) were set up in the peer configuration. Recorder 3 (Mark6–4007) was set up in the loopback mode. Recorder 2 (Mark6–4006) and Recorder 0 (Mark6–4004) were not set up at this time. Modules were grouped in pairs (slots 12 or slots 34) for the 8 Gbps input streams.

```
# The logs were stored on the Haystack ALMA server Monarth
# in a directory hierarchy mounted at: data=/alma/data/Mark6
# Mark6-4007 (loopback) - $data/4007/test-4007/out-MHO%0001,2,8,5-57000*
>> mstat?
<<  !mstat?0:0:
12:1:MHO%0001/24000/4/8:8:8:-:24000:closed:unprotected:sg:
12:2:MHO%0002/24000/4/8:8:8:-:24000:closed:unprotected:sg:
34:3:MHO%0008/32000/4/8:8:8:-:32000:closed:unprotected:sg:
34:4:MHO%0005/32000/4/8:8:8:-:32000:closed:unprotected:sg;
# Mark6-4005 (peerpush) - $data/4005/test-4005/out-MHO%0009,10,13,14-57000*
>> mstat?
<<  !mstat?0:0:
12:1:MHO%0009/24000/4/8:8:8:2077:24000:closed:unprotected:sg:
12:2:MHO%0010/24000/4/8:8:8:2078:24000:closed:unprotected:sg:
34:3:MHO%0013/32000/4/8:8:8:2922:32000:closed:unprotected:sg:
34:4:MHO%0014/32000/4/8:8:8:2955:32000:closed:unprotected:sg;
# Mark6-4008 (peerpush) - $data/4008/test-4008/out-MHO%0011,12,15,16-57000*
>> mstat?
<<  !mstat?0:0:
12:1:MHO%0011/24000/4/8:8:8:5231:24000:closed:unprotected:sg:
12:2:MHO%0012/24000/4/8:8:8:5231:24000:closed:unprotected:sg:
34:3:MHO%0015/32000/4/8:8:8:2701:32000:closed:unprotected:sg:
34:4:MHO%0016/32000/4/8:8:8:3391:32000:closed:unprotected:sg;
```

Because of the mechanics of the `peerpush.sh` script, the output is synchronized between the two machines, and thus the two output directories refer to writes to both machines. (*I.e.* the logs are duplicated in the two output directories.)

The goal of these tests was to shake out the system in anticipation of longer duration and more thorough testing *in situ* at ALMA.

The "doit" script was configured for the cabling arrangement so that could be run with an invocation specifying (output) scheduling options, labelling and target modules, *e.g.* :

```
./doit-400?.sh long N01 34
```

2.3 Results

Each invocation of `loopback.sh` or `peerpush.sh` generates a `.sched` file with the schedule, `.scans` file with the scans executed (labelled PASS or FAIL according to whether the approximate number of bytes/files on disk is correct), and `.marks` with the `mark_check` output and `.chk` with the scan_check output. These latter files contain details which are intended for visual examination. There is one line of scan_check output which may be tested for pass/fail. In both cases the examination is by scatter-gather fragment (16 fragments per VLBI scan recorded.)

Note that peerpush.sh was developed first and turned up a number of issues with the early version of the Mark6 application. These isses were fixed before the later loopback.sh script was developed. Likewise, the mark_check application preceded the development of the scan_check test. The former is more thorough, but more time consuming to execute, so the goal was to validate that scan_check gave consistent results with mark_check.

2.3.1 peerpush

Because of the mechanics of the script, the results are duplicated on both machines. To avoid double-counting (and misrepresentation of the results) we report only on the set of logs output to Mark6–4005.

There were 50 peerpush.sh schedules executed on Mark6–4005, comprising a total of 1229 simulated VLBI scans with durations ranging from 20 to 1794 seconds.

There were 16860 and 56 scan fragments that were PASS or FAIL, respectively. The failures were all related to bugs in the application or operator error.

The mark_check methodology was invoked for 26942 scan fragments; of these 23672 were Pass and 3270 were Fail. The failures were dominated by bugs in the application or the testing tools.

The scan_check methodology was invoked for 10408 scan fragments; all were passed, none failed.

2.3.2 loopback

There were 18 loopback.sh schedules executed on Mark6–4007, comprising a total of 589 simulated VLBI scans with durations ranging from 20 to 1791 seconds.

There were 7984 and 20 scan fragments that were PASS or FAIL, respectively. The failures were all related to bugs in the application or operator error.

The mark_check methodology was invoked for 3552 fragments; all were passed, none failed.

The scan_check methodology was invoked for 4576 fragments; all were passed, none failed.

2.4 Recorder Testing Summary

This initial testing was carried out to vet both the 8 modules and 2 recorders to be shipped to Chile (PAI).

No hardware issues were found with the hardware shipped to Chile.

Minor software issues were found, but as the plan is to upgrade the software in Chile, this is not an issue. The third recorder, Mark6–4007, was retained at Haystack to support ongoing development.

The other two recorders (Mark6–4004 and Mark6–4006) were subsequently unpacked and tested (using peerpush.sh) at Haystack, paired with Mark6–4007. A record of those tests was not retained: Mark6–4004 was delivered to Charlottesville to support PIC testing, and Mark6–4006 was paired with Mark6–4007 for testing in development prior to shipment to Chile and installion in the OSF computer room.

Chapter 3

Module Testing at OSF

3.1 Setup

These tests were executed between January 25 and March 5, 2014.

Recorders 1 and 4 (Mark6–4005 and Mark6–4008) were racked in a test rack in the Correlator lab at the OSF. This is a harsher environment than the final destination computer room (probably dirtier and warmer). The eight modules delivered with these recorders were subjected to repeated tests to verify that they could perform in the target environment. Additionally, with the module types (24TB and 32TB) paired up, we could potentially spot differences in performance according to disk type. (Unsurprisingly, no differences were noted, other than recording duration.)

For this test the recorders were paired, so `peerpush.sh` was used exclusively, and executed precisely as described in [RD2], Section 7.2.1. A pair of modules of each type was placed on each recorder so that a cycle of 4 executions could completely fill up all 8 modules:

```
# test the 24TB modules:
./doit-4005.sh 48TB A45 12
./doit-4008.sh 48TB A46 12
# wait & review
# test the 32TB modules:
./doit-4005.sh 64TB A47 34
./doit-4008.sh 64TB A48 34
# wait & review
```

taking a few days (13.3 hours per 24TB module pair and 17.8 hours per 32TB module pair).

3.2 Results

Because of the mechanics of the script, the results are duplicated on both machines. To avoid double-counting (and misrepresentation of the results) we report only on the set of logs output to Mark6–4008.

A total of 48 `peerpush.sh` schedules were executed, comprising a total of 2514 simulated VLBI scans with durations randomly generated (uniformly) between 20 and 1800 seconds.

There were 39680 and 32 scan fragments that were PASS or FAIL, respectively. The failures (2 VLBI scans with 16 fragments each) occurred at the end of one session where slightly more recorded data was requested than could be recorded on the disks (operator error of the greedy kind).

The `mark_check` methodology was invoked for 2210 scan fragments; of these 2210 were Pass and 0 were Fail. (By the time of this test, it had been established that the longer `mark_check` method was unnecessary, but it was invoked in 3 sessions at the end of the series (A45, A46 and A47) as a double-check.

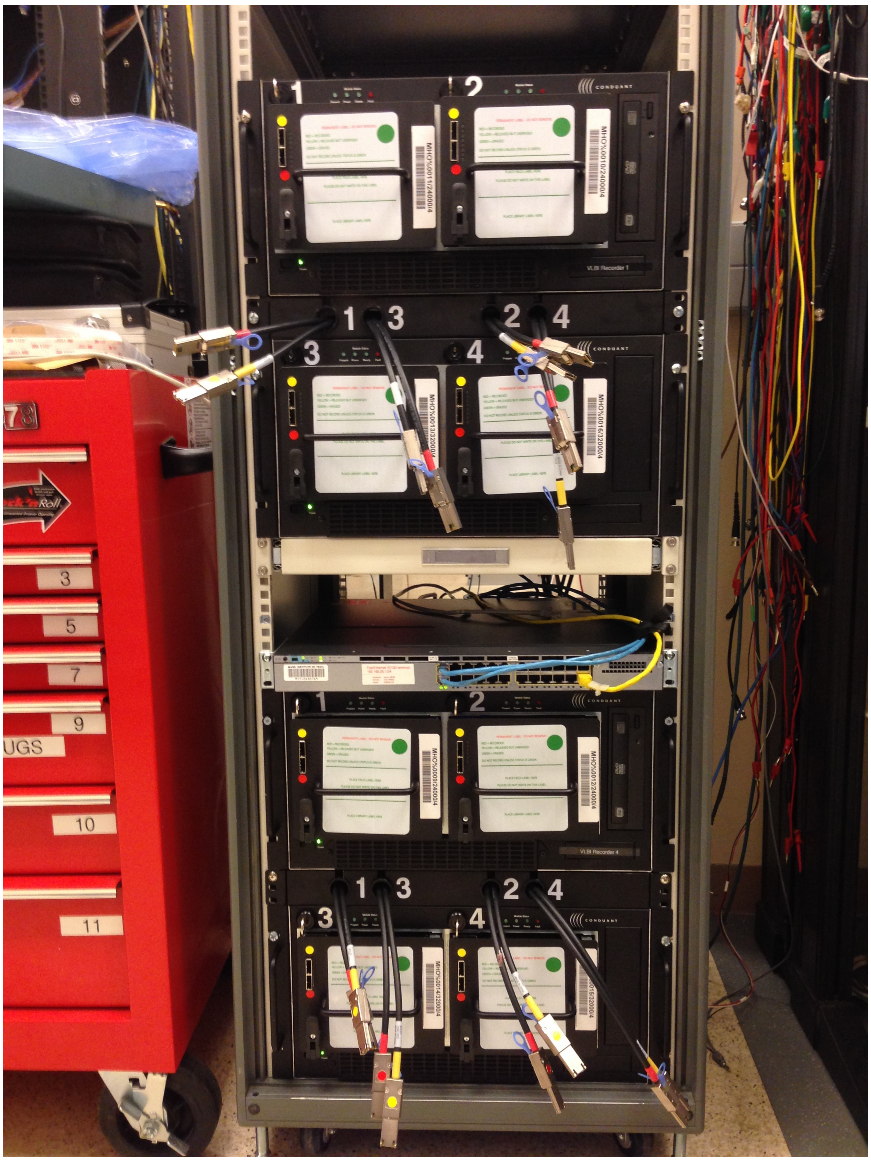

Figure 3.1: Recorder 1 and Recorder 4 housed in the test rack in the correlator lab. Note that each recorder consists of a host chassis (with slots 1 and 2) and an expansion chassis (with slots 3 and 4).

The `scan_check` methodology was invoked for 37681 scan fragments; there were precisely 3 failed cases. These were investigated and found to be “edge” conditions in the `scan_check` logic. (*I.e.* rare cases in the way the scatter-gather files are generated which were not coded.)

The number of checks is slightly less then number of sessions as there were two power-failures which intervened during the checking process, and completeness of the sample isn’t truly necessary here.

3.3 Module Duration Summary

A grand total of 707.5 recording hours (4.2 weeks) on these modules was carried out:

```
$cat $data/4005/out-clab-4005/*sched |\
 cut -d: -f2 | awk ’{s+=$1}END{print s,s/3600.0}’
2546831 707.453
$cat $data/4008/out-clab-4008/*sched |\
 cut -d: -f2 | awk ’{s+=$1}END{print s,s/3600.0}’
2547418 707.616
```

No issues were found with the disks or modules.