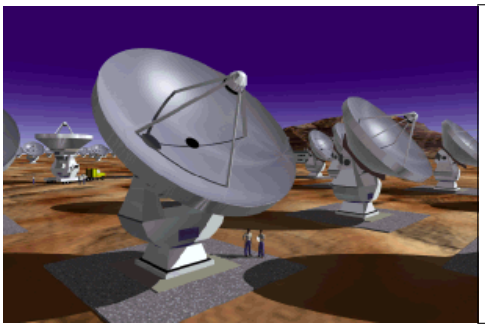

Atacama Large Millimeter Array	ALMA-SW-OL-TP-0NNN Revision:0.3 2003-05-08 *Software Tests* Raymond Rusk

Plan for AIPS++

Performance Testing

Software Tests

R. Rusk, K. Golap

Keywords: Software Test, Offline

Author Signature: Raymond Rusk	Date: 2003-05-08
Approved by: B. Glendenning, G. Raffi, K.-I. Morita	Signature:
Institute: NRAO, ESO, NAOJ	Date:
Released by:	Signature:
Institute:	Date:

Change Record

Revision	Date	Author	Section/ Page affected	Remarks
0.1	2003-04-23	Raymond Rusk	All	Initial Version
0.2	2003-04-24	Raymond Rusk	Test deliverables	Memory usage/disk IO added
0.3	2003-05-08	Raymond Rusk	Sections 2,6,9,11	Role of SSR group clarified

Contents

1 Introduction

The content and structure of this plan conform to industry standards *IEEE Std 829-1998 IEEE Standard for Software Test Documentation*, *IEEE/EIA 12207.0/12207.1 Standard for Information Technology - Software life cycle processes* and *ALMA-SW-008 The ALMA Software Documentation Standard*.

1.1 Objectives

The performance test plan for the offline data processing system supports the following objectives:

1. To detail the activities required to prepare for and conduct the benchmarks;
2. To communicate to all responsible parties the tasks that they are to perform and the schedule to be followed in performing the tasks;
3. To define the sources of information used to prepare the plan;
4. To define the test tools and environment needed to conduct the benchmarks.

1.2 Background

AIPS++ has been chosen as the foundation on which to build the ALMA Offline Data Reduction Package. A recent audit of the AIPS++ Package for ALMA Offline data processing reported in ALMA-SW Memo 19 did not address performance issues. However, the authors of that audit noted that "performance issues (which are perceived to be a possible problem with AIPS++) will likely be a driver in the next auditing cycle once these metrics are available".

Some benchmarking of AIPS++ on IRAM millimeter-wave data was done for the March 2003 Preliminary Design Review in Tuscon. That work, reported in the Memo *IRAM AIPS++ Test: Phase III*, indicates that the offline performance requirements are achievable with workstations that are likely to be available when ALMA starts producing data but that continuing and more systematic benchmarking efforts are required.

This plan is intended to provide performance measurements that will support further audits of AIPS++. It builds on previous test plan notes published by F. Viallefond and Debra Shepherd, addressing system specific issues required to perform the benchmarking tasks in an efficient and timely fashion.

1.3 Scope

This document provides a plan for the testing of AIPS++ performance on datasets reflecting typical ALMA observing patterns reduced using procedures relevant to ALMA data processing requirements. The plan describes the software test environment to be used for the testing, identifies the tests to be performed, and provides schedules for test activities.

Specifically, it is a plan to test whether AIPS++ meets the performance requirements listed in ALMA-SW-018 *ALMA Offline Data Processing Requirements* by S.T. Myers *et al.* The requirements subject to validation are:

OL-1.1-R4 The performance of the Package shall be quantifiable and commensurate with the data processing requirements of ALMA output and the scientific needs of users at a given time. The timing and reproducibility of results for a fiducial set of reduction tasks on specified test data will be benchmarked (e.g. "AIPSmarks") and compared against other packages and **a list of benchmark specifications provided and maintained by the Project**.

OL-2.2-R1 The GUI shall provide real-time feedback via standard compact displays:

OL-2.2-R1.1 Window updating must be fast (less than 0.1s on same host).

OL-2.4-R1 The UI must have basic programming facilities such as:

OL-2.4-R1.9 efficient special vector and matrix operations

OL-3.2-R3 The Package must be able to handle, efficiently and gracefully, datasets larger than main memory of the host system.

Section 2.5.1 General Imaging Requirements *Because ALMA is inherently a multi-channel instrument, spectral cube mapping shall be built in as the primary mode from the beginning. Also, due to the high volume of data that can be produced by ALMA, it is imperative that the imaging and deconvolution tools in the Package be user-friendly, efficient, and flexible. This is the workhorse of the Package as far as most users will be concerned, and suitability and success of the Package will be judged with this in mind.*

OL-7.2-R3 The plot update speed shall not be a bottleneck. Speed shall be benchmarked, and should be commensurate with comparable plotting packages.

1.4 References

In accordance with the ALMA Software Documentation Standard ALMA-SW-008, all references are provided at the end of this document.

2 Test Items

The latest (at time of test) stable release of AIPS++ will be tested. The specific list of test items and features to be tested will be defined by the Science Software Requirements (SSR) group in collaboration with the benchmark team. In general terms, the software components tested will include the fillers for raw data, utilities for FITS import and export of calibrated data, the data calibration routines, and the tools to generate "dirty" and "clean" images along with analysis features which are critical to producing scientifically valid results.

3 Features to be tested

The performance of the current stable version of AIPS++ compared to previous stable versions and its performance compared to other data reduction package will be tested.

The volume of data processed during benchmarking will correspond to the average and peak data rates expected from ALMA. Where possible performance will be determined as a function of the size (duration of observation and number of spectral channels, baselines and polarizations) of the input dataset being processed.

The Science Software Requirements committee will specify the specific features to be tested but in general terms the performance of the data filler, the calibrater, the imager and the GUI will be measured and compared to other reduction packages where appropriate.

4 Features not to be tested

This plan addresses performance concerns regarding the offline data reduction package. Features of AIPS++ unique to its future role in the science pipeline will not be tested though it is likely that most pipeline features may be exercised given the common code base.

Also, this test plan focuses on performance benchmarking. Correctness of procedures is not the primary goal. Correctness validation requires additional testing that is already being done in the AIPS++ project. However, the performance tests will incorporate automated "'sanity" checks wherever possible to ensure the correct functioning of the software under test.

5 Approach

The focus of the performance test plan approach is on automation and efficient use of test personnel time. Automation is crucial since the test suite will have to be re-run after each new AIPS++ stable release. Also, part or all of the tests will need to be re-run when a comparison package is updated though it is expected that this will occur much less frequently.

The Science Software Requirements group is responsible for contributing the list of items and features to be tested. It will be much easier for the test team if these contributions follow the guidelines outlined below:

1. Each test case should be supported by a rationale for inclusion. This must include a statement of which feature or features are being benchmarked. One or more specific offline software requirements must be referenced.
2. Each test item shall have complete input specifications. These must be provided for each package (AIPS++, AIPS, Gildas, Miriad, Gipsy, etc.) being tested. This includes an appropriate data set in a format suitable for processing and field values for all parameters that must be specified to meet the test objective. An output specification must also be provided that will allow for an automated check of whether the procedure being benchmarked ran correctly. In some cases this could be a final image, for instance, with a metric that will allow the derived image to be compared to the expected image for correctness. (If one or more comparison packages require data in a different format than AIPS++, the Science Software Requirements Group must ensure that all input datasets are functionally equivalent. Also, they must ensure that the input parameters to all comparison packages are functionally equivalent to those used in AIPS++.)
3. Each test case must be supported by an executable script for AIPS++ and any other package to which AIPS++ must be compared. If the "comparable" package does not provide its own scripting language, then detailed instructions that the test team can use to generate a reduction script in some other UNIX scripting language must be provided. It is important that test scripts restore the testing environment upon completion. This includes removal of temporary files, for instance.
4. Any special environmental or procedural needs must be indicated. This includes an estimate of the amount of disk space required if it is substantial (greater than 20GB, for instance). If special procedures are required to provide the input data in a format not used by AIPS++ or a package under comparison then these procedures must be described (and scripted if possible).

The results of testing will be posted automatically onto the web after each test run. These results will include hyper-linked test logs, incident report logs, incident reports and test summaries. Each test log will provide a chronological record of relevant details about the execution of the test. The test incident report documents any event that occurs during the testing process that requires investigation. The test incident report log records the results of that investigation. The summary report provides the results of the designated testing activities in a format suitable for supporting audits of the AIPS++ package.

Each test result will include a complete description of the test environment including software version and release number, hardware platform characteristics (processor(s), processor speed, RAM, cache, number and model of disk drives, disk drive configuration, and filesystem (eg., ReiserFS, ext3)), the operating system and compilers used to build the package. Any change in the execution environment over time will be clearly indicated. This is particularly important for regression testing.

6 Item pass/fail criteria

The test plan will report performance measurements in support of future AIPS++ audits. The AIPS++ package is still in a state of rapid development so benchmark results will provide feedback for the developers.

The results of performance testing will show whether incremental performance improvements are being achieved. It is anticipated that a future audit of AIPS++ will establish final pass/fail criteria and that the benchmark results will be one factor in the final decision.

7 Suspension criteria and resumption requirements

End-to-end testing can take many hours. Unanticipated disruptions such as power failures may occur. It is useful to establish check-points in long tests so that they can be resumed. For instance checkpoints at

the end of data filling and after calibration can be used. The Science Software Requirements group should indicate other possible breakpoints when proposing specific tests, if the test is expected to take more than two hours of wall clock time. Any special processing required to resume processing from the check-point such as clearing of flags, re-initialization of fields, etc., must be clearly indicated.

8 Test deliverables

The following documents will be generated by the performance test group and will be published on the web (URL to be determined) after each run of the test suite.

1. **Test documentation:**
   (a) Performance Test Plan (this document)
   (b) Performance Test Specifications (See the section entitled "Approach" for details. Each test case will be supported by a rationale for inclusion, inputs and expected output with a metric for correctness testing, and executable scripts)
   (c) Performance Test Reports (hyper-linked test logs, incident report logs, incident reports and test summaries will be provided on the web)
2. **Test data:**
   (a) All input parameters for the test must be clearly documented. This can be done within the test script.
   (b) The input dataset and "expected output" test files are to be provided by the SSR. The "expected output" test file might consist of an image, for instance. A metric for comparison of actual to expected test output shall be provided. Ideally, the test script will make this comparison automatically.
   (c) The output logs and reports will be published to the web. Both elapsed CPU and wall-clock time as well as memory usage and disk IO due to paging and swapping will be measured and reported in all cases. The generated images, calibrated datasets, plots, temporary files, etc. will be removed after correctness of the results has been established. The removal of generated files except those required for test documentation shall be part of the test script.

9 Testing tasks

See the task list in Table 1.

Task (2) *test design specification* is the high-level process of specifying the approach for measuring the performance of a software feature or combination of software features and identifying the associated tests. Task (3) *test case specification* includes specifying inputs, expected output(s) and any special requirements (hardware, software, intercase dependencies) for a test item.

Task (4) *test procedure specification* is a specific sequence of actions required for the execution of a test. When comparison to another reduction package is unnecessary or impossible then tasks (4) and (6) could be combined. However, in general task (4) produces a procedural description that is data reduction package independent and task (6) implements that procedure for different comparison packages.

Task (12) is the automated web-publishing of benchmark results. These will not be pass/fail results. Instead they will be measurements allowing the incremental performance changes within the AIPS++ package to be tracked over time. The Science Software Requirements group will be able to use these measurements to assess whether performance improvements within AIPS++ are meeting ALMA requirements in a timely manner.

Task (13) involves the periodic reevaluation of test specifications and data sets to determine if specifications should be expanded and data sets added or changed.

Table 1: Task list

Task	Dependency	Special skills	Responsibility	Effort	Finish
(1) Prepare test plan.	-	-	Manager,Analyst	1w	30Apr03
(2) Prepare test design specifications.	1	Science	Scientist	4w	30May03
(3) Prepare test case specifications.	2	Science	Scientist	4w	27Jun03
(4) Prepare test procedure specifications.	3	Science	Scientist	4w	25Jul03
(5) Prepare test datasets - send to test group.	3	Science	Scientist	2k	22Aug03
(6) Prepare test scripts - send to test group.	4	Science/Comp	Scientist,Analyst	4w	22Aug03
(7) Build AIPS++/AIPS/Gildas/Miriad etc.	4	Comp	Analyst,Tech	3w	22Aug03
(8) Build benchmark web page.	4	Comp	Analyst,Tech	2w	22Aug03
(9) Run tests.	5,6,7	Comp	Tech	1w	29Aug03
(10) Check out results.	9	Comp	Analyst,Tech	2d	01Sep03
(11) Resolve test incident reports.	10	Comp/Science	Scientist,Analyst	1w	08Sep03
(12) Publish benchmarks.	8,9	Comp	Tech	1d	30Aug03
(13) Develop new performance tests.	12	Science	Scientist	-	-
(14) Add new tests to benchmark suite.	13	Comp	Tech	1w	-
(15) Periodically rerun benchmarks.	12	Comp	Tech	1w	-

10 Environmental needs

1. **Hardware.**

   The testing will be done on dedicated workstations. For uniprocessor systems this means stopping the cron daemon so that the benchmark is not impacted by regularly scheduled tasks. For instance, on Red Hat Linux systems the updating of the "locate" database is run once a day typically. This process can take several minutes and is very disk intensive. It would have a major impact on any other disk activity occurring during that time.

   Multi-processor workstations may be used provided that all data reduction tasks are confined to one processor and are performed sequentially. Normal system housekeeping processing can utilize the second processor obviating the need to stop the cron daemon. However, no competing high memory usage or high-intensity disk activity processes can run while benchmarking is being done.

   All comparison benchmarks between AIPS++ and other packages must be performed on the same workstation. The test summary report must contain complete details of the test environment. This includes the processor speed, RAM and cache size and the disk and filesystem performance characteristics. On Linux systems, this information can be captured from /var/log/dmesg, /etc/fstab, procinfo, df and top, for instance.

2. **Software**

   (a) **Operating system.** All testing will be done in a Linux environment. The distribution, version and release number shall be specified. It is expected that Red Hat Linux will be used for all testing.

   (b) **Compilers.** The compiler name, release version, options and optimization level must be specified for each benchmark result.

3. **Web Site**

   A website for the automated publication of performance test results must be available. If it is secured by password, access must be provided for the testers, AIPS++ auditors and other designated parties.

11 Responsibilities

The following groups have responsibility for segments of the testing:

1. **ALMA AIPS++ test group** This group provides the overall management of the performance testing and benchmarking expertise.
2. **SSR** This group represents the end user of the offline system and will be responsible for the following activities:
   - working with AIPS++ test group to create test cases, test datasets and test procedures (in the form of scripts)
   - GUI performance testing
3. **AIPS++ auditors** This group is responsible for providing the final pass/fail criteria which will be applied to the benchmarks.

12 Staffing and training needs

12.1 Staffing.

A continuing 0.5 FTE is required for performance testing. It will be split among the following roles:

1. **Test group.**
   - 1 Test manager
   - 1 Test analyst
   - 1 Test technician
2. **SSR group.**
   - 1 Offline subsystem scientist.

12.2 Training.

Once the tests are automated the major activities can be transferred to the test technician who will provide routine maintenance. Some training will be required before the technician can assume this role. One month to learn the procedures for installing and updating AIPS++ and the various other comparison packages (AIPS, Gildas, Miriad, Gipsy, etc.) and for learning how to maintain the test suite and reports published to the web site should be sufficient.

13 Schedule

See the task list in Table 1. Hardware, software and test tools will be required for testing during a period of several years.

14 Risks and contingencies

Performance testing heavily impacts workstation availability for other purposes. A dedicated system has been acquired at DRAO in Penticton to alleviate this problem. A dedicated benchmarking machine with similar characteristics is available in Socorro.

Comparison packages may be updated from time to time. New versions will have to be installed on the benchmarking machine. Some time has been allocated for training to perform these tasks. The Offline Subsystem Scientist will indicate when an updated comparison package is available and when it should be installed.

Since the performance testing process will extend over many months or years personnel changes may occur. This risk will be alleviated by thoroughly documenting the test procedures, datasets and results. In any case, such transparency is required for the adequacy of the performance testing to be assessed easily.

Assuming that AIPS++ remains the offline package of choice, benchmarking may extend into years and more than one generation of dedicated workstations. This complicates interpretation of performance results. Before replacement, the next generation and current generation of workstation can both run benchmarks so that the results can be inter-compared.

Automated testing cannot exercise the GUI adequately. Also, the final GUI is not available. However, the Offline Subsystem Scientist has primary responsibility for running tests that ensure representative users can successfully achieve their intended tasks using the software product. GUI testing will form a natural part of that testing effort; this plan will focus on tests that can be automated.

An important concern is that comparison of the results from AIPS++ to other packages may in some instances be equivalent to comparing "apples and oranges". This risk can be mitigated by having the subsystem scientist verify that the initial data sets used to benchmark the functionality of comparison packages are equivalent and that the functionalities being tested are equivalent. If the functionalities of packages are not entirely equivalent then these differences must be indicated. Analysis of any remaining large discrepancies in benchmark results will be addressed through examination of the source code from comparison packages to ensure that it provides functionality that is logically equivalent to the corresponding code in AIPS++.

Another risk is that the benchmarking efforts may not capture all the information required for future audits. This will be alleviated through consultation with the Science Software Requirements Group regarding reporting standards and by the addition of new performance tests as the need arises.

15 Approvals

See the title page.

16 References

1. IEEE Std 829-1998, IEEE Standard for Software Test Documentation
2. IEEE/EIA 12207.0-1996 (ISO/IEC 12207) Standard for Information Technology - Software life cycle processes
3. IEEE/EIA 12207.1-1997 (ISO/IEC 12207) Standard for Information Technology - Software life cycle processes - Life cycle data
4. ALMA-SW-008, revision 4, 2001-02-21, The ALMA Software Documentation Standard, Alan Bridger
5. ALMA-SW-019, revision 1.3, 2003-01-20, Audit of the AIPS++ Package for ALMA Offline Data Processing, S. Meyers, F.Viallefond, K.-I. Morita
6. ALMA-SW-0NNN, revision 0.1, 2003-03-13, IRAM Aips++ Test: Phase III, R. Lucas, D. Broguiére, K. Golap, R. Rusk
7. 2002-12-23, Benchmark of the AIPS++ Package, F. Viallefond
8. alma-sw-ssr, draft, 2003-04-09, Offline testing plan, Debra Shepherd
9. ALMA-SW-018, revision 4.1, 2002-04-18, ALMA Offline Data Processing Requirements, S. Myers *et al*