PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

An overview of the planned CCAT software system

Tim Jenness, Martin C. Shepherd, Reinhold Schaaf, Jack Sayers, Volker Ossenkopf, et al.

Tim Jenness, Martin C. Shepherd, Reinhold Schaaf, Jack Sayers, Volker Ossenkopf, Thomas Nikola, Gaelen Marsden, Ronan Higgins, Kevin Edwards, Adam Brazier, "An overview of the planned CCAT software system," Proc. SPIE 9152, Software and Cyberinfrastructure for Astronomy III, 91522W (18 July 2014); doi: 10.1117/12.2056516

Event: SPIE Astronomical Telescopes + Instrumentation, 2014, Montréal, Quebec, Canada

An overview of the planned CCAT software system

Tim Jenness[a], Martin C. Shepherd[b], Reinhold Schaaf[c], Jack Sayers[b], Volker Ossenkopf[d], Thomas Nikola[a], Gaelen Marsden[e], Ronan Higgins[d], Kevin Edwards[f], Adam Brazier[a]

[a]Department of Astronomy, Cornell University, Ithaca, NY, 14853, USA;

[b]California Institute of Technology, 1200 E California Blvd, Pasadena, CA 91125, USA;

[c]Argelander-Institut für Astronomie, Universität Bonn, Auf dem Hügel 71, 53121 Bonn, Germany;

[d]KOSMA, I. Physikalisches Institut, Universität zu Köln, Zülpicher Str. 77, 50937 Köln, Germany;

[e]Department of Physics and Astronomy, University of British Columbia, 6224 Agricultural Road, Vancouver, BC V6T 1Z1, Canada;

[f]Department of Physics, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ABSTRACT

CCAT will be a 25 m diameter sub-millimeter telescope capable of operating in the 0.2 to 2.1 mm wavelength range. It will be located at an altitude of 5600 m on Cerro Chajnantor in northern Chile near the ALMA site. The anticipated first generation instruments include large format (60,000) kinetic inductance detector (KID) cameras, a large format heterodyne array and a direct detection multi-object spectrometer. The paper describes the architecture of the CCAT software and the development strategy.

Keywords: Astronomy software, Facilities: CCAT, Software Development, Observatories, Submillimeter: general

1. INTRODUCTION

CCAT[1–5] is a 25 m diameter sub-millimeter telescope to be built at an altitude of 5600 m on Cerro Chajnantor in Chile near the ALMA site. Operating at this altitude results in excellent transparency across all observing bands from 350 µm to 2 mm, including the potential to observe at 200 um in the best weather conditions.[5,6]

The CCAT project has identified four first generation instruments to achieve its science goals. SWCam[7–9] will have of order 60,000 detectors operating mainly at 350 µm with additional detectors at 450, 850 and 2000 µm. CHAI[10] will be a large format heterodyne array operating simultaneously in two bands (two of 850, 600 and 350 µm) with at least 32 elements per band, with the backend able to process spectra with a bandwidth of at least 4 GHz and 64k channels. LWCam[11] will be a dedicated long-wave camera operating in 5-6 bands between 750 µm and 2.1 mm with a long-wavelength goal of 3.3 mm. X-Spec[12,13] will be a multi-object spectrometer with ~ 100 beams on the sky, each covering a frequency range of 190-520 GHz in two bands simultaneously with a resolving power of $400 - 700$.

CCAT software development covers all phases of Observatory operations (see Figure 1), including observation preparation, dynamic scheduling, observation execution, data management and data reduction. In this paper we present an overview on the software design and a report on current status. The main drivers for the software design are:

1. Data rates that exceed a petabyte per year for the first generation instruments.

Further author information: (Send correspondence to T.J.)
T.J.: E-mail: tjenness@cornell.edu

Software and Cyberinfrastructure for Astronomy III, edited by Gianluca Chiozzi, Nicole M. Radziwill,
Proc. of SPIE Vol. 9152, 91522W · © 2014 SPIE · CCC code: 0277-786X/14/$18
doi: 10.1117/12.2056516

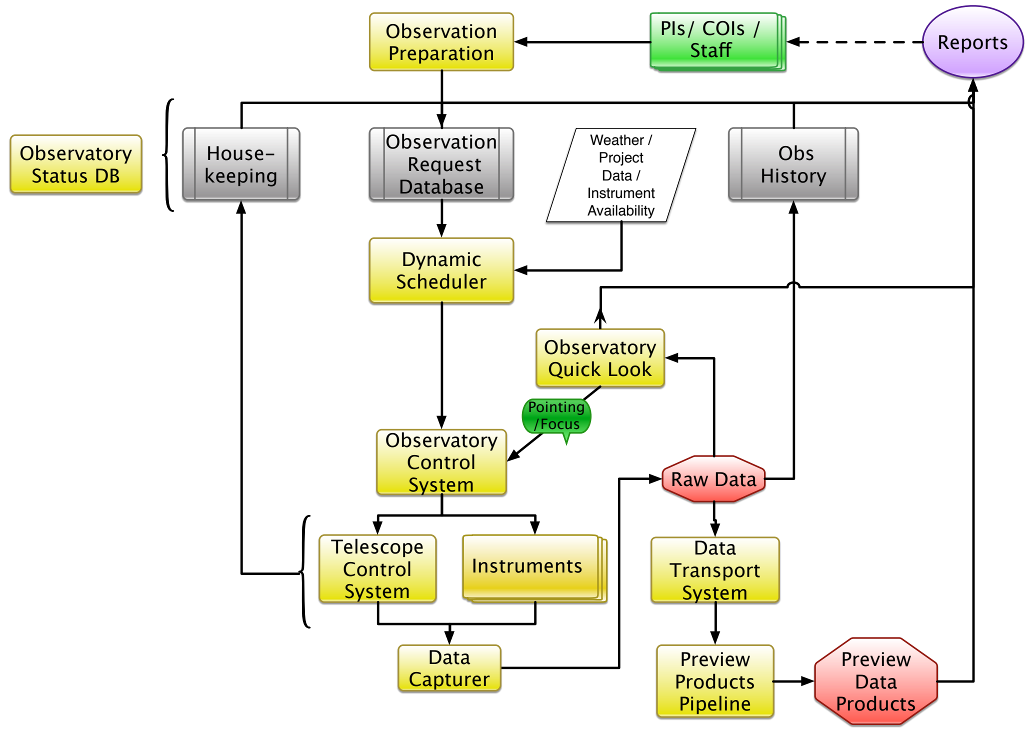

Figure 1. Overview of CCAT software systems. Observations are prepared and stored in a database to be queried by the scheduler. The Observatory Control System executes observations and data flows from the instruments and telescope to the Data Capturer, which collates the output and makes the data files available to the Observatory Quick Look pipeline and data transport system. Preview data products are generated and made available to astronomers as part of reports on the status of their projects.

2. Remote operations of a telescope at an inhospitable site, with remote monitoring of observatory status by observers and engineers.

The data rate for SWCam is expected to be of order 1 Gbps, assuming some lossless compression is employed resulting in a variable bit depth[9,14]*. The CHAI data rate for the baseline array design with 10 Hz readout for on-the-fly mapping is approximately 1.4 Gbps. The estimated annual rate for CCAT, taking weather statistics into account,[6] is of order 1.5 petabytes.

2. OBSERVATORY CONTROL SYSTEM

The design of the Observatory Control System (OCS) for the CCAT telescope is based on an approach used at a number of current and past telescopes, such as the CBI,[15] QUIET[16] and OVRO 40m[17] telescopes. Figure 2 shows the two major parts of the OCS and how they interact with user interfaces, the dynamic scheduler, the telescope control system (TCS), and various instruments.

The OCS will be a minimal system that has no knowledge of the TCS, the science instruments or any other systems until they connect to it. When any system connects to the OCS, it will send the OCS the declarations of the commands that it supports and the monitoring values that it can supply. Thereafter, interactive users, the dynamic scheduler and observing scripts will be able to send commands to these systems and receive regular updates of dynamically-selected monitoring values at 0.1 s intervals. All systems that the OCS controls will be treated as independent self-contained control-systems, each of which will perform high level operations that are initiated by commands sent to them by the OCS.

*see for example the `slimdata` algorithm, `http://sourceforge.net/projects/slimdata/`

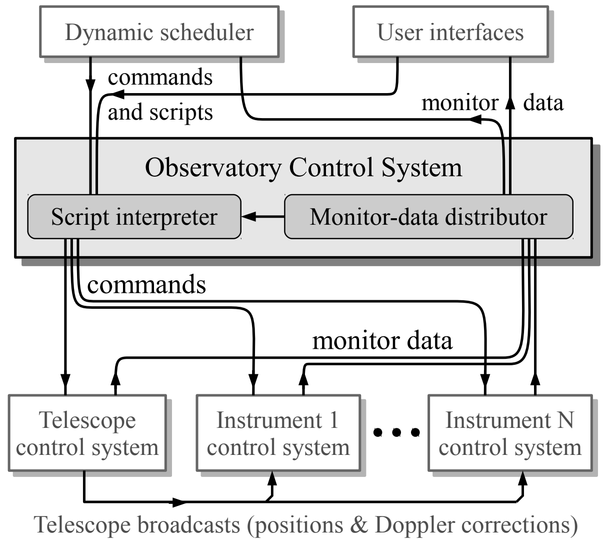

Figure 2. The topology of the Observatory Control System (OCS). The OCS contains a script interpreter, which sends commands to external systems, and a component that receives monitoring values from external systems and distributes them to user-interfaces, a dynamic scheduler and the script interpreter.

Coordination of the independent operations in the various instruments and the TCS will be performed by scripts. The scripts will be written in a custom high-level language, which has been designed for asynchronous control with bounded latencies. The language has the ability to quickly respond to monitoring feedback from any controlled system, including the ability to cancel and preempt any ongoing operation.

Communications between the OCS and controlled systems will occur on a 10 Hz communication cycle. During each cycle, a block of commands and a block of selected monitoring values will be exchanged between the OCS and each controlled system. The sizes of these blocks are too large for many industrial Ethernet protocols, and many of these protocols don't allow dynamic reconfiguration for new instruments, or the ability to send messages with varying contents. Rather than attempt to design our own protocol, TCP/IP will be used.

Using TCP/IP for control takes some care. TCP/IP is optimized for continuous streams of data sent over large distances. It quickly notices and retransmits dropped packets when the receiving TCP/IP stack sees a gap in the stream of packet sequence-numbers, but this only happens if more packets follow the dropped packets. A dropped packet at the end of a short control message isn't noticed or retransmitted for some time (0.2 s between Linux hosts). A similar problem occurs even when no packets are dropped, because of a well known interaction between Nagle's algorithm and delayed acknowledgments.[18] To remedy both of these issues, Nagle's algorithm is disabled by using the `TCP_NODELAY` configuration option, and the OCS communication scheme requires that whenever a message is sent to a recipient, the recipient send back an acknowledging reply. If this reply is not received within 25 ms, the sender transmits an extra message of a few packets, which simply asks the recipient to echo this message back to the sender. The extra packets in this echoed message reveal any gap in the stream of packets to the receiving TCP/IP stack, which then initiates TCP/IP's fast-retransmit scheme for the dropped packets.

The communication scheme described above has been tested on a cluster of 11 computers. One computer acted like the OCS server and the rest acted like client instruments. The goal was to verify that the server and each of the 10 client computers could repeatably exchange a simulated 1 kB-100 kB monitoring message and ten simulated command messages of 1 kB size within a 0.1 second deadline. During a contiguous period of 33 hours, the server and clients exchanged the above messages at successive intervals of 0.1 s, and all messages were successfully delivered well within their 0.1 s deadlines. This was not true when the test was repeated without the countermeasures discussed above. In that case many messages did miss their deadlines.

3. OBSERVATION PREPARATION

At CCAT there will be several ways of specifying an observation. For experimentation and direct control of all instrument facilities an approved observer can interact with the OCS directly and submit scripts using the custom language detailed in section 2. They are in full control of observing and bypass the dynamic scheduler.

For most observations, observers will prepare minimum schedulable blocks (MSBs) using the Observing Tool (OT). The OT can be used to prepare MSBs containing one or more observations such that the technical implementation details of a particular observing mode are hidden, allowing the astronomer to focus on the science requirements for each observation. The science specification of this MSB will be stored in a database for later querying by the dynamic scheduler. It is only once the scheduler has selected a particular MSB that the abstract view of the observing will be translated into an OCS observing script ready for execution.

Observers will also be able to supply an OCS script of the core observing mode but uploaded to the observation request database along with scheduling parameters and expected duration. The dynamic scheduler will select the MSB as usual but there will of course be no translation. Instead the scheduler will wrap the supplied script with code that will firstly ensure that the resulting data will be assigned to the correct observing project, and secondly, ensure that the script is stopped if it takes significantly longer to execute than was expected.

3.1 Tool Selection

A number of observing tools have been developed over the years[19–25] and after careful consideration the project has adopted the JCMT-OT,[21] as used at the James Clerk Maxwell Telescope (JCMT), to form the basis of the CCAT-OT for the initial development phase. The JCMT-OT is the sub-millimeter ground-based telescope version of the generic JAC-OT written by the Joint Astronomy Centre that also supports the United Kingdom Infrared Telescope (UKIRT).[26–28] It supports the HARP heterodyne receiver[29] and the SCUBA-2 bolometer array,[30] which are similar to CHAI and SWCam.

The JCMT-OT is written in Java and supports self-updating using Java Web Start. Visualisation of the instruments on the sky is provided by JSky.[31] Proposal preparation is not part of the JCMT-OT and we are considering using NorthStar,[32] which would work with minor modifications.

3.2 Modifications

The JCMT-OT will require modifications in order to support CCAT. In particular, code will need to be written to support the X-Spec multi-object spectrometer so that the individual beams can be positioned correctly.

MSB durations do not need to be extremely accurate, unlike for satellite missions, and time estimates can be done locally in the tool without requiring extensive server-side processing. An error of tens of seconds in a ten minute observation is acceptable given the use of a dynamic scheduler rather than a pre-determined night plan.

The atmospheric model used for the time estimators will have to be modified for Chajnantor and the high frequency bands supported by CHAI. We are also considering improving the spectral editor capabilities of the JCMT-OT to more closely resemble the functionality and look currently available in the ALMA-OT.[33]

The JCMT-OT source code is released under the GNU General Public License v2 for the JCMT components, with the original Gemini code using the 3-clause BSD license. The source code is available on a public repository*. The code will need some refactoring to meet the needs of CCAT users. The JCMT and UKIRT components are integrated within the code base and will need to be separated out. The code base is not supported by an accompanying set of unit test code, so there will be some effort required to bring the code up to more modern software development standards. These efforts will largely go unseen by the general astronomer. The front-end conversion of the tool to something that looks like a CCAT-OT will be relatively straightforward as seen in Figure 3 where we have already demonstrated the ability to define an SWCam component.

*`https://github.com/jac-h/observing-tool`

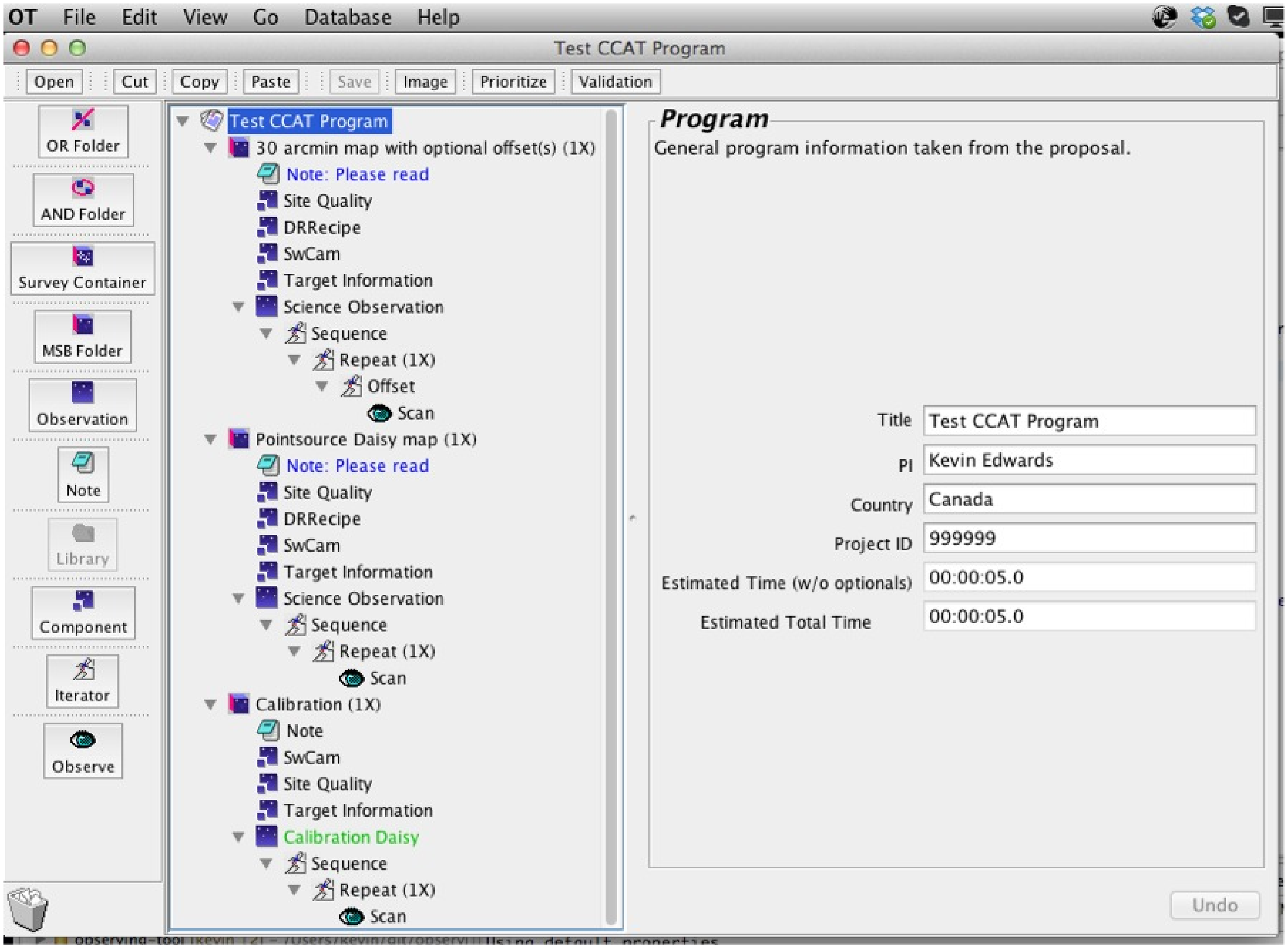

Figure 3. The CCAT-OT prototype showing a science program that includes MSBs with an SWCam component.

4. RAW DATA MODEL

The instrumental data rates are such that we have decided to use a loosely-coupled distributed data acquisition system where each system controlled by the OCS writes out a time-series independently of all other systems but where synchronization is managed by accurate recording of time stamps. It is up to the data reduction software to take the time stamps and determine which sequences are related. Each system writes out time-series using the same data model and a central data capturer task (see §5.1) collates the individual components and creates a linker file referencing them. The data capturer does not require highly synchronized coordination of data writing between systems.

We have evaluated the existing data models MBFITS,[34] NDF[35,36] and LOFAR.[37] In view of CCAT's demands, each of these data models has its advantages and disadvantages: MBFITS keeps data from different systems in different files, in accordance with the envisaged distributed data acquisition scheme; however, although MBFITS was designed and is being used for continuum and spectral line arrays conceptually similar to CCAT's first-light instruments, the data model is not flexible enough for new types of instruments like X-Spec. In contrast, NDF gives data authors wide freedom to design specialized data models on top of the general NDF model, and such data models for raw data (and data products) from continuum and spectral line arrays are in use; however, since NDF lacks mechanisms to establish links between structures in different files (such as the Hierarchical Grouping mechanism in FITS, or external links in HDF5), such links can only be expressed by location of files in the file system and directory and file names. The LOFAR Data Types (implemented in HDF5) are a family

```
<ObsID>.h5                  Group                   Basic metadata
|--OCS                      Group                   OCS metadata
|--...                                              other static metadata
|
|--TCS                      Group (ext. link)       static TCS metadata
|  |--TCS_00001             Group (ext. link)       TCS data
|   --TCS_00002             Group (ext. link)       TCS data
|
|--SWCam350                 Group (ext. link)       static SWCam350 metadata
|  |--SWCam350_00001        Group (ext. link)       SWCam350 data
|   --SWCam350_00002        Group (ext. link)       SWCam350 data
|
 --SWCam450                 Group (ext. link)       static SWCam450 metadata
   |--SWCam450_00001        Group (ext. link)       SWCam450 data
    --SWCam450_00002        Group (ext. link)       SWCam450 data
```

Figure 4. The HDF5 hierarchy of an observation with the 350 μm and 450 μm sections of SWCam. The HDF5 root group and other groups directly below the root group (like the shown OCS group which holds OCS-related metadata) contain observation-related static metadata. TCS and instrument data and metadata are stored in separate files, external links are used to establish a single HDF5 hierarchy for all data in the dataset.

of related hierarchical data models for raw data and data products for various LOFAR observing modes; they share common structures for common data and metadata and allow specialized structures. However, the data models reflect the specific structure of the LOFAR array and its observation modes too much to be used directly for a single-dish telescope with radically different observing modes.

We have decided to develop a new data model based on HDF5[38] as the low-level data format, but that shares the merits of the evaluated data models. During an observation, the data capturer, the TCS, and involved instruments write their data to HDF5 files independently. The set of these files forms a dataset that contains all data and metadata of the involved systems during this observation. In order to avoid excessive file sizes, bulk data from the TCS and science instruments will be recorded in sequences of data files which hold chunks of data for 30 s each.

In order to form a single HDF5 hierarchy from the HDF5 structures in the files of a dataset, HDF5's external links are used. The result is a HDF5 hierarchy with basic observation-related metadata at the root of the hierarchy, and TCS and instrument specific structures further down the hierarchy. Each OCS client system will write out structures in a standard way such that the TCS component of a CHAI observation will be identical to that of an SWCam observation. Furthermore, following the lead from NDF, structure layouts will be re-used wherever possible when designing the form of instrument-specific structures and, for example, the time field in every time-series table will use the same name and format to encourage code re-use and aid in cross-instrument understanding. There is, however, no requirement for each instrument to adopt data models that do not fit well with the needs of the particular instrument. This approach provides a good compromise between a well-constrained model and one with sufficient flexibility to cope with the specific needs of instruments. Figure 4 illustrates the proposed top-level HDF5 hierarchy.

HDF5's external links rely on pathnames of the referenced files; since absolute pathnames are not invariant when files are moved, only relative pathnames are used. This requires that all files of a dataset reside in a single directory tree. This can be achieved with mounts and (file-system) symbolic links and it is also likely that we will adopt the approach of using a distributed file system such as GPFS. Figure 5 sketches the resulting directory structure.

5. OBSERVATORY STATUS DATABASE

The CCAT Observatory Status Database (OSD) stores and retrieves information on all observatory operations and forms the central "brain" of the CCAT Observatory Data Management System. Comprising a heterogeneous

```
<DataRoot>/<ObsID>/          Dir                   Root directory for observation dataset
|--<ObsID>.h5                File                  Central linker file
|
|--TCS/                      Dir (symb. link)      Directory dedicated to TCS
|  |--<ObsID>_00001.h5       File                  TCS data
|   --<ObsID>_00002.h5       File                  TCS data
|
|--SWCam350/                 Dir (symb. link)      Directory dedicated to SWCam350
|  |--<ObsID>_00001.h5       File                  SWCam350 data
|   --<ObsID>_00002.h5       File                  SWCam350 data
|
 --SWCam450/                 Dir (symb. link)      Directory dedicated to SWCam450
   |--<ObsID>_00001.h5       File                  SWCam450 data
    --<ObsID>_00002.h5       File                  SWCam450 data
```

Figure 5. Directories and files for the dataset sketched in Fig. 4. The subdirectories with data from the TCS and instrument sections are mapped with mounts and symbolic links into the central directory structure. Data from the TCS and instruments is recorded in sequences of data files with 30 s of data each.

system of relational database and file system storage with file readers, the OSD has a programmatic interface for input and queries of operational CCAT data.

The OSD will have two geographically distinct components: the Live OSD at the Observatory and the Legacy OSD hosted at a CCAT Datacenter. Query access to both components will be through a single interface and by default the querying party will not know which component is serving their results.

5.1 CCAT Observatory data management

Figure 6 shows the overall data flow for CCAT data management. Key components include:

- Data Capturer: software system responsible for collating science data from the individual controlled systems, informing Obervatory Quicklook of the availability and location of the data and updating the OSD
- Housekeeping Data (HK): all non-detector data from Controlled Systems
- Data Transport System: Initiates network transfer and tracks network and physical transport of CCAT data files.

5.2 OSD Requirements

In addition to standard data storage requirements such as those relating to robustness and preservation of the integrity of relationships between data, specific key requirements on the OSD design include:

- The OSD will have to consume HK data at a rate of ∼Mbits/s 24 hours a day
- The OSD shall have capacity to hold 30 days of HK data before transfer to the Legacy OSD for permanent storage.
- The maximum allowed period after HK data are sent to the Live OSD to being available in query responses is 30 seconds.

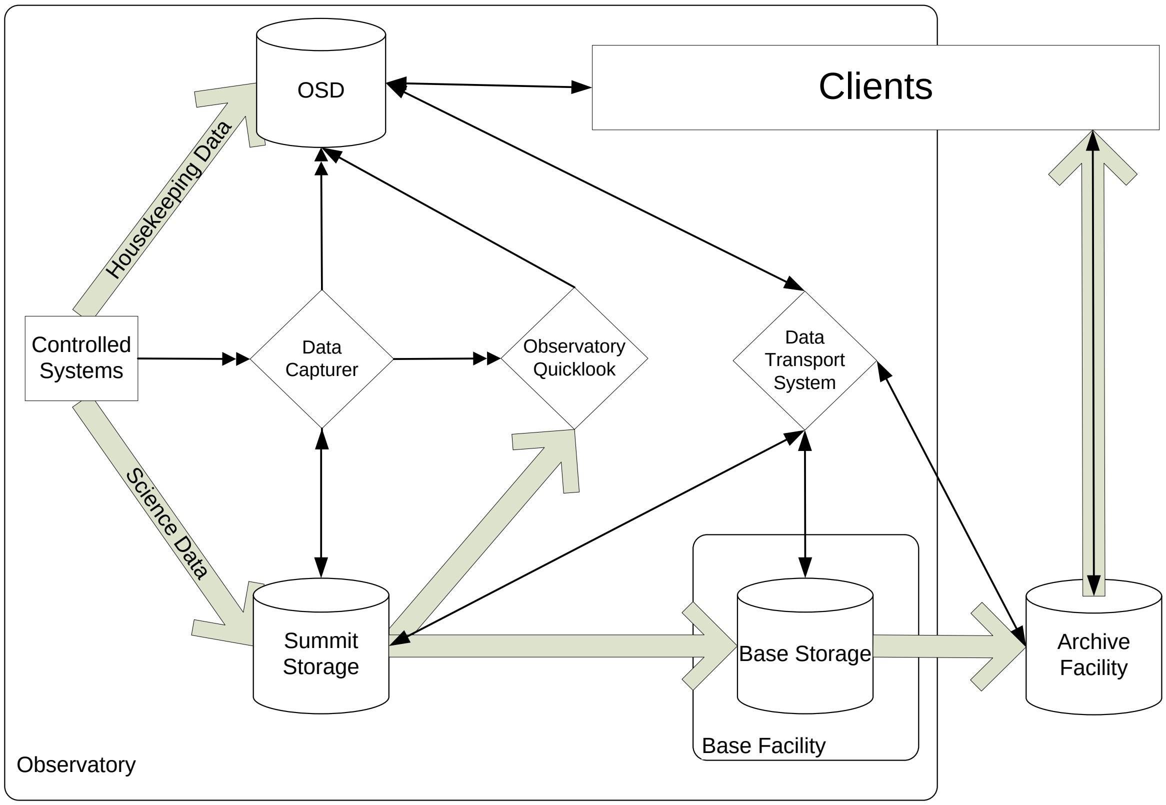

Figure 6. CCAT Data Management Schema. Thick arrows represent bulk data flow and thinner arrows represent control and metadata communications, with double-headed arrows representing "fast" connections with 1–3 s latency.

5.3 OSD Prototype

The OSD prototype is built in a development environment using the following technologies:

- Red Hat Enterprise Linux (RHEL) 6.5 for Controlled System and Data Storage machines including the file storage of HK data
- Microsoft SQL Server 2012 running on Windows Server 2012
- KVM for virtualization, running on an RHEL host
- Python 3.4 for the OSD Application Programming Interface (API)
- FreeTDS and `pyodbc` for database access (considering alternatives for ongoing development)
- Stored Procedures for typical database operations, with ownership chaining, bound parameters and delineated permissions to control access to the database.

The OSD prototype API consists of python modules to serve calls from client code, which are validated against function specifications in ancillary files and then executed, returning requested responses to the calling code. The API is installed using `setuptools` onto the client machine via a standard `setup.py` call and can be downloaded from the Github repository in its entirety.

The OSD prototype is a collection of tables simply distinguishing between data files, about which metadata are stored including the location of the file, and "data records", data which are to be stored in a SQL Server native type in database fields. A suite of stored procedures written in static parametrized SQL ensure that insertions of data and metadata preserve relationships between fields, which are enforced by foreign keys and allow efficient querying of the OSD.

6. OBSERVATION MANAGEMENT

The management of observations is a critical component of a flexibly scheduled telescope.[39–41] In general, Observations are defined by Scientists or by automated survey definition tools and have to be tracked to ensure that the highest priority observations are performed.

If someone is monitoring the observations there must be a way for time-stamped commentary to be recorded to provide additional information that is not available from the monitored computer systems. This could involve a comment on data quality, the reason why a particular instrument has been removed from the scheduler, or a statement from an engineer regarding why an instrument warmed up. The entry will contain the time the entry was made and also the time for which the entry was relevant as the comment may be made for some event that happened in the past. This system provides a narrative log of events at the telescope and why decisions have been made.

The data reduction pipelines running at the summit and base facility will generate quality assurance parameters automatically and will make this information available to the observing log. Additionally it shall be possible for people (for example the current observer, instrument team, staff or collaborators) to comment on a particular observation or a particular observation block. This can be done during observing or later on after data have been inspected more carefully.

It is important that scientists be able to inspect their data in near real-time and if necessary modify their observing program to optimize the science. All the logging information, along with pipeline products, quality assurance data, and monitoring data (such as weather statistics) will be made available to the astronomers so they can make informed decisions on data quality. There will also be a helpdesk system to allow astronomers to ask questions about their data and observing program.

7. SWCAM DATA REDUCTION

Current ground-based submillimeter instruments, such as SCUBA-2,[30] SHARC-2[42] and LABOCA,[43] have 100–1000s of detectors. SWCam will have ∼60,000 detectors across four wavelengths. Up to 48,000 detectors will be at 350 μm, the primary wavelength for the instrument. Additionally, the SWCam KID detectors will be sampled at ∼1000 Hz (to a maximum rate of 1500 Hz), while the bolometer detectors on current instruments are sampled at ∼100 Hz. This increase in sample rate is driven by the combination of smaller beam sizes and faster telescope scan rates at CCAT compared to existing facilities. The combined increase in number of detectors and sample rate for SWCam compared to current instruments results in a factor of 55 increase in data size over the largest detectors today. It will be a challenge to reduce these large datasets.

Map-making in the context of the next generation of submillimeter cameras, such as SWCam, has been described elsewhere.[44] The above-mentioned instruments all make use of iterative techniques to reduce common-mode and correlated noise.[45–48] These map-makers are based on similar algorithms; we focus on SMURF,[45,49] the SCUBA-2 map-maker, as it was developed by the current team members, is highly configurable, is part of the open source Starlink software collection[50†], and is in active development by the Joint Astronomy Centre[51‡]. Ref. 44 estimates how the SMURF run time will scale from SCUBA-2 to SWCam, and concludes that to keep up with data collection, we will need either (i) several high-end machines running on independent datasets or (ii) a cluster of machines running a parallelized version of the map-maker. The parallel option has the additional

† `http://www.starlink.ac.uk`
‡ `http://pipelinesandarchives.blogspot.com`

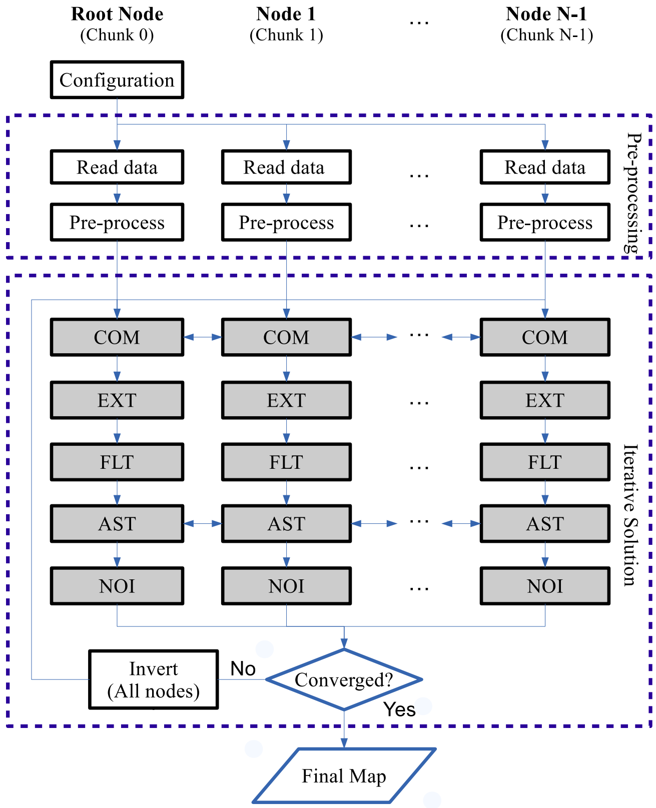

Figure 7. A schematic for how distributed-memory parallelization of SMURF might work. The individual models are indicated by a three letter code: `COM` is the common-mode removal, `EXT` is atmospheric extinction correction, `FLT` is a Fourier filter, `AST` is the astronomical signal and `NOI` is a noise model. Horizontal arrows indicate modules where inter-node communication is required. The others are trivially parallelized. Figure reproduced from Ref. 44.

advantage that larger datasets (longer than about 15 minutes of observation) can be reduced without caching to disk, as the data can be divided amongst the machines. In light of this, we have investigated how SMURF could be modified to run on a distributed-memory cluster.

Figure 7 shows a schematic of how a parallelized SMURF might be laid out. In this model, each process manages a unique chunk of data, split so that each chunk contains the full time streams for a number of detectors. The iterative algorithm is modular, solving for a number of (user-specified) models sequentially during each iteration. The program runs serially in each process, with communications between processes occuring when necessary. Some models, such as the filter model, which applies a high-pass filter to each time stream in order to reduce low-frequency detector and atmospheric noise, are trivially parallelized, as each process can apply the filter to its chunk of detectors, without the need to communicate with other processes. Other models, such as the common-mode model, which removes a signal common to all detectors at each time sample, require communication between all processes. Due to this communication overhead, in addition to the fact that the program will proceed at the rate of the slowest process, it is not expected that the run time will scale as the inverse of the number of processes.

A proof-of-concept implementation of the algorithm described above has been written[§] to explore the scaling of run time with the number of processors for a range of data sizes. The data sets used have 64, 256 and 4096 detectors, laid out in a square array, with the spacing adjusted so that the detector array has the same angular size on the sky, ensuring that the sky coverage is the same in all simulations. The simulated data sets are created with 1800 s of data and a sampling rate of 1500 Hz. The timing tests are run on Grex, a compute cluster that

[§] `https://github.com/CCATObservatory/mpi-mapmaker-test`

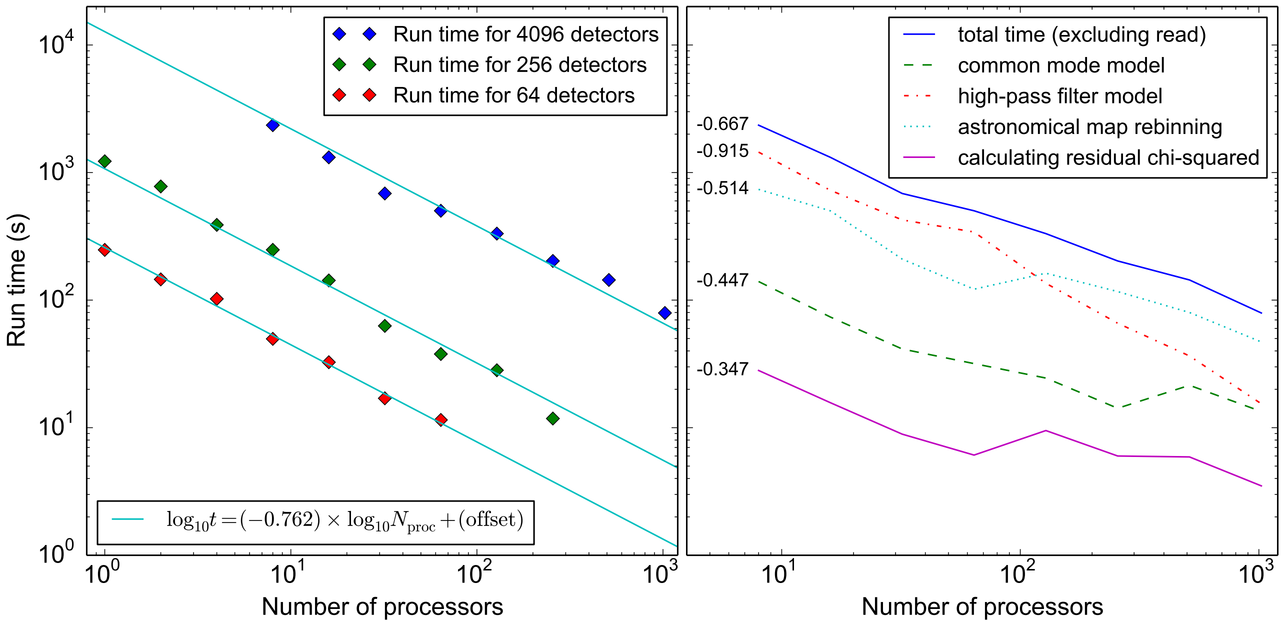

Figure 8. Results of timing tests of SWCam prototype parallel map-maker. **Left panel:** Three timing tests for three detector counts (indicated by colored points). The input data consist of 1800 s of data sampled at 1500 Hz. All three tests are fit simultaneously with a common power-law index (solid lines), showing that a common slope is consistent with the data. **Right Panel:** Breakdown of subroutine run times for the $N_{\rm det} = 4096$ data set. The best-fit power-law index for each component is indicated at the left edge of the associated curve.

is part of the WestGrid network.¶ Grex consists of a total of 316 nodes, each with two 6-core 2.66 GHz CPUs. The cluster features 24 nodes with 96 GB memory and the remaining 292 have 48 GB. For each data set, we have run the map-maker using a range of number of nodes. Where possible (namely the $N_{\rm det} = 64$ and 256 data sets), we start with one processor and increase the number by factors of two until each process is operating on one detector ($N_{\rm proc} = 64$ and 256, respectively). For the $N_{\rm ndet}$= 4096 data set, the memory requirements prohibit using fewer than 8 processors (72 GB memory required per process), and we stop at $N_{\rm proc} = 1024$, a large fraction of the available number of processors.

The results of the timing tests are shown in the left panel of Figure 8. For each of the three datasets, run time is plotted vs. number of processors used in the map-making run. The datasets are simultaneously fit with a power law with common power-law index, $t^j = A^j \times (N_{\rm proc})^{-\alpha}$, where the index j labels the three different datasets. The best-fit index is $\alpha = 0.762$; this is shallower than perfect scaling ($\alpha = 1.0$), but that is to be expected due to communication time overhead. It is encouraging, however, that the power-law relation appears to continue to large $N_{\rm proc}$. While doubling the number of available processors does not halve the run time, it does significantly improve the run time, reducing it by about 40 per cent, and additionally reduces the per-processor memory requirements (by about a factor of two).

The right-hand panel of Figure 8 shows how the run times of individual noise/signal models scale with the number of processors. It shows the breakdown of run times for the $N_{\rm det} = 4096$ dataset, including the best-fit power-law indices for each component (annotated at the left edge of each line). We see that the high-pass filter model, which runs independently in each process, not requiring communications between processes, falls steeply, with a slope of nearly $\alpha = 1.0$. The other models, the common-mode removal and map rebinning, require communication between all processes, and thus do not scale as steeply as $\alpha = 1.0$.

While the run time does not scale perfectly with the number of processors used, the improvement is still significant. Ref. 44 states that 16 minutes of SCUBA-2 data can be reduced on a dual quad-core CPU in 7

¶`http://www.westgrid.ca`

minutes using 33 GB of memory. Using the data scaling factor of 55 from SCUBA-2 to SWCam 350 μm, this same machine will take about six hours to reduce 16 minutes of SWCam 350 μm data. Using the overall best-fit power-law index of Figure 8, $\alpha = 0.762$, a cluster of 50 of the above-described machines reduces the run time by a factor of 20 to ~ 20 minutes. The memory required is about 90 GB per node. Assuming that CCAT observes 12 hours per day, the parallel algorithm described here will therefore be able to keep up with data collection using mid-level hardware available today.

8. CHAI DATA REDUCTION

Even in its smallest configuration and readout rate CHAI will be generating hundreds of thousands of spectra per night and for large area on-the-fly mapping there may be millions. Reducing those spectra requires automated pipelines capable of detecting bad spectra and removing baseline artifacts with minimal input from a human operator.

The CHAI data reduction has to perform the reference subtraction, including a correction for drifts, the frequency and intensity calibration, the flagging of known problems, such as bad channels or pixels, a baseline subtraction, and the evaluation of the resulting data quality in terms of noise and the lack of unknown artifacts. The pipeline[52–55] for the HARP instrument[29] at the JCMT is designed specifically for automated cube creation and we are using it to investigate scaling and performance issues from a 16-element instrument to larger arrays. We are testing the pipeline on two data sets. The first data set comes from the SMART[56–58] instrument on NANTEN2 providing direct comparison with the current Köln data reduction software written in CLASS which is part of the GILDAS data reduction package[59]*. The second data set comes from commissioning data from the 64-element Supercam[60] on the Heinrich Hertz Submillimeter Telescope (HHSMT). The Supercam data has eight times the detector count of HARP although the channel count, 900 channels, is nine times lower than the 8192 channels usually present in HARP spectra. Both these data sets provide different tests of the pipeline infrastructure and how the algorithm behaves as more detectors and channels are used and will provide excellent feedback into the CHAI pipeline design phase.

9. SOFTWARE DEVELOPMENT

The CCAT software team is currently distributed over three countries, seven institutions and twelve timezones and more institutions are expected to contribute as the project enters the construction phase. Distributed teams can result in difficulties in communication and the motivation of isolated team members. Fast networks, ubiquitous webcam availability, agile methodologies and advances in web site technologies continue to aid distributed software development and within CCAT we have implemented several approaches to maximize team effectiveness.

9.1 Semi-annual face to face meetings

There is still no replacement for face to face meetings to maximize information transfer between team members. Full team meetings are held every six months and are a critical aspect of team building. They allow people to resolve misundertandings that built up over the intervening months, as well as provide a social setting in the evenings and during breaks to build up a rapport with other team members that can not be achieved when you only know the person over email in a professional context.

9.2 Weekly "standups"

In the northern hemisphere summer there are twelve hours between Hawaii and Germany and it is unreasonable to expect daily full team meetings when participants in Hawaii have just woken up and those in Germany are eating their evening meal. As a compromise we have a full team video conference call each week to summarize progress and report on any tasks that are being blocked. We have investigated a number of different video conferencing technologies including Google Hangout†, GoToMeeting‡ and Zoom§. Each of these are capable

*http://www.iram.fr/IRAMFR/GILDAS
†https://plus.google.com
‡http://www.gotomeeting.com
§https://www.zoom.us

of screen sharing and ten video participants. They differ somewhat in pricing strategies and the ability for people to call in from a telephone. For example, Google Hangouts let you add people by calling their number, GoToMeeting provides call in numbers for multiple countries and Zoom provides a US toll-free number where the host must pay a per-minute charge. Ideally people would call in using apps on their smartphones and tablets if they are not using a computer but during the transition from phone conferencing to video conferencing there is still a need to support the telephone system.

9.3 Team communication

Mailing lists exist for each workpackage but email discussions can become unwieldy as a topic is discussed over many days with many levels of quoting. A mailing list is fine for a short broadcast to team members but we are also considering collaborative instant messaging tools such as Campfire,[¶], FlowDock[‖], and HipChat.[**] These tools allow general conversations to occur throughout the day and lower the barrier for asking quick questions to other team members. They do not, however, solve the issues associated with debating larger topics over many days.

We are considering discussion tools such as Discourse[††]. Fora have the best potential for simplifying long form debates on a particular topic as they allow quoting of particular paragraphs and responses to responses whilst keeping the information in a single location and not spread over a hundred emails. In some sense the code review features of Github could easily serve a similar purpose although expecting people to submit discussion topics to a git repository dedicated to this purpose may be a step too far for people (developing long-form documentation using Github is done regularly). Ideally it should be possible to upload gists[‡‡] and allow immediate inline commentary but at present Github do not support this.

9.4 Source Control and Collaboration

Distributed revision control systems, such as `git`, that are designed with distributed teams in mind are a huge aid to modern software development. Branching in a repository is now seen as an every day event rather than something that only the brave should attempt.

We have looked at both Github and Atlassian's Bitbucket and have decided to use Github since its collaboration tools are significantly more powerful and easier to use. One key aspect is the integrated wikis that are themselves hosted as git repositories and allow the use of Markdown along with many other markup languages. We develop using feature branches and make use of the code review features and issue trackers provided by Github.

We have available up to 20 private git repositories that can be used for internal document development and for early development of modules, but the default is for all CCAT source code to be developed in public under a 3-clause BSD license in a similar approach to that taken by LSST (albeit with a different, less restrictive, license).[61*]

9.5 Kanban boards

The Kanban approach to software development[62] is a very popular agile technique but becomes difficult in a distributed team when a physical Kanban board is being used. Online Kanban boards are now available from many companies and currently we favor Trello[†] although the integration with Github issues is not optimal.

[¶] `https://campfirenow.com`

[‖] `https://www.flowdock.com`

[**] `http://www.hipchat.com`

[††] `http://www.discourse.org`

[‡‡] `https://gist.github.com`

[*] For details on the LSST software licensing policy see `https://dev.lsstcorp.org/trac/wiki/SWLicense`

[†] `http://trello.com`

ACKNOWLEDGMENTS

The CCAT Submillimeter Observatory (CCAT) is owned and operated by a consortium of universities and non-profit organizations located in the United States, Canada and Germany. Specifically the CCAT Consortium is comprised of: Cornell University, California Institute of Technology (Caltech), University of Colorado at Boulder, University of Cologne, University of Bonn, Dalhousie University, McGill University, McMaster University, University of British Columbia, University of Calgary, University of Toronto, University of Waterloo, University of Western Ontario and Associated Universities, Incorporated. The CCAT Engineering Design Phase was partially supported by funding from the National Science Foundation via AST-1118243. We thank William Peters for making Supercam commissioning data available to us for testing. We also thank Steve Padin, John Carpenter and Jeff Zivick for helpful comments on the manuscript. The SWCam prototype map-maker makes use of facilities provided by WestGrid and Compute Canada Calcul Canada.‡

REFERENCES

[1] Sebring, T., "The Cornell Caltech Atacama Telescope: progress and plans 2010," in [*Ground-based and Airborne Telescopes III*], Stepp, L. M., Gilmozzi, R., and Hall, H. J., eds., *Proc. SPIE* **7733**, 77331X (2010).

[2] Woody, D., Padin, S., Chauvin, E., et al., "The CCAT 25m diameter submillimeter-wave telescope," in [*Ground-based and Airborne Telescopes IV*], Stepp, L. M., Gilmozzi, R., and Hall, H. J., eds., *Proc. SPIE* **8444**, 84442M (2012).

[3] Glenn, J. et al., "The CCAT Telescope," in [*American Astronomical Society Meeting*], *American Astronomical Society Meeting Abstracts* **221**, #150.06 (2013).

[4] Jenness, T., Brazier, A., Edwards, K., et al., "The CCAT software System," in [*Astronomical Data Analysis Software and Systems XXIII*], Manset, N. and Forshay, P., eds., *ASP Conf. Ser.* **485**, 49, ASP, San Francisco (2014). arXiv:1401.8280.

[5] Jenness, T., "Transient Alert Follow-up Planned for CCAT," in [*Hot-Wiring the Transient Universe 3*], Wozniak, P., Graham, M., and Mahabal, A., eds. (2014). arXiv:1402.1202.

[6] Radford, S. J. E., "Observing Conditions for Submillimeter Astronomy," in [*Astronomical Site Testing Data in Chile*], *Revista Mexicana de Astronomia y Astrofisica Conference Series* **41**, 87–90 (2011).

[7] Stacey, G. J., Parshley, S., Nikola, T., et al., "SWCam: the short wavelength camera for the CCAT observatory," in [*Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VII*], Holland, W. S. and Zmuidzinas, J., eds., *Proc. SPIE* **9153**, 915321 (2014).

[8] Stacey, G. J., Parshley, S., Nikola, T., et al., "The Design of the Short Wavelength Camera for the CCAT Telescope," in [*American Astronomical Society Meeting*], *American Astronomical Society Meeting Abstracts* **221**, #150.07 (2013).

[9] Rajagopalan, G., Kovacs, A., Monroe, R. M., et al., "Readout electronics for the CCAT observatory's instruments at first light and beyond," in [*Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VII*], Holland, W. S. and Zmuidzinas, J., eds., *Proc. SPIE* **9153**, 9153124 (2014).

[10] Goldsmith, P. et al., "Studying the Formation and Development of Molecular Clouds: with the CCAT Heterodyne Array Instrument (CHAI)." `http://trs-new.jpl.nasa.gov/dspace/bitstream/2014/42645/1/12-2032_A1b.pdf` (2012).

[11] Golwala, S. R. et al., "The Design and Science Goals of LWCam, the CCAT Long-Wavelength Imager," in [*American Astronomical Society Meeting*], *American Astronomical Society Meeting Abstracts* **221**, #150.08 (2013).

[12] Bradford, C. M., Hailey-Dunsheath, S., Shirokoff, E. D., Hollister, M. I., McKenney, C. M., Chapman, S., Tikhomirov, A., and Nikola, T., "X-Spec: a multi-object trans-millimeter-wave spectrometer for CCAT," in [*Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VII*], Holland, W. S. and Zmuidzinas, J., eds., *Proc. SPIE* **9153**, 915370 (2014).

‡`http://www.computecanada.ca`

[13] Bradford, C., Hailey-Dunsheath, S., Shirokoff, E., et al., "X-Spec: A Multi-Object Wideband Survey Spectrograph for CCAT," in [*American Astronomical Society Meeting*], *American Astronomical Society Meeting Abstracts* **221**, #150.09 (2013).

[14] Fowler, J. W., "The Atacama Cosmology Telescope Project," in [*Millimeter and Submillimeter Detectors for Astronomy II*], Zmuidzinas, J., Holland, W. S., and Withington, S., eds., *Proc. SPIE* **5498**, 1–10 (2004).

[15] Padin, S., Shepherd, M. C., Cartwright, J. K., et al., "The Cosmic Background Imager," *PASP* **114**, 83–97 (2002).

[16] Bischoff, C. et al., "The Q/U Imaging ExperimenT Instrument," *ApJ* **768**, 9 (2013).

[17] Richards, J. L., Max-Moerbeck, W., Pavlidou, V., et al., "Blazars in the Fermi Era: The OVRO 40 m Telescope Monitoring Program," *ApJS* **194**, 29 (2011).

[18] Cheshire, S., "TCP Performance problems caused by interaction between Nagles Algorithm and Delayed ACK." `http://www.stuartcheshire.org/papers/NagleDelayedAck/` (2005).

[19] Wampler, S., Gillies, K. K., Puxley, P. J., and Walker, S., "Science planning for the Gemini 8-m telescopes," in [*Telescope Control Systems II*], Lewis, H., ed., *Proc. SPIE* **3112**, 246–253 (1997).

[20] Frayer, D. T. et al., "Tools for the Herschel Space Observatory: Observation Planning and Data Processing," in [*American Astronomical Society Meeting Abstracts #210*], *Bulletin of the American Astronomical Society* **39**, 108 (2007).

[21] Folger, M., Bridger, A., Dent, B., Kelly, D., Adamson, A., Economou, F., Hirst, P., and Jenness, T., "A New Observing Tool for the James Clerk Maxwell Telescope," in [*Astronomical Data Analysis Software and Systems XI*], Bohlender, D. A., Durand, D., and Handley, T. H., eds., *ASP Conf. Ser.* **281**, 453 (2002).

[22] Storrie-Lombardi, L. J. and SSC Team, "Planning Observations for a SIRTF Legacy Science Proposal," in [*American Astronomical Society Meeting Abstracts*], *Bulletin of the American Astronomical Society* **31**, #119.07 (1999).

[23] Rebull, L. M. and SSC Observer Support Team, "Planning your observations with the Spitzer Space Telescope," in [*13th Cambridge Workshop on Cool Stars, Stellar Systems and the Sun*], Favata, F., Hussain, G. A. J., and Battrick, B., eds., *ESA Special Publication* **560**, 895 (2005).

[24] Bridger, A., Williams, S., McLay, S., Yatagai, H., Schilling, M., Biggs, A., Tobar, R., and Warmels, R. H., "The ALMA OT in early science: supporting multiple customers," in [*Software and Cyberinfrastructure for Astronomy II*], Radziwill, N. M. and Chiozzi, G., eds., *Proc. SPIE* **8451**, 84511A (2012).

[25] Vacca, W. D. and Sandell, G., "US SOFIA General Investigator Program: How to Propose; How to Plan," in [*American Astronomical Society Meeting Abstracts*], *Bulletin of the American Astronomical Society* **37**, #159.05 (2005).

[26] Economou, F., Jenness, T., and Rees, N. P., "Sharing code and support between heterogeneous telescopes: the UKIRT and JCMT joint software projects," in [*Observatory Operations to Optimize Scientific Return III*], Quinn, P. J., ed., *Proc. SPIE* **4844**, 321–330 (2002).

[27] Bridger, A., Wright, G. S., Economou, F., et al., "ORAC: a modern observing system for UKIRT," in [*Advanced Telescope and Instrumentation Control Software*], Lewis, H., ed., *Proc. SPIE* **4009**, 227–238 (2000).

[28] Jenness, T. and Economou, F., "Data Management at the UKIRT and JCMT," in [*Telescopes from Afar*], Gajadhar, S., Walawender, J., Genet, R., Veillet, C., Adamson, A., Martinez, J., Melnik, J., Jenness, T., and Manset, N., eds., 42 (2011). arXiv:1111.5855.

[29] Buckle, J. V., Hills, R. E., Smith, H., et al., "HARP/ACSIS: a submillimetre spectral imaging system on the James Clerk Maxwell Telescope," *MNRAS* **399**, 1026–1043 (2009).

[30] Holland, W. S., Bintley, D., Chapin, E. L., et al., "SCUBA-2: the 10 000 pixel bolometer camera on the James Clerk Maxwell Telescope," *MNRAS* **430**, 2513–2533 (2013).

[31] Dolensky, M., Albrecht, M., Albrecht, R., et al., "Java for astronomy: software development at ESO/ST-ECF.," *The Messenger* **98**, 30–32 (1999).

[32] Smit, A. and Holties, H., "NorthStar — The RadioNet Common Proposal Tool." AstroGrid Science Workshop and RadioNet/AstroGrid workshop for radio data providers, Oxford University (2006). `http://wiki.astrogrid.org/pub/Astrogrid/RadioAgenda/NSAstroGrid08122006.pdf`.

[33] Williams, S. and Bridger, A., "Spectral Line Selection in the ALMA Observing Tool," in [*Astronomical Data Analysis Software and Systems XXII*], Friedel, D. N., ed., *ASP Conf. Ser.* **475**, 373 (2013).

[34] Muders, D., Hafok, H., Wyrowski, F., et al., "APECS - the Atacama pathfinder experiment control system," *A&A* **454**, L25–L28 (2006).

[35] Jenness, T., Berry, D. S., Currie, M. J., et al., "Learning from 25 years of the extensible N-Dimensional Data Format," *Astron. Comp.* **submitted** (2014).

[36] Economou, F., Jenness, T., Currie, M. J., and Berry, D. S., "Advantages of Extensible Self-described Data Formats: Lessons Learned from NDF," in [*Astronomical Data Analysis Software and Systems XXIII*], Manset, N. and Forshay, P., eds., *ASP Conf. Ser.* **485**, 355 (2014).

[37] Alexov, A., Schellart, P., ter Veen, S., et al., "Status of LOFAR Data in HDF5 Format," in [*Astronomical Data Analysis Software and Systems XXI*], Ballester, P., Egret, D., and Lorente, N. P. F., eds., *ASP Conf. Ser.* **461**, 283 (2012).

[38] Folk, M., Heber, G., Koziol, Q., Pourmal, E., and Robinson, D., "An Overview of the HDF5 Technology Suite and Its Applications," in [*Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases*], *AD '11*, 36–47, ACM (2011).

[39] Economou, F., Jenness, T., Tilanus, R. P. J., Hirst, P., Adamson, A. J., Rippa, M., Delorey, K. K., and Isaak, K. G., "Flexible Software for Flexible Scheduling," in [*Astronomical Data Analysis Software and Systems XI*], Bohlender, D. A., Durand, D., and Handley, T. H., eds., *ASP Conf. Ser.* **281**, 488 (2002).

[40] Adamson, A. J., Tilanus, R. P., Buckle, J., Davis, G. R., Economou, F., Jenness, T., and Delorey, K., "Science returns of flexible scheduling on UKIRT and the JCMT," in [*Optimizing Scientific Return for Astronomy through Information Technologies*], Quinn, P. J. and Bridger, A., eds., *Proc. SPIE* **5493**, 24–32 (2004).

[41] Dempsey, J. T., Jenness, T., Economou, F., et al., "Setting the standard: 25 years of operating the JCMT," in [*Observatory Operations: Strategies, Processes, and Systems V*], Peck, A. B., Benn, C. R., and Seaman, R., eds., *Proc. SPIE* **9149**, 914951 (2014).

[42] Dowell, C. D., Allen, C. A., Babu, R. S., et al., "SHARC II: a Caltech submillimeter observatory facility camera with 384 pixels," in [*Millimeter and Submillimeter Detectors for Astronomy*], Phillips, T. G. and Zmuidzinas, J., eds., *Proc. SPIE* **4855**, 73–87 (2003).

[43] Siringo, G., Kreysa, E., Kovács, A., et al., "The Large APEX BOlometer CAmera LABOCA," *A&A* **497**, 945–962 (2009).

[44] Marsden, G., Brazier, A., Jenness, T., Sayers, J., and Scott, D., "Map-making for the Next Generation of Ground-based Submillimeter Instruments," in [*Astronomical Data Analysis Software and Systems XXIII*], Manset, N. and Forshay, P., eds., *ASP Conf. Ser.* **485**, 399, ASP, San Francisco (2014). arXiv:1405.0482.

[45] Chapin, E. L., Berry, D. S., Gibb, A. G., Jenness, T., Scott, D., Tilanus, R. P. J., Economou, F., and Holland, W. S., "SCUBA-2: iterative map-making with the Sub-Millimetre User Reduction Facility," *MNRAS* **430**, 2545–2573 (2013).

[46] Kovács, A., "CRUSH: fast and scalable data reduction for imaging arrays," in [*Millimeter and Submillimeter Detectors and Instrumentation for Astronomy IV*], Duncan, W. D., Holland, W. S., and Withington, S., eds., *Proc. SPIE* **7020**, 70201S (2008).

[47] Schuller, F., "BoA: a versatile software for bolometer data reduction," in [*Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VI*], Holland, W. S., ed., *Proc. SPIE* **8452**, 84521T (2012).

[48] Kovács, A., "CRUSH: data reduction and imaging for future (sub)millimeter arrays," in [*Software and Cyberinfrastructure for Astronomy III*], Chiozzi, G. and Radziwill, N. M., eds., *Proc. SPIE* **9152**, 915219 (2014).

[49] Jenness, T., Chapin, E. L., Berry, D. S., Gibb, A. G., Tilanus, R. P. J., Balfour, J., Tilanus, V., and Currie, M. J., "SMURF: SubMillimeter User Reduction Facility," (2013). Astrophysics Source Code Library ascl:1310.007.

[50] Currie, M. J., Berry, D. S., Jenness, T., Gibb, A. G., Bell, G. S., and Draper, P. W., "Starlink in 2013," in [*Astronomical Data Analysis Software and Systems XXIII*], Manset, N. and Forshay, P., eds., *ASP Conf. Ser.* **485**, 391 (2014).

[51] Thomas, H. S., "The SCUBA-2 Data Reduction Cookbook." Starlink Cookbook 21, Joint Astronomy Centre (2013).

[52] Jenness, T., Cavanagh, B., Economou, F., and Berry, D. S., "JCMT Science Archive: Advanced Heterodyne Data Products Pipeline," in [*Astronomical Data Analysis Software and Systems XVII*], Argyle, R. W., Bunclark, P. S., and Lewis, J. R., eds., *ASP Conf. Ser.* **394**, 565 (2008).

[53] Currie, M. J., "Automated Removal of Bad-Baseline Spectra from ACSIS/HARP Heterodyne Time Series," in [*Astronomical Data Analysis Software and Systems XXII*], Friedel, D. N., ed., *ASP Conf. Ser.* **475**, 341 (2013).

[54] Jenness, T., Currie, M. J., Cavanagh, B., Berry, D. S., Leech, J., and Rizzi, L., "Automated reduction of single-dish heterodyne data from the James Clerk Maxwell Telescope using ORAC-DR," *Astronomy & Computing* **in preparation** (2014).

[55] Jenness, T., Economou, F., Cavanagh, B., Currie, M., and Gibb, A., "ORAC-DR: Astronomy data reduction pipeline," (2013). Astrophysics Source Code Library ascl:1310.001.

[56] Graf, U. U., Heyminck, S., Michael, E. A., et al., "SMART: The KOSMA Sub-Millimeter Array Receiver for Two frequencies," in [*Millimeter and Submillimeter Detectors for Astronomy*], Phillips, T. G. and Zmuidzinas, J., eds., *Proc. SPIE* **4855**, 322–329 (2003).

[57] Kawamura, A., Mizuno, N., Yonekura, Y., Onishi, T., Mizuno, A., and Fukui, Y., "NANTEN2: A Submillimeter Telescope for Large Scale Surveys at Atacama," in [*Astrochemistry: Recent Successes and Current Challenges*], *IAU Symposium* **235**, 275P (2005).

[58] Graf, U. U., Honingh, C. E., Jacobs, K., Justen, M., Pütz, P., Schultz, M., Wulff, S., and Stutzki, J., "Upgrade of the SMART Focal Plane Array Receiver for NANTEN2," in [*Nineteenth International Symposium on Space Terahertz Technology*], Wild, W., ed., 488 (2008).

[59] Gildas Team, "GILDAS: Grenoble Image and Line Data Analysis Software," (2013). Astrophysics Source Code Library ascl:1305.010.

[60] Kloosterman, J., Cottam, T., Swift, B., et al., "First observations with SuperCam and future plans," in [*Millimeter, Submillimeter, and Far-Infrared Detectors and Instrumentation for Astronomy VI*], Holland, W. S., ed., *Proc. SPIE* **8452**, 845204 (2012).

[61] Axelrod, T., Kantor, J., Lupton, R. H., and Pierfederici, F., "An open source application framework for astronomical imaging pipelines," in [*Software and Cyberinfrastructure for Astronomy*], *Proc. SPIE* **7740**, 774015 (2010).

[62] Kniberg, H., [*Lean from the Trenches: Managing Large-Scale Projects with Kanban*], The Pragmatic Programmers, 1st ed. (2011). ISBN-13: 978-1-934356-85-2.