ChaRT: The Chandra Ray Tracer

Clayton Carter, Margarita Karovska, Diab Jerius, Kenny Glotfelty and Steve Beikman

Harvard-Smithsonian Center for Astrophysics, 60 Garden St., Cambridge, MA 02138

Abstract. In this paper we present the Chandra Ray Tracer (*ChaRT*), a distributed remote-computing application developed by the Chandra X-Ray Center (CXC) to simulate the High Resolution Mirror Assembly (HRMA) Point Spread Functions (PSFs). We will discuss the overall system architecture and the programmatic flow. This approach may be used as a prototype for other projects where either the software cannot be distributed and/or the system resources to run the software would be prohibitive for the general user.

1. Introduction

Chandra produces sharper images than any other X-ray telescope to date and therefore provides an opportunity for high-angular and spectral resolution studies of X-ray sources. Crucial to these studies is the knowledge of the characteristics of the PSF. The blurring of the Chandra PSF is introduced by the HRMA PSF, the aspect, the limited size of detector pixels and detector effects. Simulating the HRMA PSF is the first and most important step in obtaining a good model of the Chandra PSF for a given observation.

Currently available PSF library files (Karovska et al. 2000) provide the general users with a grid of precomputed monochromatic HRMA PSFs. To calculate a HRMA PSF matching the location and spectrum of the observed source, users interpolate between model PSFs on a coarse spatial and energy grid using the *CIAO* tool `mkpsf`[1]. Model PSFs derived in this way suffer from a limiting precision which makes them unsuitable for very detailed analysis. To address this, the CXC has developed *ChaRT*, a soft- and hardware system designed to provide the general user community with access to the most precise and up to date HRMA PSF simulations available.

In the following, we describe the details of the *ChaRT* design and implementation, as well as the data products.

[1] *CIAO* is the Chandra Interactive Analysis of Observations data analysis system. *CIAO* is available at `http://cxc.harvard.edu/ciao`

2. ChaRT System Design

The HRMA was designed to produce on-axis images with better than one arc-second resolution. However, the shape and size of the HRMA PSFs vary significantly with source location in the telescope field of view, as well as with the spectral energy distribution of the source. As such, in order to carry out spatial analysis of the *Chandra* data, each PSF must be simulated individually.

For calibration purposes, the *Chandra* Mission Support Team developed ray tracing code (*SAOsac*, Jerius et al 2000) that can simulate the HRMA PSF using the official mirror model, and includes many of the details of the HRMA's physical construction. These include the stray light baffles and support structures as well as a detailed model of the reflective properties of the mirror surface. *SAOsac*, however, was designed for engineering support and analysis, and – as a result – is currently unsuitable for release to the general Chandra community. (Though work to change this is ongoing.)

ChaRT is a soft- and hardware system designed to provide *Chandra* users with access to the *SAOsac* ray tracing routines and allow them to carry out HRMA PSF simulations. Using *ChaRT*'s user friendly web interface, the HRMA PSF can be simulated for point sources at multiple locations, each source characterized by either a monochromatic energy or an input spectrum. *ChaRT* verifies and submits the simulation parameters and notifies the user when their simulated PSFs are available for download via FTP. This is accomplished using middleware, developed at the CXC, that works in conjunction with some commonly available software components (Apache[2] and PBS[3], for instance) on a distributed computer network located at the CXC.

2.1. Goals and Implementation

In developing *ChaRT*, we sought to produce an application that was simple to use and configure, easy to maintain, lightweight and extensible. Our goal was to design this system as fault tolerant, scalable and resistant to abuse and attacks.

To that end, we created a set of Perl[4] CGI scripts which provide a simple interface for the user to enter source specifications for PSF simulations. These scripts were written to allow for flexible user input and data output, easily customized interfaces and simple configuration. After the scripts verify and translate the specifications, a series of jobs (representing one request) are submitted to a server for rendering. After the servers have finished rendering each source, the data is packaged into a gzipped tar archive and staged for FTP retrieval. The user is then notified via email that their request has been fulfilled and the working directory (unique to each user) is removed. (The FTP staging area is also automatically cleaned up on a regular basis.)

The rendering servers are located at the CXC and are self-managed by a *PBS* server which schedules and preempts the simulations based upon priorities and a complex set of interdependencies. The use of *PBS* automatically makes

---

[2] `http://www.apache.org`

[3] `http://www.openpbs.org`

[4] `http://www.cpan.org`

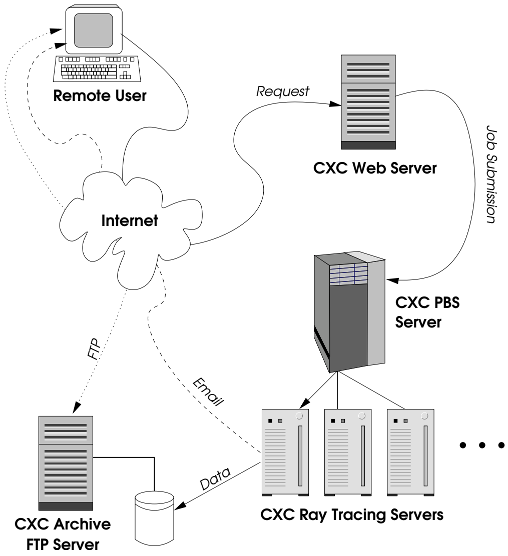

Figure 1. The software and hardware architecture of *ChaRT*.

the ray tracing system a distributed system, but it also allows the network of rendering machines to be easily scaled to suit future needs.

We installed numerous checks and limits which are applied to everything from the users of the system to the actual simulation jobs. To keep the system responsive, upper bounds have been placed on the number of simultaneous users as well as the number of sources specifiable per simulation. To protect the system from being monopolized, we limit the number of requests a single user may run at once, as well as the maximum run time of each simulation. Additionally, *ChaRT* will be publicly accessible and we check all inputs to ensure no malicious commands are attempted. These limits and checks are easily configurable and will be adjusted in the future based on users requirements and usage.

2.2. Software and Hardware Requirements

The *ChaRT* system is mostly comprised of Free Software components including Perl (and its CGI module), Bourne Shell, Apache and GNU `gzip` and `tar`. The freely available *PBS* batch system[5] is used to manage the rendering network and, of course, *SAOsac* is used to run the simulations.

While *ChaRT* could perform on a single machine, it is currently implemented such that the user interface (Perl scripts) are run along with Apache on a server dedicated to web tasks while *PBS* and the rendering jobs are run on a separate set of dedicated computers. (See Figure 1.)

[5]The license used by *PBS* is too restrictive for it be considered truly free software. It is still, however, freely available and redistributable.

3. Data Products

Our initial tests show that simulation runtimes average between 1 and 20 minutes, during which time no user interaction is required. When the simulation has completed, users will be notified by email that their data are ready for retrieval. This email includes a complete listing of the request (for user's archival purposes) as well as all information needed to retrieve the data from the CXC's public FTP server, CDA[6].

As an auxiliary data product, we record a log of every request made of the system. With this, we will be able to monitor trends in the usage of *ChaRT* and tune the system to better suit the needs of our users.

When users retrieve their data, they'll receive a package in gzipped tar format. This package will contain a directory hierarchy which expands to one directory per source, each containing one FITS file for each simulation of that source. Each FITS file includes a list of rays which can be later projected to the detector planes using the *CIAO* tool `psf_project_ray` to obtain 2-D model HRMA PSF images. Furthermore, the *ChaRT* ray files can be input in the `MARX` simulator (Wise et al. 1997) to generate realistic Chandra PSFs which also include the instrumental effects.

4. Future Work

Currently, users are only able to specify point sources. In the future, we will make it possible for users to run simulations with more complicated source specifications, including the use of images as source definitions. We would also like to facilitate on-the-fly generation of spectra.

Additionally, we would like to adapt the *ChaRT* system design to similar applications which require either the use of remote software or substantial computing resources.

Acknowledgments. This project is supported in part by the Chandra X-ray Center under NASA contract NAS8-39073.

References

Jerius, D. et al. 1995, in ASP Conf. Ser., Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes (San Francisco: ASP)

Karovska, M. et al. 2000, in ASP Conf. Ser., Vol. 238, Astronomical Data Analysis Software and Systems X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP)

Wise, M. et al. 1997, in ASP Conf. Ser., Vol. 125, Astronomical Data Analysis Software and Systems VI, ed. G. Hunt & H. E. Payne (San Francisco: ASP)

---

[6]`ftp://cda.cfa.harvard.edu`