

## ОБРАБОТКА ДАННЫХ СОЛНЕЧНЫХ ТЕЛЕСКОПОВ В ПРОЦЕССЕ НАБЛЮДЕНИЙ

В.Е. Томин, Д.Ю. Колобов, А.В. Киселев

### SOLAR TELESCOPE DATA PROCESSING DURING OBSERVATIONS

V.E. Tomin, D.Y. Kolobov, A.V. Kiselev

Анализ данных астрофизического эксперимента тесно связан с рядом технических задач, одна из которых – обработка данных в реальном времени – является особо актуальной для современных солнечных телескопов. Предварительный анализ данных позволяет своевременно определить необходимость корректировки хода эксперимента и более эффективно использовать наблюдательное время. При этом успешность последующего полноценного научного анализа во многом зависит от доступности исходных (сырых) данных эксперимента. Обсуждается решение, где основным подходом является конвейерное продвижение данных между этапами эксперимента. Это одновременно позволяет производить редуцированную обработку и сохранение исходных данных, осуществлять просмотр обработанных данных и их предварительный анализ в реальном времени.

Analysis of astrophysical experiment data is associated with a number of technical tasks. Online data processing is one of such tasks, which is most important for modern solar telescopes. Preliminary data analysis allows us to opportunely determine the need for experiment adjustment and to increase the efficiency of the time usage available for observations. At the same time, the success of the complete scientific analysis depends on the availability of the original (raw) experiment data. We discuss the solution in which the main approach is the pipelined data handling among experiment stages. This allows us to simultaneously conduct the data reduction and raw data storing, to preview the processed data and to perform preliminary data analysis in real time.

Анализ данных астрофизического инструмента тесно связан с рядом сложных технических задач. Одной из актуальных задач является обработка данных солнечных телескопов в реальном времени. Оценить приемлемость результата эксперимента можно разными способами. Традиционно опытный наблюдатель выполняет оценку по визуальному анализу исходных спектрограмм. При таком подходе результат напрямую определяется опытом наблюдателя. Другой подход заключается в проведении полного цикла обработки данных и получении физических величин и их производных прямо в ходе эксперимента. Существенную сложность к этому добавляют временные ограничения: наблюдатель должен видеть результат в режиме реального времени. Последнее подразумевает, что обработка данных производится с некоторыми компромиссами, которые, однако, не должны влиять на качественную оценку результатов. Такое решение позволит более эффективно использовать наблюдательное время и снизит влияние наблюдателя на качество получаемых данных.

Получение и обработка данных в общем случае представляют собой две независимых операции. Тем не менее существует ряд обстоятельств, которые делают эти операции тесно связанными. Можно выделить два типа обработки, которые предшествуют научному анализу. Первый тип – это редуцированная обработка, которая затрагивает изображение целиком, не зависит от его содержимого (спектральных линий, участков солнечной поверхности), но зависит от используемой оптической схемы. Ее результатом являются те же физические величины, что и в исходных данных (интенсивность). Применение редуцированной обработки позволяет избавиться от избыточной, с точки зрения физического эксперимента, информации. Другой тип будем называть предварительной обработкой. Ее результатом являются величины, описывающие поведение конкрет-

ного участка спектра (смещение и ширину линии, интенсивность в крыле линии). Сигналы, соответствующие физическим характеристикам солнечной плазмы, формируются непосредственно из этих данных.

Необходимость применения редуцированной обработки во время эксперимента определяется его типом. Для одних достаточно записать и показать пользователю сырые данные, другие можно провести только с использованием редуцированной обработки. Становится возможным проведение экспериментов с получением большого потока данных с последующим суммированием и записью на жесткий диск ЭВМ. В отдельных случаях редуцированная обработка может требоваться для улучшения визуального восприятия изображений спектра наблюдателем. Для этого может использоваться коррекция за плоское поле и учет темнового тока. Для последующего научного анализа необходимо сохранять оригинальные данные эксперимента (сырые данные, либо данные, прошедшие редуцированную обработку без потерь). Таким образом, с технической точки зрения операции обработки и записи данных должны быть полностью независимыми, то есть должна быть обеспечена возможность их одновременного выполнения.

Одно из решений заключается в том, что алгоритм конкретного эксперимента строится из более простых алгоритмов. В качестве последних можно выделить следующие типовые операции: получение данных, их сохранение и преобразование. Конечная задача – получение из спектров некоторой физической информации, ее сохранение и представление ее пользователю – представляется также рядом более простых алгоритмов, которые зависят от конкретного эксперимента. С точки зрения программного обеспечения ряд типовых алгоритмов можно представить в виде независимых друг от друга модулей, при этом передача данных от модуля к модулю выпол-

няется единым способом. Предварительная обработка может быть реализована аналогичным образом. Это позволит использовать единый механизм передачи исходных данных на обработку и результата обработки в модуль, осуществляющий их запись. Внутри модуля предварительной обработки отдельные операции (например вычисление V-параметра Стокса) можно реализовать в отдельных загружаемых модулях. Алгоритмы предварительной обработки должны быть реализованы таким образом, чтобы была возможна возможность использования данных разных размерностей, а также разных типов.

Алгоритмы редуцированной обработки можно представить в виде последовательности простых операций, где промежуточные результаты хранятся в отдельных кольцевых буферах. Хранение промежуточных результатов упрощает контроль выполнения алгоритма и его отладку, а также позволяет реализовать ветвящуюся структуру параллельных вычислений. Такую архитектуру можно назвать конвейером, где происходит преобразование данных в процессе продвижения их от буфера к буферу.

Для продвижения данных в конвейере используется распространенный шаблон проектирования производитель–потребитель (producer–consumer). Для записи данных в буфер из различных частей программы используется специальный класс объектов «Производитель данных» (Producer). С его помощью производится запись данных в свободную ячейку кольцевого буфера. Для выполнения преобразований над данными используются специальные программные модули – клиенты. Набор клиентов для обработки данных конкретного буфера находится в отдельном классе объектов «Потребитель данных» (Consumer). Объекты потребители работают в собственных потоках для исключения блокировок буферов длительными вычислениями, производимыми в клиентах. Потребители также решают задачу извлечения данных из кольцевого буфера и передачи их набору клиентов. Этот подход позволяет добиться одновременного получения, обработки и сохранения данных. Структура конвейера определяется в конфигурационном файле – ее изменение не требует перекомпиляции приложения. Таким образом, каждому алгоритму получения и обработки данных эксперимента соответствует конфигурация конвейера.

Этап подготовки данных к научному анализу, который следует за предварительной обработкой, обычно выполняется исследователем интерактивно. Во время наблюдений на аналогичном этапе должны использоваться те же самые алгоритмы, что и при научном анализе, с той лишь разницей, что вмешательство пользователя не требуется. Решение может быть основано на использовании типичных шаблонов проводимого эксперимента: измерение магнитного поля [Кобанов, 2001] с построением магнитограммы; измерение лучевых скоростей с построением двумерной карты; регистрация и анализ изменений этих величин со временем для одной пространственной точки или для всей пространственной области и др. Инструментом, используемым для реализации алгоритмов подобного типа, является язык Python с интерактивным интерпретатором IPython

[<http://ipython.org/>] и пакетами для обработки научных данных Numpy [<http://numpy.org/>], Scipy [<http://scipy.org/>] и Matplotlib [<http://matplotlib.org/>].

Отдельные этапы обработки данных могут быть реализованы с помощью стороннего программного обеспечения (например «Корректор данных» на языке Python, который используется для расчета и учета кадра плоского поля). Это требует способа передачи данных между различными приложениями и/или ЭВМ. Поскольку ресурсоемкие операции по преобразованию данных не должны препятствовать получению и сохранению исходной информации, в решении предусмотрена возможность организации распределенных вычислений. Формат и размерность данных могут существенно меняться в зависимости от типа эксперимента. При обработке после наблюдений используется промежуточный файл формата HDF5, в котором сохраняется результат предварительной обработки для последующей загрузки в интерактивной среде анализа данных. Существуют различные способы передачи данных между элементами конвейера: один из них – использование разделяемой памяти. Он используется для построения конвейера в пределах одной ЭВМ. Другой подход – передача данных через локальную сеть.

Задача распределенных вычислений решается с помощью библиотеки ZeroMQ [<http://zeromq.org/>]. Здесь под этим понимается возможность использования для получения и обработки данных разных ЭВМ. Для каждого приложения реализуется интерфейс для отправки и приема данных через локальную сеть. Транспортный слой ZeroMQ позволяет передавать произвольные массивы данных. Сырые данные передаются в неизменном виде, а модель с их описанием из внутреннего представления сериализуется в текстовый формат JSON.

Для сложных экспериментов – например, для наблюдений с использованием многоволнового спектрополяриметра – необходимо создание дополнительного приложения – менеджера эксперимента, с помощью которого будет осуществляться настройка и управление отдельными узлами (камерами, алгоритмами коррекции данных) как одним целым.

На рисунке представлена типичная схема организации обработки данных во время наблюдений. На одной ЭВМ происходит получение данных с камер, их редуцированная обработка и запись в файл. Затем данные поступают на другую ЭВМ по сети, где производится коррекция за плоское поле и учет темного тока. Скорректированные данные представляются пользователю для контроля хода эксперимента и проходят обработку, результатом которой являются карты лучевых скоростей и магнитограммы.

Все программное обеспечение (ПО) является открытым и кроссплатформенным. Под этим подразумевается возможность сборки и использования ПО на распространенных программных платформах, таких как GNU/Linux, MS Windows. Текущую платформу определяет аппаратное обеспечение, необходимое для проведения эксперимента. Для разработки ПО для получения данных был выбран язык C++ и набор библиотек для построения графического интерфейса Qt4. Для обработки и представления результатов пользователю выбран интерпретируемый язык Python.



Типичная схема организации обработки данных во время наблюдений.

Представленный выше подход был опробован при наблюдениях на Горизонтальном автоматизированном солнечном телескопе Саянской солнечной обсерватории в 2013 г. В результате создан работоспособный прототип ПО, где в рамках единой концепции осуществлено одновременное получение данных, их редуцирующая обработка, сохранение и предварительный анализ.

Работа выполнена при частичной поддержке гранта РФФИ 12-02-33110 мол\_a\_вед и гранта Президента Российской Федерации МК-497.2012.2, Министерства образования и науки Российской Федерации ГК 14.518.11.7047, соглашения 8407.

## СПИСОК ЛИТЕРАТУРЫ

Кобанов Н.И., Макарич Д.В. Измерения дифференциальной лучевой скорости и продольного магнитного поля на Солнце с помощью пзс-фотоприемников. Часть II. Результаты первых наблюдений в безмодуляционном режиме // Приборы и техника эксперимента. 2001. № 4. С. 110–116.

IPython: Interactive computing // URL: <http://ipython.org/> (дата обращения 26.08.2013).

Matplotlib: python plotting // <http://matplotlib.org/> (дата обращения 26.08.2013).

Numpy // <http://numpy.org> (дата обращения 26.08.2013).

Scipy // <http://scipy.org> (дата обращения 26.08.2013).

The Simplest Way to Connect Pieces – zeromq // <http://zeromq.org> (дата обращения 26.08.2013).

*Институт солнечно-земной физики СО РАН, Иркутск, Россия*