

СЛУЖБА ОБМЕНА СООБЩЕНИЯМИ PyChannel В ПРОГРАММНОМ КОМПЛЕКСЕ Sonix+

*И. А. Морковников*¹, *Л. А. Трунтова*

Объединенный институт ядерных исследований, Дубна

Рассматривается разработка службы обмена сообщениями PyChannel для программного комплекса Sonix+, обеспечивающего управление измерительными установками на реакторе ИБР-2 в Лаборатории нейтронной физики им. И. М. Франка Объединенного института ядерных исследований.

Необходимость в этом определялась тем, что текущая реализация протокола межмодульного взаимодействия на основе базы данных Varman не может обеспечить обмен данными между модулями, запущенными на разных компьютерах и серверах, а также взаимодействие 64- и 32-разрядных программ.

Описывается адаптация PyChannel под потребности разных программ, входящих в состав программного комплекса, с целью создания унифицированной службы для расширения возможностей Sonix+.

We consider the development of the PyChannel messaging service for the Sonix+ software package, which provides control of the measuring facilities at the IBR-2 reactor at the Frank Laboratory of Neutron Physics of the Joint Institute for Nuclear Research.

The need for this was determined by the fact that the current implementation of the inter-module communication protocol based on the Varman database cannot provide data exchange between modules running on different computers and servers, as well as the interaction of 64- and 32-bit programs.

We describe the adaptation of PyChannel to the needs of different programs that are part of the software package in order to create a unified service to extend the capabilities of Sonix+.

PACS: 29.50.+v; 28.41.-i

ВВЕДЕНИЕ

Одно из направлений развития программного комплекса Sonix+ [1], обеспечивающего управление измерительными установками на реакторе ИБР-2, — создание сервисов для организации удобной работы пользователей [2].

В качестве таких сервисов были разработаны система удаленного наблюдения за ходом эксперимента WebSonix [3] и файловое хранилище для хранения результатов измерений со спектрометров ИБР-2 [4].

Взаимодействие WebSonix с модулями Sonix+ обеспечивало устройство CChannel. К сожалению, реализация устройства значительно ограничивала отзывчивость интерфейса WebSonix, что требовало либо усовершенствования CChannel, либо создания нового модуля.

¹E-mail: bl@nf.jinr.ru

У графического интерфейса пользователя Sonix+ есть ограничение на количество используемой оперативной памяти в 2 ГБ при отображении данных, накапливаемых во время измерения. Для преодоления данного ограничения потребовалось обеспечить взаимодействие между 64- и 32-разрядными приложениями программного комплекса.

Добавление интеграции с системами автоматической обработки и отложенной записи результатов измерений в файловое хранилище для спектрометров ИБР-2 потребовало создания дополнительного слоя обмена данными с модулями Sonix+.

Для решения перечисленных выше задач была разработана служба PyChannel.

РАЗРАБОТКА PyChannel ДЛЯ WebSonix

PyChannel разрабатывался как замена CChannel для оптимизации обмена данными между устройствами Sonix+ и WebSonix с целью уменьшения времени загрузки и обновления веб-страниц с состоянием измерения. Так, например, загрузка основной веб-страницы со всеми данными о состоянии измерения требовала 8 с.

Основной недостаток модуля CChannel — последовательная обработка запросов. Так как WebSonix всегда делает несколько запросов к устройствам Sonix+, это приводит к задержкам при обновлении информации о текущем измерении. Некоторые запросы, например, создание изображения с массивом спектральных распределений, могут выполняться несколько минут, на протяжении которых WebSonix не сможет получать информацию об устройствах с научной установки. Поэтому основным требованием к PyChannel стало параллельное выполнение запросов.

Кроме того, WebSonix — многопользовательский сервис, что приводит к постоянной отправке идентичных запросов от нескольких пользователей. Для предотвращения повторной обработки данных в таких ситуациях необходимо кэширование на стороне сервера.

Следует отметить, что изначально CChannel реализовывал для WebSonix механизм вызова удаленных процедур (Remote Procedure Call, RPC). В качестве дальнейшего развития было предложено создать связующее программное обеспечение, ориентированное на обработку сообщений (message-oriented middleware, MOM) [5].

Данный механизм позволяет реализовать следующие функции: маршрутизацию сообщений (что позволит распределять запросы от WebSonix между несколькими процессами на стороне Sonix+) и обработку сообщений при передаче (что позволит кэшировать запросы).

Для выполнения описанных задач необходимо специальное приложение, которое называется брокером сообщений. Существуют как уже готовые решения (Apache Kafka, RabbitMQ), так и специальные библиотеки для создания своих собственных брокеров сообщений, например, ZeroMQ [6], который позволяет адаптировать протокол передачи сообщений под решаемую задачу.

Так как PyChannel первоначально разрабатывался для веб-интерфейса Sonix+, нам прежде всего была необходима низкая задержка при передаче данных и дополнительная обработка передаваемых сообщений в брокере сообщений. При этом нет никакого смысла в долгом хранении сообщений, так как информация, скорее всего, потеряет свою актуальность. Кроме того, во время разработки PyChannel предполагалось, что может потребоваться модернизация логики обмена сообщениями (например, в некоторых случаях может потребоваться RPC). ZeroMQ подошел в качестве основы

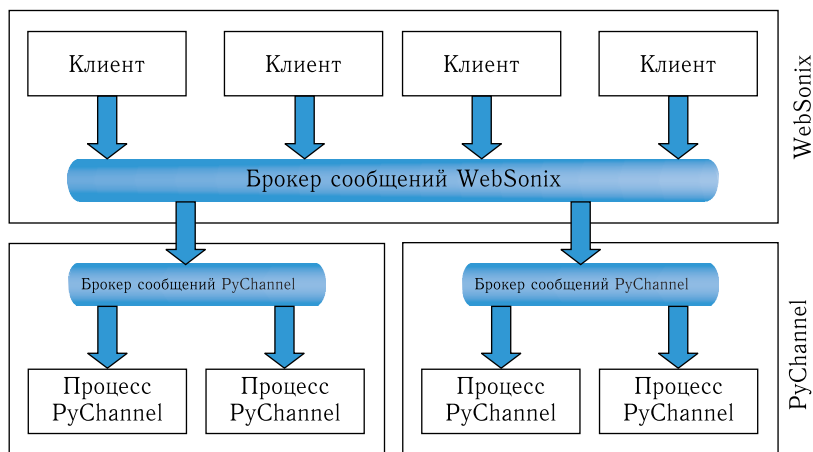


Рис. 1. Структура взаимодействия PyChannel с WebSonic

для PyChannel, так как обладает низкой задержкой при передаче сообщений и позволяет самостоятельно организовать логику обмена сообщениями.

В итоге WebSonic взаимодействует с устройствами Sonix+ через PyChannel по следующему принципу (рис. 1). На стороне WebSonic множество клиентов с разных научных установок создают запросы, которые проходят через брокер сообщений WebSonic, проверяющий наличие запроса в ассоциативном массиве, содержащем: запрос, список клиентов, сделавших запрос, время удаления записи из массива и результат выполнения запроса, если он уже выполнен. Если запрос уже выполнен и текущее время меньше или равно времени удаления записи, то возвращается результат из ассоциативного массива. Если текущее время больше времени удаления, то старая запись удаляется и создается новая. Если ожидается выполнение запроса, то идентификатор клиента для возврата результата запроса добавляется в ассоциативный массив. Если запрос необходимо выполнить, то брокер сообщений WebSonic направляет его на указанную научную установку. Далее брокер сообщений PyChannel распределяет запросы между своими процессами, которые запрашивают данные у устройств Sonix+ и проводят их обработку, после чего результат возвращается брокеру сообщений WebSonic, добавляет его в соответствующую запись в ассоциативном массиве и возвращает всем клиентам, сделавшим этот запрос.

Во время тестов замена CChannel на PyChannel значительно увеличила отзывчивость интерфейса, сократив время выполнения простых запросов WebSonic с 1 с до 6 мс, а сложных с 1 с до 731 мс. При этом время загрузки основной web-страницы сократилось с 8 до 2 с.

АДАПТАЦИЯ PyChannel ДЛЯ ОБМЕНА СООБЩЕНИЯМИ МЕЖДУ МОДУЛЯМИ Sonix+

В последние годы в ЛНФ ОИЯИ все более активно применяются DAQ-контроллеры [7] с форматом выходных данных в виде списка событий. Работа со списками событий вместо массивов спектральных распределений потребовала реорганизации программного комплекса Sonix+ [8].

Одно из основных преимуществ списка событий — построение спектральных распределений в высоком разрешении, что, однако, требует значительного количества оперативной памяти.

Программа для отображения данных SpectraViewer работает отдельно от Sonix+ в 64-разрядном режиме. Однако протокол межмодульного взаимодействия программного комплекса поддерживает работу только с 32-разрядными приложениями, что ограничивает количество используемой памяти до 2 ГБ [9]. Это уменьшает максимально возможное разрешение, в котором графический интерфейс пользователя (GUI) Sonix+ может отображать данные, накапливаемые во время измерения.

Для получения информации о текущем состоянии эксперимента и отправке команд через базу данных Varman с помощью протокола межмодульного взаимодействия из графического интерфейса Sonix+ и программы для просмотра спектров SpectraViewer было предложено транслировать запросы через PyChannel (рис. 2), что потребовало значительной доработки программы. Основные изменения заключаются в разделении PyChannel на отдельные модули и добавлении возможности работы с внешними программами для гибкого интегрирования приложений в Sonix+. Трансляцию запросов от 64-разрядных приложений обеспечивают три внешних приложения:

PyProtocolDll — организует работу с библиотекой protocol_dll, которая отвечает за запись в протокол Sonix+;

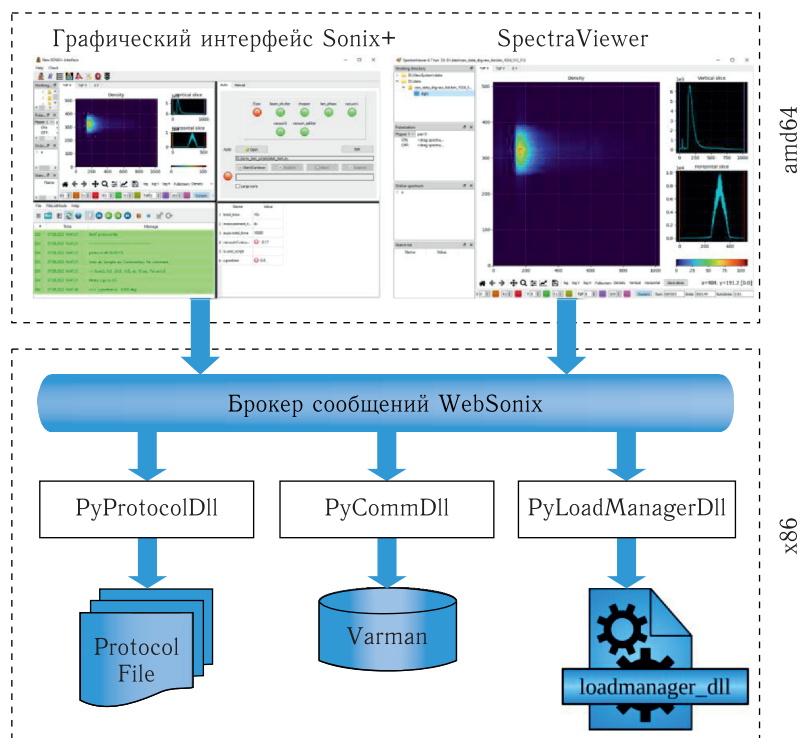


Рис. 2. Схема подключения 64-разрядного интерфейса к 32-разрядным управляющим модулям Sonix+ через PyChannel

PyCommDll — обеспечивает отправку команд и получение информации из базы данных Varman;

PyLoadManagerDll — позволяет получать сигналы от устройств Sonix+ об окончании загрузки с помощью библиотеки loadmanager_dll.

При запуске приложение проверяет доступность Sonix+ и свою разрядность. В случае если Sonix+ недоступен, SpectraViewer запускается в offline режиме (без взаимодействия с Sonix+), а графический интерфейс Sonix+ завершает свою работу с ошибкой. Далее, если Sonix+ работает, то программы проверяют свою разрядность. В 64-разрядном режиме программы связываются с Sonix+ через PyChannel, а в 32-разрядном режиме подключение осуществляется через протокол межмодульного взаимодействия программного комплекса.

Необходимо отметить, что данная реализация PyChannel добавляет дополнительные промежуточные модули для связи с программным комплексом, что приводит к увеличению времени обмена данными.

ПРОБЛЕМЫ ИСПОЛЬЗОВАНИЯ ФАЙЛОВОГО ХРАНИЛИЩА ДЛЯ СПЕКТРОМЕТРОВ ИБР-2 И ИХ РЕШЕНИЕ С ПОМОЩЬЮ PyChannel

При эксплуатации центрального файлового хранилища для спектрометров ИБР-2 (далее хранилище) был выявлен ряд недочетов, которые значительно затрудняют его эксплуатацию:

- в хранилище пишутся не только данные с результатами измерений, но и результаты тестовых запусков Sonix+;
- в случае потери связи с хранилищем пользовательские файлы нужно записывать вручную;
- у пользователей отсутствует возможность создания систем автоматической обработки данных из хранилища.

Для решения данных задач необходимо организовать дополнительный слой обмена данными между устройствами Sonix+ и хранилищем пользовательских файлов, который будет:

- маркировать измерения по циклам реактора или, если измерение было проведено во время перерыва, пометить его как тестовое;
- вести реестр измерений на управляющем компьютере и автоматически дописывать данные, которые были получены во время потери связи с хранилищем;
- уведомлять пользовательские системы автоматической обработки данных о поступлении новой информации;
- обеспечивать запись в хранилище данных, полученных в результате обработки пользователями;
- вести реестр и индексировать все данные, записанные в хранилище, для их дальнейшего быстрого поиска по параметрам измерений.

Для решения данных задач потребовалось организовать набор программ, связанных друг с другом, для управления данными как на сервере (модуль хранилища), так и на управляющем компьютере (модуль хранилища и загрузчик в хранилище). PyChannel может обеспечить обмен сообщениями между модулями, однако Sonix+ работает исключительно под ОС Windows, а хранилище организовано на ОС

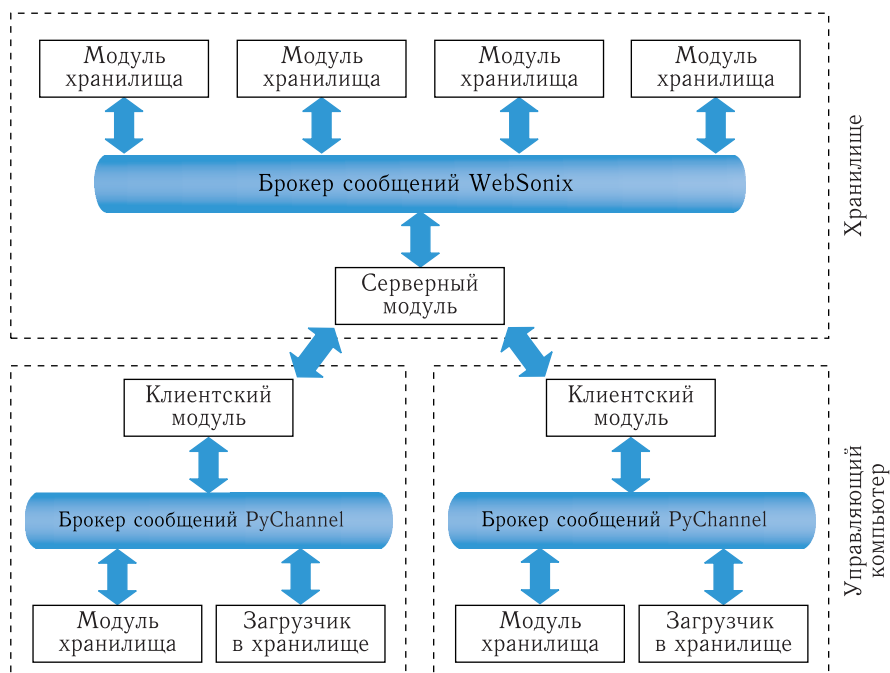


Рис. 3. Структура PyChannel для хранилища

GNU/Linux, что потребовало дополнительной адаптации для работы на сервере. Кроме того, потребовалось создать отдельные модули (серверный модуль и клиентский модуль) для организации связи между экземплярами PyChannel на разных физических устройствах (рис. 3).

В идеале даже перезагрузка или выключение Sonix+ не должно прерывать работу модулей, отвечающих за запись данных в хранилище, что решается с помощью оформления PyChannel в качестве службы Windows [10]. При таком подходе отправку данных сможет прервать только выключение или перезагрузка компьютера, управляющего экспериментом. Кроме того, система отправит сигнал о завершении работы и предоставит время для легитимного завершения работы службы, что является одним из главных условий для поддержания реестра измерений в рабочем состоянии.

ЗАКЛЮЧЕНИЕ

Предложена концепция службы для обмена сообщениями PyChannel. В процессе разработки принципы организации службы несколько раз модифицировались в соответствии с расширением списка решаемых задач.

Служба реализована и обеспечивает дополнительный уровень обмена данными между модулями Sonix+ в случае, когда протокол межмодульного взаимодействия программного комплекса неприменим.

Авторы выражают признательность В. И. Боднарчуку за поддержку работы и ценные замечания при подготовке данной рукописи.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования РФ (соглашение № 075-10-2021-115 от 13 октября 2021 г. (внутренний номер 15.СИН.21.0021)).

СПИСОК ЛИТЕРАТУРЫ

1. *Kirilov A. S.* Instrument Control Software at the IBR-2: Directions of Development. JINR Preprint E10-2017-89. Dubna, 2017.
2. *Кирилов А. С., Юдин В. Е.* Реализация базы данных реального времени для управления экспериментом в среде MS Windows. Препринт ОИЯИ P13-2003-11. Дубна, 2003.
3. *Morkovnikov I. A., Kirilov A. S.* Upgrading WebSonix – Remote Instrument Control System Experiment at the IBR-2 Reactor // Proc. of XXIV Intern. Symp. “Nuclear Electronics & Computing” (NEC’2013), Varna, Bulgaria, Sept. 9–16, 2013. Dubna, 2013. P. 181–185.
4. *Кирилов А. С., Морковников И. А.* О концепции файлового хранилища для спектрометров ИЯУ ИБР-2. Препринт ОИЯИ P10-2018-23. Дубна, 2018.
5. *Curry E.* Message-Oriented Middleware // Middleware for Communications. John Wiley & Sons, Ltd, 2005. P. 1–28.
6. <https://zeromq.org>
7. *Кирилов А. С., Мурашкевич С. М.* Адаптация программного комплекса Sonix+ для работы с DAQ-контроллерами DeLiDAQ-2 и диджитайзером N6730 фирмы CAEN // Письма в ЭЧАЯ. 2023. Т. 20, № 2(247). С. 127–135.
8. *Кирилов А. С. и др.* Реорганизация программного комплекса Sonix+ для работы с данными в виде списка событий. Препринт ОИЯИ P10-2023-14. Дубна, 2023.
9. <https://learn.microsoft.com/ru-ru/windows/win32/memory/memory-limits-for-windows-releases>
10. [https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc783643\(v=ws.10\)?redirectedfrom=msdn#services-overview-1](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc783643(v=ws.10)?redirectedfrom=msdn#services-overview-1)

Получено 19 июня 2023 г.