

P10-2023-27

И. А. Морковников, Л. А. Трунтова

СЛУЖБА ОБМЕНА СООБЩЕНИЯМИ PyChannel  
В ПРОГРАММНОМ КОМПЛЕКСЕ Sonix+

Направлено в журнал «Письма в ЭЧАЯ»

Морковников И. А., Трунтова Л. А.

P10-2023-27

Служба обмена сообщениями PyChannel  
в программном комплексе Sonix+

Рассматривается разработка службы обмена сообщениями PyChannel для программного комплекса Sonix+, обеспечивающего управление измерительными установками на реакторе ИБР-2 в ЛНФ ОИЯИ.

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

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

Работа выполнена в Лаборатории нейтронной физики им. И. М. Франка ОИЯИ.

Препринт Объединенного института ядерных исследований. Дубна, 2023

Morkovnikov I. A., Truntova L. A.

P10-2023-27

PyChannel Messaging Service in 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 FLNP JINR.

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 with different operating systems, as well as the interaction of 64-bit and 32-bit programs.

The article describes 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+.

The investigation has been performed at the Frank Laboratory of Neutron Physics, JINR.

Preprint of the Joint Institute for Nuclear Research. Dubna, 2023

## **ВВЕДЕНИЕ**

Одно из направлений развития программного комплекса Sonix+ — создание сервисов для организации удобной работы пользователей [1, 2].

В качестве таких сервисов были разработаны система удаленного наблюдения за ходом эксперимента WebSonix [3] и центральное хранилище данных ЛНФ [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.

Для увеличения максимального возможного разрешения, в котором графический интерфейс пользователя Sonix+ может отображать данные, накапливаемые во время измерения, потребовалось обеспечить взаимодействие между 64- и 32-разрядными приложениями программного комплекса.

Добавление интеграции с системами автоматической обработки и отложенной записи результатов измерений в центральное хранилище данных ЛНФ потребовало создания дополнительного слоя обмена данными с модулями Sonix+.

Для решения перечисленных выше задач разработана служба PyChannel.

### **1. РАЗРАБОТКА PyChannel ДЛЯ WebSonix**

PyChannel разрабатывался как замена CChannel для оптимизации обмена данными между устройствами Sonix+ и WebSonix с целью уменьшения времени обновления web-страниц с состоянием измерения.

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 таких ситуаций необходимо кэширование на стороне сервера.

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

Наиболее интересные функции, которые позволяет реализовать данный механизм: маршрутизация сообщений (что позволит распределять запросы от WebSonix между несколькими процессами на стороне Sonix+) и обработка 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 и был взят за основу для реализации PyChannel.

В итоге WebSonix взаимодействует с устройствами Sonix+ через PyChannel по принципу, отраженному на рис. 1. На стороне WebSonix множество клиентов с разных научных установок создают запросы, которые проходят через приложение, кэширующее и направляющее их на указанные научные установки. Далее PyChannel распределяет запросы между своими процессами, которые запрашивают данные у устройств Sonix+ и проводят их обработку, после чего результат возвращается клиентам.

Во время тестов замена CChannel на PyChannel значительно увеличила отзывчивость интерфейса WebSonix.

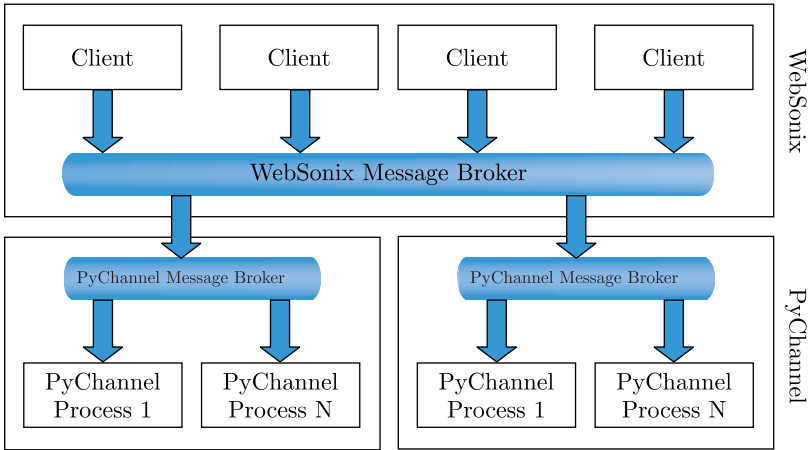


Рис. 1. Структура взаимодействия PyChannel с WebSonix

## 2. 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, которое, 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 максимальное возможное разрешение, в котором графический интерфейс пользователя (GUI) Sonix+ может отображать данные, накапливаемые во время измерения.

Для отправки команд из GUI в протокол межмодульного взаимодействия Sonix было предложено транслировать их 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+.

Необходимо отметить, что данная реализация PyChannel добавляет большой слой абстракции, который приводит к увеличению времени обмена данными.

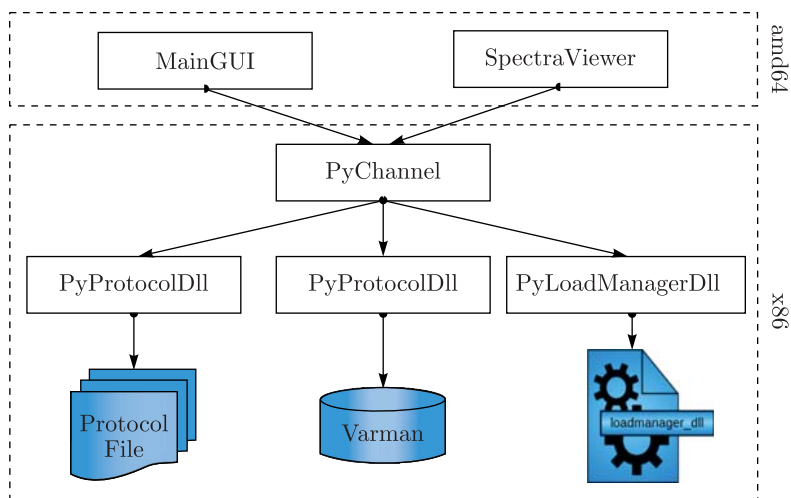


Рис. 2. Схема подключения 64-разрядного интерфейса к 32-разрядным управляющим модулям Sonix+ через PyChannel

## 1. ЦЕНТРАЛЬНОЕ ХРАНИЛИЩЕ ДАННЫХ ЛНФ

При эксплуатации центрального хранилища данных ЛНФ (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 Sopix+;
- 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 Sopix+ 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 решения данных задач необходимо организовать набор программ, связанных друг с другом, для управления данными как на сервере, так и на управляющем компьютере. PyChannel может обеспечить обмен со-

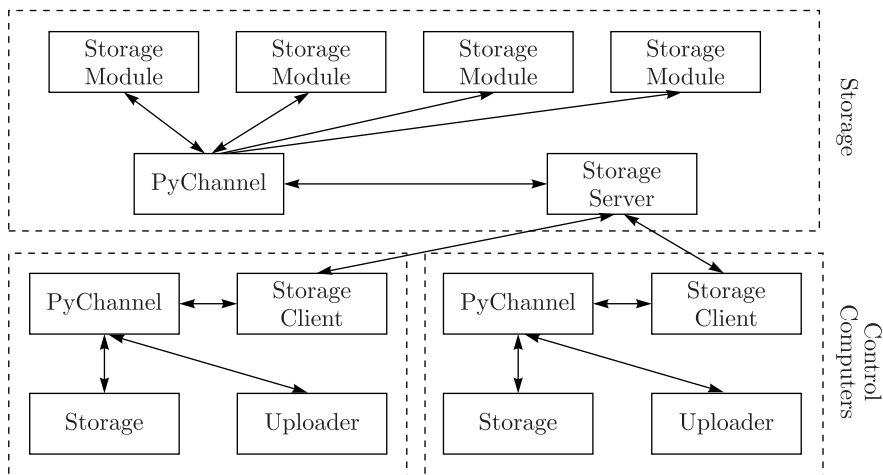


Рис. 3. Структура PyChannel для хранилища

общениями между приложениями, однако для этого его потребовалось адаптировать для работы на сервере и создать отдельный 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. 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

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

На момент написания статьи 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.

## 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 on 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 / Ed. by Q. H. Mahmoud. 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 Soplх+ для работы с данными в виде списка событий. Препринт ОИЯИ Р10-2023-14. 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 1 июня 2023 г.



Редактор *М. И. Зарубина*

Подписано в печать 20.09.2023.

Формат 60 × 90/16. Бумага офсетная. Печать цифровая.

Усл. печ. л. 0,75. Уч.-изд. л. 0,60. Тираж 125 экз. Заказ № 60731.

Издательский отдел Объединенного института ядерных исследований  
141980, г. Дубна, Московская обл., ул. Жолио-Кюри, 6.

E-mail: [publish@jinr.ru](mailto:publish@jinr.ru)

[www.jinr.ru/publish/](http://www.jinr.ru/publish/)