

Программный комплекс для распределённой системы управления TANGO

А. С. Бондяков, А. О. Кондратьев*

Международная межправительственная организация Объединенный институт ядерных исследований, г. Дубна, Российская Федерация

Адрес: 141980, Российская Федерация, г. Дубна, Московская область, ул. Жолио-Кюри, д. 6

* kondratyev@jinr.ru

Аннотация

В настоящее время, научные исследования в области физики высоких энергий предполагают использование ускорительных комплексов различного уровня сложности, экспериментальных установок, а также значительных вычислительных ресурсов. Для эффективного управления этими комплексами используются специализированные системы, представляющие собой объектно-ориентированные распределенные системы управления аппаратным оборудованием. В данной статье представлен разработанный программный комплекс для распределенной системы управления *Tango Controls*, позволяющий получать, обрабатывать данные, полученные с сервера устройств *Tango Controls*, а также отображать полученные данные с помощью веб-интерфейса в виде графиков и данных телеметрии. Безусловным достоинством *Tango Controls* являются его кроссплатформенность, открытый исходный код, а также универсальный инструментарий, что позволяет использовать *Tango Controls* в широком диапазоне аппаратных решений. Распределенная система *Tango Controls* используется для создания систем управления аппаратными ресурсами. Доступ к аппаратным ресурсам осуществляется посредством распределенного объекта *Tango Controls*. Распределенный объект в *Tango Controls* называется устройством и создается как объект в процессе-контейнере, называемом сервером устройств. Сервер устройств реализует сетевое взаимодействие и связывается с базой данных конфигурации и клиентами. В процессе работы сервер устройств создает экземпляры устройств, отображающие логические сущности компонентов оборудования. Для взаимодействия клиентов с серверами устройств используется *Tango* протокол. Серверы и клиенты устройств *Tango* могут быть написаны на *Python*, *C++* или *Java*. *Tango* поставляется с полным набором инструментов для разработки, контроля, мониторинга и т.д. Разработанный программный комплекс является масштабируемым, и протестирован на отказоустойчивость и безопасность.

Ключевые слова: программный комплекс, сервер устройств, распределенная система управления

Авторы заявляют об отсутствии конфликта интересов.

Для цитирования: Бондяков А. С., Кондратьев А. О. Программный комплекс для распределённой системы управления TANGO // Современные информационные технологии и ИТ-образование. 2022. Т. 18, № 1. С. 47-53. doi: <https://doi.org/10.25559/SITITO.18.202201.47-53>

© Бондяков А. С., Кондратьев А. О., 2022



Контент доступен под лицензией Creative Commons Attribution 4.0 License.
The content is available under Creative Commons Attribution 4.0 License.



Software Complex for Distributed Control System TANGO

A. S. Bondyakov, A. O. Kondratyev*

Joint Institute for Nuclear Research, Dubna, Russian Federation

Address: 6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation

* kondratyev@jinr.ru

Abstract

Today, any research in the field of high energy physics involves the use of the most complex accelerator complexes, experimental facilities, as well as huge computing resources. Therefore, to control these complexes, specialized systems are used, such as Tango Controls and WinCC OA, which are object-oriented distributed control systems for hardware equipment. To solve the problem, the Tango Controls system was chosen. It has a hardware-independent toolkit that allows you to integrate Tango Controls into any hardware. Also, the system has a lot of both desktop and Web applications, which greatly simplifies development. Tango Controls is an open source distributed system that can run on one or hundreds of machines at the same time. It uses a client-server interaction model, as well as an omniorb implementation of CORBA as a network protocol. CORBA technology is a software mechanism that integrates isolated systems and allows programs written in different programming languages and running in different network nodes to interact with each other. The distributed Tango Controls system is used primarily to provide access to hardware resources. Access is programmed in processes called Device Servers, which implement the classes that provide access to the resource. During operation, the server creates device instances that represent the logical entities of the hardware components. Tango protocol is used for interaction between clients and device servers. This article presents the developed software complex for the distributed control system Tango Controls, which allows us to read, process data received from the Server of the Tango Controls device, and also display the received data on the Web interface in the form of graphs and telemetry data. The developed software complex is scalable, fault-tolerant and tested for performance.

Keywords: software complex, device server, distributed control system

The authors declare no conflict of interest.

For citation: Bondyakov A.S., Kondratyev A.O. Software Complex for Distributed Control System TANGO. *Sovremennye informacionnye tehnologii i IT-obrazovanie = Modern Information Technologies and IT-Education*. 2022; 18(1):47-53. doi: <https://doi.org/10.25559/SITITO.18.202201.47-53>



Введение

На сегодняшний день, различные научные исследования в области физики высоких энергий и биофизики, микробиологии и медицины, и пр., предполагают использование различных комплексов ускорителей элементарных частиц, включающих в себя миллионы сверхчувствительных детекторов, которые способны улавливать мельчайшие частицы. Для обработки, хранения и анализа данных, полученных в ходе экспериментов, необходимы значительные вычислительные мощности.

Одним из таких ускорительных комплексов является Большой адронный коллайдер (БАК), расположенный в ЦЕРН (*European Organization of Nuclear Research* – англ.), представляющий собой ускоритель длиной более 26 километров, с расположенными на нём четырьмя основными детекторами: *ATLAS*, *ALICE*, *CMS*, *LHCb*, а также тремя вспомогательными: *TOTEM*, *LHCf*, *MoEDAL*. Для управления, хранения и обработки данных, получаемых с ускорителя БАК и детекторов, используется распределенная вычислительная сеть *LCG* (*LHC Computing Grid* – англ.), использующая технологию ГРИД [1-3], которая позволяет объединить в единую вычислительную среду ресурсы, расположенные в разных точках земного шара¹.

Также, ярким примером развития комплексов ускорителей элементарных частиц является *NICA* (*Nuclotron based Ion Collider Facility* – англ.), строящийся в Объединенном институте ядерных исследований [4]. Комплекс включает в себя детекторы: *MPD* (*Multi-Purpose Detector* – англ.) – предназначенный для проведения экспериментов в области ядерной физики при столкновении пучков ядер тяжелых элементов², *SPD* (*Spin Physics Detector* – англ.) – предназначенный для экспериментов по физике спина при столкновении пучков ядер легких элементов [5], а также *BM@N* (*Baryonic Matter at Nuclotron* – англ.) – предназначенный для изучения взаимодействия пучков тяжелых ионов с фиксированными мишенями [6].

В свою очередь, Многофункциональный информационно-вычислительный комплекс Объединенного института ядерных исследований (МИВК ОИЯИ) [7-23], расположенный в Лаборатории информационных технологий имени М. Г. Мещерякова, обеспечивает высокую производительность, развитую систему хранения данных, высокую надёжность и доступность, а также масштабируемость и развитую программную среду для различных групп пользователей. МИВК ОИЯИ позволяет обрабатывать и хранить данные, полученные в ходе экспериментов.

Для управления узлами вышеуказанных ускорительных комплексов, а также для того, чтобы полученные в ходе экспериментов данные поступали, обрабатывались и хранились на вычислительных ресурсах, необходимо иметь возможность программного доступа к аппаратным ресурсам данных комплексов.

Наиболее популярными программными решениями в этой об-

ласти являются объектно-ориентированные системы *Simatic WinCC OA* (*Windows Control Panel Open Architecture* – англ.)³ и *Tango* (*TACO Next Generation Objects* – англ.) *Controls*⁴ [24].

WinCC OA – проприетарное программное обеспечение, разработанное для применения в сложных проектах со специфическими требованиями к архитектуре, а также для решения индивидуальных задач. *WinCC OA* имеет архитектуру клиент-сервер и построено по модульному принципу, т.е. функционально разделено на драйверы (программные процессы), которые, в свою очередь, могут быть распределены по аппаратным серверам в сети. Обмен данными между драйверами происходит посредством использования протокола *TCP/IP*. *WinCC OA* является кроссплатформенным программным обеспечением и может быть установлено на операционные системы *Windows* и *Linux*, а также платформу виртуализации *VMware ESXi*.

Управление проектами в *WinCC OA* основано на объектно-ориентированном подходе. Объекты представляются в виде точек данных, характеризующих образ конкретного физического устройства или процесса. Написание пользовательских сценариев (скриптов) происходит на языке *CONTROL++*, синтаксис которого подобен языку программирования *C/C++*. Такие скрипты могут являться как обработчиками событий, так и представлять собой процедуры обработки данных. В состав данного программного обеспечения также входят клиентские приложения для персональных компьютеров и мобильных устройств, а также множество пакетов расширений.

Tango Controls – представляет собой объектно-ориентированную систему с открытым исходным кодом, разработанную для управления экспериментальными установками, ускорителями, а также различным программным обеспечением и оборудованием. *Tango* является распределенной системой управления, что позволяет ей использовать для работы несколько аппаратных серверов одновременно. В качестве сетевого протокола система использует *omniORB* реализацию *CORBA* [25], представляющую собой технологический стандарт написания распределенных приложений, а также обеспечивающий полное взаимодействие систем управления аппаратными ресурсами между собой вне зависимости от операционной системы, под управлением которой работает она работает.

Также как и в *WinCC OA*, *Tango* основана на принципах объектно-ориентированного подхода. Ее модель поддерживает атрибуты, методы и свойства. Однако, ключевым отличием *Tango Controls* от *WinCC OA*, которая рассматривает систему управления как набор сигналов, считывает и записывает значения процессов, является концепция устройств (*device* – англ.) и классов устройств (*device class* – англ.). Устройства создаются серверами устройств (*device server* – англ.), процессами, реализующими классы устройств, которые, в свою очередь, отвечают за преобразование аппаратных протоколов связи в протоколы связи *Tango Controls*. Каждое устройство имеет со-

¹ Buncic P., Krzewicki M., Vande Vyvre P. Technical Design Report for the Upgrade of the Online-Offline Computing System. CERN-LHCC-2015-006, ALICE-TDR-019. CERN; 2015 [Электронный ресурс]. URL: <https://cds.cern.ch/record/2011297> (дата обращения: 20.01.2022).

² MPD Collaboration. The MultiPurpose Detector – MPD. Conceptual Design Report. Dubna: JINR, 2012. 259 p.

³ Соловьев С. Ю., Космин А. С. Построение распределенных систем сбора и обработки данных на базе платформы SIMATIC WinCC Open Architecture [Электронный ресурс] // «ИСУП» Информатизация и системы управления в промышленности. 2015. № 4(58). С. 70-75. URL: <https://isup.ru/articles/2/7611> (дата обращения: 20.01.2022); SIMATIC WinCC Open Architecture Portal [Электронный ресурс] // ETM professional control GmbH, 2022. URL: <https://www.winccoa.com> (дата обращения: 20.01.2022).

⁴ Tango Controls [Электронный ресурс] // Tango Controls, 2022. URL: <https://www.tango-controls.org> (дата обращения: 20.01.2022).



стояние, ноль или более атрибутов. Таким образом, осуществляется управление таким оборудованием как двигатели, клапаны, осциллографы и т. д.

Tango была разработана для управления как большими, так и малыми системами. Каждая система имеет централизованную базу данных, действующую как постоянное хранилище динамических настроек, в которой хранятся данные конфигурации, используемые при запуске сервера устройств.

На сегодняшний день, *Tango* используется для управления не только ускорителями, но и экспериментальными лазерами, аэродинамическими трубами, а совсем недавно он был принят крупнейшим в мире радиотелескопом в качестве основной системы управления.

Tango Controls и *WinCC OA* эффективно справляются со своими задачами и обладают похожими возможностями, но нужно отметить, в первую очередь, что *WinCC OA* является проприетарным программным обеспечением, в то время как *Tango Controls* представляет собой систему с открытым исходным кодом, что, в конечном итоге, и предопределило выбор в пользу *Tango Controls*.

Цель Исследования

Цель исследования – разработка программного комплекса для распределённой системы управления *TANGO*, позволяющего

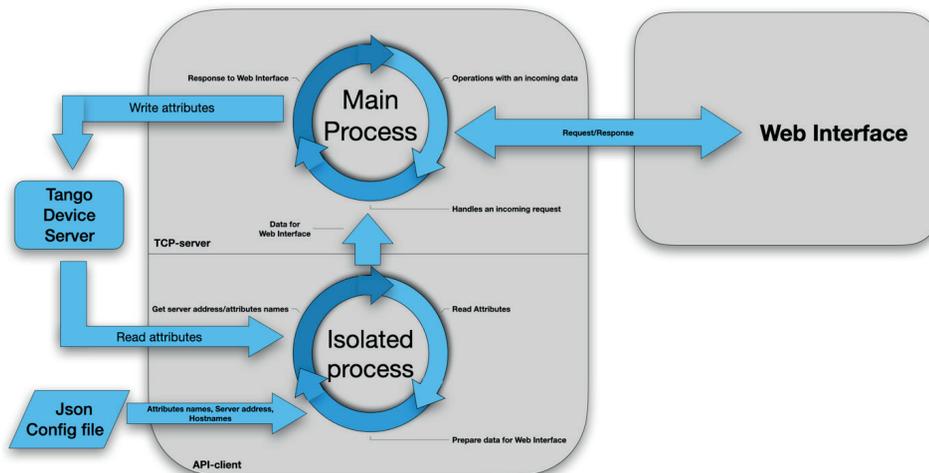
получать, обрабатывать, а также отображать на веб-интерфейсе в виде графиков и телеметрии данные, полученные с аппаратного оборудования. Протестировать разработанный программный комплект на отказоустойчивость.

Основная часть

Разработанный программный комплекс представляет собой набор взаимодействующих между собой компонентов:

- *Tango* клиент, осуществляющий получение и обработку, а также передачу данных серверу устройств *Tango*;
- Веб-интерфейс, отображающий данные, полученные с сервера устройств *Tango* клиентом, в виде графиков и данных телеметрии, с возможностью изменения значений некоторых атрибутов сервера устройств *Tango*.
- TCP-сервер, реализованный на языке высокого уровня C++, являющийся серверной частью веб-интерфейса и обрабатывающий все входящие запросы.
- Конфигурационный файл в формате *JSON*, в котором хранятся имена атрибутов, необходимых для получения данных, а также адрес для подключения к серверу устройств *Tango*.

Принципиальная схема работы программного комплекса представлена на Рис. 1.



Р и с. 1. Схема работы разработанного программного комплекса

Fig. 1. Scheme of work of the developed software complex

Алгоритм взаимодействия компонентов (составных частей) программного комплекса реализован следующим образом:

Tango Клиент

Каждые 500 миллисекунд, в изолированном от основного процесса, *Tango* клиент получает из конфигурационного файла адрес для подключения к серверу устройств *Tango*, а также имена атрибутов, необходимых для считывания. Это реализовано для того, чтобы иметь возможность изменять адрес подключения и списки атрибутов для считывания без необходи-

мости отключения и переконфигурирования клиента. Далее происходит попытка подключения к серверу устройств *Tango* по заданному ранее адресу, а также попытка считать указанные атрибуты. Если, по каким-то причинам, не удаётся считать тот или иной атрибут, его значение становится равным -1. Затем, из считанных атрибутов формируется объект типа *JSON* для отправки на веб-интерфейс.

Благодаря тому, что процесс, в котором работает клиент изолирован от основного, клиент может получать данные с большого количества серверов устройств *Tango* одновременно.



TCP-сервер

TCP-сервер разработан на языке высокого уровня C++, реализован с использованием интерфейса неблокирующего сокета, запускается в главном процессе и является серверной частью веб-интерфейса.

Для организации безопасной передачи данных клиента TCP-серверу, была реализована следующая схема: каждый раз, когда клиент подключается к TCP-серверу и пытается передать запрос, выполняется проверка, имеет ли право *hostname*, который содержится в заголовке запроса, передавать данные на TCP-сервер. Если проверка не пройдена, то соединение для таких клиентов закрывается. Если же подключенный клиент имеет право на передачу данных серверу, то происходит считывание запроса и его обработка. Всего реализовано 3 типа запросов, которые обрабатывает TCP-сервер:

1. Запрос на получение данных от веб-интерфейса.

При получении данного запроса, TCP-сервер в ответ отправляет объект типа *JSON*, сформированный *Tango* клиентом в изолированном процессе, на веб-интерфейс. В данном объекте находятся актуальные данные, полученные с сервера устройств *Tango*.

2. Запрос на изменение атрибутов сервера устройств *Tango*.

Получая подобный запрос от веб-интерфейса, сервер вычлняет из запроса имена атрибутов, значения которых необходимо изменить, а также сами значения данных атрибутов. Затем происходит проверка соответствия имён атрибутов из запроса с именами атрибутов сервера устройств *Tango*, проверка возможности изменения данных атрибутов, а также совпадает ли тип данных для значений атрибутов из запроса и атрибутов сервера устройства *Tango*. После этого, вызывается метод отправки атрибутов на сервер устройств *Tango*.

3. Все остальные запросы.

Если входящий запрос не является запросом на получение данных от веб-интерфейса или запросом на изменение атрибутов сервера устройства *Tango*, TCP-сервер отправляет клиен-

ту текстовое сообщение об ошибке.

В случае, если по каким-либо причинам, после подключения, клиент не начал передавать запрос, он будет отключён по тайм ауту. Данное решение реализовано с помощью метода опроса сокетов – *poll*⁵.

Веб-интерфейс

Веб-интерфейс программного комплекса представлен в виде структурной схемы (Рис. 2), на которой расположены:

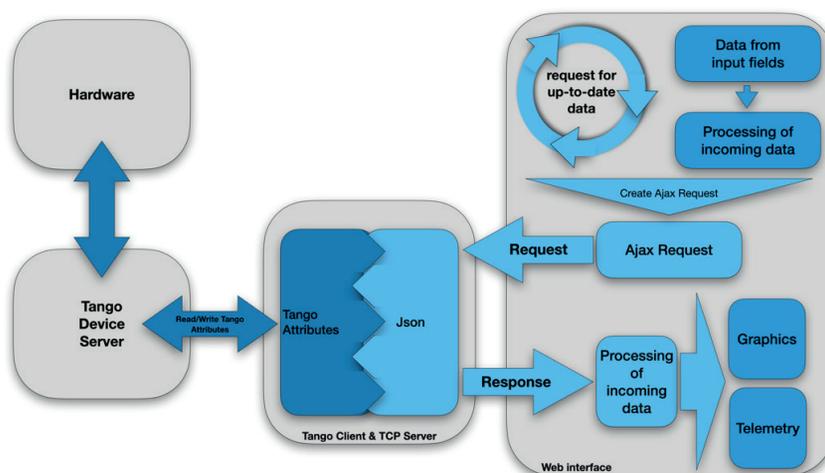
- графики для визуального отображения значений атрибутов, полученных с сервера устройств *Tango*;
- блок телеметрии;
- поля ввода для отправки новых значений атрибутов на сервер устройств *Tango*;

Каждую секунду на веб-интерфейсе автоматически формируется *ajax* запрос к TCP-серверу на получение актуальных значений атрибутов сервера устройств *Tango*. В случае неполучения ответа от TCP-сервера, тайм аут запроса составляет 300 миллисекунд.

Для изменения значений атрибутов сервера устройств *Tango*, значения из полей ввода, введённые вручную, также помещаются в *ajax* запрос и отправляются на TCP-сервер. В случае, если поле ввода осталось пустым, отправляется последнее актуальное значение. Если же данные в поле ввода, по тем или иным причинам, некорректны, отправка не происходит, и высвечивается сообщение об ошибке.

Для сброса значений атрибутов, создаётся *ajax* запрос на изменение атрибутов сервера устройств *Tango*, в котором значения соответствующих атрибутов равны 0. Далее, осуществляется отправка запроса на TCP-сервер.

При выполнении операций с аппаратным устройством, посредством веб-интерфейса, для исключения случайного нажатия кнопок, которое может привести к нестабильной работе сервера устройств *Tango*, реализован механизм подтверждения действия.



Р и с. 2. Структурная схема веб-интерфейса программного комплекса для распределённой системы управления TANGO

F i g. 2. Structural diagram of the Web interface of the software complex for the distributed control system TANGO

⁵ Kerrisk M. poll(2) – Linux manual page, 2021 [Электронный ресурс]. URL: <https://www.tango-controls.org> (дата обращения: 20.01.2022).



Полученные результаты

В результате проведённых исследований, представлены и проанализированы наиболее важные элементы разработанного алгоритма программного комплекса для распределённой системы управления TANGO. Посредством встроенных инструментов TANGO, проведён ряд тестов на предмет корректной работы представленного в статье программного комплекса.

Заключение

В результате проделанной работы, был разработан и реализован масштабируемый программный комплекс, позволяю-

щий считывать и обрабатывать данные, полученные с сервера устройств Tango, корректно отображать их с помощью веб-интерфейса в виде графиков и телеметрии. Используя возможности веб-интерфейса и TCP-сервера, данный комплекс позволяет корректно выполнять изменения значений атрибутов сервера устройств Tango. Разработан алгоритм и представлено описание компонентов (составных частей) данного программного комплекса. Представленный в данной статье, программный комплекс можно рекомендовать в качестве базового алгоритма для разработки систем управления таким оборудованием как двигатели, клапаны, осциллографы и т. д.

References

- [1] Foka P, et al. Overview of result from ALICE at the CERN LHC. *Journal of Physics: Conference Series*. 2013; 455:012004. (In Eng.) doi: <https://doi.org/10.1088/1742-6596/455/1/012004>
- [2] Płoskoń M., et al. Overview of results from ALICE. *Journal of Physics: Conference Series*. 2014; 509:012003. (In Eng.) doi: <https://doi.org/10.1088/1742-6596/509/1/012003>
- [3] Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA; 1998. 572 p. (In Eng.)
- [4] Baranov A.V., Balashov N.A., Kutovskiy N.A., Semenov R.N. JINR cloud infrastructure evolution. *Physics of Particles and Nuclei Letters*. 2016; 13(5):672-675. (In Eng.) doi: <https://doi.org/10.1134/S1547477116050071>
- [5] Kouznetsov O., Savin I. Spin Physics Experiments at NICA-SPD. *Nuclear and Particle Physics Proceedings*. 2017; 282-284:20-26. (In Eng.) doi: <https://doi.org/10.1016/j.nuclphysbps.2016.12.005>
- [6] Baranov D., Kapishin M., Mamontova T, et al. The BM@N experiment at JINR: Status and Physics Program. *KnE Energy*. 2018; 3(1):291-296. (In Eng.) doi: <https://doi.org/10.18502/ken.v3i1.1757>
- [7] Korenkov V., Nechaevskiy A., Ososkov G., Priakhina D., Trofimov V. A Probabilistic Approach to the Simulation of Data Processing Centers. *EPJ Web of Conferences: Mathematical Modeling and Computational Physics 2019 (MMCP 2019)*. 2020; 226:03012. (In Eng.) doi: <https://doi.org/10.1051/epjconf/202022603012>
- [8] Balashov N., Baranov A.I., Belov S., Kadochnikov I., Korenkov V., Kutovskiy N., Nechaevskiy A., Pelevanyuk I. Advanced Scheduling in IaaS Clouds. *EPJ Web of Conferences: 23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018)*. 2019; 214:07011. (In Eng.) doi: <https://doi.org/10.1051/epjconf/201921407011>
- [9] Kadochnikov I.S., Papoyan V.V. Blocking Strategies to Accelerate Record Matching for Big Data Integration. *CEUR Workshop Proceedings*. 2019; 2507:219-224. Available at: <http://ceur-ws.org/Vol-2507/219-224-paper-38.pdf> (accessed 20.01.2022). (In Eng.)
- [10] Baginyan A., Balandin A., Balashov N., Dolbilov A., Gavrish A., Golunov A., Gromova N., Kashunin I., Korenkov V., Kutovskiy N., et al. Current Status of the MICC: an Overview. *CEUR Workshop Proceedings*. 2021; 3041:1-8. Available at: <http://ceur-ws.org/Vol-3041/1-8-paper-1.pdf> (accessed 20.01.2022). (In Eng.)
- [11] Belov S., Kadochnikov I., Korenkov V., Reshetnikov A., Semenov R., Zrellov P. Data Analysis Platform for Stream and Batch Data Processing on Hybrid Computing Resources. *CEUR Workshop Proceedings*. 2021; 3041:174-179. Available at: <http://ceur-ws.org/Vol-3041/174-179-paper-32.pdf> (accessed 20.01.2022). (In Eng.)
- [12] Balashov N., Kutovskiy N., Makhalkin A., Mazhitova Y., Pelevanyuk I., Semenov R. Distributed information and computing infrastructure of JINR member states' organizations. *AIP Conference Proceedings*. 2021; 2377(1):040001. (In Eng.) doi: <https://doi.org/10.1063/5.0063809>
- [13] Balashov N., Kutovskiy N., Priakhina D., Sokolov I. Evolution and Perspectives of the Service for Parallel Applications Running at JINR Multifunctional Information and Computing Complex. *EPJ Web of Conferences: Mathematical Modeling and Computational Physics 2019 (MMCP 2019)*. 2020; 226:03002. (In Eng.) doi: <https://doi.org/10.1051/epjconf/202022603002>
- [14] Baginyan A.S., Balandin A.I., Dolbilov A.G., et al. GRID at JINR. *CEUR Workshop Proceedings*. 2019; 2507:321-325. Available at: <http://ceur-ws.org/Vol-2507/321-325-paper-58.pdf> (accessed 20.01.2022). (In Eng.)
- [15] Belov S.D., Kadochnikov I.S., Korenkov V.V., Kutovskiy N.A., Pelevanyuk I.S., Semenov R.N., Zrellov P.V. Integration of the parallel resources to the distributed cloud infrastructures for large scale projects. *CEUR Workshop Proceedings*. 2020; 2772:58-64. Available at: <http://ceur-ws.org/Vol-2772/58-64-paper-9.pdf> (accessed 20.01.2022). (In Eng.)
- [16] Podgainy D., Belaykov D., Nechaevskiy A., Streltsova O., Vorontsov A., Zuev M. IT Solutions for JINR Tasks on the "GOVORUN" Supercomputer. *CEUR Workshop Proceedings*. 2021; 3041:612-618. Available at: <http://ceur-ws.org/Vol-3041/612-618-paper-113.pdf> (accessed 20.01.2022). (In Eng.)



- [17] Balashov N., Kuprikov I., Kutovskiy N., Makhalkin A., Mazhitova Ye., Pelevanyuk I., Semenov R. JINR Distributed Information and Computing Environment: Participants, Features and Challenges. *CEUR Workshop Proceedings*. 2021; 3041:280-284. Available at: <http://ceur-ws.org/Vol-3041/280-284-paper-52.pdf> (accessed 20.01.2022). (In Eng.)
- [18] Dolbilov A., Kashunin I., Korenkov V., Kutovskiy N., Mitsyn V., Podgainy D., Stretsova O., Strizh T., Trofimov V., Vorontsov A. Multifunctional Information and Computing Complex of JINR: Status and Perspectives. *CEUR Workshop Proceedings*. 2019; 2507:16-22. Available at: <http://ceur-ws.org/Vol-2507/16-22-paper-3.pdf> (accessed 20.01.2022). (In Eng.)
- [19] Balashov N., Baranov A., Kutovskiy N., Makhalkin A., Mazhitova Y., Pelevanyuk I., Semenov R. Present Status and Main Directions of the JINR Cloud Development. *CEUR Workshop Proceedings*. 2019; 2507:185-189. Available at: <http://ceur-ws.org/Vol-2507/185-189-paper-32.pdf> (accessed 20.01.2022). (In Eng.)
- [20] Balashov N.A., Kutovskiy N.A., Sokolov I.V. Problem-oriented interface for MICC. *AIP Conference Proceedings*. 2021; 2377(1):040002. (In Eng.) doi: <https://doi.org/10.1063/5.0064008>
- [21] Balashov N., Kuprikov I., Kutovskiy N., Makhalkin A., Mazhitova Y., Semenov R. Quantitative and Qualitative Changes in the JINR Cloud Infrastructure. *CEUR Workshop Proceedings*. 2021; 3041:275-279. Available at: <http://ceur-ws.org/Vol-3041/275-279-paper-51.pdf> (accessed 20.01.2022). (In Eng.)
- [22] Korenkov V., Dolbilov A., Mitsyn V., Kashunin I., Kutovskiy N., Podgainy D., Streltsova O., Strizh T., Trofimov V., Zrellov P. The JINR distributed computing environment. *EPJ Web of Conferences: 23rd International Conference on Computing in High Energy and Nuclear Physics (CHEP 2018)*. 2019; 214:03009. (In Eng.) doi: <https://doi.org/10.1051/epjconf/201921403009>
- [23] Belyakov D., Dolbilov A., Moshkin A., Pelevanyuk I., Podgainy D., Rogachevsky O., Streltsova O., Zuev M. Using the "Govorun" Supercomputer for the NICA Megaproject. *CEUR Workshop Proceedings*. 2019; 2507:316-320. Available at: <http://ceur-ws.org/Vol-2507/316-320-paper-57.pdf> (accessed 20.01.2022). (In Eng.)
- [24] Andreev V.A., Volkov V.I., Gorbachev E.V., Isadov V.A., Kirichenko A.E., Romanov S.V., Sedykh G.S. TANGO standard software to control the Nuclotron beam slow extraction. *Physics of Particles and Nuclei Letters*. 2016; 13(5):605-608. (In Eng.) doi: <https://doi.org/10.1134/S1547477116050034>
- [25] Gupta A., Kar S. The Common Object Request Broker Architecture (CORBA) and its Notification Service. *IETE Technical Review*. 2002; 19(1-2):31-45. (In Eng.) doi: <https://doi.org/10.1080/02564602.2002.11417009>

*Поступила 20.01.2022; одобрена после рецензирования 25.02.2022; принята к публикации 12.03.2022.
Submitted 20.01.2022; approved after reviewing 25.02.2022; accepted for publication 12.03.2022.*

Об авторах:

Бондяков Алексей Сергеевич, инженер-программист Лаборатории информационных технологий имени М. Г. Мещерякова, Международная межправительственная организация Объединенный институт ядерных исследований (141980, Российская Федерация, г. Дубна, Московская область, ул. Жолио-Кюри, д. 6), кандидат технических наук, **ORCID: <https://orcid.org/0000-0003-0429-3931>**, aleksey@jinr.ru

Кондратьев Андрей Олегович, инженер-программист Лаборатории информационных технологий имени М. Г. Мещерякова, Международная межправительственная организация Объединенный институт ядерных исследований (141980, Российская Федерация, г. Дубна, Московская область, ул. Жолио-Кюри, д. 6), **ORCID: <https://orcid.org/0000-0001-6203-9160>**, konratyev@jinr.ru

Все авторы прочитали и одобрили окончательный вариант рукописи.

About the authors:

Aleksey S. Bondyakov, Software Engineer of the Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation), Cand.Sci. (Tech.), **ORCID: <https://orcid.org/0000-0003-0429-3931>**, aleksey@jinr.ru

Andrey O. Kondratyev, Software Engineer of the Meshcheryakov Laboratory of Information Technologies, Joint Institute for Nuclear Research (6 Joliot-Curie St., Dubna 141980, Moscow region, Russian Federation), **ORCID: <https://orcid.org/0000-0001-6203-9160>**, konratyev@jinr.ru

All authors have read and approved the final manuscript.

