



**International Conference
"Distributed Computing and Grid-Technologies
in Science and Education"**

June 30 – July 5, 2014, Dubna, Russia

GRID'2014

Book of Abstracts

**DISTRIBUTED COMPUTING
AND GRID-TECHNOLOGIES IN SCIENCE
AND EDUCATION**

Book of Abstracts of the 6th International Conference

Dubna, June 30 – July 5, 2014

**РАСПРЕДЕЛЕННЫЕ ВЫЧИСЛЕНИЯ
И ГРИД-ТЕХНОЛОГИИ В НАУКЕ
И ОБРАЗОВАНИИ**

Тезисы докладов 6-й Международной конференции

Дубна, 30 июня – 5 июля 2014 г.

**RECENT DEVELOPMENTS IN THE CONTRIBUTION OF
DFCTI/IFIN-HH TO THE WLCG COLLABORATION21**
CIUBANCAN Mihai, IVANOAIKA Teodor, DULEA Mihnea

**THE DEVELOPMENT OF AN ARM SYSTEM ON CHIP BASED
PROCESSING UNIT FOR DATA STREAM COMPUTING22**
COX Mitchell

**DERIVING SEMANTICS FROM WS-BPEL SPECIFICATIONS OF
PARALLEL BUSINESS PROCESSES ON AN EXAMPLE22**
DIMITROV Vladimir

**PILITE: A UNIFIED INTERFACE TO LOCAL RESOURCE
MANAGERS ON SUPERCOMPUTING RESOURCES23**
DUBENSKAYA Yulia, KRYUKOV Alexander, DEMICHEV Andrey, PRIKHODKO Nikolay

**HIGH-PERFORMANCE AND GRID COMPUTING AT INCDTIM,
CLUJ-NAPOCA, ROMANIA24**
FLOARE Calin Gabriel, FARCAS Felix, ADAM Gheorghe

RUNNING APPLICATIONS ON A HYBRID CLUSTER24
GAIDUCHOK Vladimir, YUZHANIN Nikolai, GANKEVICH Ivan, BOGDANOV Alexander

**EFFICIENT PROCESSING AND CLASSIFICATION OF WAVE
ENERGY SPECTRUM DATA WITH A DISTRIBUTED PIPELINE.....25**
GANKEVICH Ivan, DEGTYAREV Alexander

**APPLICATIONS OF ON-DEMAND VIRTUAL CLUSTERS TO HIGH
PERFORMANCE COMPUTING26**
GANKEVICH Ivan, KORKHOV Vladimir, BALYAN Serob, ABRAAMYAN Suren

**REVIEW OF JULIA PROGRAMMING LANGUAGE FOR SCIENTIFIC
COMPUTING27**
GEVORKYAN Migran, KULYABOV Dmitry, SEVASTYANOV Leonid

**NUMERICAL SIMULATION OF COMPLEX SEISMIC PROBLEMS IN
HETEROGENEOUS MEDIA USING HIGH-PERFORMANCE
COMPUTING SYSTEMS28**
GOLUBEV Vasily, PETROV Igor, KHOKHLOV Nikolay, FAVORSKAYA Alena,
BABICHEV Dmitry

INTEGRATION OF THE COMPUTING CLUSTER INTO THE INFORMATION SYSTEM OF FACILITY	36
KULYABOV Dmitry, GEVORKYAN Migran, SEVASTYANOV Leonid	
CLOUD INFRASTRUCTURE AT JINR.....	37
KUTOVSKIY Nikolay, BALASHOV Nikita, BARANOV Alexandr, SEMENOV Roman	
STORAGE DATABASE IN CLOUD PROCESSING	37
KYAW Thurein, BOGDANOV Alexander	
HARDWARE PLATFORMS OF PARALLEL AND DISTRIBUTED SIMULATION TECHNOLOGY	38
KYAW Wunna, MYO MIN Swe, DEGTYAREV Alexander	
TIER-1 FOR ALICE, ATLAS & LHCb AT THE KURCHATOV INSTITUTE (NRC KI). CURRENT STATUS	38
LAZIN Yury	
ALLOCATION STEINER POINTS IN EUCLIDEAN STEINER TREE PROBLEM BY MEANS OF MATLAB PACKAGE	38
LOTAREV Dmitriy	
TOOL FOR UTILIZING IDLE RESOURCES OF COMPUTING CLUSTERS IN VOLUNTEER COMPUTING	39
MANZYUK Maxim, ZAIKIN Oleg, POSYPKIN Mikhail	
DEFINING VOLUNTEER COMPUTING: A FORMAL APPROACH.....	40
MAROSI Attila Csaba, LOVAS Robert	
USE OF NESTED VIRTUALIZATION WHEN WORKING WITH PRIVATE OPENSTACK CLOUD AND HYPERVISOR VMWARE ESXi	41
MINUKHIN Sergii, LEONTIEV Igor	
HIGH-THROUGHPUT PARALLEL PIPELINED DATA PROCESSING SYSTEM FOR REMOTE EARTH SENSING BIG DATA IN THE CLOUDS	42
NOVIKOV Alexander, POYDA Alexey, AULOV Vasiliy	
ACTIVITIES AND PERSPECTIVES AT ARMENIAN GRID SITE	43
OGANEZOV Hovhannes, ASTSATRYAN Hrachya	

REVIEW OF JULIA PROGRAMMING LANGUAGE FOR SCIENTIFIC COMPUTING

GEVORKYAN Migran, KULYABOV Dmitry, SEVASTYANOV Leonid
mngevorkyan@sci.pfu.edu.ru, yamadharma@gmail.com, leonid.sevast@gmail.com

Peoples' Friendship University of Russia

Julia is new high-level programming language that supports multiple programming styles and primarily focuses on scientific computing [1]. Language has been developed at the Massachusetts Institute of Technology (MIT) in 2009 under the MIT license and is available through the public repository on GitHub [2]. Language is under active development and its latest version is 0.3.

Language has built-in tools for parallel and distributed computing, and supports multithreading (coroutines). Developers have mentioned that Julia does not impose any particular style of parallelism on the user but provides a set of blocks that allows implementing parallelism as programmer likes.

Julia is scientific oriented language and it influence on all aspects of language. There are built-in mathematical oriented data types in Julia, such as rational and complex numbers. Dynamic type conversion avoids a loss of accuracy, which is important in the scientific calculations. Also a large variety of built-in mathematical functions and a large number of libraries are present.

One of the main goals of the language was to achieve high performance close to the C/Fortran, and clarity interpreted languages. Besides compiling source files into an executable program Julia has the ability to execute language constructs interactively (like interpreted languages), as well as an interface to support iPython.

Another key feature of the language is simple API for functions and subroutines of C/Fortran/Python. The standard library has lots of built-in wrappers for popular libraries (eg, BLAS and LAPACK). Also, there is a possibility to connect external libraries C / Fortran / Python.

Julia supports multiple programming paradigms: procedural, object-oriented, functional programming. Language is under active development and is replenished with libraries for various fields of mathematics, physics and engineering sciences.

1. The Julia Language — <http://julialang.org/>

2. The Julia Language: GitHub — <https://github.com/JuliaLang>