

DUBNA

Integration of geographically distributed heterogeneous resources based on the DIRAC Interware

Speaker: Igor Pelevanyuk

Main participants

DIRAC:

Igor Pelevanyk, Andrey Tsaregorodtzev

Experiments:

MPD: Oleg Rogachevskiy, Andrey Moshkin

BM@N: Konstantin Gertsenberger

Responsibles for resources:

Tier-1,Tier-2, EOS: Valery Mitsyn

dCache: Vladimir Trofimov

Cloud: Nikolay Kutovskiy and the team

Govorun: Dmitry Podgainy and the team

LHEP cluster: Boris Schinov, Andrey Dolbilov

UNAM cluster: Luciano Diaz

Administrative support:

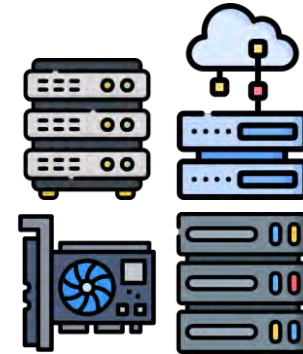
Vladimir Korenkov, Tatiana Strizh, Petr Zrelov

Principles of Integration

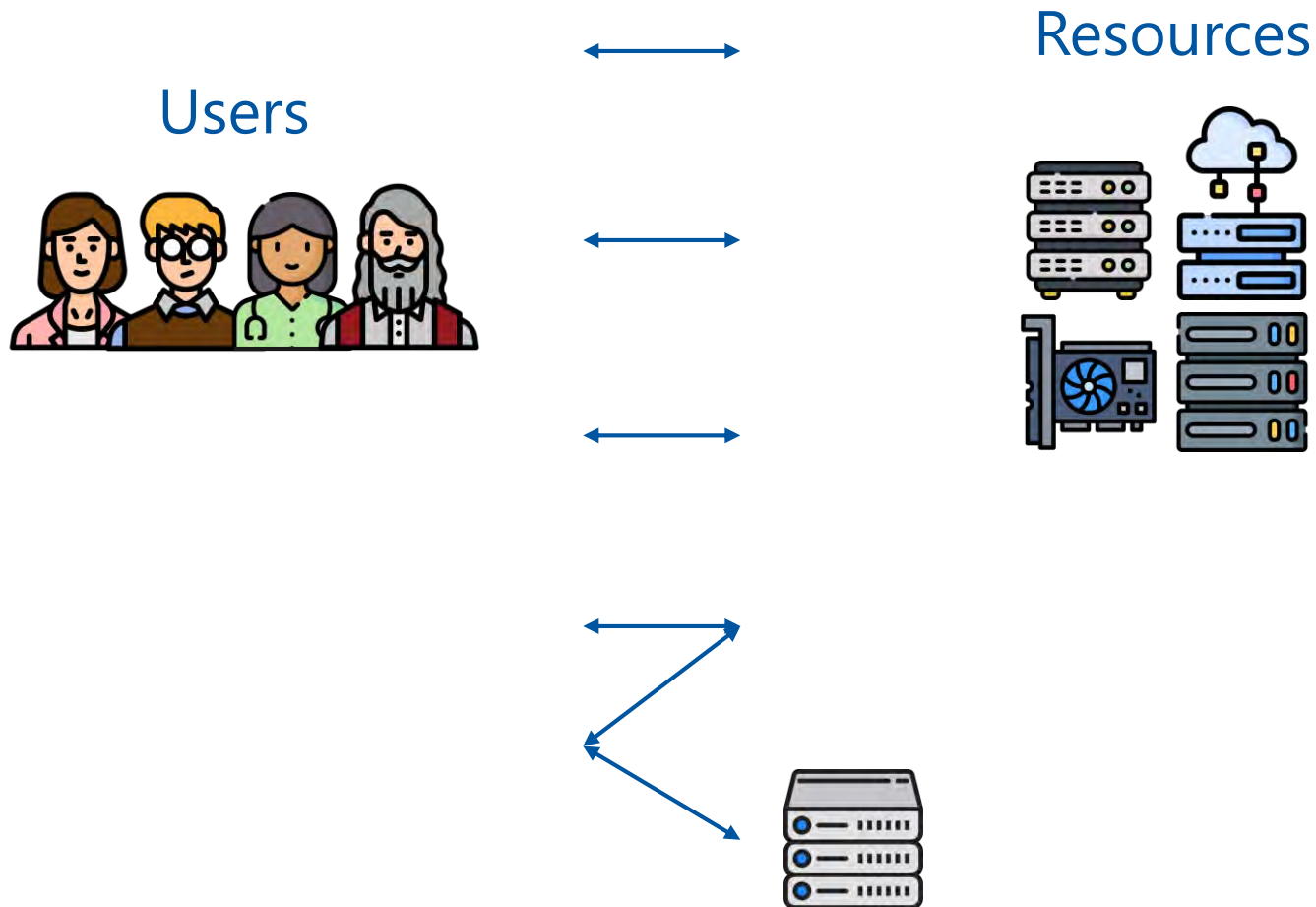
Users



Resources



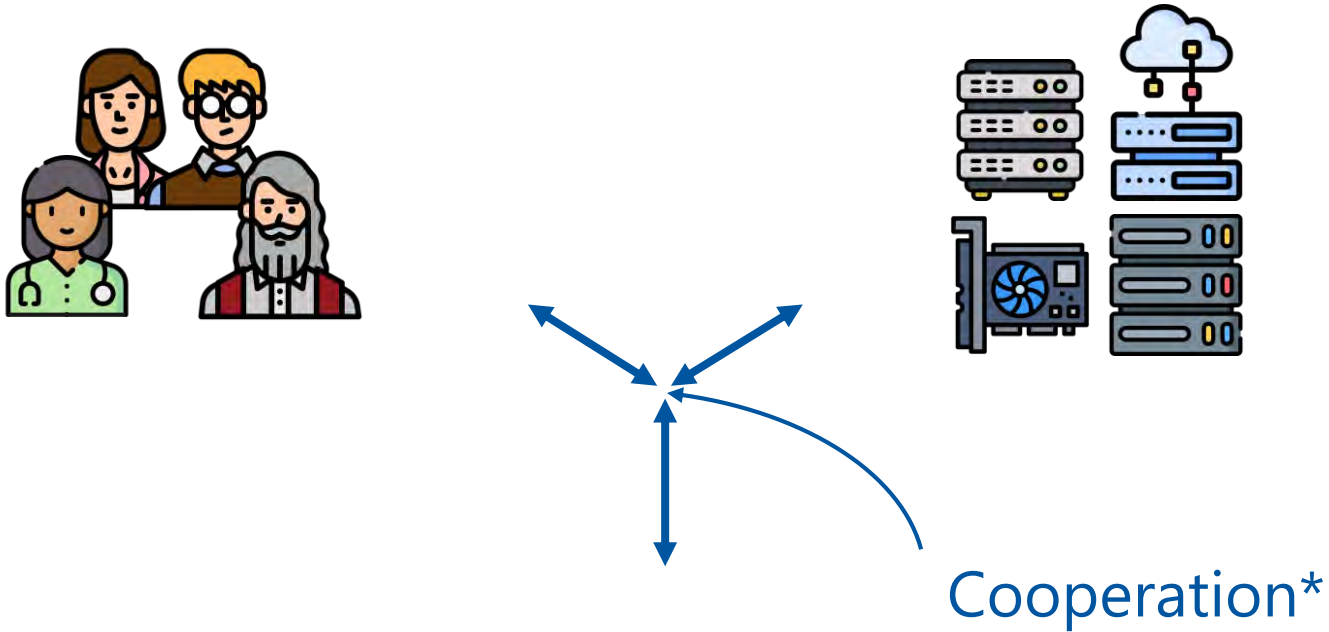
Principles of Integration








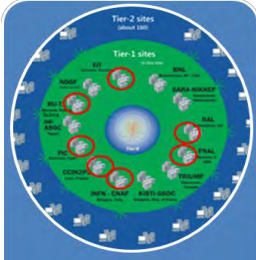


No silver bullet



Principles of Integration

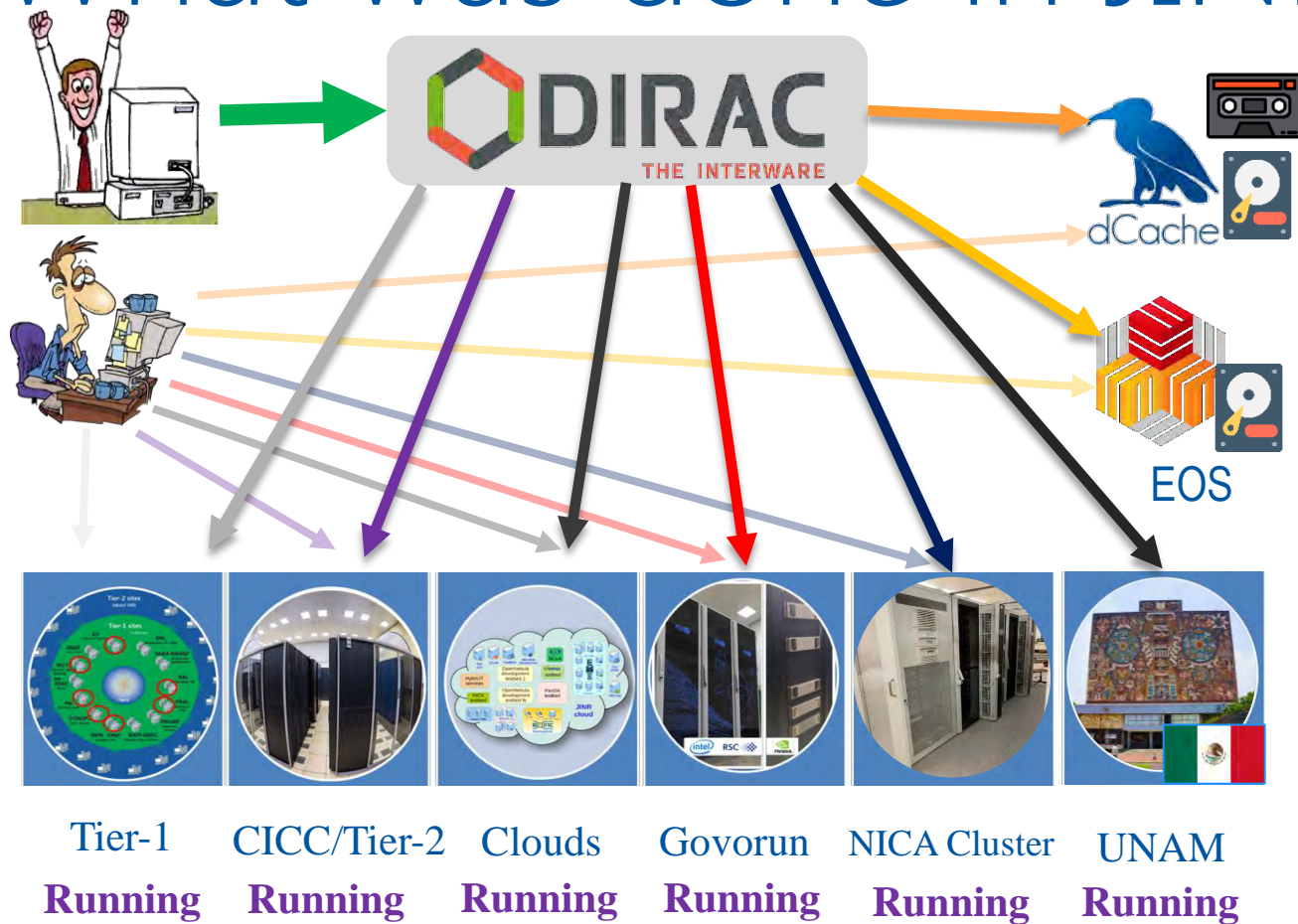


MICC Resources

Storage	 EOS	 dCache Disk, Tape	 ceph	
Protocol	local, root	GridFTP, root	local	local
Auth Storage	Kerb. , x509	x509	ceph key	<i>HybriLIT</i>
Auth Jobs	Kerb. x509	x509	SSO	<i>HybriLIT</i>
Job Submit.	Torque Grid	Grid	OpenNebula	Slurm
Component				
	Tier-2/CICC	Tier-1	Cloud	Govoron/HybriLIT

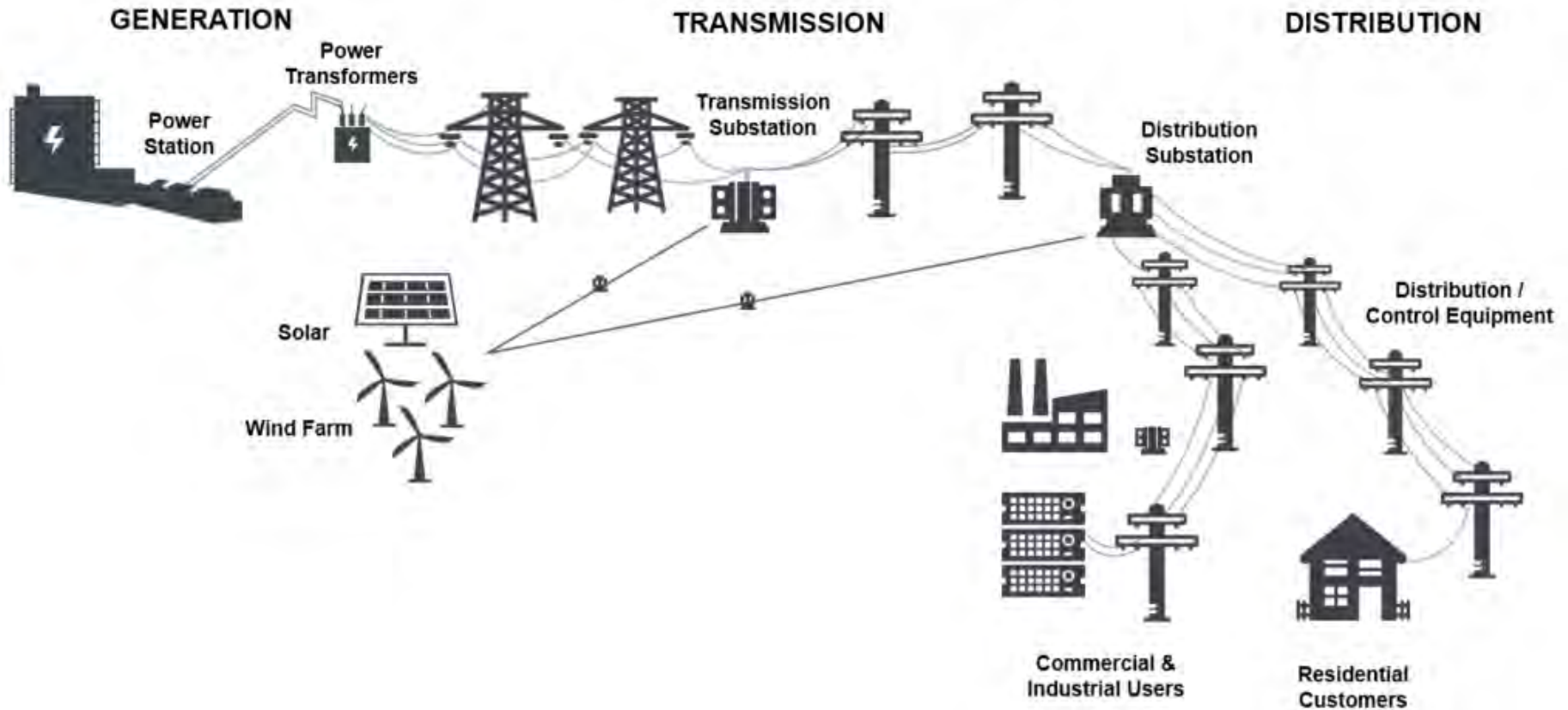
* This is a simplified slide to demonstrate complexity and variability of protocols and accesses approaches

What was done in JINR

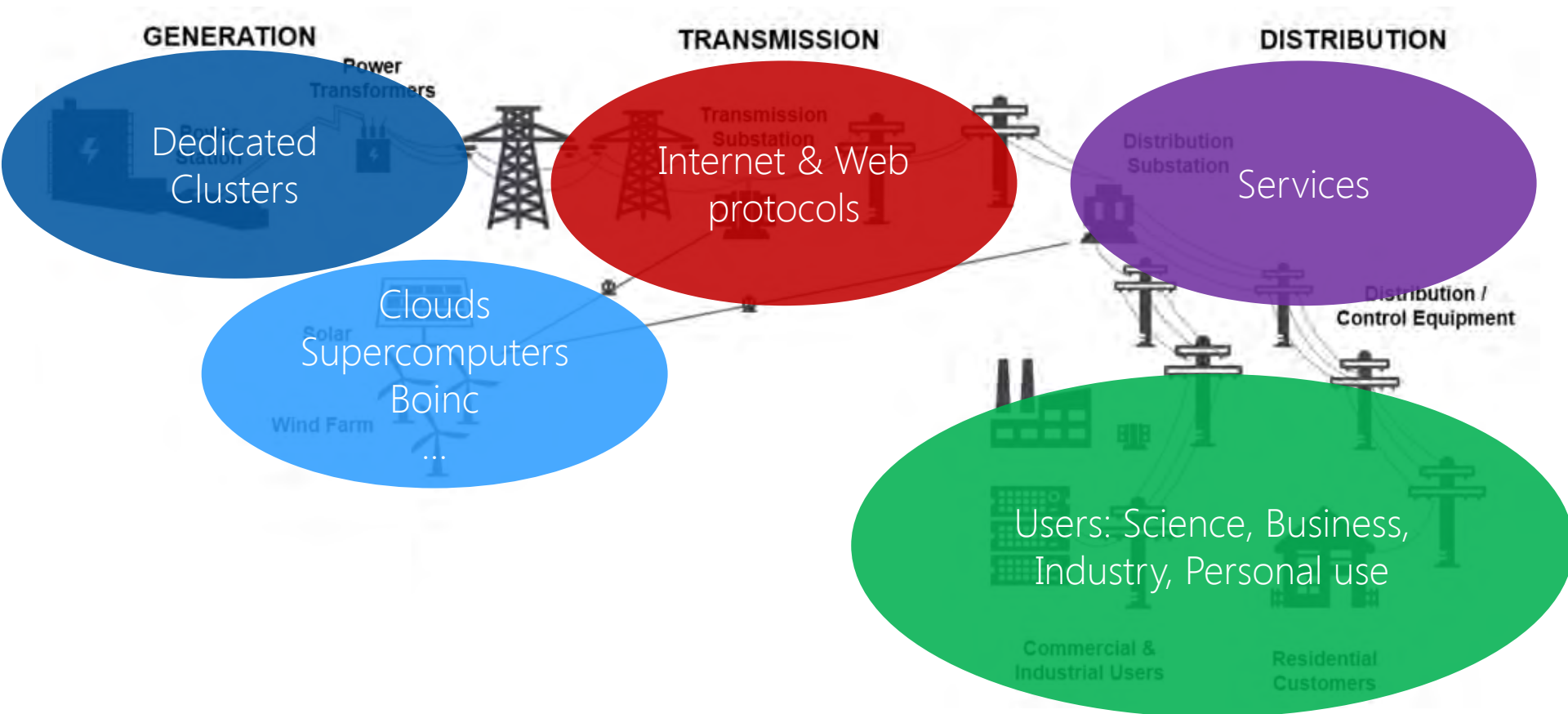


The computing resources of the JINR Multifunctional Information and Computing Complex, clouds in JINR Member-States, cluster from Mexico University were combined using the DIRAC Interware.

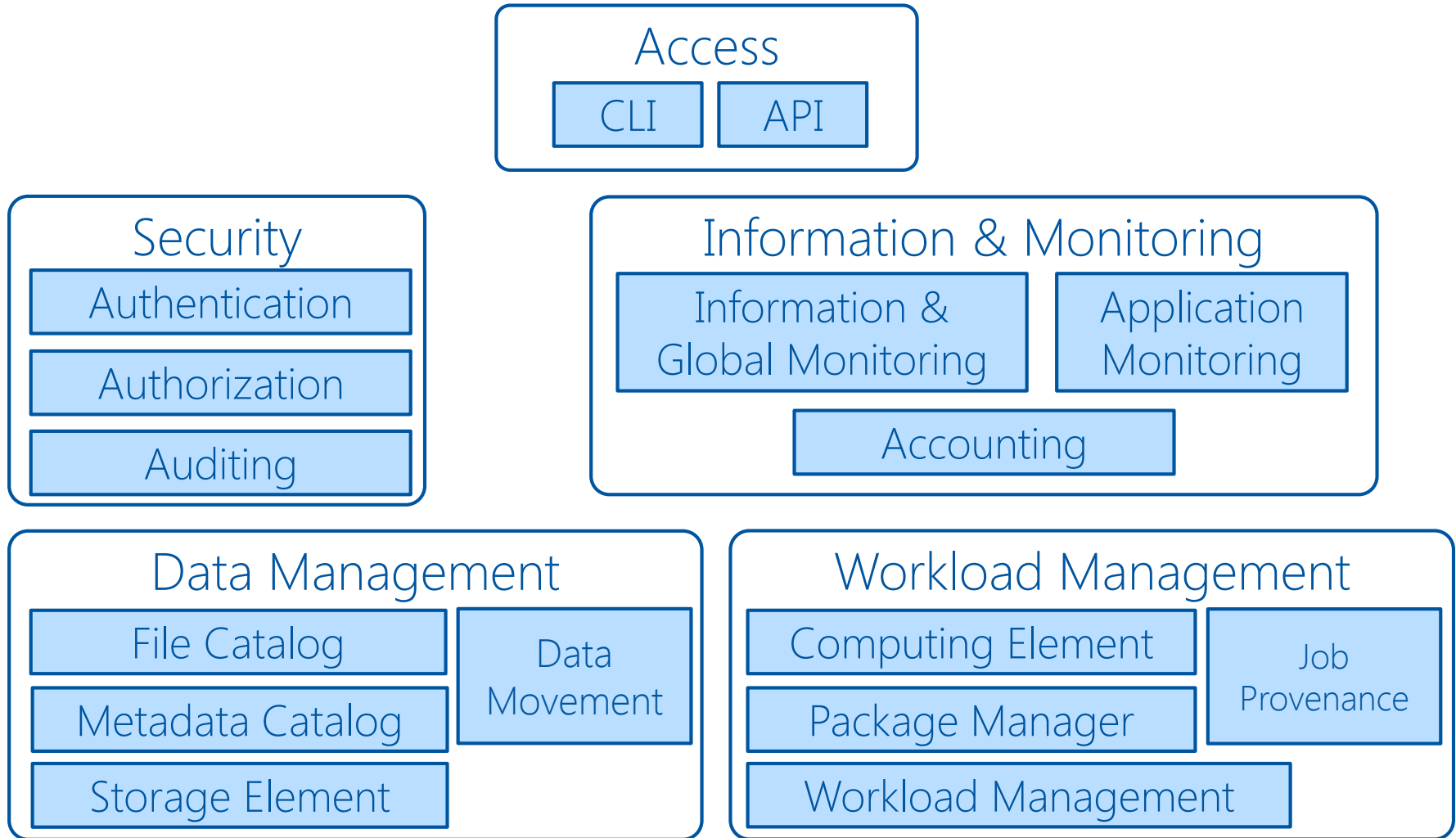
What is grid?



What is grid?

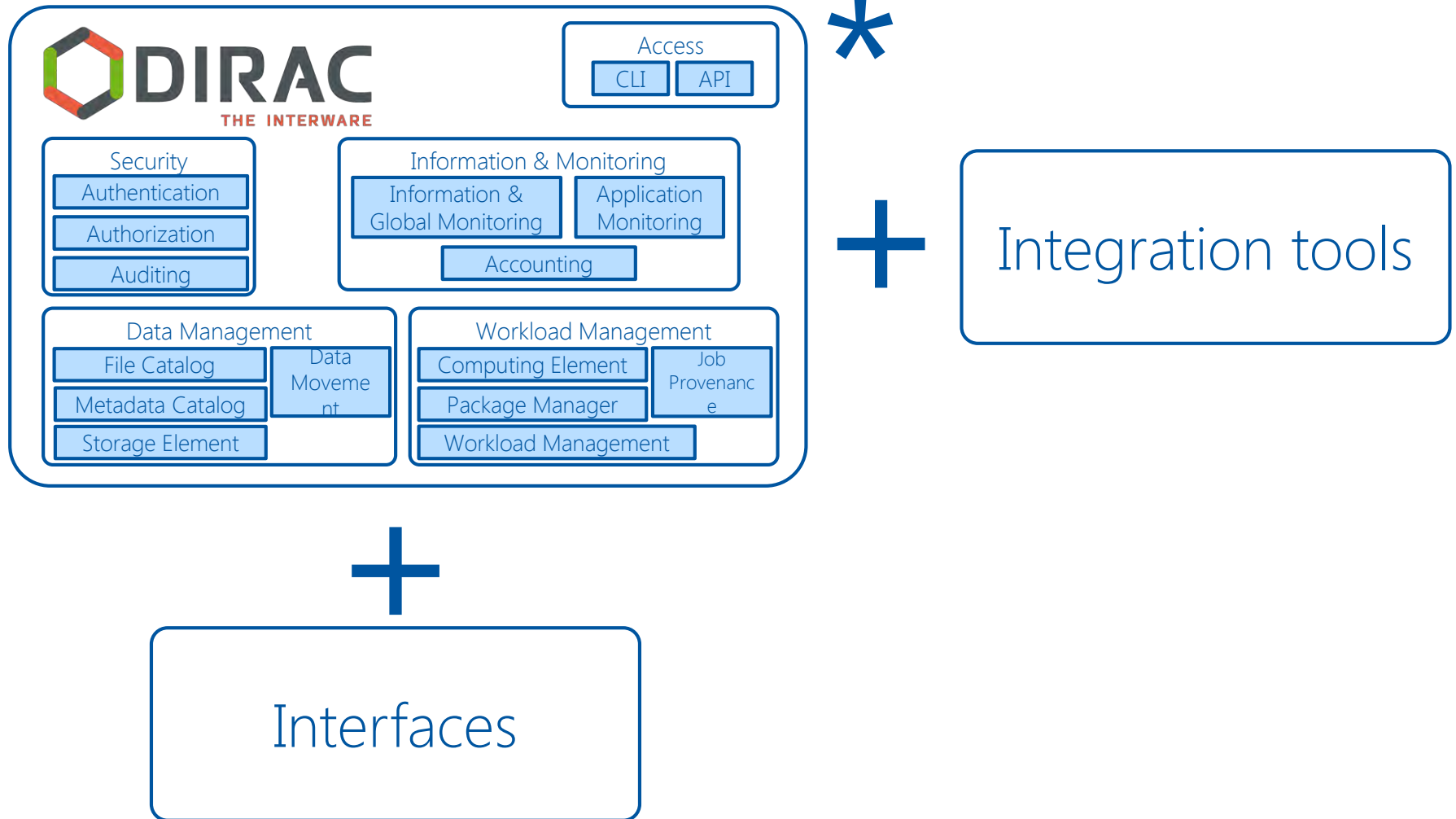


What is grid?



*** This is logical schema. Real schema may include different realization of services and different protocols of interconnection**

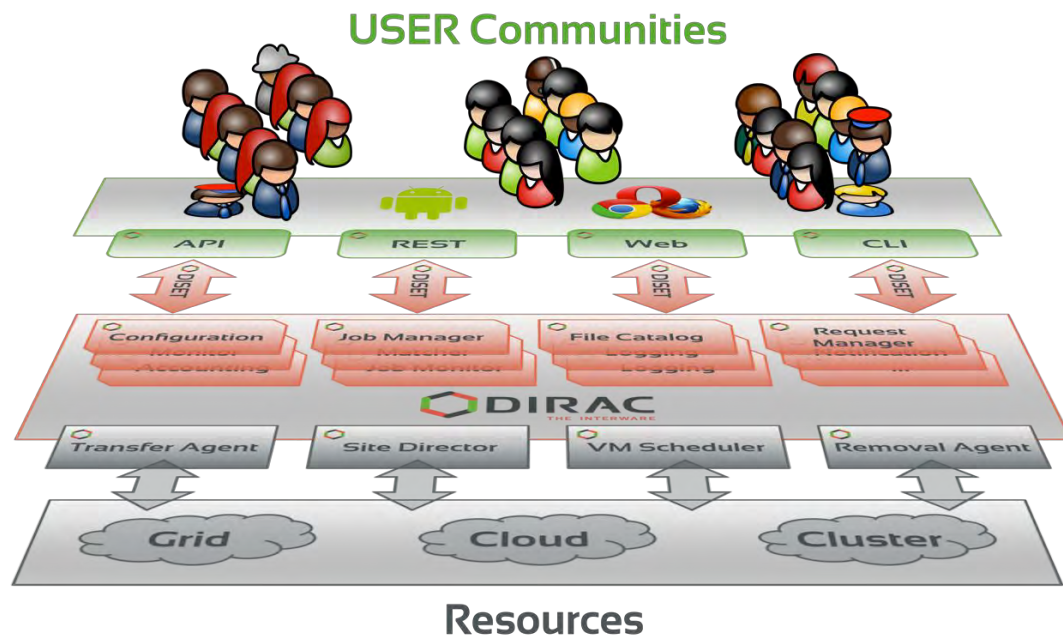
So, what is DIRAC?



*** This schema is just for demonstration. Some components are omitted and some are not included in DIRAC, like Package Manager.**

What is DIRAC?

DIRAC provides all the necessary components to build ad-hoc grid infrastructures **interconnecting** computing resources of different types, allowing **interoperability** and simplifying **interfaces**. This allows to speak about the DIRAC *interware*.



Web

CLI

API

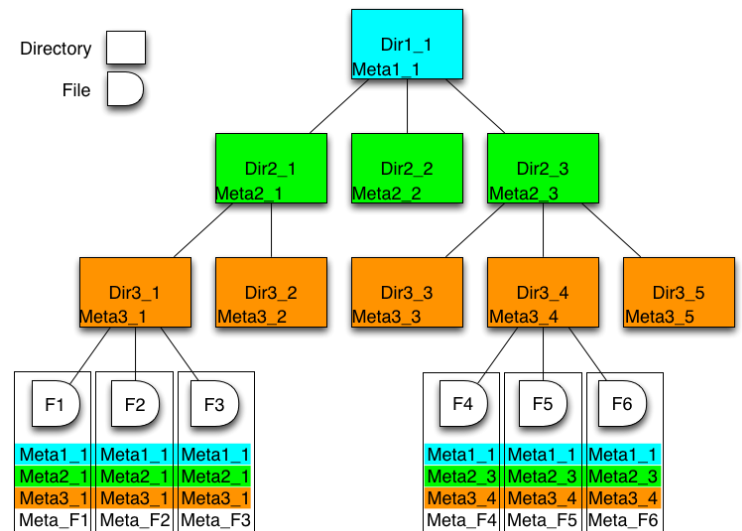
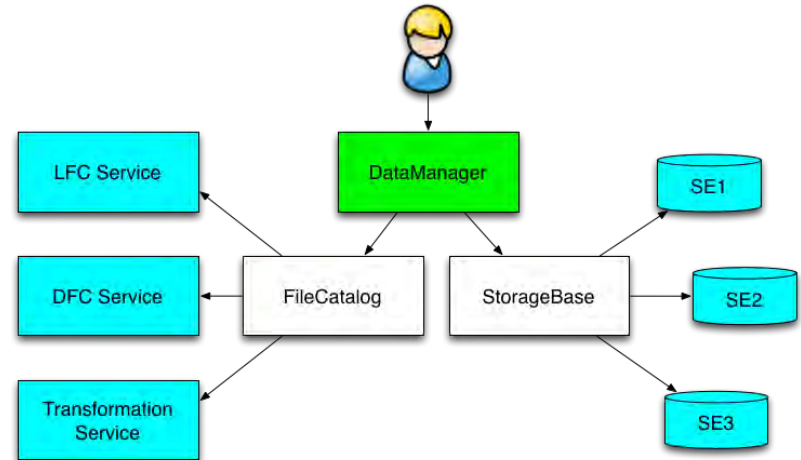
REST

Storage resources

- Storage element abstraction with a client implementation for each access protocol
 - DIPS, SRM, XROOTD, RFIO, etc
 - gfal2 based plugin gives access to all protocols supported by the library: DCAP, WebDAV, S3 ...
- Each SE is seen by the clients as a logical entity
 - SE's can be configured with multiple protocols

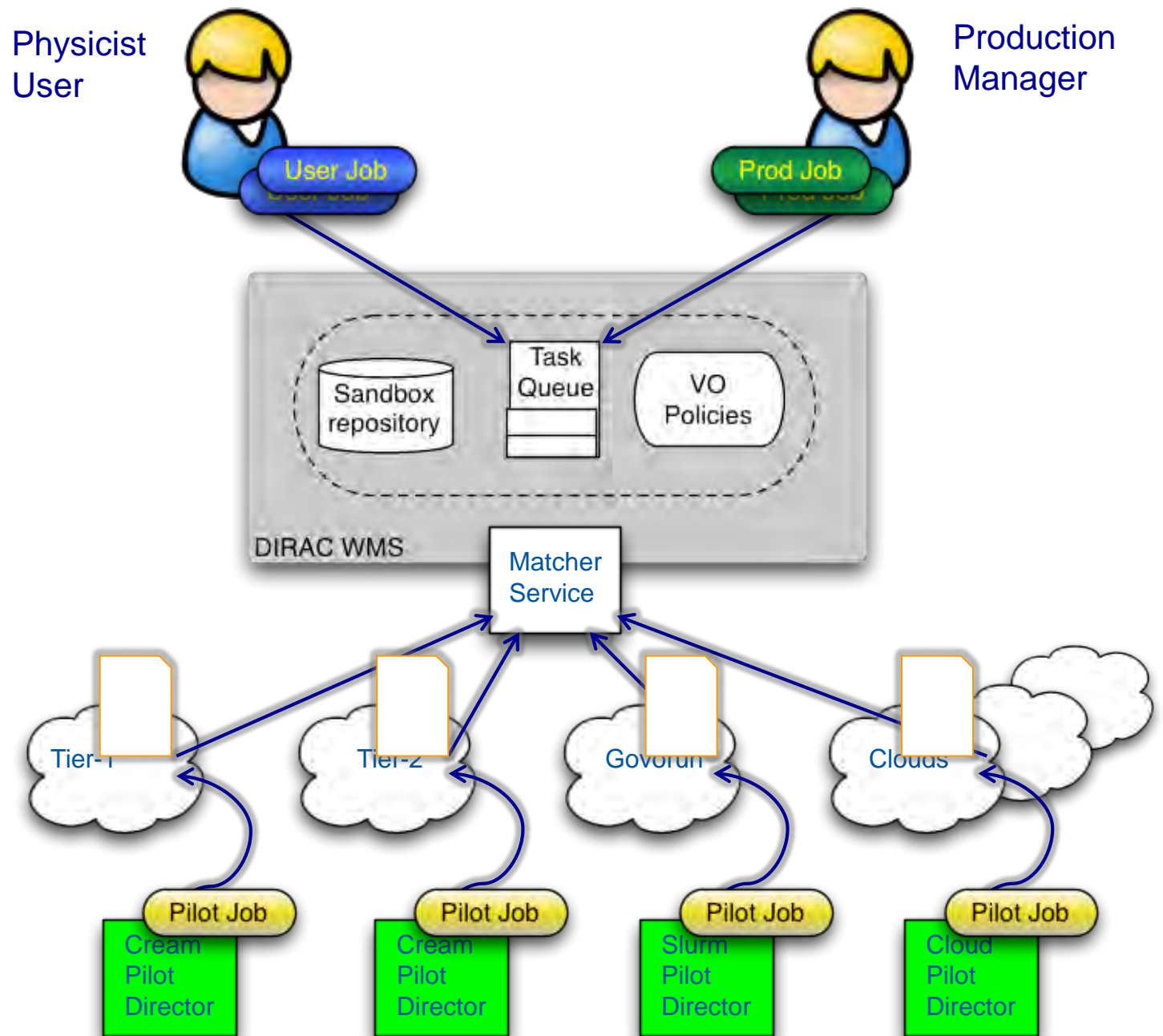
Storage resources

- DIRAC File Catalog(DFC) is maintaining a single global logical name space
- A user sees it as a single catalog with additional features
- DataManager is a single client interface for logical data operations
- DFC also may host Metadata
 - User defined metadata
 - The same hierarchy for metadata as for the logical name space
 - Metadata associated with files and directories
 - Allow for efficient searches
 - Efficient Storage Usage reports



Computing resources

- Computing Elements
 - Conventional GRIDs: gLite/EMI, VDT, ARC
- Clouds
 - OpenNebula, OpenStack, EC2(Amazon), OCCl
- Computing clusters
 - Resources available at Universities and scientific laboratories
 - LSF, BQS, SGE, PBS/Torque, Condor, SLURM
- HPC Centers, or Supercomputers
 - Computing centers oriented towards massively parallel applications using specialized hardware
- Volunteer Computing
 - Mostly based on BOINC technology SETI@Home, LHC@Home, etc



What is DIRAC job

Input of Job

Executable name: /bin/l`s`

Arguments: -la *.txt

Input Sandbox : temp.sh,
test.exe, program.py, any.cnf

Output Sandbox: std.out,
std.err

Output of Job

Job result: Done, Failed

Minor Status: Data uploaded

Output files: std.out, std.err

DIRAC standard job workflow

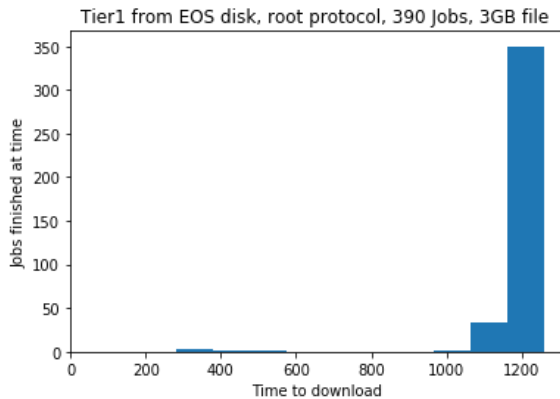
1. Initial configuration
2. Input data download
3. Processing
4. Output data upload
5. Finalization

Main HEP use-cases

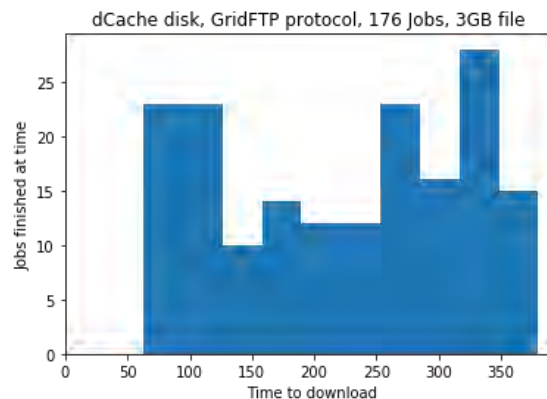
	MC-Generation	Reconstruction	Merge files
InputSize	Very Small	Big	Big
OutputSize	Variable	May be smaller	Big
Disk usage	Variable	High	High

Measuring transfers speed

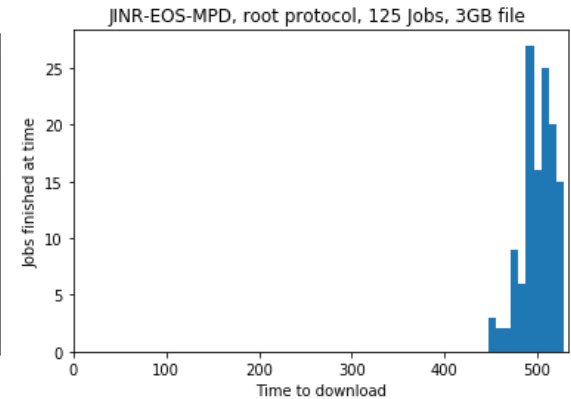
$$\text{Transfer speed} = \frac{\text{Jobs amount} * \text{File size}}{\text{Slower download time}}$$



EOS disk - root
Jobs: 390
Transferred: 1170.0GB
Transfer speed: 0.93GB/s



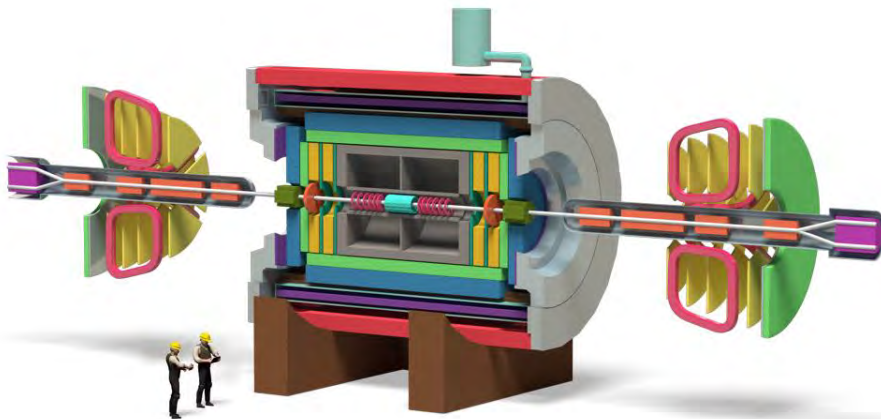
dCache disk - gridFTP
Jobs: 176
Transferred: 528.0GB
Transfer speed: 1.39GB/s



EOS disk - root (from cloud)
Jobs: 125
Transferred: 375.0GB
Transfer speed: 0.71GB/s

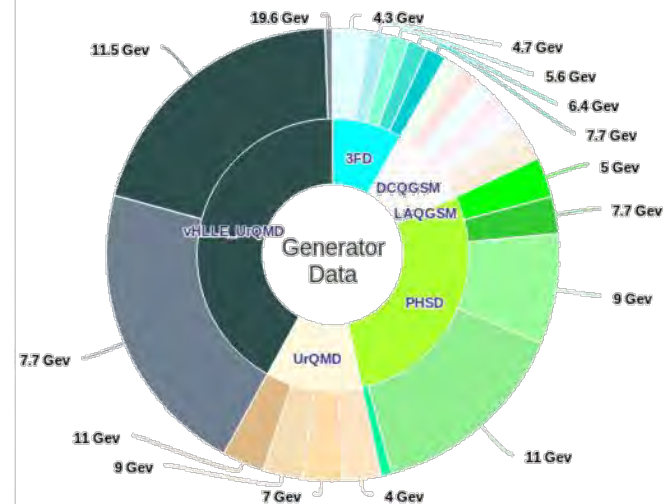
MPD MC generation

The MPD(Multi Purpose Detector) apparatus has been designed as a 4π spectrometer capable of detecting of charged hadrons, electrons and photons in heavy-ion collisions at high luminosity in the energy range of the NICA collider. To reach this goal, the detector will comprise a precise 3-D tracking system and a high-performance particle identification (PID) system based on the time-of-flight measurements and calorimetry.



MPD detector

Monte-Carlo generation plan

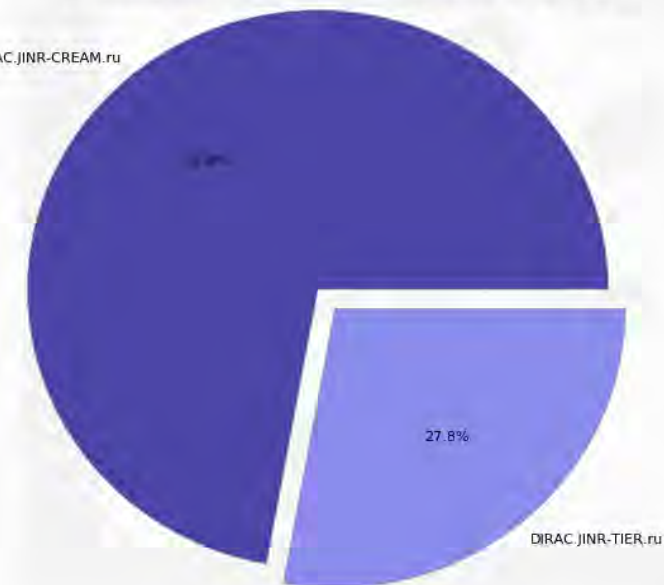


MPD MC generation

- Andrey Moshkin is responsible for launch of all jobs which generated for MPD using DIRAC infrastructure.

Total Number of Jobs by Site

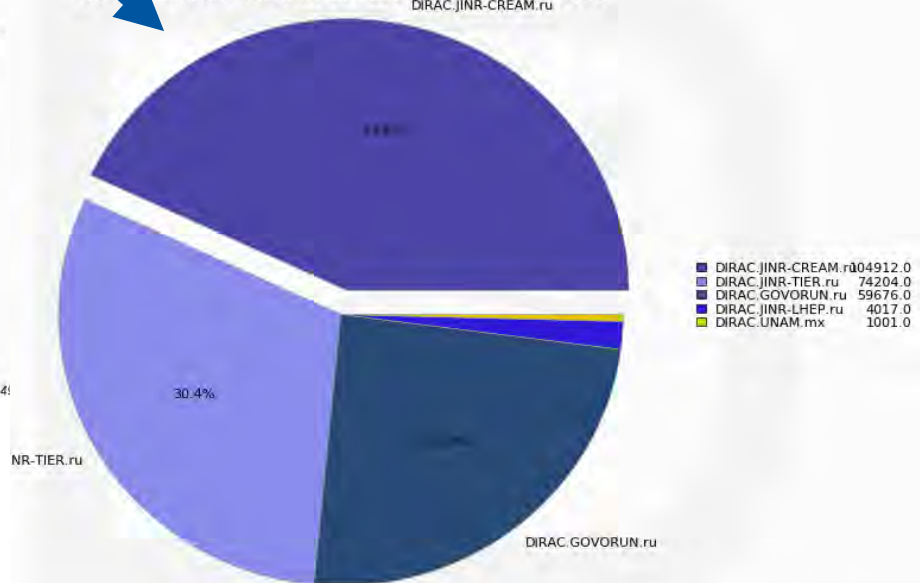
15 Weeks from Week 21 of 2019 to Week 37 of 2019



Generated on 2019-09-16 21:25:41

Total Number of Jobs by Site

96 Weeks from Week 37 of 2018 to Week 30 of 2020



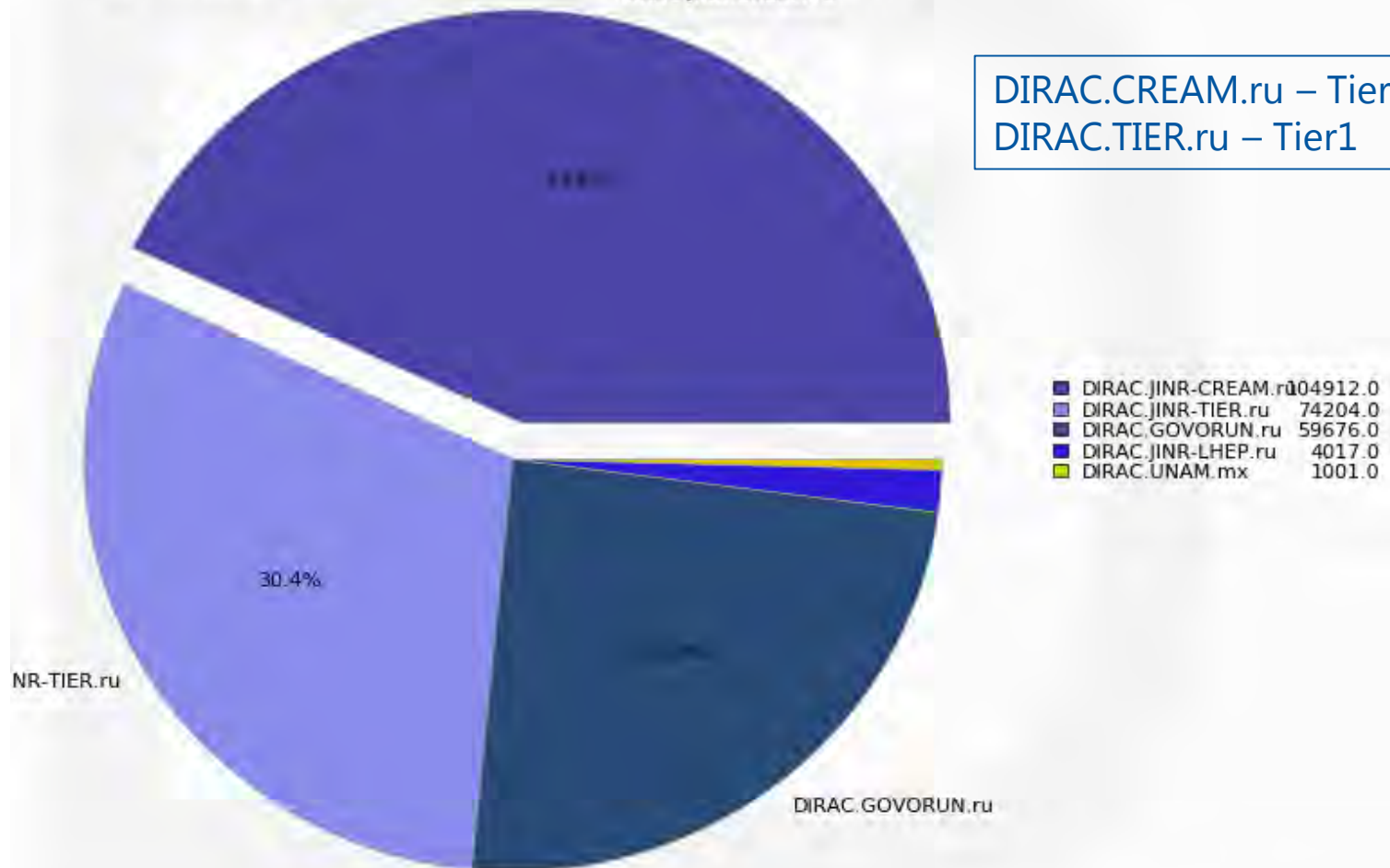
Generated on 2020-07-28 15:03:49 UTC

MPD Total Successful jobs

Total Number of Jobs by Site

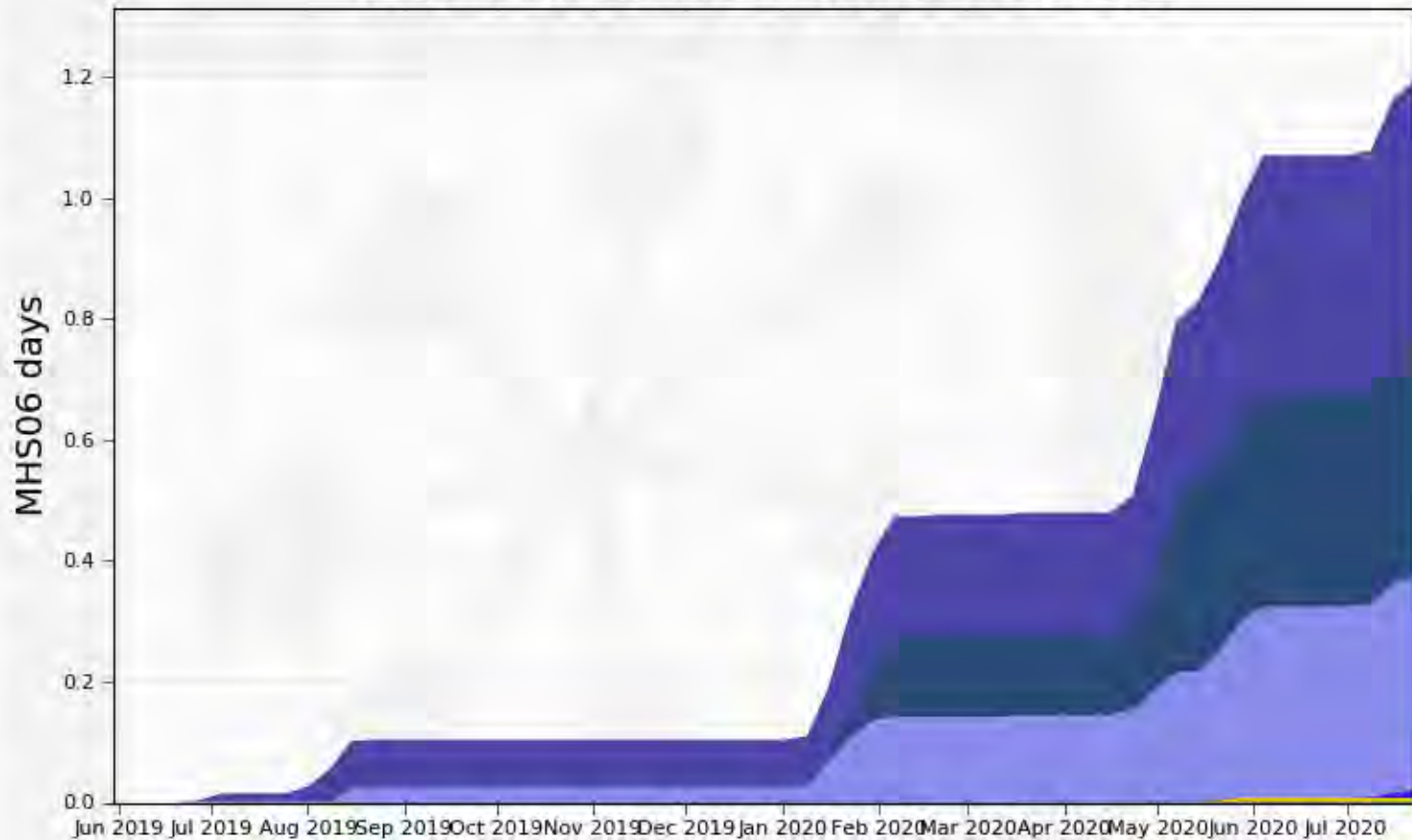
96 Weeks from Week 37 of 2018 to Week 30 of 2020

DIRAC.JINR-CREAM.ru



MPD Total

Normalized CPU used by Site
60 Weeks from Week 21 of 2019 to Week 29 of 2020



Max: 1.20, Average: 0.36, Current: 1.20

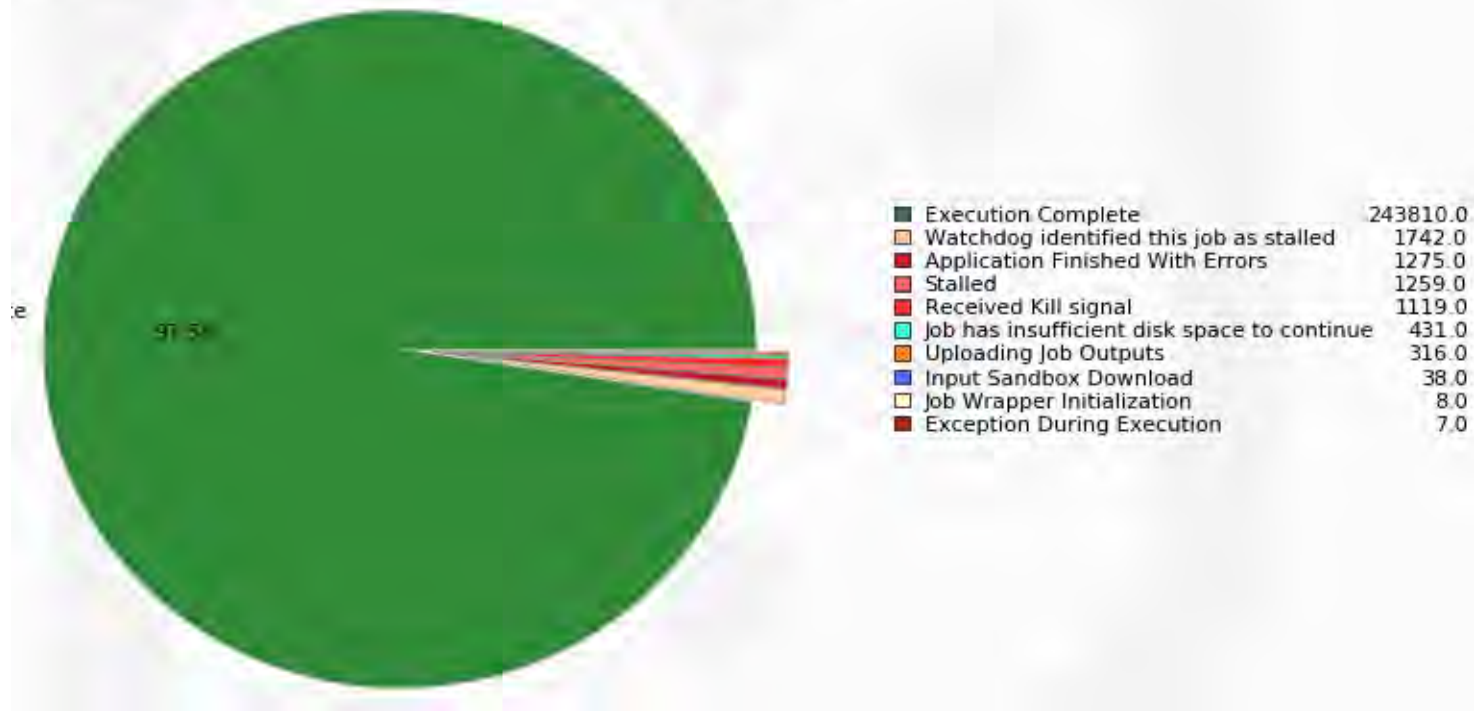
DIRAC.JINR-CREAM.ru 0.4 DIRAC.JINR-TIER.ru 0.4 DIRAC.UNAM.mx 0.0
DIRAC.GOVORUN.ru 0.4 DIRAC.JINR-LHEP.ru 0.0

Generated on 2020-07-28 15:24:24 UTC

MPD Total job by status

Total Number of Jobs by FinalMinorStatus

96 Weeks from Week 37 of 2018 to Week 30 of 2020

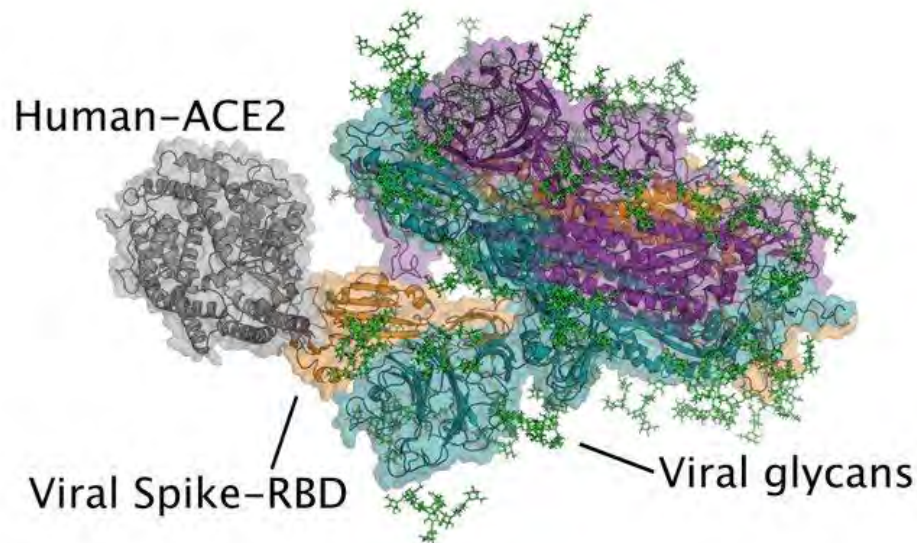


Conclusion on MPD

- Cooperation is the key.
- In JINR DIRAC performance was not a bottleneck at any time.
- Monte-Carlo generation for MPD is a success. It became possible thanks to cooperation of many people and teams.

Folding@Home

Folding@home (FAH or F@h) is a distributed computing project aimed to help scientists develop new therapeutics to a variety of diseases by the means of simulating protein dynamics. This includes the process of protein folding and the movements of proteins, and is reliant on the simulations run on the volunteers' personal computers.

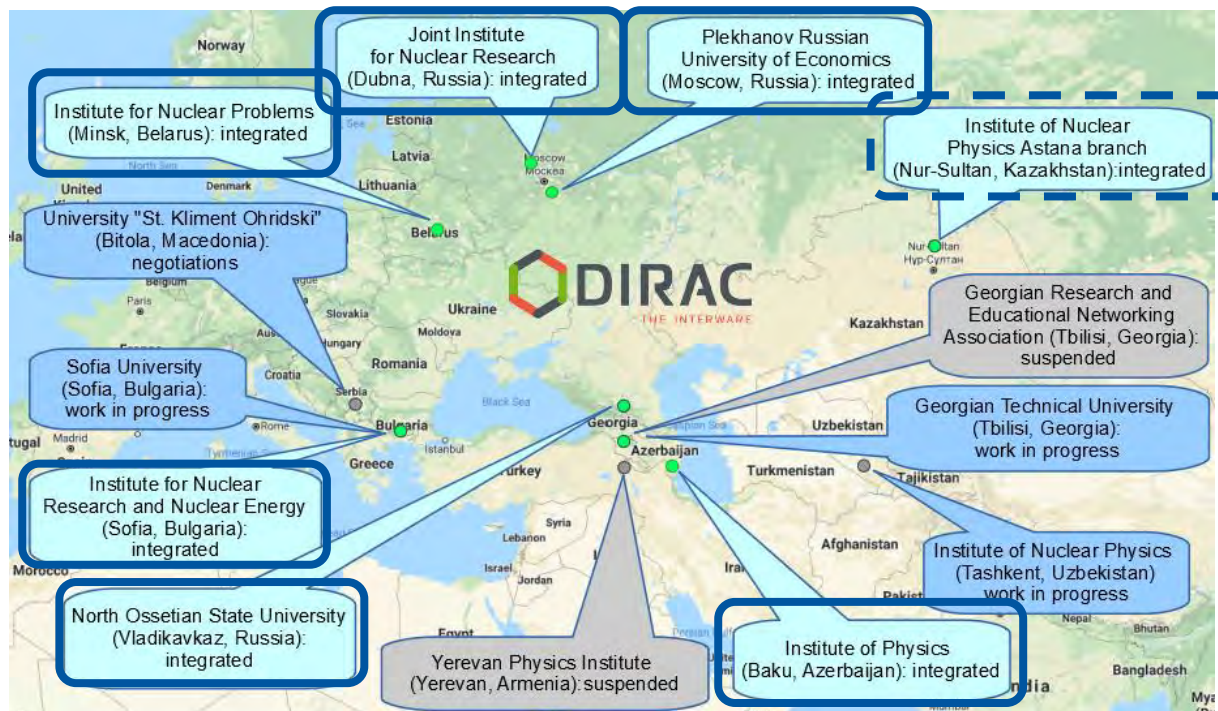


The project utilizes central processing units (CPUs), graphics processing units (GPUs), and other devices. With heightened interest in the project as a result of the COVID-19 pandemic, the system achieved a speed of approximately 1.22 exaflops by late March 2020 and reaching 2.43 exaflops by April 12, 2020, making it the world's first exaflop computing system.

JINR Cloud + Member States

Integration of JINR Member States and JINR cloud to DIRAC was performed. It was done to join resources for solving common tasks as well as to distribute a peak load across resources of partner organizations to speed up receiving of scientific results.

All integrated clouds are based on OpenNebula platform.



Clouds Fighting COVID-19

In March 2020, Folding@home launched a program to assist researchers around the world who are working on finding a cure and learning more about the coronavirus pandemic. The initial wave of projects simulate potentially druggable protein targets from SARS-CoV-2 virus, and the related SARS-CoV virus, about which there is significantly more data available.

Team: Joint Institute for Nuclear Research

Date of last work unit 2020-07-28 15:57:30
Active CPUs within 50 days 4,034
Team Id 265602
Grand Score [14,803,449](#)
Work Unit Count [6,074](#)
Team Ranking 8083 of 254436
Homepage <http://www.jinr.ru/main-en/>

Team members

Rank	Name	Credit	WUs
82,626	CLOUD.JINR.ru	7,400,592	2,994
96,763	CLOUD.PRUE.ru	5,668,413	2,352
222,721	CLOUD.INP.by	994,797	388
321,635	CLOUD.NOSU.ru	362,878	123
322,405	CLOUD.IPANAS.az	360,424	203
N/A	CLOUD.INRNE.bg	16,345	14

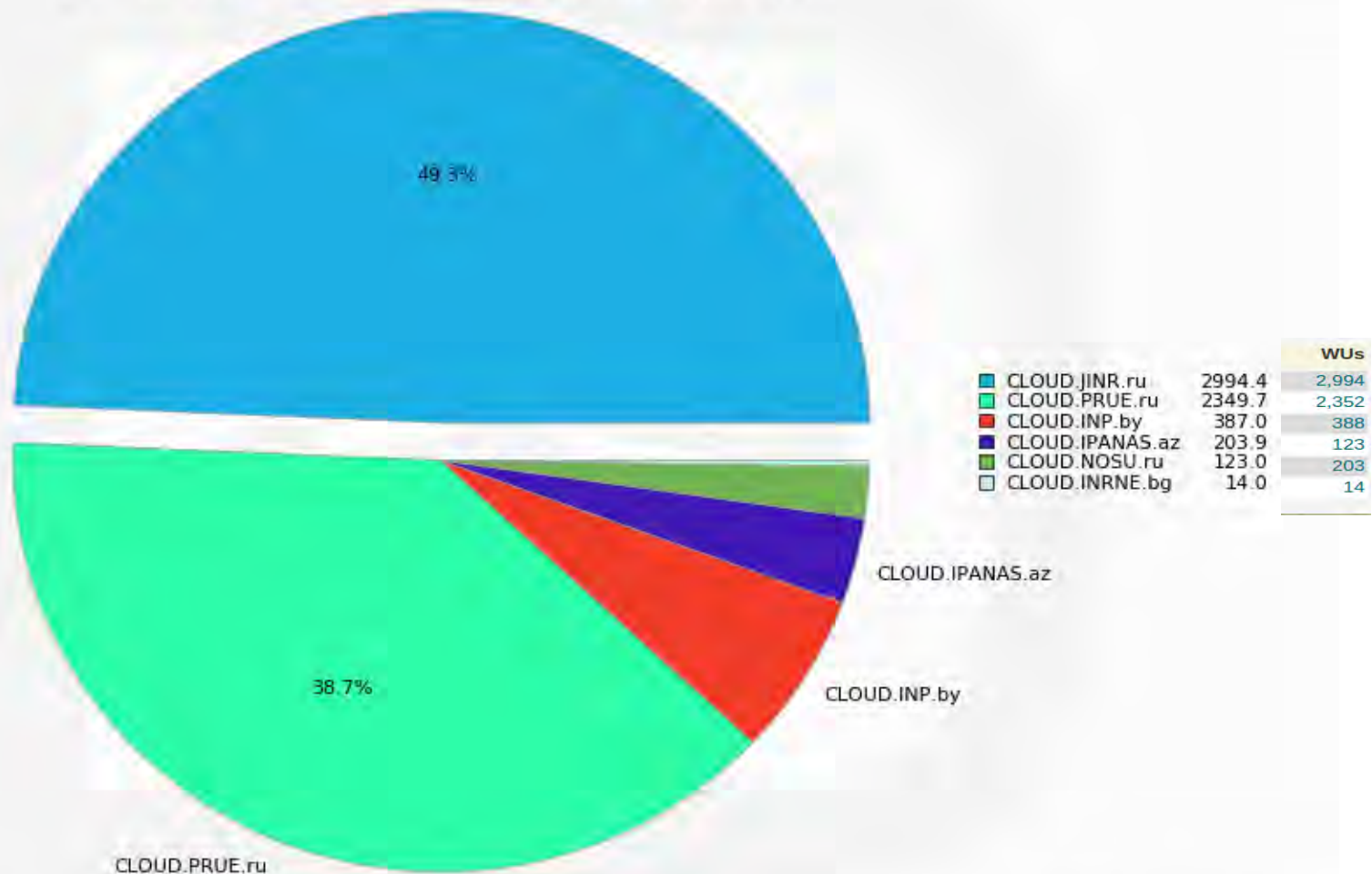
It was decided to use idle cloud resources to participate in F@H.

Our priorities:

1. Do not interfere with other VMs
2. Control of the usage
3. Accounting
4. Only COVID-19 jobs accepted

Clouds Fighting COVID-19

Total Number of Jobs by Site
10 Weeks from Week 19 of 2020 to Week 30 of 2020



Generated on 2020-07-28 16:42:57 UTC

Conclusion on F@H

- DIRAC allowed running Folding@home jobs within a week after initial idea.
- After clouds integration, “usually” no special efforts required from local administrators.
- Small chunks of idle resources could be utilized.
- In case of problems, stopping Folding@home is matter of several commands.

Benchmarks

- Benchmark is the act of running a computer program, a set of programs, or other operations, in order to assess the **relative performance** of an object, normally by running a number of standard tests and trials against it.



Benchmark of one aspect



Synthetic test



Benchmarks in grid

- HEP-SPEC2006 is a standard benchmark in computing for high energy physics.

“We propose a new name for the benchmark: HEP-SPEC06. This acknowledges the fact that we use a benchmark derived from the SPEC CPU2006 benchmark suite, but with a clearly defined way of running it (tuned for HEP: on Linux, with gcc compiler, with the optimization switches we defined and in Multiple speed mode), and underlines the difference between it and the other benchmarks of the SPEC family. “[1]

- The DIRAC Benchmark 2012 (DB12) is a good alternative to HEP-SPEC2006 [2].

This benchmark was originally created for prediction of the duration of LHCb Monte-Carlo tasks. It is fast (takes around 60 seconds), and it runs every time the DIRAC pilot job agent starts execution.

[1] Michele Michelotto et al. A Comparison of HEP code with SPEC benchmarks on multi-core worker nodes, Journal of Physics: Conference Series 219 (2010) 052009 doi:10.1088/1742-6596/219/5/052009

[2] P Charpentier, Benchmarking worker nodes using LHCb productions and comparing with HEPspec06, IOP Conf. Series: Journal of Physics: Conf. Series 898 (2017) 082011 doi:10.1088/1742-6596/898/8/082011

Analogy example

Piece of road from point A to B ←———— A Monte-Carlo task

Speed of the car ←———— Performance of the computer

Time to complete ←———— Time to complete

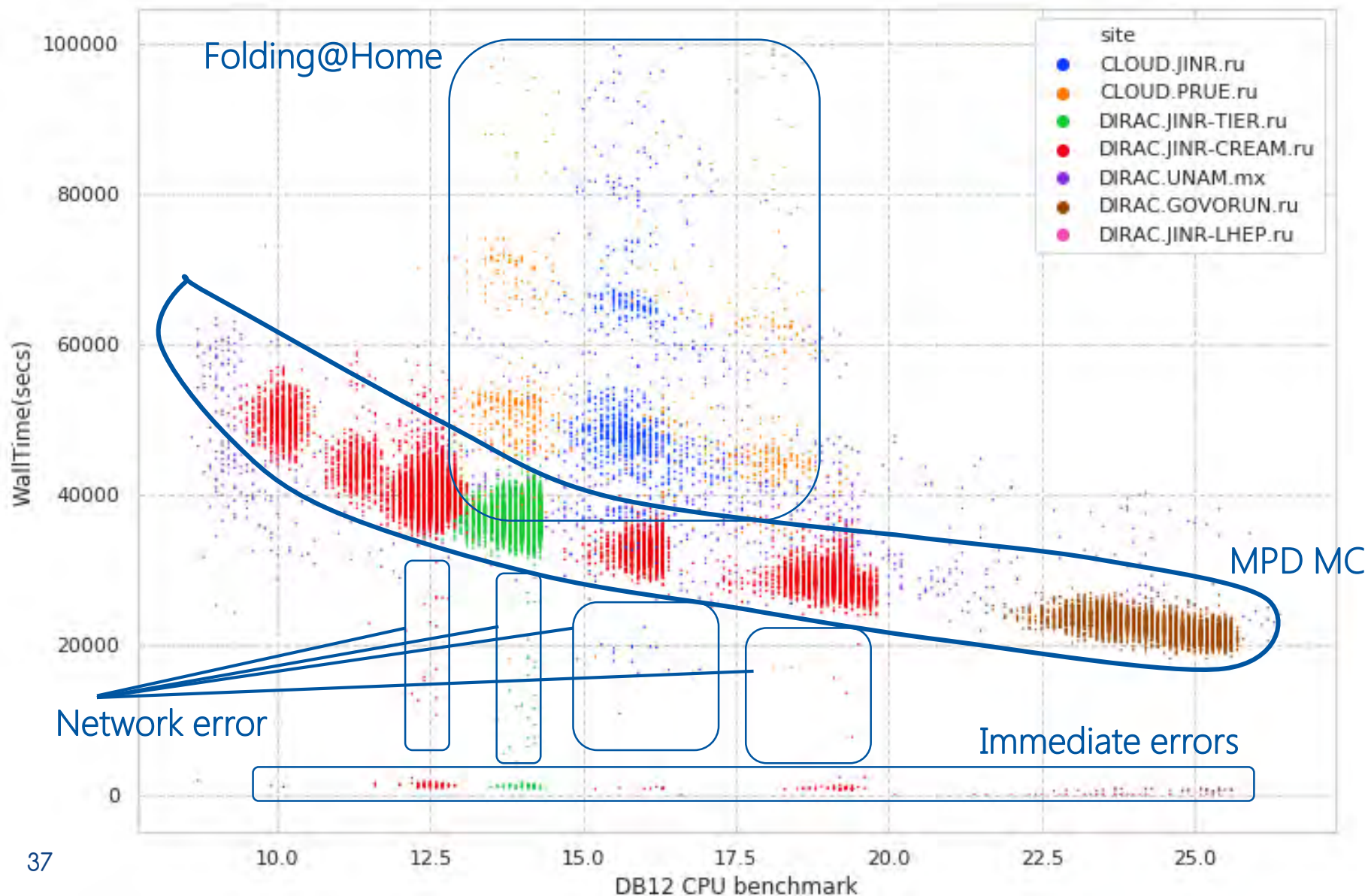
$$Time = \frac{Amount\ of\ work}{Speed\ of\ computer}$$

DB12 gives results like: 10(old slow core), 17 (standard server core), 27 (high performance core)

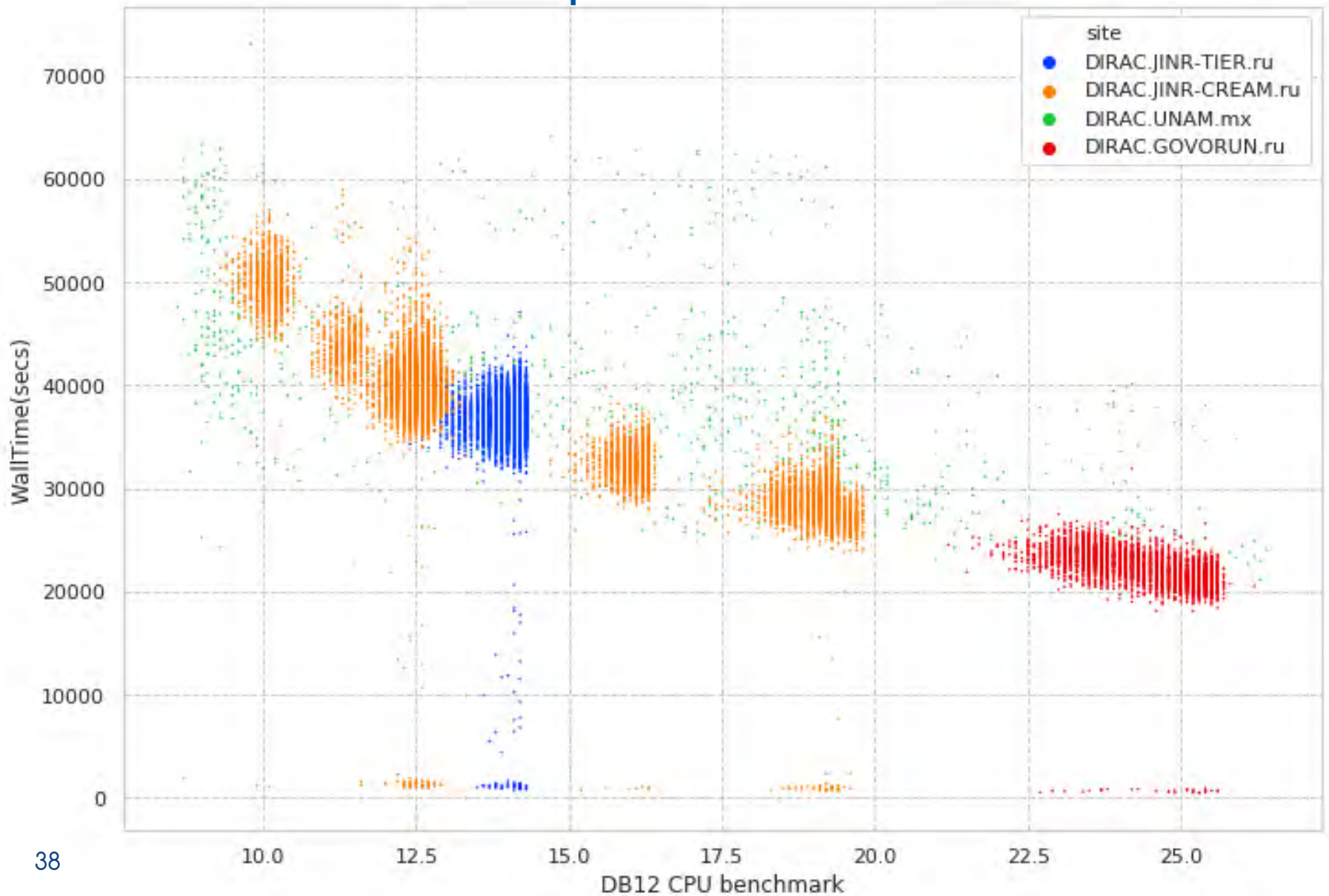
What if we build a plot, where X is DB12 result, Y is time in seconds. Then, every point on the plot represent one job.

It will be useless if all jobs are unique and different. But, in real life there are usually many similar jobs.

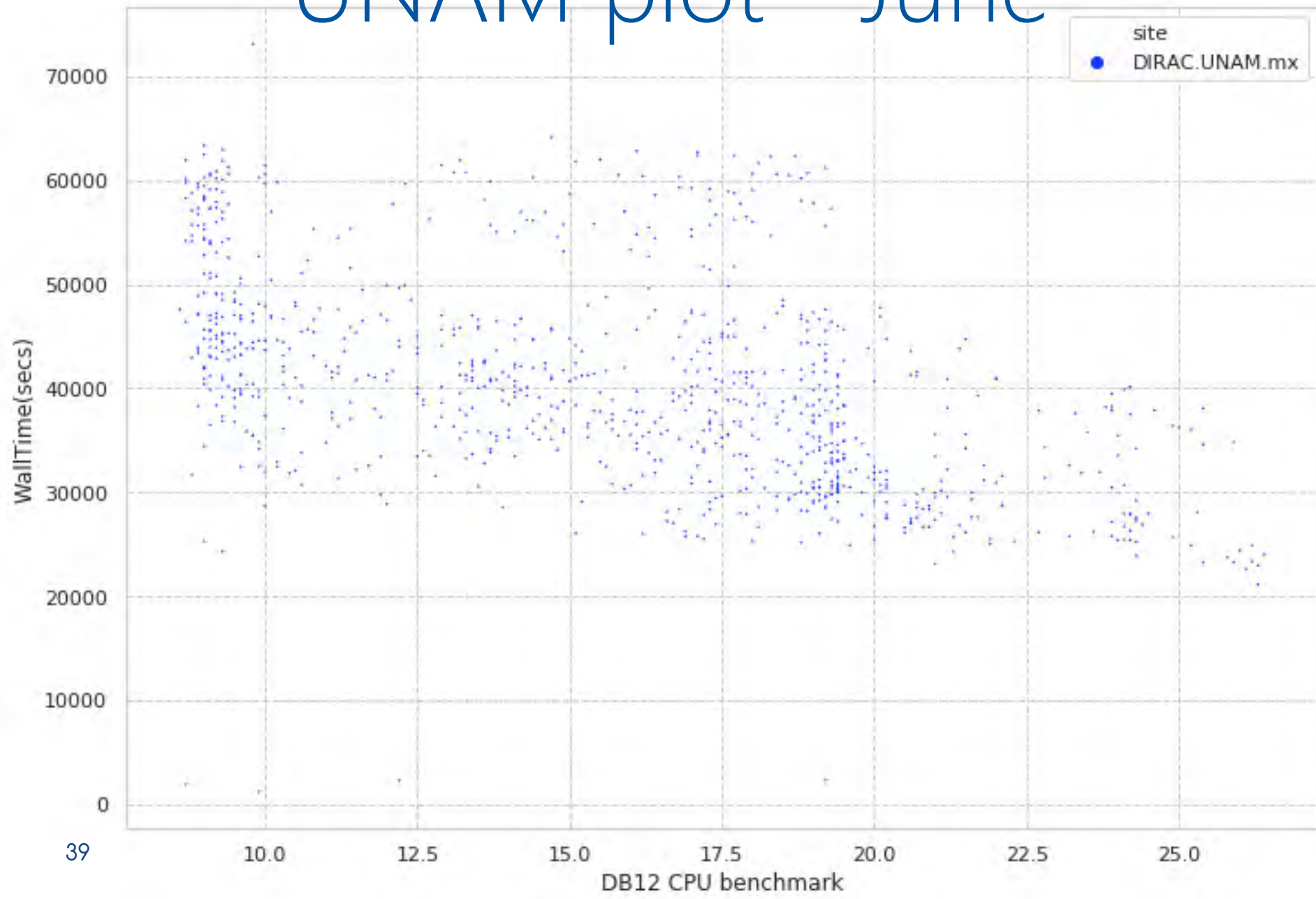
The all plot - June



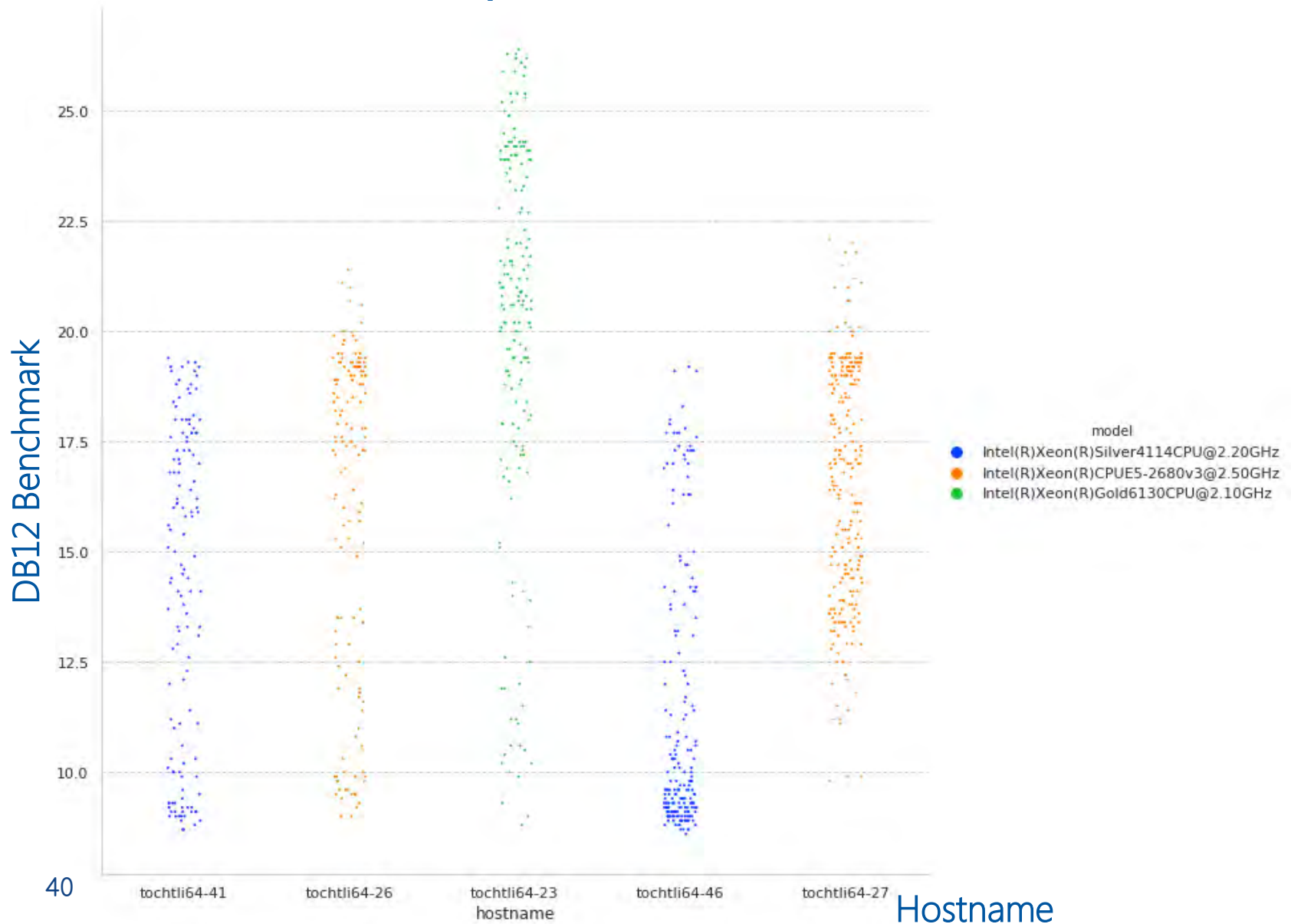
MPD plot - June



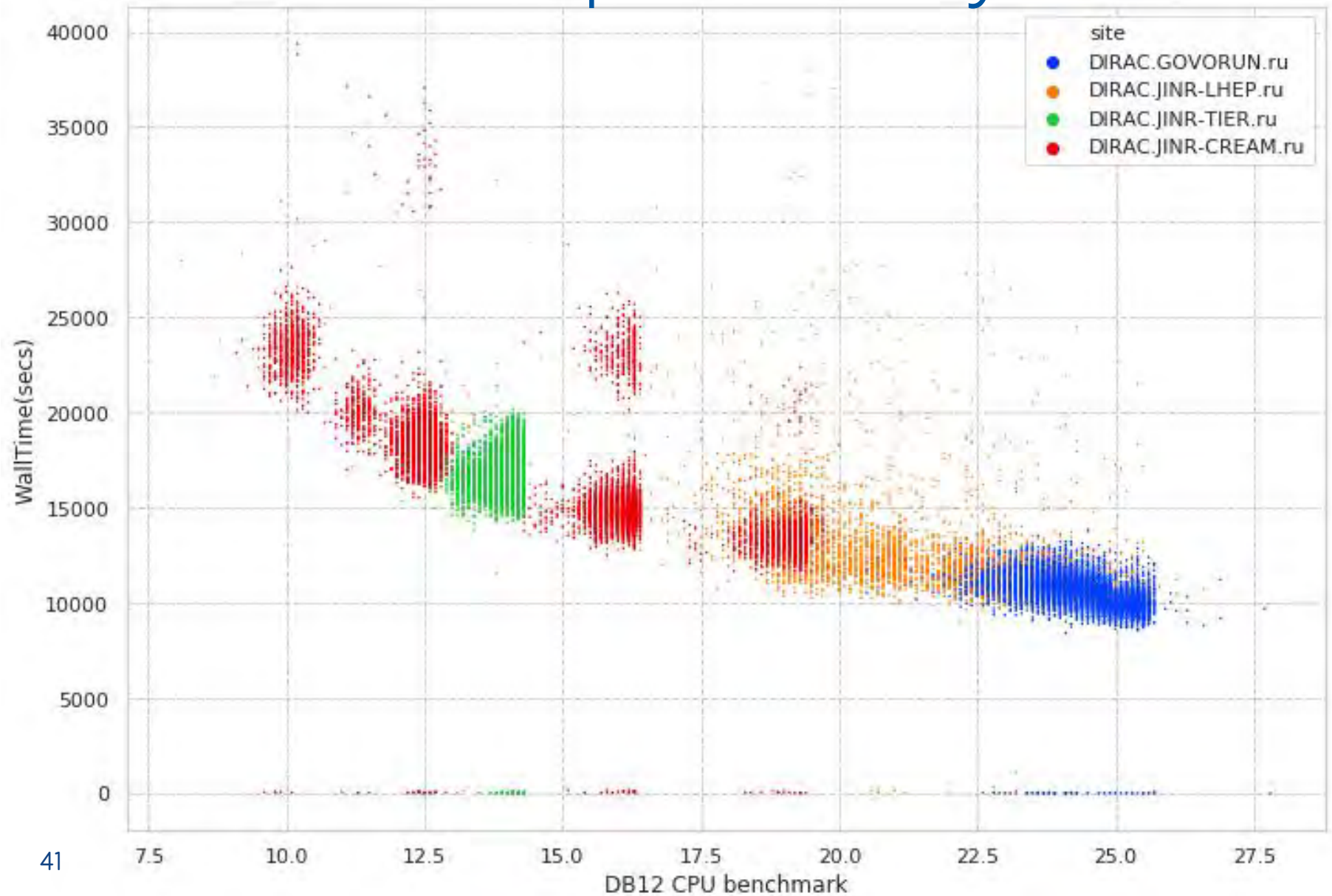
UNAM plot - June



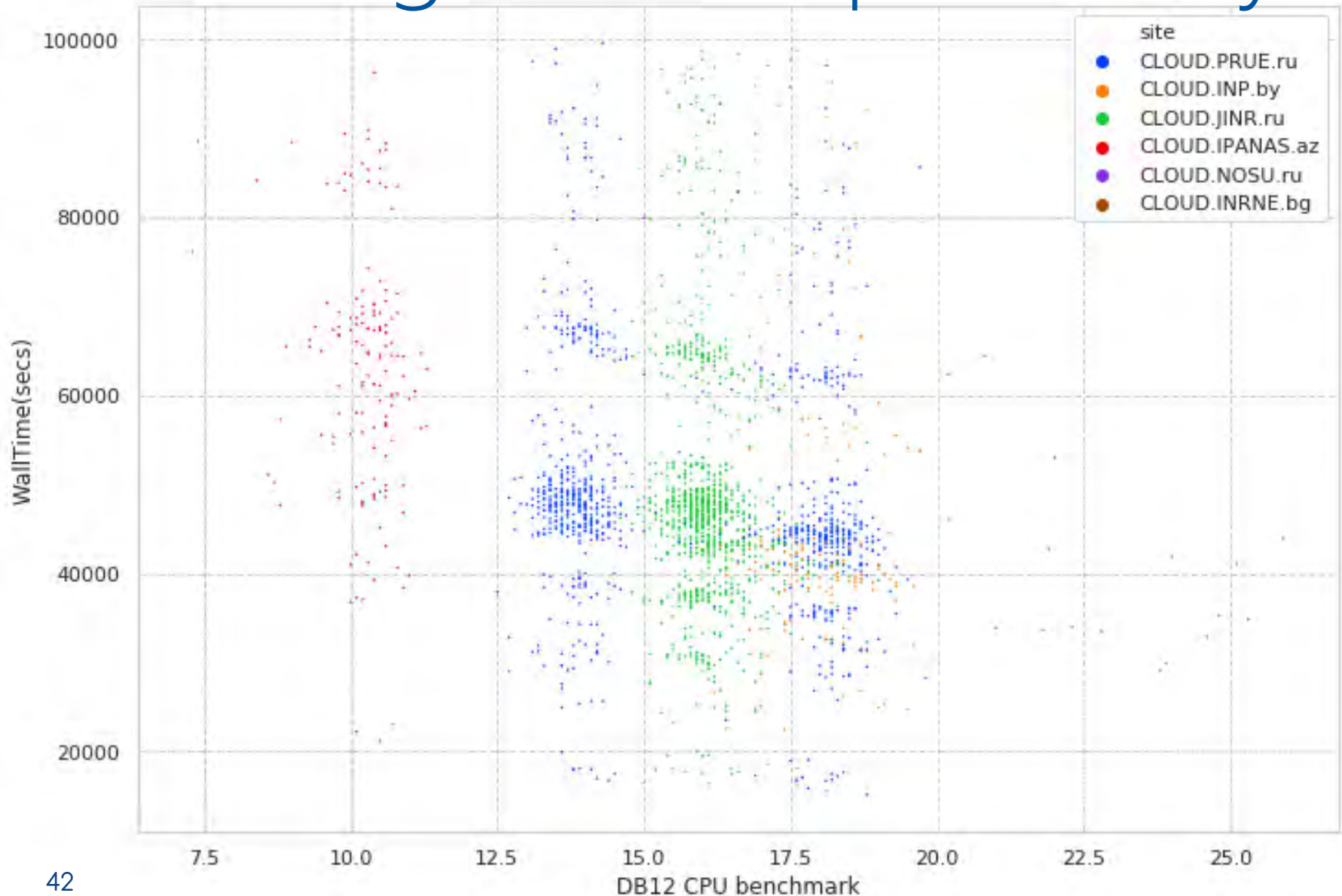
UNAM processors - June



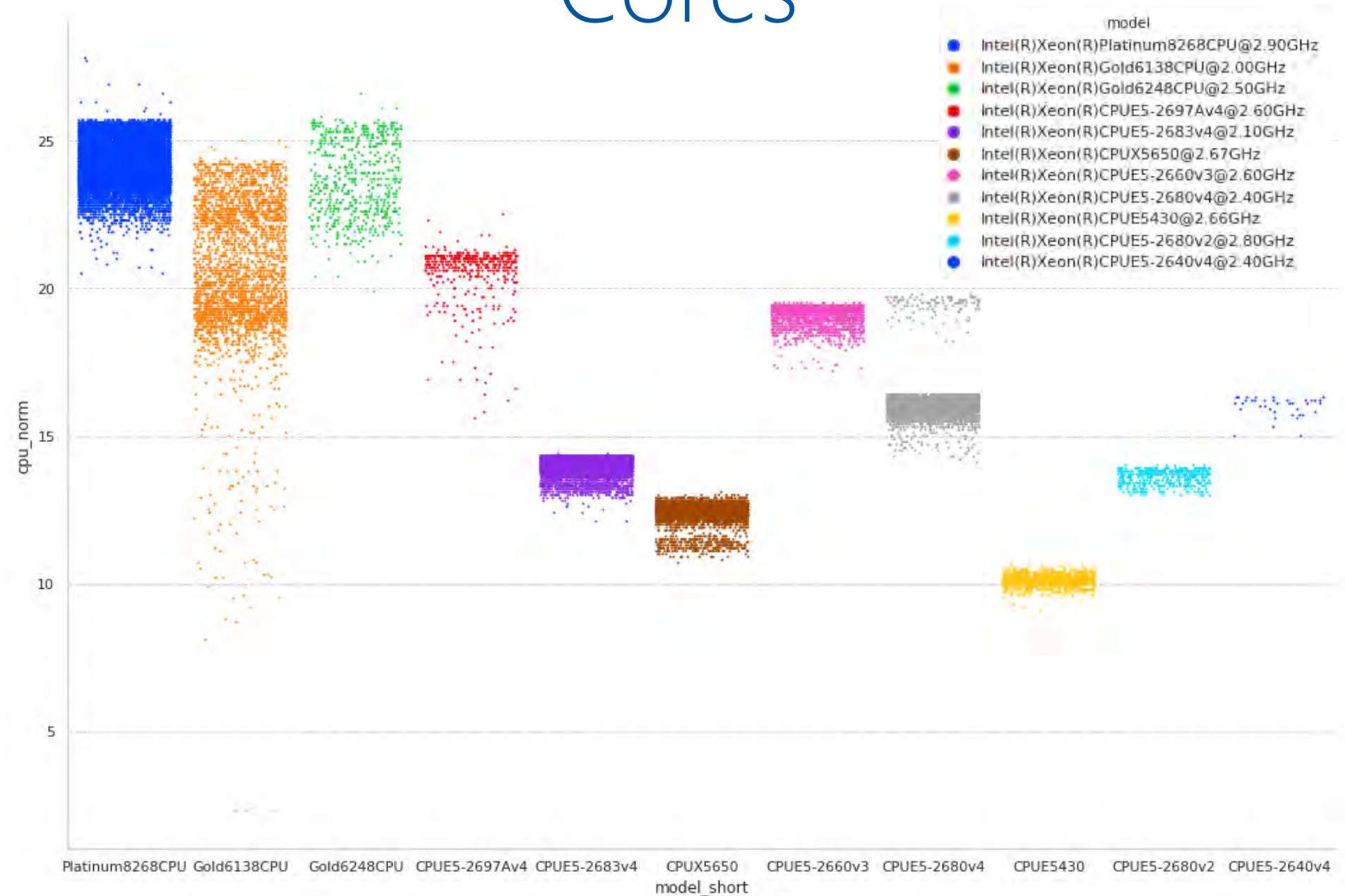
MPD plot - July



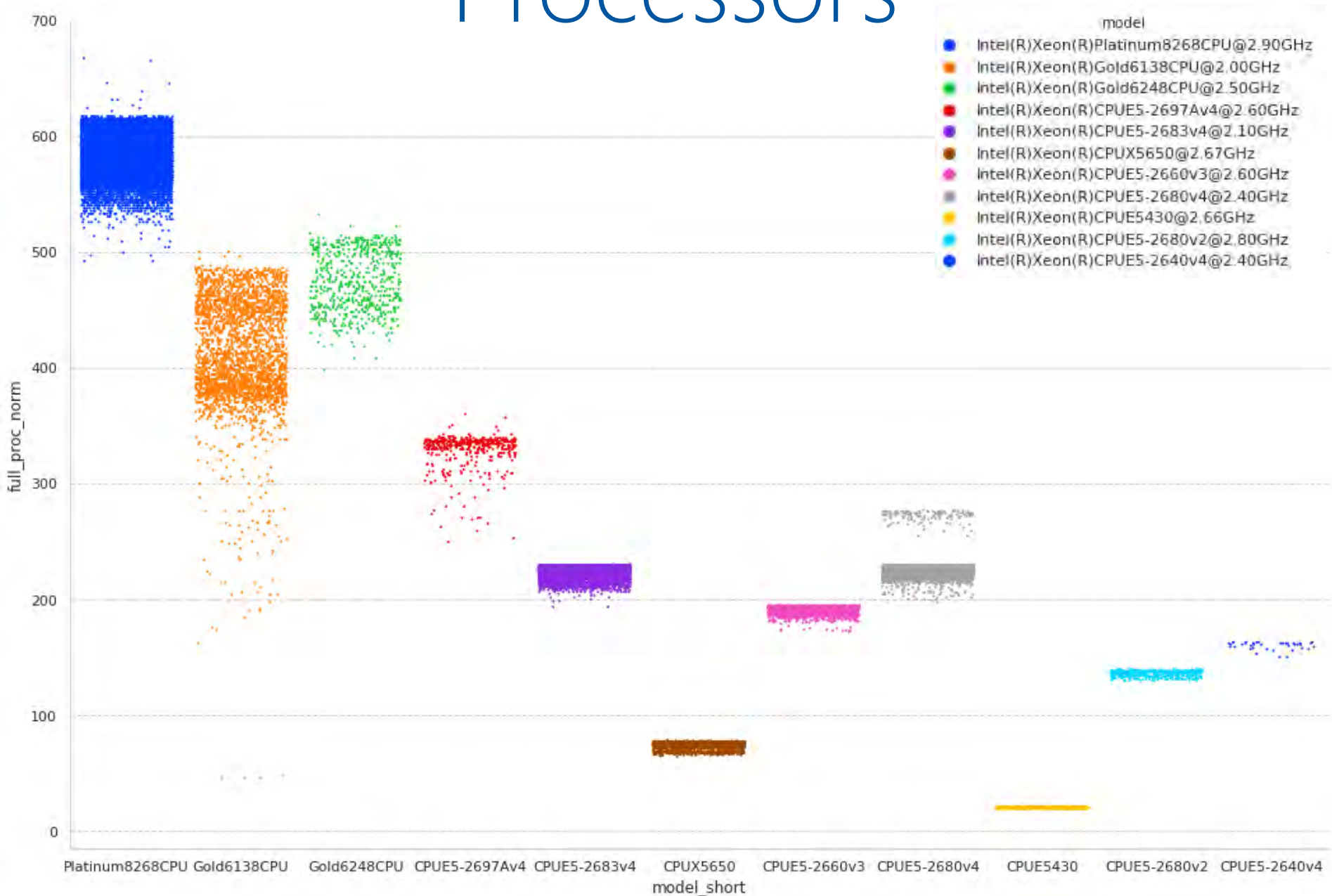
Folding@home plot - July



Cores

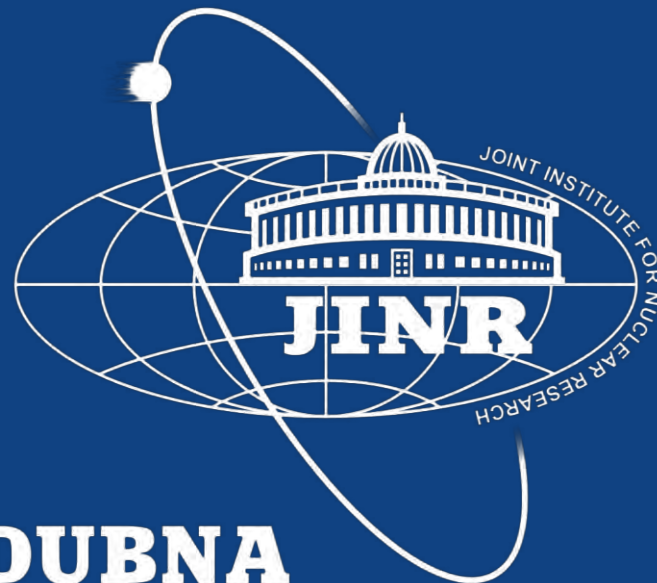


Processors



Conclusions

- Cooperation is a key.
- DIRAC appeared to be useful tool for running jobs over distributed resources.
- In JINR DIRAC performance was not a bottleneck at any time.
- Monte-Carlo generation for MPD is a success. It became possible thanks to cooperation of many people and teams.
- Folding@home on clouds looks like to be a success.
- When the system is operational, and users submit jobs, we may get intelligence about the performance, structure, components of heterogeneous resources “almost” “for free”.



DUBNA

LHCb DIRAC performance

