
JINR
MLiT

Distributed computing with DIRAC Interware

Igor Pelevanyuk, MLIT

14 November 2022

Throughput vs Performance

Throughput vs Performance

High Performance - Sharing the workload of interdependent processes over multiple cores

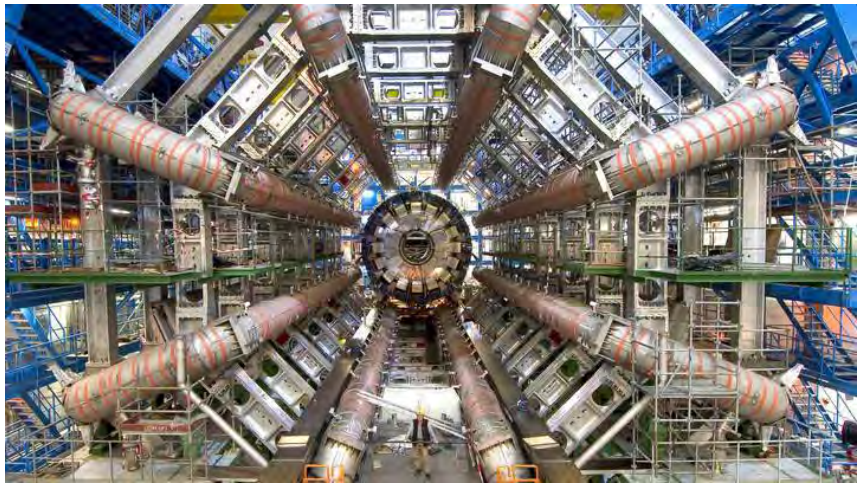
---

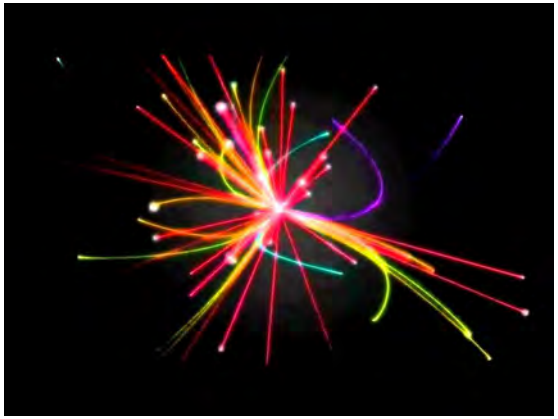
Focus of this talk

High Throughput - Many independent processes that can run in 1 or few cores on the same computer

Example of jobs: Monte-Carlo generation, Data reconstruction

Experiment



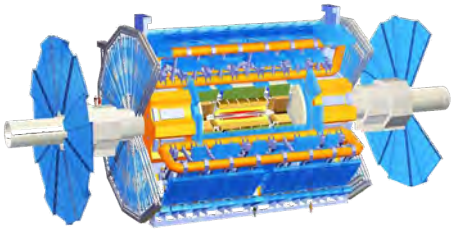


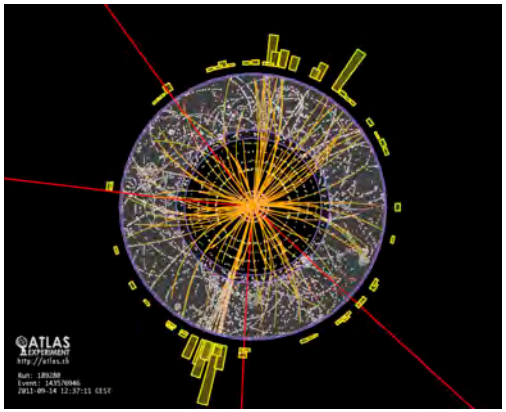
Raw data

```
CH0:0.001;
CH2:0.14;
CH4:0.34;
…
CH98039232:0.08;
```

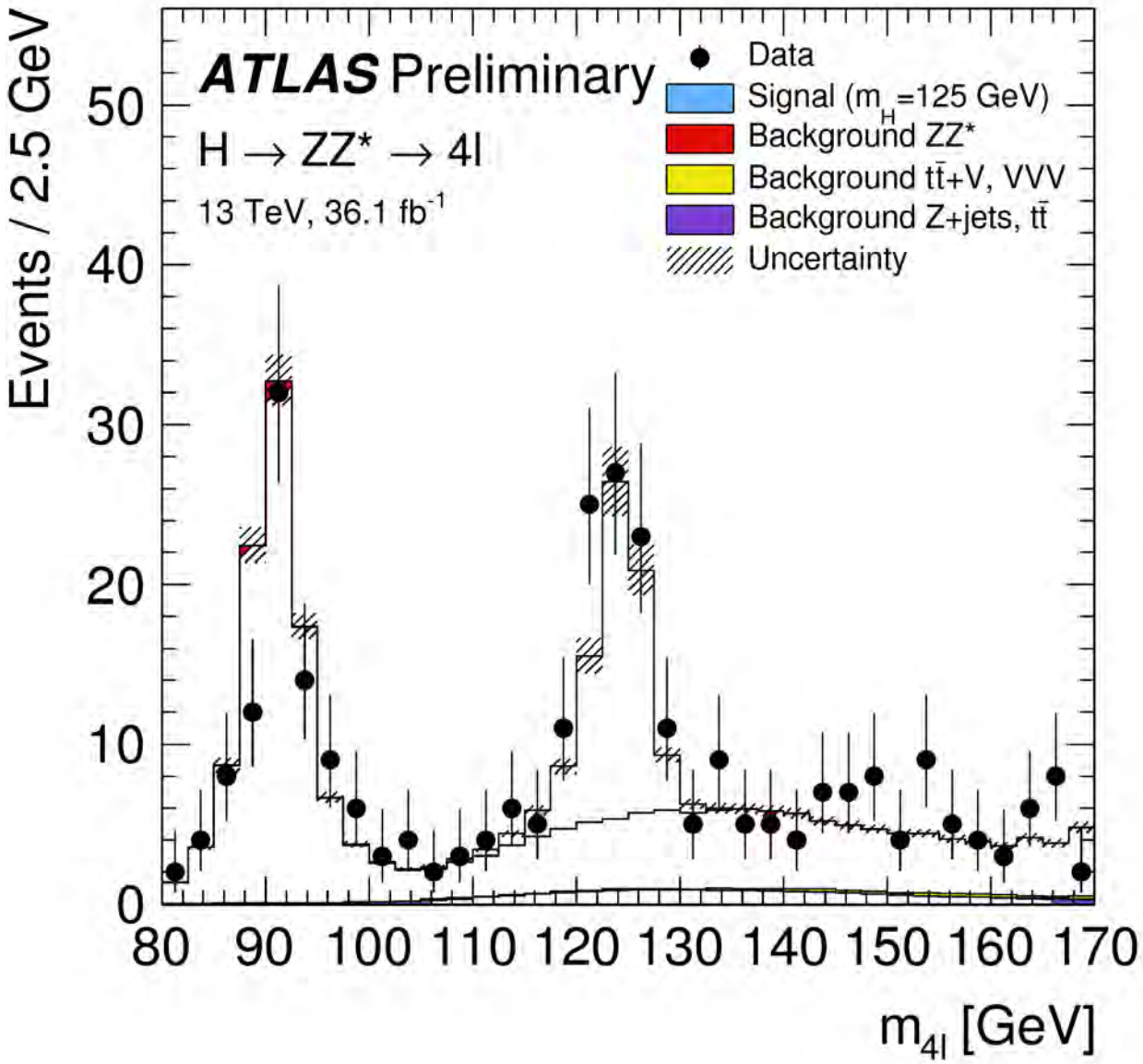
Reconstruction

```
CH0:0.001;
CH2:0.14;
CH4:0.34;
…
CH98039232:0.08;
```

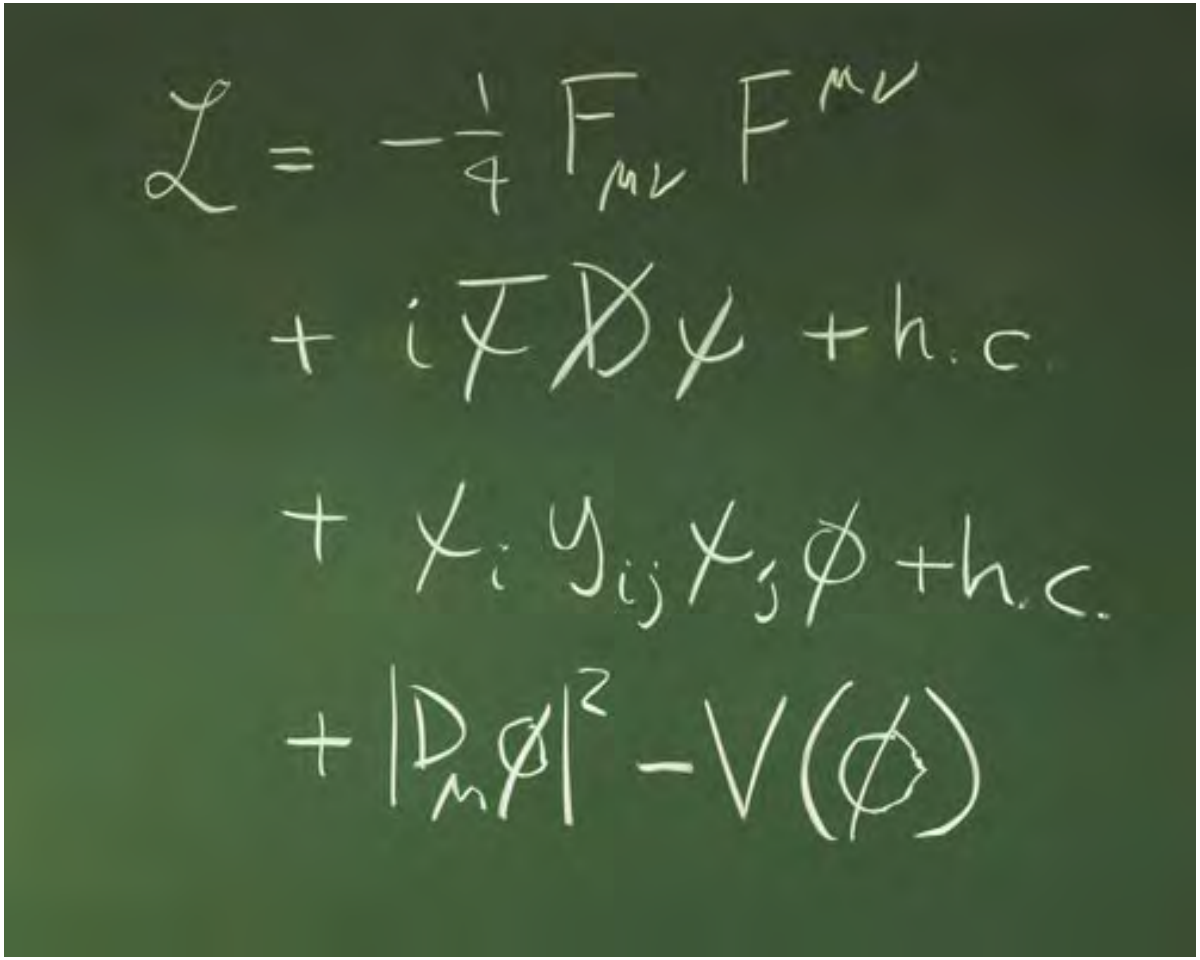
+  + Reconstruction algorithm

= 

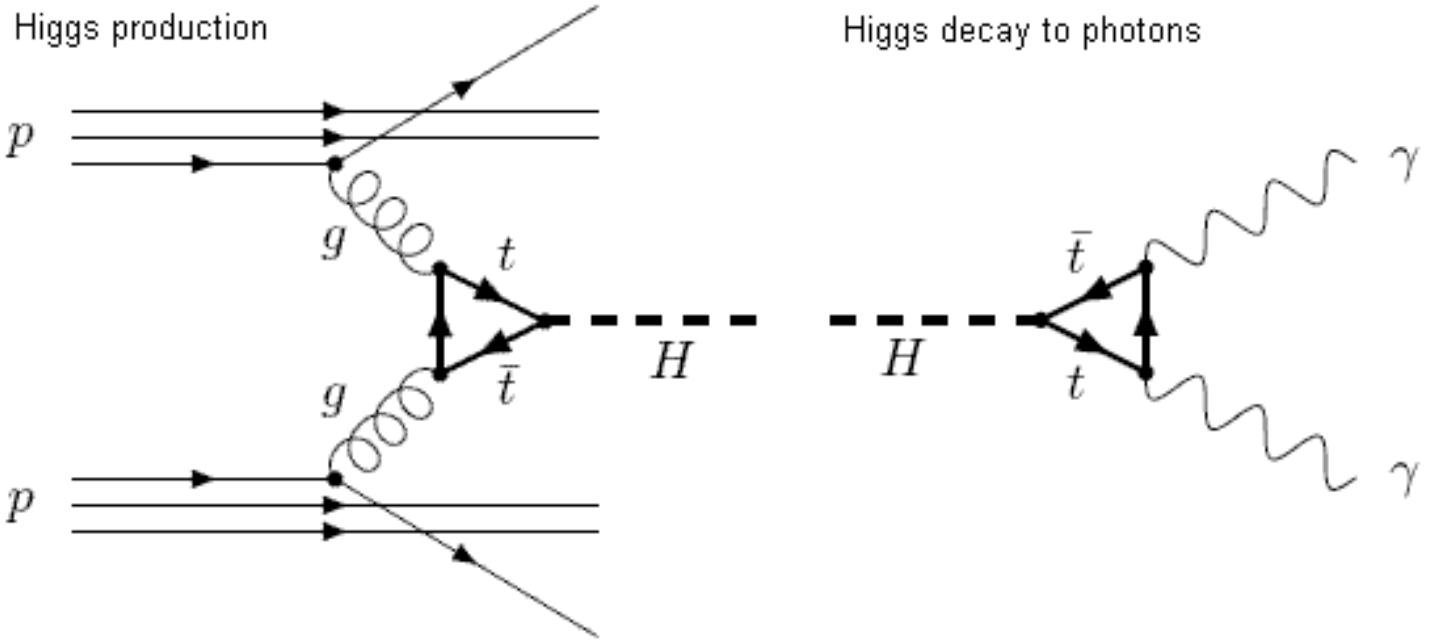
But that is not enough



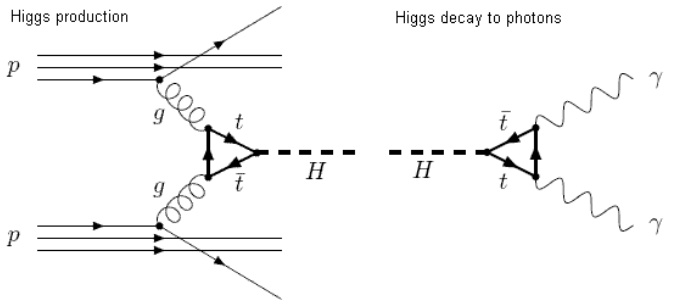
Model


$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu}$
$+ i\bar{\chi} \not{D} \chi + h.c.$
$+ \chi_i y_{ij} \chi_j \phi + h.c.$
$+ |D_\mu \phi|^2 - V(\phi)$

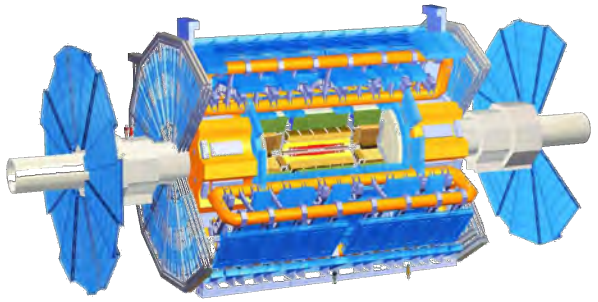
Event


Higgs production
Higgs decay to photons
p
p
g
g
t
$\bar{t}$
H
H
$\bar{t}$
t
γ
γ

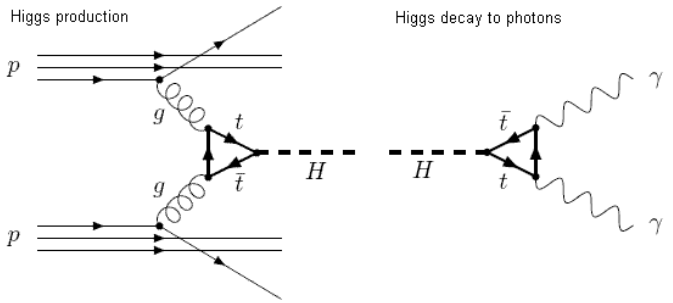
Simulation



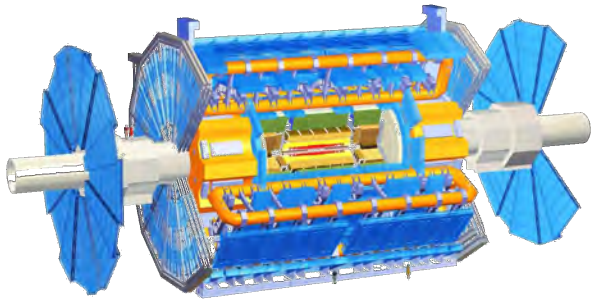
+



Simulation

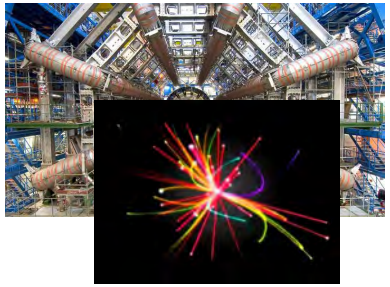


+

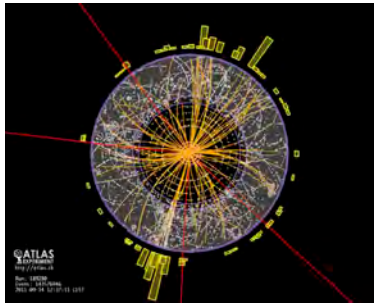


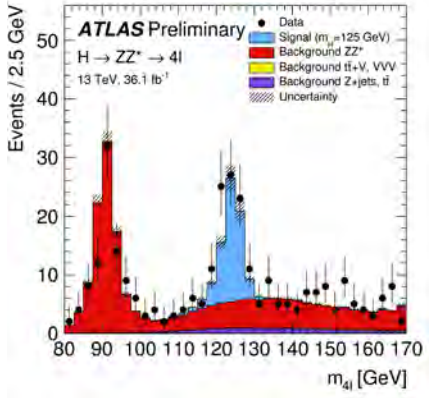
```
CH0:0.001;
CH2:0.14;
CH4:0.34;
…
CH98039232:0.08;
```

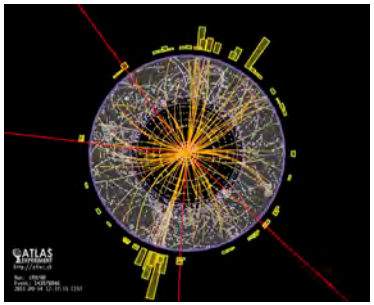
Full workflow

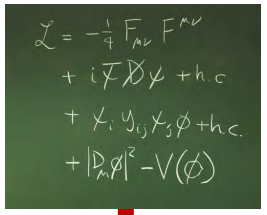


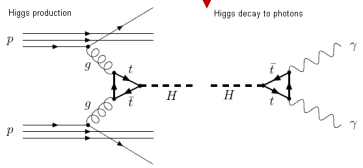
CH0:0.001;
CH2:0.14;
CH4:0.34;
…
CH98039232:0.08;





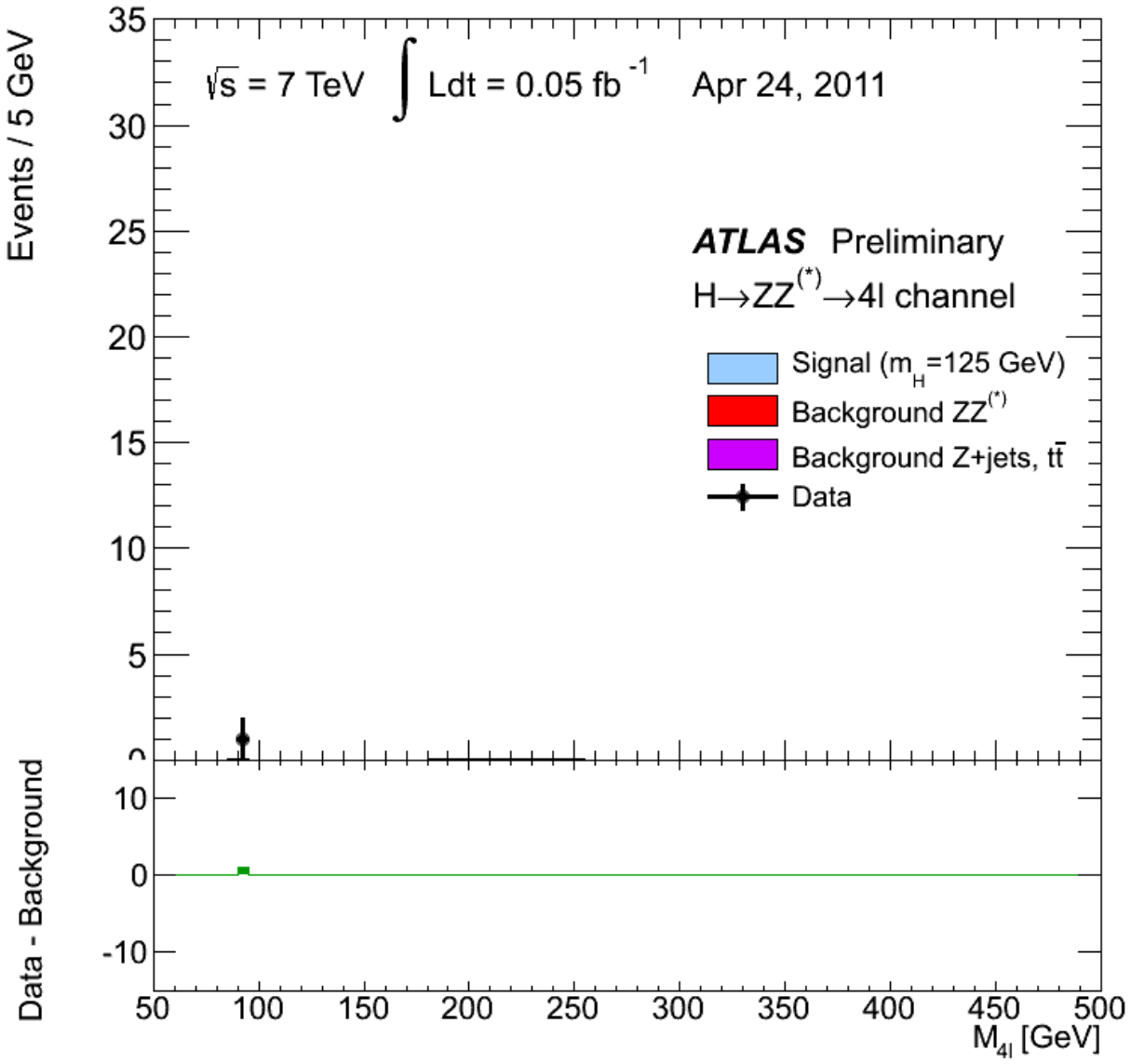






CH0:0.001;
CH2:0.14;
CH4:0.34;
…
CH98039232:0.08;

Result

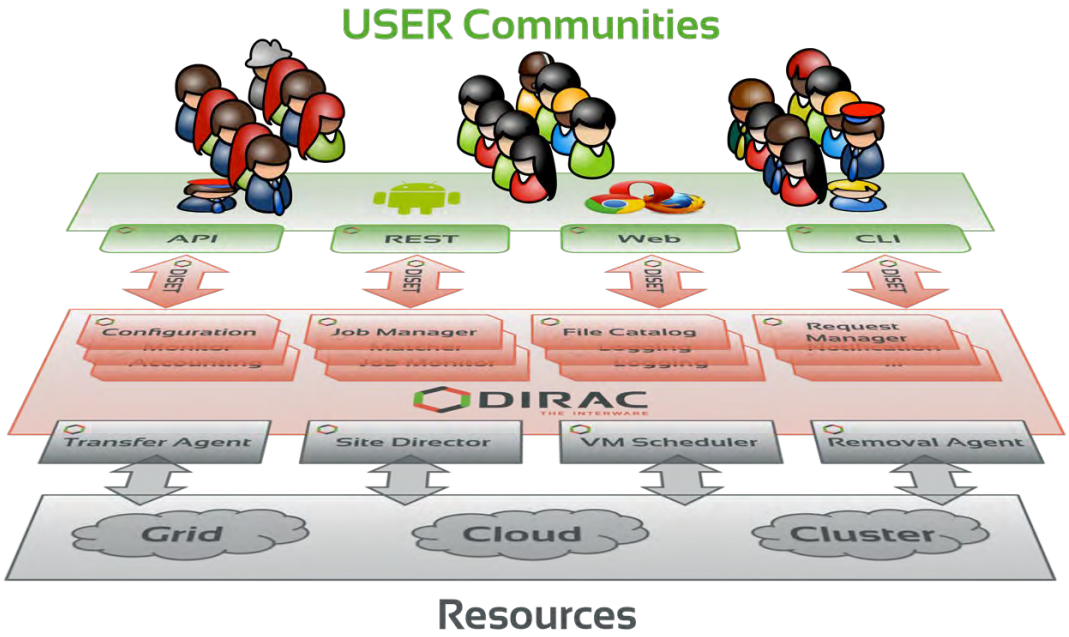

√s = 7 TeV ∫ Ldt = 0.05 fb-1 Apr 24, 2011
ATLAS Preliminary
H→ZZ(*)→4l channel
Signal (mH=125 GeV)
Background ZZ(*)
Background Z+jets, tt̄
Data
Events / 5 GeV
Data - Background
M4l [GeV]

Birth of DIRAC

- LHC experiments, all developed their own middleware
  - PanDA, AliEn, glideIn WMS, PhEDEx, …
- DIRAC is developed originally for the LHCb experiment
- The experience collected with a production grid system of a large HEP experiment is very valuable
  - Several new experiments expressed interest in using this software relying on its proven in practice utility
- In 2009 the core DIRAC development team decided to generalize the software to make it suitable for any user community.
- The results of this work allow to offer DIRAC as a general purpose distributed computing framework

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

Why DIRAC?

1. Single system for all aspects of computing

User Interface

API

Central configuration

Workload management

Data management

Integration tools

File Catalog

Workflow management

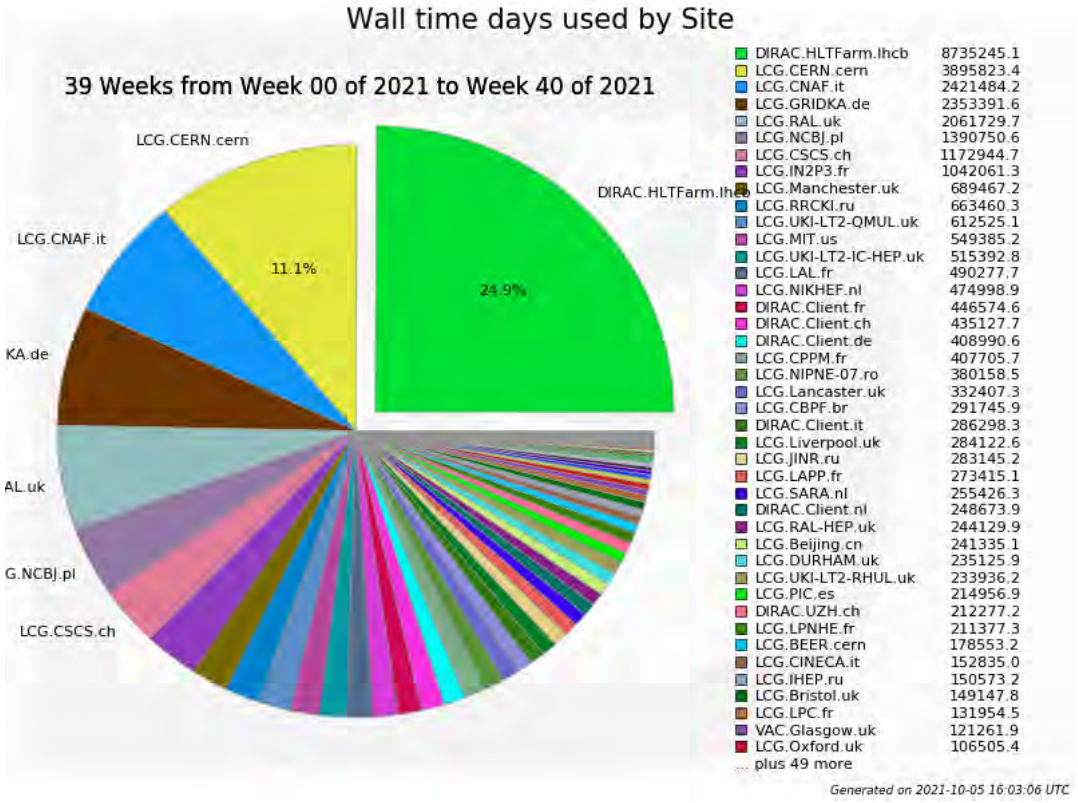
Metadata management

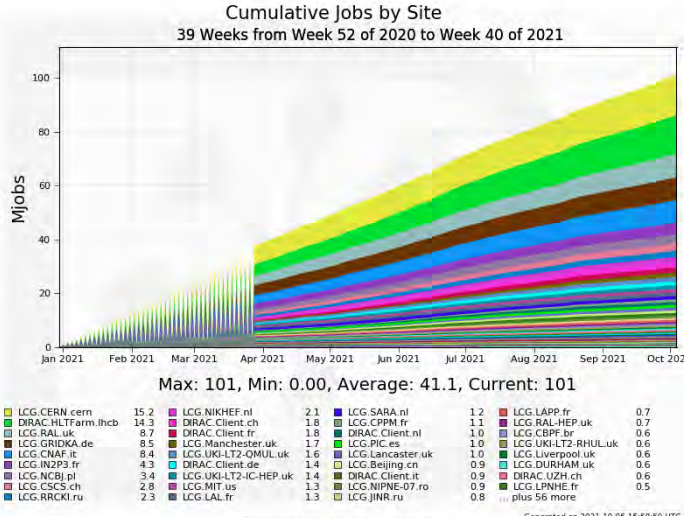
Accounting

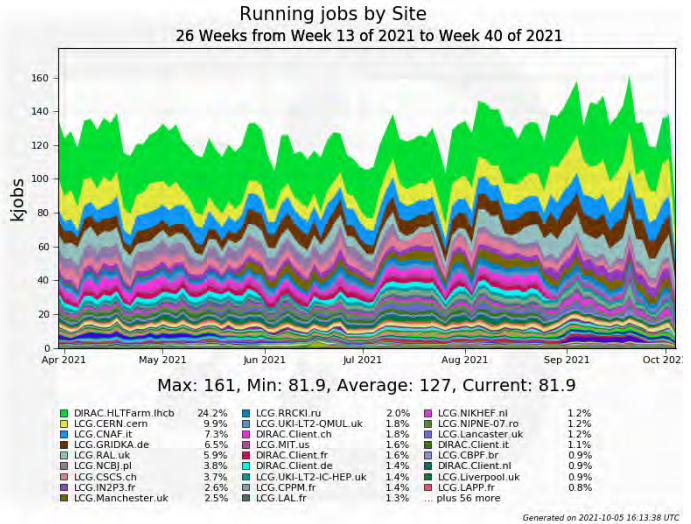
Management

Why DIRAC?

2. Good performance







Why DIRAC?

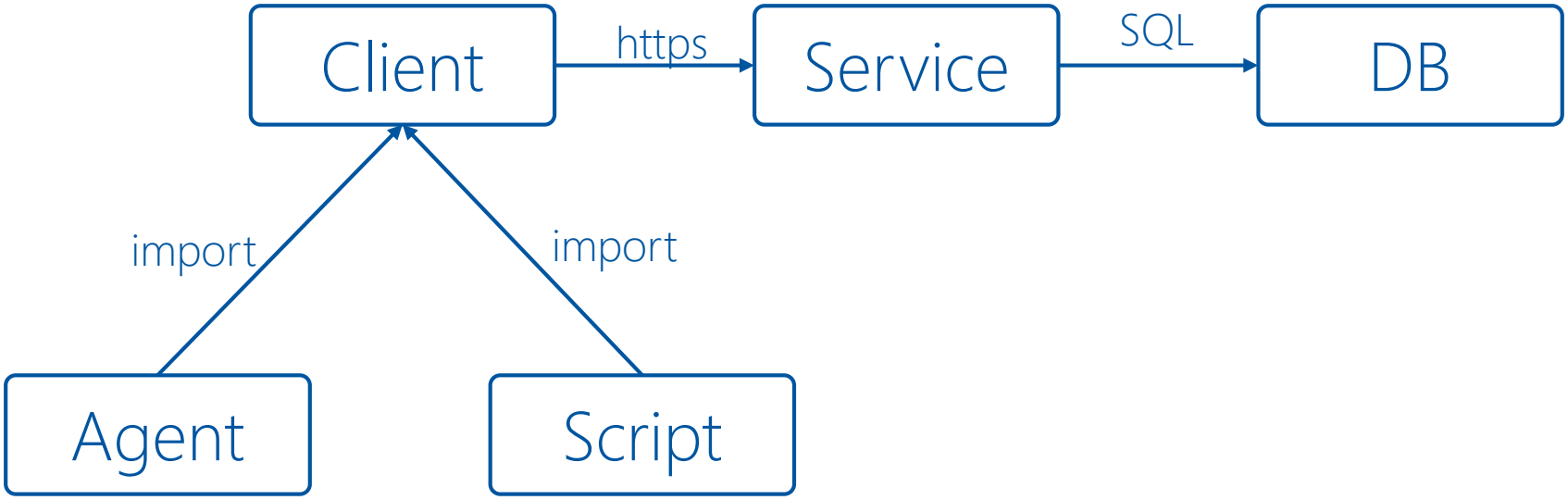
3. Active users and developers community

- Dedicated installations
  - LHCb, Belle II, CTA
- Multi-community services
  - ILC, CALICE
  - IHEP: BES III, Juno, CEPC
  - FG-DIRAC
  - GridPP
  - DIRAC4EGI
  - PNNL
  - DIRAC@JINR
  - DIRAC@CNAF
- Several DIRAC evaluations are ongoing
  - Auger, ELI, NICA, Virgo, LSST, …

What is DIRAC?

- DIRAC is a multi-agent system written in Python.
- DIRAC implements Service Oriented Architecture.
- Whole system is built with the following *bricks*:
  - Services
  - Clients
  - DBs
  - Agents
  - Scripts
- Repository: https://github.com/DIRACGrid

What is DIRAC?



DIRAC standard job workflow

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

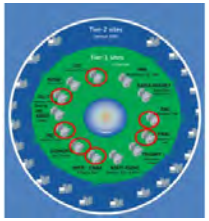
Job example

```
dirac-configure config.cfg

dirac-dms-get-file /bmn/raw/AuAu/data1002.raw

root –l –q –b reco.MC(“data1002.raw”)
#or
root –l –q –b reco.MC(/eos/nica/bmn/AuAu/data1002.raw)

dirac-dms-put-file /bmn/reco/AuAu/data1002.root \\
                                    data1002.root \\
                                    JINR-EOS-BMN

rm –f data1002.raw data1002.root
```

Workload management

Submit thousand of jobs to DIRAC Job Queue

DIRAC
THE INTERWARE


Tier-1


CICC/Tier-2


Clouds

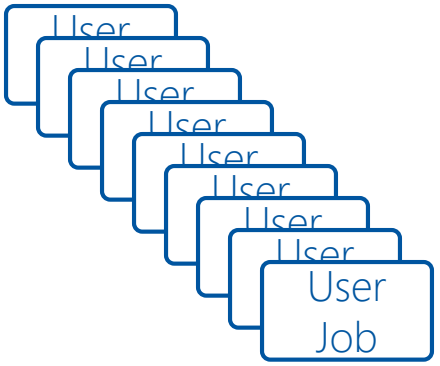

Govorun


NICA Cluster

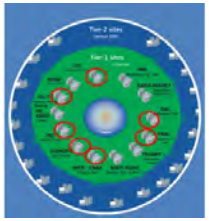

UNAM

Workload management

Submit thousand of jobs to DIRAC Job Queue



DIRAC
THE INTERWARE


Tier-1


CICC/Tier-2


Clouds

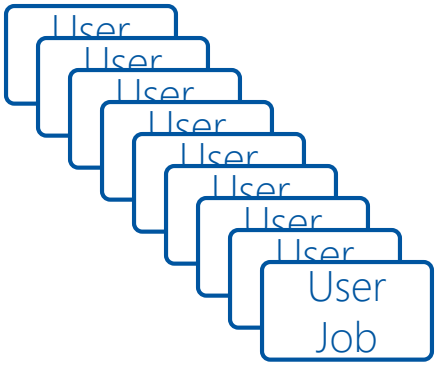

Govorun


NICA Cluster


UNAM

Workload management

Submit thousand of jobs to DIRAC Job Queue



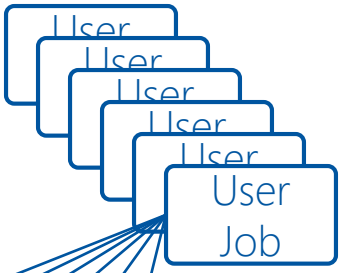
DIRAC
THE INTERWARE

Pilot job | Pilot job | Pilot job | Pilot job | Pilot job | Pilot job

Tier-1 | CICC/Tier-2 | Clouds | Govorun | NICA Cluster | UNAM

Workload management

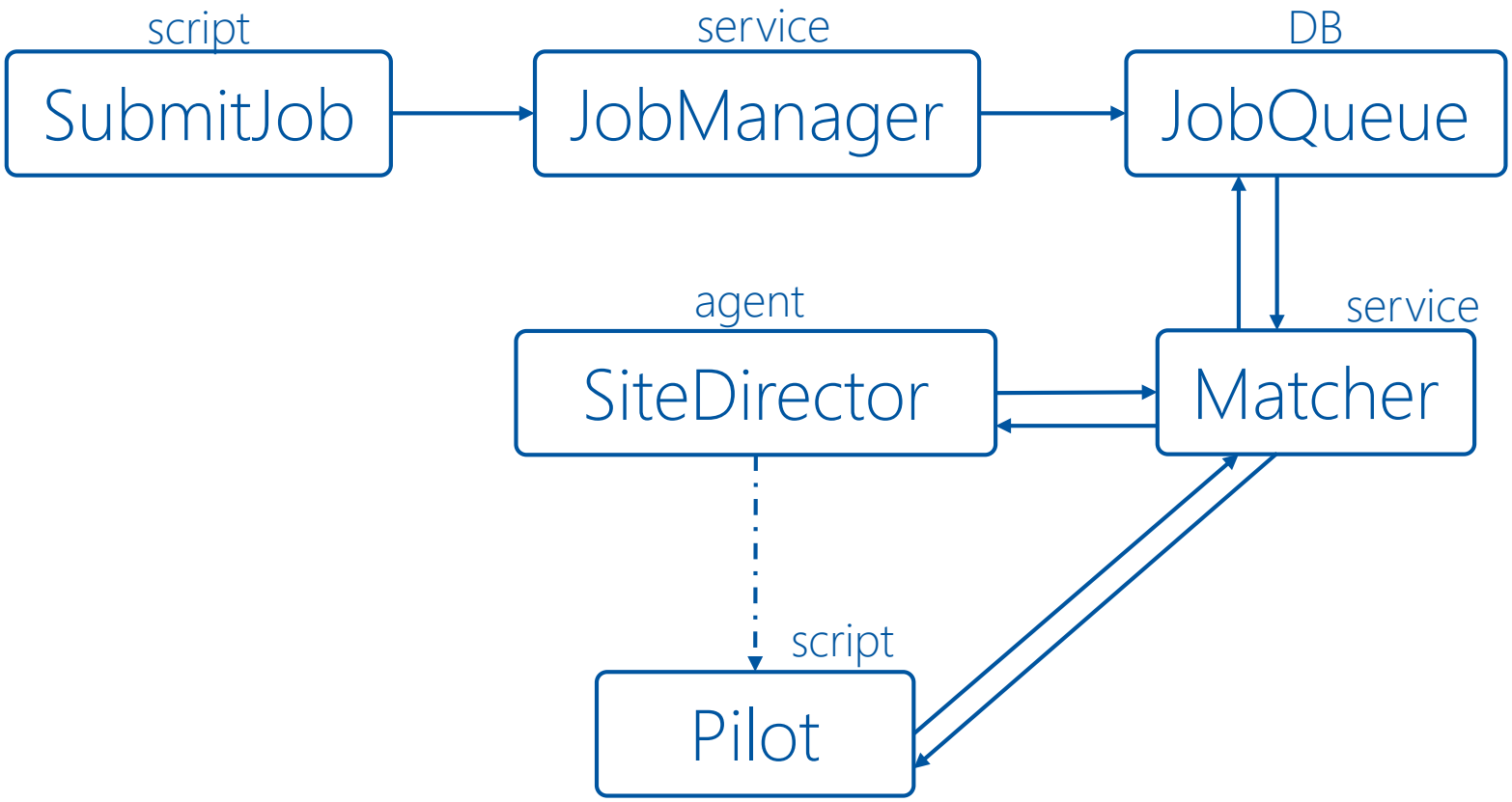
Submit thousand of jobs to DIRAC Job Queue



DIRAC
THE INTERWARE

Tier-1 | CICC/Tier-2 | Clouds | Govorun | NICA Cluster | UNAM

How to achieve that?



Computing resources

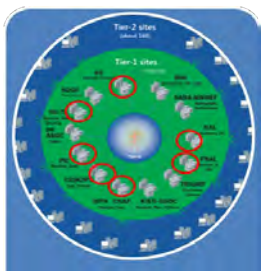
- Computing Elements
- Clouds
- Computing clusters (not included in any grid infrastructure)
  - Resources available at Universities and scientific laboratories
- High Performance Computing (HPC) Centers, or Supercomputers
- Volunteer Computing
  - Mostly based on BOINC technology
  - SETI@Home, LHC@Home, etc

JINR Main Resources

EOS LIT | **EOS HybriLIT** | **EOS LHEP**

Tapes Disks | EOS | lustre | EOS | EOS | ceph

root – transfer protocol
X509 – authentication
VOMS - authorization

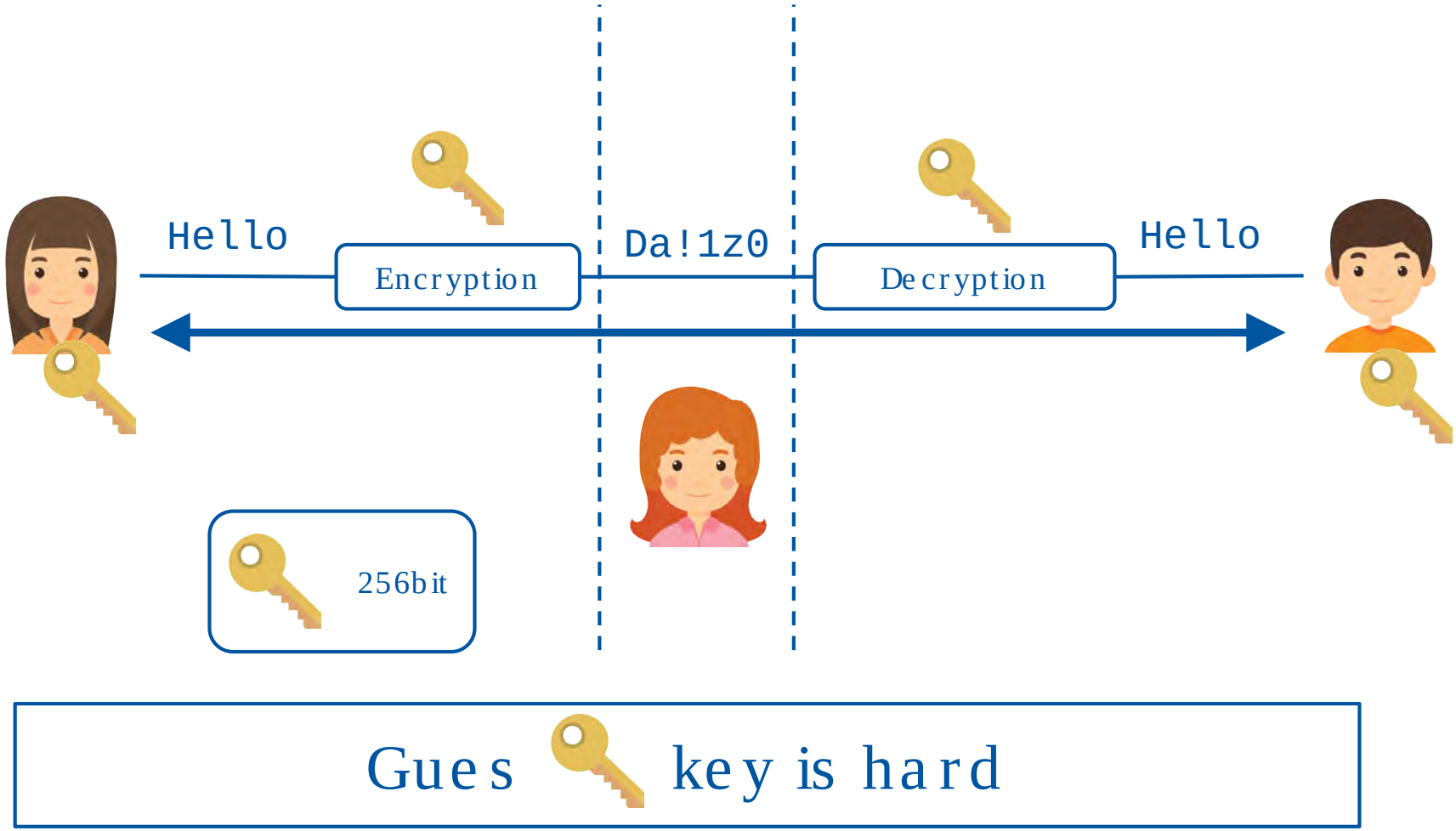





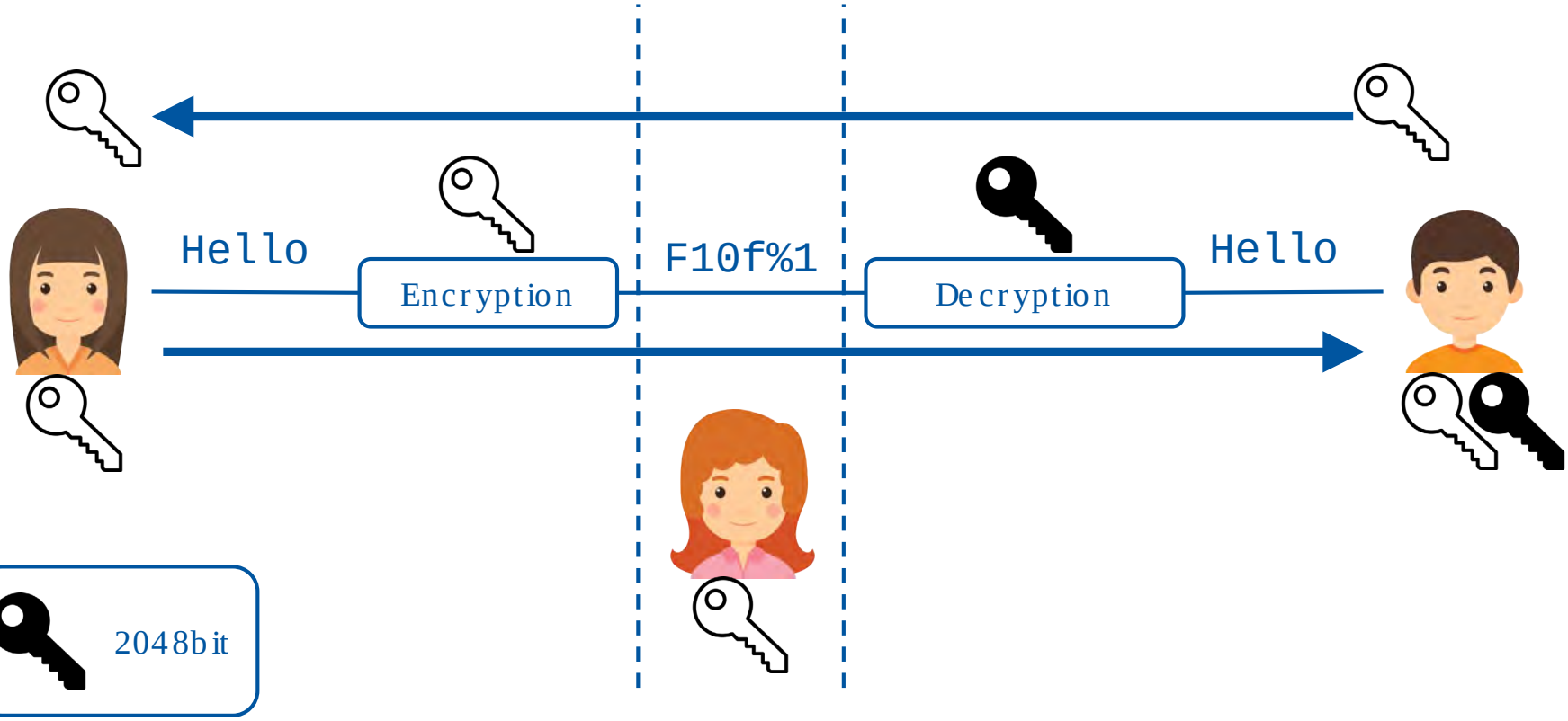
* Any computer with xrdcp, your certificate, or proxy of your certificate**

**How to pass credentials to processes to write?
Proxy is the answer!

Symmetric-key algorithm

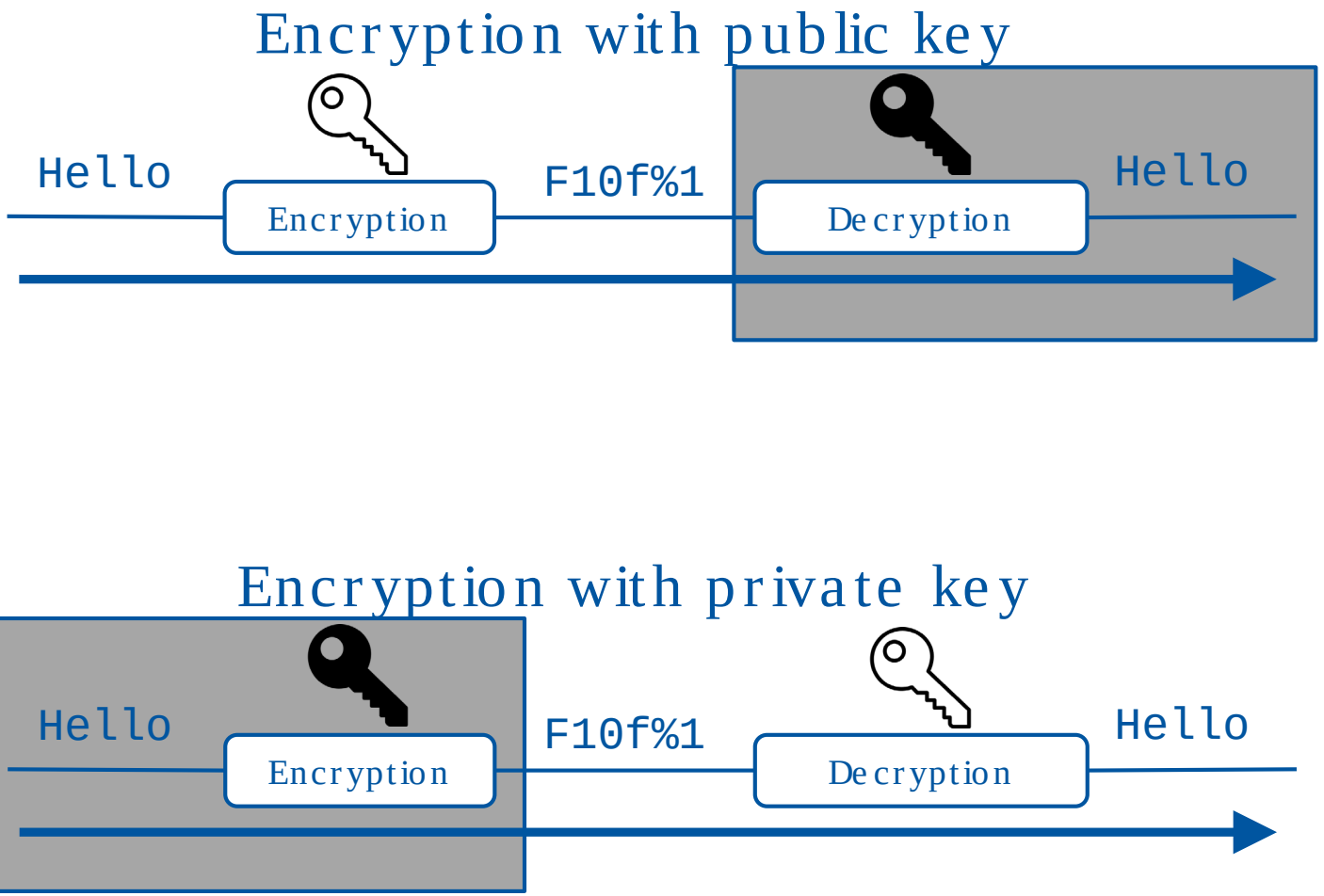


Asymmetric-key algorithm

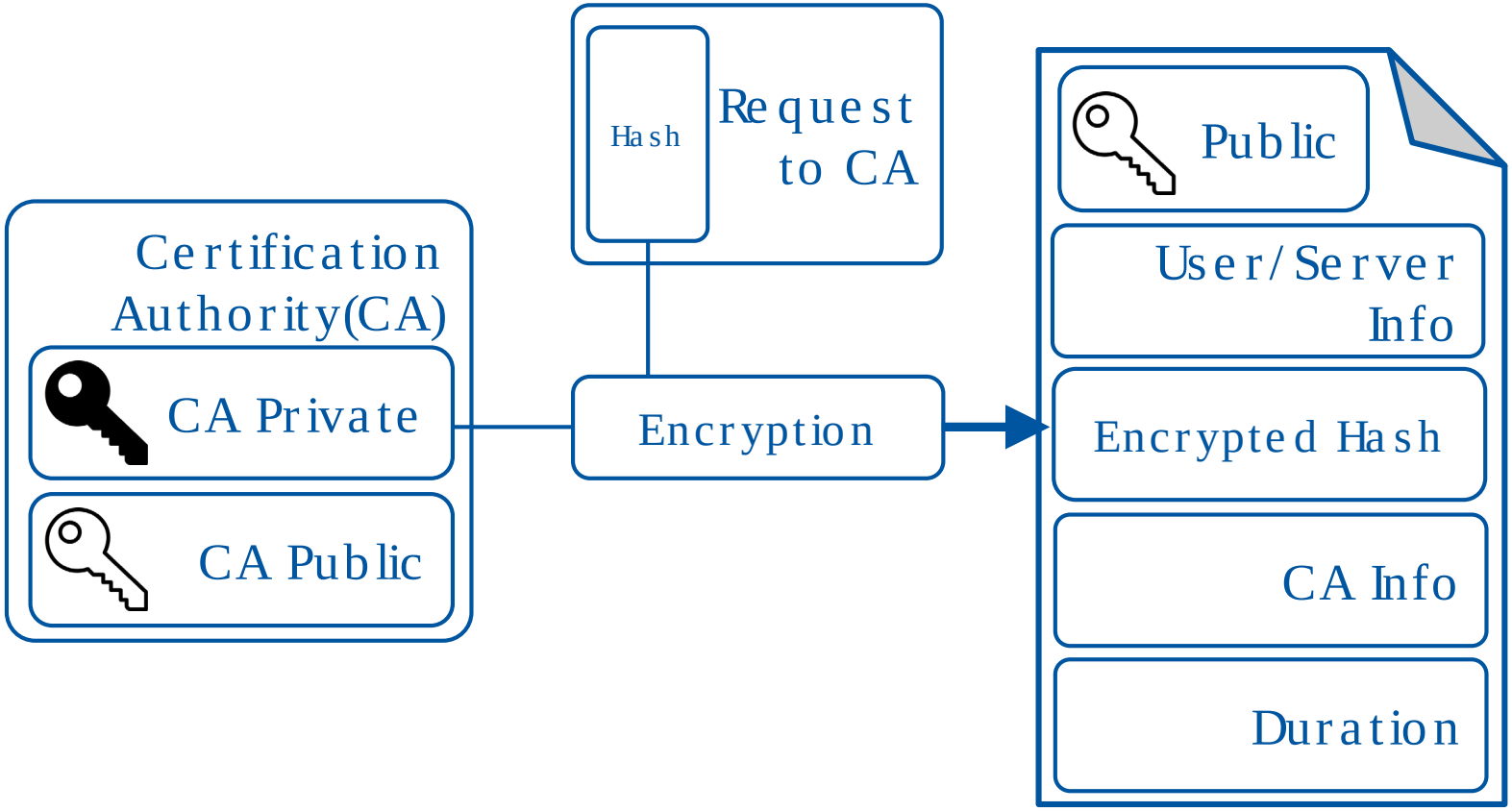


Gues key is hard

Asymmetric-key algorithm

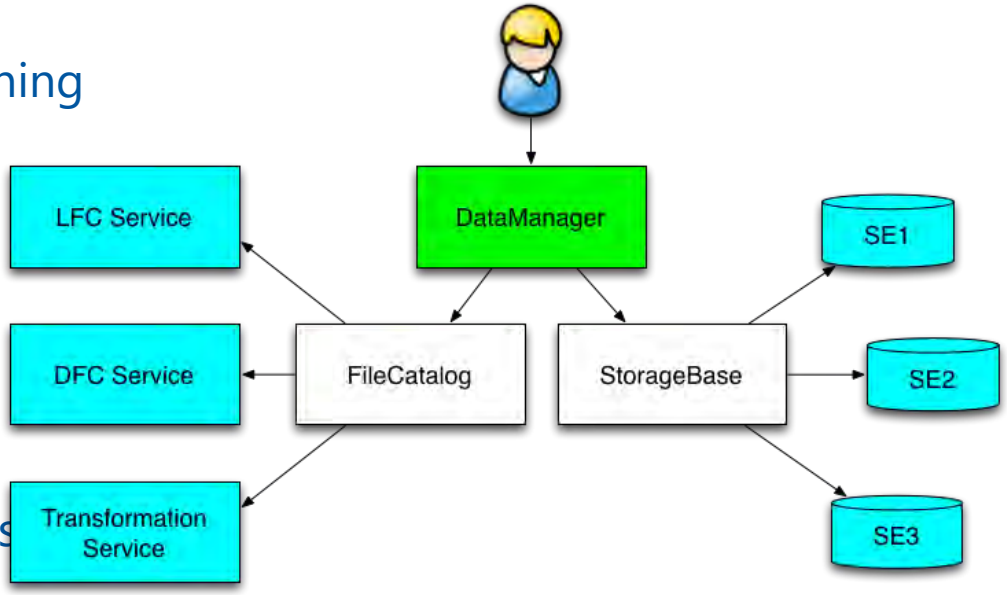


Making a certificate



Data Management

- 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



- Upload file(example):

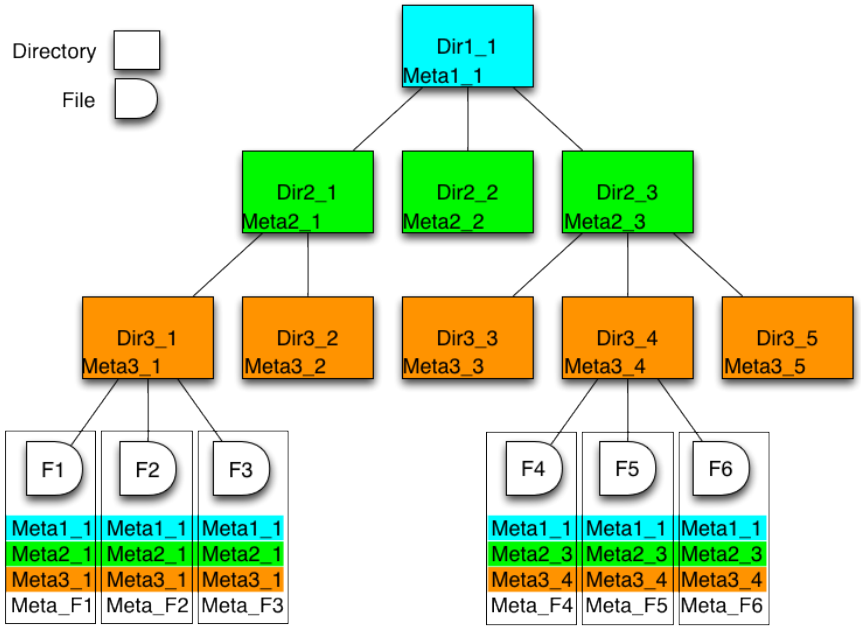
```
dirac-dms-add-file /mpd/vo/model/DQGSM/data.root ./data.roor EOS-MPD
```

- Download file(example):

```
dirac-dms-get-file /mpd/vo/model/DQGSM/data.root
```

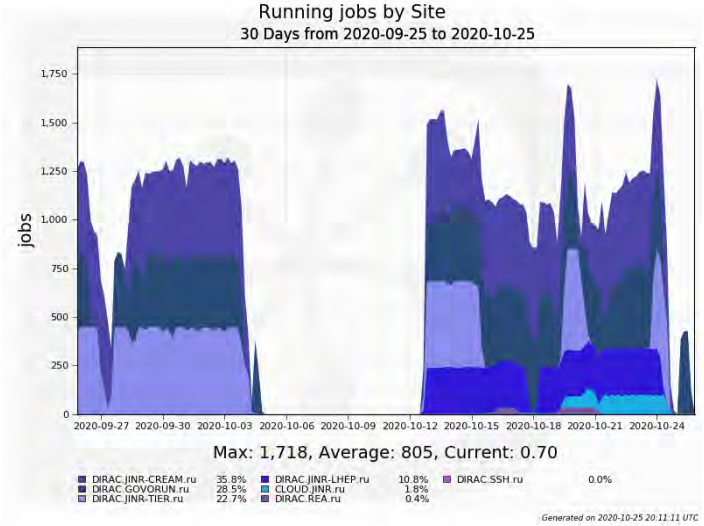
Meta Data

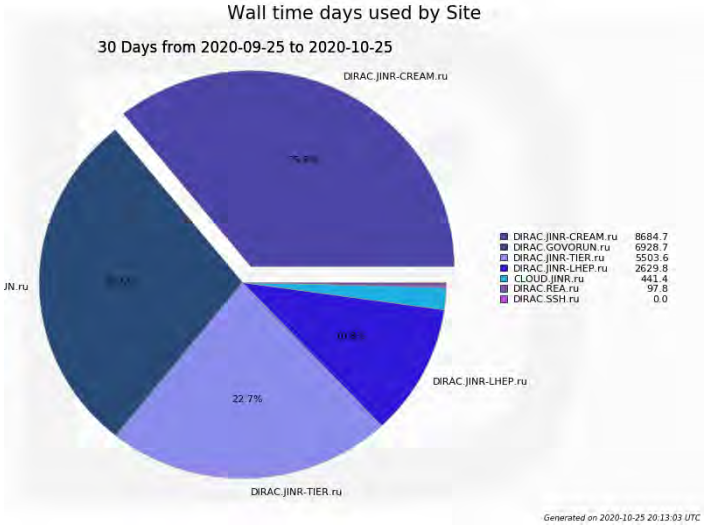
- DFC also may host Metadata
  - User defined metadata
  - The same hierarchy for metadata as for the logical name space
  - Metadata associated with files and directories
  - Allow for efficient searches
  - Efficient Storage Usage reports

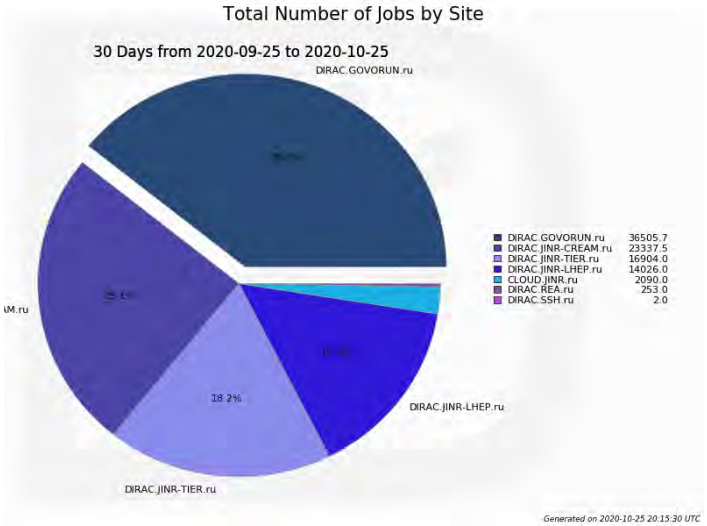


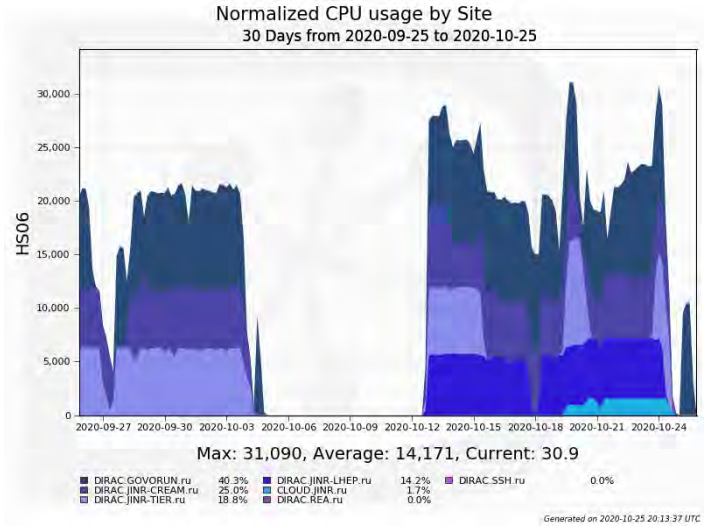

```
find /experiment/model LastAccess<01-10-2020
GaussVersion=v1,v2 SE=EOS-MPD Name=*.raw
```

Accounting


Running jobs by Site
30 Days from 2020-09-25 to 2020-10-25
jobs
1,750
1,500
1,250
1,000
750
500
250
0
2020-09-27 2020-09-30 2020-10-03 2020-10-06 2020-10-09 2020-10-12 2020-10-15 2020-10-18 2020-10-21 2020-10-24
Max: 1,718, Average: 805, Current: 0.70
DIRAC.JINR-CREAM.ru 35.8%
DIRAC.GOVORUN.ru 28.5%
DIRAC.JINR-TIER.ru 22.7%
DIRAC.JINR-LHEP.ru 10.8%
CLOUD.JINR.ru 1.8%
DIRAC.REA.ru 0.4%
DIRAC.SSH.ru 0.0%
Generated on 2020-10-25 20:11:11 UTC


Wall time days used by Site
30 Days from 2020-09-25 to 2020-10-25
DIRAC.JINR-CREAM.ru
DIRAC.JINR-LHEP.ru
22.7%
DIRAC.JINR-TIER.ru
DIRAC.JINR-CREAM.ru 8684.7
DIRAC.GOVORUN.ru 6928.7
DIRAC.JINR-TIER.ru 5503.6
DIRAC.JINR-LHEP.ru 2629.8
CLOUD.JINR.ru 441.4
DIRAC.REA.ru 97.8
DIRAC.SSH.ru 0.0
Generated on 2020-10-25 20:13:03 UTC


Total Number of Jobs by Site
30 Days from 2020-09-25 to 2020-10-25
DIRAC.GOVORUN.ru
18.2%
DIRAC.JINR-LHEP.ru
DIRAC.JINR-TIER.ru
DIRAC.GOVORUN.ru 36505.7
DIRAC.JINR-CREAM.ru 23337.5
DIRAC.JINR-TIER.ru 16904.0
DIRAC.JINR-LHEP.ru 14026.0
CLOUD.JINR.ru 2090.0
DIRAC.REA.ru 253.0
DIRAC.SSH.ru 2.0
Generated on 2020-10-25 20:15:30 UTC


Normalized CPU usage by Site
30 Days from 2020-09-25 to 2020-10-25
HS06
30,000
25,000
20,000
15,000
10,000
5,000
0
2020-09-27 2020-09-30 2020-10-03 2020-10-06 2020-10-09 2020-10-12 2020-10-15 2020-10-18 2020-10-21 2020-10-24
Max: 31,090, Average: 14,171, Current: 30.9
DIRAC.GOVORUN.ru 40.3%
DIRAC.JINR-CREAM.ru 25.0%
DIRAC.JINR-TIER.ru 18.8%
DIRAC.JINR-LHEP.ru 14.2%
CLOUD.JINR.ru 1.7%
DIRAC.REA.ru 0.0%
DIRAC.SSH.ru 0.0%
Generated on 2020-10-25 20:13:37 UTC

MultiVO support

Installing, configuring and supporting DIRAC requires a lot of efforts.

CEPC* and JUNO** have joined to BES-III computing infrastructure.

*Circular Electron Positron Collider

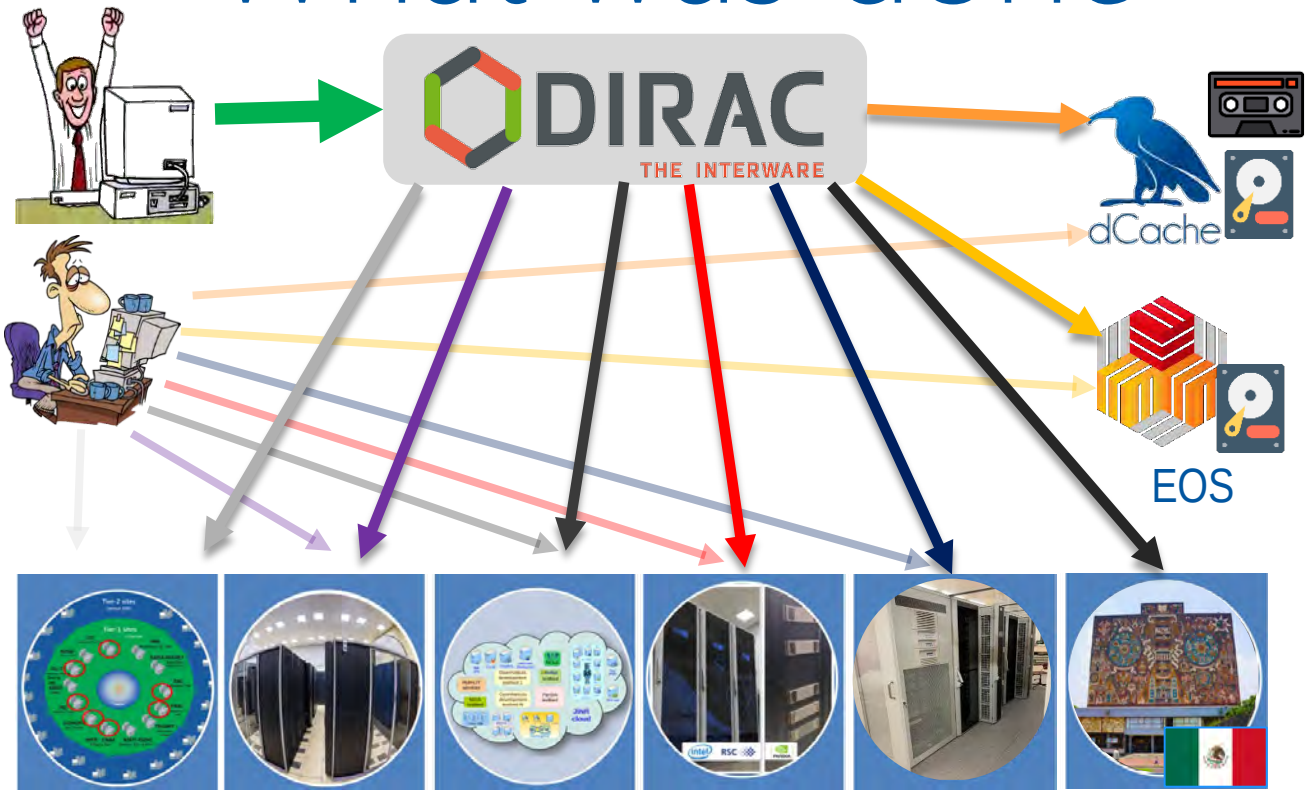
**Jiangmen Underground Neutrino Observatory

Resource utilization

Support needs

Feature development

What was done



Tier-1	CICC/Tier-2	Clouds	Govorun	NICA Cluster	UNAM
Running	**Running**	**Running**	**Running**	**Running**	**Running**

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.

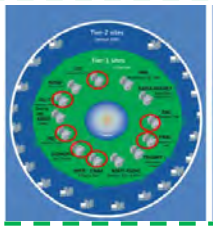
History of DIRAC at JINR

2013 – Development of monitoring system for BES-III installation. First tries to setup and configure DIRAC infrastructure.

2017 – DIRAC Interware installed; basic configuration done. Used for educational purposes. dCache storage integrated, Tier2 integrated.
2018 – HybriLIT integrated. JINR cloud integrated using OCCI protocol. Tests of full cycle of Monte-Carlo for BM@N were performed.
2019 – Clouds of JINR Member-States integrated by module developed in JINR. MPD starts using DIRAC for massive Monte-Carlo production. Tier1, Govorun and EOS integrated in DIRAC.
2020 – Folding@Home jobs submitted to clouds via DIRAC. Baikal-GVD jobs submitted to JINR and PRUE clouds.
2021 – First tests for SPD Monte-Carlo successfully done. First million jobs done!
2022– Total walltime exceeds 1000 years. DIRAC in JINR updated to use Python 3.

Computing Resources

JINR


Tier-1
920 slots


CICC/Tier-2
1000 slots


Cloud
80 slots


Govorun
184-870 slots


NICA Cluster
250 slots


UNAM
100 slots

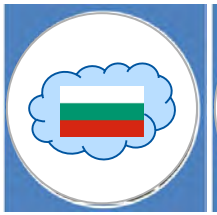
MPD
collaboration

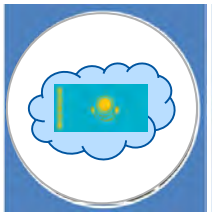
Clouds


IPANAS
16 slots


INP
132 slots


SU
48 slots


INRNE
20 slots


INP
50 slots


REA Plehanova
132 slots


NOSU
84 slots


STI-SCI
98 slots

Total amount of cores exceeds 3000

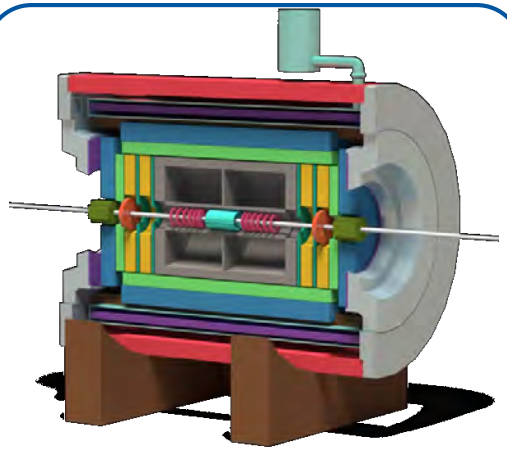
Steps

1. DIRAC setup, configuration, development and tuning

2. Integration of computing and storage resources

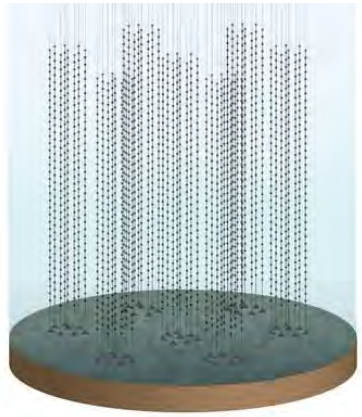
3. Elaboration of approaches for effective use of resources

What do we use DIRAC for?



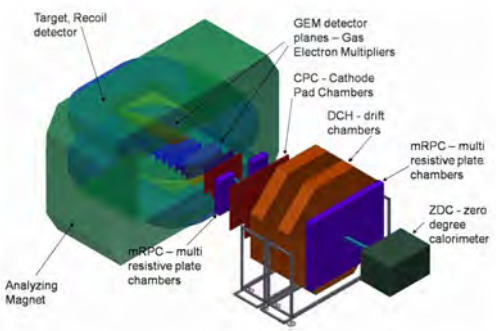
MPD@NICA

Monte-Carlo – Real
Analysis – Maybe



Baikal-GVD

Monte-Carlo – Real



BM@N

Monte-Carlo – Real
Reconstruction – Tests

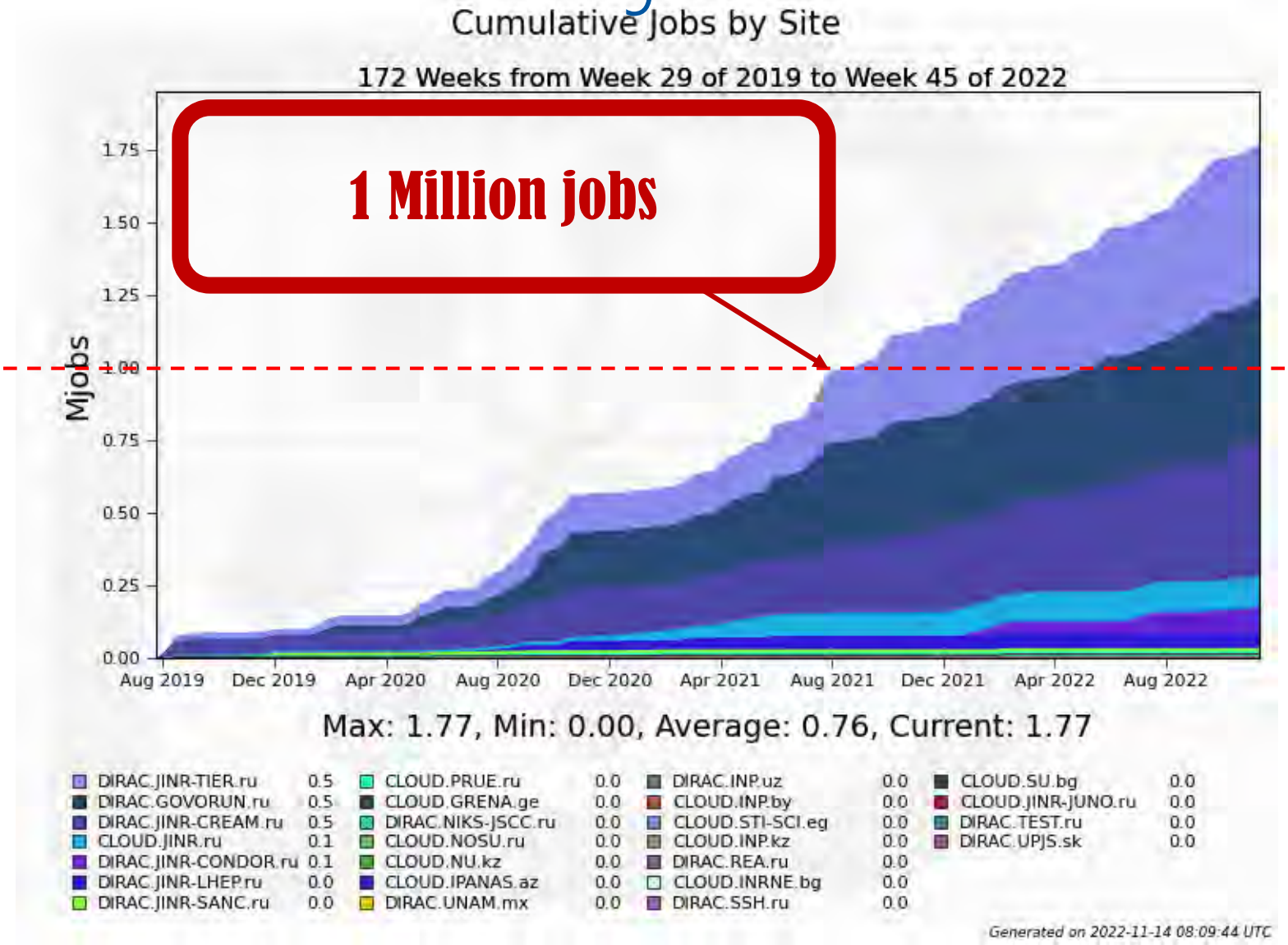
SPD@NICA

Monte-Carlo – Real

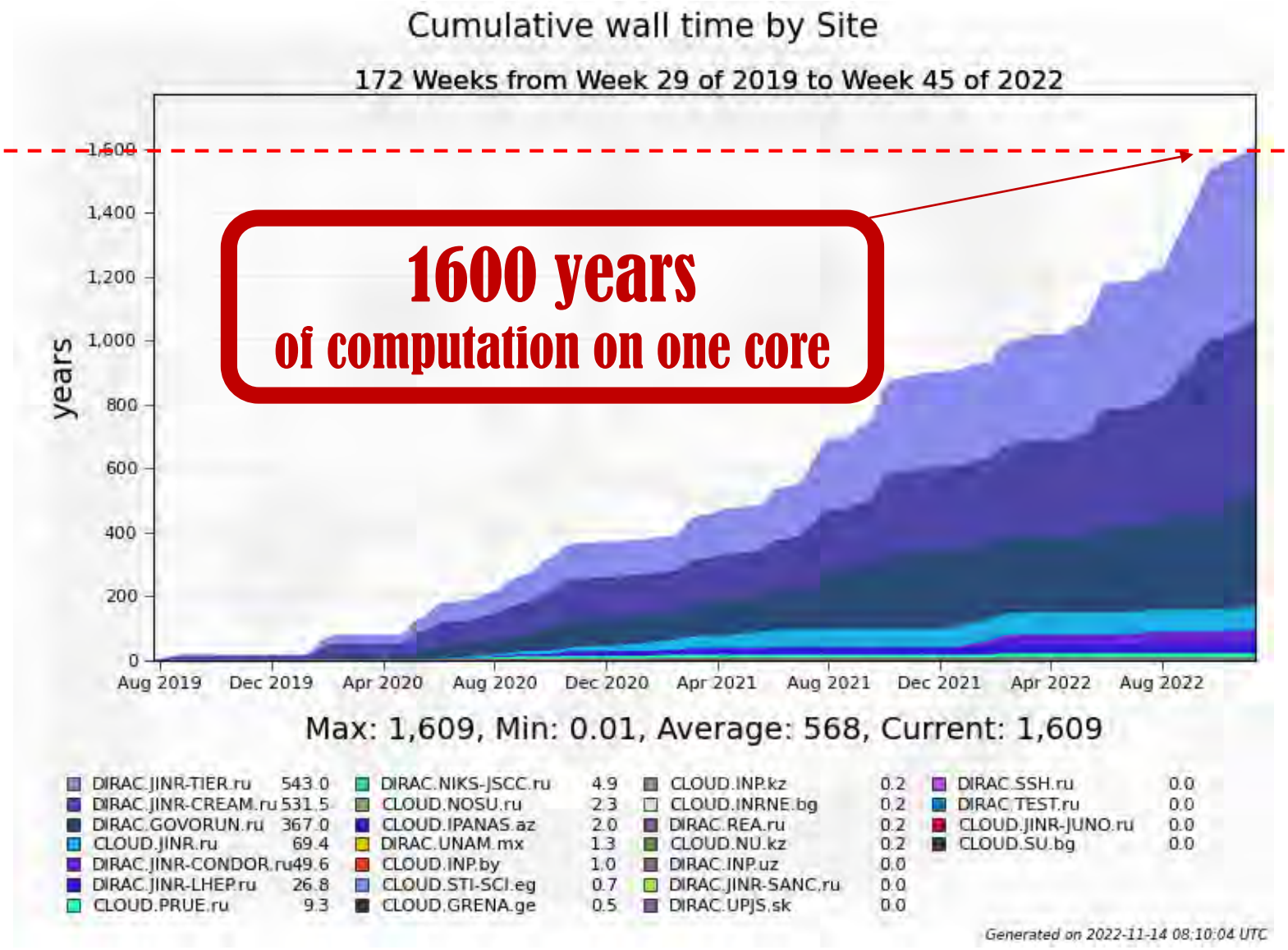
Folding@HOME

Teaching

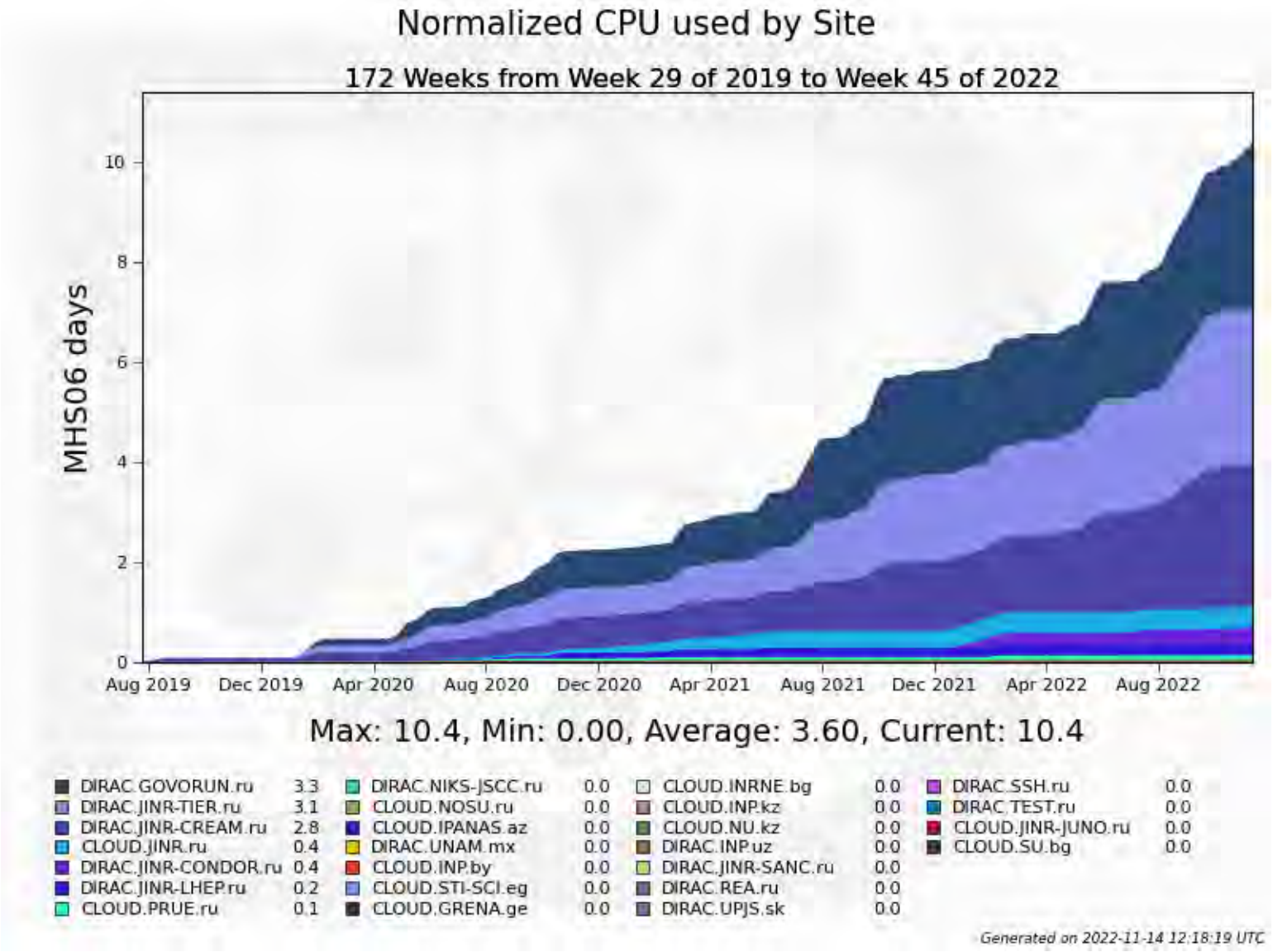
Statistics: jobs done



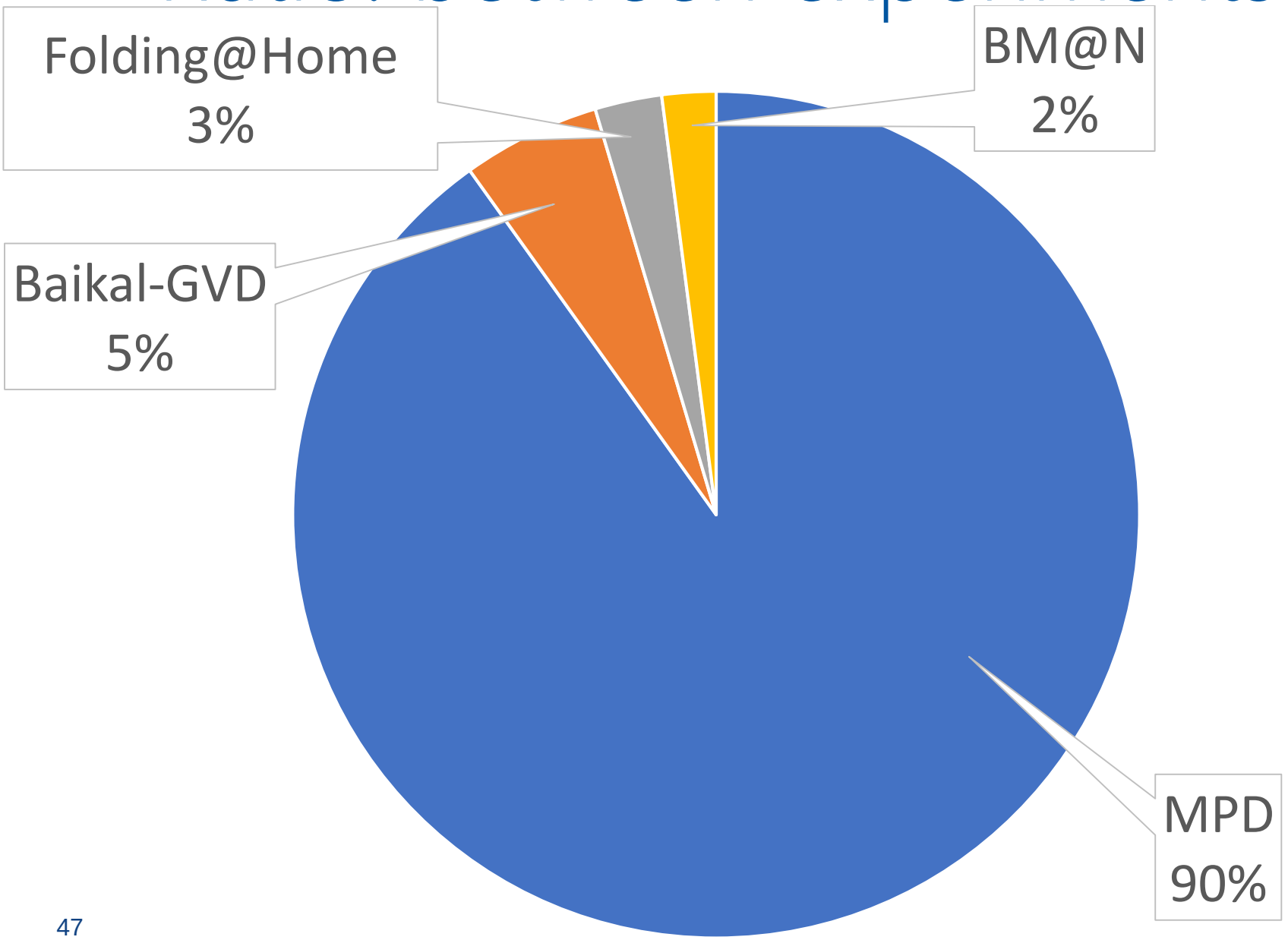
Statistics: walltime



Statistics: normalized time



Ratio: between experiments



Individual CPU core performance study

- Centralized job management gives possibility for centralized and unified performance study of different computing resources.
- Before running user jobs DIRAC Pilots execute benchmark for CPU core they are running on.
- Benchmark is DiracBenchmark2012 or DB12. It evaluate just CPU core performance. Disk I/O, RAM speed, Network, CPU caches and other highly important aspects of performance are **neglected by DB12.**

DB12 benchmark study

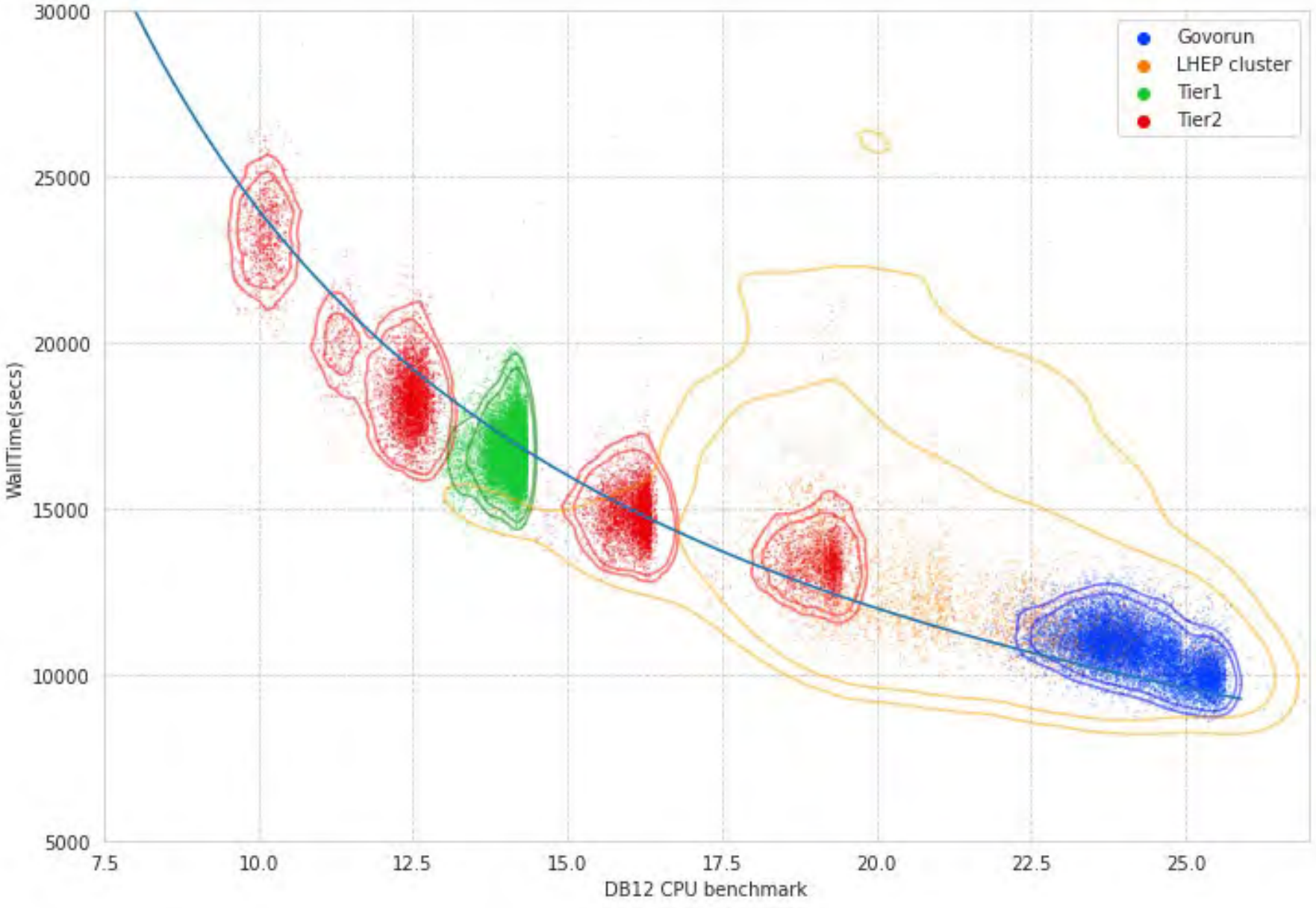
Piece of road from point A to B	←	A Monte-Carlo task
Speed of the car	←	Performance of the computer
Time to complete	←	Time to complete

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

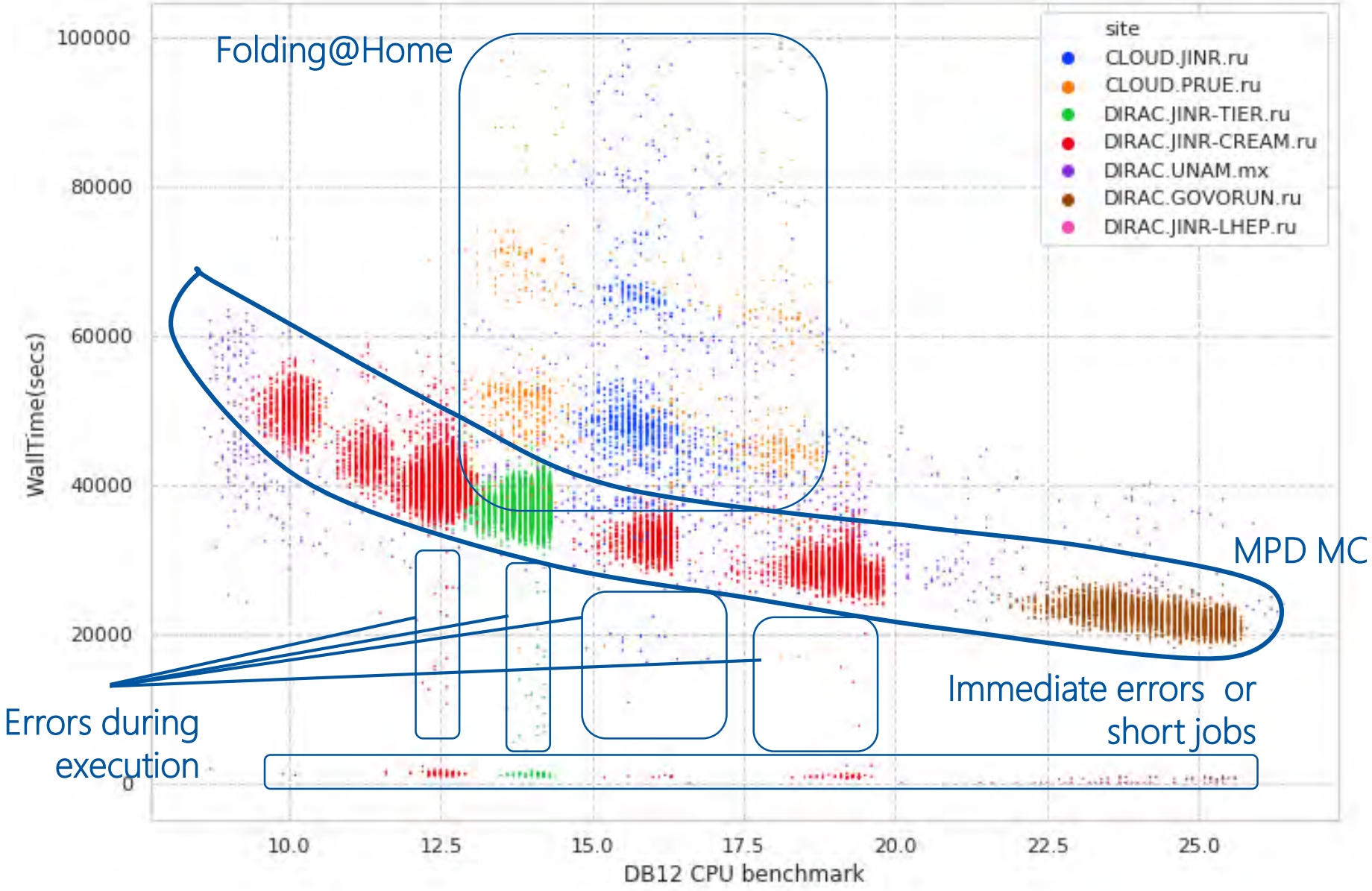
DB12 gives results like: 10(old slow core), 17 (standard server core), 27 (high performance core)

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 would be mostly useless if all jobs were unique and different.
But, in the real life there are usually many similar jobs.

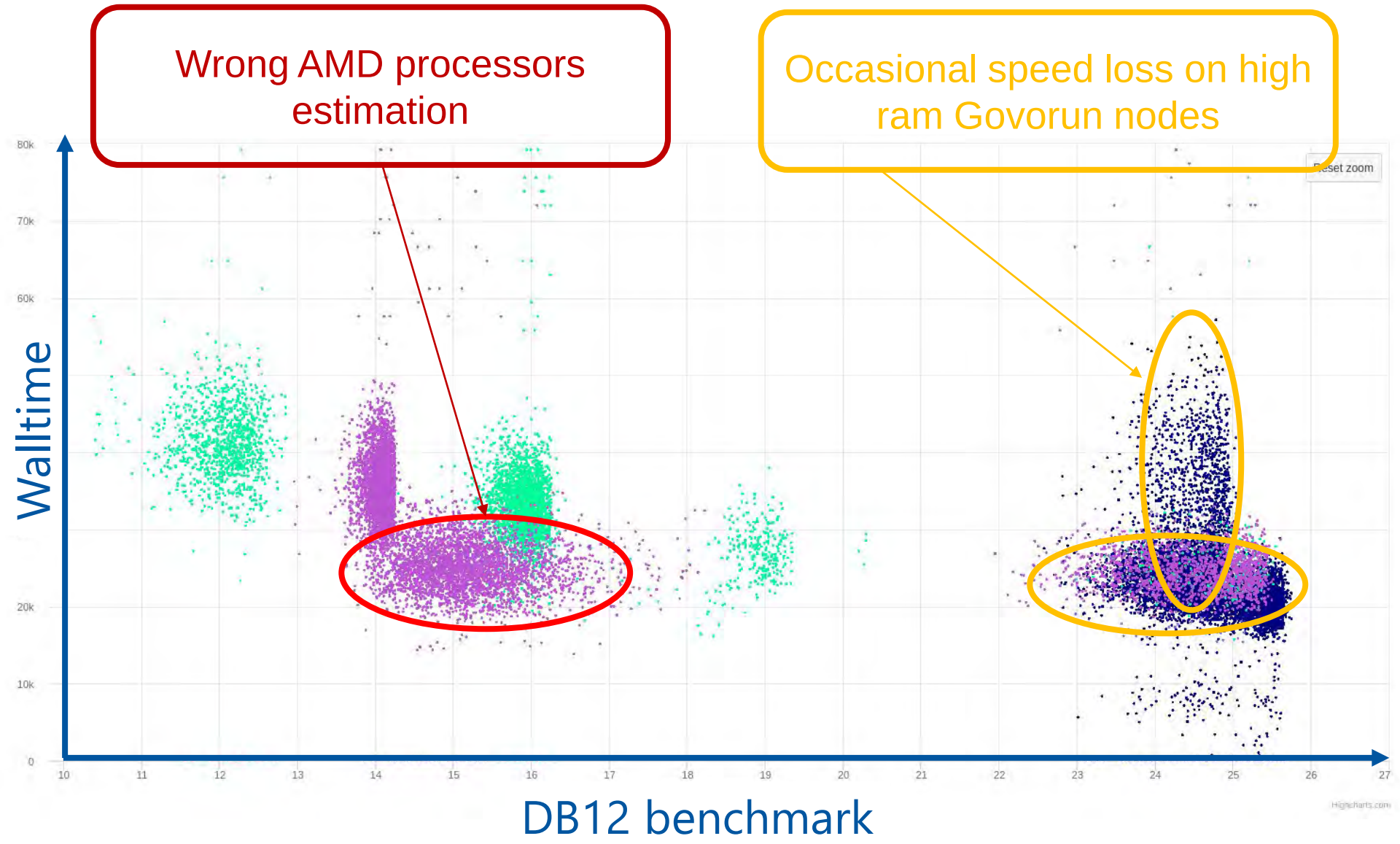
Performance analysis



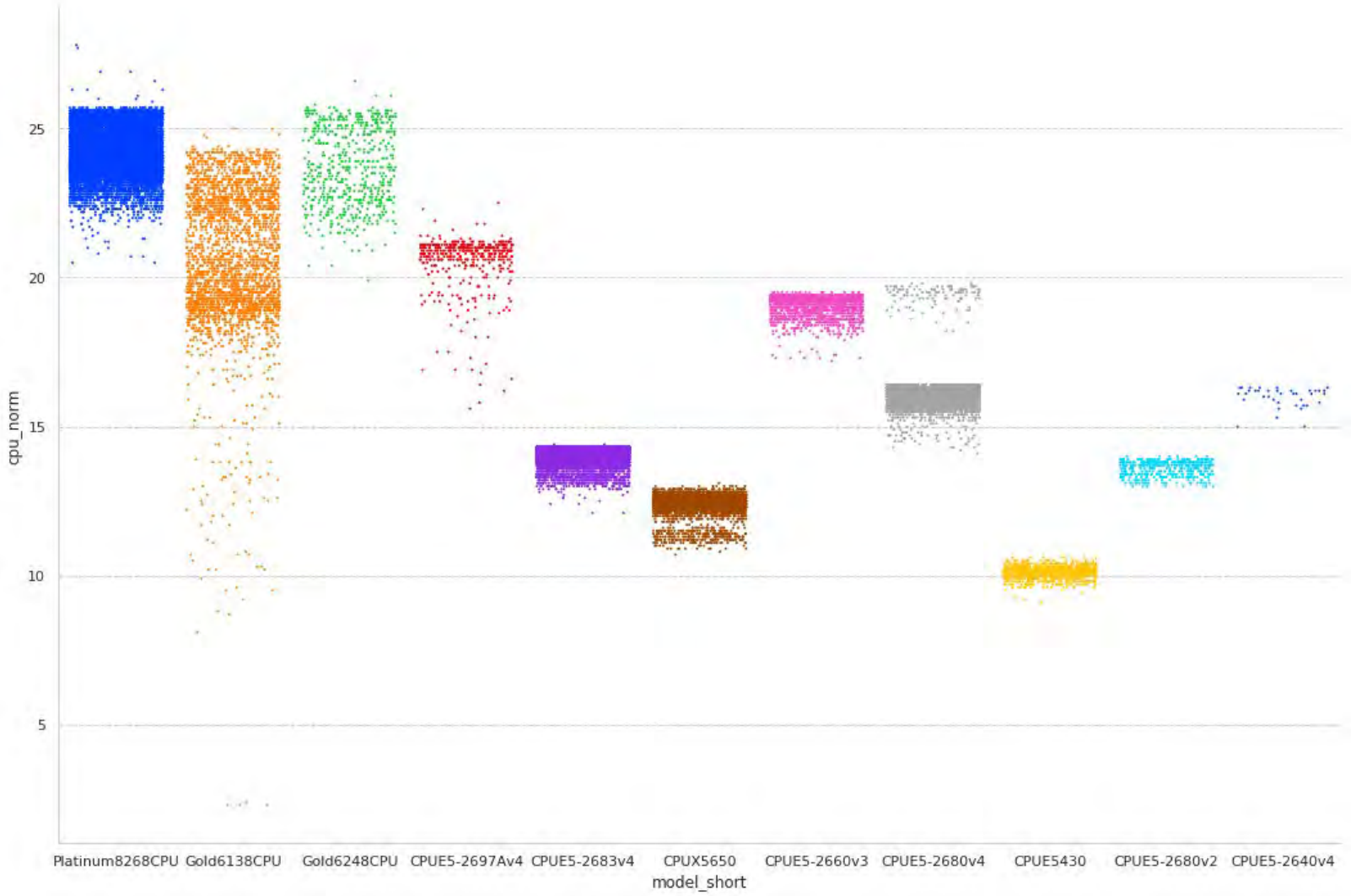
Performance analysis



Discoveries



CPU core performance



Total CPU performance

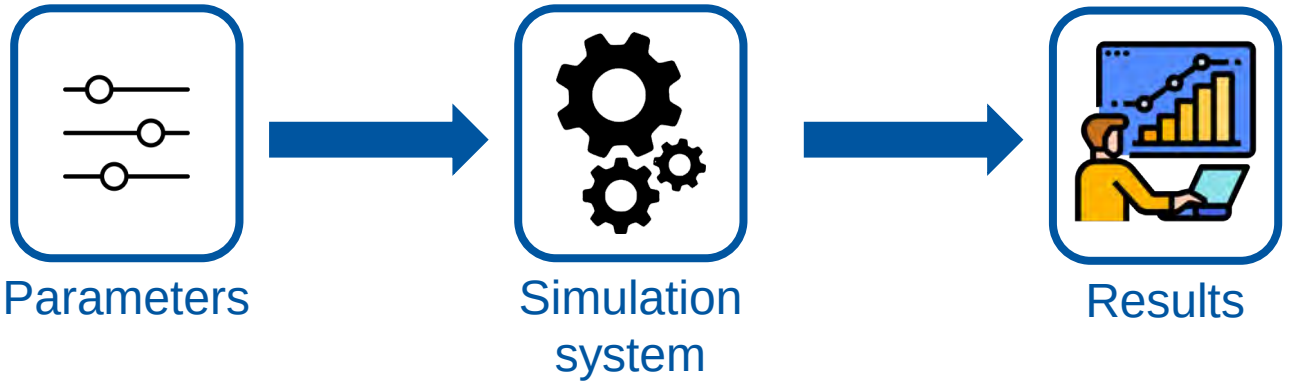
Questions that we ask

- Are we efficient in our job execution?
  - We have no access to monitoring information on a remote computing resources
  - Even if we have monitoring, it may be spoiled by other jobs
- Can we utilize more computing resources if we would have them in our disposal?
  - It is not so difficult to submit jobs on another thousand of cores, but will network handle them efficiently?

Parameters example

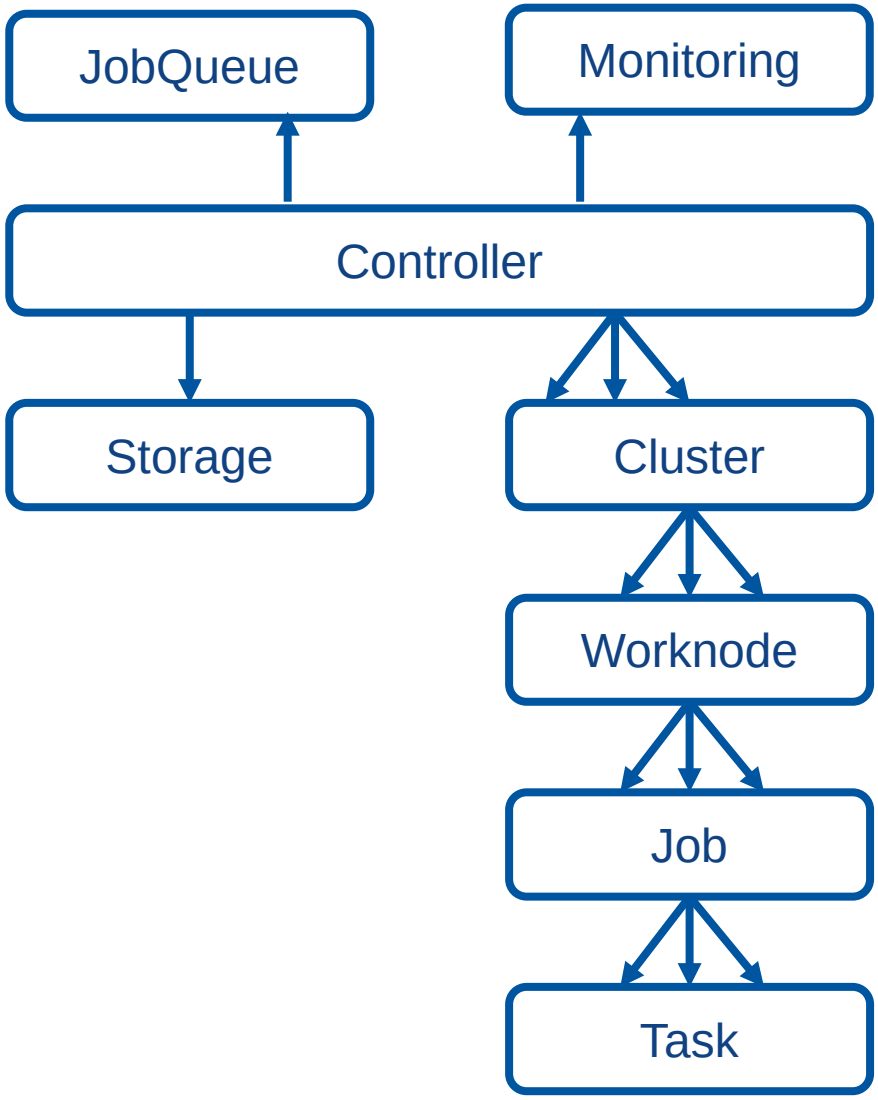
- If we have **20** computing worknodes, **40** cores available on each worknode.
- If **100 MB/s** maximum disk writing speed on each worknode.
- If new **40 GB** RAW file appears every **90** seconds. 105000 events in each RAW file.
- If each event processing time is 0.5 sec – one file processing will last for **14.5** hours.

Simulation is a solution

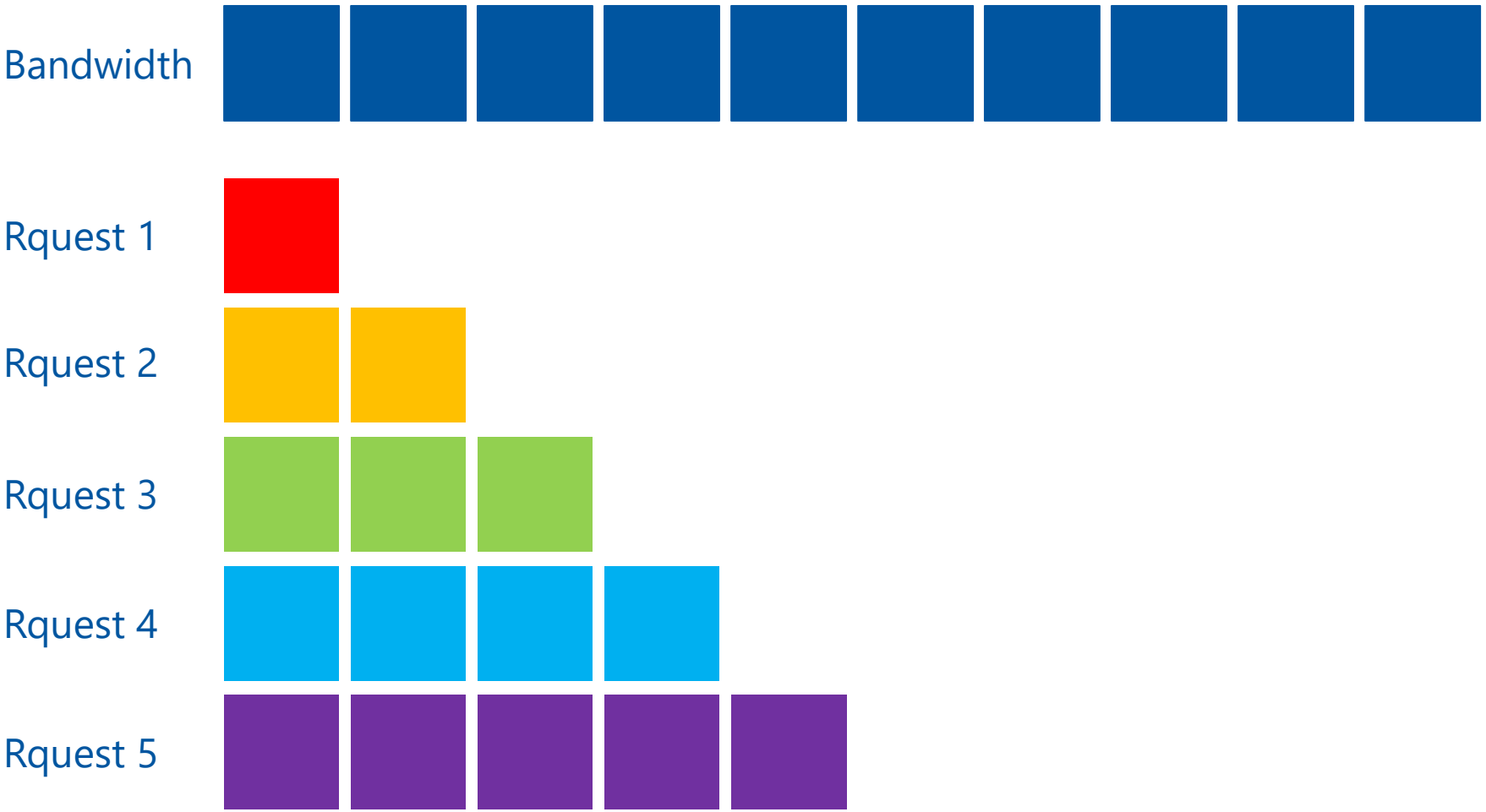


Simulation system

- Written in python to predict CPU, RAM, network and disk load
- Uses data about performance of resources integrated in DIRAC
- It is used to check the behavior of DIRAC jobs in real infrastructure.
- Simulation is done every second, but period may be increased for speeding up simulation.
- InfluxDB is used for results storage and visualization



Most important part
transfer simulation



Simple approach

Bandwidth

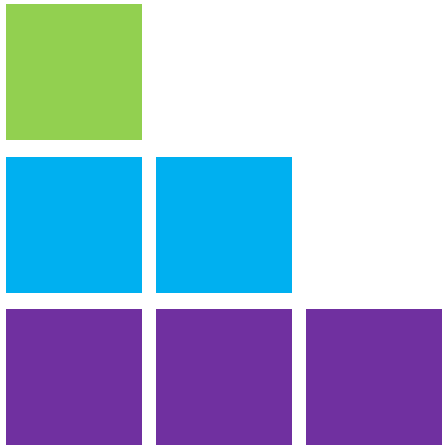
Rquest 1

Rquest 2

Rquest 3

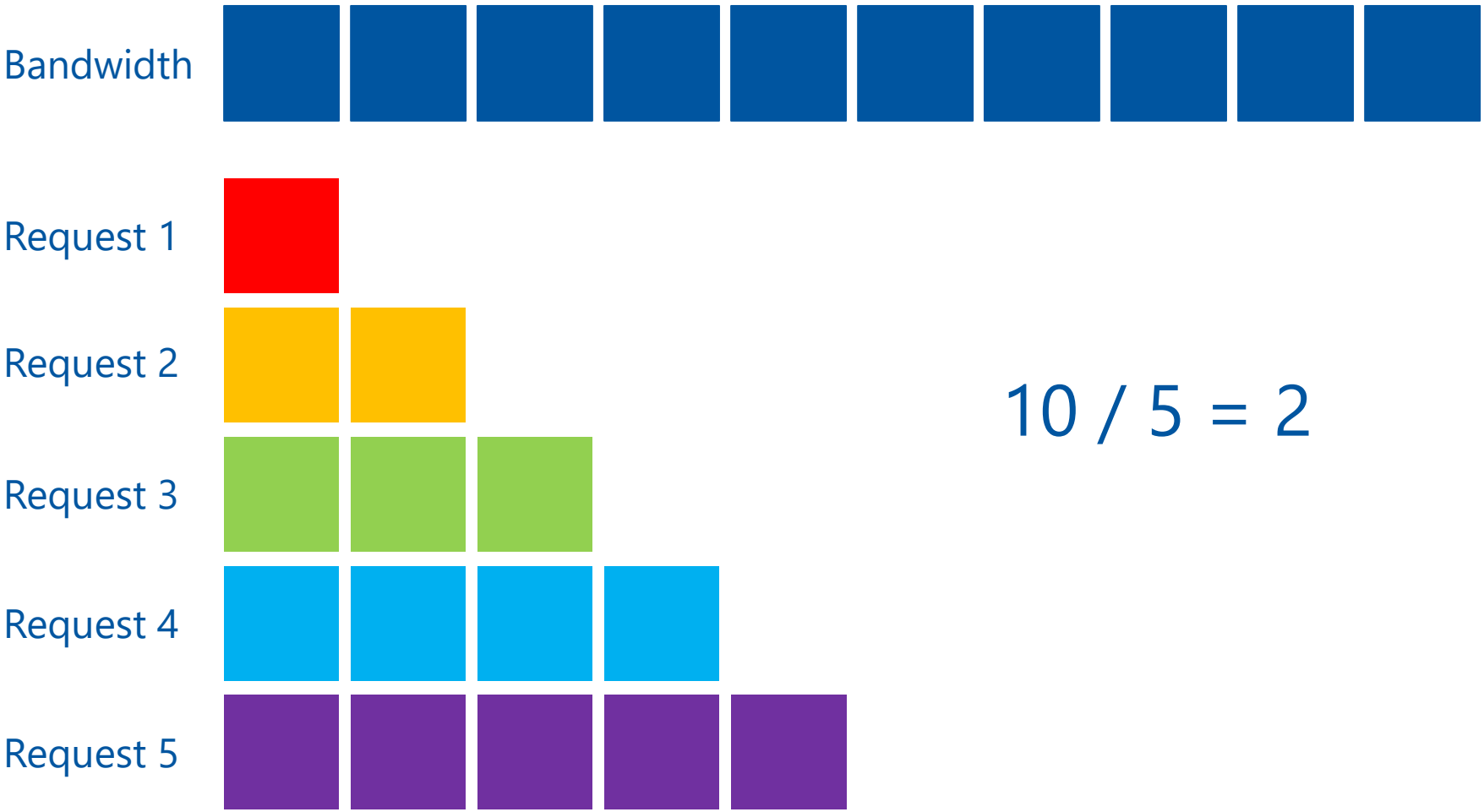
Rquest 4

Rquest 5

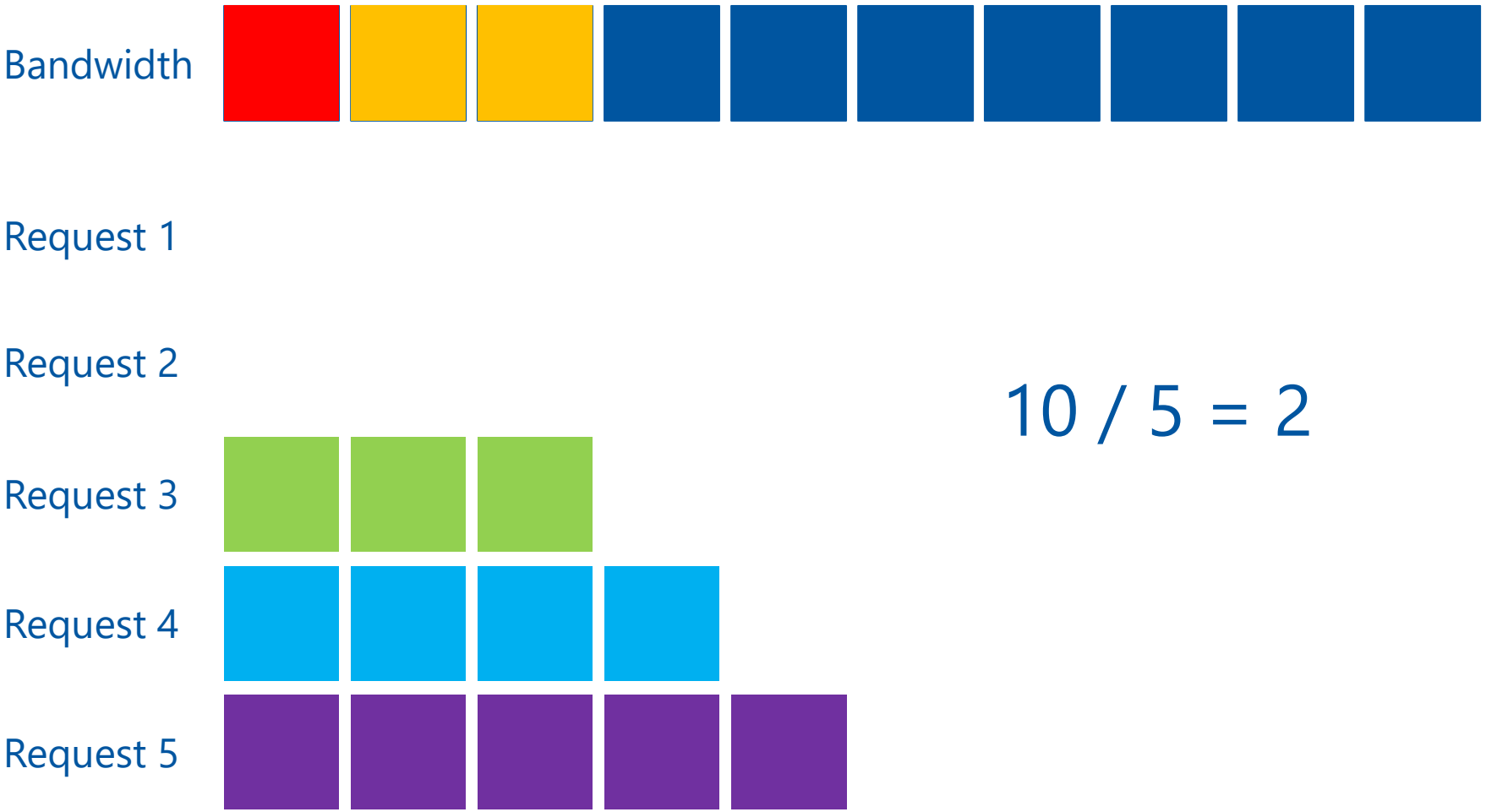


10 / 5 = 2

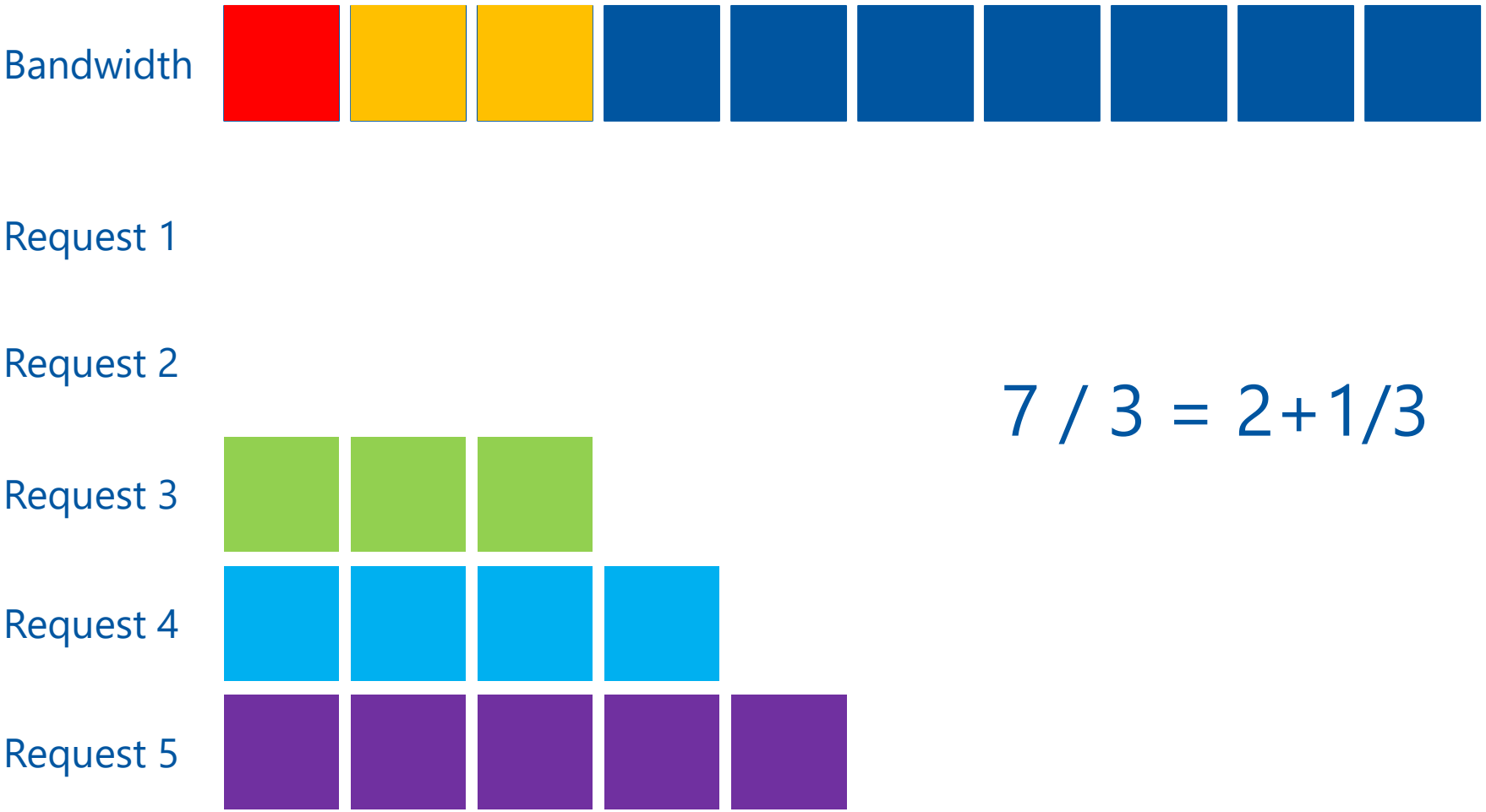
Used approach



Try to apply share



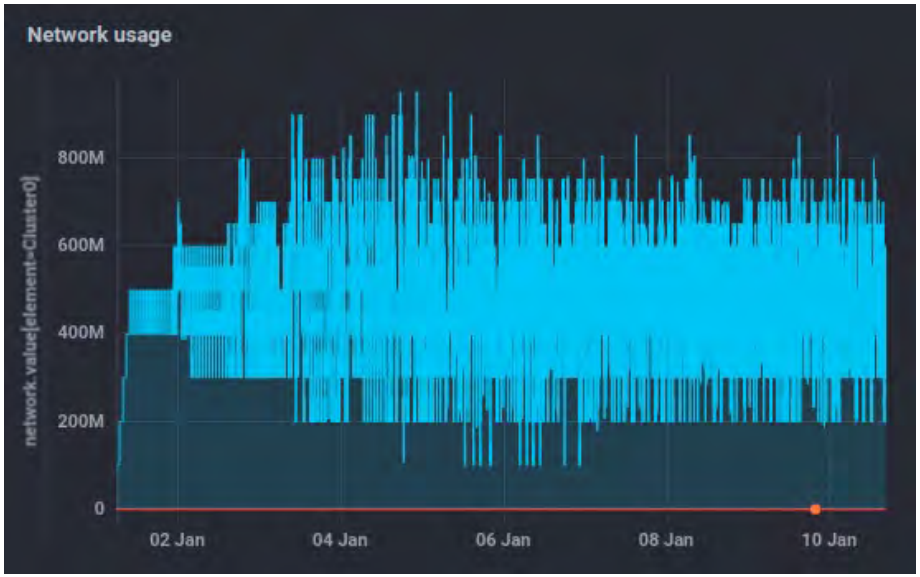
Recalculate share

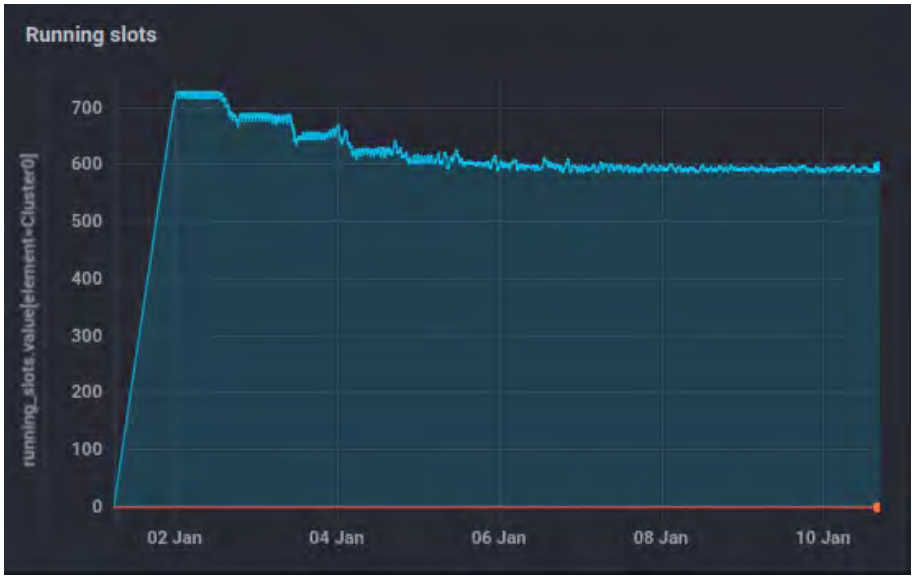


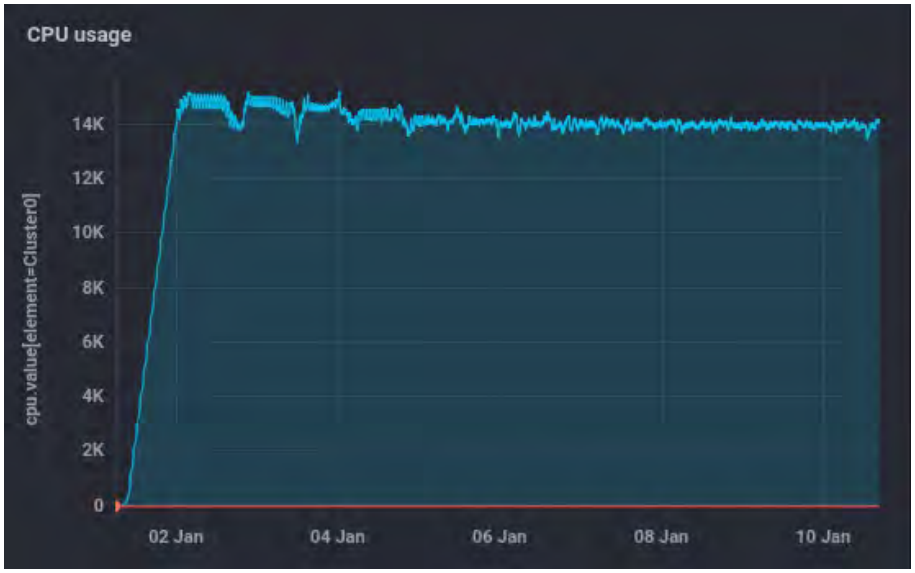
Apply share


Bandwidth
Request 1
Request 2
Request 3
Request 4
Request 5
7 / 3 = 2+1/3
These are left for the next simulation period

Simulation results



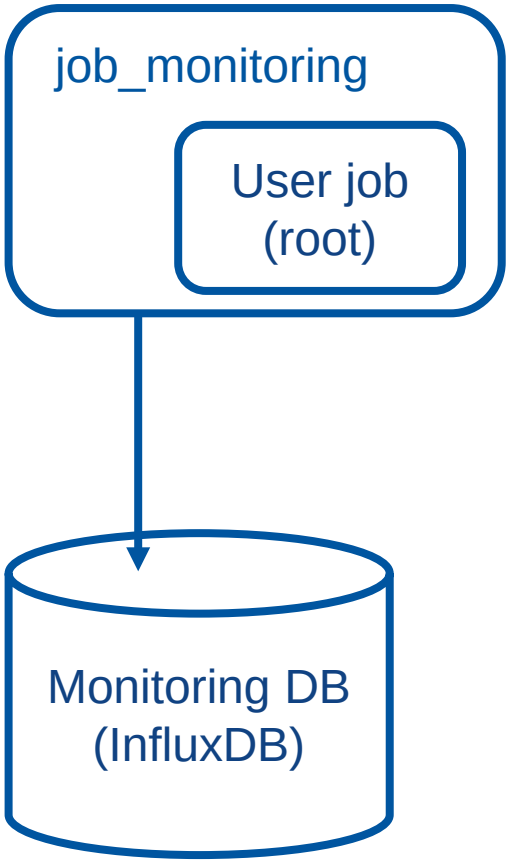




User job monitoring

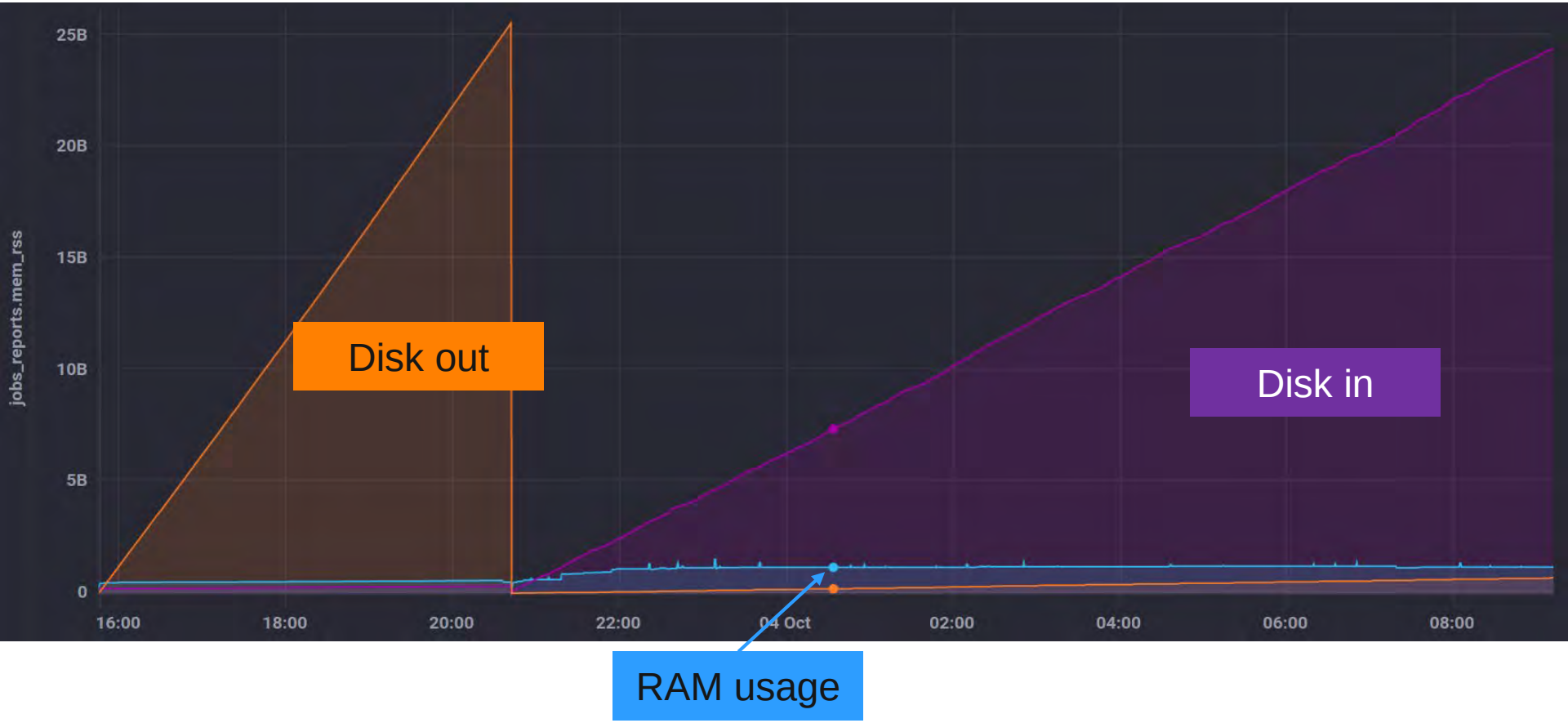
```
$ root macro.c(input)
```


```
$ job_monitoring root macro.c(input)
```

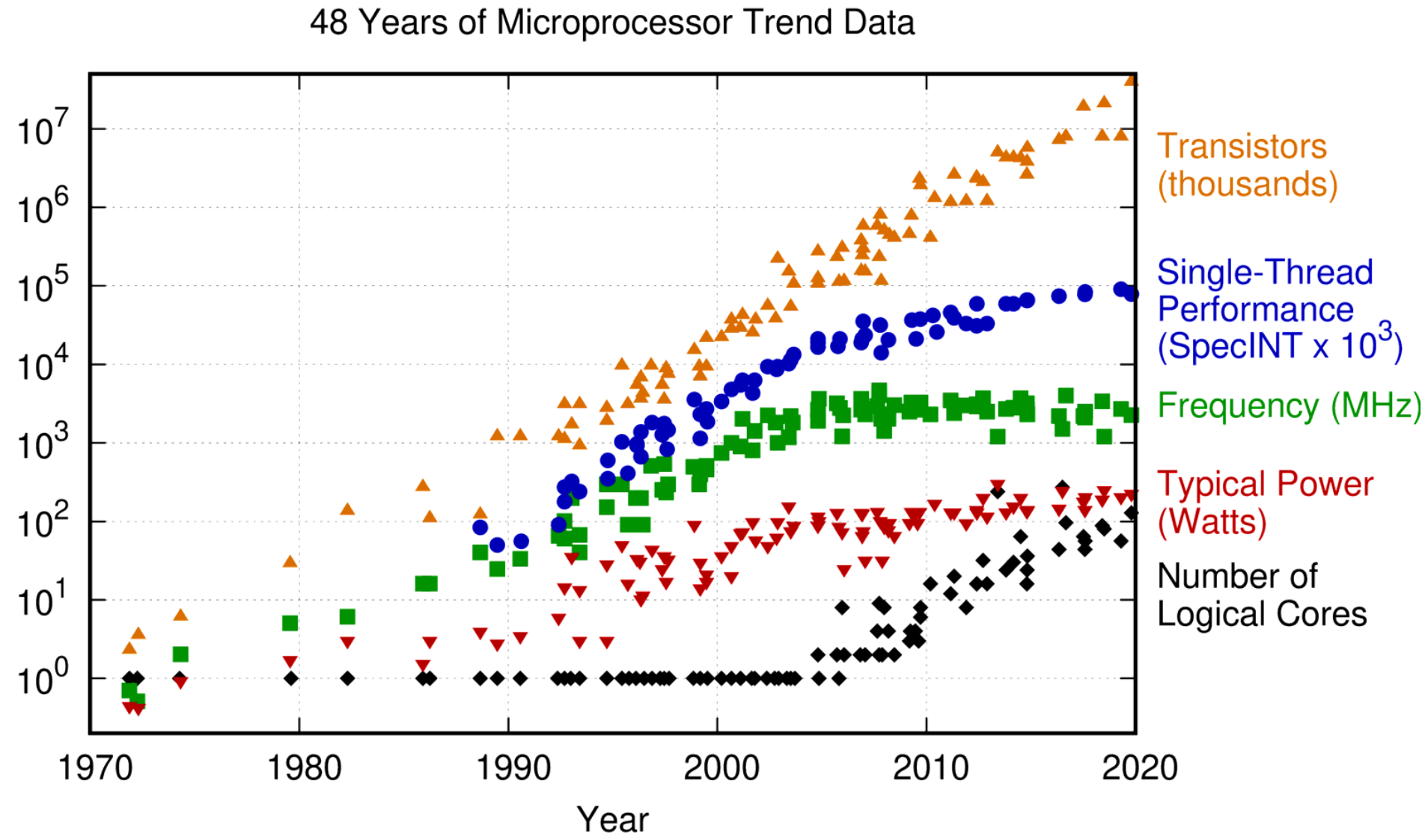


User job monitoring

GenToDst job on Govorun



Why do we do all that?



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2019 by K. Rupp

What to do with that

Go Parallel

HPC, Supercomputers

Go Distributed

Grid,
clusters

Go Effective

New algorithms,
machine learning

List of participants

DIRAC: Igor Pelevanyk, Andrey Tsaregorodtzev
Baikal-GVD: Dmitry Zaborov
BM@N: Konstantin Gertsenberger, Dmitry Tsvetkov
MPD: Oleg Rogachevskiy, Andrey Moshkin
SPD: Alexey Zhemchugov, Katherin Shtejer
Responsible for resources:
Govorun: Dmitry Podgainy, Dmitry Belyakov, Maxim Zuev
LHEP cluster: Boris Schinov
Tier-1,Tier-2, EOS: Valery Mitsyn
Cloud: Nikolay Kutovskiy, Nikita Balashov
dCache: Vladimir Trofimov

Detailed articles

1. Gergel, V., V. Korenkov, I. Pelevanyuk, M. Sapunov, A. Tsaregorodtsev, and P. Zrelov. 2017. **Hybrid Distributed Computing Service Based on the DIRAC Interware**.
2. Korenkov, V., Pelevanyuk, I. & Tsaregorodtsev, A. 2019, "**Dirac system as a mediator between hybrid resources and data intensive domains**", CEUR Workshop Proceedings, pp. 73.
3. Balashov, N.A., Kuchumov, R.I., Kutovskiy, N.A., Pelevanyuk, I.S., Petrunin, V.N. & Tsaregorodtsev, A.Y. 2019, "**Cloud integration within the DIRAC Interware**", CEUR Workshop Proceedings, pp. 256.
4. Korenkov, V., Pelevanyuk, I. & Tsaregorodtsev, A. 2020, **Integration of the JINR hybrid computing resources with the DIRAC interware for data intensive applications**.
5. Kutovskiy, N., Mitsyn, V., Moshkin, A., Pelevanyuk, I., Podgayny, D., Rogachevsky, O., Shchinov, B., Trofimov, V. & Tsaregorodtsev, A. 2021, "**Integration of Distributed Heterogeneous Computing Resources for the MPD Experiment with DIRAC Interware**", Physics of Particles and Nuclei, vol. 52, no. 4, pp. 835-841.
6. Pelevanyuk, I., "**Performance evaluation of computing resources with DIRAC interware**", AIP Conference Proceedings 2377, 040006 (2021)


JINR
MLiT