Introduction to the Worldwide LHC Computing Grid

Lectures On Modern Scientific Programming

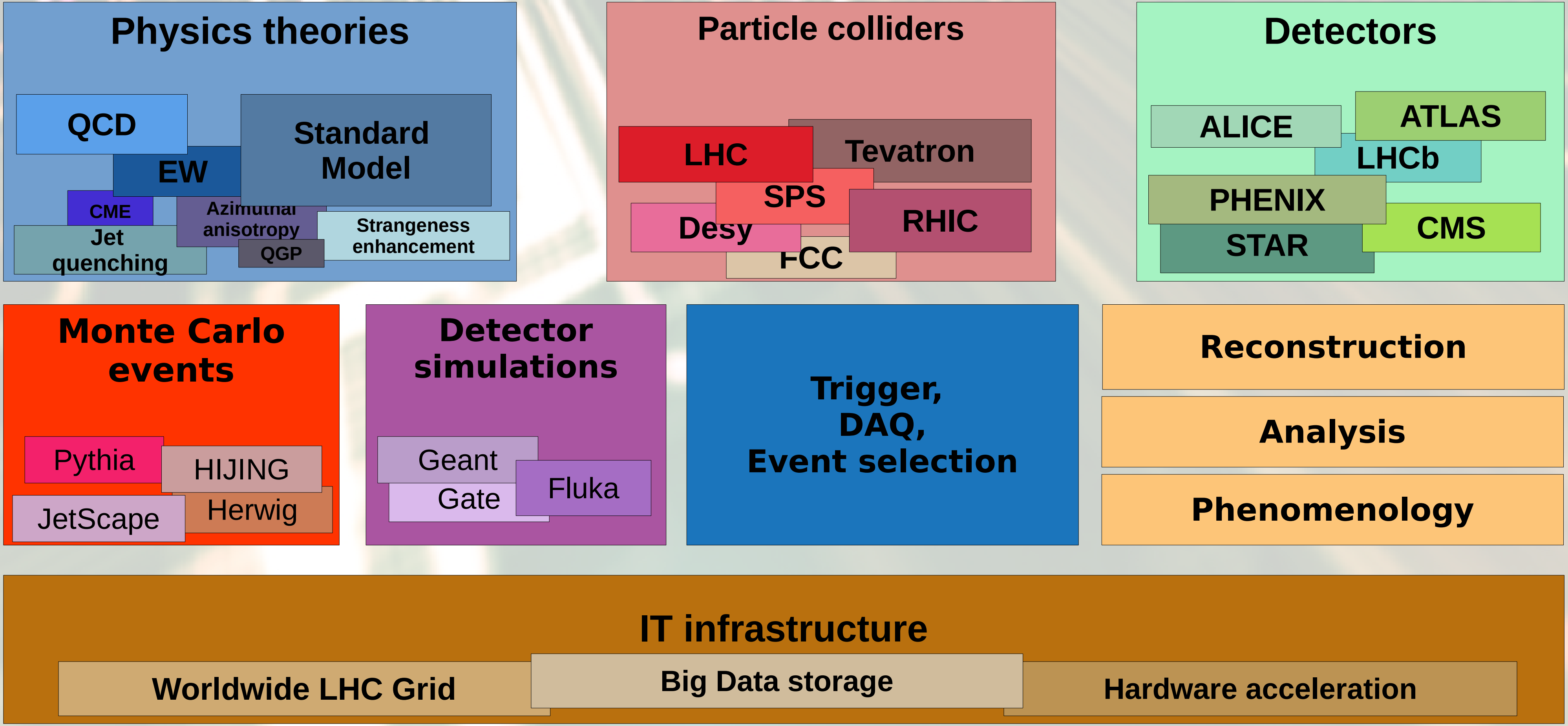
Wigner Scientific Computing Laboratory

12-13 11 2023

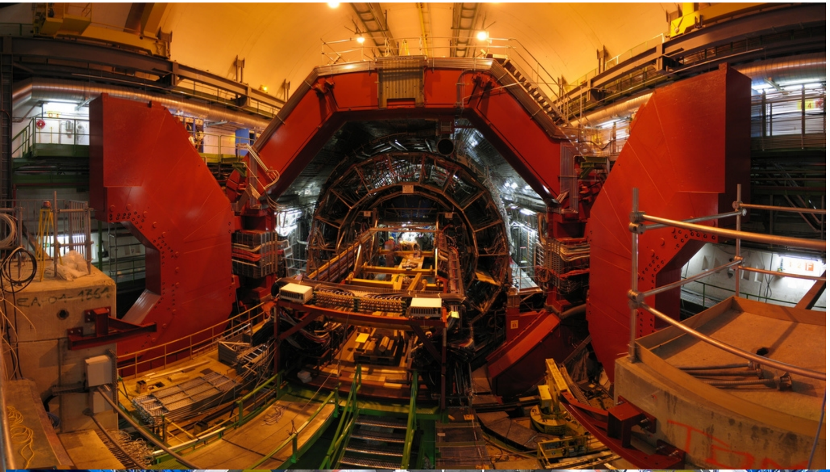
GÁBOR BÍRÓ
biro.gabor@wigner.hun-ren.hu

HUN-REN Magyar Kutatási Hálózat

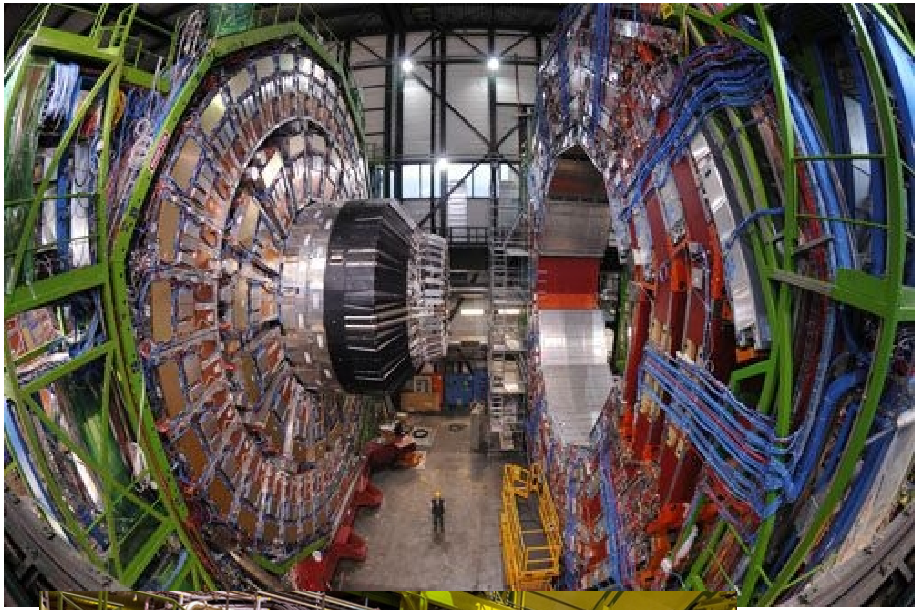
Ingredients of HEP

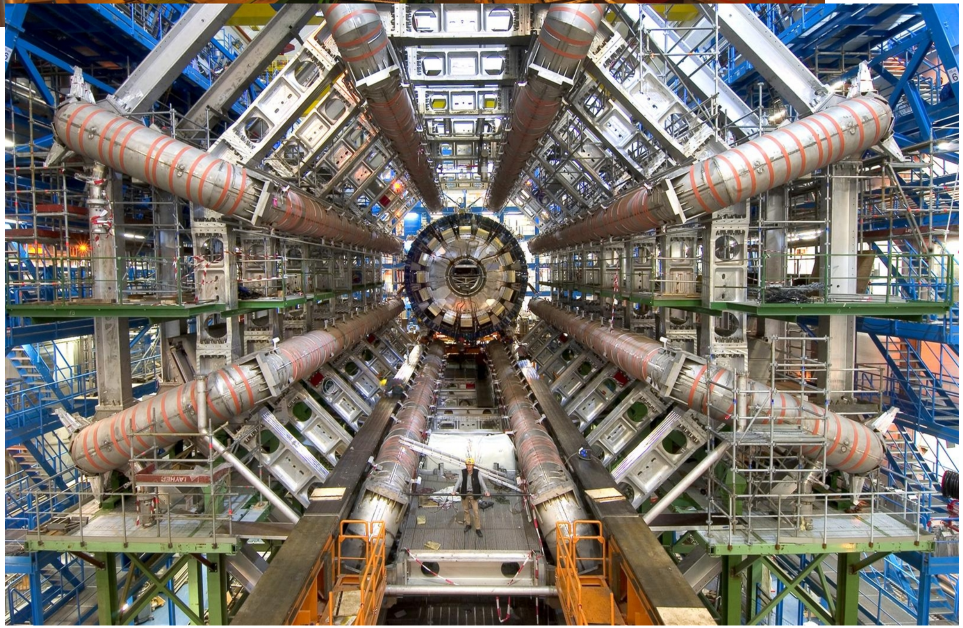



LHCb
ATLAS
CERN Meyrin
CERN Prévessin
SPS 7 km
SUISSE
FRANCE
ALICE
CMS
LHC
LHC 27 km

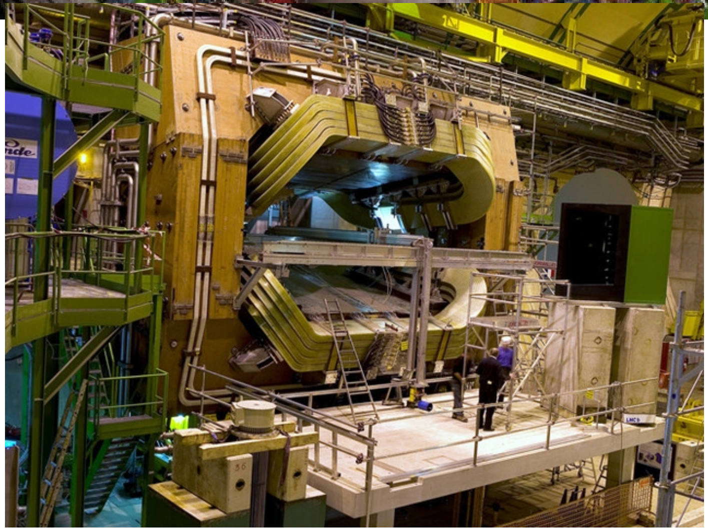


ALICE
CMS

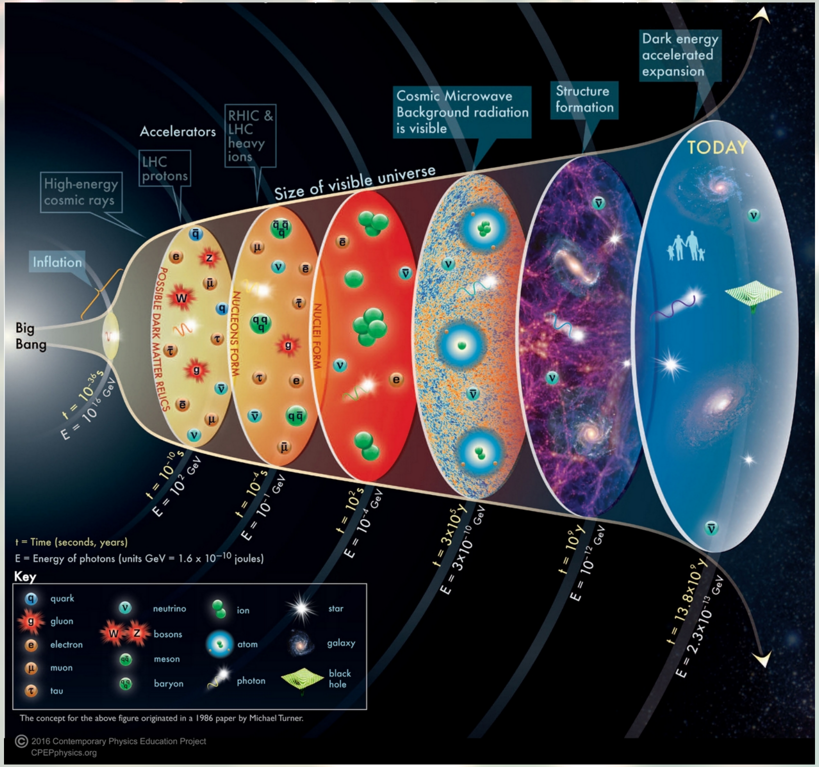




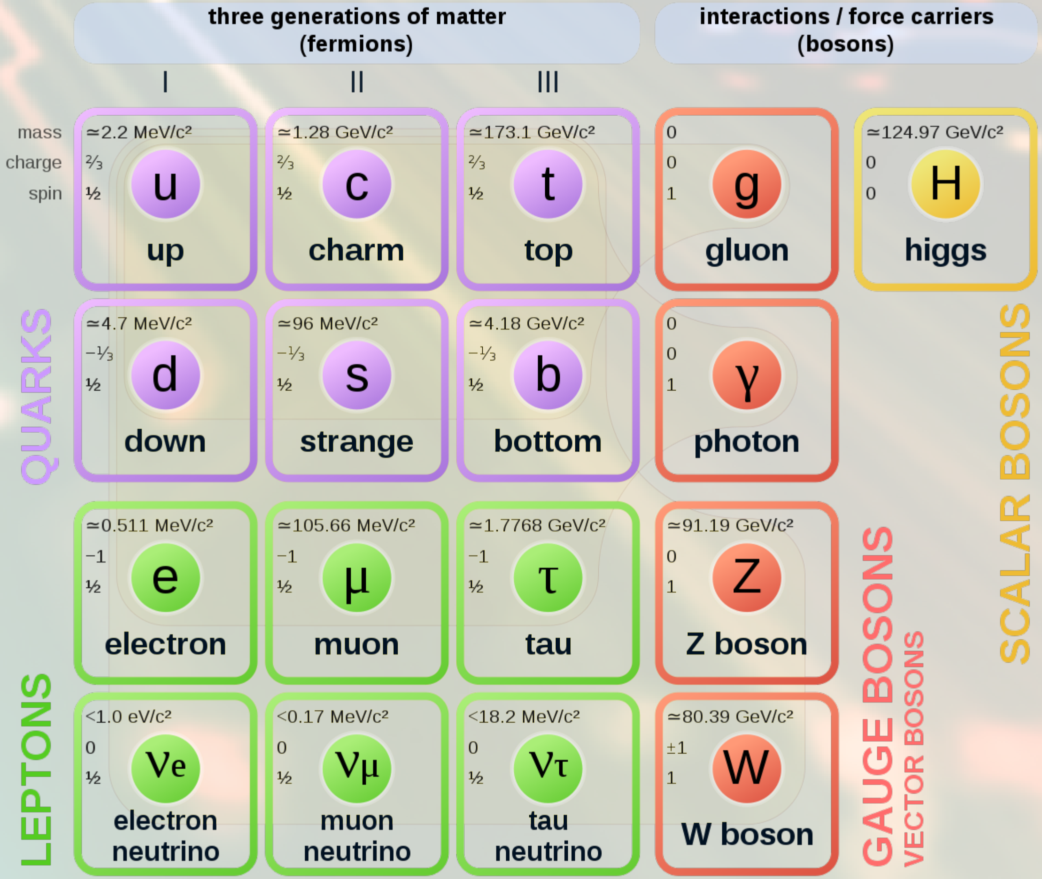
ATLAS
LHCb



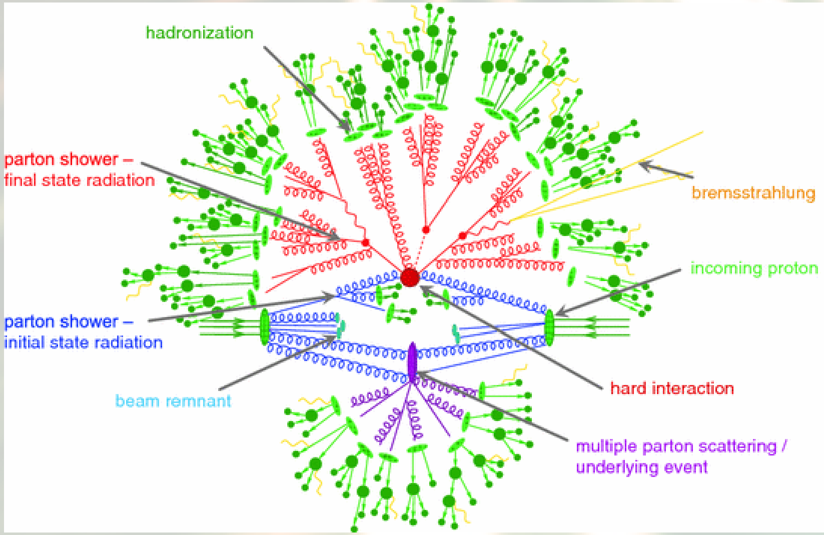
Ingredients of HEP

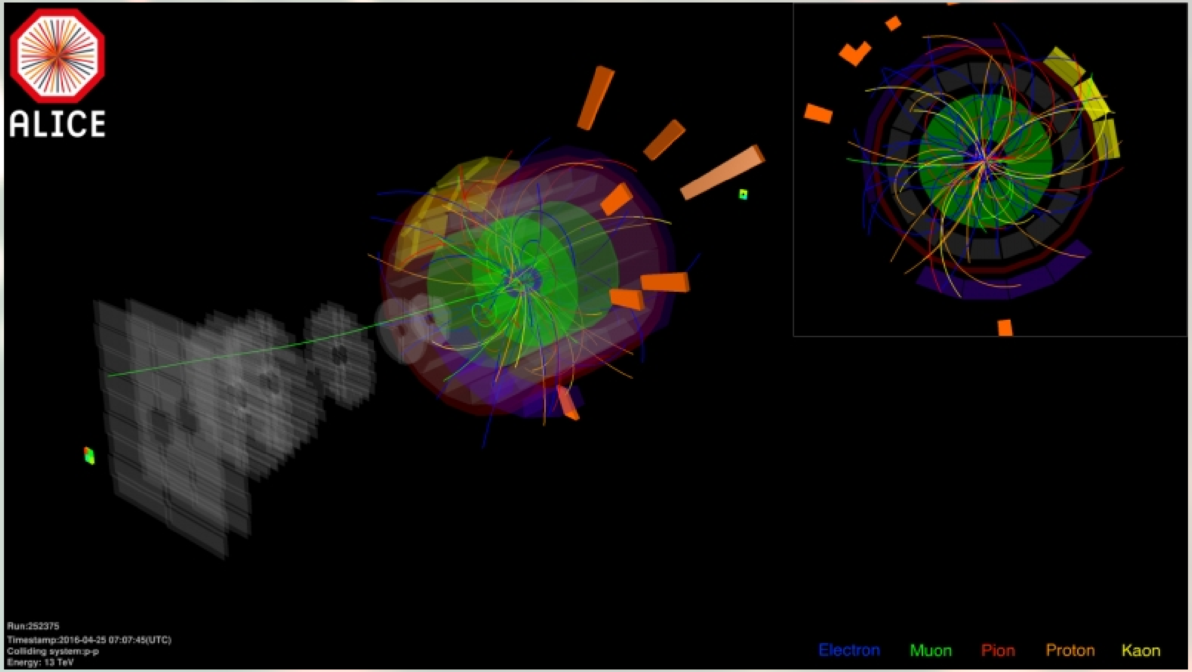


Standard Model of Elementary Particles

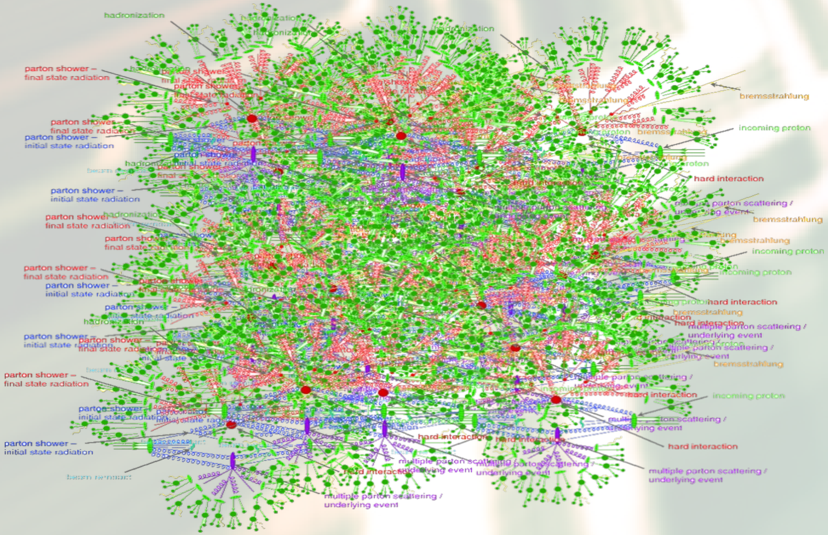


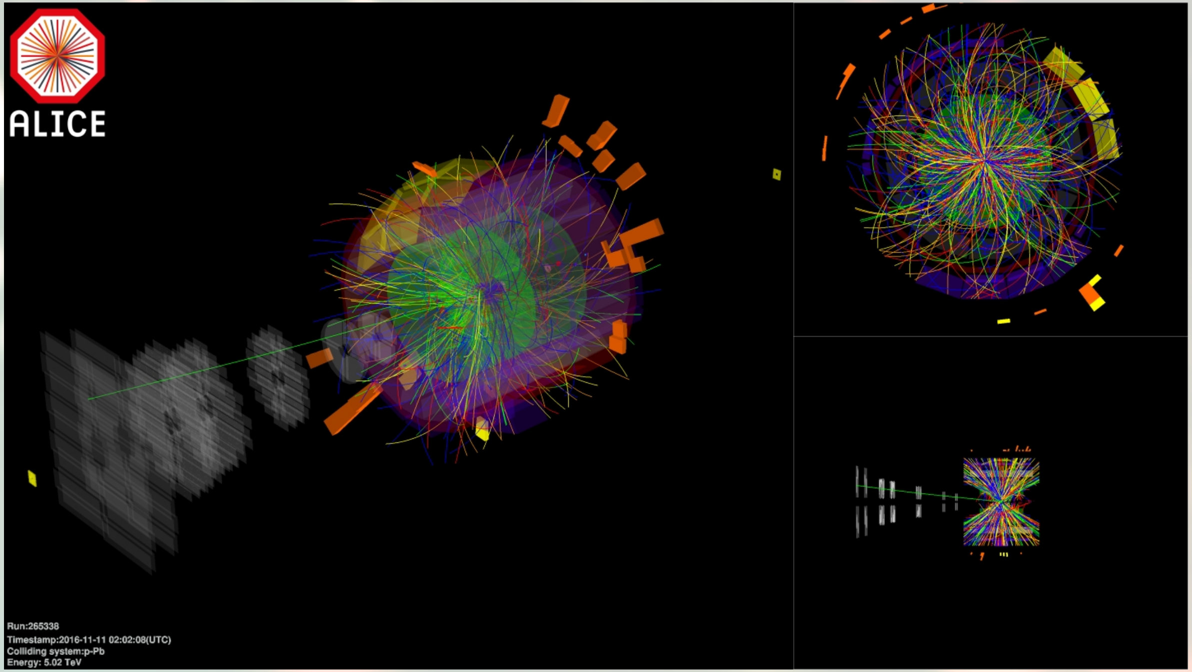
Ingredients of HEP



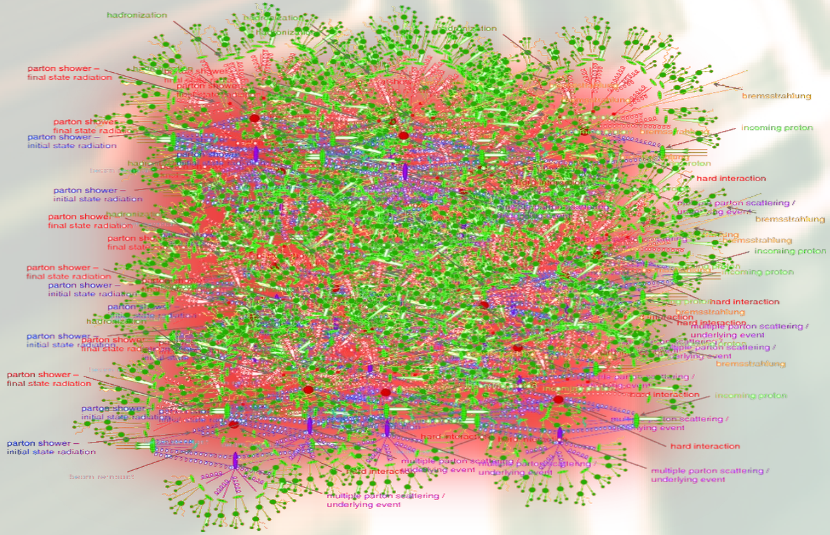


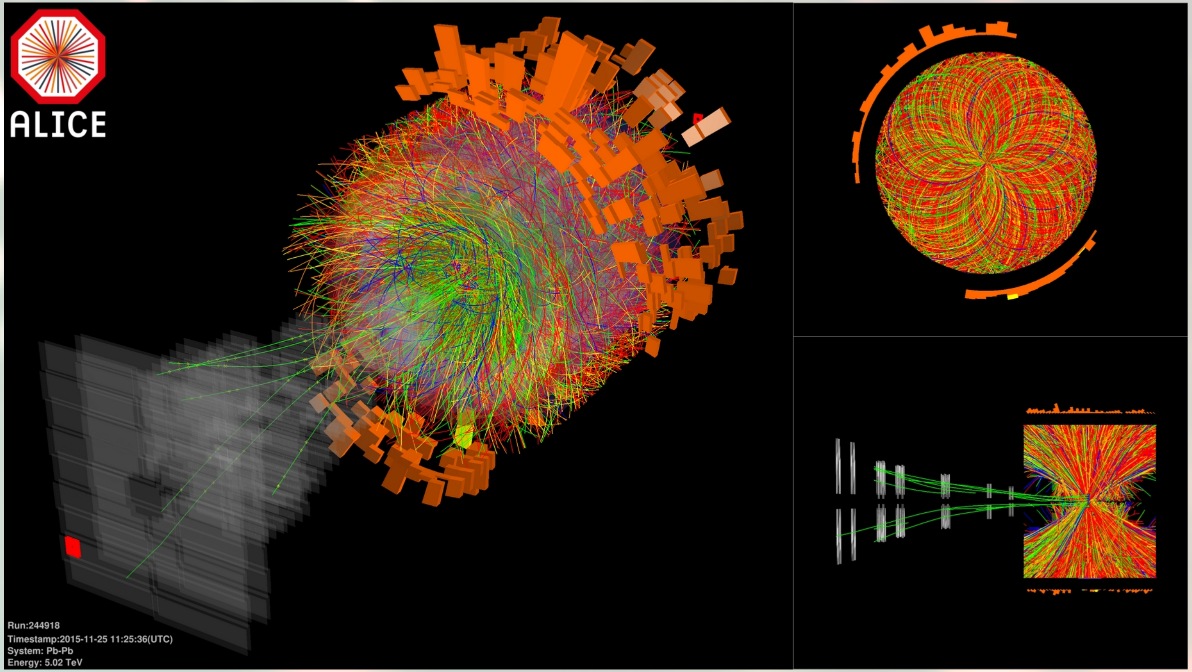
Ingredients of HEP



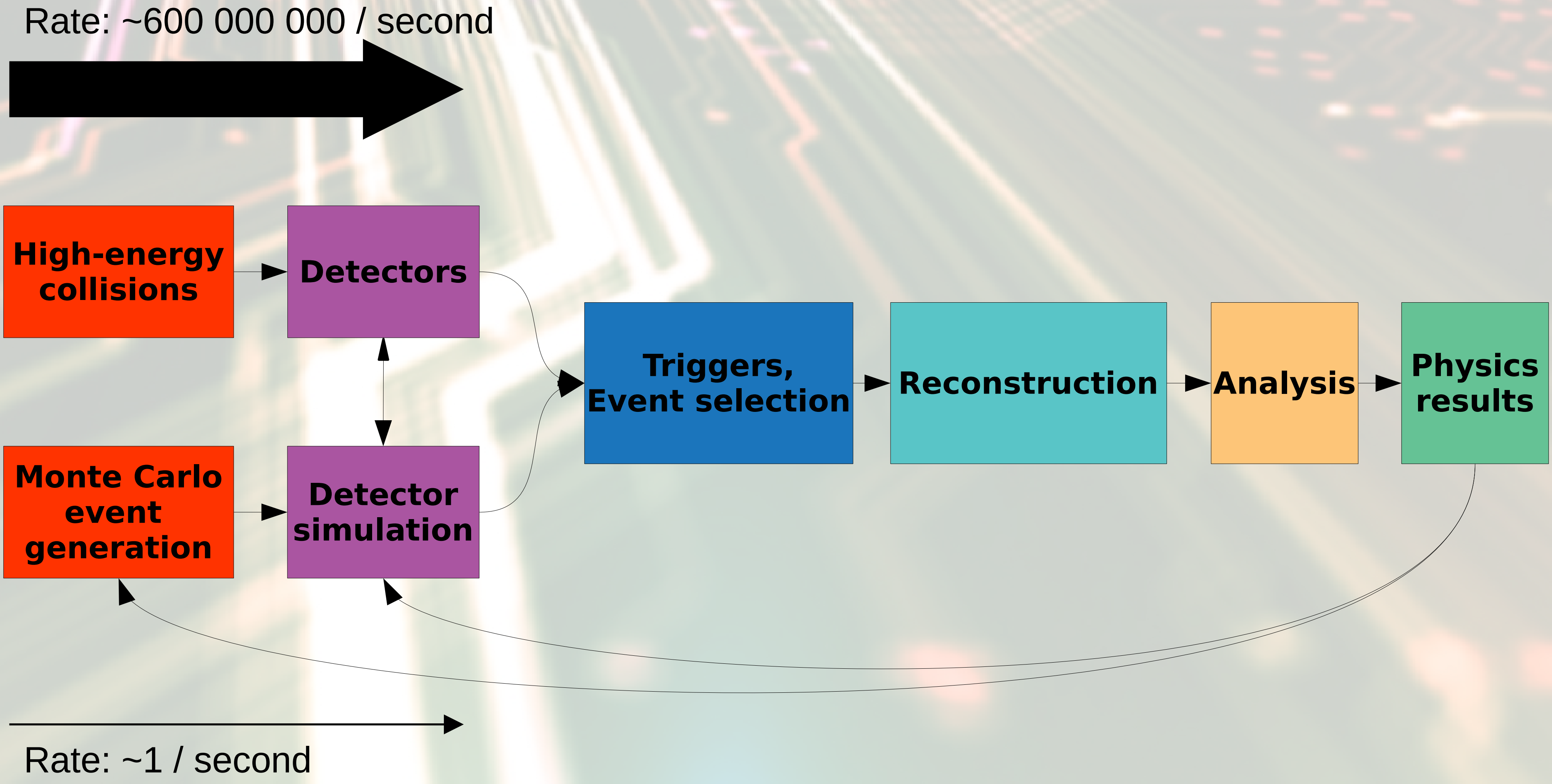


Ingredients of HEP

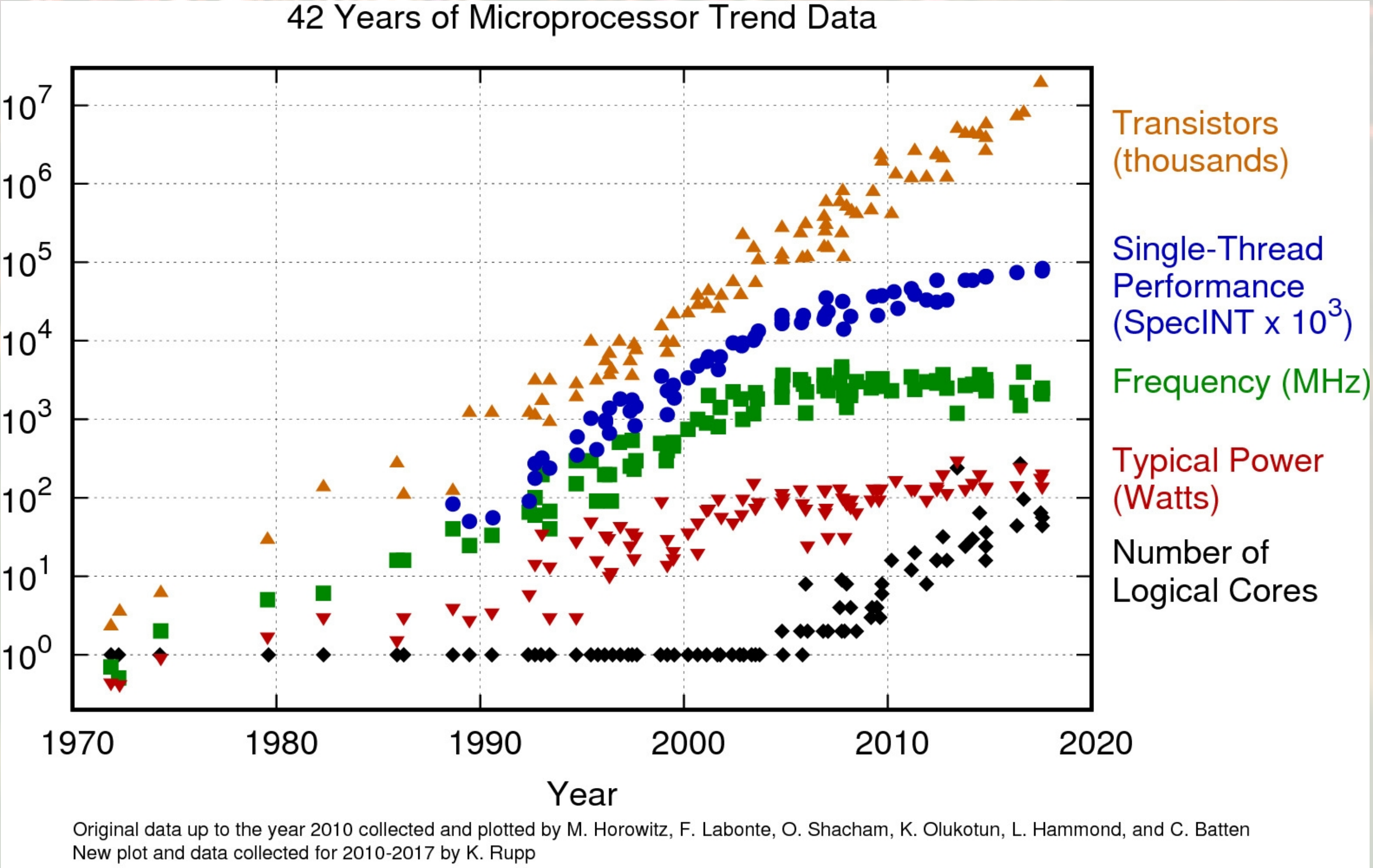




The PROBLEM

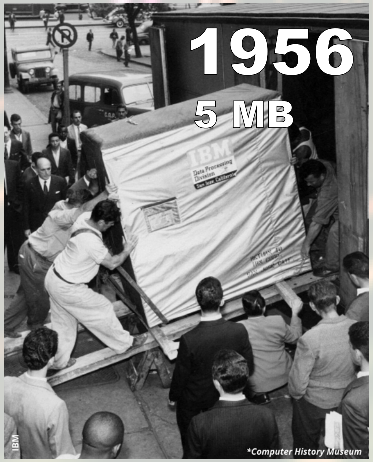

Rate: ~600 000 000 / second
High-energy collisions
Detectors
Monte Carlo event generation
Detector simulation
Triggers, Event selection
Reconstruction
Analysis
Physics results
Rate: ~1 / second

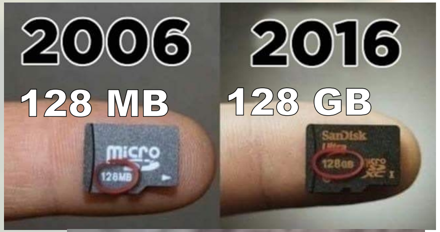
The PROBLEM



42 Years of Microprocessor Trend Data
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
1970
1980
1990
2000
2010
2020
Year
Transistors (thousands)
Single-Thread Performance (SpecINT x 10^3)
Frequency (MHz)
Typical Power (Watts)
Number of Logical Cores
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 2010-2017 by K. Rupp

The PROBLEM



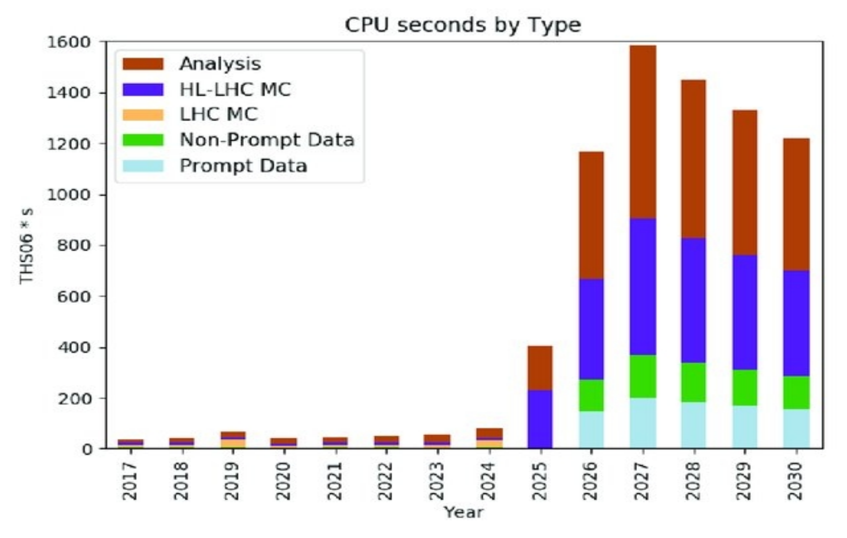


LHC in numbers: **2013** and **now:**

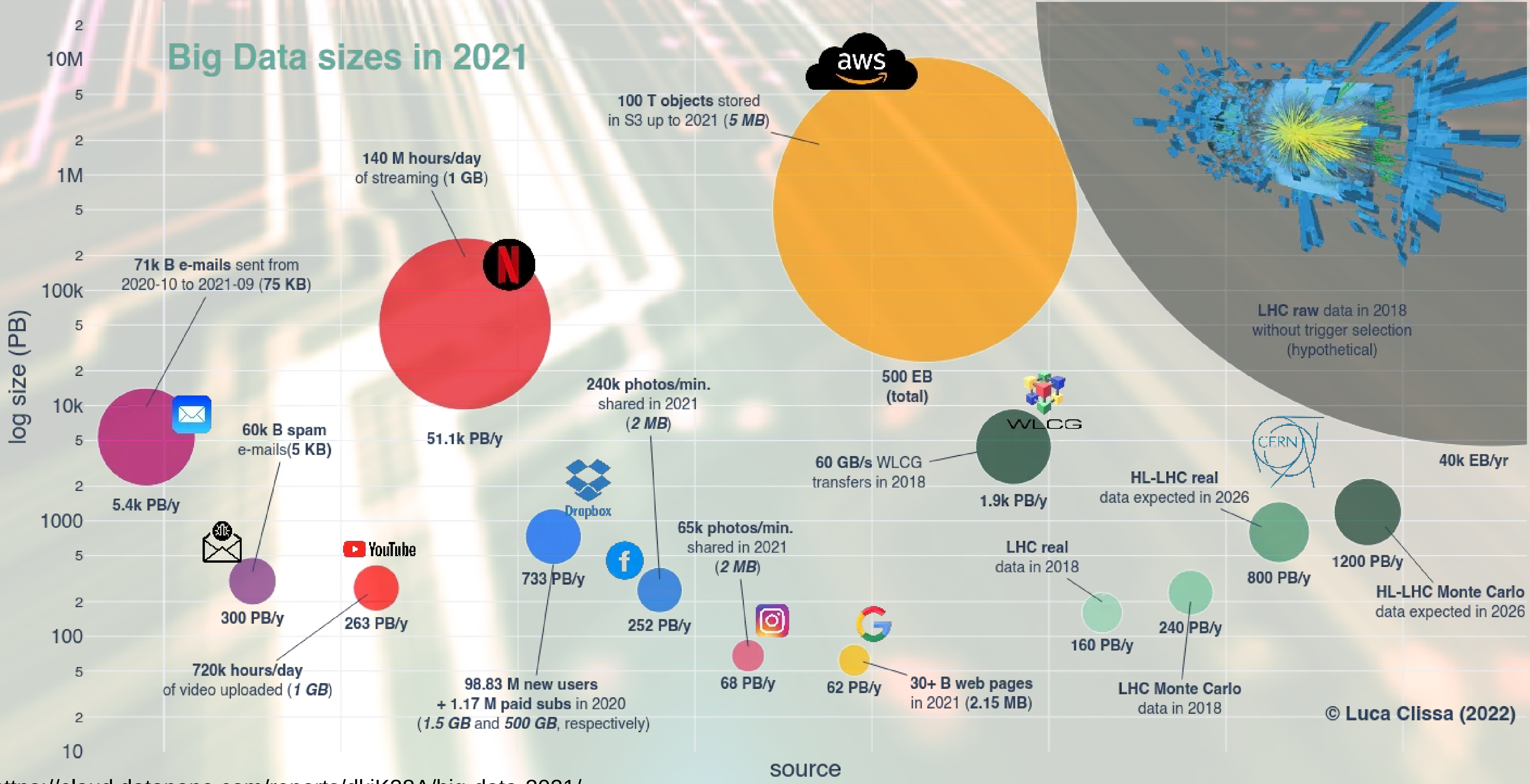
Data:	15 PB/year	vs 200+ PB/year
Tape:	180 PB	vs 740+ PB
Disk:	200 PB	vs 570+ PB
HS06:	2M	vs 100+ B

Storing and distributing the data is only one side of the challange

→ reconstruction, analysis, simulations...

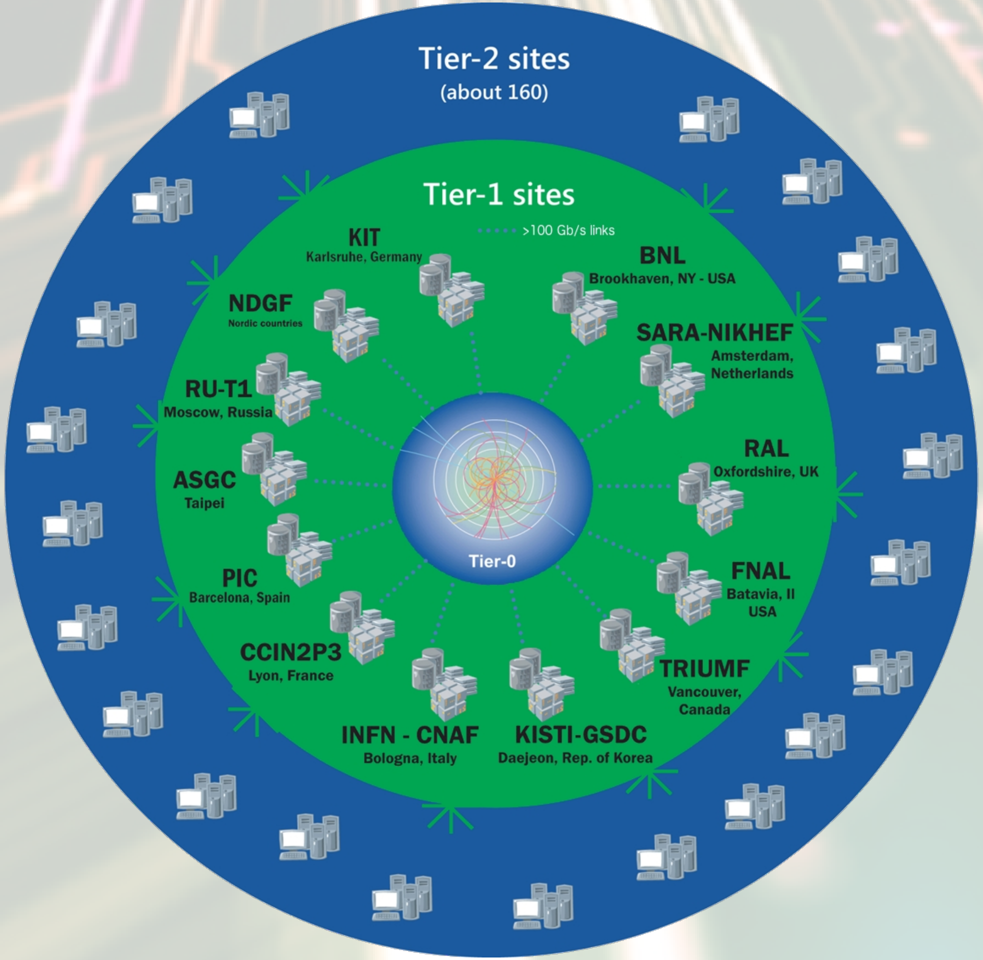


The PROBLEM



https://cloud.datapane.com/reports/dkjK28A/big-data-2021/

The SOLUTION



WLCG
Worldwide LHC Computing Grid

Mission: “provide global computing resources for the storage, distribution and analysis of the data generated by the LHC.”

Today:

1.4 million computer cores

1.5 exabytes of storage

170+ sites in 42 countries

The world's largest computing grid

CERN: 20% of total resources

WLCG

WLCG Worldwide LHC Computing Grid

LHC experiments were designed and built from 1984 onwards

- the challenge of LHC computing was only tackled seriously only at the end of the '90s

Mission: collect, distribute, process and **preserve (!)** data

- **Data:** not just "physics" → also documentation, software + environment, know-how...

Time changes also the perspective:

- 100TB per LEP (Large Electron-Positron Collider) experiment: immensely challenging at the time → now "trivial" for both CPU and storage

Grid: 24/7 service globalwide (was not trivial at the beginning)

- Distributed Computing = Distributed Spending (and = Distributed Discoveries!)

Recent emerging challenges: adoption of opportunistic resources

– High Performance Computing (supercomputers)

– Volunteer Computing (general public, LHC@Home → ~1%)

Other related topics with increasing importance: sustainability, carbon footprint, electricity costs...

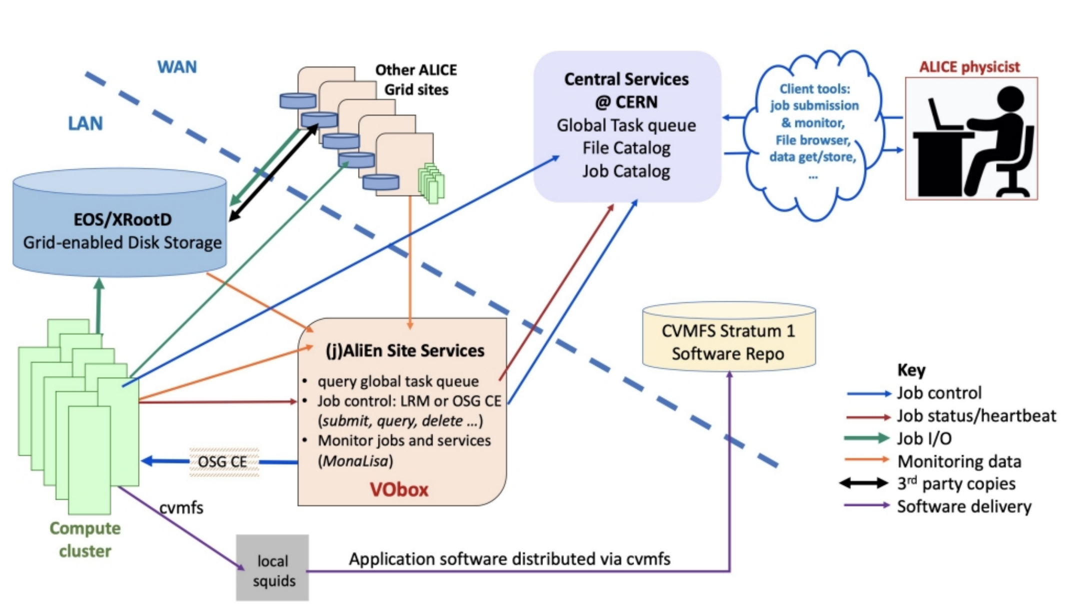
WLCG

WLCG
Worldwide LHC Computing Grid

Main Grid ingredients:

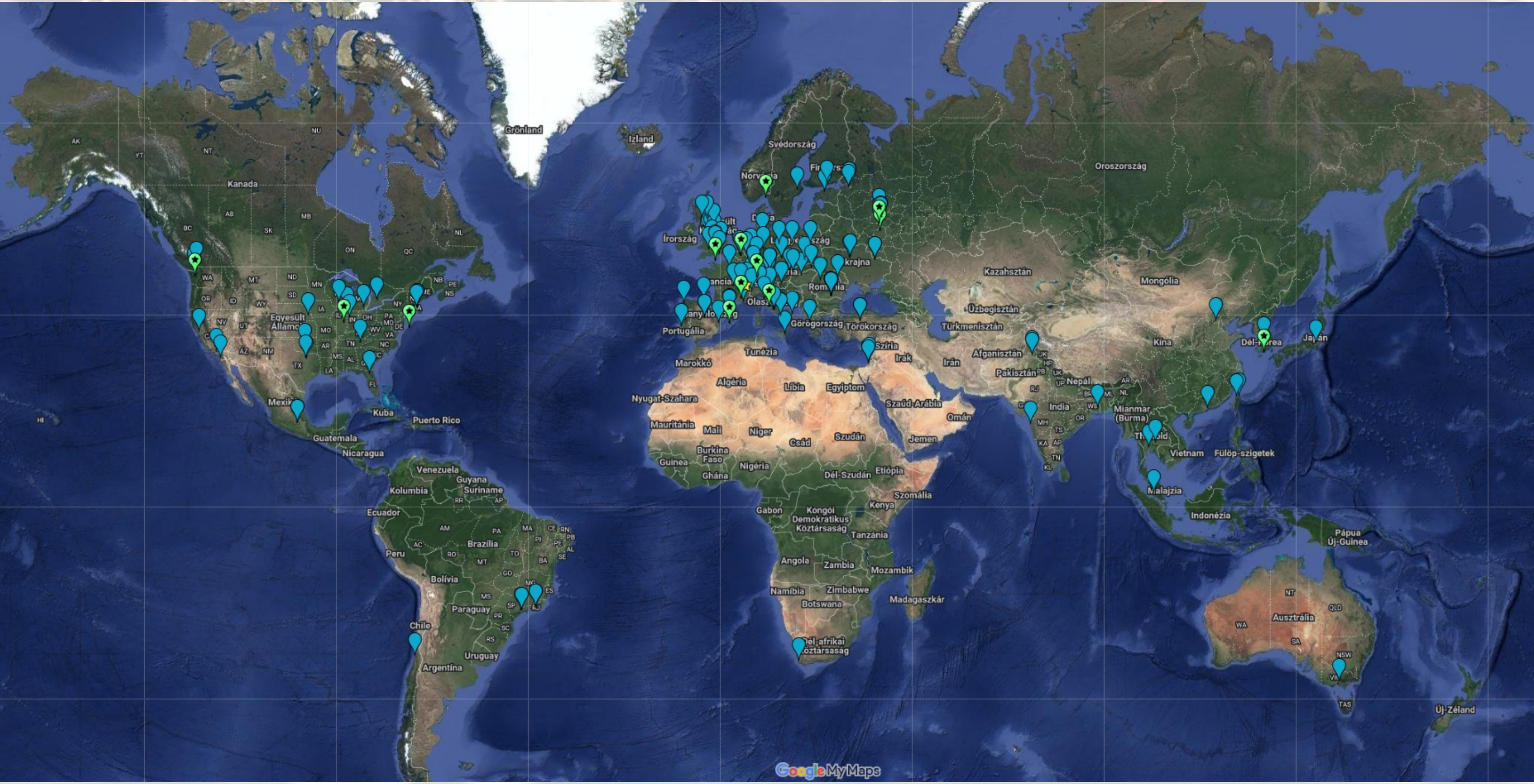
- Computing Elements
- Storage Elements
- Information system
  - You should only be sure that the data that you want are on the grid
  - Don't need to know where the data is
  - Don't need to know where your job is going to be running

- X509-based authentication



https://conferences.lbl.gov/event/644/#2-discussion-of-doe-review-tal

High Performance Computing	vs	**Grid:**
– Massively parallel	vs	(single-node) workload
– Low I/O	vs	high I/O
– Restricted environment	vs	open environment
– Multiple CPU/OS flavours	vs	single (CPU)/OS flavours
– username/password	vs	x509 cerfiticate



Grönland
Izland
Svédország
Oroszország
Kanada
Írország
Portugália
Görögország
Törökország
Kazahsztán
Mongólia
Üzbegisztán
Türkmenisztán
Afganisztán
Pakisztán
Kína
Irán
Irak
India
Mianmar (Burma)
Vietnam
Fülöp-szigetek
Indonézia
Pápua Új-Guinea
Ausztrália
Új-Zéland
Marokkó
Tunézia
Algéria
Líbia
Egyiptom
Nyugat-Szahara
Szaúd-Arábia
Omán
Jemen
Mauritánia
Mali
Niger
Csád
Szudán
Burkina Faso
Guinea
Ghána
Nigéria
Dél-Szudán
Etiópia
Szomália
Kenya
Gabon
Kongói Demokratikus Köztársaság
Tanzánia
Angola
Zambia
Mozambik
Namíbia
Zimbabwe
Botswana
Madagaszkár
Kuba
Puerto Rico
Guatemala
Nicaragua
Venezuela
Guyana
Suriname
Kolumbia
Ecuador
Brazília
Peru
Bolívia
Paraguay
Chile
Uruguay
Argentína
Google My Maps

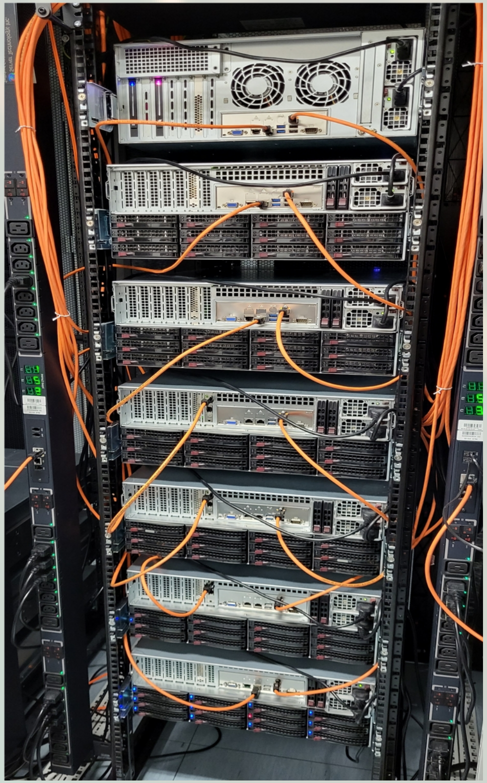
Tier-2 + AF in Budapest

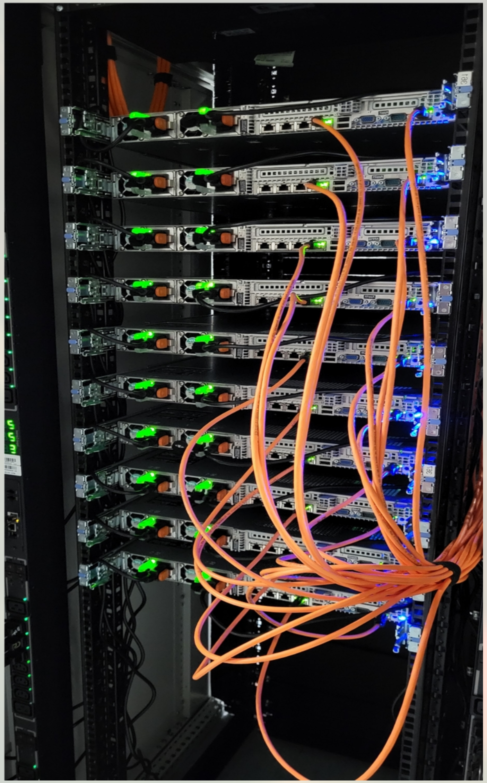
2003
7th site to join WLCG
50 cores (32bit) + 1.8 TB

2013

~600 cores (64bit) + 290 TB

2023



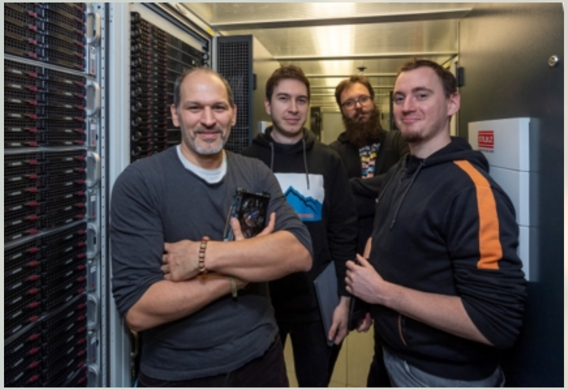


Since spring of 2022: at the Wigner DC

4000 cores (shared between CMS (2/3) and ALICE (1/3))

3.6 PB disk

Tier-2 + AF in Budapest



WSCLAB

WIGNER SCIENTIFIC COMPUTING LABORATORY

EXCELLENT RESEARCH INFRASTRUCTURE HUNGARY

(Other research projects)

- **HIJING++**
  - Next-gen heavy-ion Monte Carlo event generator
- **Machine learning for plasma channel profiling**
  - Monitoring tool for the AWAKE experiment
- **Modeling hadronization with Machine Learning techniques**
- **proton-CT**
  - Novel medical imaging method with ALICE-developed detectors
  - Accelerating the image reconstruction with Machine Learning
- **ALICE, CMS publications**
- **Contribution to COVID-19 research**
- **Gravitational waves**
  - Ligo/Virgo
- **High-precision calculations for nuclear reactor dynamics**
- **Collaboration Spotting**
- **QA centre for the ALICE TPC upgrade**
- **CRU development for DAQ**
- **Participation in the Quantum Technology Initiative**
  - National Quantum Technology Program
  - Dedicated grants
  - Investment on Maxeler DataFlow Machines 2022Q2
- ...

Simulation of Photonic Quantum Computers Enhanced by Data-Flow Engines

Special Theme 20 December 2021 Last Updated: 21 January 2022 Hits: 1889

by Peter Rakyta (ELTE), Ágoston Kaposi, Zoltán Kolarovszki, Tamás Kozsik (ELTE), and Zoltán Zimborás (Wigner)

Introducing HIJING++: the Heavy Ion Monte Carlo Generator for the High-Luminosity LHC Era

Gábor Bíró[a,b], Gergely Gábor Barnaföldi[b], Gábor Papp[a], Miklos Gyulassy[b,c,d,e], Péter Lévai[b], Xin-Nian Wang[c,d] and Ben-Wei Zhang[c]

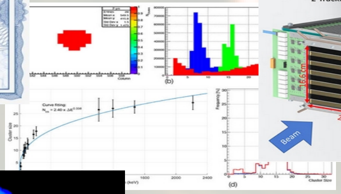
PHYSICAL REVIEW D 105, 114022 (2022)

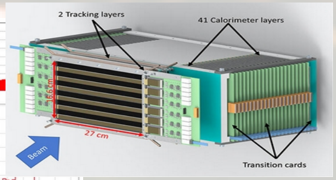
Estimating elliptic flow coefficient in heavy ion collisions using deep learning

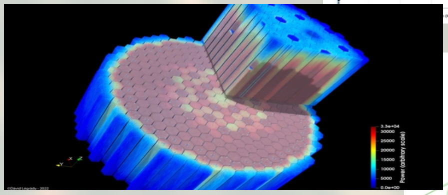
Neelkamal Mallick,[1] Suraj Prasad,[1] Aditya Nath Mishra,[2] Raghunath Sahoo,[1,3,*] and Gergely Gábor Barnaföldi[2]

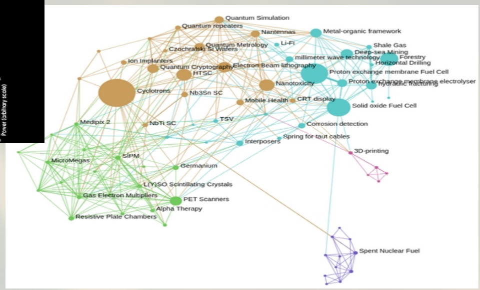
Machine learning methods for Schlieren imaging of a plasma channel in tenuous atomic vapor

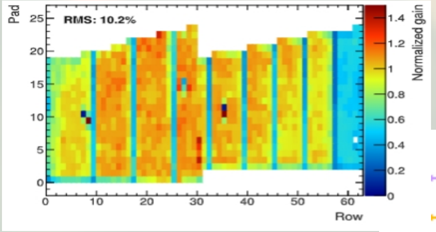
Gábor Bíró,[1] Mihály Pocsai,[1] Imre F. Barna,[1] Joshua T. Moody,[2] and Gábor Demeter[1]

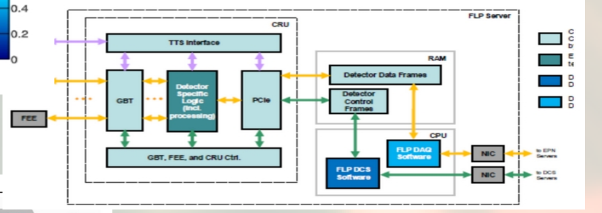




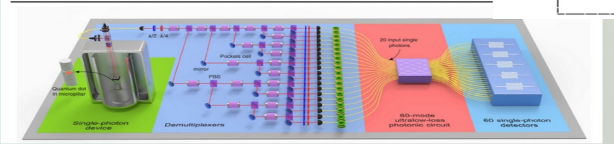








PHYSICAL REVIEW LETTERS 123, 250503 (2019)



How does it look like in practice?

You will need:

0) Some coding skills

Analysis codes (also in small scales), collaboration frameworks...

1) Authentication and authorization

Registration with an LHC-recognised VO (~usership at an experiment), CERN Grid User certificate

2) Have a working environment

The way to submit an analysis can be very different among the experiments

3) The **Grid** itself

- Batch system, resource broker, resource discovery, accounting, monitoring...
- How much resource do I need? Which data do I need? Where is that data accessible? Where are resources available (close to the data)? What is the expected output?...

How does it look like in practice?

You will need:

0) Some coding skills

Analysis codes (also in small scales), collaboration frameworks...

1) Authentication and authorization

Registration with an LHC-recognised VO (~usership at an experi

2) Have a working environment

The way to submit an analysis can be very different among the

3) The Grid itself

- Batch system, resource broker, resource discovery, accounti
- How much resource do I need? Which data do I need? Wh available (close to the data)? What is the expected output?...

```
}
//
AliAnalysisTaskMyTask::AliAnalysisTaskMyTask(const char* name) : AliAnalysisTaskSE(name),
    fAOD(0), fOutputList(0), fHistPt(0)
{
    // constructor
    DefineInput(0, TChain::Class());    // define the input of the analysis: in this case we take a
                                        // this chain is created by the analysis manager, so no ne
                                        // it does its work automatically
    DefineOutput(1, TList::Class());    // define the output of the analysis: in this case it's a
                                        // you can add more output objects by calling DefineOutput
                                        // if you add more output objects, make sure to call PostDa
                                        // make changes to your AddTask macro!
}
//_____________________________________________________________________________
AliAnalysisTaskMyTask::~AliAnalysisTaskMyTask()
{
    // destructor
    if(fOutputList) {
        delete fOutputList;     // at the end of your task, it is deleted from memory by calling t
    }
}
//_____________________________________________________________________________
void AliAnalysisTaskMyTask::UserCreateOutputObjects()
{
    // create output objects
    //
    // this function is called ONCE at the start of your analysis (RUNTIME)
    // here you ceate the histograms that you want to use
    //
    // the histograms are in this case added to a tlist, this list is in the end saved
    // to an output file
    //
    fOutputList = new TList();          // this is a list which will contain all of your histogram
                                        // at the end of the analysis, the contents of this list a
                                        // to the output file
    fOutputList->SetOwner(kTRUE);       // memory stuff: the list is owner of all objects it conta
                                        // if requested (dont worry about this now)

    // example of a histogram
    fHistPt = new TH1F("fHistPt", "fHistPt", 100, 0, 10);       // create your histogra
    fOutputList->Add(fHistPt);          // don't forget to add it to the list! the list will be wr
                                        // your histogram in the output file, add it to the list!

    PostData(1, fOutputList);           // postdata will notify the analysis manager of changes /
                                        // fOutputList object. the manager will in the end take ca
                                        // so it needs to know what's in the output
}
//_____________________________________________________________________________
void AliAnalysisTaskMyTask::UserExec(Option_t *)
{
    // user exec
```

How does it look like in practice?

You will need:

0) Some coding skills

Analysis codes (also in small scales), collaboration frameworks...

1) Authentication and authorization

Registration with an LHC-recognised VO (~usership at an experiment), CERN Grid User certificate

2) Have a working environment

The way to submit an analysis can be very different

3) The **Grid** itself

- Batch system, resource broker, resource discover
- How much resource do I need? Which data do available (close to the data)? What is the expected output?...

```
[gbiro@lxplus810][~]
$ /cvmfs/alice.cern.ch/bin/alienv enter AliPhysics/vAN-20230920_02-1
[AliPhysics/vAN-20230920_02-1] ~ > alien-token-init gbiro
Enter PEM pass phrase:
DN >>> C=ch/O=AliEn2/CN=Users/CN=gbiro/OU=gbiro
ISSUER >>> C=ch/O=AliEn2/CN=AliEn CA
BEGIN >>> 2023-10-11 12:53:31
EXPIRE >>> 2023-11-11 15:53:31
[AliPhysics/vAN-20230920_02-1] ~ > alien.py
Welcome to the ALICE GRID
support mail: adrian.sevcenco@cern.ch

AliEn[gbiro]:/alice/cern.ch/user/g/gbiro/ >
```

WLCG Worldwide LHC Computing Grid

How does it look like in practice?

You will need:

0) Some coding skills

Analysis codes (also in small scales), collaboration frameworks...

1) Authentication and authorization

Registration with an LHC-recognised VO (~usership at an experiment), CERN Grid User certificate

2) Have a working environment

The way to submit an analysis can be very different among the experiments

3) The

- Batc
- How avai

```
[AliPhysics/vAN-20230920_02-1] ~ > alienv list
Currently Loaded Modulefiles:
 1) BASE/1.0                          10) libpng/v1.6.34-124      19) lzma/v5.2.3-54                     28) VMC/v2-0-30            37) AliRoot/v5-09-59f_02-1
 2) GCC-Toolchain/v10.2.0-alice2-9    11) sqlite/v3.15.0-46       20) GSL/v1.16-95                       29) GEANT3/v4-1-18         38) RooUnfold/V02-00-01-alice5-98
 3) AliEn-Runtime/v2-19-le-113        12) libffi/v3.2.1-41        21) libxml2/v2.9.3-66                  30) ZeroMQ/v4.3.3-32       39) treelite/8498081-23
 4) FreeType/v2.10.1-75               13) Python/v3.9.12-2        22) XRootD/v5.5.0-2                    31) GEANT4/v11.0.3-1       40) KFParticle/v1.1-5-1
 5) protobuf/v3.15.8-12               14) Python-modules/1.0-322  23) TBB/v2021.5.0-21                   32) vgm/v5-0-20            41) AliPhysics/vAN-20230920_02-1
 6) OpenSSL/v1.1.1m-2                 15) lz4/v1.9.3-29           24) ROOT/v6-26-04-patches-alice2-8     33) GEANT4_VMC/v6-1-p2-2
 7) utf8proc/v2.6.1-13                16) boost/v1.75.0-79        25) DPMJET/v19.1.2-alice3-16           34) Vc/1.4.1-87
 8) Clang/v13.0.0-28                  17) xsimd/8.1.0-4           26) cgal/4.12.2-86                     35) xjalienfs/1.4.5-1
 9) zlib/v1.2.8-94                    18) arrow/v9.0.0-alice1-7   27) fastjet/v3.4.0_1.045-alice1-17     36) JAliEn-ROOT/0.6.8-5
[AliPhysics/vAN-20230920_02-1] ~ >
```

WLCG Worldwide LHC Computing Grid

How does it look like in practice?

You will need:

0) Some coding skills

Analysis codes (also in small scales), collaboration

```
void runAnalysis()
{
    // set if you want to run the analysis locally (kTRUE), or on grid (kFALSE)
    Bool_t local = kFALSE;
    // if you run on grid, specify test mode (kTRUE) or full grid model (kFALSE)
    Bool_t gridTest = kTRUE;

```

1) Authentication and authoriza

Registration with an LHC-recognised VO (~usership at an experiment), CERN Grid User certificate

2) Have a working environment

The way to submit an analysis can be very different among the experiments

3) The **Grid** itself

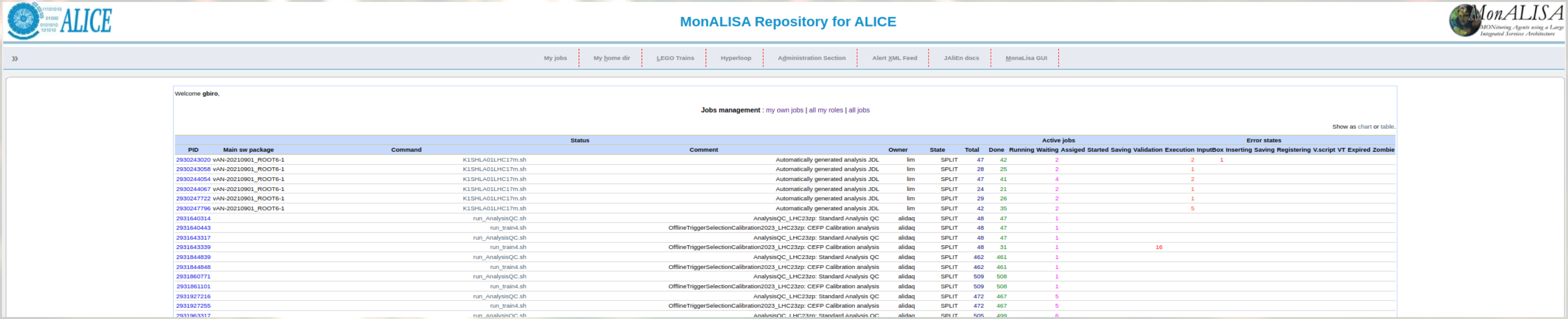
- Batch system, resource broker, resource discovery, accounting, monitoring...
- How much resource do I need? Which data do I need? Where is that data accessible? Where are resources available (close to the data)? What is the expected output?...

WLCG

Worldwide LHC Computing Grid

How does it look like in practice?

You will need:



ALICE

MonALISA Repository for ALICE

MonALISA — MONitoring Agents using a Large Integrated Services Architecture

» My jobs | My home dir | LEGO Trains | Hyperloop | Administration Section | Alert XML Feed | JAliEn docs | MonaLisa GUI

Welcome **gbiro**,

Jobs management : my own jobs | all my roles | all jobs

Show as chart or table.

		Status						Active jobs						Error states								
PID	Main sw package	Command	Comment	Owner	State	Total	Done	Running	Waiting	Assiged	Started	Saving	Validation	Execution	InputBox	Inserting	Saving	Registering	V.script	VT	Expired	Zombie
2930243020	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	47	42		2					2	1							
2930243058	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	28	25		2					1								
2930244054	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	47	41		4					2								
2930244067	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	24	21		2					1								
2930247722	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	29	26		2					1								
2930247796	vAN-20210901_ROOT6-1	K1SHLA01LHC17m.sh	Automatically generated analysis JDL	lim	SPLIT	42	35		2					5								
2931640314		run_AnalysisQC.sh	AnalysisQC_LHC23zp: Standard Analysis QC	alidaq	SPLIT	48	47		1													
2931640443		run_train4.sh	OfflineTriggerSelectionCalibration2023_LHC23zp: CEFP Calibration analysis	alidaq	SPLIT	48	47		1													
2931643317		run_AnalysisQC.sh	AnalysisQC_LHC23zp: Standard Analysis QC	alidaq	SPLIT	48	47		1													
2931643339		run_train4.sh	OfflineTriggerSelectionCalibration2023_LHC23zp: CEFP Calibration analysis	alidaq	SPLIT	48	31		1				16									
2931844839		run_AnalysisQC.sh	AnalysisQC_LHC23zp: Standard Analysis QC	alidaq	SPLIT	462	461		1													
2931844848		run_train4.sh	OfflineTriggerSelectionCalibration2023_LHC23zp: CEFP Calibration analysis	alidaq	SPLIT	462	461		1													
2931860771		run_AnalysisQC.sh	AnalysisQC_LHC23zo: Standard Analysis QC	alidaq	SPLIT	509	508		1													
2931861101		run_train4.sh	OfflineTriggerSelectionCalibration2023_LHC23zo: CEFP Calibration analysis	alidaq	SPLIT	509	508		1													
2931927216		run_AnalysisQC.sh	AnalysisQC_LHC23zp: Standard Analysis QC	alidaq	SPLIT	472	467		5													
2931927255		run_train4.sh	OfflineTriggerSelectionCalibration2023_LHC23zp: CEFP Calibration analysis	alidaq	SPLIT	472	467		5													
2931963317		run_AnalysisQC.sh	AnalysisQC_LHC23zp: Standard Analysis QC	alidaq	SPLIT	505	499		6													

The way to submit an analysis can be very different among the experiments

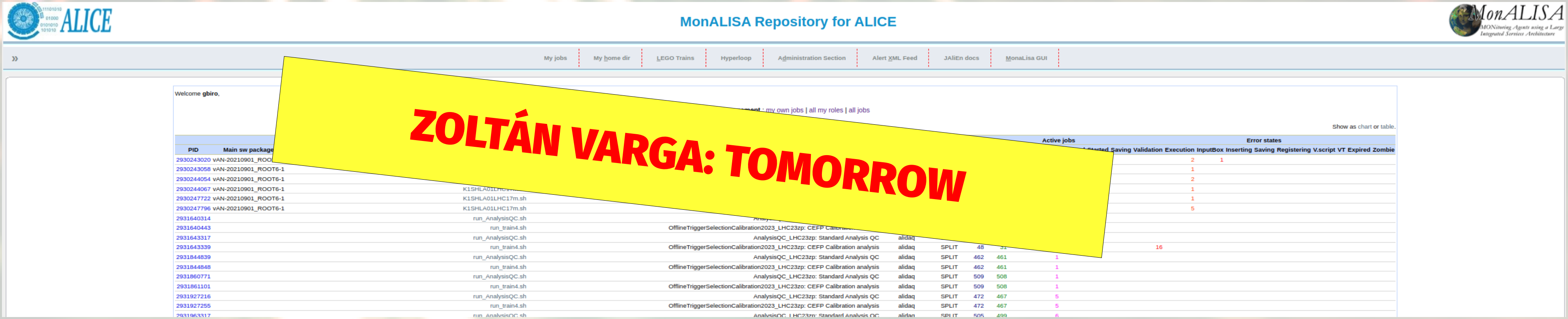
3) The **Grid** itself

- Batch system, resource broker, resource discovery, accounting, monitoring...
- How much resource do I need? Which data do I need? Where is that data accessible? Where are resources available (close to the data)? What is the expected output?...

WLCG

How does it look like in practice?

You will need:



The way to submit an analysis can be very different among the experiments

3) The **Grid** itself

- Batch system, resource broker, resource discovery, accounting, monitoring...
- How much resource do I need? Which data do I need? Where is that data accessible? Where are resources available (close to the data)? What is the expected output?...

Thank you!

Portals, resources to read:

- https://home.cern/science/computing
- https://wlcg-public.web.cern.ch/
- https://wigner.hu/hu/wsclab
- http://gpu.wigner.hu
- https://wignerdc.wigner.hu
- https://aliceo2group.github.io/analysis-framework/docs/
- https://alice-doc.github.io/alice-analysis-tutorial/
- https://docs.egi.eu/users/compute/cloud-compute/

Acknowledgement: OTKA K135515, NKFIH 2019-2.1.6-NEMZKI-2019-00011, 2020-2.1.1-ED-2021-00179, the Wigner Scientific Computing Laboratory and MILAB RRF-2.3.1-21-2022-00004.