Computing / Software needs in the next 3-5 years

S. Pagan Griso

Feb 23rd, 2024

Princeton
Muon Collider Organizational Workshop

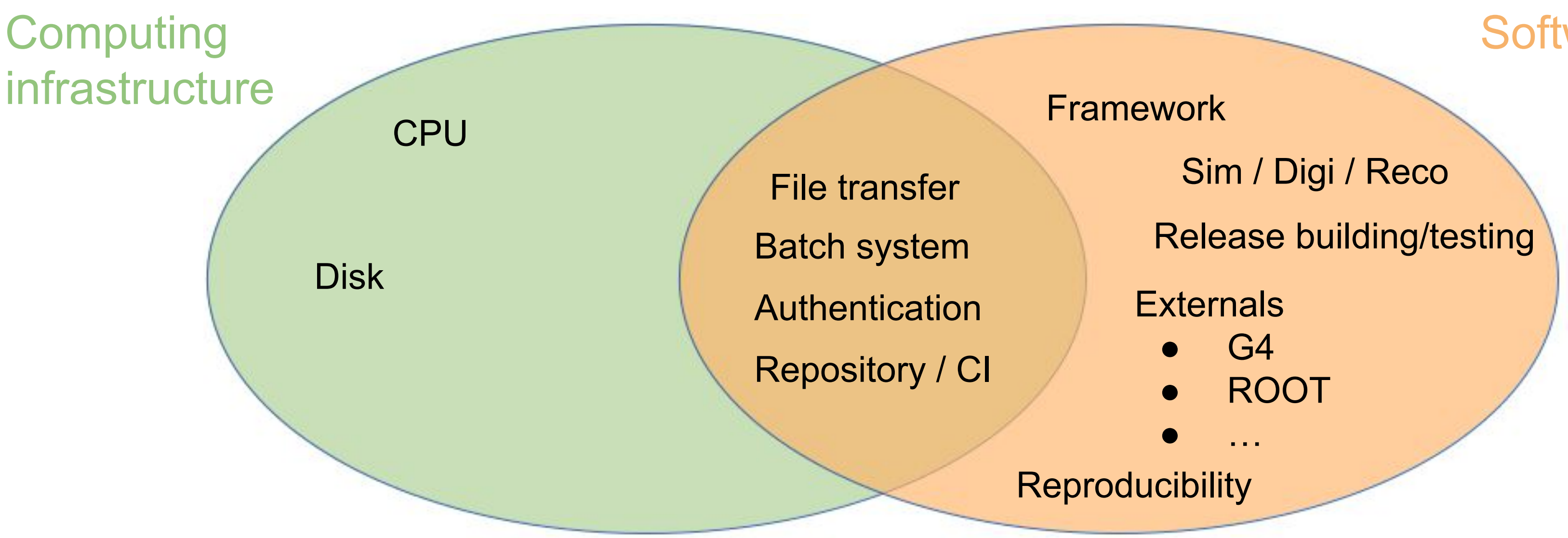
BERKELEY LAB
Lawrence Berkeley National Laboratory

Outline

Brief summary of state-of-art of Muon Collider computing and software

- focus on detector design and related tasks, but most is generic and applies to accelerator studies as well

Highlight where US contributions or dedicated resources would be beneficial



and reliable, updated documentation !!

Disk Resources

Disk resources for detector and accelerator studies mostly driven by large BIB

Approximate Size / Event [MB]			
heavily-filtered	trimmed	full truth	full truth (low threshold)
80	400	8,400	36,000

“Shared” resources available

Site	Disk Space [TB]	Notes
CERN	100	EOS
INFN-CNAF Tier-1	150	Grid Storage Element
CloudVeneto	90 + 70	INFN-based cloud storage
Snowmass.io	??	“Local” (network mounted) disk on cluster

Wish-list:

- some storage easily accessible by newcomers, input BIB for productions
- large storage where CPU is or easily reachable for large MC campaigns
    - output data should also be easily accessible

CPU Resources

CPU requirements for full simulation studies are the main driver

- BIB production and simulation
  - varying detector geometry, accelerator lattice, nozzle configuration, …
- performance and physics studies

Full simulation	Approximate time/event [min]
Physics + BIB Overlay	1-2
BIB simulation	1500

"Shared" resources available

Site	# CPUs	Notes
CERN	> 150	Condor Cluster, priority-based (~4GB memory / CPU)
INFN-CNAF Tier-1	6 CE	Condor Cluster
CloudVeneto	200	~4GB memory / CPU
Snowmass.io	??	Condor Cluster

Wish-list

- relatively homogeneous computing setup to ease using multiple resources
- ratio of CPU/memory a concern currently, since reconstruction jobs around 8GB of RAM in processing one event
  - especially problematic for HPC centers
  - strong motivation for intra-event parallelization (multi-threading software)

CPU/Disk infrastructure

Resources are nothing if you can't easily interact with them

- authentication
- transfer files between storages
- run large jobs where storage is

Existing resources

Site	Authentication	File Transfer	Notes
CERN	CERN SSO	xroot, local access	Need CERN account
INFN-CNAF Tier-1	Grid VO muoncollider	grid tools (gsiftp, …)	Need CERN account + grid certificate + request to join muon collider VO
CloudVeneto	INFN-AAI		Possible but impractical for non-INFN users
Snowmass.io	Custom Account	scp, …	Quick account creation, but disconnected from everything else.

Harmonization of authentication and easy transfer of data across sites is on the radar of some of the INFN people.
Would be really good to ensure we adapt solutions that are compatible with what we could get in the US (not at all obvious)

CPU/Disk: resources needs (US focus)

First, should endure we can easily leverage existing resources

- unlikely to have large dedicated US resources in the short term

Ideal to have some US-based resources to facilitate studies

- snowmass cluster was an excellent example of a successful model!
- HPC centers would provide less out-of-the-box experience, but possible

Tried to to a toy model to estimate resource needs for the next ~5 years

- focus on disk space, which is likely the main bottleneck
- CPU-hours estimate based on a decently modern CPU in realistic systems with O(10-100) CPUs per nodes - large variability
- Using state-of-art software, that has plenty of room for improvement

Key guidelines:

- split by activity type
- estimate number of concurrent studies performed to be kept at the same time

Results on the next slide, but a few conclusions here:

- full simulation studies of full analysis-chain for key physics processes likely to be a major driver for disk space, will need to prioritize and maintain generated samples carefully

CPU/Disk: resources needs (US focus)

Bottom-up estimates	2024-2030				
	n. indep. studies	samples/study	events/sample	size/event [MB]	total size [GB]
Accelerator Design					
- BIB simulations (per lattice configuration)	1	5	10	8400	420
- MARS/Fluka developments	2	2	10	36000	1440
- hardware acceleration (e.g. GPU)					1000
- others	2	5			1000
BIB simulations (MDI optimizations, Detector configurations, ...)	2	10	10	36000	7200
Core software tools					
- framework development	2	5	1,000	403	4030
- analysis tools development	1	5	10,000	75.9	3793
Generators					1000
Performance					
- layout optimizations	5	5	1,000	75.9	1897
- digitization dev	5	5	100	403	1008
- reconstruction algorithms	5	10	1,000	403	20150
- common benchmark samples	1	10	10,000	403	40300
Physics					
- full simulation	10	5	100,000	75.9	379310
- fast simulation	10	10	10,000,000	0.01	10000
				TOT [TB]	**473**

Similar for CPU (backup). Happy to play more with this spreadsheet, but overall:

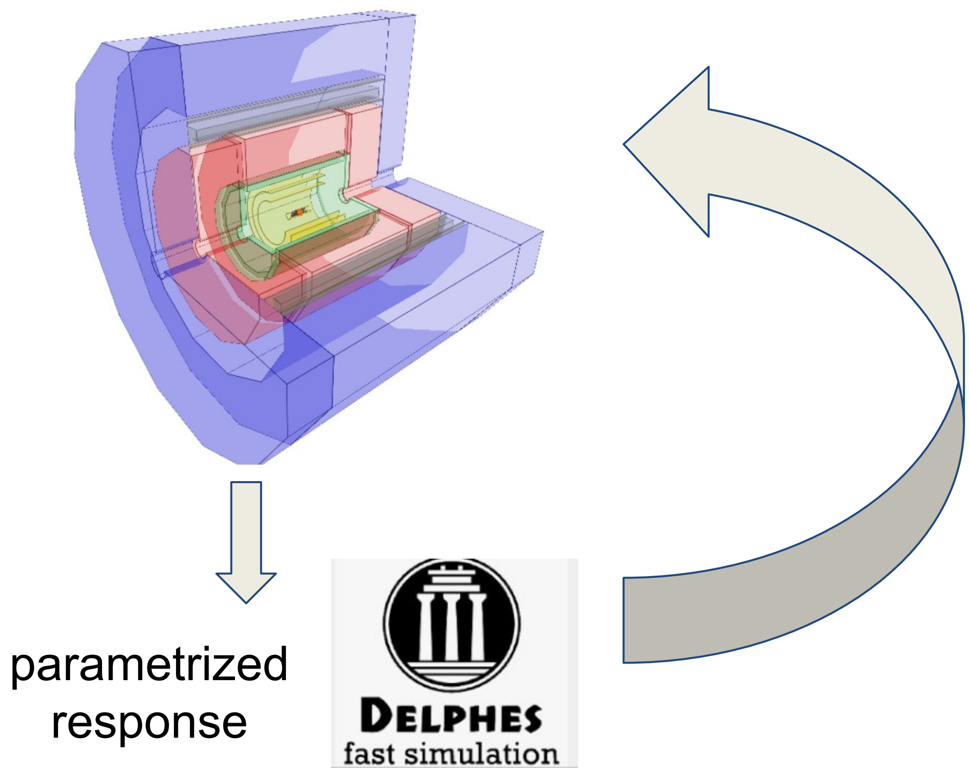
- O(100 TB), 100k CPU-hrs for most low-level studies should be a good start,
- ~500TB, 2M CPU-hrs for benchmark analyses using full simulations

Software

Large suite of softwares used for theory / detector / accelerator studies.

Theory needs:

- well maintained set of fast simulation parameters and variations
- close interaction with detector studies should be easy(ier)
- keep momentum in developing key software that allows accurate predictions of processes relevant for multi-TeV muon colliders



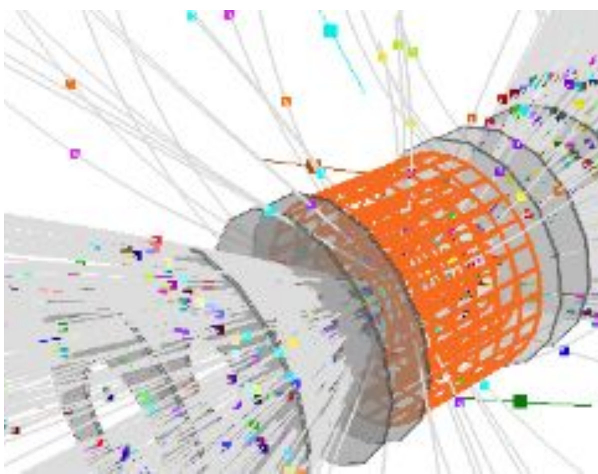
Software

Large suite of softwares used for theory / detector / accelerator studies.

Accelerator needs:

- many dedicated small software suites I'm sure I'm unaware of
  - usage of HPC centers for detailed (e.g. material) simulations
  - handled by dedicated grants and local resources? any need of shared resources?
- effort in developing and validating a FLUKA-based BIB simulation in a realistic accelerator design against reference MARS simulation results (MAP)
  - should MARS effort be revived or better to focus on a single common simulation effort?
  - detailed (and painful) low-level validation, debugging and optimization will be in continuous needs for the years to come

JINST **19** T02015
02/23/2024



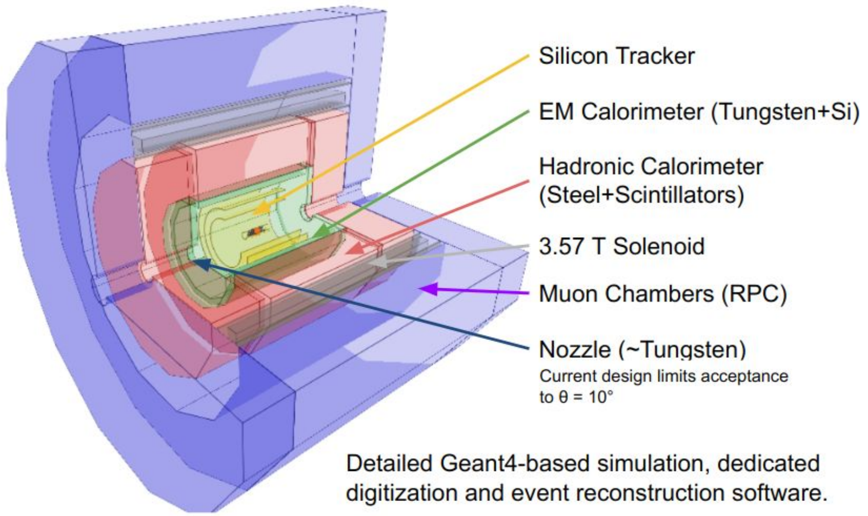
Simulation Source	MARS15		FLUKA		
Beam Energy [GeV]	62.5	750	750	1500	5000
MDI Optimization	yes	yes	yes	no	no
Muon decay length [m]	3.9×10^5	46.7×10^5	46.7×10^5	93.5×10^5	311.7×10^5
Muon decays/m per beam	51.3×10^5	4.3×10^5	4.3×10^5	2.1×10^5	0.64×10^5
γ/BX ($E_\gamma > 0.1$ MeV)	170×10^6	86×10^6	51×10^6	70×10^6	116×10^6
n/BX ($E_n > 1$ MeV)	65×10^6	76×10^6	110×10^6	91×10^6	89×10^6
$e^\pm$/BX ($E_e > 0.1$ MeV)	1.3×10^6	0.75×10^6	0.86×10^6	1.1×10^6	0.95×10^6
$h^\pm$/BX ($E_h > 0.1$ MeV)	0.011×10^6	0.032×10^6	0.017×10^6	0.020×10^6	0.034×10^6
$\mu^\pm$/BX ($E_\mu > 0.1$ MeV)	0.0012×10^6	0.0015×10^6	0.0031×10^6	0.0033×10^6	0.0030×10^6

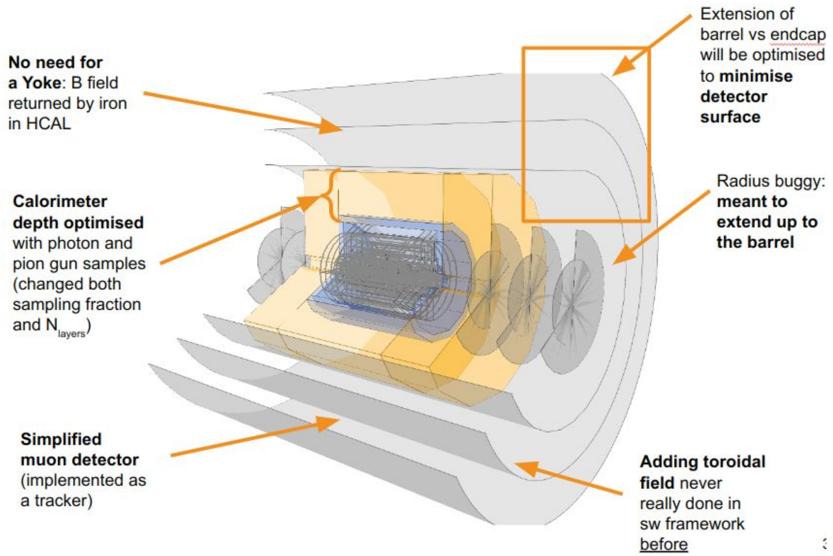
Software

Large suite of softwares used for theory / detector / accelerator studies.

Detector needs:

- the most critical piece of software is a unified framework to develop, test and validate a complete detector design **collaboratively** (see also next slides)
- Taskforce being setup to provide a roadmap for a simple, supported and unified way to run and develop such a code past from where we are
- Need to balance:
    - exploration of new ideas and concepts
    - realistic simulation that help defining technological requirements





Detector Software Infrastructure

Software repositories currently mostly on github

- gitlab group now available: https://gitlab.cern.ch/muon-collider/
- new code and migrations can be moved there, but no clear plan yet

Software distribution largely automated via gitlab, now three methods:

- CVMFS: `source /cvmfs/muoncollider.cern.ch/release/2.8/setup.sh`
  - can modified individual packages using `spack` commands
  - 300GB of space available at CERN for software distribution
- Container (docker, aptainer)
  - great out-of-the-box experience, older method to easily develop packages does not work anymore out of the box and needs fixing/work.
  - some mix of distributions to be sorted out (CERN and infn registry)
- Local install
  - requires native Alma Linux 9, good for heavy code developments touching multiple packages (e.g. EDM changes)

Overall a decently well covered area, but long-term maintenance and more user-friendly approach (vs power-users) urgently needed.

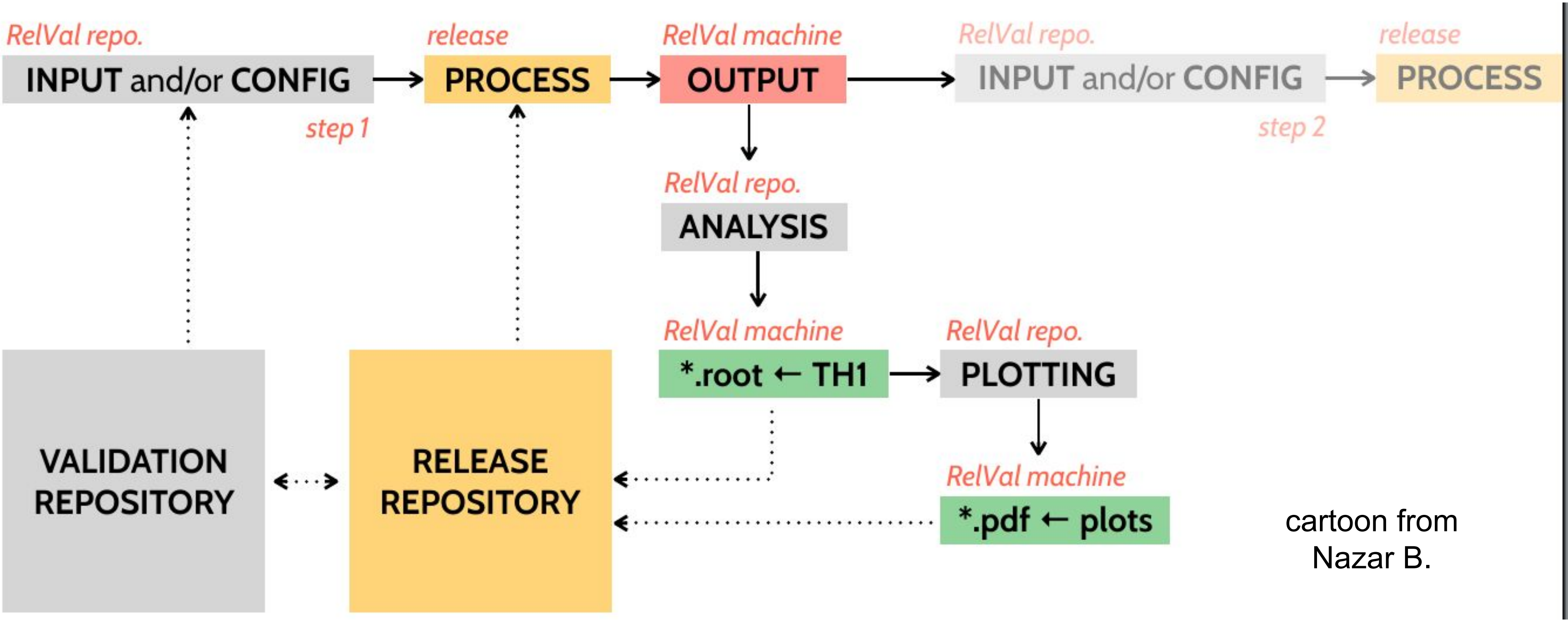
- Local (US) proxies that cache CVMFS would be a great performance boost!

Detector Software Release Validation

Work has started to create a robust software infrastructure to:

- validate changes in an integrated and unified way for everyone to use
- provide up-to-date workflows that can serve as baseline for further developments and usable by everyone (also easier to debug problems)
- keep track of common setups and settings that can easily be shared

This will be a critical tool for the developments in the next years



cartoon from Nazar B.

key4hep migration

Main components of our detector simulation software stack

1. **LCIO** → Event Data Model
2. **DD4hep** → Detector Geometry and simulation w/ GEANT 4
3. **Marlin** → Framework for event reconstruction
4. **ILCSoft** → How to put everything together…

Most component inherited from (old) ILC/CLIC based studies.

New software packages now becoming available, among many Key4Hep is emerging

- more modern approach, based on recent Gaudi framework
- flexible event data model
- supports multi-threading software (if algorithms are designed to do so)
- easy and flexible python based configuration (instead of .xml files)

key4hep migration

Main components of our detector simulation software stack

1. LCIO → EDM4Hep
2. DD4hep → DD4Hep
3. Marlin → Gaudi
4. ILCSoft → Spack

Short-term need to re-assess a good development environment for everyone

- already some effort duplication that needs to converge

Most algorithms are now running in a backward-compatibility mode

- does not take advantage of the new framework, practically

Lots of room of medium/long-term contributions in porting/improving/reinventing older algorithms to modern ones, natively in the new framework.

- excellent range of entry-point and advanced developments

Critical area, more details in backup.

Detector studies

Low-level geometry description and detector digitization

- short-term: development should focus on flexibility in testing different configurations more easily, and if possible to lower the resource usage
- medium-term: important to assess impact of approximations on technology requirements

Higher-level algorithm developments

- to prove we can extract the physics we need
- critical ingredient for technology requirements for detector R&D

Component	**Maturity (1 .. 5)**	**US effort / EU effort**
Inner Tracking	3	med/high
Calorimeter, Jets	3	low
Calorimeter, e/ɣ	1	med
Tau Reconstruction	1	high
Central muons	2	low
Fwd muons	1 (-> 3)	med/high

Any of these areas need still lots of development.

Plus many more under-developed areas

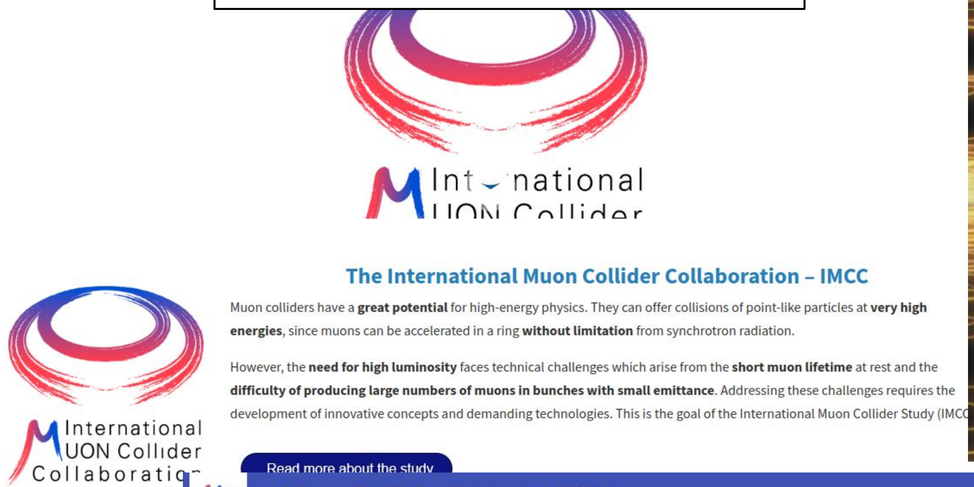
- essentially anything not listed..
- some key areas of attention, e.g. luminosity

Special mention: interaction region shielding requires unique and strong collaboration between detector, accelerator and theory experts!

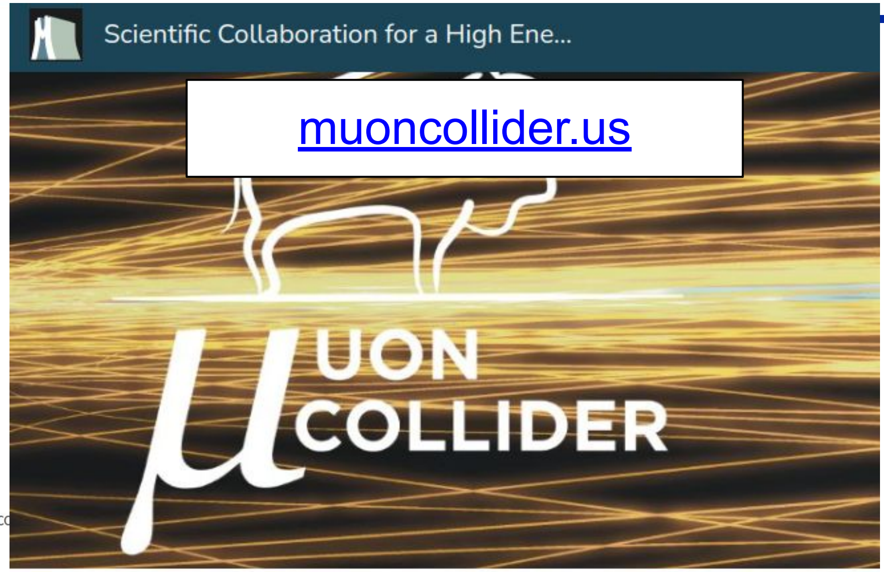
- lots achieved and lots to do

Documentation and Software support !!!

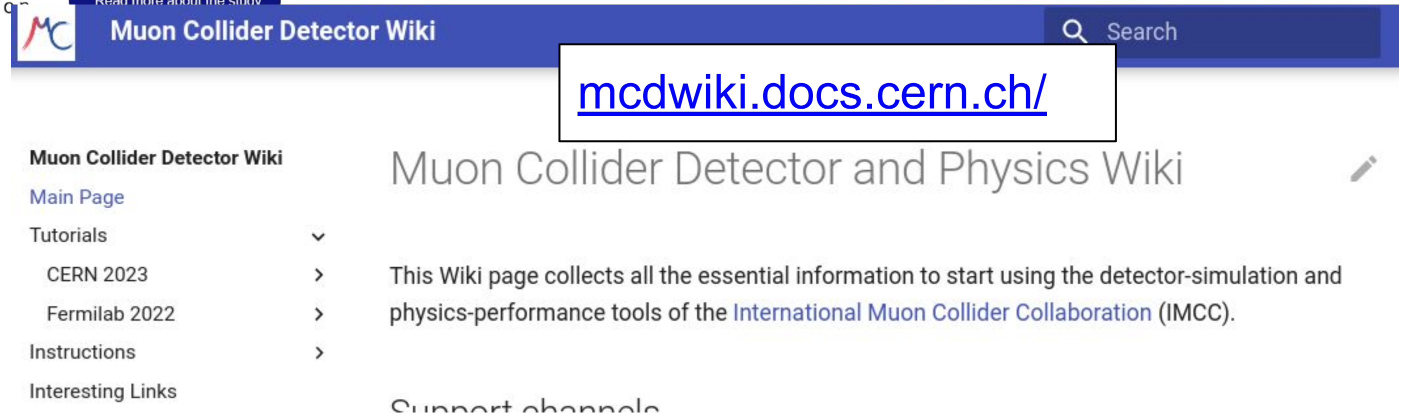
muoncollider.web.cern.ch



muoncollider.us



mcdwiki.docs.cern.ch/



Great entry point for learning and helping to keep our documentation useful !!

- and to ease the work of colleagues that have been doing that so far

Lots of needs for user support to facilitate newcomers' first studies.

Conclusions

Large ecosystem of software needed to support quantitative studies.

Lots of developments have happened in the last few years, but some of this software needs to qualitatively evolve to support a large(r) collaborative effort

- BIB simulation very resource-intensive, yet a necessary foundation of many of our realistic studies
- realistic detector simulation in need of a medium-term framework and necessitates significant person-power to make that transition
- user support and documentation are an integral critical piece of a functioning computing and software ecosystem

Dedicated computing resources in US will be useful to support a growing community

- short-term: make best use of what's available
- medium-term: prepare to ensure dedicated resources that are easily compatible with existing ones if at all possible
  - authentication and choice of underlying "middleware" software suite
  - preparation to make use of HPC centers efficiently

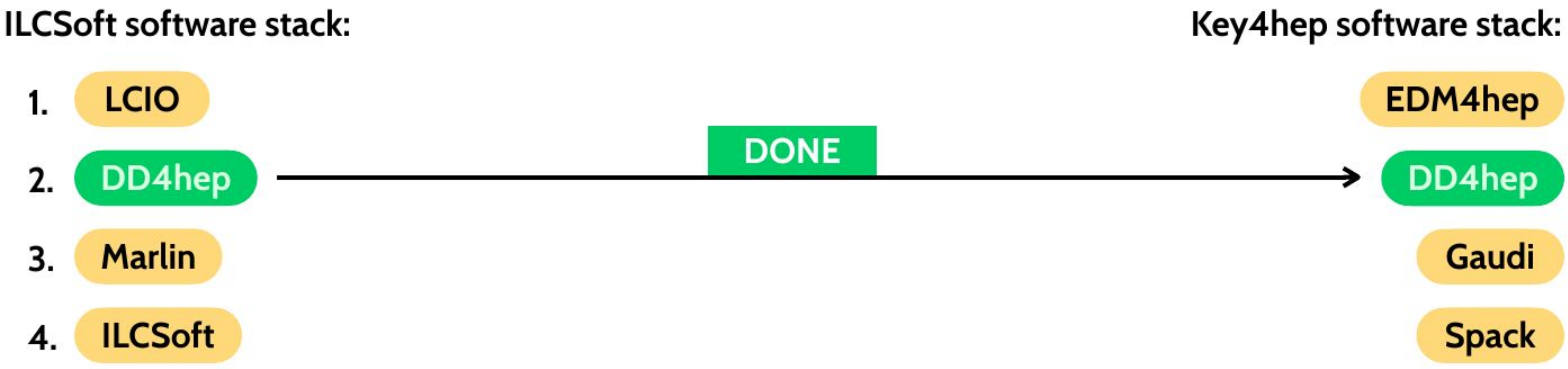
BACKUP

CPU projections

Bottom-up estimates	2024-2030				
	n. indep. studies	samples/study	events/sample	sec/event [s]	CPU hours [hrs]
Accelerator Design					
- BIB simulations (per lattice configuration)	1	10	10	90000	2500
- MARS/Fluka developments	2	20	10	90000	10000
- hardware acceleration (e.g. GPU)			10	90000	0
- others	2	10	10	90000	5000
BIB simulations (MDI optimizations, Detector config	2	10	10	90000	5000
Core software tools					
- framework development	2	50	1,000	130	3611
- analysis tools development	1	50	10,000	130	18056
Generators				130	0
Performance					
- layout optimizations	5	50	1,000	130	9028
- digitization dev	5	100	100	130	1806
- reconstruction algorithms	5	200	1,000	130	36111
- common benchmark samples	1	20	10,000	130	7222
Physics					
- full simulation	10	50	100,000	130	1805556
- fast simulation	10	250	10,000,000	0.01	69444
				TOT [CPU-hrs]	**1973333**

key4hep migration: status

original cartoons from Nazar B.



Geometry description is compatible with new key4hep software stack to first order

- some subtleties I think might be there, but can be address as we encounter them

key4hep migration: status

original cartoons from Nazar B.

Now fully migrated from custom scripts to modern package manager.

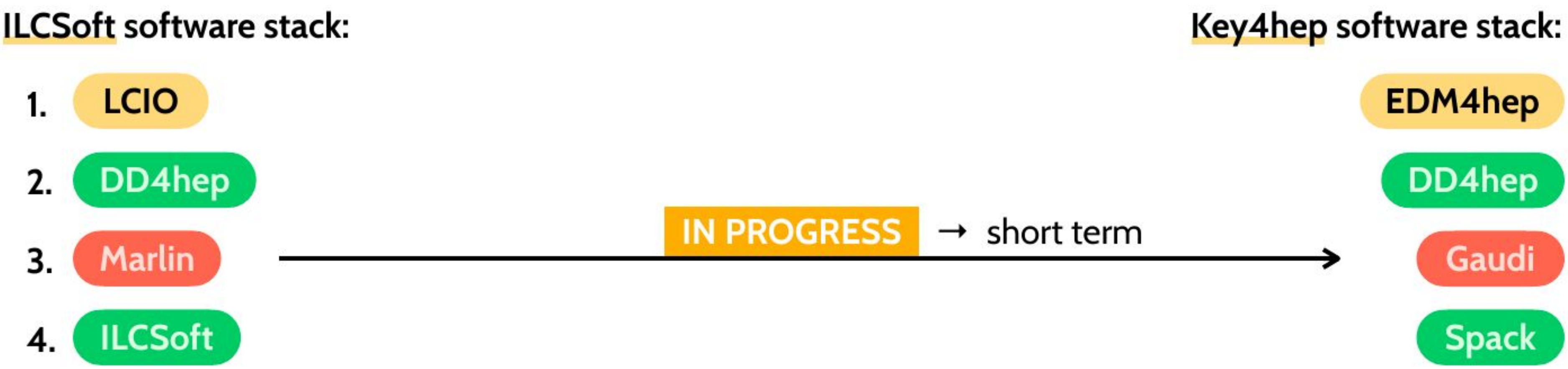
Makes setup very easy from cvmfs.
Easy out-of-the-box experience if you just want to “run things out of the box”.

To do development, some documentation exists but tutorials on how to use the new setup efficiently for software development are lacking details.

- more on documentation later
- some clear guidelines for software development still a bit lacking: key aspect where more input from various experience would be very valuable!

key4hep migration: status

original cartoons
from Nazar B.



Migration of algorithms tools to run in the new software stack.

Short-term solution: wrap existing algorithm.

- this is essentially done, at least in some implementations

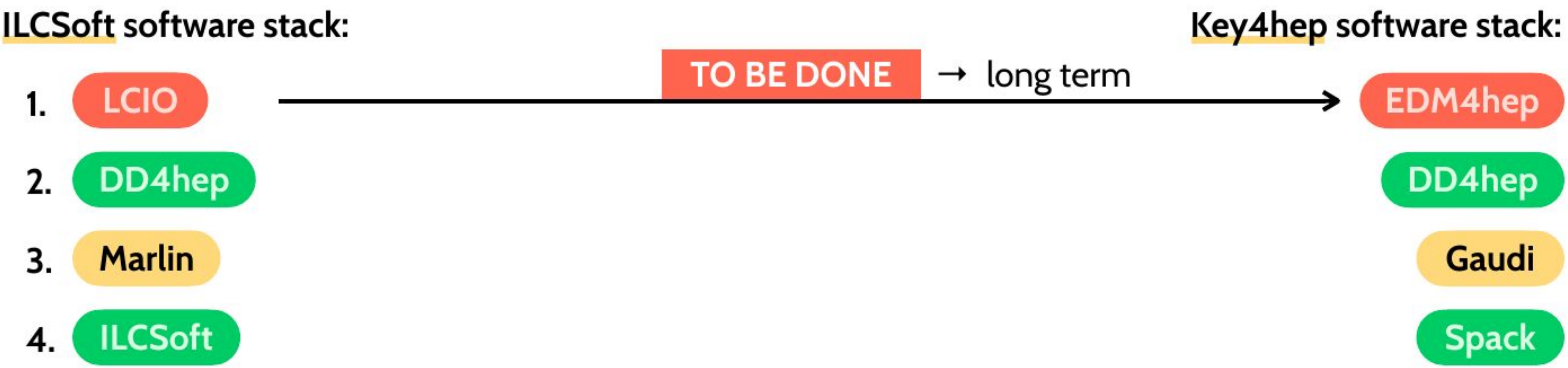
Runs ok, but can’t allow us to all of the native advantages of Gaudi

- configuration files can be migrated to python (some part of it already exists, but not integrated with “standard” software)
- algorithms need to be rewritten if to take advantage of multi-thread capabilities!

Lots of room for small (and large!) tasks that have lasting impact.

key4hep migration: status

original cartoons from Nazar B.



Event Data Model largely similar but not the same.

Requires rewriting algorithms to support it

- several technical issues and performance hit in delaying that, although doable with workarounds for the time being

Makes most sense to do together with algorithm migration to support multi-thread and native Gaudi layout.