

Task 1.5: New computing strategies for data modeling and interpretation (cont'd)

Daniele Massaro
CERN, IT-FTI-PSE
On behalf of NGT T1.5

Next Generation Triggers 1st Technical Workshop
25th November 2024



NextGen
Next Generation Triggers

Task 1.5

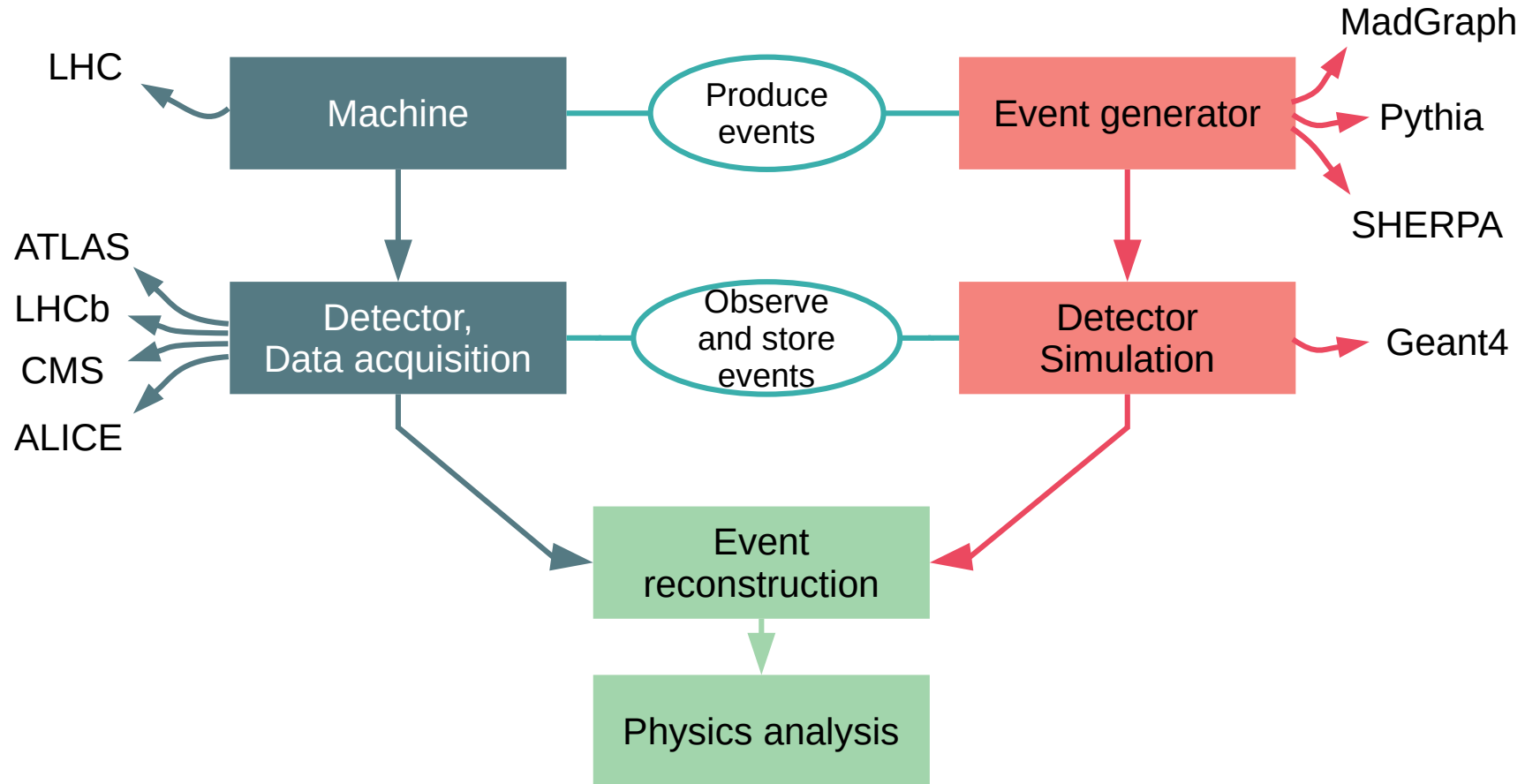
[...] The work package goals include: the porting and optimization of current event-generation codes and higher-order perturbative calculations to state-of-the-art and future hardware architectures, particularly GPUs; [...]

[NextGen Triggers Proposal, p. 19]

In this talk:

- event generators and their motivations;
- status of MadGraph port on GPU;
- plans for next year.

Event generators in the grand scheme



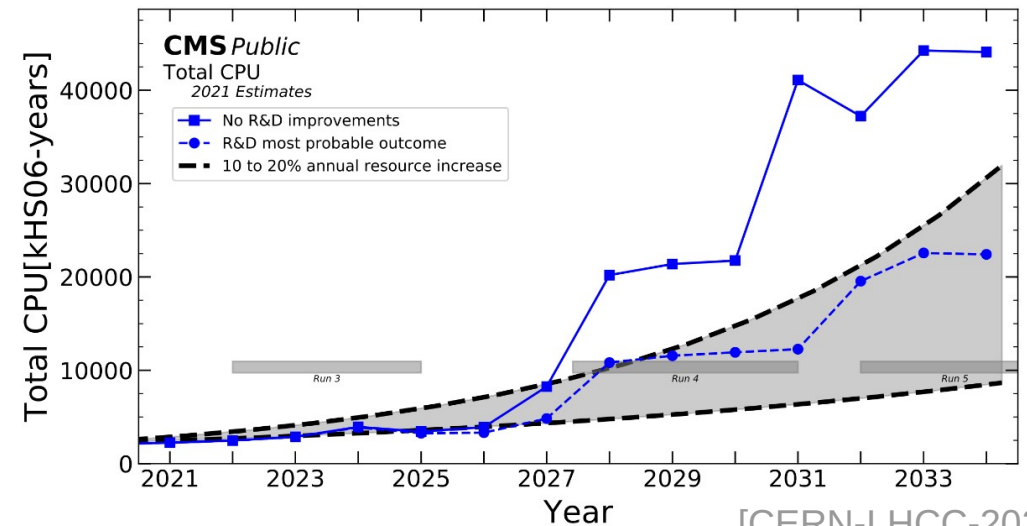
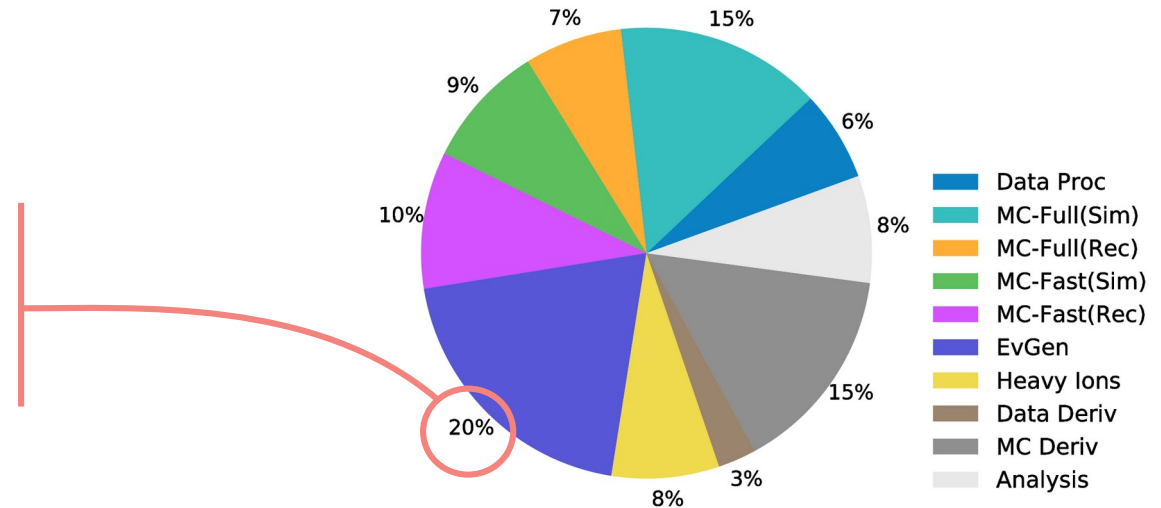
[T. Sjostrand, "Monte Carlo generators for the LHC (1/4)"]

Motivations

- Monte-Carlo event generation for HL-LHC is estimated to be **20%** of the total computing resources.
- Predicted CPU resources are not enough to cover experiments needs.

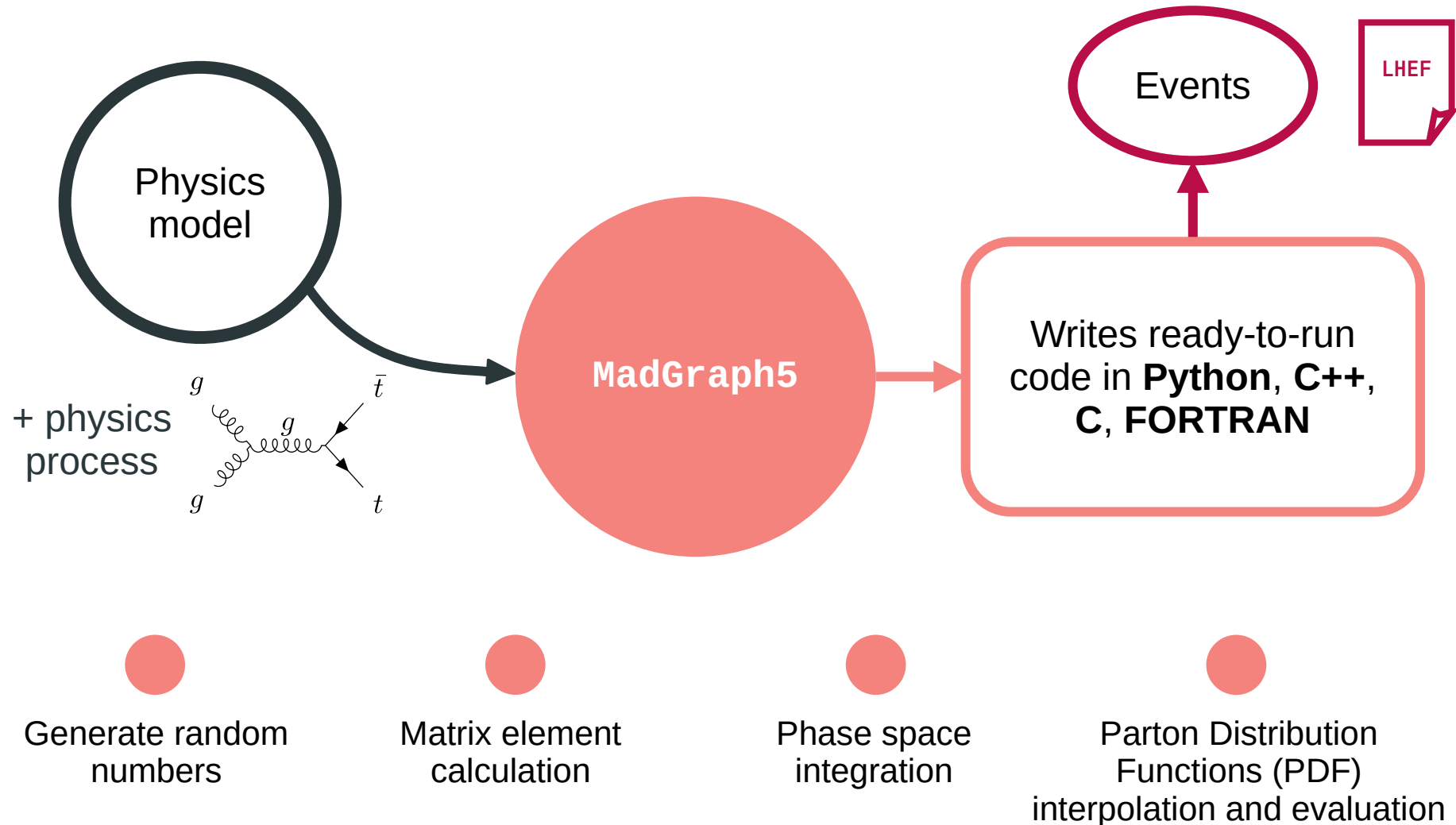
Solution: use GPUs!
 IT-FTI has been working since few years on this topic.

ATLAS Preliminary
 2020 Computing Model -CPU: 2030: Baseline



[CERN-LHCC-2022-005]

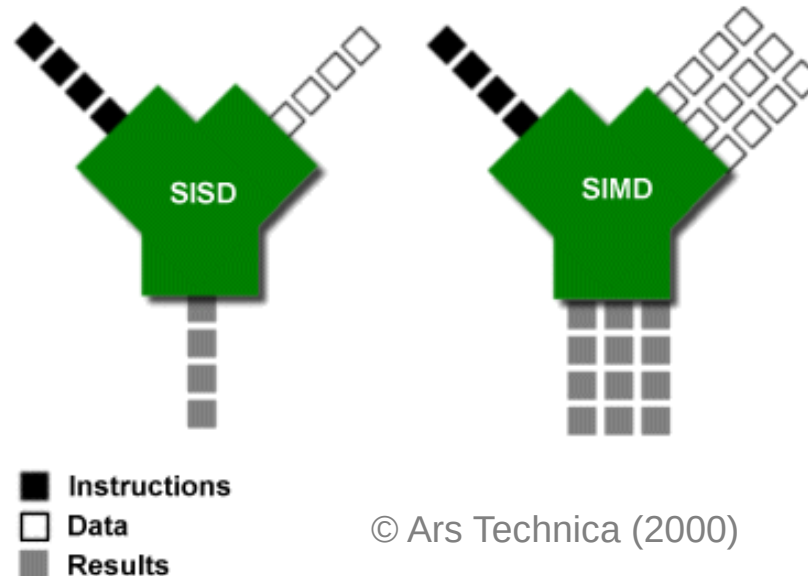
MadGraph5



Sequential vs data-parallel processing

Sequential processing

Single Instruction Single Data:
1 input and 1 output per cycle.



© Ars Technica (2000)

Data-parallel processing (lockstep processing)

Single Instruction Multiple Data:
N inputs and N outputs per cycle

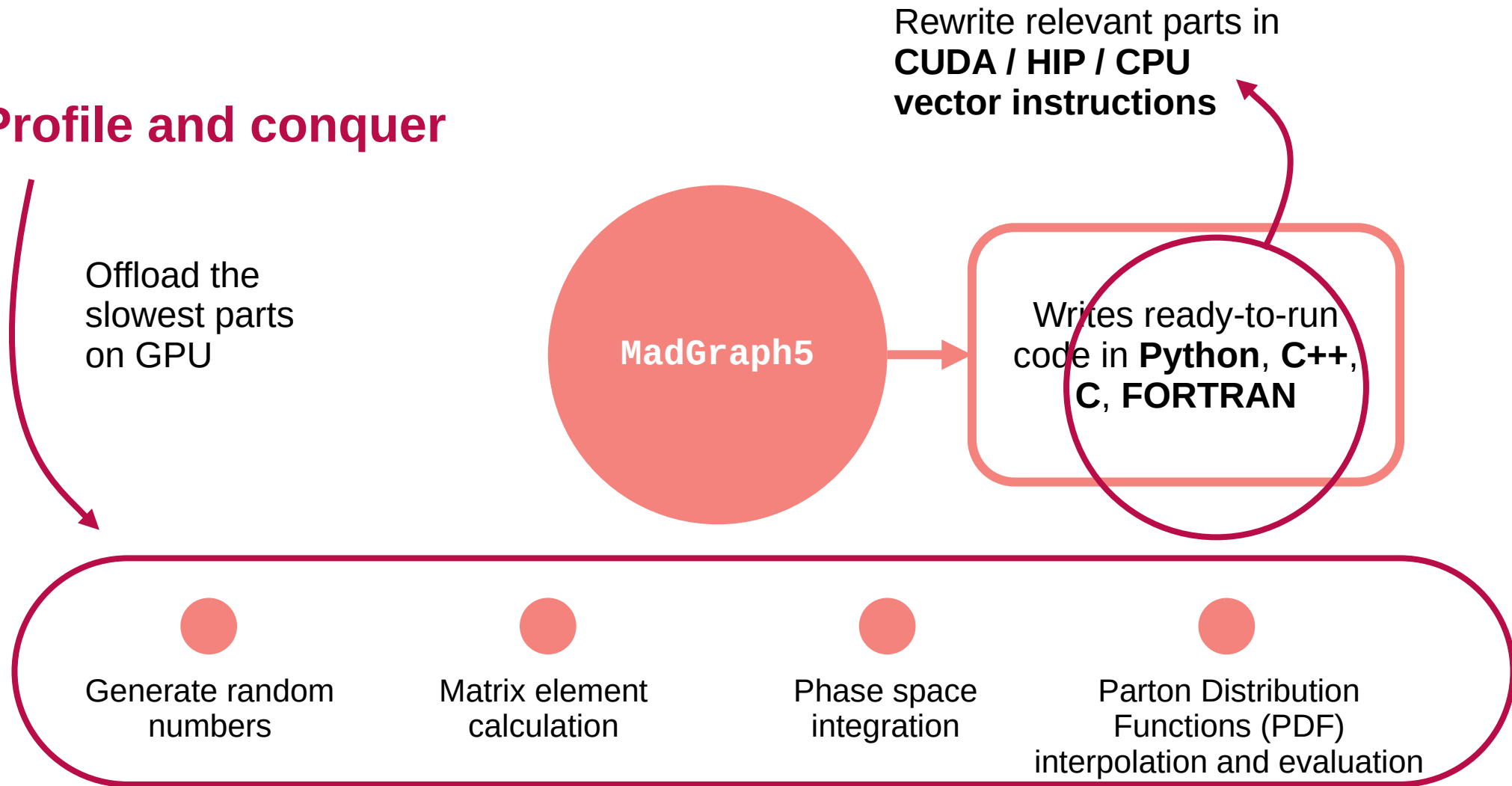
GPUs ⇒ SIMT
(CUDA)

Vector CPUs ⇒ SIMD
(C++)

2 main strategies to go from sequential to data-parallel

MadGraph5

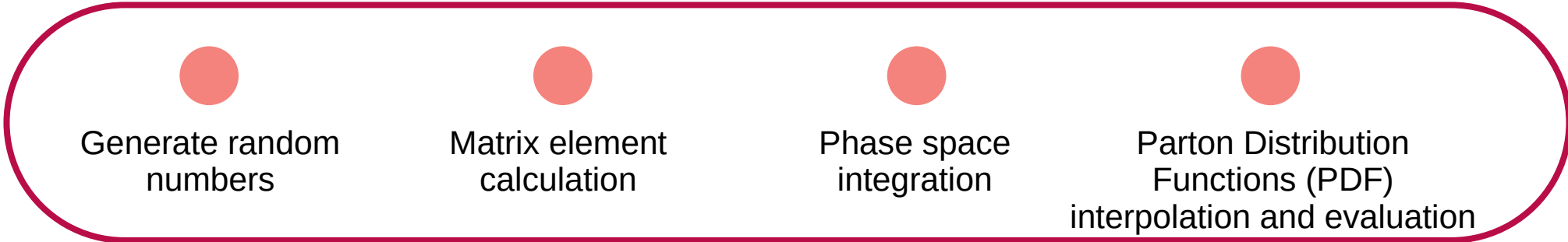
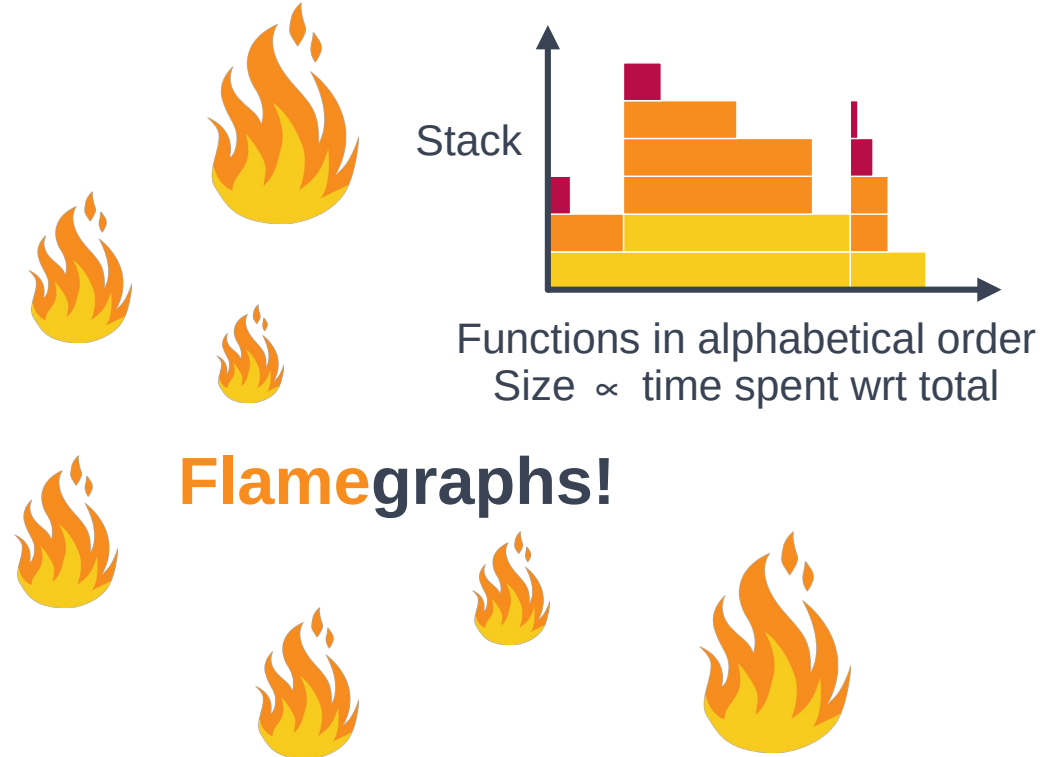
Profile and conquer



MadGraph5

Profile and conquer

Offload the slowest parts on GPU



g g → t t g g : FORTRAN

~ 97% running time



Generate random numbers

Matrix element calculation

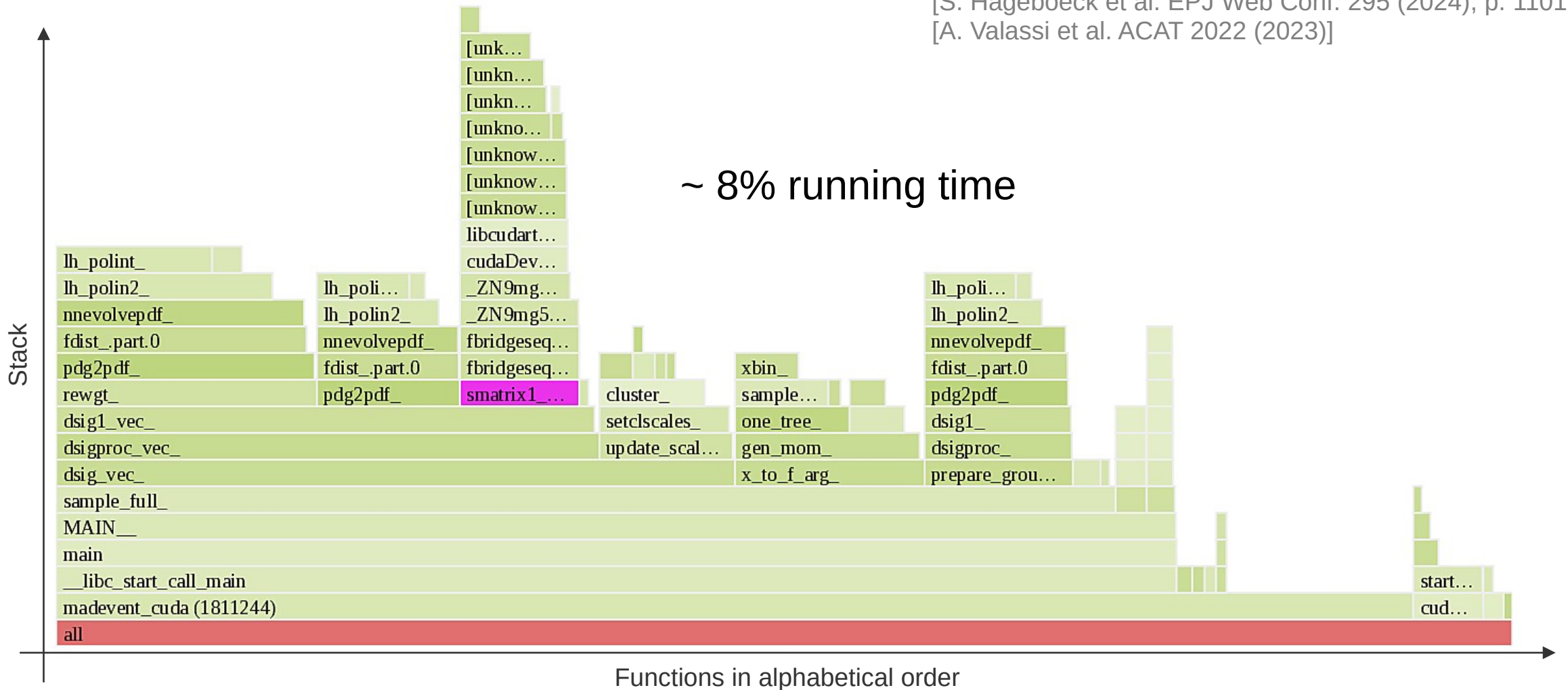
Phase space integration

Parton Distribution Functions (PDF) interpolation and evaluation

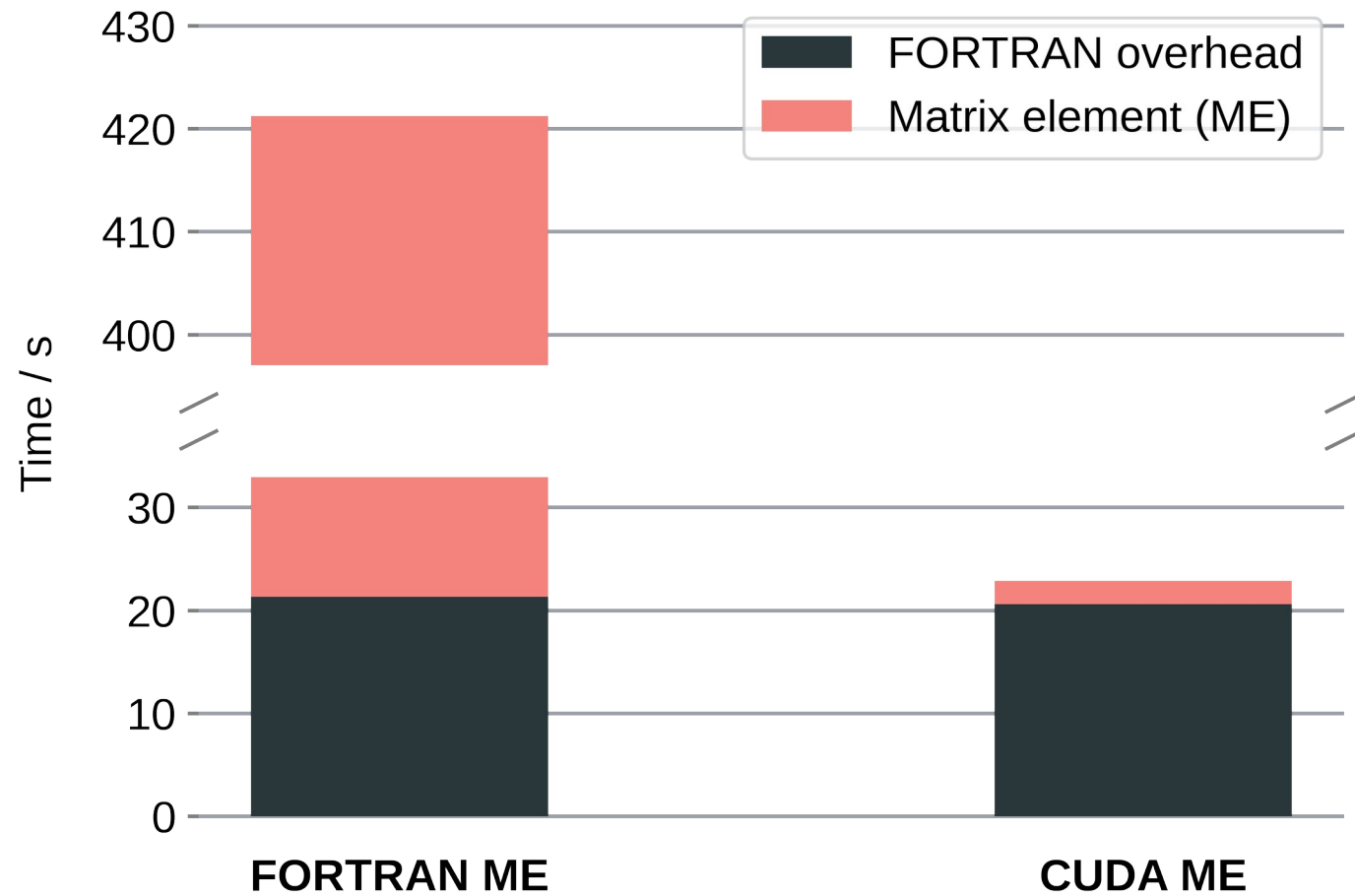
g g → t t̄ g g : CUDA

[S. Hageboeck et al. EPJ Web Conf. 295 (2024), p. 11013]
 [A. Valassi et al. ACAT 2022 (2023)]

~ 8% running time

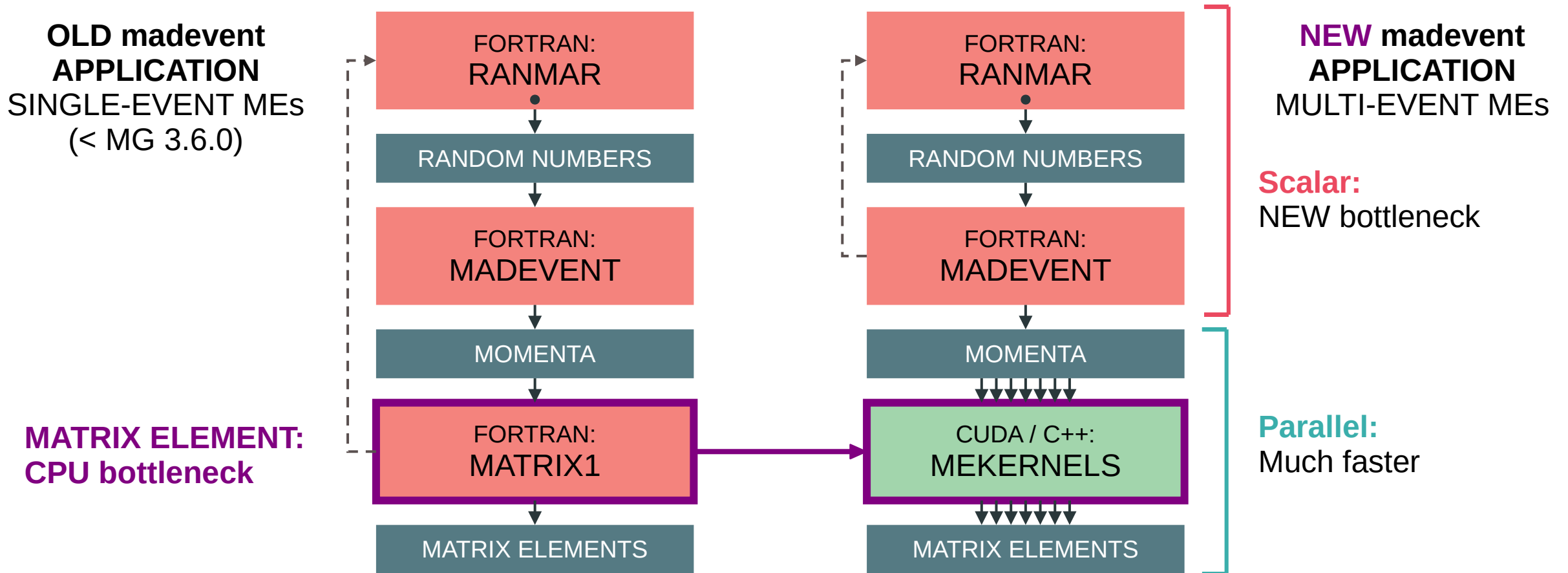


g g → t t g g : CUDA



Results obtained for GPU NVIDIA A100

The new architecture



Status 2024 (the leading-year)

[A. Valassi. CHEP24]

- First release for acceleration of **leading-order** processes on GPUs and vector CPUs available on **GitHub**:
 - Excellent collaboration with O. Mattelaer (UCLouvain, main MadGraph developer);
 - Already very good speedups of the application via hardware acceleration of matrix elements.
- Available for NVIDIA and AMD GPUs. SYCL port being worked on.
- Work on integration with CMS (S. Bhattacharya, J. Choi).



Madgraph5_aMC@NLO
on GPUs and vector CPUs:
towards production

*The 5-year journey to the
first LO release CUDACPP v1.00.00*

Andrea Valassi (CERN)

on behalf of the MG5AMC CUDACPP development team

[https://ir](https://...)

Quantifying the Computational Speedup with MG4GPU for CMS Workflow

Introduction
The most time-consuming aspect of Hard Scattering is the Matrix Element (ME) calculation. The Madgraph4GPU (MG4GPU) project is dedicated to porting the ME calculation to vectorized CPUs and GPUs.

First Implementation in CMS!

Physics Processes
Conducted extensive testing of the most common and complex LO processes within the CMS framework, encompassing:
– Drell-Yan (DY) production, capable of generating up to four jets
– Top-Quark Pair (TT) production, capable of producing up to three jets

HPC Configurations
– Lxplus HTCondor Pool with AMD EPYC 16 CPUs + A100 / Intel Xeon 48 CPUs + H100
– Seoul National University HTCondor Pool with Intel Xeon 88 CPUs
– NERSC Perimeter SLURM batch with AMD EPYC 40x CPUs

Performance Results
Gridpack Production

Production Time	FORTRAN	CPP-AVX2	CUDA
DY+tt2jj	424h 35m	133h 38m	9h 32m
TT+tt2jj	253h 35m	155h 28m	3h 5m

– Different levels of parallelism employed, normalized to 16 parallel madevent executions.
– CPP-AVX2 exhibits approximately 2-3x speedup; CUDA achieves O(10)x.

Event Generation – Jet-Binned study (TT)

– Multiple events are generated with a single madevent execution
– As the complexity of events increases, the production time required to execute then also tends to grow.
– Refining ME requires a significant amount of time for FORTRAN/PP-AVX2, resulting in substantial differences for inclusive TT and TT+3j, but not for CUDA.

Event Generation – Inclusive study

– Multiple events are generated with a single madevent execution, exhibiting a linear trend.
– CPP-AVX2 demonstrates approximately 1.5x speedup; CUDA achieves approximately 7x speedup when generating 100k events.

Conclusion
Madgraph demonstrates substantial speedups in both gridpack production and event generation for the CMS workflow, making it a promising candidate for future large-scale simulation.

Reference
[1] A. Valassi et al, Developments in Performance and Portability for MadGraph_aMC@NLO
[2] CMS Collaboration, Quantifying the computational speedup with madgraph4gpu for CMS workflow

CONTACT: choi@cern.ch
cms-generator-conveners-mefgal@cern.ch
cms-physics-conveners-gdnic@cern.ch

October 19 - 25, 2024
CHEP 2024

[S. Roiser. CERN IT & HEP Software Foundation. LHC Monte-Carlo Working Group, 14/11/2024]



25/11/24

[J. Choi. CHEP24]

Status 2024 (the leading-year)

- Acquired strong experience in the development on GPU & SIMD architecture.
- Eager to test on latest architecture and chips:
 - NVIDIA Hopper, Blackwell, Grace-Hopper *superchip*;
 - AMD Instinct MI300X;
 - ARM, RISC-V.

- **Floating-point precision is a challenge:**

→ we need **double precision** for Feynman diagrams;

→ regarding future GPUs (like Blackwell for AI)...you sure get more FP64, but you pay the price of **many** FP4 tensor cores.

Architecture	NVIDIA Blackwell	NVIDIA Hopper	NVIDIA Ampere	NVIDIA Volta
Year	2024	2022	2020	2017
GPU	GB200	H100	A100	V100
FP64	90	34	9.7	7.8
FP32	180	67	19.5	15.7
FP16	N/A	134	78	N/A
FP64	90	67	19.5	N/A
TF32	5 000	989	312	N/A
FP16	10 000	1979	624	125
FP8	20 000	3958	N/A	N/A
FP4	40 000	N/A	N/A	N/A

Expressed in TFLOPs.

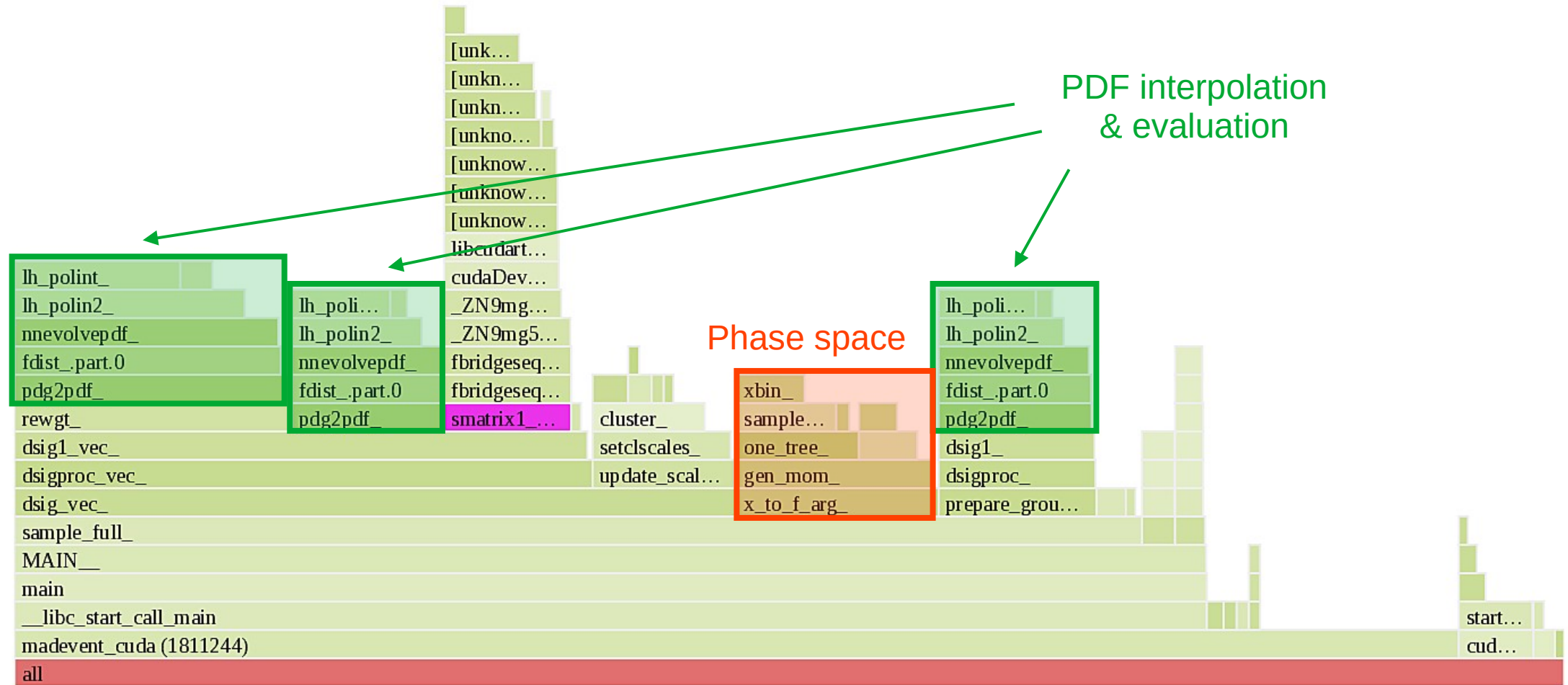
[A. Valassi. CHEP24]

Plans for 2025 (the next-to-leading year)

- 1) Main topic will be to port **Next-to-Leading Order** (NLO) computations to GPU:
 - ◆ How to deal with high-precision floating point calculations?
 - ◆ Double precision to lower precision FLOPs ratio will stay constant (or at worst decrease) in future GPUs.
 - ◆ How to deal with quadruple precision (mainly needed for loop calculation)?
- 2) Enrico Bothmann (one of SHERPA's authors) will join the team in January 2025.
 - ◆ Main contributor to PEPPER: GPU version of SHERPA.
 - ◆ Plan to implement NLO in PEPPER.

[S. Roiser. *CERN IT & HEP Software Foundation*. LHC Monte-Carlo Working Group, 14/11/2024]

Plans for 2025 (the next-to-leading year)



Plans for 2025 (the next-to-leading year)

- 1) Main topic will be to port **Next-to-Leading Order** (NLO) computations to GPU:
 - ◆ How to deal with high-precision floating point calculations?
 - ◆ Double precision to lower precision FLOPs ratio will stay constant (or at worst decrease) in future GPUs.
 - ◆ How to deal with quadruple precision (mainly needed for loop calculation)?
- 2) Enrico Bothmann (one of SHERPA's authors) will join the team in January 2025.
 - ◆ Main contributor to PEPPER: GPU version of SHERPA.
 - ◆ Plan to implement NLO in PEPPER.
- 3) Make more parts of the MadGraph workflow suitable for hardware accelerated execution (phase space, PDF interpolation & evaluation).

[S. Roiser. *CERN IT & HEP Software Foundation*. LHC Monte-Carlo Working Group, 14/11/2024]

Next-to-next plans

- Investigation on machine learning techniques for phase-space sampling optimization:
 - MadNIS method in the MadGraph event generator. [T. Heimes et al. SciPost Phys. 15, 141 (2023)]
[T. Heimes et al. SciPost Phys. 17, 023 (2024)]
- Collaborate with experiments on deployment of accelerated codes.
- Investigate more areas of software engineering in Monte-Carlo event generation:
 - interest in going beyond hard scattering, e.g. shower calculations;
 - investigation on negative weights.

[S. Roiser. *CERN IT & HEP Software Foundation*. LHC Monte-Carlo Working Group, 14/11/2024]



NextGen
Next Generation Triggers