

IWOCL 2024



The 12th International Workshop on OpenCL and SYCL

Improving Performance Portability of the Procedurally Generated High Energy Physics Event Generator MadGraph Using SYCL.

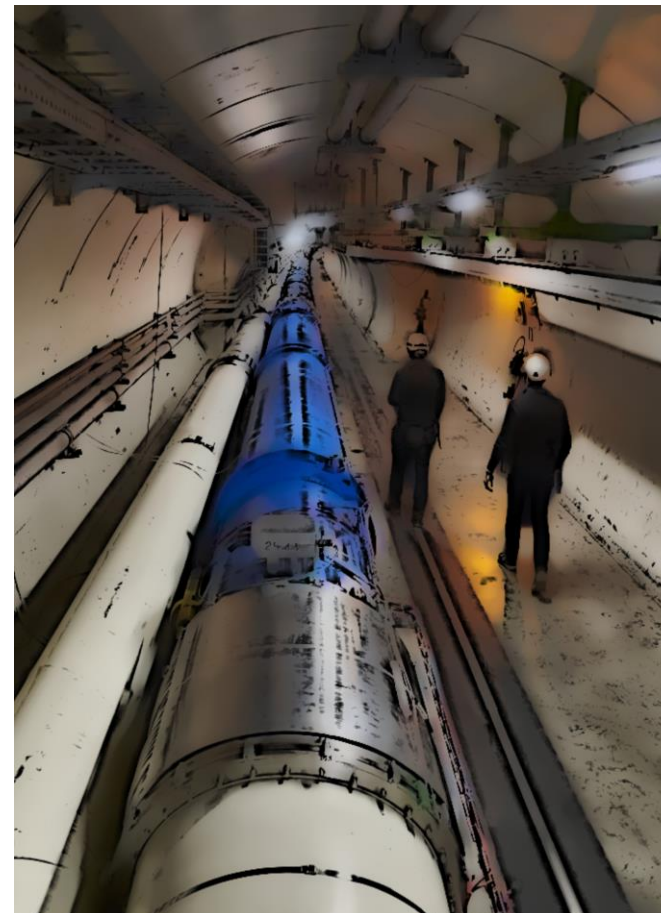
Nathan Nichols, Argonne National Laboratory

J. Taylor Childers and Tyler James Burch, Argonne National Laboratory, and Laurence Field, CERN.

APRIL 8-11, 2024 | CHICAGO, USA | IWOCL.ORG

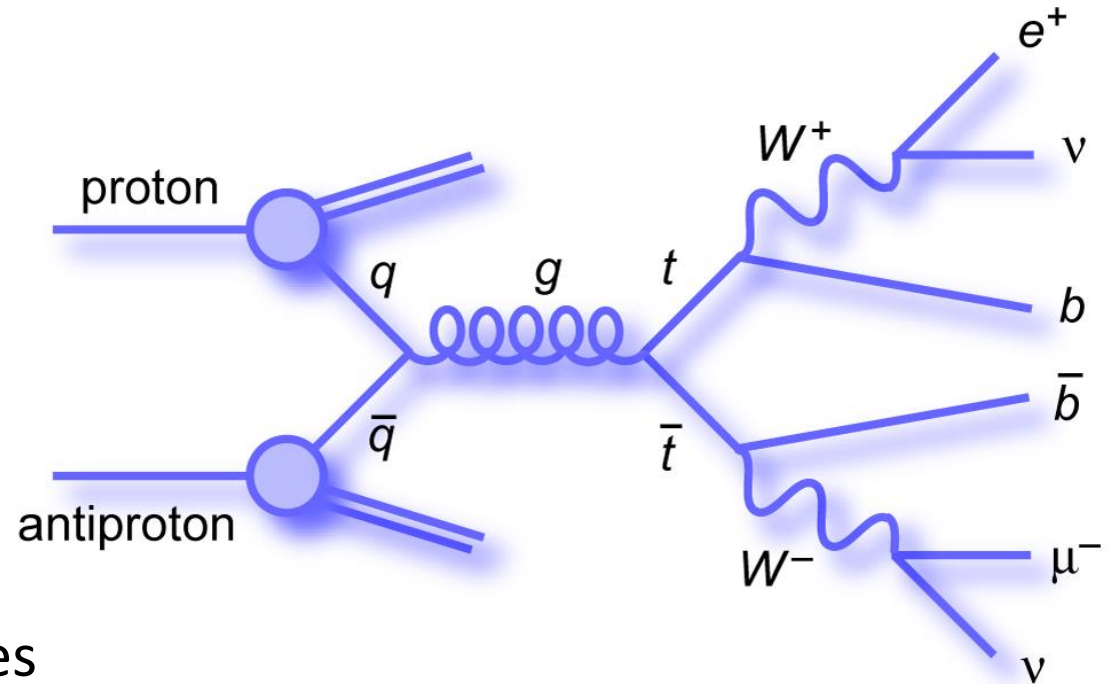
Introduction – Portable Solutions in High Energy Physics

- Introduction
 - What is MadGraph?
 - High level overview
 - A look inside the main kernel
- Performance and Scaling
 - Single Node
 - SYCL vector types
 - HPC machine runs

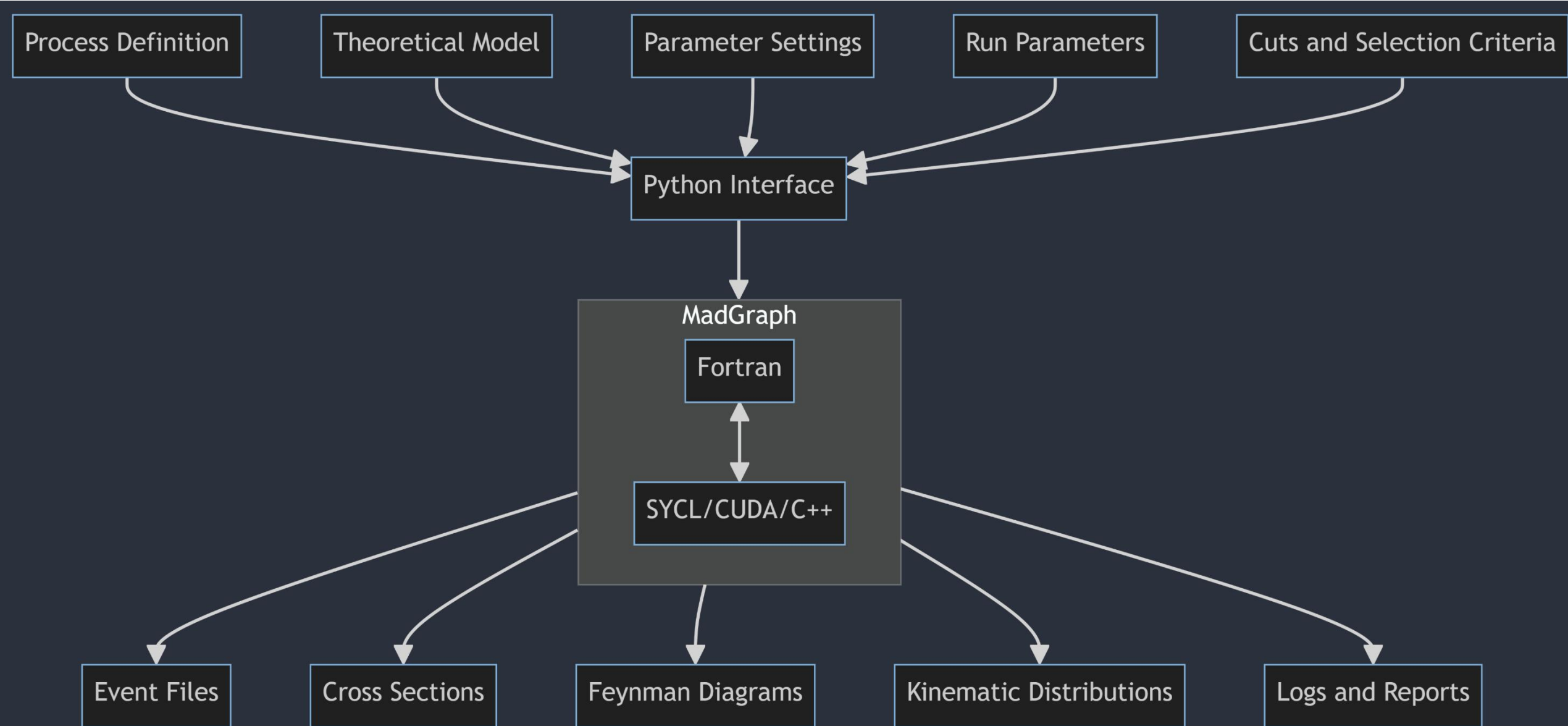


MadGraph – Advancing HEP with SYCL

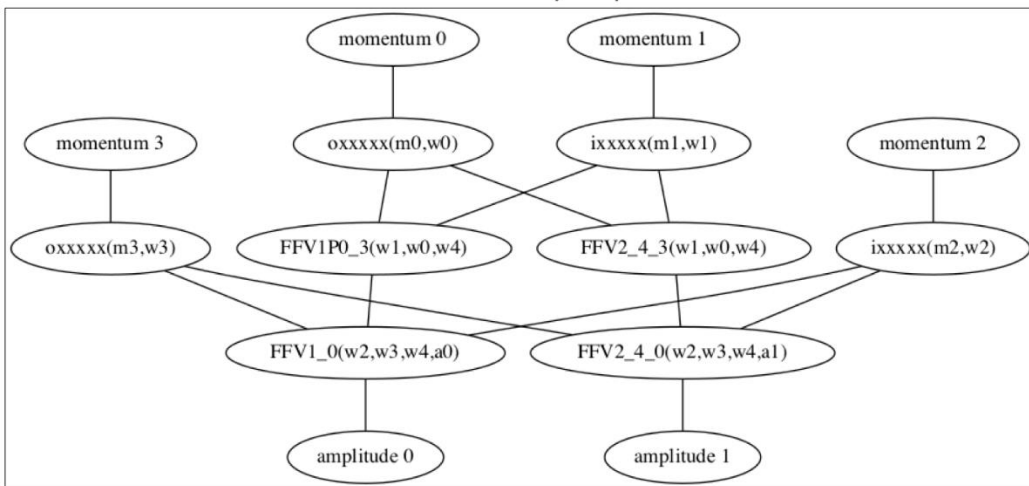
- What is MadGraph?
- Role in High Energy Physics
 - Generating Feynman Diagrams
 - Calculating Cross-Sections
 - Event Generation
- Integral for Research
 - Facilitates High Energy Physics Studies
 - Using SYCL enables Cross-Platform Portability



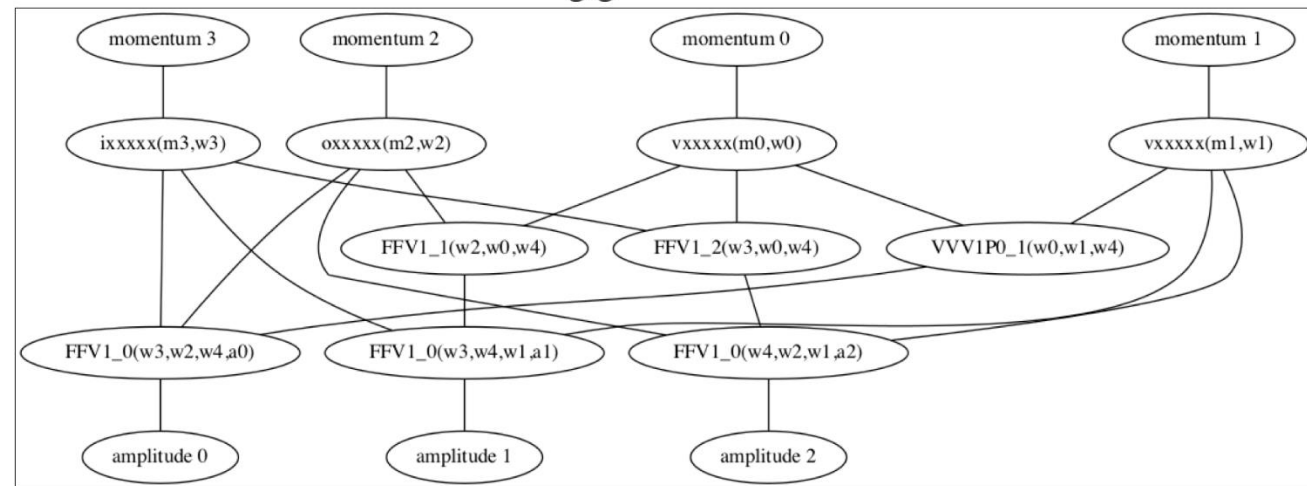
The Ins and Outs of MadGraph



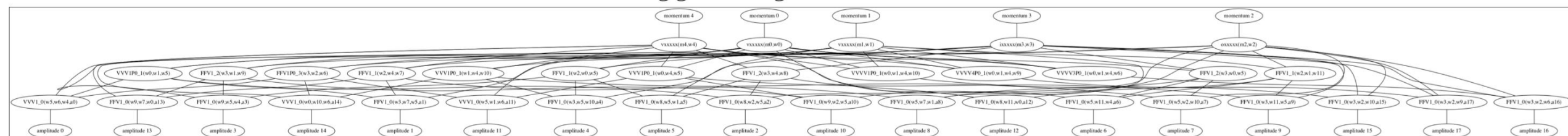
$$e^+e^- \rightarrow \mu^+\mu^-$$



$$gg \rightarrow t\bar{t}$$

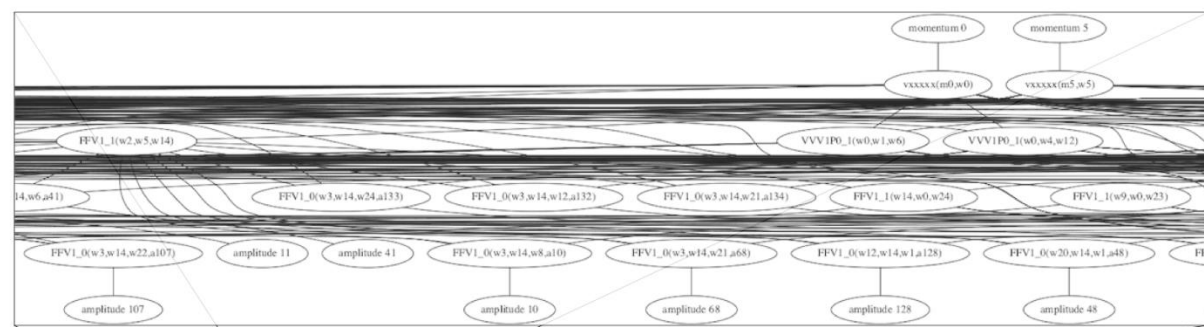


$$gg \rightarrow t\bar{t}g$$



Process	Generated Function Calls
$e^+e^- \rightarrow \mu^+\mu^-$	6
$gg \rightarrow t\bar{t}$	12
$gg \rightarrow t\bar{t}g$	36
$gg \rightarrow t\bar{t}gg$	222
$gg \rightarrow t\bar{t}ggg$	2274

$$gg \rightarrow t\bar{t}gg$$



Experimental Setup – Hardware + Software Environment

- Hardware Specifications

- JLSE Testbed
 - Nvidia A100
 - Nvidia V100
 - AMD MI50
 - AMD MI100
 - AMD MI250
 - Skylake 8180M
- ALCF Clusters
 - Aurora*
 - Sunspot*
 - Polaris

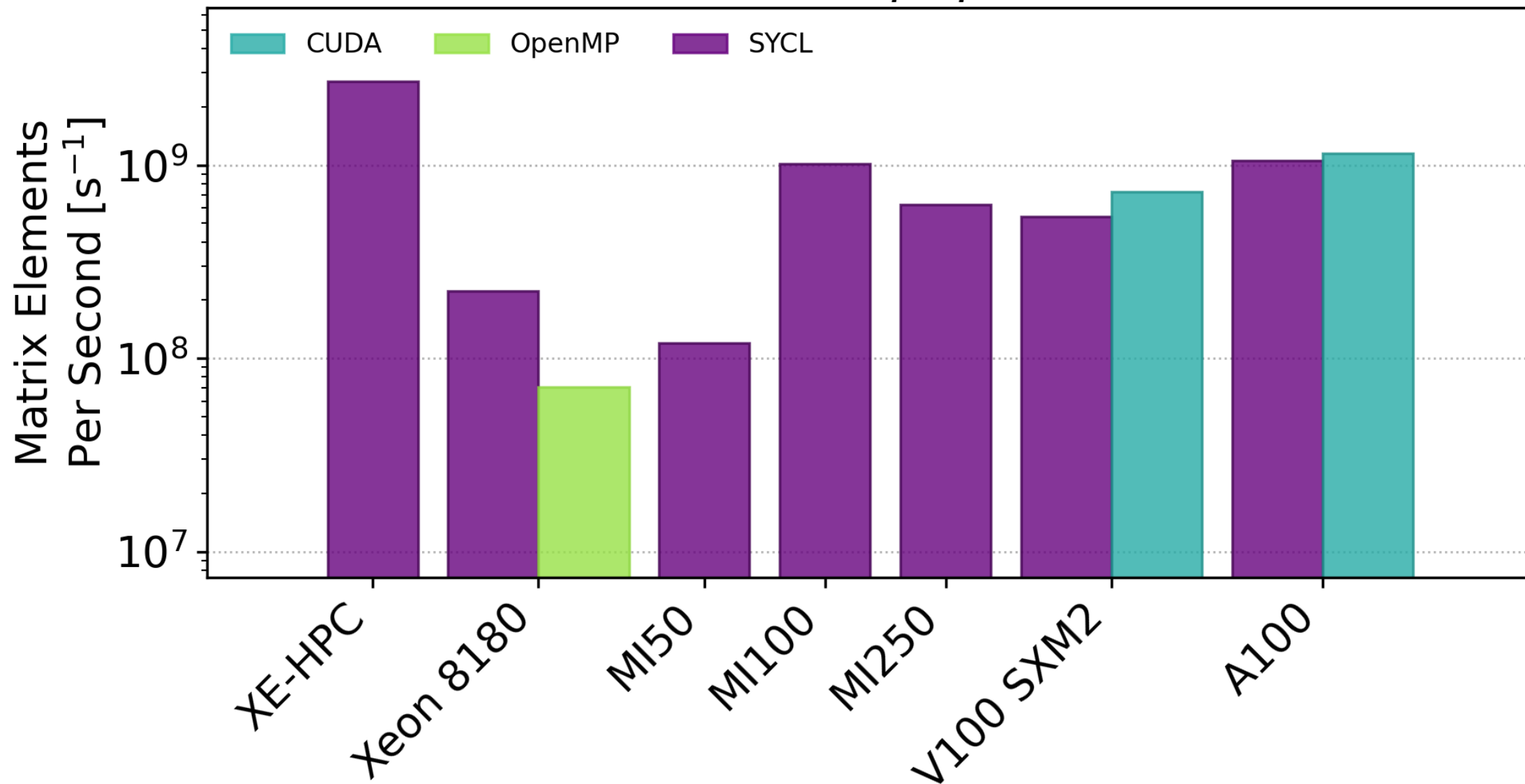
- Software Tools

- oneAPI DPC++
- GCC 11.3
- CUDA 11.4 and 11.8
- ROCm 4.5.0 and ROCm 4.5.3

*Intel provided access to early experimental versions of oneAPI DPC++ to target the Intel GPUs

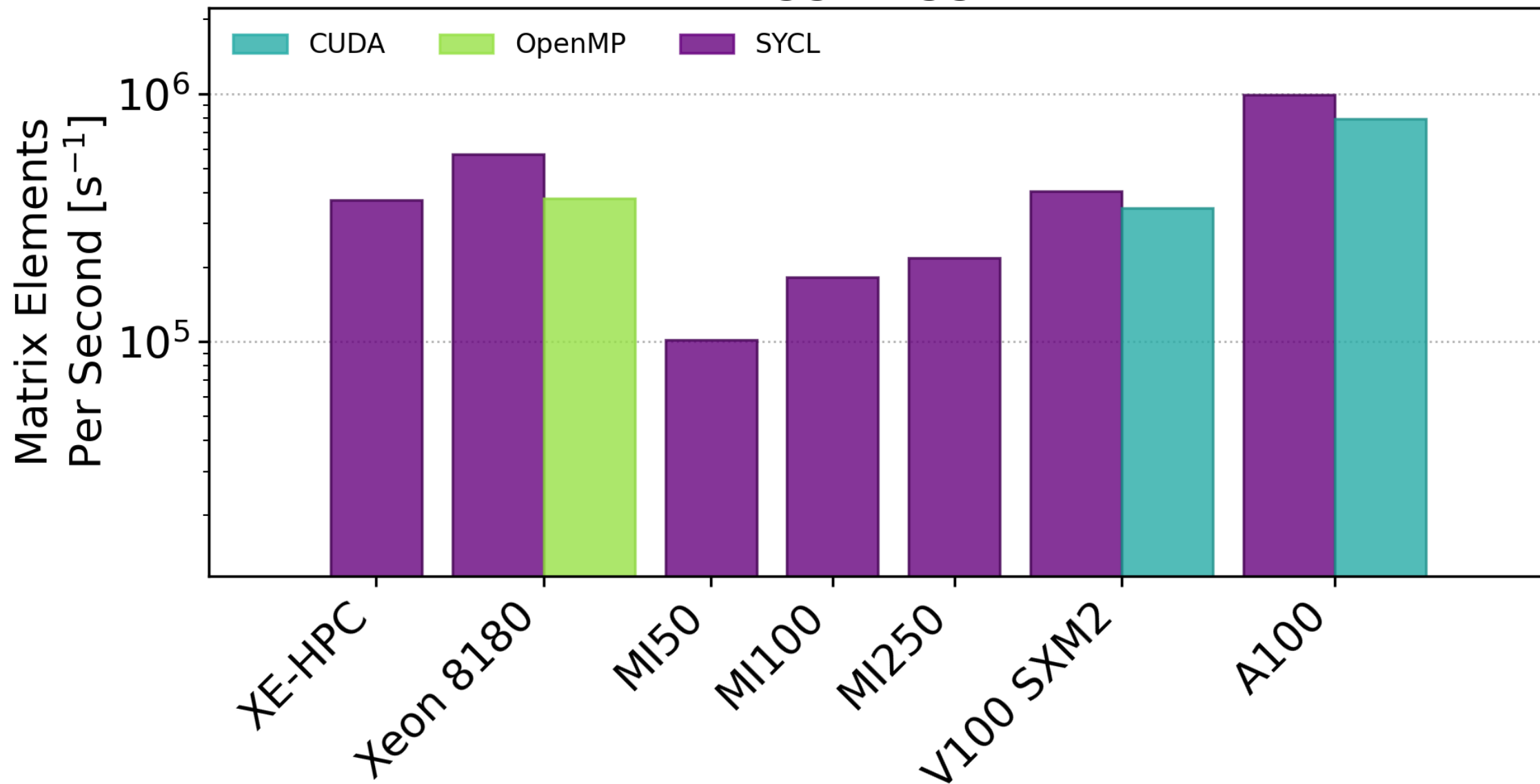
Results – Performance Comparison

$$e^+e^- \rightarrow \mu^+\mu^-$$



Results – Performance Comparison

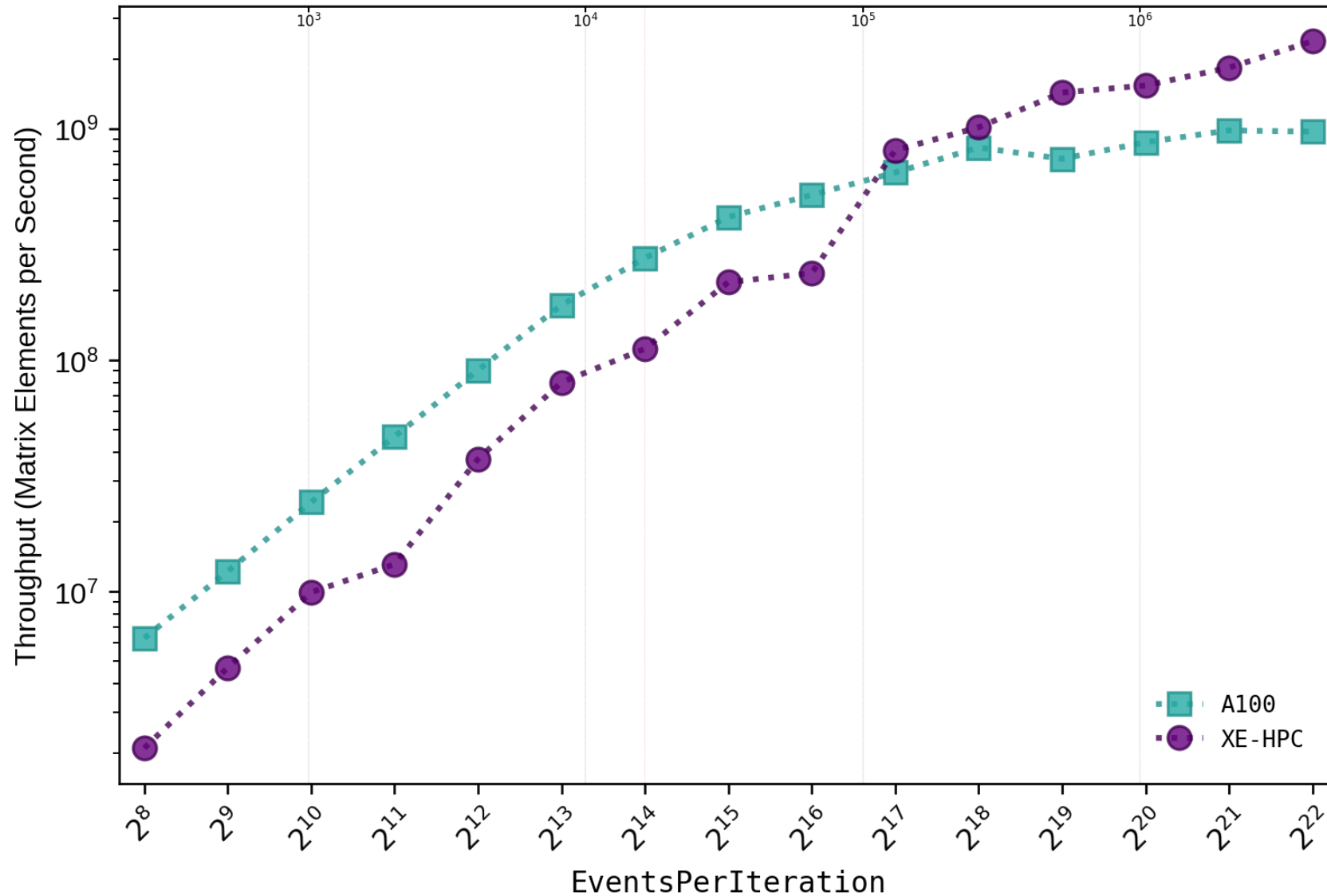
$gg \rightarrow t\bar{t}gg$



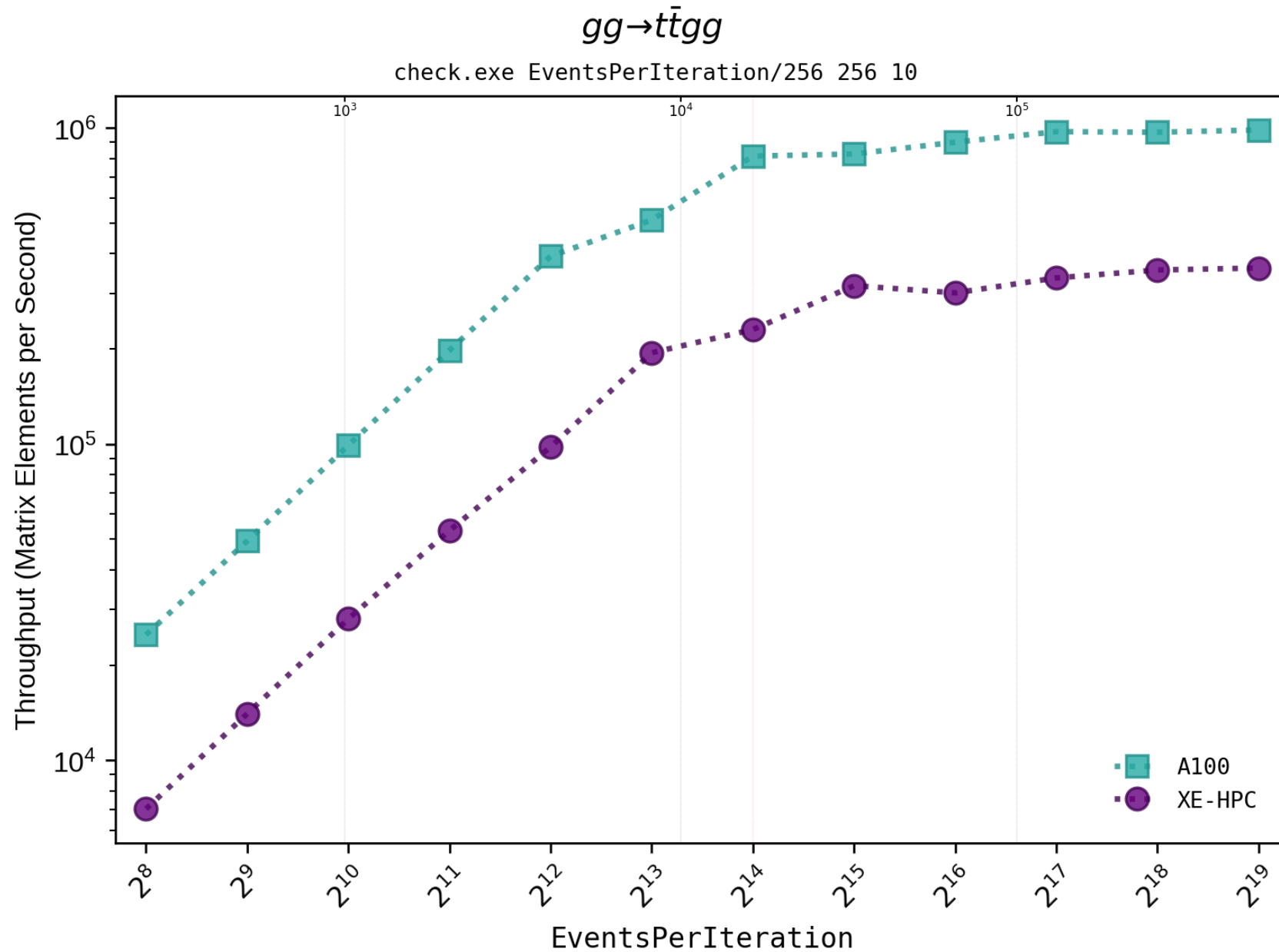
Results – Parameter Scaling on Single-Node Devices

$$e^+e^- \rightarrow \mu^+\mu^-$$

check.exe EventsPerIteration/256 256 10



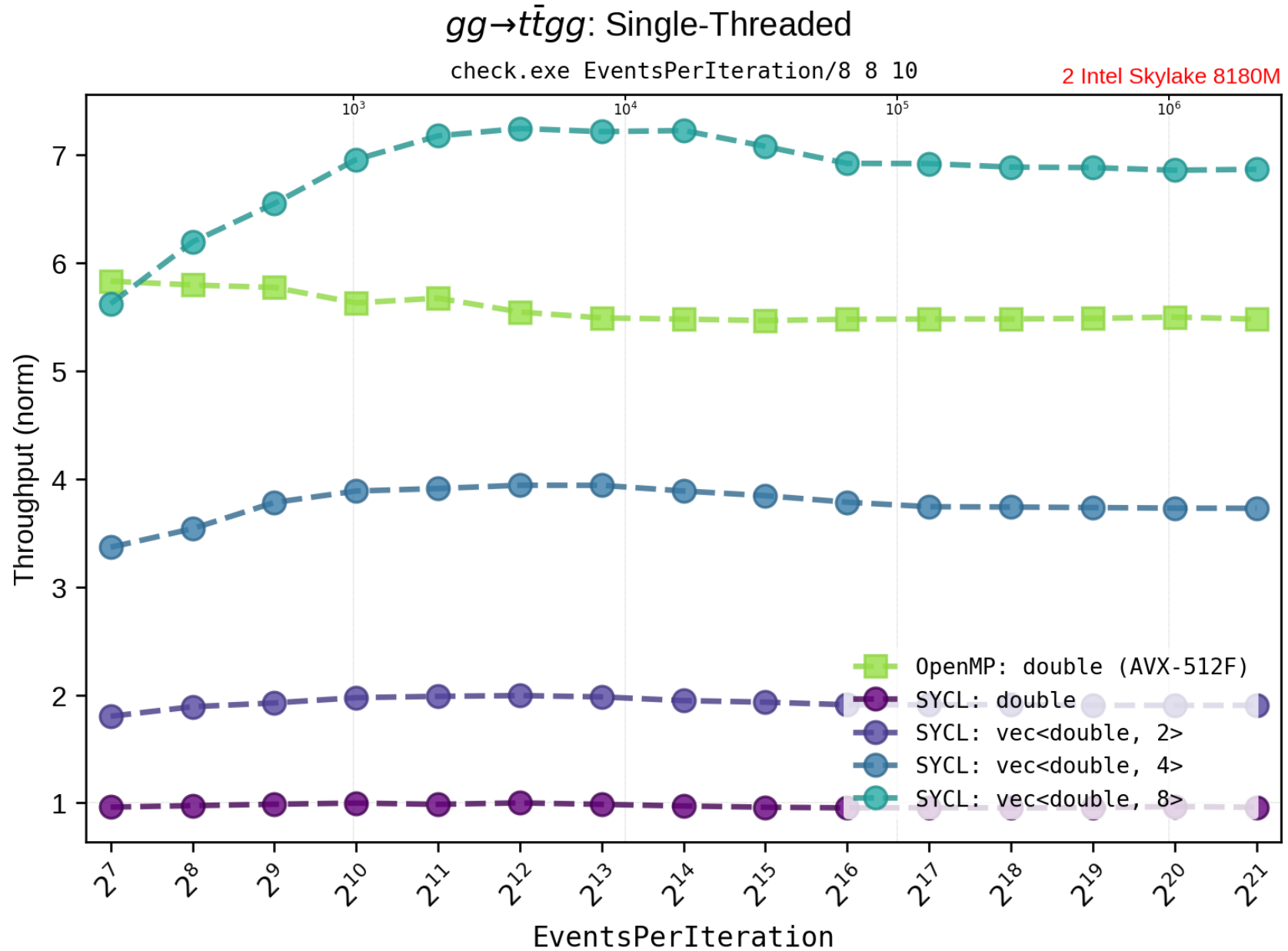
Results – Parameter Scaling on Single-Node Devices



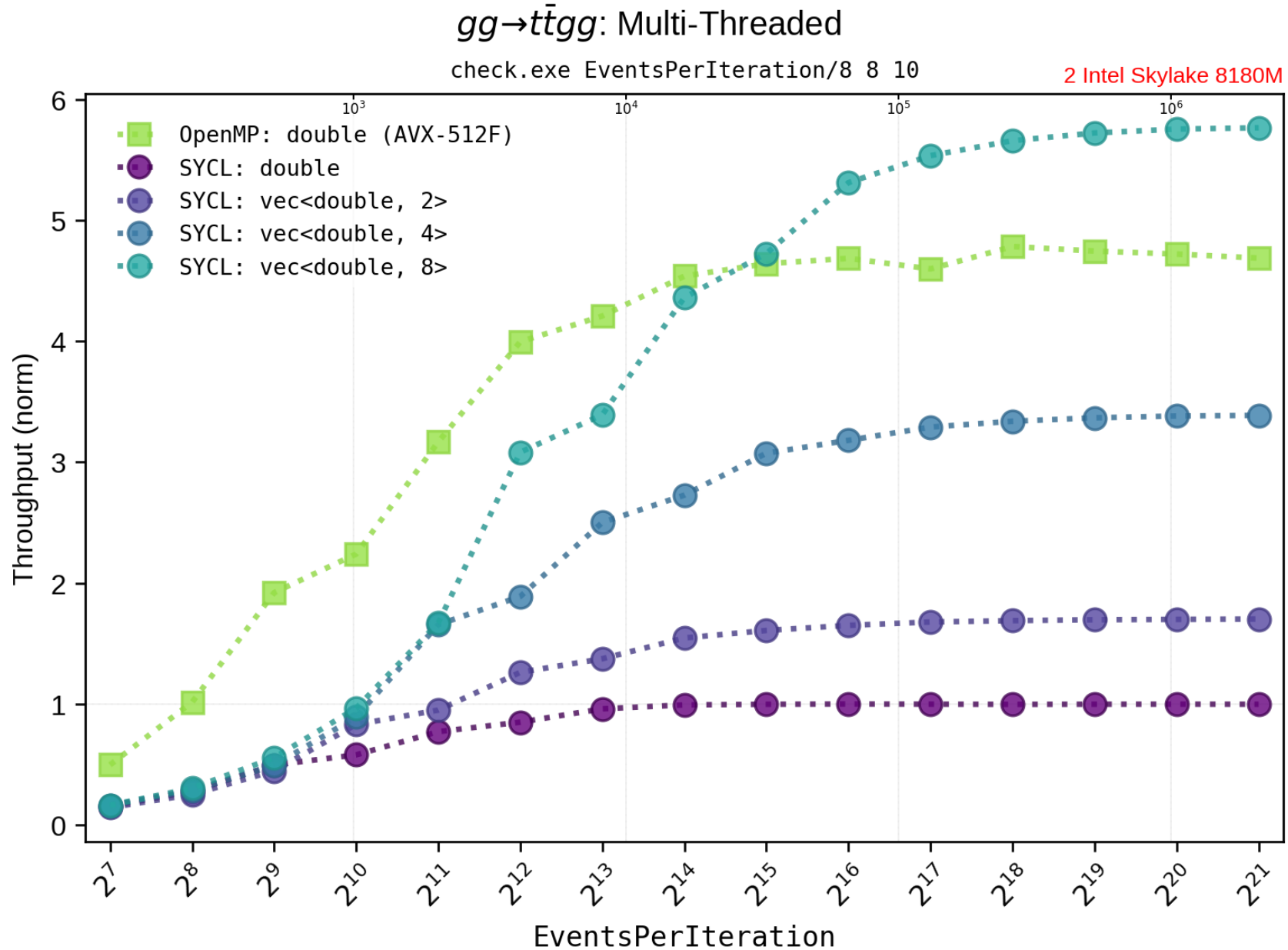
Performance on the CPU?

```
typedef sycl::vec<fptype, MGONGPU_VEC_DIM> fptype_sv;  
typedef sycl::vec<long, MGONGPU_VEC_DIM> int_sv;  
typedef sycl::vec<long, MGONGPU_VEC_DIM> bool_sv;  
#if defined MGONGPU_COMPLEX_CXSAMPL  
    typedef mgOnGpu::cxsmpl<fptype_sv> cxttype_sv;  
#elif defined MGONGPU_COMPLEX_EXTRAS  
    typedef extras::complex<fptype_sv> cxttype_sv;  
#elif defined MGONGPU_COMPLEX_STD  
    typedef std::complex<fptype_sv> cxttype_sv;  
#elif MGONGPU_COMPLEX_ONEAPI  
    typedef sycl::ext::oneapi::experimental::complex<fptype_sv> cxttype_sv;  
#elif MGONGPU_COMPLEX_CUTHRUST  
    typedef thrust::complex<fptype_sv> cxttype_sv;  
#elif MGONGPU_COMPLEX_SYCLCPLX  
    typedef sycl::ext::cplx::complex<fptype_sv> cxttype_sv;  
#else  
    #error Unconfigured vector complex type. Add details to `mgOnGpuVectors.h` or set MGONGPU_VEC_DIM to 1 in `mgOnGpuConfig.h`.  
#endif
```

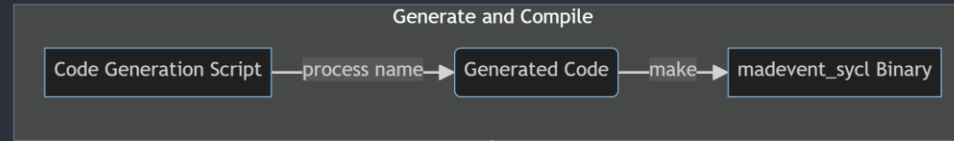
Results – SYCL Vector Type vs. Vector Intrinsic



Results – SYCL Vector Type vs. Vector Intrinsics



MadGraph Workflow – A Simplified Diagram



MadGraph Scaling Scripts

Scheduler

initialize jobs

MPI4py

copy binary and inputs

Node 0

copy binary and inputs

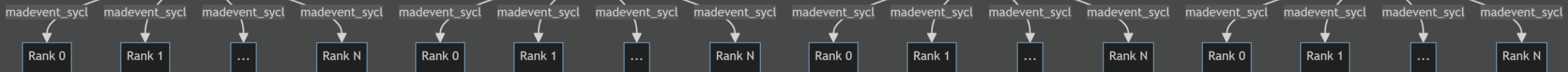
Node 1

copy binary and inputs

...

copy binary and inputs

Node M

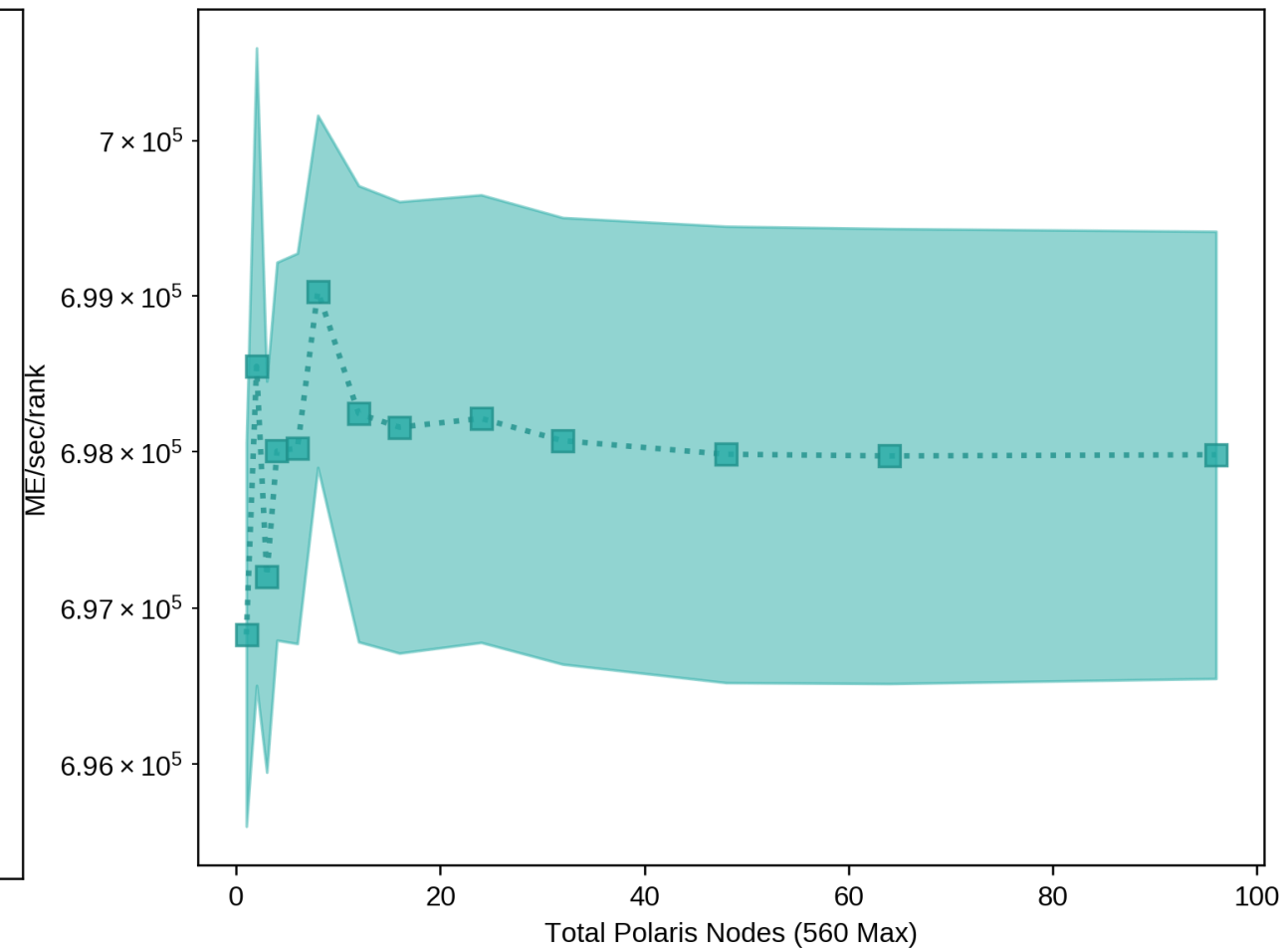
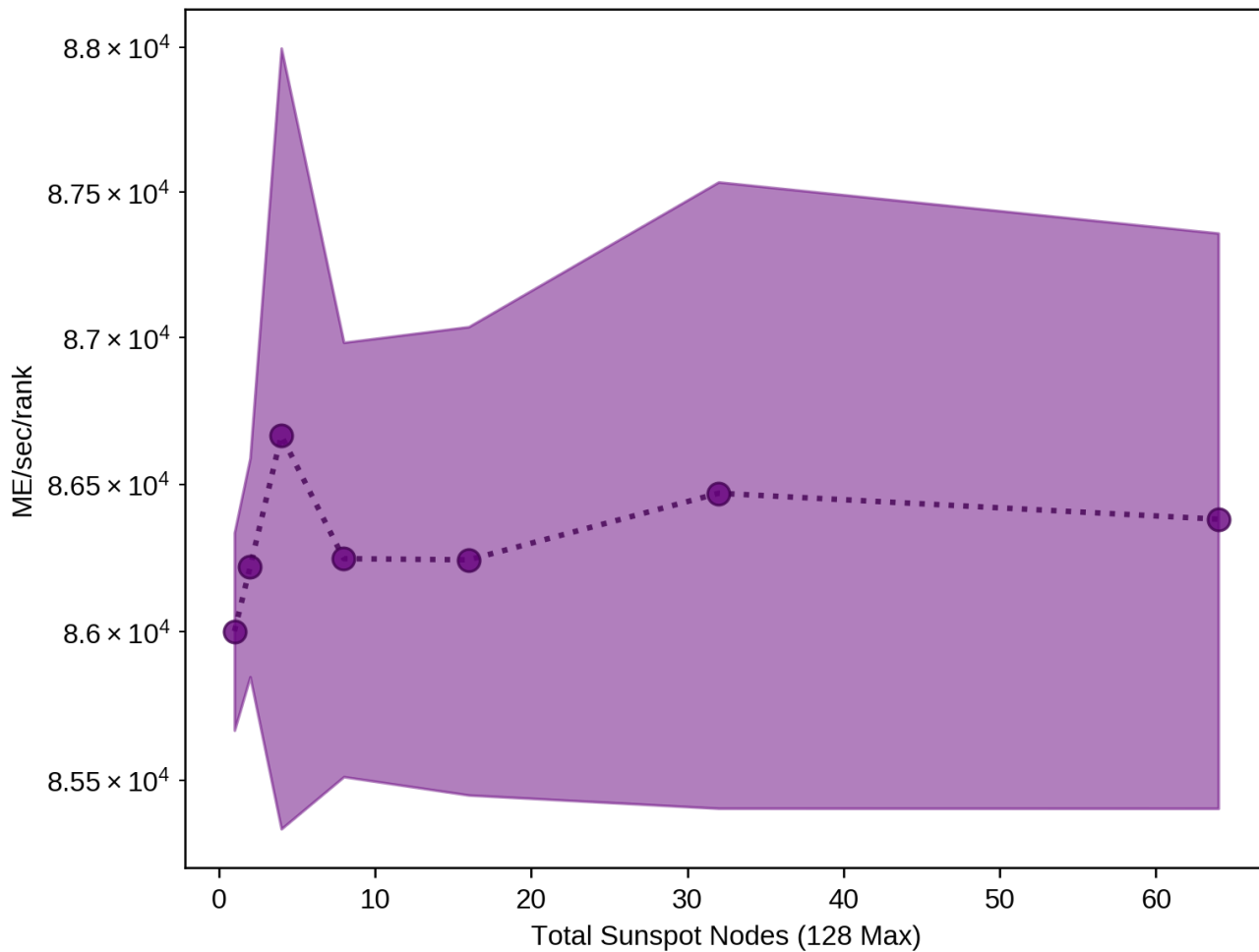


MPI4py

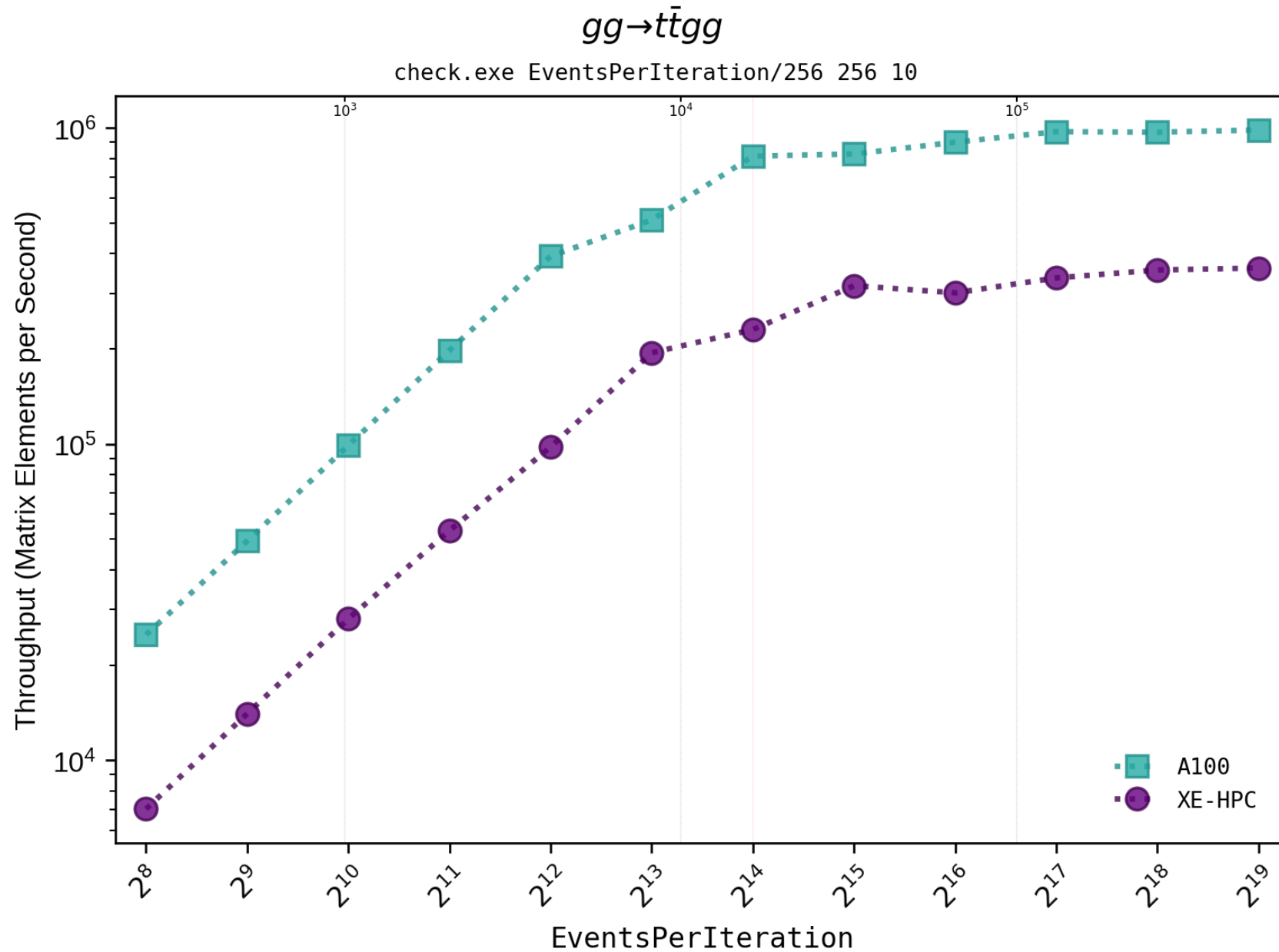
gather results

Output

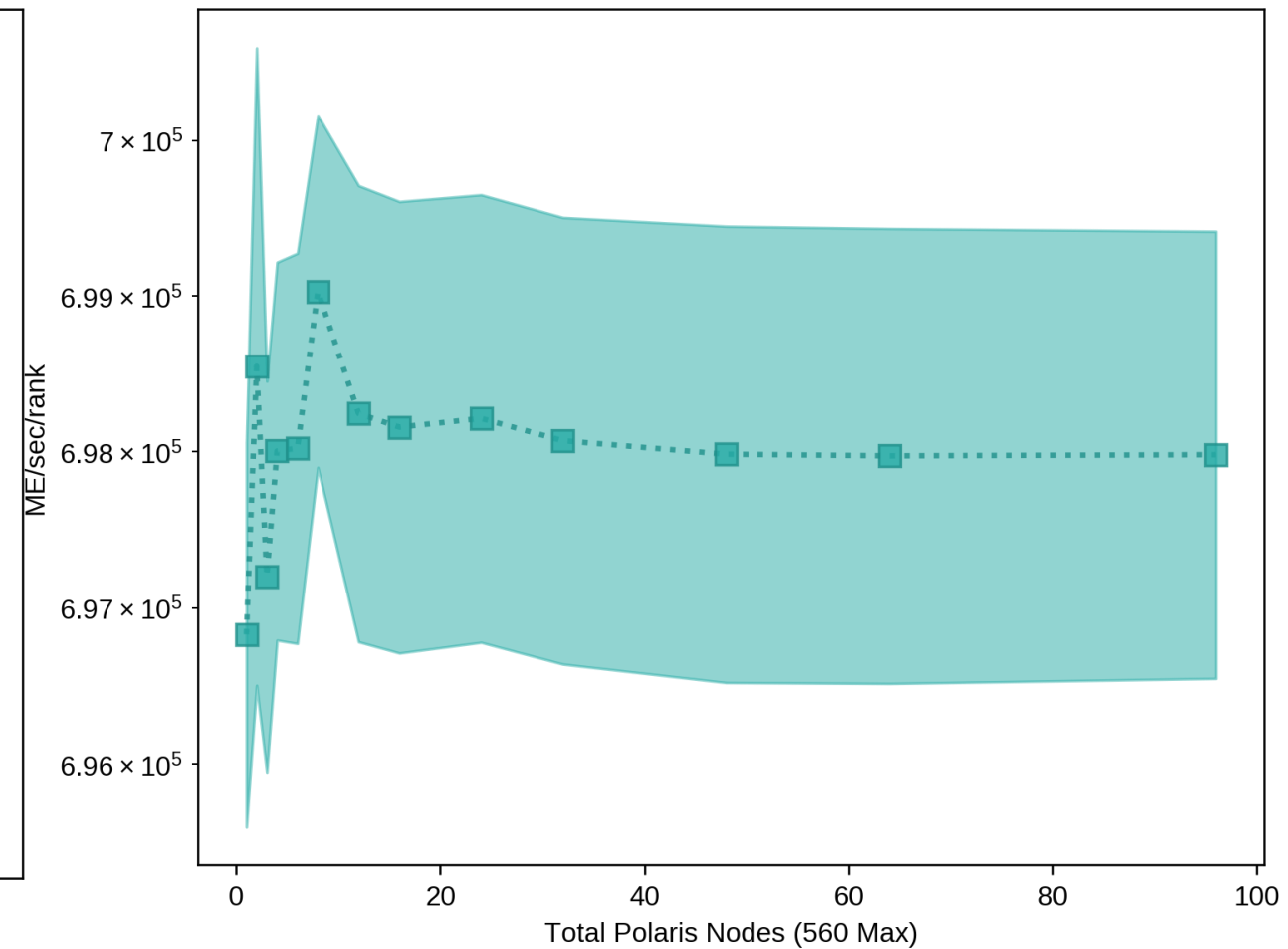
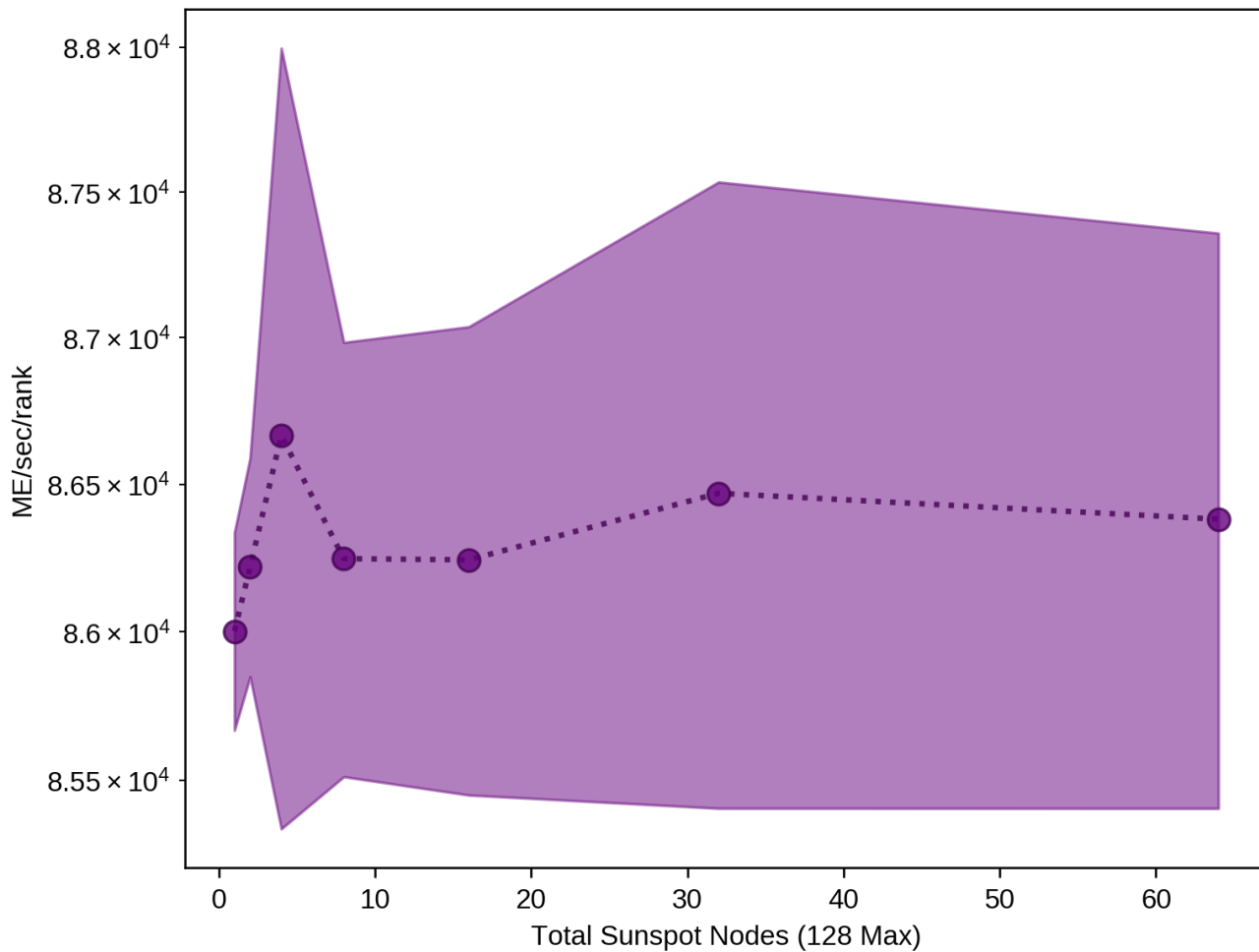
Results – Kernel Performance at Small Scale



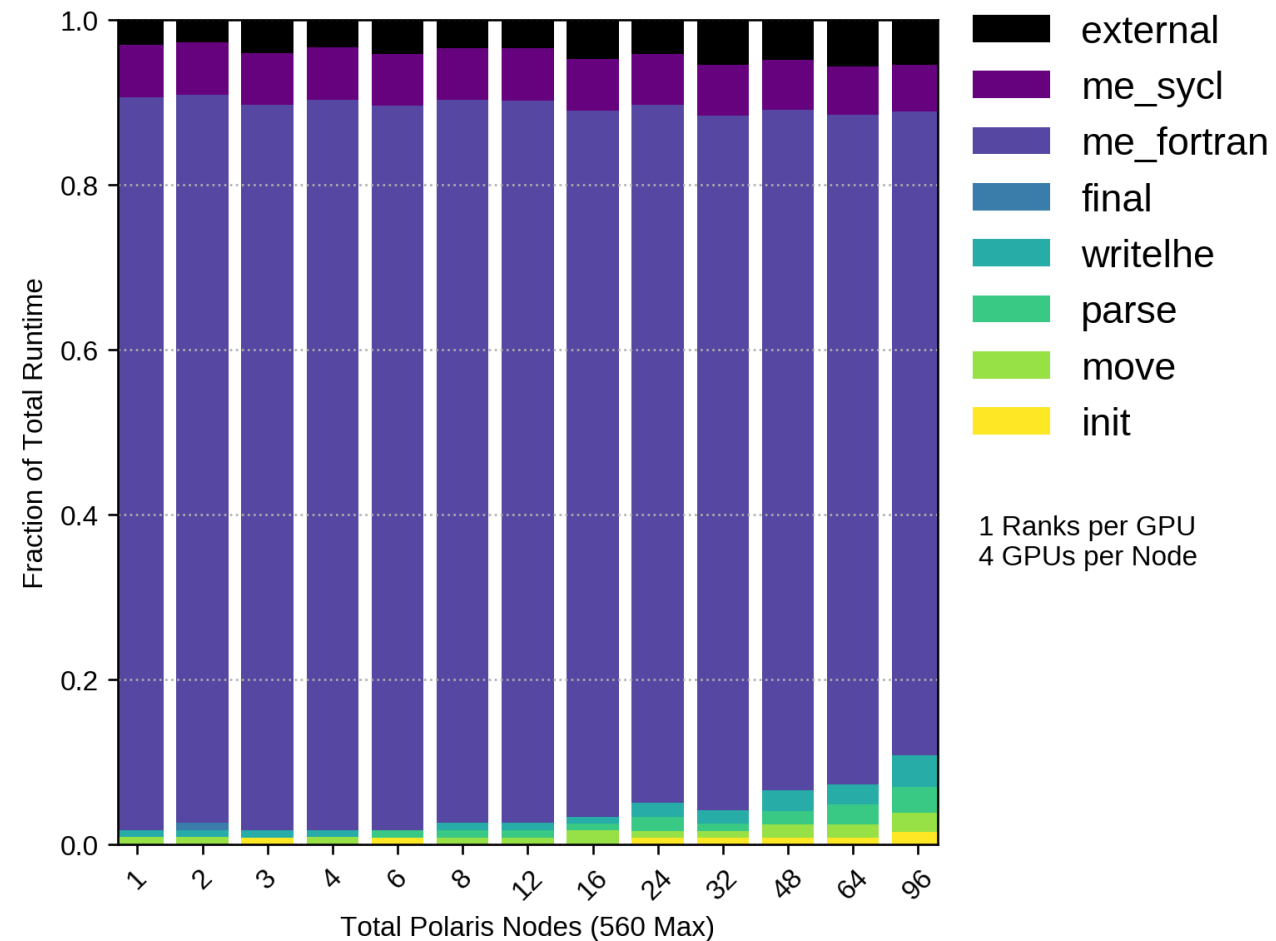
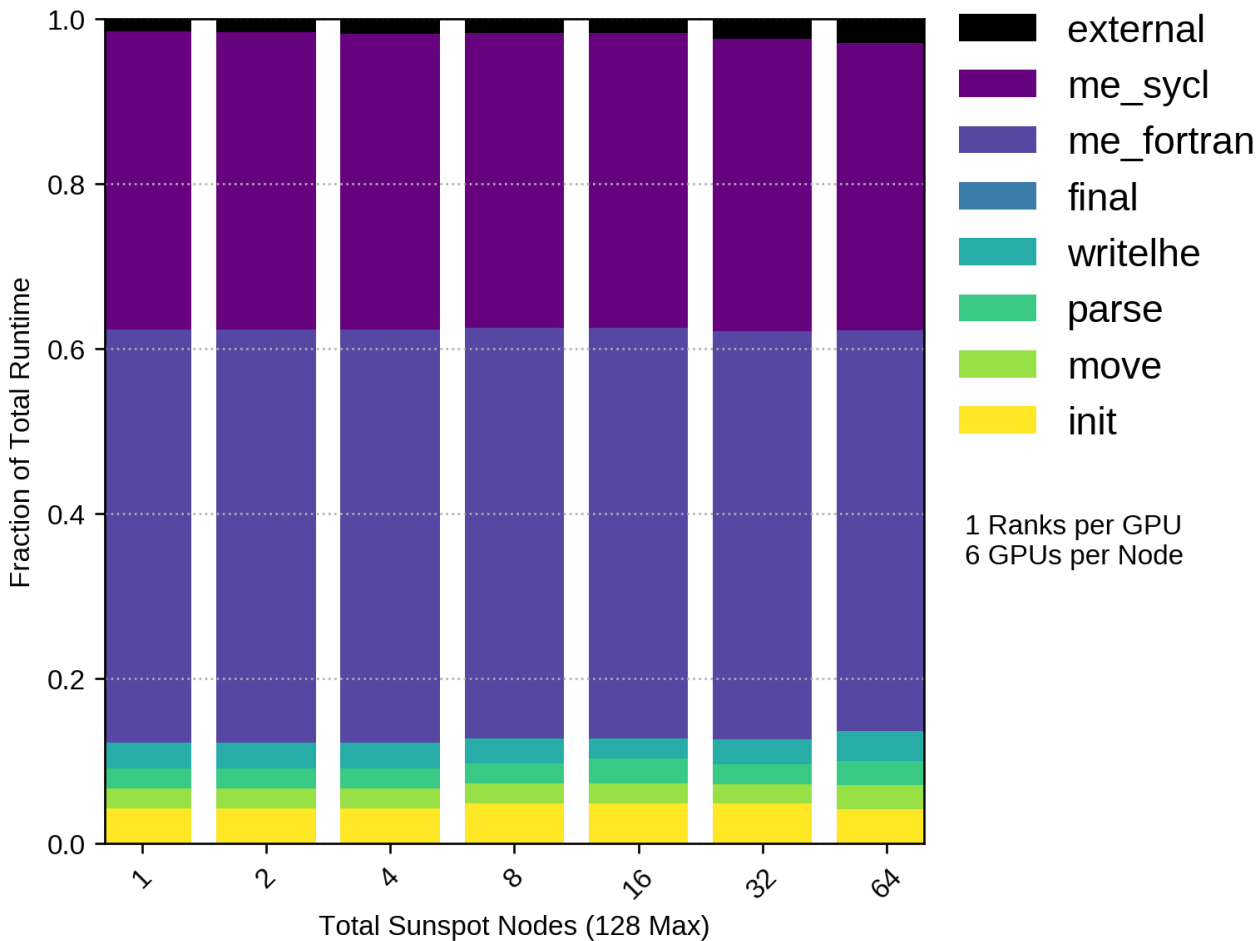
Results – Parameter Scaling on Single-Node Devices



Results – Kernel Performance at Small Scale

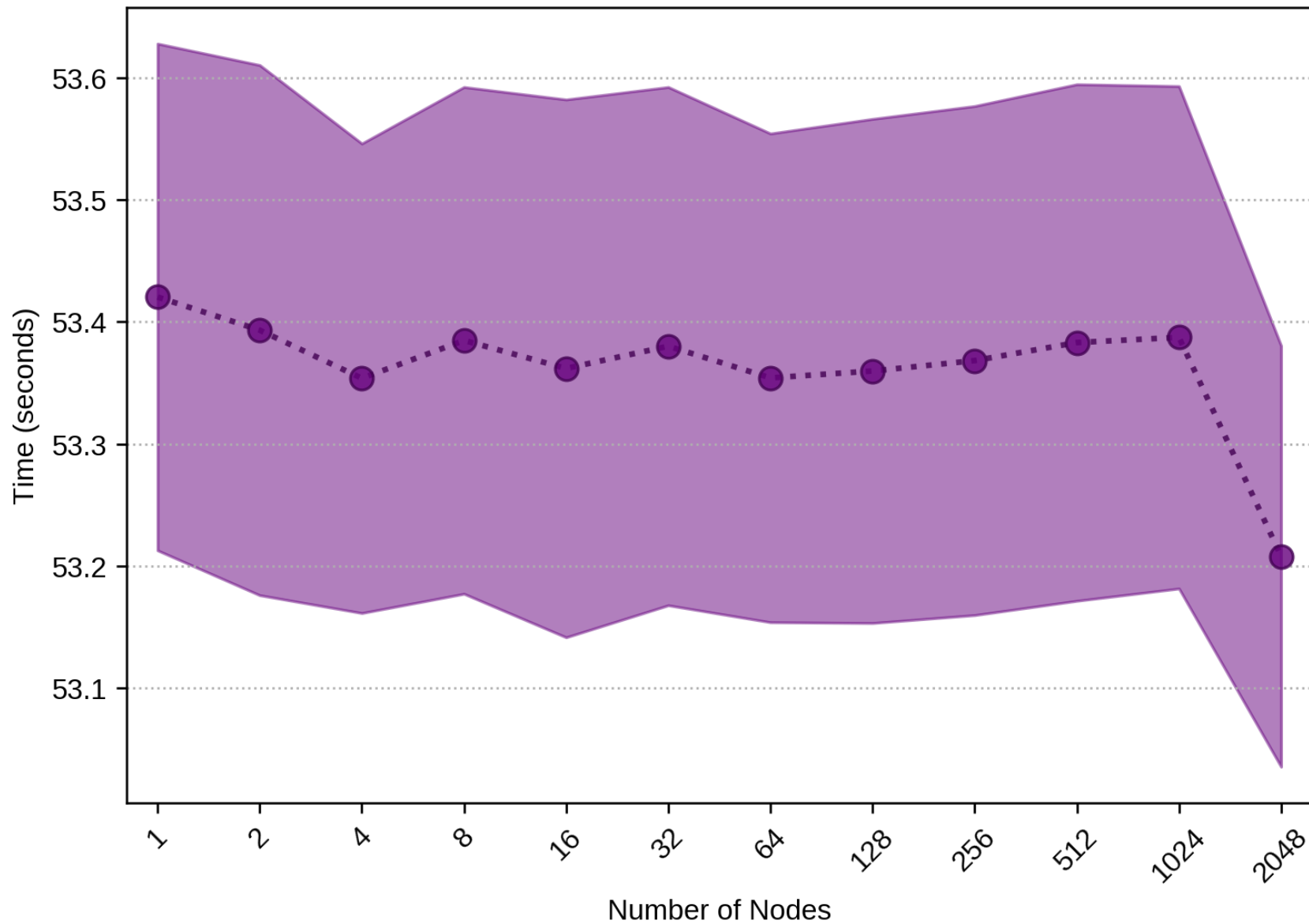


Results – Timing Components on Different HPC Systems



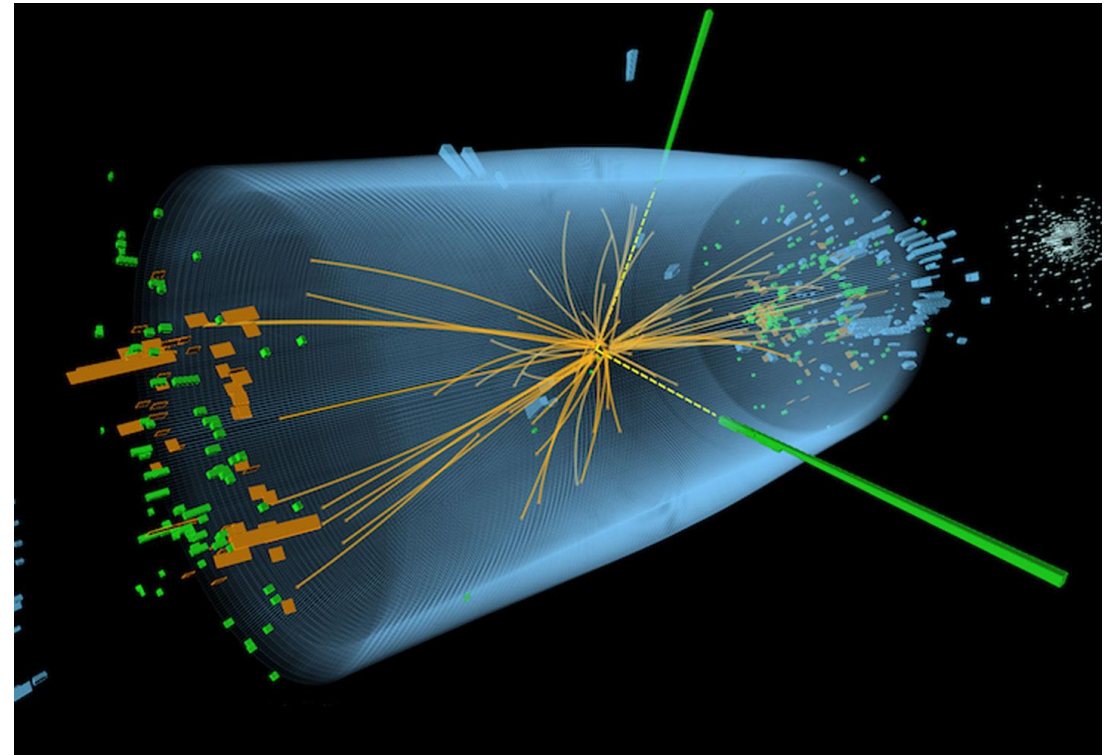
Time Spent in SYCL per rank on Aurora

(12 ranks per node)



Summary of Findings

- Portable Solution for Event Generation
- SYCL Performance
 - Similar or better than native implementations
- Parameter Scaling
 - Fine tune for better performance
- SYCL Vector Types
 - A vectorization route for complex codebases
- Scaling on Different Systems
 - Code scales efficiently



Acknowledgments

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

We gratefully acknowledge the computing resources provided and operated by the Joint Laboratory for System Evaluation (JLSE) at Argonne National Laboratory.

We thank Thomas Applencourt (ALCF), Colleen Bertoni (ALCF), and Brian Holland (Intel COE) for their support while developing and testing this software and members of the MadGraph GPU development team.

Questions?

Hardware Specifications – JLSE (Supplemental)

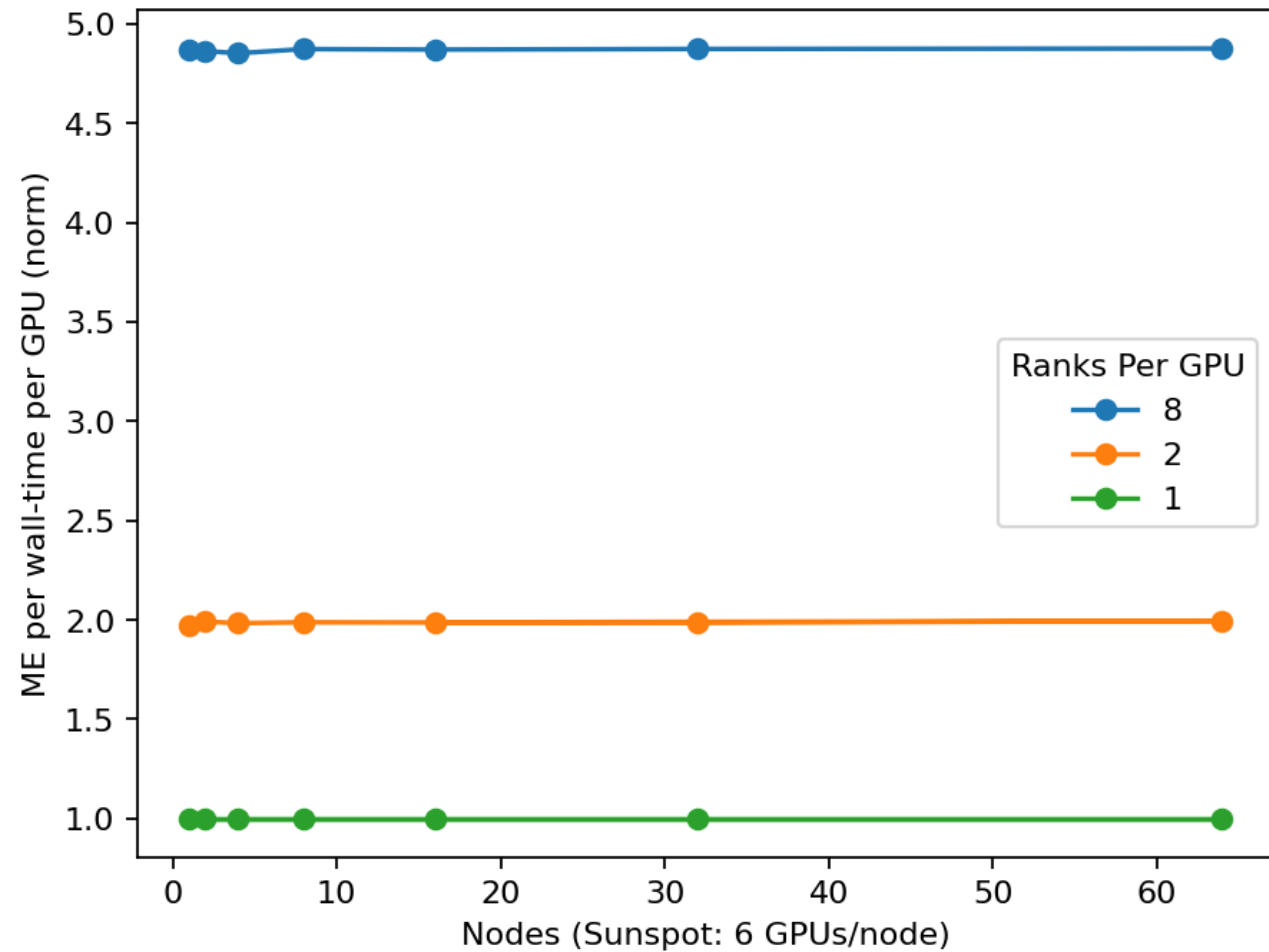
- Nvidia A100
 - Model: Gigabyte G242-Z11
 - CPU: AMD 7532 32c 2.4Ghz
 - RAM: DDR4-3200 256GB
 - GPU: 1x Nvidia A100 40GB PCIe 4.0
 - Networking: Mellanox ConnectX-6 EDR
 - Storage: Intel P4510 2TB NVMe
- Nvidia V100
 - Model: SuperMicro SuperServer 1029GQ-TVRT
 - CPU: 2x Intel Xeon Gold 6152 22c 2.10GHz
 - RAM: 192GB DDR4-2666
 - GPU: 4x Nvidia Tesla V100 SXM2 w/32GB HBM2
 - Networking: Mellanox ConnectX-5 EDR
- Skylake
 - Model: Intel S2600WF
 - CPU: 12 - 2x Intel Xeon Platinum 8180M CPU @ 2.50GHz
 - RAM: 768GB
 - Networking: EDR IB
 - Storage: Intel P4500 1TB NVMe SSD, Intel P4800X 375GB NVMe Optane SSD
- AMD MI50
 - Model: Gigabyte G482-Z51
 - CPU: 2x AMD EPYC 7742 64c Rome
 - GPU: 4x AMD MI50 32GB GPUs
 - RAM: 256GB DDR-3200 RAM
 - Networking: Infinity Fabric
- AMD MI100
 - CPU: 2x AMD EPYC 7543 32c (Milan)
 - GPU: 4x AMD MI100 32GB GPUs
 - RAM: 512GB DDR4-3200
 - Networking: Infinity Fabric
- AMD MI250
 - Model: Supermicro AS-4124GQ-TNMI
 - CPU: 2x AMD EPYC 7713 64c (Milan)
 - GPU: 4x AMD Instinct MI250 64GB HBM2e PCIe Gen4
 - RAM: 512GB DDR4-3200
 - Networking: EDR InfiniBand

ALCF System Details (Supplemental)

- Polaris
 - CPU: AMD EPYC Milan 7543P 32 core
 - RAM: 512 GB DDR4
 - GPUs: 4 Nvidia A100 with NVLink
 - Storage: 1.6TB SSD RAID0 per node
 - Network: Slingshot 10 (upgrading to Slingshot 11 in 2023)
 - Total Nodes: 560
- Aurora (Brief Overview)
 - CPUs: 2 Intel Xeon CPU Max Series processors
 - RAM: 64GB HBM on each, 512GB DDR5 each
 - GPUs: 6 Intel Data Center GPU Max Series
 - Network: 8 Slingshot 11 fabric endpoints
 - Total Nodes: 10624
- Sunspot (Testbed System Identical to Aurora)
 - Total Nodes: 128

Multiple Ranks to Fill GPUs

$gg \rightarrow t\bar{t}gg$



$gg \rightarrow t\bar{t}gg$

