# HiRep on GPUs

Sofie Martins, University of Southern Denmark

Nordic Lattice Meeting 2024, Lund University

# What is HiRep?

**Hi**gher **Rep**resentations of Wilson Fermions in F, ADJ, 2S, 2AS of SU(N) and SO(N) for any number of colors

$$D_m\psi(x) \equiv (D + m_0)\psi(x)$$
$$= \left(\frac{4}{a} + m_0\right)\psi(x)$$
$$-\frac{1}{2a}\sum_{\mu}\left[(1-\gamma_\mu)U^R(x,\mu)\psi(x+\mu)\right.$$
$$\left.+\,(1+\gamma_\mu)U^R(x-\mu,\mu)^\dagger\psi(x-\mu)\right]$$

[Del Debbio et. al., 2010, 0805.2058, Phys. Rev. D.]

**github.com/claudiopica/HiRep**

Currently working on this with **Claudio Pica** and **Antonio Rago**

Previous contributions by Erik Kjellgren and Emiliano Molinaro

# Topics

Implementation

Software Quality

HiRep on LUMI-G

Outlook

# Implementation

# Features

**Hi**gher **Rep**resentations of Wilson Fermions

- Fundamental, Adjoint, 2-index symmetric, 2-index antisymmetric
- SU(N) and SO(N)

Symanzik-Improvement

- Clover-improvement
- exponentiated clover improvement
- Lüscher-Weisz Gauge Action

Efficient Configuration Generation

- Monomials: HMC, RHMC, Hasenbusch Acceleration, Twisted Mass
- A selection of inverters: Conjugate gradient, BiCGstab, QMR$\gamma_5$, …
- Integrators: 2$^{nd}$ and 4$^{th}$ order Omelyan integrators, Leapfrog

Measurements

- Determination of spectrum for connected and disconnected contributions, Wilson flow, multi-level, glueballs, and many other features

# GPU Acceleration

## Higher Representations of Wilson Fermions

- Fundamental, Adjoint, 2-index symmetric, 2-index antisymmetric
- SU(N) and SO(N)

## Symanzik-Improvement

- Clover-improvement
- exponentiated clover improvement
- Lüscher-Weisz Gauge Action

## Efficient Configuration Generation

- Monomials: HMC, RHMC, Hasenbusch Acceleration, Twisted Mass
- A selection of inverters: Conjugate gradient, BiCGstab, QMR$\gamma_5$, …
- Integrators: 2$^{nd}$ and 4$^{th}$ order Omelyan integrators, Leapfrog

## Measurements

- Determination of spectrum for connected and disconnected contributions, Wilson flow, multi-level, glueballs, and many other features

# Using Hirep – Have a look at our new documentation!



HiRep
- Introduction
- Getting Started
- User Guide
  - Input File Configuration
  - Supported Features
  - Analysis
- Developer Guide
- Code Reference

## Input File Configuration

### Integrator

The HiRep code uses a multilevel integrator and each integrator level has to be specified in the input file.

```
integrator {
    level = 0
    type = o2mn
    steps = 10
}
```

| Variable | Description |
|---|---|
| `level` | unique integrator level (level 0 is the outermost level) |
| `type` | integrator type (see below) |
| `steps` | number of integration steps |

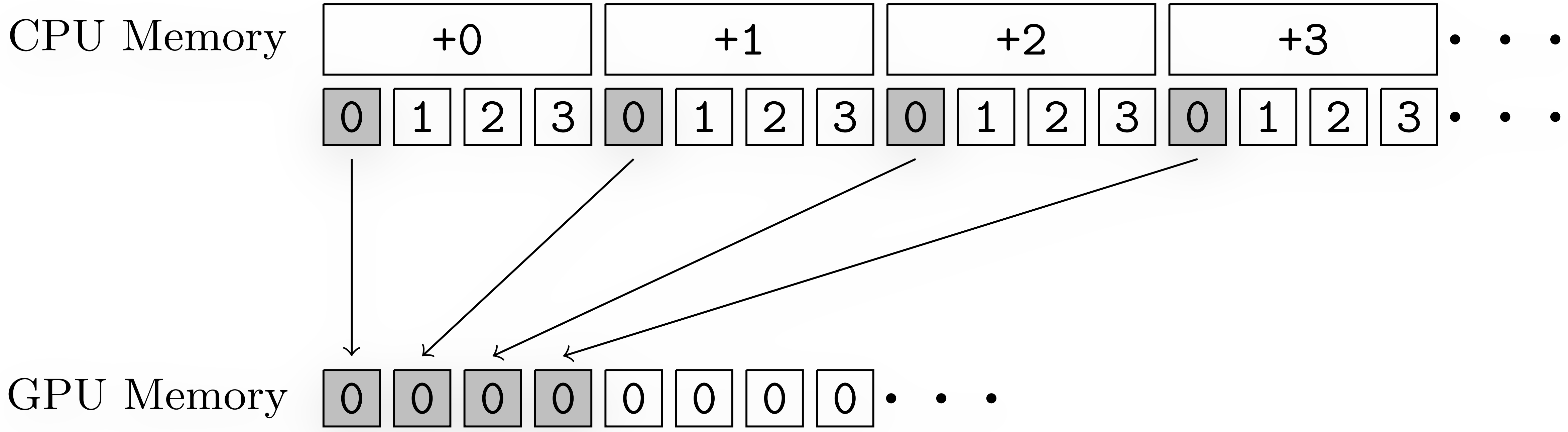
The table below show the different integrators implemented in the HiRep code. The last column in the table show how many times the next integrator level will be called in each iteration of the given integrator.

| Type | Description | Next level calls |
|---|---|---|
| `lf` | leap-frog integrator | 1 |

Table of Contents

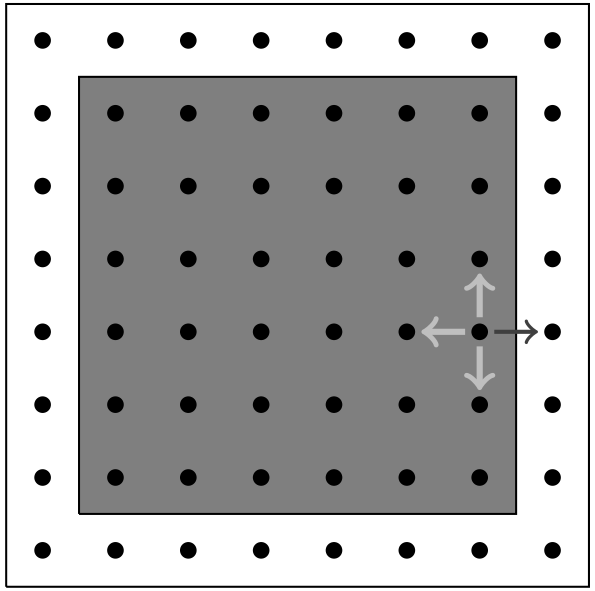
Integrator
Plaquette gauge
Lüscher-Weisz gauge
HMC Parameters
Twisted Mass
Twisted Mass Alternative
Hasenbusch
TM Hasenbusch
TM Hasenbusch Alternative
RHMC
Chronological Inverter

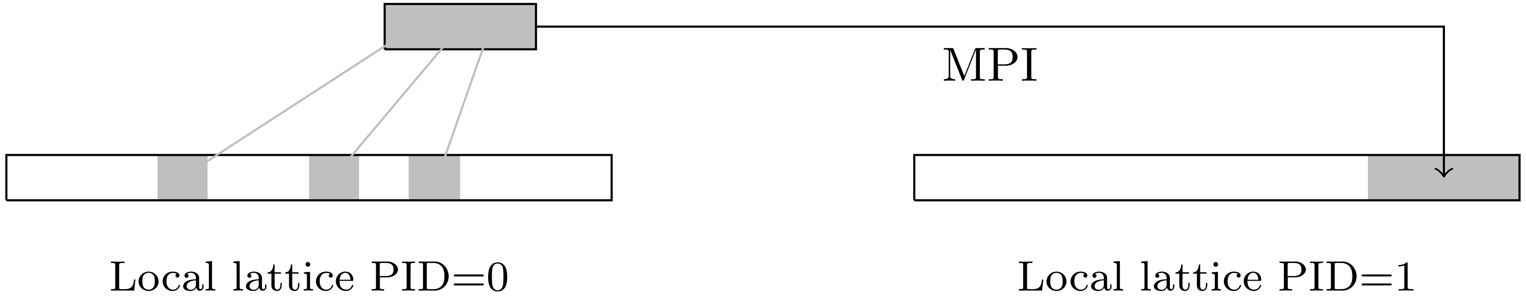
# An efficient Wilson-Dirac operator on GPUs




[SM et. al, 2024, 2405.19294, to appear in PoS]

# An efficient Wilson-Dirac operator on GPUs

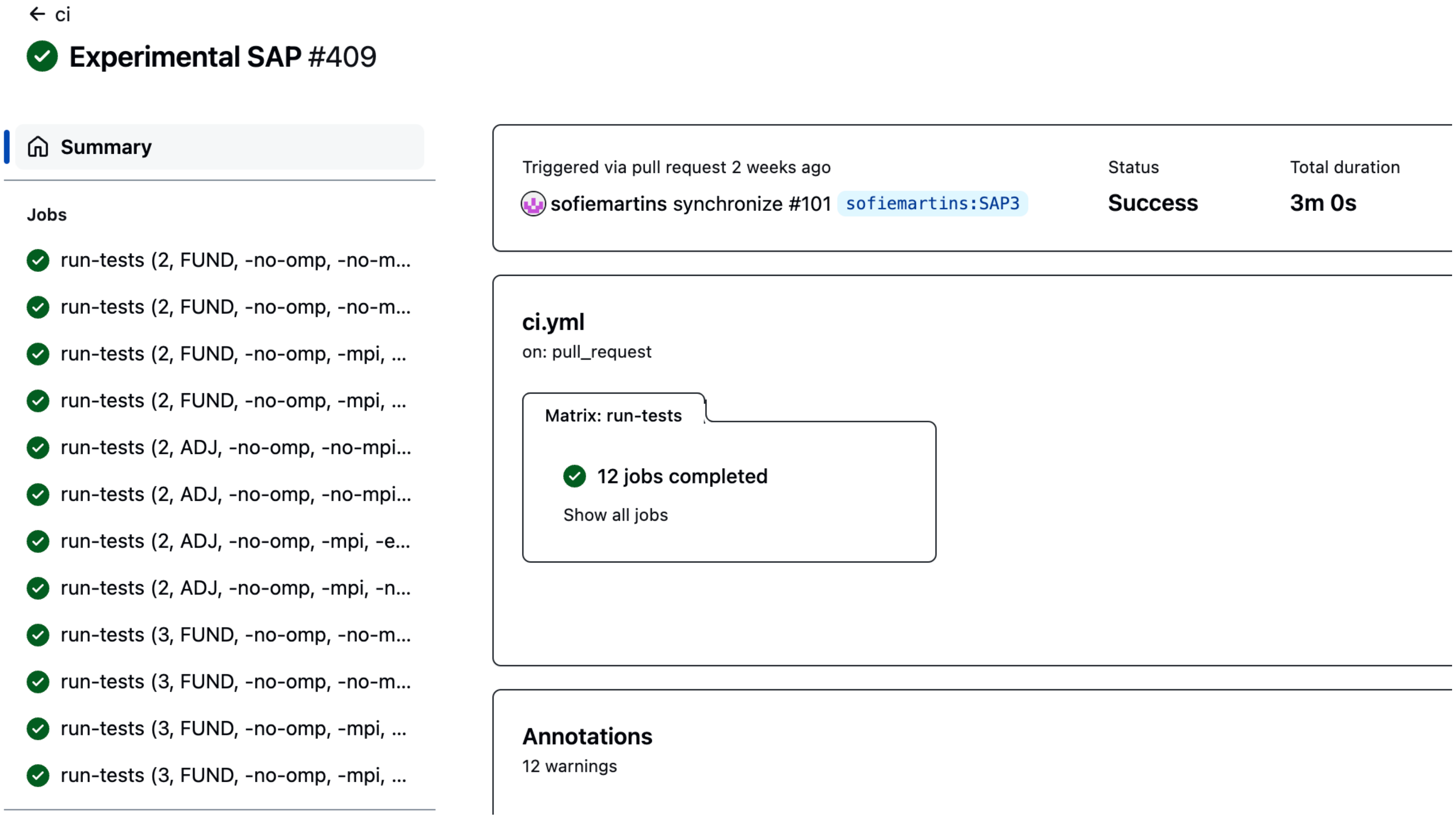





[SM et. al, 2024, 2405.19294, to appear in PoS]

# Software Quality

# CI / code coverage



# CI / code coverage

**codecov** (bot) commented 3 weeks ago • edited ▾ ···

## Codecov Report

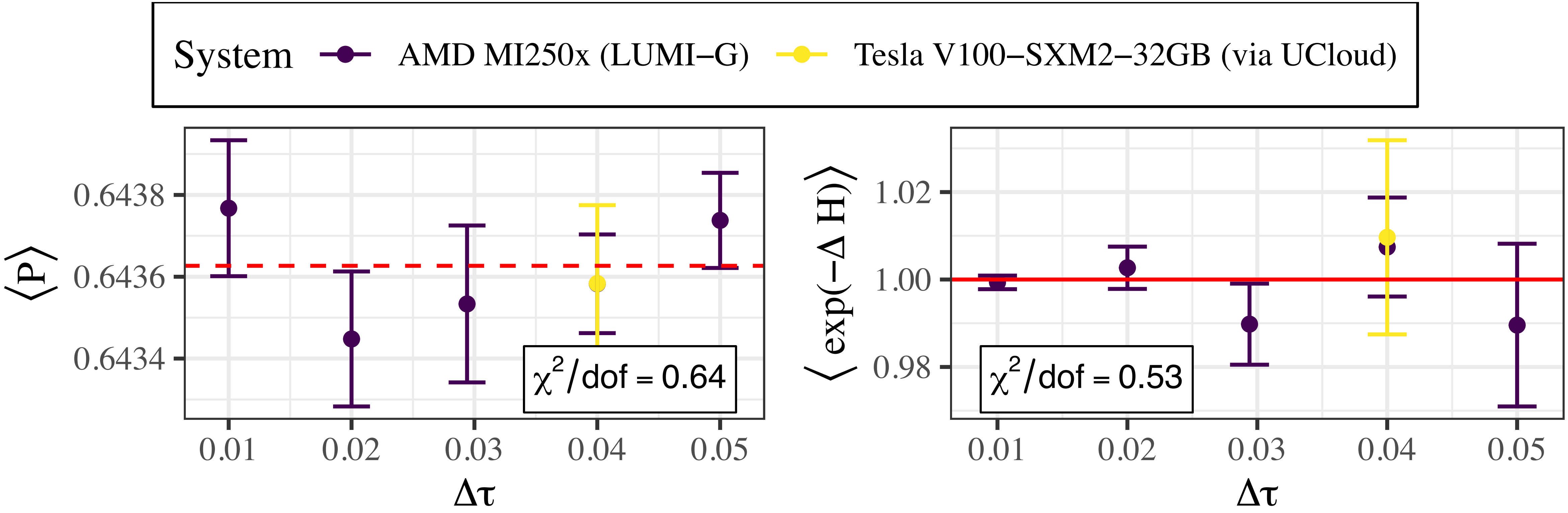
All modified and coverable lines are covered by tests ✅

> Project coverage is 49.26%. Comparing base (`7661b2d`) to head (`f78717b`).

▶ Additional details and impacted files

☔ View full report in Codecov by Sentry.
📢 Have feedback on the report? Share it here.

# The algorithm, SU(2) with adjoint fermions



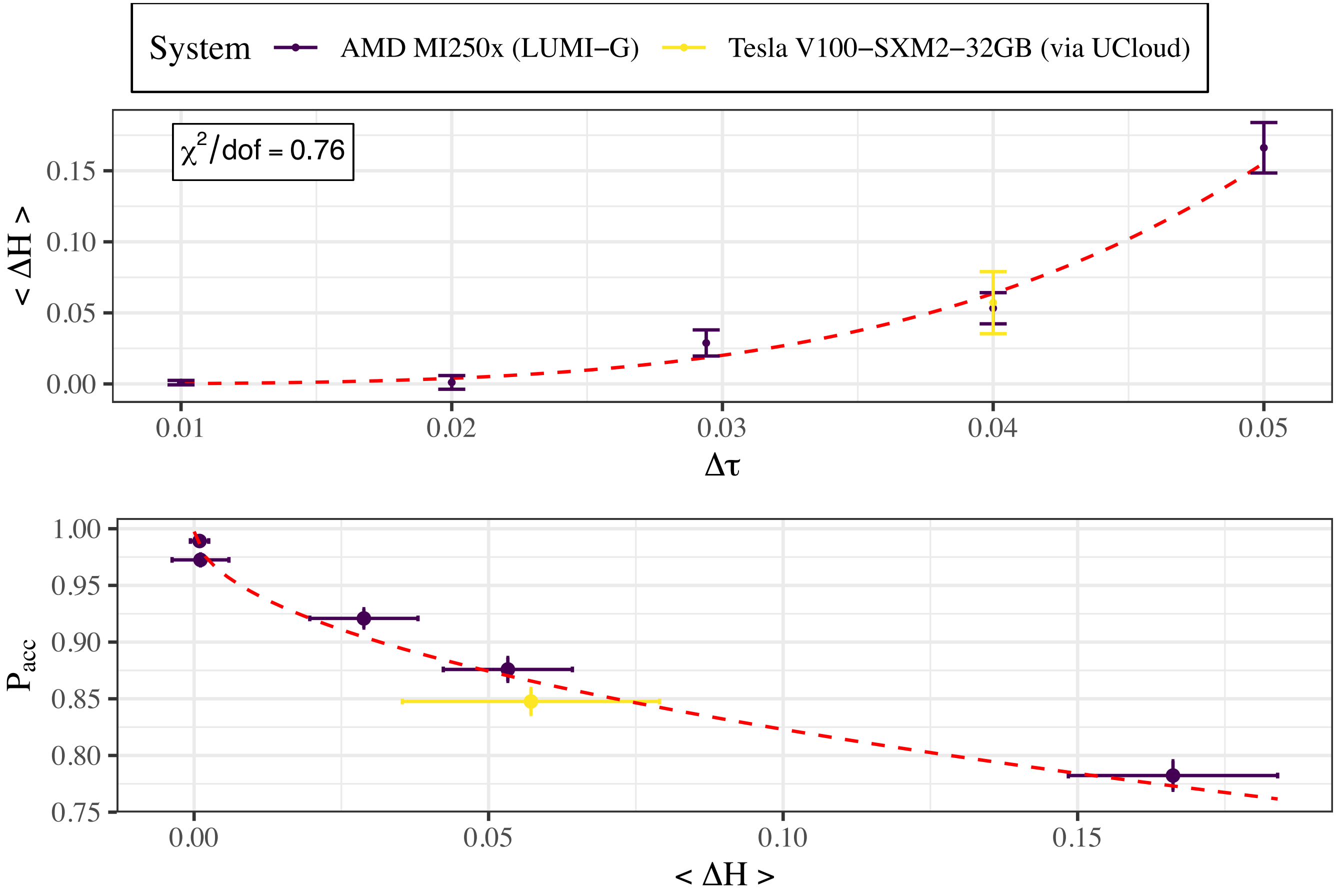

Testing Metropolis step (independence of plaquette value of step size)

$\langle \exp(-\Delta H) \rangle = 1$

[Del Debbio et. al., 2010, 0805.2058, Phys. Rev. D.] and references therein

[SM et. al, 2024, 2405.19294, to appear in PoS]

# The algorithm, SU(2) with adjoint fermions




Testing the 2nd order Omelyan Integrator

$$\Delta H \sim \Delta\tau^4$$

Testing correctness of The algorithm

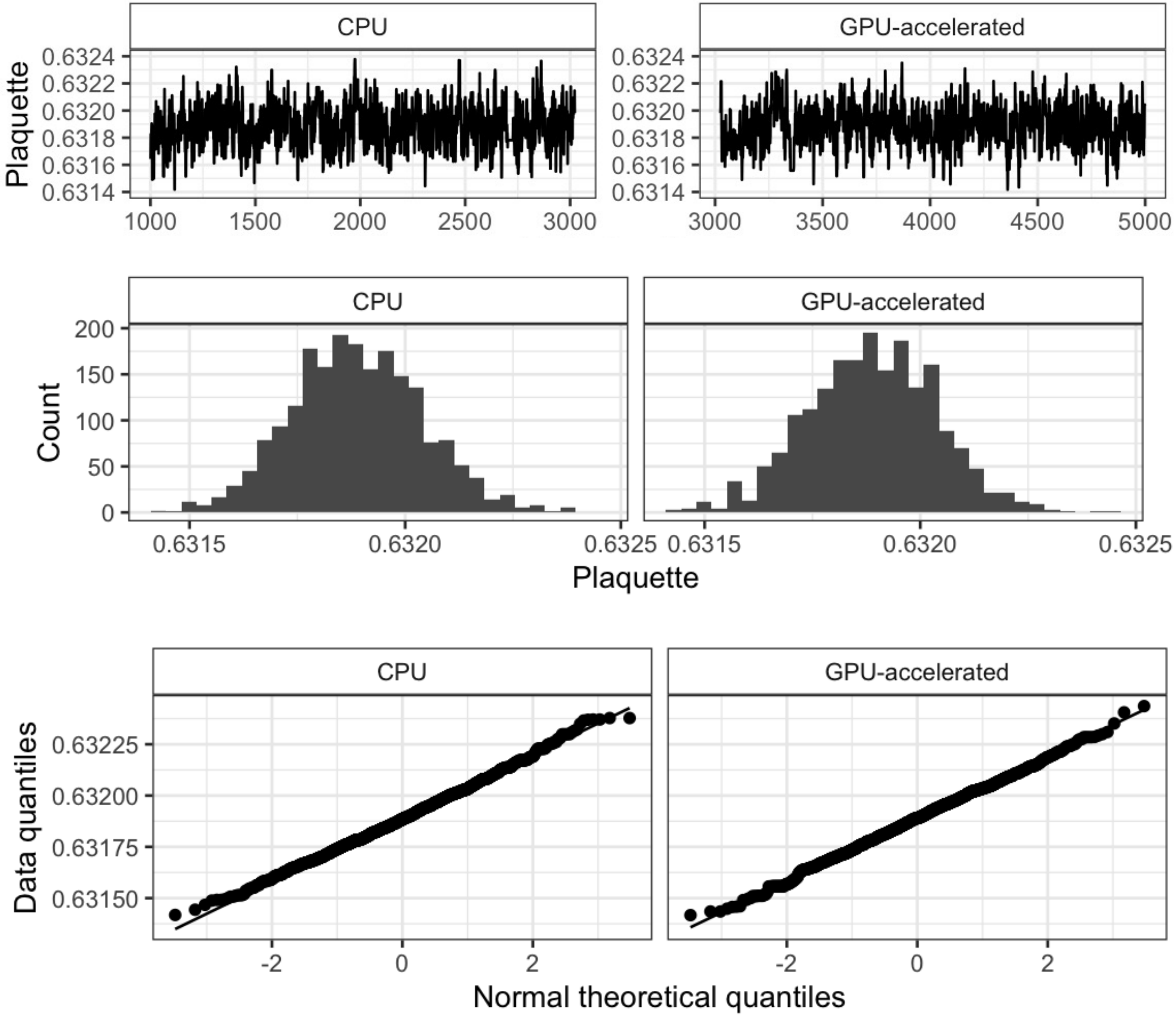
$$P_{acc} \cong \mathrm{erfc}\left(\sqrt{\frac{\langle \Delta H \rangle}{2}}\right)$$

[Del Debbio et. al., 2010, 0805.2058, Phys. Rev. D.] and references therein

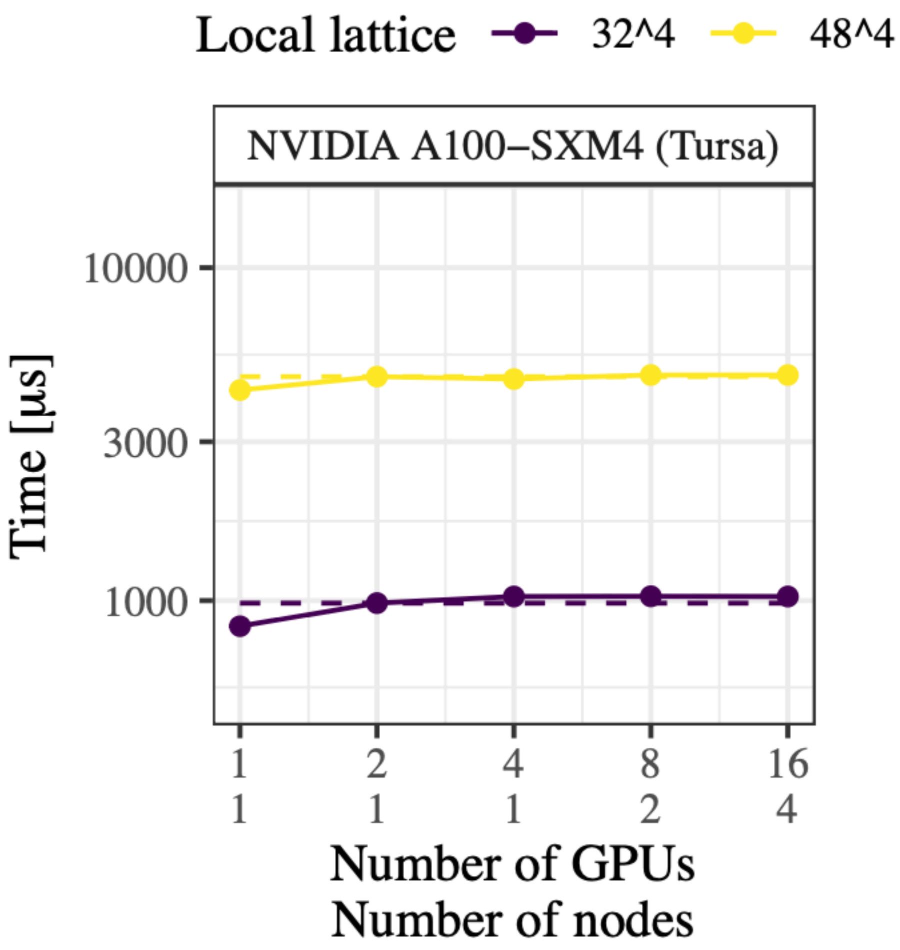
[SM et. al, 2024, 2405.19294, to appear in PoS]

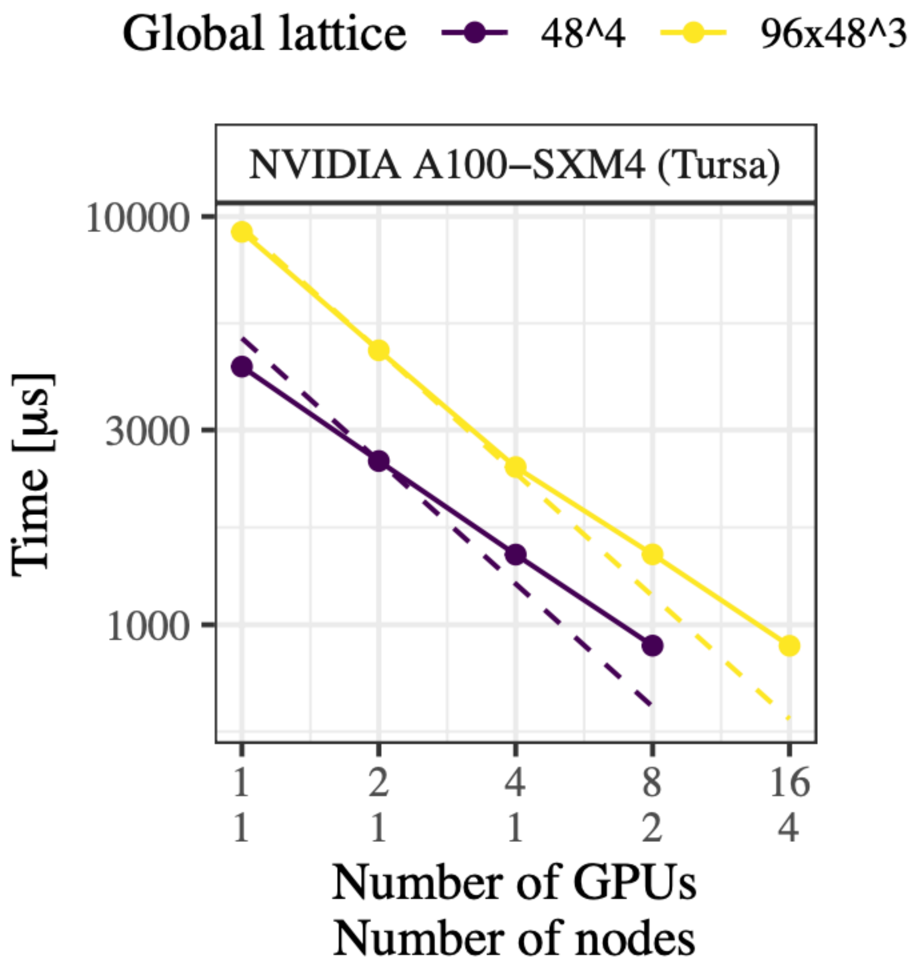
# Validation runs (SU(2)F)

- Very chiral ensembles with Hasenbusch Acceleration
- Single trajectory benchmark test: 1 node on tursa replaces 14 nodes on DIaL3 for 48^4 lattice
- Heavier validation run below: 36^4 Lattice for SU(2) with two fundamental flavors



# Performance on NVIDIA GPUs




[SM et. al, 2024, 2405.19294, to appear in PoS]

# Profiling


32^4, SU(3), fermions in the fundamental representation

# HiRep on LUMI-G

# Cray compiler toolchains

1. Use WITH_GPU and WITH_NEW_GEOMETRY for GPU support
2. Use COMMS_NONBLOCKING (only on LUMI-G!) for the fastest communications
3. Use mpicc compiler wrapper and replace the underlying C++ compiler with hipcc
4. Use –offload-arch=gfx90a in GPUFLAGS and LDFLAGS

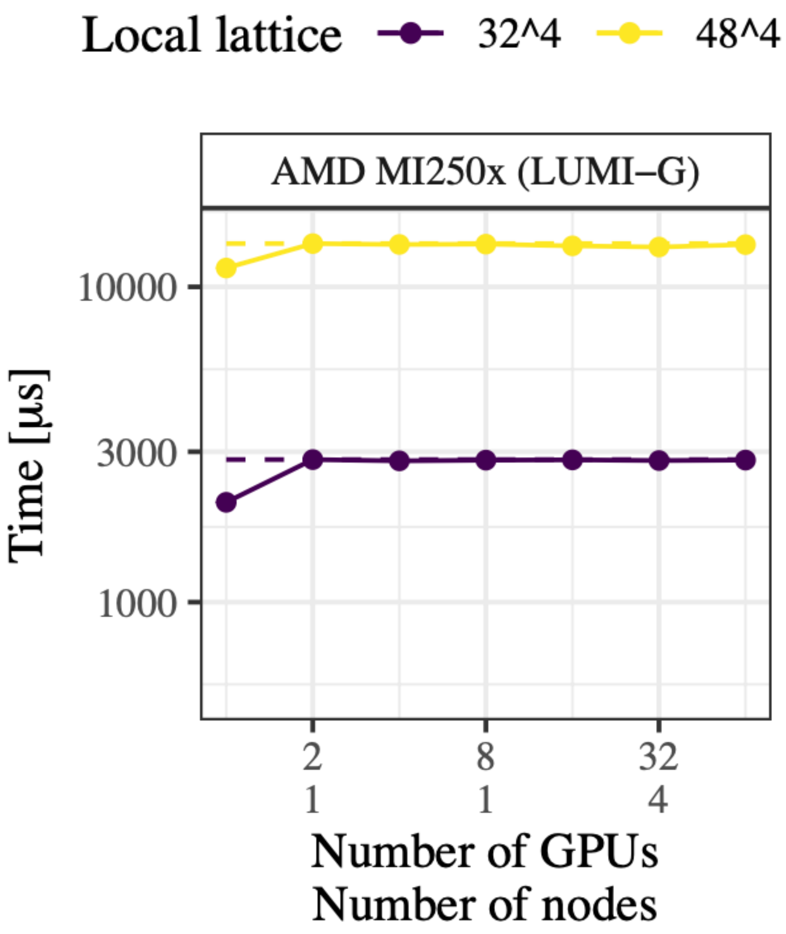
```
NG = 2
REPR = REPR_FUNDAMENTAL
GAUGE_GROUP = GAUGE_SUN
MACRO += BC_T_PERIODIC
MACRO += BC_X_PERIODIC
MACRO += BC_Y_PERIODIC
MACRO += BC_Z_PERIODIC
MACRO += UPDATE_EO
MACRO += NDEBUG
MACRO += CHECK_SPINOR_MATCHING
MACRO += IO_FLUSH
MACRO += WITH_MPI
MACRO += WITH_GPU
MACRO += WITH_NEW_GEOMETRY
MACRO += FIXED_STRIDE
MACRO += WITH_EXPCLOVER
MACRO += CUDA_CHECK_ERROR
MACRO += HIP
MACRO += COMMS_NONBLOCKING
ENV = MPICH_CC=hipcc
CC = gcc
MPICC = cc
CFLAGS = -Wall -O3
NVCC = mpicc
GPUFLAGS = -w --offload-arch=gfx90a
INCLUDE =
LDFLAGS = --offload-arch=gfx90a
```

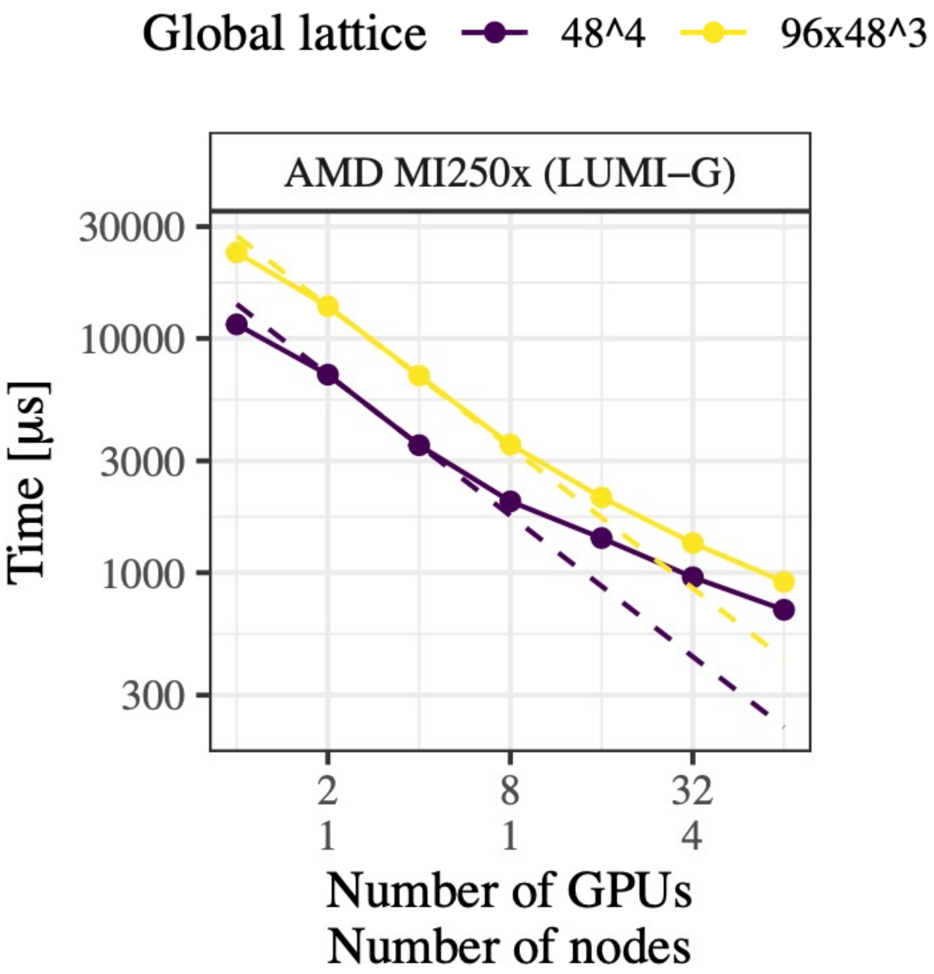
# HiRep selects the GPUs itself (using hwloc)

- Compile with hwloc if possible
- Select –ntasks-per-node not –-ntasks for multi-node jobs
- Do not use a wrapper script that pins the processes to the visible GPUs

```
#!/bin/bash -l
#SBATCH --job-name=test_hmc
#SBATCH --output=out/hmc.out%j
#SBATCH --error=err/hmc.err%j
#SBATCH --partition=small-g
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=8
#SBATCH --gpus-per-node=8
#SBATCH --time=3-00:00:00
#SBATCH --account=<project>

module load LUMI partition/G
module load PrgEnv-cray
module load craype-accel-amd-gfx90a
export LC_CTYPE=en_US.UTF-8
export LC_ALL=en_US.UTF-8
export PATH=$PATH:`pwd`/Make/
export MPICH_CC=hipcc


export OMP_NUM_THREADS=1
export MPICH_GPU_SUPPORT_ENABLED=1

srun -n 16 ./hmc -i input_file.in -o output_file.out
```

# Blocking and Nonblocking communications

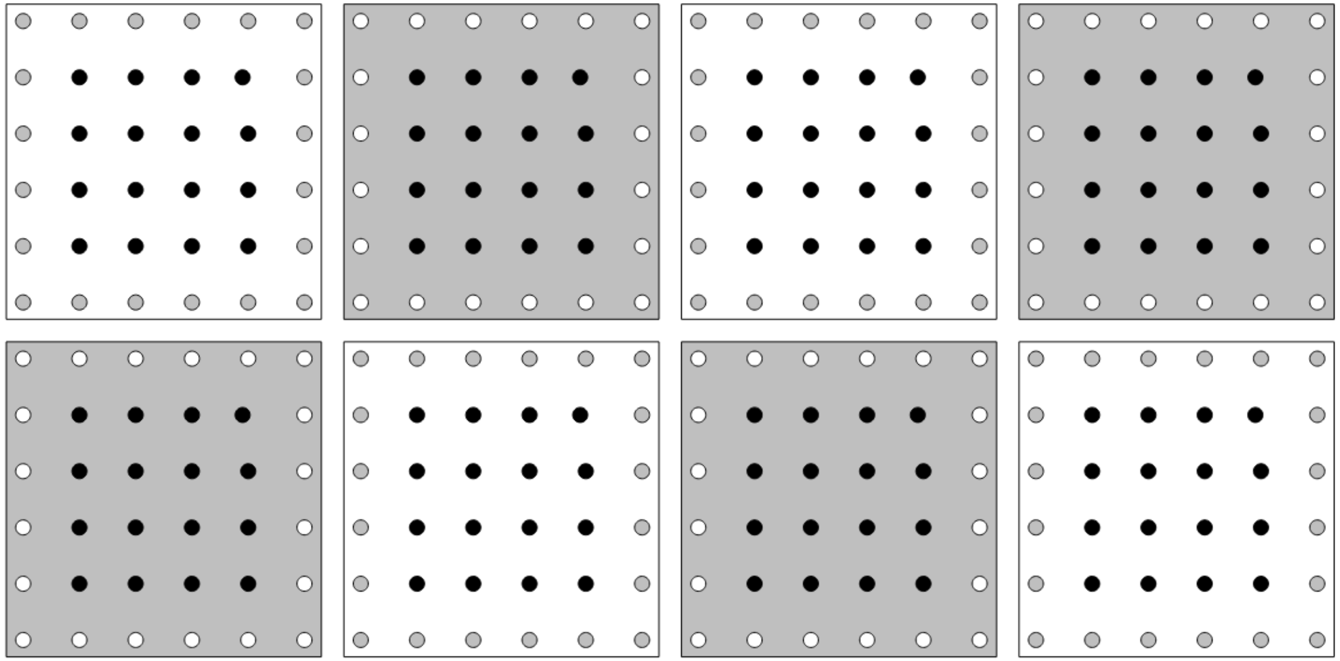



[SM et. al, 2024, 2405.19294, to appear in PoS]

# Outlook

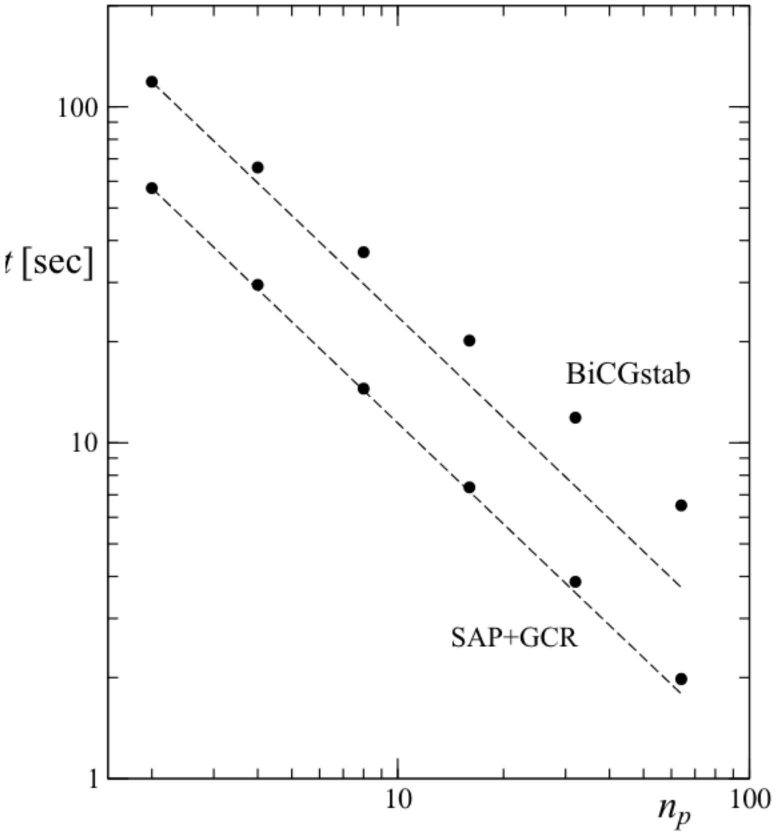
# Domain decomposition

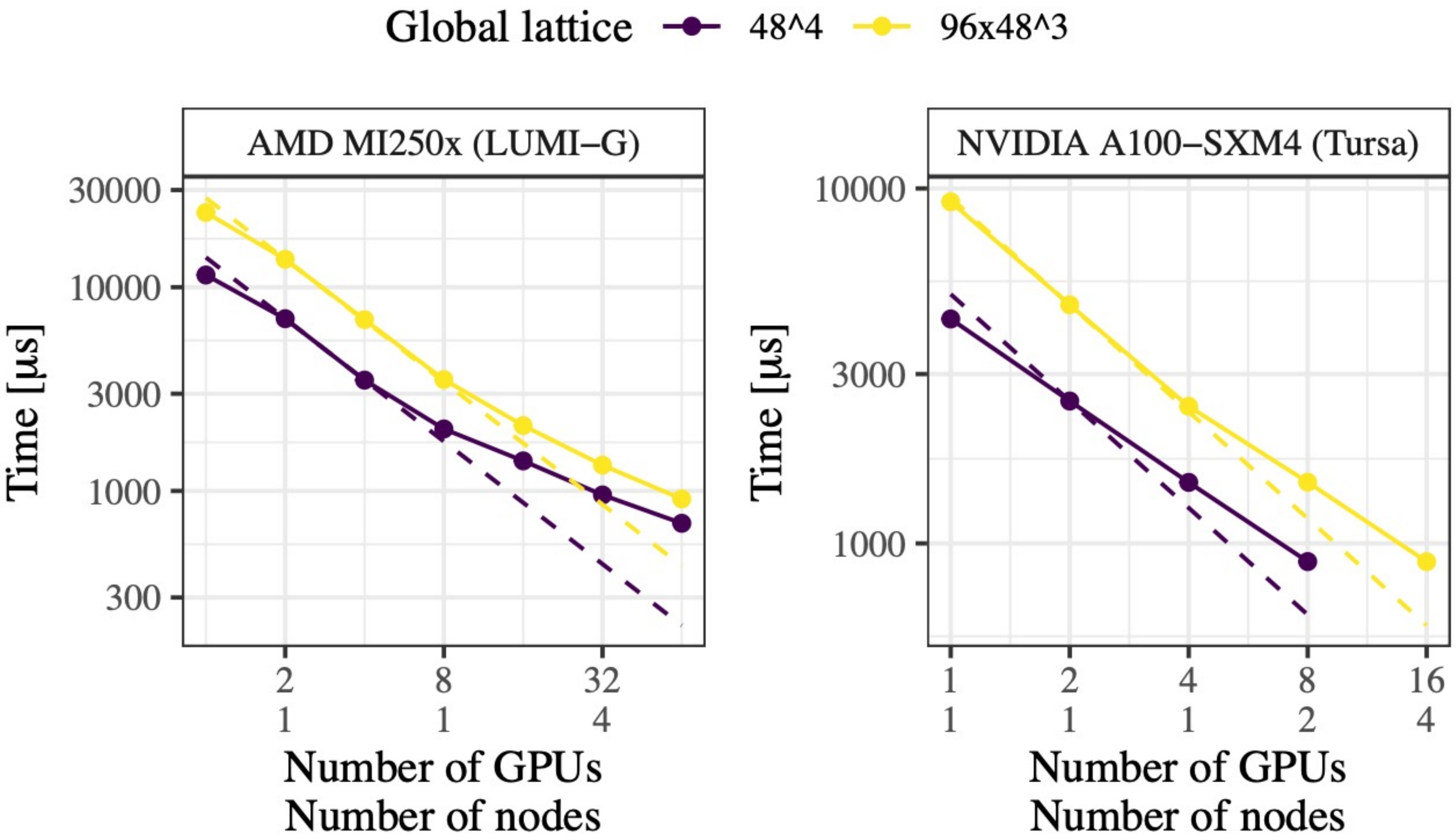


[M. Lüscher, 2004, hep-lat/0310048, Comput. Phys. Commun.]

# Domain Decomposition

[SM et. al, 2024, 2405.19294, to appear in PoS]




[M. Lüscher, 2004, hep-lat/0310048, Comput. Phys. Commun.]

# Conclusion

- A new version of HiRep is available that supports GPU acceleration
- The software is correct and scales well
- More features are on the way, for example domain decomposition and measurements

# Acknowledgements

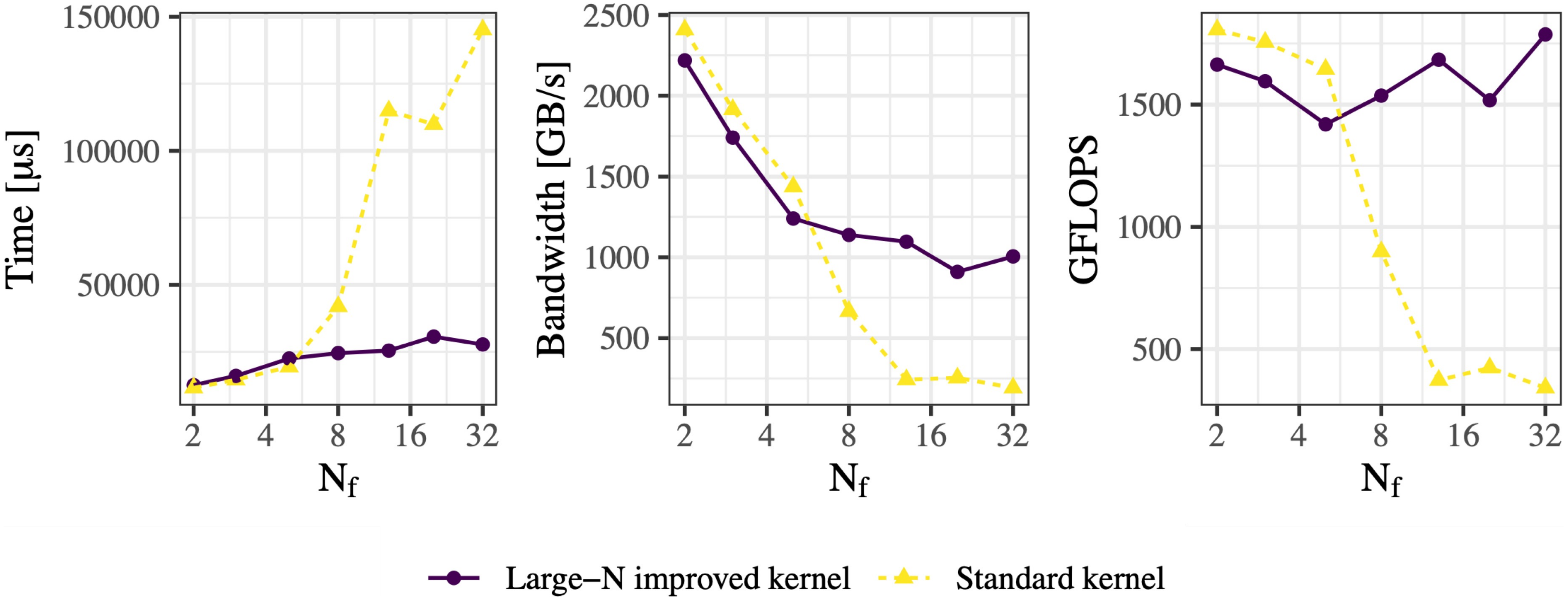
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement №813942.

Testing, development, and benchmarking was done on

- LUMI-G  using allocations provided by the Danish eInfrastructure Consortium (DeiC-SDU-N5-2024055)
- UCloud DeiC Interactive HPC system managed by the eScience Center at the University of Southern Denmark

# Backup

# Performance on NVIDIA GPUs




[SM et. al, 2024, 2405.19294, to appear in PoS]