Practical Issues in Cosmic String Simulations

David Daverio, PhD Student, Université de Genève

Work in collaboration with:
Martin Kunz & Mark Hindmarsh

Contents

- General Description
- LATfield2d
- LAtfield2d I/O
  - Parallel I/O
  - External non-blocking I/O server
- Some picture of cosmic string in a 2048^3 lattice

General Description

Abelian Higgs Model in FRW background

$$S = -\int d^4x\sqrt{g}\left(D_\mu\phi^* D^\mu\phi + V(\phi) + \frac{1}{4e^2}F_{\mu\nu}F^{\mu\nu} \right)$$

Metric : $ds^2 = a^2(\tau)\left(-d\tau^2 + dx^2\right)$

Covariant derivative : $D_\mu = \partial_\mu - iA\mu$

Potential : $V(\phi) = \frac{1}{2}\lambda\left(|\phi|^2 - m^2\right)^2$

General Description

Field simulation on a 4096^3 structured & static mesh.

Abelian Higgs :

2+2+3+3+10+10+10 = 40 float/site

$40 * 4 * 4096^3 = 11$ Tb

4096 * dx /(2dt) = 10240 time step

Need to FFT the 10 components of the E-M tensor
(FFT of 2.75 Tb!)

Need a highly scalable parallel code/framework !

LATfield2d

A c++ framework that can manage the parallelization of lattice based simulation

First version : LATfield (developed by Neil Bevis & Mark Hindmarsh)

1d mpi process grid, only serial I/O, FFT algorithm not optimized for the framework

Second version : LATfield2d (developed by DD)

2d mpi process grid, parallel I/O using HDF5 library, FFT algorithm for 3d cubic lattice optimized (based on 1d FFTW), External I/O server.

LATfield2d I/O

- Parallel I/O using HDF5-Parallel
  - Automatically recognizes data type
  - Automatically recognizes field type (scalar, vector,...)
  - One single call: field.saveHDF5("filename")
  - Output file directly readable in VisIt (visualization toolkit) even for parallel visualization (using Pixie)
- External I/O server (Beta)
  - Non-blocking I/O
  - Data transfers bandwidth up to 100 Tb/s (with ~2000 I/O cores on a Cray XE6)

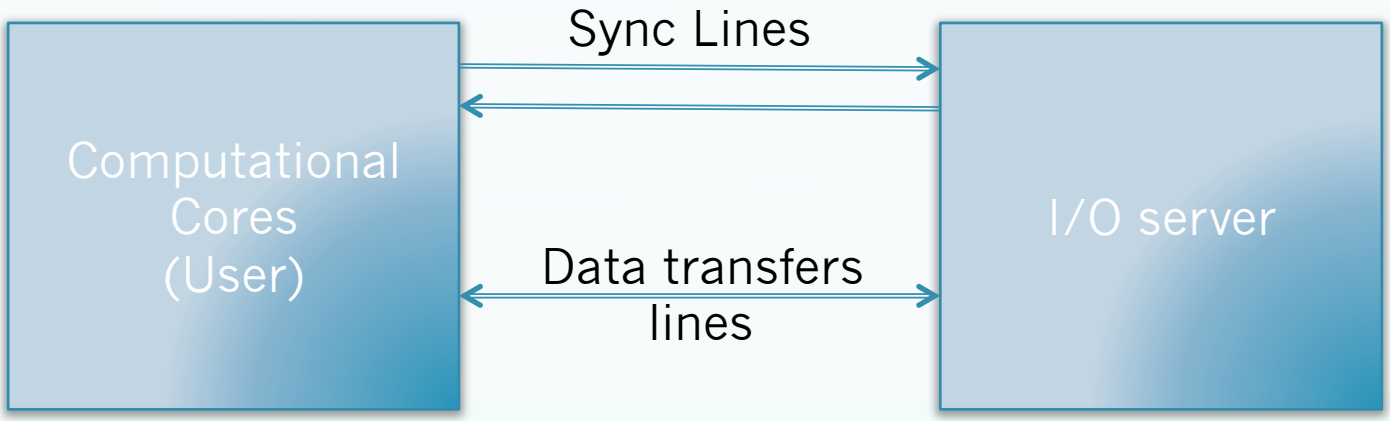
A non-blocking I/O

Idea: Create a I/O which works like a GUI: I/O event are queued like in a GUI.

Snapshots are taken only if the I/O is ready to download data.

This allows to save as many snapshots as possible, without overloading execution time. This is important due to the cost of 1 run: ~450'000 cpuhours:
(And we need at least 10 runs for statistics!)

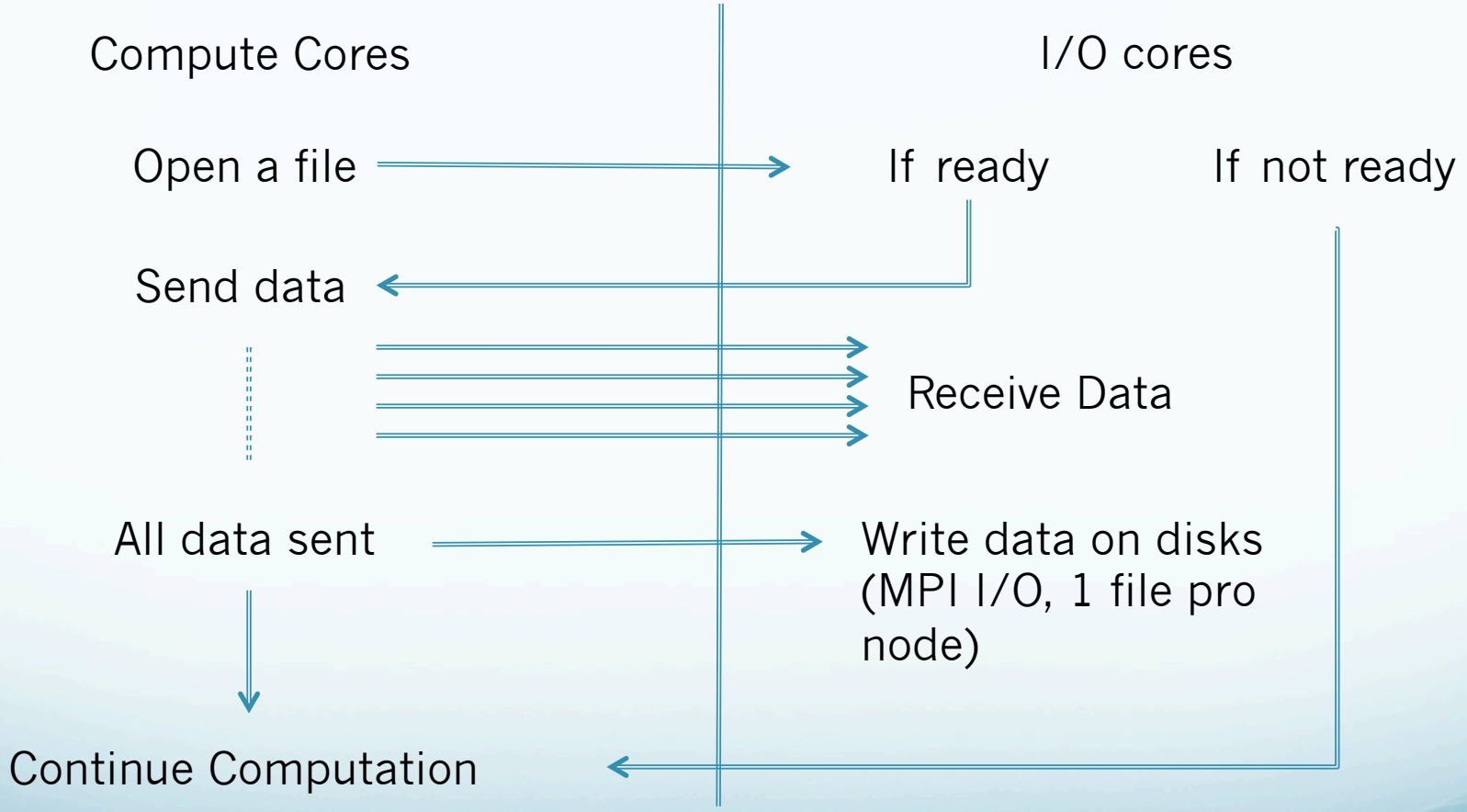
A non-blocking I/O



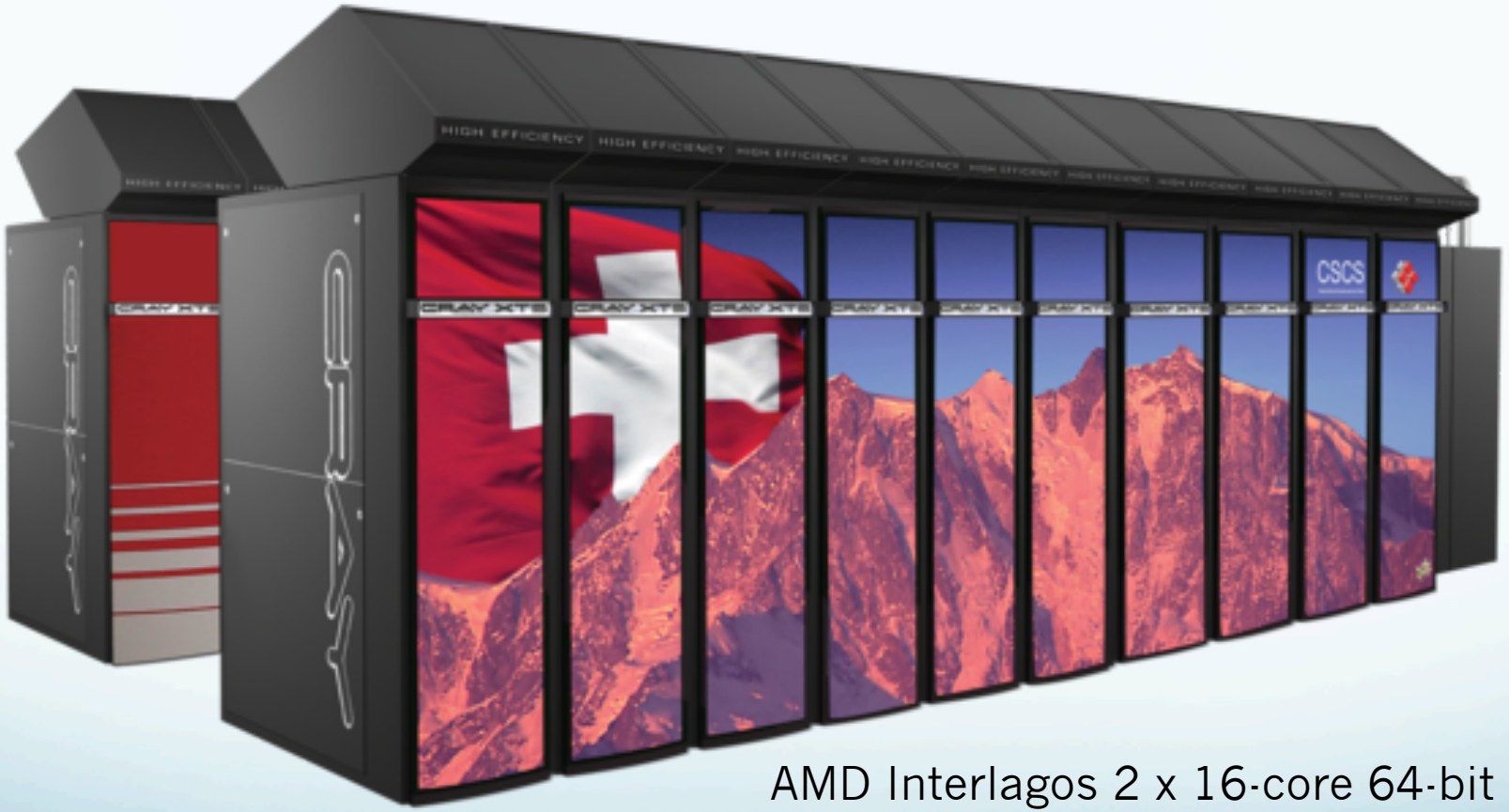
One-sided communication

Mpi send-rec

A non-blocking I/O



Monte Rosa (Cray XE6)



AMD Interlagos 2 x 16-core 64-bit CPUs
47872 cores, 402 Tflops
CSCS, Lugano, Switzerland

Some Pictures