ALPSCore:
Libraries for Physics Simulations
Core libraries of the ALPS project

Alexander Gaenko Andrey Antipov James LeBlanc Emanuel Gull

What is ALPSCore?

http://alpscore.org/

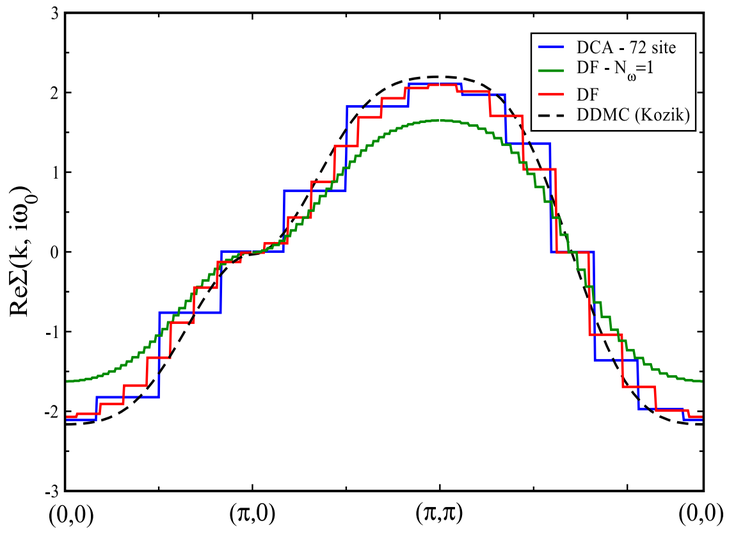
1. Open-source libraries and a framework for physics HPC simulations.
2. Aims to provide generic algorithms and utilities for physics problems.
3. Strives to increase software reuse in computational physics community.
4. Based on Algorighm and Libraries for Physics Simulations (ALPS, alps.comp-phys.org)
5. Takes most essential components of ALPS, providing compact C++ implementation and short development cycle.
6. Aims to be easy to maintain, install and use.

Why use ALPSCore?

1. Easy to install, simple interface
2. Ready-made command line and parameter files reading
3. Built-in statistics accumulation with error estimates
4. Easy checkpointing and restarting
5. Effortless MPI parallelization
6. Compact (~550 files, 400K lines of C++) with minimal external library dependence
7. Reduces time to develop and test complex scientific applications

Who uses ALPSCore?

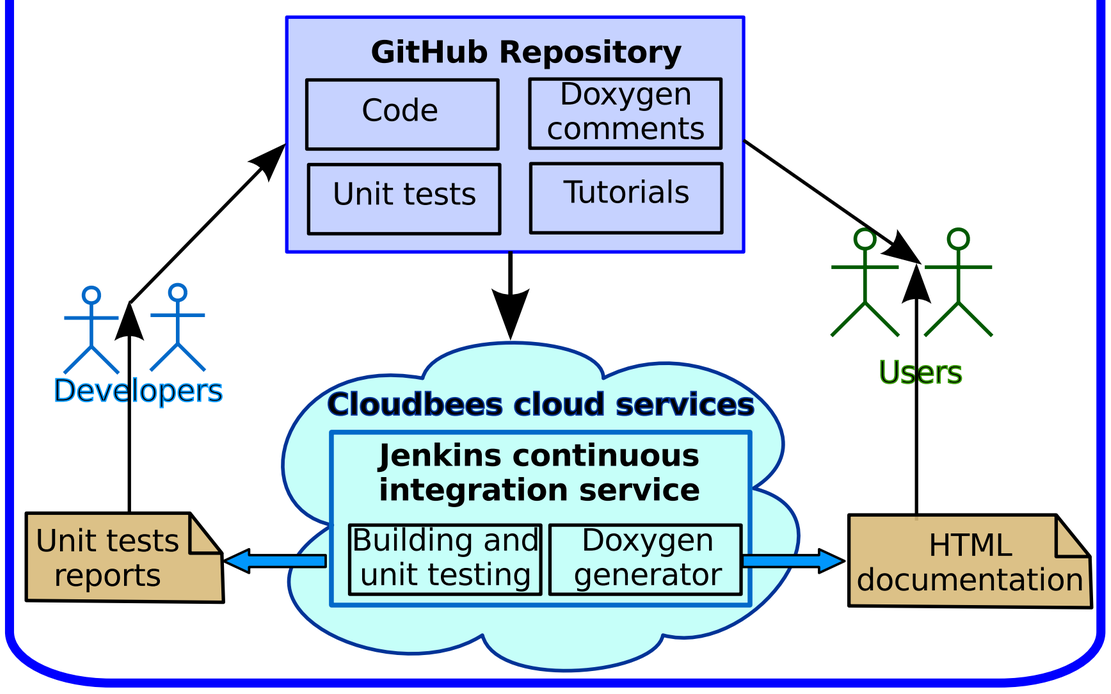
OpenDF: http://dx.doi.org/10.5281/zenodo.15007



ALPSCore package

- **params** — (Access to paramer file and command line)
- **accumulators** — (Gather statistical data with error estimates)
- **archive** — (Save to, restore from HDF5 files)
- **Monte Carlo scheduler** — (Distribute work via MPI, set termination criteria)
- **Python bindings** — (Access from Python scripts)
- **Tutorials** — (E.g., Ising model MC simulation)

Development Workflow



Example: Using accumulators

```
// Declare quantities:
using alps::accumulators;
accumulator_set measurem;
measurem << FullBinningAccumulator<double>("quantity1")
         << FullBinningAccumulator<double>("quantity2");

// ....
// Measure quantity at each MC step:
double q1=get_quantity1();
double q2=get_quantity2();
measurem["quantity1"] << q1;
measurem["quantity2"] << q2;

// ....
// Extract results
result_set results(measurem);
result_wrapper r1=results["quantity1"];
result_wrapper r2=results["quantity2"];
// Do arithmetics, get statistics:
int nmeasure = count(r1);
double mean_ratio = (r1/r2).mean<double>();
```

Accumulator *features*

- MeanAccumulator
- NoBinningAccumulator
- LogBinningAccumulator
- FullBinningAccumulator

Mean of a ratio -- not a ratio of means

Installing and using ALPSCore

Installation from source

```
$ cmake -DCMAKE_INSTALL_PREFIX=/where/to/install \
        -DBOOST_ROOT=/path/to/boost \
        -DENABLE_MPI=true \
        -DHDF5_DIR=/path/to/hdf5 \
        /path/to/alpscore-src
$ make
$ make test
$ make install
```

MacPorts and Debian packages: coming soon!

To use: put in your CMakeLists.txt

```
find_package(ALPSCore REQUIRED COMPONENTS hdf5 accumulators mc params)
  include_directories(${ALPSCore_INCLUDE_DIRS})
  link_libraries(${ALPSCore_LIBRARIES})
```

and invoke your cmake:

```
$ cmake your_project_options\
  -DALPS_ROOT=/path/to/alpscore/
```

UNIVERSITY OF MICHIGAN