

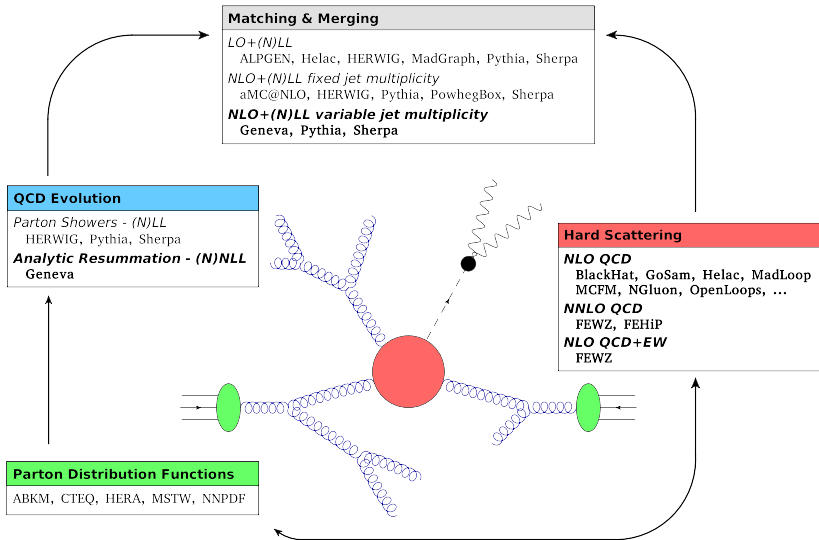
High Performance Computing for HEP Theory

Stefan Höche

SCA Meeting
SLAC, 04/17/13

HEP Theory and Computing

- ▶ Fixed-order (N)NLO calculations
 - ▶ Test new theoretical concepts in high-multiplicity processes
 - ▶ Generate parton-level event samples for theory & experiment
- ▶ QCD resummation
 - ▶ Validate algorithms for high fixed-order & logarithmic accuracy
 - ▶ Compute predictions / uncertainty estimates for many observables
- ▶ MC event generators
 - ▶ Validate algorithms to simulate high-multiplicity events at NLO
 - ▶ Tuning, release preparation, tests of applicability
- ▶ Interpretation of LHC data in terms of models for new physics
 - ▶ Generate large events samples, analyse for many hypotheses
 - ▶ Archive data and/or plots for usage by experiments
- ▶ ...



Snowmass Study on HPC for HEP Theory

Fast track:

- ▶ Draft white paper “The computing needs of theoretical high energy physics at the Energy Frontier” [Bern,Boughezal,Dixon,Hewett,SH,Petriello,Reina,Rizzo]
- ▶ Includes three case studies:
 - ▶ Perturbative QCD at NLO & NNLO
 - ▶ New physics searches in the pMSSM

Slow track:

- ▶ Started at EF East Coast meeting (BNL, 04/03-04/06)
 - ▶ Tutorial w/ “Hello world” and real-life HPC examples
 - ▶ Stirred up interest & signed up 14 people for case studies
 - ▶ Awaiting more input from community regarding use cases
- ▶ First follow-up planned at LoopFest (FSU, 05/13-05/16)
 - ▶ Feasibility studies on parallelization
 - ▶ Estimates of required resources per calculation
 - ▶ Wishlist of calculations

HPC Use Case #1: Higher-Order Perturbative QCD

Current status:

- ▶ No conceptual problems in breaking down NLO calculations into standard modular steps, automate them, and make them public
- ▶ Dedicated effort started in the context of Snowmass 2013 to address this project and have a realistic estimate of necessary resources
- ▶ First exploratory studies at NERSC show enormous potential to utilize codes which are readily available

Expected benefits of HPC:

- ▶ Various existing calculations/software can be made public within a common well-tested framework
- ▶ Allows experimental studies involving extensive multiple runs without depending on the computer- and manpower available to code authors
- ▶ New cutting-edge NLO and NNLO calculations can fully exploit parallel architectures and the availability of large resources

HTC vs. HPC

MC simulations / NLO pQCD calculations can be split into

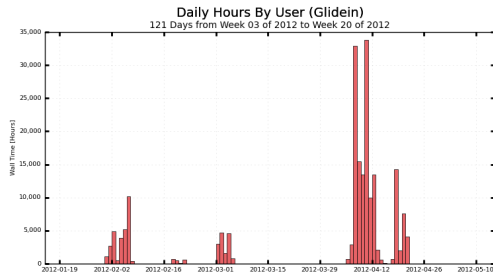
- ▶ Integration step
 - ▶ Determine total cross section and maximum for MC simulation
 - ▶ Use adaptive MC integrators to reduce variance
 - ▶ Store results in form of weight factors / grids
- ▶ Event generation step
 - ▶ Use weight factors / grids to increase efficiency
 - ▶ Produce full events rather than cross sections only
(Parton shower, Hadronization, Hadron decays, ...)

Natural separation into HPC and HTC domain

Optimizing resource usage could mean

- ▶ Integration performed in parallel
- ▶ Event generation distributed

HTC Test Case: MC Event Generator on OSG (Sherpa)

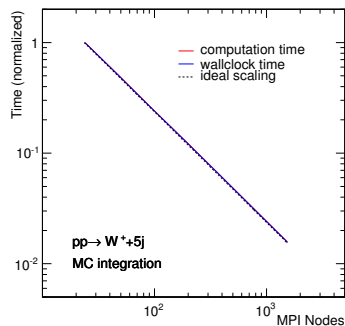


- ▶ SM backgrounds for MSSM parameter scan
- ▶ Tests of NLO ME+PS merging methods
- ▶ Single core application
- ▶ 195k CPU hours
1TB of data

Source: OSG accounting (<http://gratiaweb.grid.iu.edu/>)

- ▶ Need to pre-stage large binaries / libraries
→ good for production but not for development
- ▶ Steep learning curve for users w/o Grid experience
→ needs more user support than a local resource

HPC Test Case: NLO pQCD MC at NERSC (BlackHat)



- ▶ $pp \rightarrow W^+ + 5\text{jets}$ at NLO
Here: 1 out of 55 channels
- ▶ Adaptive integrator
Multichannel MC + VEGAS
- ▶ Variable number of points
per optimization step

- ▶ Optimum performance at around 100 - 1000 cores
- ▶ Can be improved further by OpenMP (tested at UCLA)

HPC Use Case #2: New Physics Searches

Current status:

- ▶ Development of tools to explore parameter spaces of new physics beyond the Standard Model in as much generality as possible
- ▶ Availability and efficiency will become more and more crucial during extensive data analysis at down times of LHC and after upgrade

Expected benefits of HPC:

- ▶ Availability of large resources with short I/O times
→ greatly increases throughput
- ▶ HPC data centers facilitate easy exchange of data between theory and experiment

Interpretation of LHC results in terms of BSM models

Three-stage process

1. Event generation using (N)LO Monte-Carlo program
2. Object reconstruction based on simple detector simulation
existing tools like PGS and Delphes
3. Mockup of public experimental analyses
or design of new, dedicated ones

Do this for $O(100k)$ parameter points!

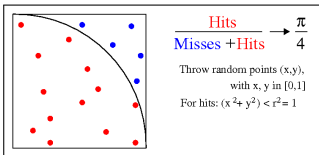
- ▶ Each run is an independent single-core job
- ▶ I/O is the main bottleneck after events are generated
- ▶ Storing output can become prohibitively expensive

HPC Tutorial at EF East Coast meeting

<http://snowmass2013.org/tiki-index.php?page=HPC+Tutorial+at+BNL>



Snowmass on the Mississippi s.k.a. CSS 2013



Quick Links

▼ [TWiki registration](#)

▼ Pre-meetings

- Community Planning
- Meeting
- All pre-Snowmass
- Meetings

Groups

- Energy Frontier
- Intensity Frontier
- Cosmic Frontier
- Frontier Capabilities
- Instrumentation
- Frontier
- Computing Frontier
- Education and Outreach
- Theory Panel

Google Search

- © snowmass2013.org
- © WWW

HPC Tutorial at BNL

A tutorial on High Performance Computing for HEP Theory will take place at the [Energy Frontier East Coast Meeting](#) on Thursday, 04/04, 12:30-2:00pm in Room C. It is part of the discussion on [how to best use HPC facilities for HEP theory](#).

If you plan to attend, please sign up for a NERSC computing account beforehand by sending an email to shoehc@slac.stanford.edu. You can use this account after the tutorial for HPC studies.

The tutorial will cover simple MPI / OpenMP programming and an example for using MPI to compute NLO cross sections efficiently. Please also plan to attend the general session on HPC for THEP on Wednesday, 04/03, 2:00-3:00pm in Room B.

USEFUL LINKS

[Getting started at NERSC](#)
[NERSC Tutorials](#)

INTRODUCTION

For details on the HPC for THEP project please refer to its [home page](#). Some remarks on the HEP are also given in this [talk](#).

The tutorial will exemplify the relatively simple structure of parallel programs compared to code. Participants should indicate their availability for exploratory studies in the course of chart a roadmap for the use High Performance Computing in HEP theory with the help of

INSTRUCTIONS

Log on to NERSC and switch the compiler suite to gcc

```
ssh <username>@hopper.nersc.gov
module unload PrgEnv-gpu
module load PrgEnv-gcc
module load trading
```

Copy the examples using

```
cp -r $EXAMPLES
```

Log on to a mom node (a service node that is used to parse batch jobs and launch parallel 24 cores available for your tests. Sessions time out after 30 minutes.

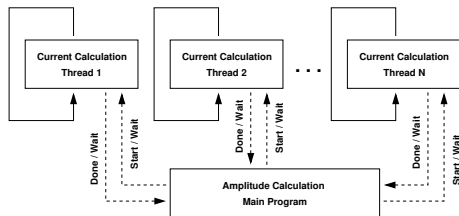
```
qsub -IV -q interactive -l mppwidth=24
```

Simple C++ MPI program

```
1 // attach to MPI
2 MPI::Init(argc, argv);
3 // get size of and rank in MPI environment
4 int size = MPI::COMM_WORLD.Get_size();
5 int rank = MPI::COMM_WORLD.Get_rank();
6 // initialize random number generator
7 srand(rank);
8 // hit it!
9 double n = 1.0e6, mysum = 0.0;
10 for (double i = 0.0; i < n; ++i) {
11     double x = rand() / (double)RANDJMAX;
12     double y = rand() / (double)RANDJMAX;
13     if (x*x+y*y < 1.0) ++mysum;
14 }
15 // collect results
16 n = size;
17 double sum;
18 MPI::COMM_WORLD.Reduce(&mysum, &sum, 1, MPI::DOUBLE, MPI::SUM, 0);
19 if (rank == 0) {
20     // compute final answer
21     double pi = 4.0 * sum / n;
22     double sig = 4.0 * sqrt(((sum/n - sum*sum/n/n) / (n-1.0)));
23     std::cout << "pi = " << pi << " +- " << sig << std::endl;
24 }
25 // detach from MPI
26 MPI::Finalize();
```

HPC Tutorial at EF East Coast meeting

- ▶ Covered MPI and OpenMP
- ▶ Explained thread-safety and race conditions
- ▶ C++, FORTRAN and Python examples
- ▶ Pointed out use cases and references
 - ▶ MPI → Adaptive Monte-Carlo integration
 - ▶ OpenMP → Recursive relations in perturbative QCD



- ▶ Real-life example: cross section calculation for $pp \rightarrow t\bar{t}H$

Summary

- ▶ Draft white paper on HPC/HTC for HEP circulated collecting responses from community
- ▶ HPC at NERSC tested for cutting edge pQCD / MC ~ scaling shown, ready for production
- ▶ Small-scale MPI runs also tested on OSG ... it works

Ongoing Activities

- ▶ Signing up people for Snowmass HPC/HTC study
- ▶ Building knowledge base “HPC for THEP”
- ▶ Testing other architectures (IBM BlueGene)