

Lattice QCD Workflows

requirements from a user's perspective

James N. Simone

Fermilab – Dec 18, 2006

`simone@fnal.gov`

— Need for a workflow system —

- QCD workflows commonly exploit a large supercomputer's capacity by running many independent streams of batch jobs. Shepherding these jobs is time consuming.
- Resuming runs after crashes is tedious. Programming recovery logic into scripts is complex and prone to error. Duplicating successful runs, however, is wasteful.
- Unorchestrated job streams compete for system resources; leads to bottlenecks and/or demand beyond system's capacity to cope.
- Recording the provenance of scientific results is important.

— Domain-specific definitions —

Lattice: Four dimensional space-time grid consisting of sites with coordinates (x, y, z, t) . Links connect each site to its nearest neighbor sites.

(Gauge) configuration: A snapshot of the gluon field. Represented by a $SU(3)$ matrix on every lattice link. File size: $3 \times 3 \times \text{sizeof}(\text{complex}) \times 4 \times L_s^3 \times L_t$ bytes. now: 0.61, 1.8 and 4.6 GB

Ensemble: An ordered collection of configurations sharing the same physics parameters e.g. lattice spacing, number of sea quarks and their masses. A new configuration is generated by inputting an existing configuration and evolving it (e.g. via HMC) a number of steps in simulation time.

Configuration generation



- “genU” program runs on a large number (e.g. 256) nodes; input: the previous configuration; output: a new configuration.
- In this example, a new configuration is produced every 6 MC steps.
- Branches: same input files, different random number generator seeds. Increases the number of streams producing configurations.
- Typically, config. generation is stopped when an ensemble of 600 or more gauge configs is produced. May resume later in order to increase statistics.

— Domain-specific definitions —

Quark Propagator: Quark variables live on lattice sites.

At every site, heavy quarks are represented by a $4 \times 4(\text{spin}) \times 3 \times 3(\text{color})$ complex matrix. Light (staggered) quarks are represented by a 3×3 complex matrix. Quark propagator files are produced (and saved) as intermediate results. On a $40^3 \times 96$ lattice: 7.1 GB (hQ) and 0.44 GB (sQ).

2- and 3-pt functions: Created by combining quark propagators. Typically stored as L_t complex numbers. Hundreds of n -point functions may be produced in the analysis of an ensemble.

Campaign: A coordinated set of calculations aimed at determining a specific set of physical quantities – for example, the masses and decay constants of the D

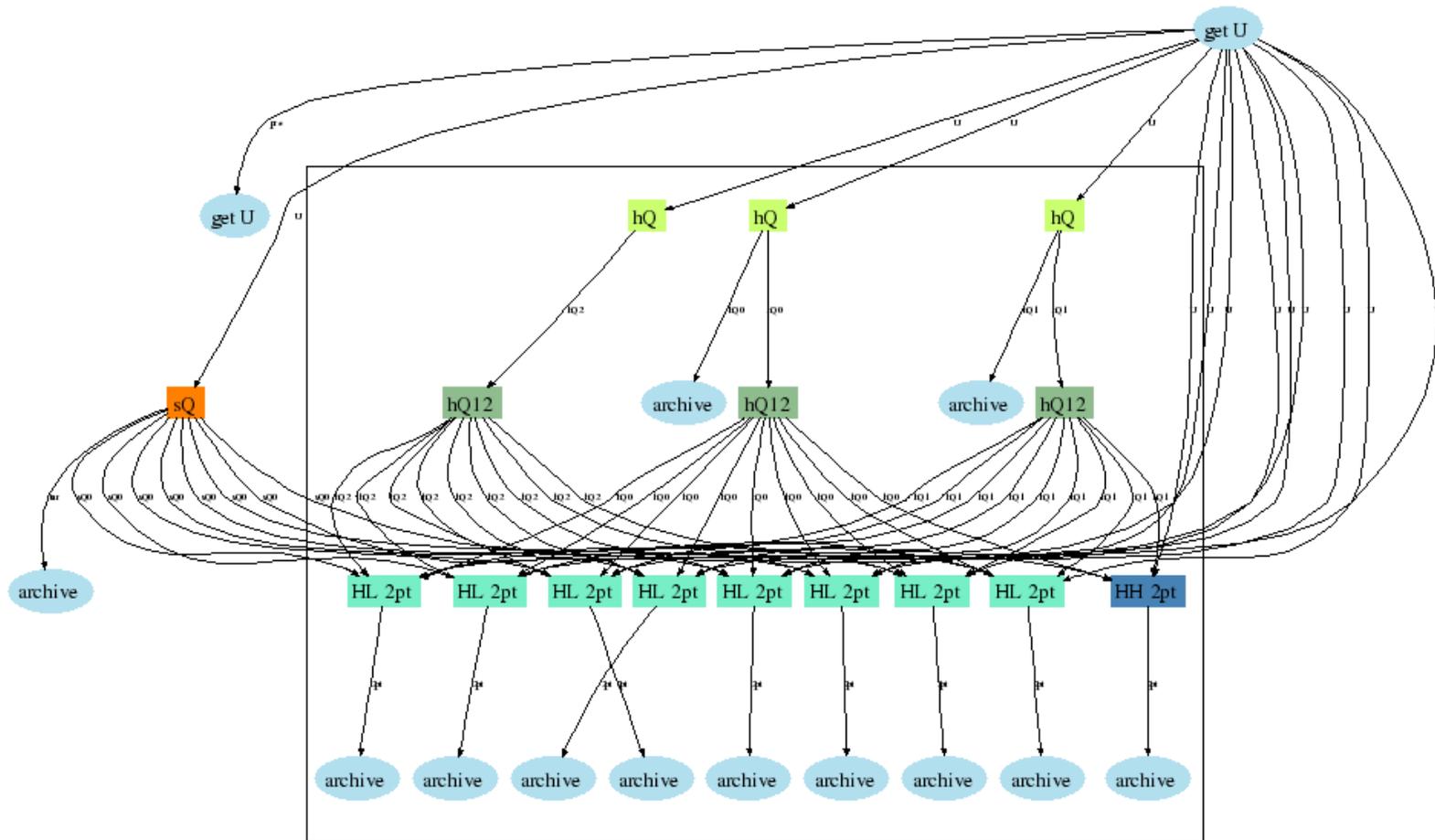
and B mesons. A typical campaign consists of taking an ensemble of configurations and using them to create intermediate data products (e.g. quark propagators) and computing n -point functions for every configuration.

An important feature of such a campaign is that calculations done for each configuration are independent of the other configurations. Several execution streams of a campaign may be run to exploit the capacity of a computer facility.

A campaign could consist of a single workflow, where intermediate products are used immediately, or it could be decomposed into multiple workflows, with intermediate products stored for later use.

Two-point campaign

for U in ensemble :



Participants

getU: (pre)fetch gauge configuration; unix proc.

input: config. number

output: config. file

hQ: solve heavy quark prop.; MPI $np = 64$

input: gauge config

output: heavy quark file

hQ12: reformat heavy quark prop.; unix proc.

input: heavy quark file

output: reformatted heavy quark file

sQ: multi-solve stag. quark props.; MPI $np = 64$

input: gauge config

output: ~ 8 staggered quark files

HH 2-pt: heavy-heavy 2-pts; MPI $np = 4$

input: gauge config; 3 heavy quark props.

output: ~ 80 2-pt functions

HL 2-pt: heavy-light 2-pts; MPI $np = 4$

input: gauge config; 1 stag. and 3 heavy quark props.

output: ~ 80 2-pt functions

arch: archive data products; unix proc.

input: file(s) to archive

output: none

Participant wrappers

\$ Staggered -help

Staggered arguments

```
-single (run single cpu)
-qdp (use MILC QDP version)
-ib ncpus (use mvapich; -single overrides this flag)
-gm ncpus (use myrinet; -single overrides this flag)
-vmi ncpus:[myrinet|openib]; (use vmi; -single overrides this flag)
    (NOTE: -ib -gm and -vmi are mutually exclusive)
-dim x,y,z,t
-beta b.bbb (required yet unused!)
-sea n:m.mmm,n:m.mmm (flavors:mass required yet unused!)
-u0 n.nnnnn mean-field tadpole
-source local,t (repeat arg -source type,tsrc for multiple sources)
-gauge fgauge ('none' for cold config.)
-fix (default is no fixing)
-fixout [none|fixgFile] (save fixed gauge in fixgFile)
-mass m.mmm [-mass d.ddd] (stag. bare mass; repeat arg for multimass)
-iterations nnn maximum CG iterations
-precision n.nne-dd (target absolute? precision)
-output proto (prototype output filename; files e.g. proto_m0.007_t0)
-cd dirname (change directory to dirname first)
-verbose print larger number of diagnostic messages
-bin dir (alternate directory for staggered executables)
-env ARG=value [-env ...] (set environment variable(s) for executable)
```

run framework

- Custom (perl) A 'run' command to control/monitor campaign execution.
- Used (by me) for about a decade (ACPMAPS + 3 gen. clusters)
- Can split campaign into separate streams of execution.
- Requires perl expertise!
- Run directs participant stdin/stdout to history directory. Data product provenance.
- History (semi)-automates resume from last successful **milestone**.

Definitions

Participant: An object that transforms inputs into outputs. All participants are considered atomic operations from the executing workflow's point of view.

Milestone: The persistent state of the last intermediate result reached by a workflow instance. A milestone can be used for recovering a workflow instance or to extend a campaign...

Milestones and resume

```

$ less LOGS/Sc.log
501722 hQ_0.0923_2S [11750.kaon2.fnal.gov] was RUNNING now DONE
501722 hQ_0.0923_2S [11750.kaon2.fnal.gov] DONE (status: 0) 31146 sec
501722 twoPoint_0.0923 [11843.kaon2.fnal.gov] QUEUED
501722 twoPoint_0.0923 [11843.kaon2.fnal.gov] was QUEUED now RUNNING
501722 twoPoint_0.0923 [11843.kaon2.fnal.gov] was RUNNING now FAILED
501722 twoPoint_0.0923 [11843.kaon2.fnal.gov] FAILED (status: 1) 10809 sec
501722 FAILED 51692 s
run@kaon1.fnal.gov 0.24 (Exp) (simone) [3516]
000366 already done
001302 already done
#
501722 starting... Thu Nov 23 10:23:02 CST 2006
501722 already completed twoPoint_0.127
501722 already completed hQ_0.0923_1S
501722 already completed lightQuark
501722 already completed hQ_0.0923_2S
501722 already completed hQ_0.127_2S
501722 already completed cacheU
501722 already completed hQ_0.0923_d
501722 already completed hQ_0.127_1S
501722 already completed hQ_0.127_d
501722 twoPoint_0.0923 [11939.kaon2.fnal.gov] QUEUED
501722 twoPoint_0.0923 [11939.kaon2.fnal.gov] was QUEUED now RUNNING
501722 twoPoint_0.0923 [11939.kaon2.fnal.gov] was RUNNING now DONE
501722 twoPoint_0.0923 [11939.kaon2.fnal.gov] DONE (status: 0) 47528 sec
501722 completed successfully 47588 s

```