All the way through demonstration

The purpose is to demonstrate an actual LQCD project execution. This purpose could and was met with a small example normally used to test cluster software configuration. This demonstration shows that batch scripts and inputs to the MILC application SU3_rmd can be generated where the various parameters, batch script templates and MILC input templates can be stored in a relational database.

Minor cosmetic edits have been done to reduce line width in this document.

A table of contents

1. Brief description of processing, page 2
2. Initiation, page 3

   How the job is initiated.
3. The shell environment parameter set, page 3

   This is evidence_environ(17). Its name is derived from the "driver" script parameters.
4. The generated shell environment script, page 3

   A source script generated by the "driver" script to inject the job environment parameters into the "driver" script.
5. The Project Parameter Set, page 3

   This is evidence_params(4). Its name is derived from evidence_environ(17), page 3, parameter set values. This provides the job parameters.
6. The qsub script parameter set, page 4

   This is evidence_script(13). Its name is derived from evidence_environ(17), page 3, parameter set values. This provides the template for the qsub submission.
7. The generated qsub script, page 5

   The qsub script generated by and submitted by the "driver" script.
8. MILC input template, page 6

   This is evidence_input(4). Its name is derived from evidence_environ(17), page 3, parameter set values. This provides the template for the MILC job input.
9. A generated MILC input, with nS being 4, page 6

   In this example, three MILC jobs are executed for varying lattice sizes. This one of the generated MILC job inputs.
10. "driver", page 7

    Used in the initiation step by the user.
11. the main source file of the pandora gem, page 9

    This provides several service routines using in the templates.
12. The MILC output, page 12

    For the curious, a MILC job output for one the MILC job inputs.

Brief description of processing

1. the following items were preloaded into the data base:
   (a) the parameters used by the application, evidence_params(4), page 3.
   (b) the script describing the shell script which generates the MPI jobs, evidence_script(13), page 4.
   (c) the template for the MILC input file, evidence_input(4), page 6.
   (d) the shell environment variable values parameter set, evidence_environ(17), page 3, used by the driver script used to drive the processing.

   The reason that this parameter set value has such a high version is that any change in one of the preceeding items has a tendency to change this item as well.

2. the driver script has not been placed in the data base; but, it could be.

   The driver script currently does not log the executions it manages. This is a change that perhaps should be made.

   It is very likely that the driver script would need changes to allow it to work effectively and efficently on the clusters and with the work flow management system. As an example, the driver script might have a third parameter of which cluster to which to submit the work and would tailor the work for that cluster.

3. the driver script has two arguments, name and version.
   (a) It avails from data base, the shell environment values, from the parameter set which supplies the project working directory and the versions of the other parameter sets used by the job, {name}_environ({environ_version}).
   (b) It changes to the project working directory (which needs to be in an area NFS served to the worker nodes, which do not have access to the "outside" world.
   (c) It creates a temporary workarea directory within the project working directory so that generated files are visible to the worker nodes. It also generates a parameter set containing the name of this temporary workarea directory.
   (d) It avails an unprocessed parameter set in FHiCL form describing the template MILC input into the temporary workarea directory within the project working directory, {name}_input({input_version}).
   (e) It avails an unprocessed parameter set in FHiCL form describing the project specific parameters into the temporary workarea directory within the project working directory, {name}_params({parameters_version}).
   (f) It avails a processed parameter set in FHiCL form describing the project generation script using the following inputs:
      i. the script parameter set template, {name}_script({script_version}),
      ii. the project parameter set {name}_params({parameters_version}),
      iii. the shell environment values {name}_environ({version}),
      iv. the parameter set containing the name of the temporary workarea directory

      The result of this processing is the script submitted by qsub.

4. the script submitted by qsub, which runs in a worker, uses only data embedded in the qsub script and files placed into the temporary work area directory and generates a set of MILC input decks, one for each in the project parameters sizes array using the input file, the parameters file and a generated file containing the current entry from the sizes array and submits those MILC input decks via the specified mpi run program.

Note that the script loops over the work units specified in the project parameter set and generates two separate, different work units with the MILC input in this case.

Note that the work units are generated by applying modifications to a default work unit description. One set of modifications is empty. The other set is not.

Initiation

How the script was executed, less shell prompt text.

```
$ driver evidence 17
1259548.jpsi2.fnal.gov
$ qstat | grep herber
1259548.jpsi2     script      herber       0 R test_jpsi
```

The shell environment parameter set

This parameter set, evidence_environ(17), selects the other parameter sets used.

```
home: "~herber/toy",
input: 4,
parameters: 4,
prefix: "evidence",
script: 13
```

The generated shell environment script

Part of the execution of the driver shell script is the injection of shell environment variables into its environment specifying the various parameter sets used.

```
export home=~herber/toy
export prefix="evidence"
export parameters=4
export script=13
export input=4
```

The Project Parameter Set

This is parameter set, evidence_params(4).

```
executable_path: "./su3_rmd_qdp_mvapich",
expansion_factor: 1,
flavors: [2, 1],
home: "~herber/toy",
input_template: "evidence_input",
mpi_path: "/usr/local/mvapich/bin/mpirun_rsh_8",
```

```
pbs: {
  account: "lqcdadmin",
  destination: "test_jpsi",
  join: "oe",
  resources: {nodes: 1, walltime: "0:30:00"}
},
seed: 1234,
sizes: [4, 6, 8],
tag: "lprocs",
work_unit_default: {
  beta: 5.6,
  error_for_propagator: .125,
  error_per_site: .125,
  masses: [0.125, 0.125],
  max_cg_iterations: 100,
  max_cg_restarts: 5,
  microcanonical_time_step: 0.2,
  steps_per_trajectory: 5,
  traj_between_meas: 1,
  trajecs: 1,
  u0: 1.,
  warms: 0
},
work_units: [
  {},
  {
    error_for_propagator: 1.e-5,
    error_per_site: 1.e-5,
    max_cg_iterations: 300,
    microcanonical_time_step: 0.01
  }
]
```

The qsub script parameter set

This is parameter set, evidence_script(13).

```
$<: require 'pandora' >
#!/bin/bash
$< generate_PBS >
T=/tmp/$$
mkdir $T
trap 'rm -fr $T $temporary' 0 1 2 3 4 5 6 7 8 10 12 13 14 15
export temporary=$< Enquote(@temporary) >
export home=$< @home >
export prefix=$< Enquote(@prefix) >
export parameters=$< Enquote(@parameters) >
export script=$< Enquote(@script) >
export input=$< Enquote(@input) >

echo ===== uname -a =====
```

```
uname -a

cd $<@home>
for ns in $< @sizes.list >; do
echo "nS : `expr $ns \* $<@expansion_factor>`" > $T/nS
# the submitted script is saved in the home directory
mangle $temporary/input $temporary/parameters $T/nS \
  >in_1procs.$ns
date > out_1procs.$ns
ls -l ./su3_rmd_qdp_mvapich >> out_1procs.$ns
$<@mpi_path> -np 1 $<@executable_path> in_1procs.$ns \
  >> out_1procs.$ns
date >> out_1procs.$ns
echo Finished `date`
done
```

The generated qsub script

```
#!/bin/bash
#PBS -A lqcdadmin
#PBS -j oe
#PBS -q test_jpsi
#PBS -l nodes=1
#PBS -l walltime="0:30:00"
T=/tmp/$$
mkdir $T
trap 'rm -fr $T $temporary' 0 1 2 3 4 5 6 7 8 10 12 13 14 15
export temporary="/home/h_k/herber/toy/14069"
export home=~herber/toy
export prefix="evidence"
export parameters=4
export script=13
export input=4

echo ===== uname -a =====
uname -a

cd ~herber/toy
for ns in 4 6 8; do
echo "nS : `expr $ns \* 1`" > $T/nS
# the submitted script is saved in the home directory
mangle $temporary/input $temporary/parameters $T/nS \
  >in_1procs.$ns
date > out_1procs.$ns
ls -l ./su3_rmd_qdp_mvapich >> out_1procs.$ns
/usr/local/mvapich/bin/mpirun_rsh_8 -np 1 \
  ./su3_rmd_qdp_mvapich in_1procs.$ns >> out_1procs.$ns
date >> out_1procs.$ns
echo Finished `date`
done
```

MILC input template

This is the parameter set, evidence_input(4)

```
$<: require 'pandora' >
prompt 0
$< ol(@flavors,"nflavors",1) >
nx $<@nS>
ny $<@nS>
nz $<@nS>
nt $<@nS>
iseed $<@seed>
$<: @first = true >
$<: for @work_unit in @work_units >
$<: @values = Marshal.load(Marshal.dump(@work_unit_default)) >
$<: @values.merge!(@work_unit) >

warms $<@values.warms>
trajecs $<@values.trajecs>
traj_between_meas $<@values.traj_between_meas>
beta $<@values.beta>
$< ol(@values.masses,"mass",1) >
u0 $<@values.u0>
microcanonical_time_step $<@values.microcanonical_time_step>
steps_per_trajectory $<@values.steps_per_trajectory>
max_cg_iterations $<@values.max_cg_iterations>
max_cg_restarts $<@values.max_cg_restarts>
error_per_site $<@values.error_per_site>
error_for_propagator $<@values.error_for_propagator>
$<: if @first >
$<:   @first = false >
fresh
$<: else >
continue
$<: end >
forget
$<: end >
```

A generated MILC input, with nS being 4

```
prompt 0
nflavors1 2
nflavors2 1
nx 4
ny 4
nz 4
nt 4
iseed 1234

warms 0
```

```
trajecs 1
traj_between_meas 1
beta 5.6
mass1 0.125
mass2 0.125
u0 1.
microcanonical_time_step 0.2
steps_per_trajectory 5
max_cg_iterations 100
max_cg_restarts 5
error_per_site .125
error_for_propagator .125
fresh
forget

warms 0
trajecs 1
traj_between_meas 1
beta 5.6
mass1 0.125
mass2 0.125
u0 1.
microcanonical_time_step 0.01
steps_per_trajectory 5
max_cg_iterations 300
max_cg_restarts 5
error_per_site 1.e-5
error_for_propagator 1.e-5
continue
forget
```

driver

```
#!/bin/bash
# use ":" to suppress debugging output; use "" to receive it
debug=""
parameterSetServer="https://oink.fnal.gov:32766"
export T=/tmp/$$
trap 'rm -fr $T' 0 1 2 3 4 5 6 7 8 10 12 13 14 15
mkdir $T

if [ $# != 2 ]; then
  echo $(basename $0) jobset_name jobset_version
  exit 1
fi
name="$1"
version="$2"

cat - <<\EOF >$T/template
$<: require 'pandora' >
export home=$< @home >
export prefix=$< Enquote(@prefix) >
export parameters=$< Enquote(@parameters) >
```

```
export script=$< Enquote(@script) >
export input=$< Enquote(@input) >
EOF

mangle $T/template "db::${name}_environ($version)" >$T/environ

$debug echo ===== $T/environ =====
$debug cat $T/environ
$debug echo ==========

source $T/environ

if [ "$home" ]; then
  if cd $home; then
    :
  else
    echo Unable to change working directory to ${home}.
    exit 1
  fi
else
  echo Unable to determine file storage location.
  exit 1
fi

export temporary=$(/bin/pwd)/$$

$debug echo ===== temporary is $temporary =====

if [ "$temporary" ]; then
  if mkdir $home/$$; then
    echo "temporary : \"$temporary\"" >$temporary/temporary
    cp $T/environ $temporary
  else
    echo Unable to create temporary storage directory, ${temporary}.
    exit 1
  fi
fi

$debug echo ===== $temporary/temporary =====
$debug cat $temporary/temporary
$debug echo ==========

#
# The MILC input template and the parameter set has to availed
# at this time as the worker nodes do not have access to the
# outside world and the parameter set server
#

curl -sk \
  "$parameterSetServer/parameters/fetch/${prefix}_input($input)/plain" \
  >$temporary/input

$debug echo ===== $temporary/input =====
$debug cat $temporary/input
$debug echo ==========

curl -sk \
  "$parameterSetServer/parameters/fetch/${prefix}_params($parameters)/plain" \
```

```
  >$temporary/parameters

$debug echo ===== $temporary/parameters =====
$debug cat $temporary/parameters
$debug echo ==========

mangle "db::${prefix}_script($script)" \
       "db::${prefix}_params($parameters)" \
       "db::${name}_environ($version)" \
       "$temporary/temporary" \
       >$temporary/script

$debug echo ===== $temporary/script =====
$debug cat $temporary/script
$debug echo ==========

$debug grep '^#PBS ' $temporary/script
qsub $temporary/script
```

The main pandora gem source file

```
module Pandora

  # :stopdoc:
  LIBPATH = ::File.expand_path(::File.dirname(__FILE__)) + ::File::SEPARATOR
  PATH = ::File.dirname(LIBPATH) + ::File::SEPARATOR
  # :startdoc:

  # Returns the version string for the library.
  #
  def self.version
    @version ||= File.read(path('version.txt')).strip
  end

  # Returns the library path for the module. If any arguments are given,
  # they will be joined to the end of the libray path using
  # <tt>File.join</tt>.
  #
  def self.libpath( *args, &block )
    rv =  args.empty? ? LIBPATH : ::File.join(LIBPATH, args.flatten)
    if block
      begin
        $LOAD_PATH.unshift LIBPATH
        rv = block.call
      ensure
        $LOAD_PATH.shift
      end
    end
    return rv
  end

  # Returns the lpath for the module. If any arguments are given,
  # they will be joined to the end of the path using
  # <tt>File.join</tt>.
  #
```

```
  def self.path( *args, &block )
    rv = args.empty? ? PATH : ::File.join(PATH, args.flatten)
    if block
      begin
        $LOAD_PATH.unshift PATH
        rv = block.call
      ensure
        $LOAD_PATH.shift
      end
    end
    return rv
  end

  # Utility method used to require all files ending in .rb that lie in the
  # directory below this file that has the same name as the filename passed
  # in. Optionally, a specific _directory_ name can be passed in such that
  # the _filename_ does not have to be equivalent to the directory.
  #
  def self.require_all_libs_relative_to( fname, dir = nil )
    dir ||= ::File.basename(fname, '.*')
    search_me = ::File.expand_path(
        ::File.join(::File.dirname(fname), dir, '**', '*.rb'))

    Dir.glob(search_me).sort.each {|rb| require rb}
  end

end  # module Pandora

Pandora.require_all_libs_relative_to(__FILE__)

def isNumeric(string)
  begin
    Kernel.Float(string)
  rescue
    return false
  end
  return true
end
def enquote(value)
  if value.nil? then "" else '"'+value.to_s.gsub(/["\\]/,'\\\\\&')+'"' end
end
def Enquote(value)
  isNumeric(value) ? value : enquote(value)
end
def ol(data,tag,start)
  return_value = ""
  for value in data
    return_value << tag << start.to_s << " " << Enquote(value) << "\n"
    start += 1
  end
  return_value
end
def generate_PBS
  return_value = ""
  if @pbs
    if @pbs.has_key?("after")
      return_value << "#PBS -a " << @pbs["after"].to_s << "\n"
    end
```

```
    if @pbs.has_key?("account")
      return_value << "#PBS -A " << @pbs["account"].to_s << "\n"
    end
    if @pbs.has_key?("wait_for_server")
      return_value << "#PBS -b " << @pbs["wait_for_server"].to_s << "\n"
    end
    if @pbs.has_key?("checkpoint")
      return_value << "#PBS -c " << @pbs["checkpoint"].to_s << "\n"
    end
    if @pbs.has_key?("directive_prefix")
      return_value << "#PBS -C " << @pbs["directive_prefix"].to_s << "\n"
    end
    if @pbs.has_key?("working_directory")
      return_value << "#PBS -d " << @pbs["working_directory"].to_s << "\n"
    end
    if @pbs.has_key?("root_directory")
      return_value << "#PBS -D " << @pbs["root_directory"].to_s << "\n"
    end
    if @pbs.has_key?("error_stream")
      return_value << "#PBS -e " << @pbs["error_stream"].to_s << "\n"
    end
    if @pbs.has_key?("hold")
      return_value << "#PBS -h\n"
    end
    if @pbs.has_key?("join")
      return_value << "#PBS -j " << @pbs["join"].to_s << "\n"
    end
    if @pbs.has_key?("keep")
      return_value << "#PBS -k " << @pbs["keep"].to_s << "\n"
    end
    if @pbs.has_key?("mail")
      return_value << "#PBS -m " << @pbs["mail"].to_s << "\n"
    end
    if @pbs.has_key?("mail_to")
      return_value << "#PBS -M " << @pbs["mail_to"].to_s << "\n"
    end
    if @pbs.has_key?("jobname")
      return_value << "#PBS -N " << @pbs["jobname"].to_s << "\n"
    end
    if @pbs.has_key?("output")
      return_value << "#PBS -o " << @pbs["output"].to_s << "\n"
    end
    if @pbs.has_key?("priority ")
      return_value << "#PBS -p " << @pbs["priority"].to_s << "\n"
    end
    if @pbs.has_key?("destination")
      return_value << "#PBS -q " << @pbs["destination"].to_s << "\n"
    end
    if @pbs.has_key?("rerunable")
      return_value << "#PBS -r " << @pbs["rerunable"].to_s << "\n"
    end
    if @pbs.has_key?("shell")
      return_value << "#PBS -S " << @pbs["shell"].to_s << "\n"
    end
    if @pbs.has_key?("job_array")
      return_value << "#PBS -t " << @pbs["job_array"].to_s << "\n"
    end
    if @pbs.has_key?("wrapper_name")
```

```
        return_value << "#PBS -T " << @pbs["wrapper_name"].to_s << "\n"
      end
      if @pbs.has_key?("user_list")
        return_value << "#PBS -u " << @pbs["user_list"].to_s << "\n"
      end
      if @pbs.has_key?("export_list")
        return_value << "#PBS -v " << @pbs["export_list"].to_s << "\n"
      end
      if @pbs.has_key?("export_all")
        return_value << "#PBS -v\n"
      end
      if @pbs.has_key?("X11_forwarding")
        return_value << "#PBS -X\n"
      end
      if @pbs.has_key?("silent")
        return_value << "#PBS -z\n"
      end
      if  @pbs.has_key?("resources")
        @pbs["resources"].each{|key,value|
          return_value << "#PBS -l " << key.to_s << "=" << Enquote(value) << "\n"
        }
      end
    end
    return_value
  end
```

The MILC output

```
Fri Oct 22 11:44:23 CDT 2010
-rwxr-xr-x  1 herber cdf 3313866 Oct 18 20:03 ./su3_rmd_qdp_mvapich
SU3 with improved KS action
Microcanonical simulation with refreshing
MIMD version 7 $Name:  $
Machine = QMP (portable), with 1 nodes
R algorithm
Options selected...
Generic single precision
C_GLOBAL_INLINE
SSE_GLOBAL_INLINE
DBLSTORE_FN
D_FN_GATHER13
FEWSUMS
KS_MULTIFF=FNMAT
VECLENGTH=4
type 0 for no prompts  or 1 for prompts
nflavors1 2
nflavors2 1
nx 4
ny 4
nz 4
nt 4
iseed 1234
LAYOUT = Hypercubes, options = QDP
ON EACH NODE 4 x 4 x 4 x 4
Mallocing 0.252 MBytes per node for lattice
Made lattice
Made nn gathers
Made 3nn gathers
Finished setup


warms 0
trajecs 1
traj_between_meas 1
beta 5.6
mass1 0.125
```

```
mass2 0.125
u0 1
microcanonical_time_step 0.2
steps_per_trajectory 5
max_cg_iterations 100
max_cg_restarts 5
error_per_site 0.125
error_for_propagator 0.125
fresh
forget
unit gauge configuration loaded
CHECK PLAQ: 3.000000e+00 3.000000e+00
CHECK NERSC LINKTR: 1.000000e+00 CKSUM: 0
Unitarity checked.  Max deviation 0.00e+00
Time to check unitarity = 1.389980e-04
Symanzik 1x1 + 1x2 + 1x1x1 action
gauge_action: total_dyn_flavors = 3
loop coefficients: nloop rep loop_coeff  multiplicity
                    0 0      1.000000e+00     6
                    1 0      -5.000000e-02     12
                    2 0      0.000000e+00     16
"O(a^2): couplings(pi)=0, Naik term, No O(a^2) errors, tadpole weights"
path coefficients: npath  path_coeff  multiplicity
                    0      6.250000e-01     8
                    1      -4.166667e-02     8
                    2      -6.250000e-02     48
                    3      1.562500e-02     192
                    4      -2.604167e-03     384
                    5      -6.250000e-02     48
WARMUPS COMPLETED
LLTIME(Fat): time = 7.045031e-03 (Asqtad opt) mflops = 2.239563e+03
LLTIME(long): time = 2.989769e-04 (Asqtad opt) mflops = 1.544681e+03
LLTIME(Fat): time = 6.981134e-03 (Asqtad opt) mflops = 2.260061e+03
LLTIME(long): time = 2.398491e-04 (Asqtad opt) mflops = 1.925477e+03
LLTIME(Fat): time = 6.917953e-03 (Asqtad opt) mflops = 2.280702e+03
LLTIME(long): time = 2.470016e-04 (Asqtad opt) mflops = 1.869720e+03
CONGRAD5: time = 3.969669e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.061938e+03
CONGRAD5: time = 3.139973e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.871015e+03
GFTIME:   time = 1.387215e-02 (Symanzik1) mflops = 1.784895e+03
GFREMAP:  time = 1.170635e-04
FFTIME:  time = 3.538299e-02 (qdp ASVEC) terms = 2 mflops = 3.139809e+03
LLTIME(Fat): time = 7.335186e-03 (Asqtad opt) mflops = 2.150974e+03
LLTIME(long): time = 2.589226e-04 (Asqtad opt) mflops = 1.783637e+03
LLTIME(Fat): time = 7.400036e-03 (Asqtad opt) mflops = 2.132124e+03
LLTIME(long): time = 2.591610e-04 (Asqtad opt) mflops = 1.781997e+03
LLTIME(Fat): time = 7.281065e-03 (Asqtad opt) mflops = 2.166962e+03
LLTIME(long): time = 2.310276e-04 (Asqtad opt) mflops = 1.998999e+03
CONGRAD5: time = 3.290176e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.694294e+03
CONGRAD5: time = 3.030300e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 4.011114e+03
GFTIME:   time = 1.379585e-02 (Symanzik1) mflops = 1.794765e+03
GFREMAP:  time = 1.180172e-04
FFTIME:  time = 3.515601e-02 (qdp ASVEC) terms = 2 mflops = 3.160080e+03
LLTIME(Fat): time = 7.111073e-03 (Asqtad opt) mflops = 2.218764e+03
LLTIME(long): time = 2.398491e-04 (Asqtad opt) mflops = 1.925477e+03
LLTIME(Fat): time = 7.072210e-03 (Asqtad opt) mflops = 2.230956e+03
LLTIME(long): time = 2.479553e-04 (Asqtad opt) mflops = 1.862529e+03
LLTIME(Fat): time = 7.141829e-03 (Asqtad opt) mflops = 2.209209e+03
LLTIME(long): time = 2.481937e-04 (Asqtad opt) mflops = 1.860740e+03
CONGRAD5: time = 3.309250e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.673002e+03
CONGRAD5: time = 3.101826e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.918621e+03
GFTIME:   time = 1.375699e-02 (Symanzik1) mflops = 1.799836e+03
GFREMAP:  time = 1.282692e-04
FFTIME:  time = 3.534913e-02 (qdp ASVEC) terms = 2 mflops = 3.142816e+03
LLTIME(Fat): time = 7.162094e-03 (Asqtad opt) mflops = 2.202958e+03
LLTIME(long): time = 2.570152e-04 (Asqtad opt) mflops = 1.796874e+03
LLTIME(Fat): time = 7.070065e-03 (Asqtad opt) mflops = 2.231633e+03
LLTIME(long): time = 2.849102e-04 (Asqtad opt) mflops = 1.620946e+03
LLTIME(Fat): time = 7.074118e-03 (Asqtad opt) mflops = 2.230355e+03
LLTIME(long): time = 2.379417e-04 (Asqtad opt) mflops = 1.940912e+03
CONGRAD5: time = 3.521442e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.451676e+03
CONGRAD5: time = 2.977848e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 4.081766e+03
GFTIME:   time = 1.370001e-02 (Symanzik1) mflops = 1.807322e+03
GFREMAP:  time = 1.187325e-04
FFTIME:  time = 3.521514e-02 (qdp ASVEC) terms = 2 mflops = 3.154774e+03
LLTIME(Fat): time = 6.934881e-03 (Asqtad opt) mflops = 2.275135e+03
LLTIME(long): time = 2.570152e-04 (Asqtad opt) mflops = 1.796874e+03
LLTIME(Fat): time = 7.174015e-03 (Asqtad opt) mflops = 2.199297e+03
LLTIME(long): time = 2.439022e-04 (Asqtad opt) mflops = 1.893480e+03
LLTIME(Fat): time = 7.146835e-03 (Asqtad opt) mflops = 2.207661e+03
```

```
LLTIME(long): time = 2.331734e-04 (Asqtad opt) mflops = 1.980604e+03
CONGRAD5: time = 3.349781e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.628560e+03
CONGRAD5: time = 3.240108e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.751381e+03
GFTIME:   time = 1.370716e-02 (Symanzik1) mflops = 1.806378e+03
GFREMAP:  time = 1.289845e-04
FFTIME:  time = 3.526402e-02 (qdp ASVEC) terms = 2 mflops = 3.150402e+03
PLAQ:   2.509844        2.511637
P_LOOP: 1.456498e+00    1.194501e-02
G_LOOP:  0  0  4        2.502886e+00    ( 0 1 7 6  )
G_LOOP:  0  1  4        2.504996e+00    ( 0 2 7 5  )
G_LOOP:  0  2  4        2.515832e+00    ( 0 3 7 4  )
G_LOOP:  0  3  4        2.521651e+00    ( 1 2 6 5  )
G_LOOP:  0  4  4        2.531774e+00    ( 1 3 6 4  )
G_LOOP:  0  5  4        2.487305e+00    ( 2 3 5 4  )
G_LOOP:  1  0  6        2.146579e+00    ( 0 0 1 7 7 6  )
G_LOOP:  1  1  6        2.132736e+00    ( 0 0 2 7 7 5  )
G_LOOP:  1  2  6        2.178479e+00    ( 0 0 3 7 7 4  )
G_LOOP:  1  3  6        2.151570e+00    ( 1 1 0 6 6 7  )
G_LOOP:  1  4  6        2.156863e+00    ( 1 1 2 6 6 5  )
G_LOOP:  1  5  6        2.175868e+00    ( 1 1 3 6 6 4  )
G_LOOP:  1  6  6        2.159641e+00    ( 2 2 0 5 5 7  )
G_LOOP:  1  7  6        2.156407e+00    ( 2 2 1 5 5 6  )
G_LOOP:  1  8  6        2.087983e+00    ( 2 2 3 5 5 4  )
G_LOOP:  1  9  6        2.129029e+00    ( 3 3 0 4 4 7  )
G_LOOP:  1  10  6       2.181840e+00    ( 3 3 1 4 4 6  )
G_LOOP:  1  11  6       2.077079e+00    ( 3 3 2 4 4 5  )
G_LOOP:  2  0  6        2.152359e+00    ( 0 1 2 7 6 5  )
G_LOOP:  2  1  6        2.193425e+00    ( 0 1 5 7 6 2  )
G_LOOP:  2  2  6        2.132191e+00    ( 0 6 2 7 1 5  )
G_LOOP:  2  3  6        2.165951e+00    ( 0 6 5 7 1 2  )
G_LOOP:  2  4  6        2.154626e+00    ( 0 1 3 7 6 4  )
G_LOOP:  2  5  6        2.241443e+00    ( 0 1 4 7 6 3  )
G_LOOP:  2  6  6        2.173004e+00    ( 0 6 3 7 1 4  )
G_LOOP:  2  7  6        2.214450e+00    ( 0 6 4 7 1 3  )
G_LOOP:  2  8  6        2.160568e+00    ( 0 2 3 7 5 4  )
G_LOOP:  2  9  6        2.120590e+00    ( 0 2 4 7 5 3  )
G_LOOP:  2  10  6       2.151300e+00    ( 0 5 3 7 2 4  )
G_LOOP:  2  11  6       2.163890e+00    ( 0 5 4 7 2 3  )
G_LOOP:  2  12  6       2.150064e+00    ( 1 2 3 6 5 4  )
G_LOOP:  2  13  6       2.177677e+00    ( 1 2 4 6 5 3  )
G_LOOP:  2  14  6       2.116887e+00    ( 1 5 3 6 2 4  )
G_LOOP:  2  15  6       2.165419e+00    ( 1 5 4 6 2 3  )
GACTION: 2.422259e+00
LLTIME(Fat): time = 7.035971e-03 (Asqtad opt) mflops = 2.242447e+03
LLTIME(long): time = 2.450943e-04 (Asqtad opt) mflops = 1.884271e+03
CONGRAD5: time = 3.259182e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.729427e+03
CONGRAD5: time = 4.329681e-04 (fn_qdp F) masses = 1 iters = 5 mflops = 3.509173e+03
PBP: mass 1.250000e-01     1.008234e-01  2.756273e-01  2.487829e-04  7.759240e-03 ( 1 of 1 )
FACTION: mass = 1.250000e-01,  1.521820e+00 ( 1 of 1 )
CONGRAD5: time = 3.628731e-04 (fn_qdp F) masses = 1 iters = 4 mflops = 3.349623e+03
CONGRAD5: time = 4.010201e-04 (fn_qdp F) masses = 1 iters = 5 mflops = 3.788738e+03
PBP: mass 1.250000e-01     1.110369e-01  1.856585e-01  3.854217e-02  3.877672e-02 ( 1 of 1 )
FACTION: mass = 1.250000e-01,  1.392258e+00 ( 1 of 1 )
RUNNING COMPLETED
average cg iters for step= 8.000000e+00
Time = 4.000711e-01 seconds
total_iters = 116


warms 0
trajecs 1
traj_between_meas 1
beta 5.6
mass1 0.125
mass2 0.125
u0 1
microcanonical_time_step 0.01
steps_per_trajectory 5
max_cg_iterations 300
max_cg_restarts 5
error_per_site 1e-05
error_for_propagator 1e-05
continue
forget
CHECK PLAQ: 2.509844e+00 2.511637e+00
CHECK NERSC LINKTR: 6.441766e-01 CKSUM: 245a7e98
Unitarity checked.  Max deviation 2.38e-07
Time to check unitarity = 1.449585e-04
Symanzik 1x1 + 1x2 + 1x1x1 action
```

```
gauge_action: total_dyn_flavors = 3
loop coefficients: nloop rep loop_coeff  multiplicity
                   0 0      1.000000e+00     6
                   1 0      -5.000000e-02     12
                   2 0      0.000000e+00     16
"O(a^2): couplings(pi)=0, Naik term, No O(a^2) errors, tadpole weights"
path coefficients: npath  path_coeff  multiplicity
                   0      6.250000e-01     8
                   1      -4.166667e-02     8
                   2      -6.250000e-02     48
                   3      1.562500e-02     192
                   4      -2.604167e-03     384
                   5      -6.250000e-02     48
WARMUPS COMPLETED
LLTIME(Fat): time = 7.111073e-03 (Asqtad opt) mflops = 2.218764e+03
LLTIME(long): time = 2.319813e-04 (Asqtad opt) mflops = 1.990781e+03
LLTIME(Fat): time = 7.122993e-03 (Asqtad opt) mflops = 2.215051e+03
LLTIME(long): time = 2.410412e-04 (Asqtad opt) mflops = 1.915955e+03
LLTIME(Fat): time = 7.045984e-03 (Asqtad opt) mflops = 2.239260e+03
LLTIME(long): time = 2.501011e-04 (Asqtad opt) mflops = 1.846549e+03
CONGRAD5: time = 2.038956e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.172928e+03
CONGRAD5: time = 1.996994e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.260612e+03
GFTIME:   time = 1.378202e-02 (Symanzik1) mflops = 1.796566e+03
GFREMAP:  time = 1.268387e-04
FFTIME:  time = 3.529406e-02 (qdp ASVEC) terms = 2 mflops = 3.147720e+03
LLTIME(Fat): time = 7.012129e-03 (Asqtad opt) mflops = 2.250072e+03
LLTIME(long): time = 2.520084e-04 (Asqtad opt) mflops = 1.832574e+03
LLTIME(Fat): time = 7.206917e-03 (Asqtad opt) mflops = 2.189257e+03
LLTIME(long): time = 2.379417e-04 (Asqtad opt) mflops = 1.940912e+03
LLTIME(Fat): time = 7.144928e-03 (Asqtad opt) mflops = 2.208251e+03
LLTIME(long): time = 2.310276e-04 (Asqtad opt) mflops = 1.998999e+03
CONGRAD5: time = 2.090931e-03 (fn_qdp F) masses = 1 iters = 29 mflops = 4.214528e+03
CONGRAD5: time = 2.003908e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.245911e+03
GFTIME:   time = 1.361990e-02 (Symanzik1) mflops = 1.817952e+03
GFREMAP:  time = 1.249313e-04
FFTIME:  time = 3.522491e-02 (qdp ASVEC) terms = 2 mflops = 3.153899e+03
LLTIME(Fat): time = 6.984949e-03 (Asqtad opt) mflops = 2.258827e+03
LLTIME(long): time = 2.501011e-04 (Asqtad opt) mflops = 1.846549e+03
LLTIME(Fat): time = 7.112026e-03 (Asqtad opt) mflops = 2.218467e+03
LLTIME(long): time = 2.460480e-04 (Asqtad opt) mflops = 1.876967e+03
LLTIME(Fat): time = 7.344007e-03 (Asqtad opt) mflops = 2.148390e+03
LLTIME(long): time = 2.620220e-04 (Asqtad opt) mflops = 1.762539e+03
CONGRAD5: time = 2.022028e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.207863e+03
CONGRAD5: time = 2.066135e-03 (fn_qdp F) masses = 1 iters = 29 mflops = 4.265107e+03
GFTIME:   time = 1.376581e-02 (Symanzik1) mflops = 1.798682e+03
GFREMAP:  time = 1.277924e-04
FFTIME:  time = 3.529906e-02 (qdp ASVEC) terms = 2 mflops = 3.147274e+03
LLTIME(Fat): time = 7.027149e-03 (Asqtad opt) mflops = 2.245262e+03
LLTIME(long): time = 2.348423e-04 (Asqtad opt) mflops = 1.966528e+03
LLTIME(Fat): time = 7.151127e-03 (Asqtad opt) mflops = 2.206336e+03
LLTIME(long): time = 2.360344e-04 (Asqtad opt) mflops = 1.956596e+03
LLTIME(Fat): time = 7.328033e-03 (Asqtad opt) mflops = 2.153073e+03
LLTIME(long): time = 2.379417e-04 (Asqtad opt) mflops = 1.940912e+03
CONGRAD5: time = 2.011061e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.230810e+03
CONGRAD5: time = 2.055883e-03 (fn_qdp F) masses = 1 iters = 29 mflops = 4.286375e+03
GFTIME:   time = 1.368809e-02 (Symanzik1) mflops = 1.808896e+03
GFREMAP:  time = 1.296997e-04
FFTIME:  time = 3.534794e-02 (qdp ASVEC) terms = 2 mflops = 3.142922e+03
LLTIME(Fat): time = 7.043839e-03 (Asqtad opt) mflops = 2.239942e+03
LLTIME(long): time = 2.470016e-04 (Asqtad opt) mflops = 1.869720e+03
LLTIME(Fat): time = 7.184982e-03 (Asqtad opt) mflops = 2.195940e+03
LLTIME(long): time = 2.388954e-04 (Asqtad opt) mflops = 1.933164e+03
LLTIME(Fat): time = 7.173061e-03 (Asqtad opt) mflops = 2.199590e+03
LLTIME(long): time = 2.670288e-04 (Asqtad opt) mflops = 1.729491e+03
CONGRAD5: time = 2.084017e-03 (fn_qdp F) masses = 1 iters = 29 mflops = 4.228511e+03
CONGRAD5: time = 2.069950e-03 (fn_qdp F) masses = 1 iters = 29 mflops = 4.257247e+03
GFTIME:   time = 1.371884e-02 (Symanzik1) mflops = 1.804840e+03
GFREMAP:  time = 1.440048e-04
FFTIME:  time = 3.535008e-02 (qdp ASVEC) terms = 2 mflops = 3.142731e+03
PLAQ:   2.490742        2.491316
P_LOOP: 1.451587e+00    2.214976e-02
G_LOOP:  0  0  4        2.490684e+00    ( 0 1 7 6  )
G_LOOP:  0  1  4        2.485330e+00    ( 0 2 7 5  )
G_LOOP:  0  2  4        2.496663e+00    ( 0 3 7 4  )
G_LOOP:  0  3  4        2.496211e+00    ( 1 2 6 5  )
G_LOOP:  0  4  4        2.508267e+00    ( 1 3 6 4  )
G_LOOP:  0  5  4        2.469018e+00    ( 2 3 5 4  )
G_LOOP:  1  0  6        2.132149e+00    ( 0 0 1 7 7 6  )
G_LOOP:  1  1  6        2.108180e+00    ( 0 0 2 7 7 5  )
```

```
G_LOOP:  1  2  6        2.157102e+00    ( 0 0 3 7 7 4  )
G_LOOP:  1  3  6        2.137895e+00    ( 1 1 0 6 6 7  )
G_LOOP:  1  4  6        2.110994e+00    ( 1 1 2 6 6 5  )
G_LOOP:  1  5  6        2.140435e+00    ( 1 1 3 6 6 4  )
G_LOOP:  1  6  6        2.139998e+00    ( 2 2 0 5 5 7  )
G_LOOP:  1  7  6        2.115979e+00    ( 2 2 1 5 5 6  )
G_LOOP:  1  8  6        2.071196e+00    ( 2 2 3 5 5 4  )
G_LOOP:  1  9  6        2.105433e+00    ( 3 3 0 4 4 7  )
G_LOOP:  1  10  6       2.152247e+00    ( 3 3 1 4 4 6  )
G_LOOP:  1  11  6       2.052770e+00    ( 3 3 2 4 4 5  )
G_LOOP:  2  0  6        2.131558e+00    ( 0 1 2 7 6 5  )
G_LOOP:  2  1  6        2.171789e+00    ( 0 1 5 7 6 2  )
G_LOOP:  2  2  6        2.109513e+00    ( 0 6 2 7 1 5  )
G_LOOP:  2  3  6        2.130251e+00    ( 0 6 5 7 1 2  )
G_LOOP:  2  4  6        2.129706e+00    ( 0 1 3 7 6 4  )
G_LOOP:  2  5  6        2.209257e+00    ( 0 1 4 7 6 3  )
G_LOOP:  2  6  6        2.135318e+00    ( 0 6 3 7 1 4  )
G_LOOP:  2  7  6        2.191343e+00    ( 0 6 4 7 1 3  )
G_LOOP:  2  8  6        2.132178e+00    ( 0 2 3 7 5 4  )
G_LOOP:  2  9  6        2.077263e+00    ( 0 2 4 7 5 3  )
G_LOOP:  2  10  6       2.125443e+00    ( 0 5 3 7 2 4  )
G_LOOP:  2  11  6       2.135004e+00    ( 0 5 4 7 2 3  )
G_LOOP:  2  12  6       2.111974e+00    ( 1 2 3 6 5 4  )
G_LOOP:  2  13  6       2.138512e+00    ( 1 2 4 6 5 3  )
G_LOOP:  2  14  6       2.084108e+00    ( 1 5 3 6 2 4  )
G_LOOP:  2  15  6       2.129911e+00    ( 1 5 4 6 2 3  )
GACTION: 2.525046e+00
LLTIME(Fat): time = 7.232904e-03 (Asqtad opt) mflops = 2.181391e+03
LLTIME(long): time = 2.310276e-04 (Asqtad opt) mflops = 1.998999e+03
CONGRAD5: time = 2.027035e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.197469e+03
CONGRAD5: time = 5.040169e-04 (fn_qdp F) masses = 1 iters = 6 mflops = 3.617403e+03
PBP: mass 1.250000e-01     1.078435e-01  1.604667e-01  9.419164e-03  9.417929e-03 ( 1 of 1 )
FACTION: mass = 1.250000e-01,  1.568152e+00 ( 1 of 1 )
CONGRAD5: time = 2.028942e-03 (fn_qdp F) masses = 1 iters = 28 mflops = 4.193523e+03
CONGRAD5: time = 4.990101e-04 (fn_qdp F) masses = 1 iters = 6 mflops = 3.653698e+03
PBP: mass 1.250000e-01     3.667666e-02  2.228681e-01  1.773735e-02  1.773602e-02 ( 1 of 1 )
FACTION: mass = 1.250000e-01,  1.502651e+00 ( 1 of 1 )
RUNNING COMPLETED
average cg iters for step= 5.700000e+01
Time = 4.207032e-01 seconds
total_iters = 822


get_i(0): EOF on input.
exit: Fri Oct 22 11:44:24 2010

Fri Oct 22 11:44:24 CDT 2010
```