HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Performing Parameter Sweeps on Odyssey

Plamen Krastev, PhD
FAS Research Computing
Harvard University
plamenkrastev@fas.harvard.edu

April 8, 2015

Goal

Use SLURM commands and the power of BASH shell scripts to perform effective parameter sweeps on Odyssey.

Goal

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Use SLURM commands and the power of BASH shell scripts to perform effective parameter sweeps on Odyssey.

Outline

- ❑ Introduction / Purpose
- ❑ Serial parameter sweeps
- ❑ Parallel parameter sweeps
  - Multiple jobs submission – Bash Shell Scripts
  - Multiple jobs submission – Job Arrays
  - Parallelization of serial applications
- ❑ Advanced topics – Submitting Multiple Jobs with Job Dependencies

Introduction / Purpose

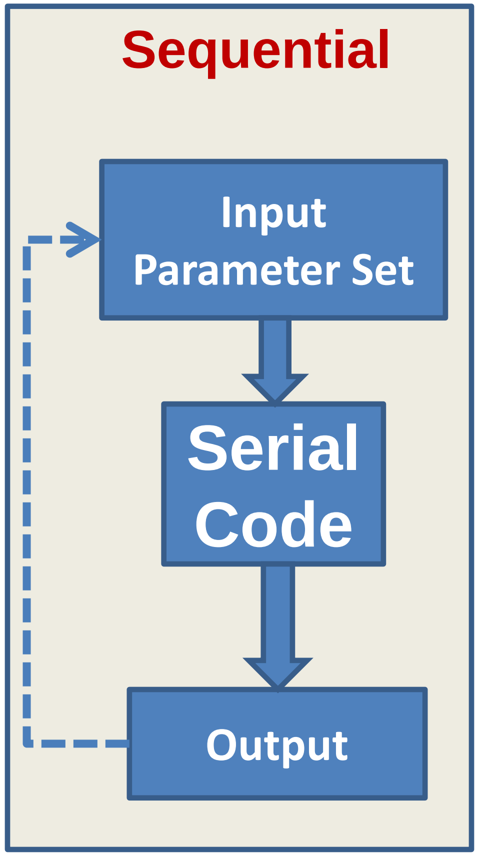
HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Parameter sweep applications are a class of applications in which the **same code is run multiple times using unique set of input parameter values**. This includes varying one parameter over a range of values, or varying multiple parameters over a large multidimensional space.

Introduction / Purpose

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

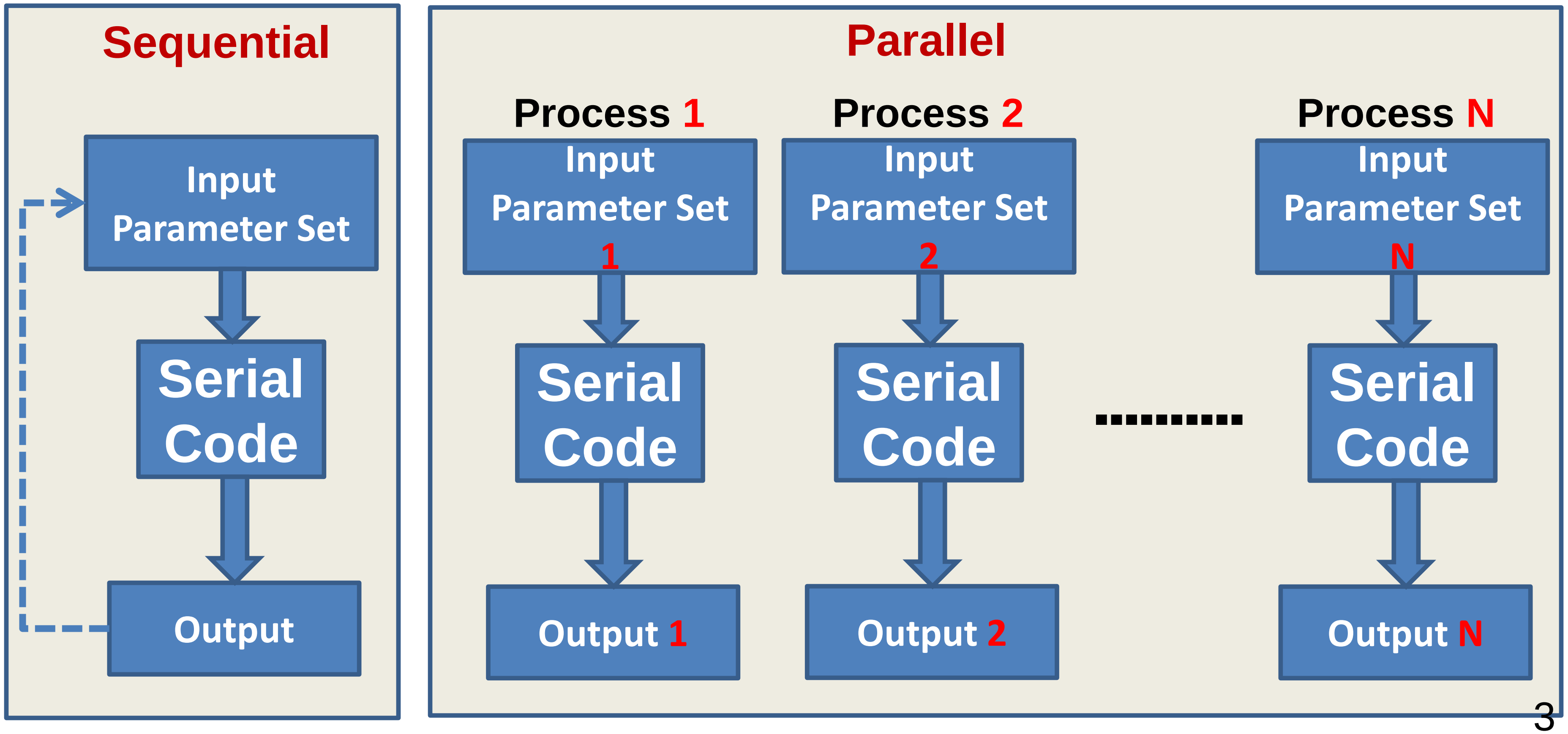
Parameter sweep applications are a class of applications in which the **same code is run multiple times using unique set of input parameter values**. This includes varying one parameter over a range of values, or varying multiple parameters over a large multidimensional space.



Introduction / Purpose

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Parameter sweep applications are a class of applications in which the **same code is run multiple times using unique set of input parameter values**. This includes varying one parameter over a range of values, or varying multiple parameters over a large multidimensional space.



Serial Parameter Sweeps

Example: Serial Matlab program computing the sum of integers from 1 to N for **N = [100, 200, 300]**. N is a parameter passed to the program from the Matlab command line.

Serial Parameter Sweeps

Example: Serial Matlab program computing the sum of integers from 1 to N for **N = [100, 200, 300]**. N is a parameter passed to the program from the Matlab command line.

```
%=========================================
% Function: serial_sum( N )
%           Calculates integer sum from 1 to N
%=========================================
function s = serial_sum(N)
  s = 0;
  for i = 1:N
    s = s + i;
  end
  fprintf('Sum of numbers from 1 to %d is %d.\n', N, s);
end
```

Serial Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

serial_sum.sh: Bash shell script for running the serial code for each value of the parameter N (100, 200, and 300). Runs are performed in serial.

```
#!/bin/bash
parameter_list=( 100 200 300 )
i=0
for N in "${parameter_list[@]}"
do
    i=$(($i+1))
    echo "Iteration: $i"
    echo "Parameter value: $N"
    echo "Starting Matlab ..."
    matlab -nodesktop -nosplash -r "serial_sum($N);exit"
    echo "“
done
```

Serial Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

serial_sum.run: Batch-job submission script to submit the serial parameter sweep job.

```
#!/bin/bash
#SBATCH -J serial_sum
#SBATCH -o serial_sum.out
#SBATCH -e serial_sum.err
#SBATCH -p general
#SBATCH -n 1
#SBATCH -t 30
#SBATCH --mem=2000
sh serial_sum.sh
```

Serial Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

serial_sum.run: Batch-job submission script to submit the serial parameter sweep job.

```
#!/bin/bash
#SBATCH -J serial_sum
#SBATCH -o serial_sum.out
#SBATCH -e serial_sum.err
#SBATCH -p general
#SBATCH -n 1
#SBATCH -t 30
#SBATCH --mem=2000
sh serial_sum.sh
```

Use

```
sbatch serial_sum.run
```

to submit the job to the queue.

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Submitting multiple jobs with the help of Bash shell scripts: Jobs are submitted simultaneously with separate parameter sets.

```
#!/bin/bash
parameter_list=( 100 200 300 )
i=0
job_name="parallel_sum"
for N in "${parameter_list[@]}"
do
    i=$(($i+1))
    echo "Submitting job $i ..."
    echo "Parameter value: $N"
    sbatch -J ${job_name}_${i}     \
           -o ${job_name}_${i}.out \
           -e ${job_name}_${i}.err \
           -p general              \
           -n 1                    \
           -t 30                   \
           --mem=2000              \
    --wrap="matlab -nodesktop -nosplash -r 'serial_sum($N);exit'"
    echo ""
    sleep 1
done
```

Parallel Parameter Sweeps

Submitting multiple jobs with the help of Job Arrays: Jobs are submitted simultaneously with different parameter sets. Parameter sets are read from separate input files for each job instance.

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

<u>Submitting multiple jobs with the help of Job Arrays</u>: Jobs are submitted simultaneously with different parameter sets. Parameter sets are read from separate input files for each job instance.

```
%=======================================
% Function: serial_sum( infile )
%          Calculates integer sum from 1 to N
%=======================================
function s = serial_sum(infile)
  fid = fopen(infile,'r');
  m = fgets(fid);
  fclose(fid);
  N = sscanf(m,'%d');
  s = 0;
  for i = 1:N
    s = s + i;
  end
  fprintf('Sum of numbers from 1 to %d is %d.\n', N, s);
end
```

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Submitting multiple jobs with the help of Job Arrays: Jobs are submitted simultaneously with different parameter sets. Parameter sets are read from separate input files for each job instance.

Batch-job submission script for the array job

```
#!/bin/bash
#SBATCH -J parallel_sum
#SBATCH -o output_%a.out
#SBATCH -e error_%a.err
#SBATCH -p general
#SBATCH -n 1
#SBATCH -t 30
#SBATCH --mem=2000
#SBATCH --array=1-3
matlab -nodesktop -nosplash -r "serial_sum('input_${SLURM_ARRAY_TASK_ID}');exit"
```

Parallel Parameter Sweeps

Parallelization of serial applications: A typical parameter sweep application consists of a for loop which repeatedly executes the same code, usually in a function. A unique set of arguments is supplied to the function in each iteration.

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Parallelization of serial applications: A typical parameter sweep application consists of a for loop which repeatedly executes the same code, usually in a function. A unique set of arguments is supplied to the function in each iteration.

spar.m: serial Matlab script

```
%=========================================
% Matlab code for calling the function serial_sum
% with different parameter values (100, 200, 300)
%=========================================
parameter_list = [100, 200, 300];
N = 3;
for i = 1: N
   serial_sum(parameter_list(i));
end
exit;
```

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Parallelization of serial applications: A typical parameter sweep application consists of a for loop which repeatedly executes the same code, usually in a function. A unique set of arguments is supplied to the function in each iteration.

```
#!/bin/bash
#SBATCH -J serial
#SBATCH -o serial.out
#SBATCH -e serial.err
#SBATCH -p general
#SBATCH -n 1
#SBATCH -t 30
#SBATCH --mem=2000
matlab -nosplash -nodesktop -r "spar"
```

Parallel Parameter Sweeps

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Parallelization of serial applications: A typical parameter sweep application consists of a for loop which repeatedly executes the same code, usually in a function. A unique set of arguments is supplied to the function in each iteration.

ppar.m: parallel Matlab script using 3 Matlab workers with PCT

```
%======================================
% Parallel Matlab code for calling the function serial_sum
% with different parameter values (100, 200, 300)
%======================================
% create a local cluster object
pc = parcluster('local')
parpool(pc, 3)
parameter_list = [100, 200, 300];
N = 3;
parfor i = 1: N
  serial_sum(parameter_list(i));
end
delete(gcp);
exit;
```

Parallel Parameter Sweeps

Parallelization of serial applications: A typical parameter sweep application consists of a for loop which repeatedly executes the same code, usually in a function. A unique set of arguments is supplied to the function in each iteration.

```
#!/bin/bash
#SBATCH -J parallel
#SBATCH -o parallel.out
#SBATCH -e parallel.err
#SBATCH -p general
#SBATCH -n 3
#SBATCH -N 1
#SBATCH -t 30
#SBATCH --mem=2000
matlab-default -nosplash -nodesktop -r "ppar"
```

Job Dependencies

HARVARD Faculty of Arts and Sciences RESEARCH COMPUTING

Example: Submitting three jobs (step_1, step_2, step_3). Subsequent job starts if only the previous job has completed successfully.

```
#!/bin/bash
n_steps=3
for i in `seq 1 ${n_steps}`;
do
    echo "Iteration: $i"
    command="sbatch --dependency=afterok:$latest_id step_${i}.run"
    if [ $i = 1 ]; then
        command="sbatch step_${i}.run"
    fi
    latest_id=$($command | awk ' { print $4 }')
    echo "Successfully submitted job $latest_id."
done
```

Job Dependencies

HARVARD
Faculty of Arts and Sciences
RESEARCH COMPUTING

Example: Submitting three jobs (step_1, step_2, step_3). Subsequent job starts if only the previous job has completed successfully.

```
#!/bin/bash
n_steps=3
for i in `seq 1 ${n_steps}`;
do
    echo "Iteration: $i"
    command="sbatch --dependency=afterok:$latest_id step_${i}.run"
    if [ $i = 1 ]; then
        command="sbatch step_${i}.run"
    fi
    latest_id=$($command | awk ' { print $4 }')
    echo "Successfully submitted job $latest_id."
done
```

Get Job ID