Parallel programming in Stata

Brian Quistorff

Department of Economics
University of Maryland

Cluster Mini-talks, 2014

Easy Wins For Parallelism

Common situations where you need to do repeated work with no data dependencies between repetitions:

1. Calculations are per-observation (e.g. each observation is a set of parameters for which a model must be solved)
2. Calculations involve whole dataset but still independent (e.g. permutations and bootstraps).

Easy Wins For Parallelism

Common situations where you need to do repeated work with no data dependencies between repetitions:

1. Calculations are per-observation (e.g. each observation is a set of parameters for which a model must be solved)
2. Calculations involve whole dataset but still independent (e.g. permutations and bootstraps).

Easy Wins For Parallelism

Common situations where you need to do repeated work with no data dependencies between repetitions:

1. Calculations are per-observation (e.g. each observation is a set of parameters for which a model must be solved)
2. Calculations involve whole dataset but still independent (e.g. permutations and bootstraps).

Can't I just use Stata-MP?

- Stata-MP (which is on the cluster) has coded some some commands to take advantage of multiple processors but doesn't allow explicit parallelization by the programer.
- See which commands can be sped up: `http://www.stata.com/statamp/statamp.pdf`

Can't I just use Stata-MP?

- Stata-MP (which is on the cluster) has coded some some commands to take advantage of multiple processors but doesn't allow explicit parallelization by the programer.
- See which commands can be sped up: `http://www.stata.com/statamp/statamp.pdf`

Stata's -parallel- module

- Documentation:
  `http://fmwww.bc.edu/repec/bocode/p/parallel.pdf`
- Can run a .do file or command (I'll stick to commands here) in parallel.
- By default does parallelization type #1 well. Breaks up dataset passes off to workers and then aggregates the results.
- I'll show how to do type #2 tasks.

Stata's -parallel- module

- Documentation:
  `http://fmwww.bc.edu/repec/bocode/p/parallel.pdf`
- Can run a .do file or command (I'll stick to commands here) in parallel.
- By default does parallelization type #1 well. Breaks up dataset passes off to workers and then aggregates the results.
- I'll show how to do type #2 tasks.

Stata's -parallel- module

- Documentation:
  `http://fmwww.bc.edu/repec/bocode/p/parallel.pdf`
- Can run a .do file or command (I'll stick to commands here) in parallel.
- By default does parallelization type #1 well. Breaks up dataset passes off to workers and then aggregates the results.
- I'll show how to do type #2 tasks.

Stata's -parallel- module

- Documentation:
  `http://fmwww.bc.edu/repec/bocode/p/parallel.pdf`
- Can run a .do file or command (I'll stick to commands here) in parallel.
- By default does parallelization type #1 well. Breaks up dataset passes off to workers and then aggregates the results.
- I'll show how to do type #2 tasks.

Basic coding guides

- Keep intermediate files around until sure computation worked (because debugging is harder than normal).
- If you need different parameters for different workers (# of reps, seed) use globals.
- Make switching between parallel and non-parallel both easy stylistically and result in the same output. For the latter, do just the random commands in serial and track what the seed would be at for each chunk. Otherwise, have to worry about problems of joint randomization in parallel RNGs of Stata (seeOzier - Perils of simulation).
- (see code)

Basic coding guides

- Keep intermediate files around until sure computation worked (because debugging is harder than normal).
- If you need different parameters for different workers (# of reps, seed) use globals.
- Make switching between parallel and non-parallel both easy stylistically and result in the same output. For the latter, do just the random commands in serial and track what the seed would be at for each chunk. Otherwise, have to worry about problems of joint randomization in parallel RNGs of Stata (seeOzier - Perils of simulation).
- (see code)

Basic coding guides

- Keep intermediate files around until sure computation worked (because debugging is harder than normal).
- If you need different parameters for different workers (# of reps, seed) use globals.
- Make switching between parallel and non-parallel both easy stylistically and result in the same output. For the latter, do just the random commands in serial and track what the seed would be at for each chunk. Otherwise, have to worry about problems of joint randomization in parallel RNGs of Stata (seeOzier - Perils of simulation).

- (see code)

Basic coding guides

- Keep intermediate files around until sure computation worked (because debugging is harder than normal).
- If you need different parameters for different workers (# of reps, seed) use globals.
- Make switching between parallel and non-parallel both easy stylistically and result in the same output. For the latter, do just the random commands in serial and track what the seed would be at for each chunk. Otherwise, have to worry about problems of joint randomization in parallel RNGs of Stata (seeOzier - Perils of simulation).

- (see code)

Debugging

- Check the intermediate _-pll${pid}_do${pll_instance}.log along with your own log
- Use -set trace on- (and tracedepth)
- Always test with smaller reps/dataset before running big.

Debugging

- Check the intermediate _-pll${pid}_do${pll_instance}.log along with your own log
- Use -set trace on- (and tracedepth)
- Always test with smaller reps/dataset before running big.

Debugging

- Check the intermediate _pll${pid}_do${pll_instance}.log along with your own log
- Use -set trace on- (and tracedepth)
- Always test with smaller reps/dataset before running big.

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-

Other notes

- -parallel- doesn't copy settings such as adopath and matsize.
- -parallel- doesn't copy scalars or matrices (so use globals or mata objects).
- -parallel- has the same "instance id" for a computer (so can't run two parallel on same machine (or I think even across cluster machines) without resetting seed)
- -parallel- doesn't work in batch mode for windows
- Copied mata objects will move after -parallel- so earlier pointers will be incorrect.
- The [,] is required for -parallel subcommand-