Report to Users

Alan Riley

Vice President, Software Development
StataCorp LP

2006 German Stata Users Group meeting, Mannheim, Germany

Outline

1. Stata Press
2. Stata 9
3. New development
   - Stata 9.1
   - Stata 9.2 - Mata structures
   - Stata 9.2 - work faster

Stata Press

Most active year ever

- Stata Journal indexed
- Two revised editions of existing books
- Four new books published
- Seven books in progress

Stata Press

Stata Journal

- 6th year of publication
- Special edition - Stata 20th anniversary
- Now indexed

Thomson Scientific citation indexes

- Science Citation Index Expanded
- CompuMath Citation index

Stata Press

Stata Journal

- 6th year of publication
- Special edition - Stata 20th anniversary
- Now indexed

Thomson Scientific citation indexes

- Science Citation Index Expanded
- CompuMath Citation index

More than doubled number of books published

Revised editions, 2005

- **Regression Models for Categorial Dependent Variables Using Stata, 2nd Edition**
  by J. Scott Long, Jeremy Freese

- **Maximum Likelihood Estimation with Stata, 3rd Edition**
  by William Gould, Jeffrey Pitblado, William Sribney

More than doubled number of books published

Revised editions, 2005

- **Regression Models for Categorial Dependent Variables Using Stata, 2nd Edition**
  by J. Scott Long, Jeremy Freese

- **Maximum Likelihood Estimation with Stata, 3rd Edition**
  by William Gould, Jeffrey Pitblado, William Sribney

Stata Press

More than doubled number of books published

New books, 2005

- **Data Analysis With Stata**
  by Ulrich Kohler and Frauke Kreuter
- **Multilevel and Longitudinal Modeling Using Stata**
  by Sophia Rabe-Hesketh and Anders Skrondal
- **A Gentle Introduction to Stata**
  by Alan Acock
- **An Introduction to Stata for Health Researchers**
  by Svend Juul

Stata Press

Forthcoming books, 2006

- **An Introduction to Modern Econometrics Using Stata**
  by Christopher F. Baum
- **Generalized Linear Models and Extensions, 2nd Edition**
  by James Hardin, Joseph Hilbe
- **A Guide to Stochastic Frontier Models: Specification and Estimation**
  by Subal Kumbhakar, Hung-Jen Wang
- **An Introduction to Forecasting Time Series Using Stata**
  by Robert Yaffee
- **The 123s of Survey Statistics with Stata**
  by Nicholas Winter
- **Applied Microeconometrics Using Stata**
  by A. Colin Cameron, Pravin K. Trivedi

Stata Press

Forthcoming books, 2007

- **Data Management Using Stata**
  by Michael Mitchell

Stata 9

- Released April 2005
- 20th anniversary
- Largest release ever

Stata 9

Stata 1, January 1985

- 44 commands
- 175 pages of documentation

Stata 8, January 2003

- over 600 commands
- 4652 pages of documentation

Stata 9, April 2005

- over 700 commands including new matrix language Mata
- 6413 pages of documentation

Stata 9

Stata 1, January 1985

- 44 commands
- 175 pages of documentation

Stata 8, January 2003

- over 600 commands
- 4652 pages of documentation

Stata 9, April 2005

- over 700 commands including new matrix language Mata
- 6413 pages of documentation

Stata 9

Stata 1, January 1985

- 44 commands
- 175 pages of documentation

Stata 8, January 2003

- over 600 commands
- 4652 pages of documentation

Stata 9, April 2005

- over 700 commands including new matrix language Mata
- 6413 pages of documentation

Stata 9

Ongoing development

- Continued release-as-we-go strategy
- Stata 9.1
- Stata 9.2
  - Mata structures
  - Work faster

Stata 9.1

- Multiple log files
- Faster survey linearization
- More stored estimation results
- New Mata functions (permutation, string, regular expression, binary I/O)
- Sized PNG and TIFF exported graphs
- `adoupdate`
- And more...

Mata structures

Set of variables tied together under a single name

```
struct structname {
        declaration(s)
}
```

Example

```
struct mystruct {
        real scalar    n1, n2
        real matrix    x
}
```

Mata structures

Set of variables tied together under a single name

```
struct structname {
        declaration(s)
}
```

Example

```
struct mystruct {
        real scalar    n1, n2
        real matrix    x
}
```

Mata structures

```
struct myresult {
        real scalar    yoverx
        real scalar    xovery
}

struct myresult scalar myfunc(real scalar x, real scalar y)
{
        struct myresult scalar    res

        res.yoverx = y/x
        res.xovery = x/y

        return(res)
}

...
struct myresult scalar results
...

results = myfunc(3, 4)
```

Mata structures

You can have vectors and matrices of structures

```
struct mystruct scalar     t
struct mystruct vector     t
struct mystruct rowvector  t
struct mystruct colvector  t
struct mystruct matrix     t

t[2,3].n1
```

Structures can contain vectors and matrices

```
t[2,3].x[9,2]
```

Mata structures

You can have vectors and matrices of structures

```
struct mystruct scalar     t
struct mystruct vector     t
struct mystruct rowvector  t
struct mystruct colvector  t
struct mystruct matrix     t

t[2,3].n1
```

Structures can contain vectors and matrices

```
t[2,3].x[9,2]
```

Mata structures

Structures can contain other structures

```
struct myresult {
        real scalar    yoverx
        real scalar    xovery
}

struct someresults {
        struct myresult scalar  res1, res2
}

...
struct someresults scalar myres
...

myres.res1 = myfunc(3, 4)
myres.res2 = myfunc(5, 6)
```

Mata structures

Advantages of structures

- Organization
- Convenience (return multiple results)
- Abstraction (handles)

Stata 9.2

Moore's Law

- Computer processing power doubles every 18 months
- Max transistors per chip has doubled every 24 months
- To maintain, industry must improve at rate of **1% per week**

Stata/MP

Work faster – work in parallel

- new 'flavor' of Stata capable of performing symmetric multiprocessing (SMP)
- same capabilities as Stata/SE, but faster due to parallelization of central routines
- for dual core, multicore, or multiprocessor computers
- `http://www.stata.com/statamp/`

Difference between 'processor' and 'core'

- processor: central processing unit, or CPU
- core: computation engine of a CPU with integer and floating point processing units

Stata/MP

Work faster – work in parallel

- new 'flavor' of Stata capable of performing symmetric multiprocessing (SMP)
- same capabilities as Stata/SE, but faster due to parallelization of central routines
- for dual core, multicore, or multiprocessor computers
- `http://www.stata.com/statamp/`

Difference between 'processor' and 'core'

- processor: central processing unit, or CPU
- core: computation engine of a CPU with integer and floating point processing units

Stata/MP

Design requirements

- 100% compatible with Stata/SE, Intercooled Stata, and Small Stata
- No end-user programming necessary to obtain speed ups
- No changes necessary to do-files, user-written programs, or datasets
- Priority given to estimation commands

Stata/MP

Supports 2 to 32 processors or cores on

- Macintosh OSX (Intel)
- 32-bit Windows
- 64-bit Windows (x86-64)
- 64-bit Windows (Itanium)
- 32-bit Linux
- 64-bit Linux (x86-64)
- 64-bit Linux (Itanium)
- 64-bit Solaris (Sparc)

Stata/MP

Perfection, in theory

- 100% efficiency is twice as fast on 2 processors/cores
- Speed doubles for every doubling of number of processors
- Execution time halves for every doubling of number of processors

Amdahl's Law

F: sequential/non-parallelizable fraction

N: number of processors

Maximum speed up: $\frac{1}{F+\frac{1-F}{N}}$

Stata/MP

Perfection, in theory

- 100% efficiency is twice as fast on 2 processors/cores
- Speed doubles for every doubling of number of processors
- Execution time halves for every doubling of number of processors

Amdahl's Law

F: sequential/non-parallelizable fraction

N: number of processors

Maximum speed up: $\frac{1}{F+\frac{1-F}{N}}$

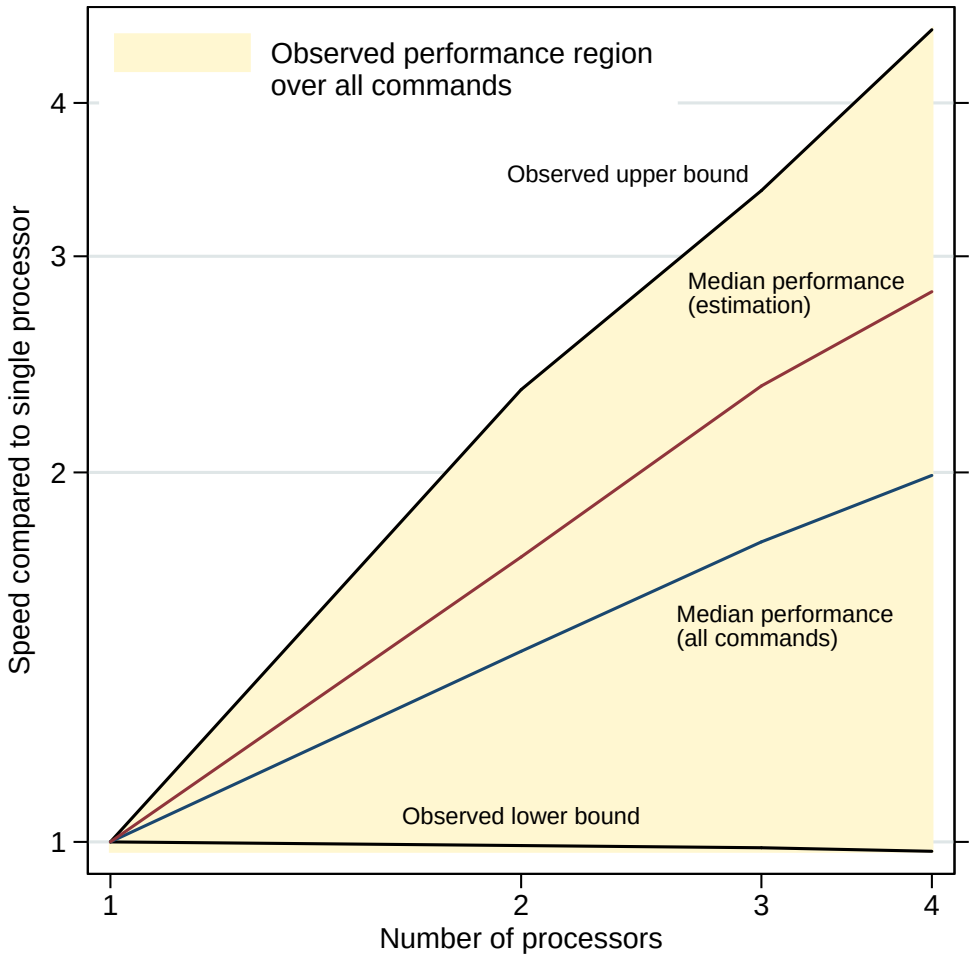
How much faster?

- Median speed up (overall)
  - 72% efficiency
  - 2 CPUs: 1.4
  - 3 CPUs: 1.75
  - 4 CPUs: 2.0
- Median speed up (estimation comands)
  - 88% efficiency
  - 2 CPUs: 1.7
  - 3 CPUs: 2.3
  - 4 CPUs: 2.8

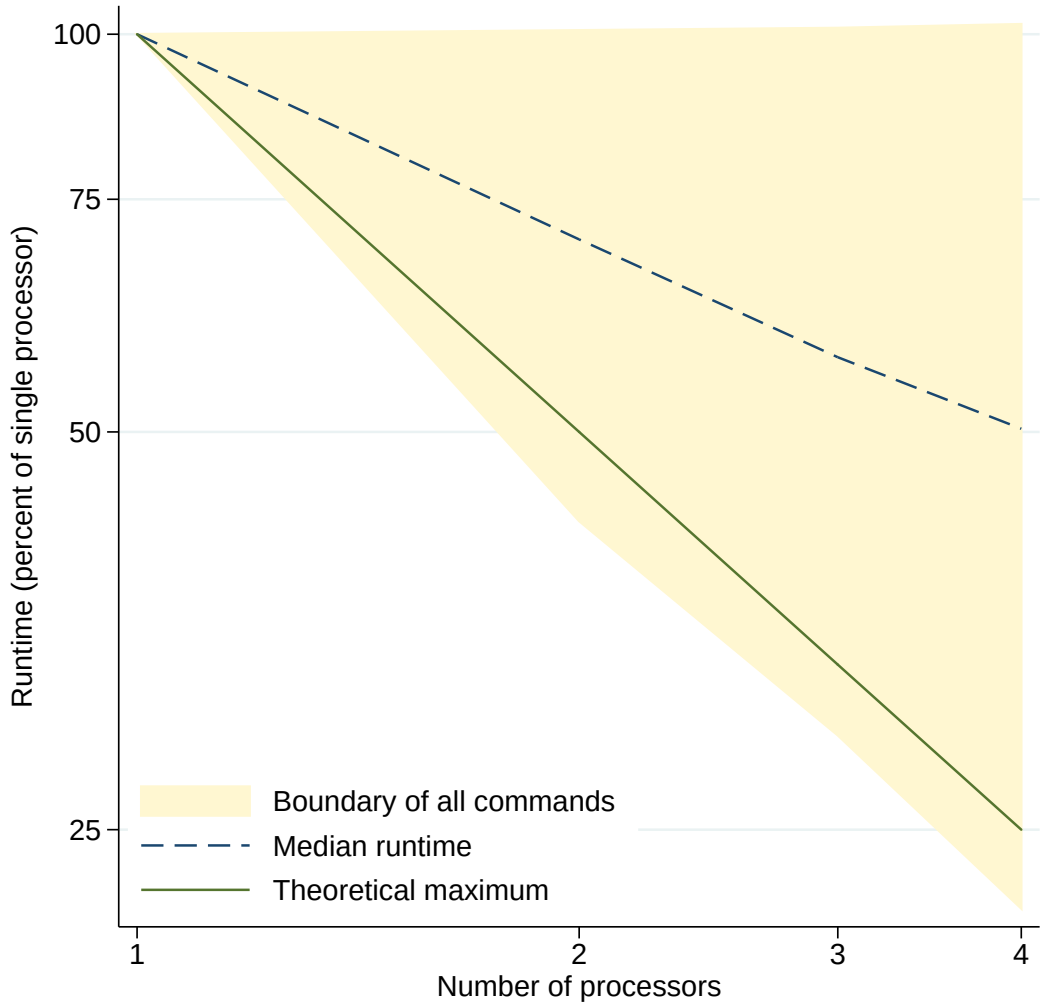
How much faster?

- Median speed up (overall)
  - 72% efficiency
  - 2 CPUs: 1.4
  - 3 CPUs: 1.75
  - 4 CPUs: 2.0
- Median speed up (estimation comands)
  - 88% efficiency
  - 2 CPUs: 1.7
  - 3 CPUs: 2.3
  - 4 CPUs: 2.8

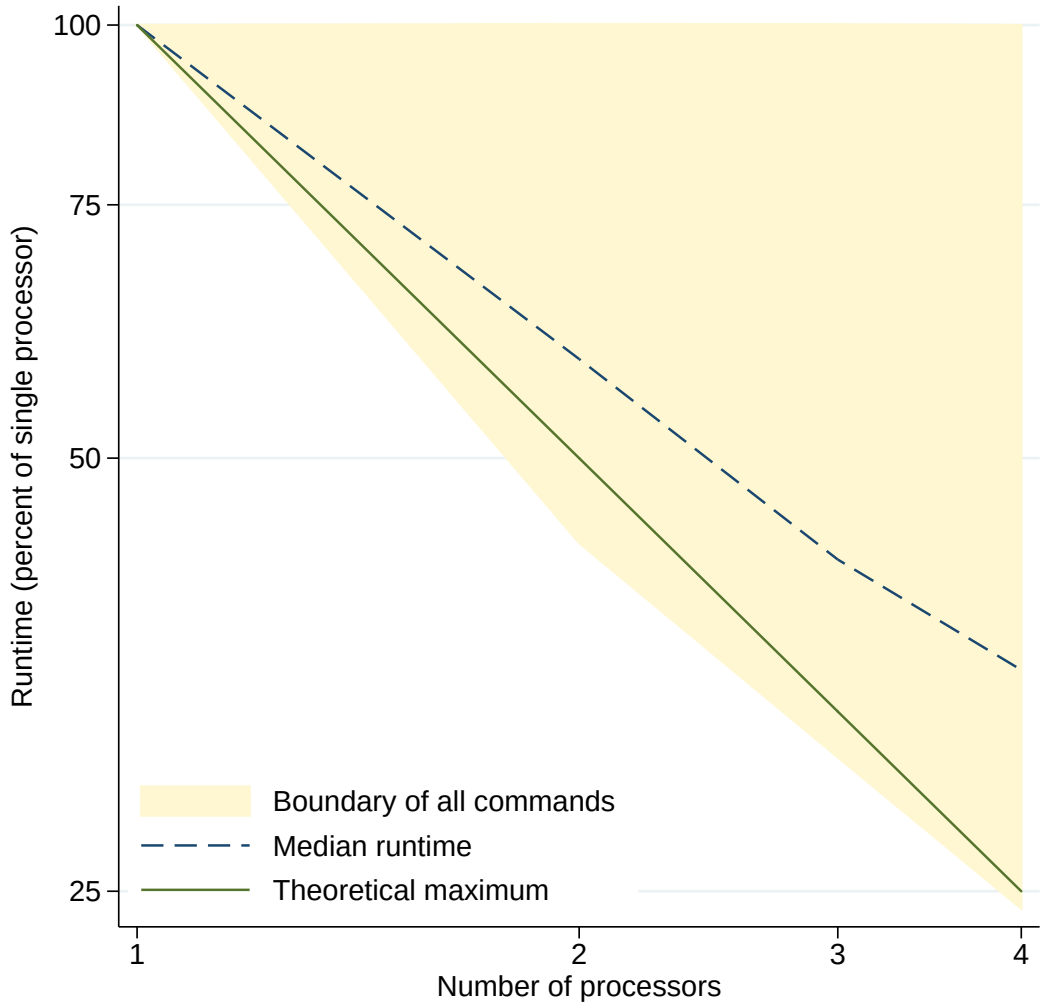
Stata/MP


Observed performance region
over all commands
Observed upper bound
Median performance
(estimation)
Median performance
(all commands)
Observed lower bound
Speed compared to single processor
1
2
3
4
Number of processors

Stata/MP - All commands



Stata/MP - Estimation commands



Comments on median results

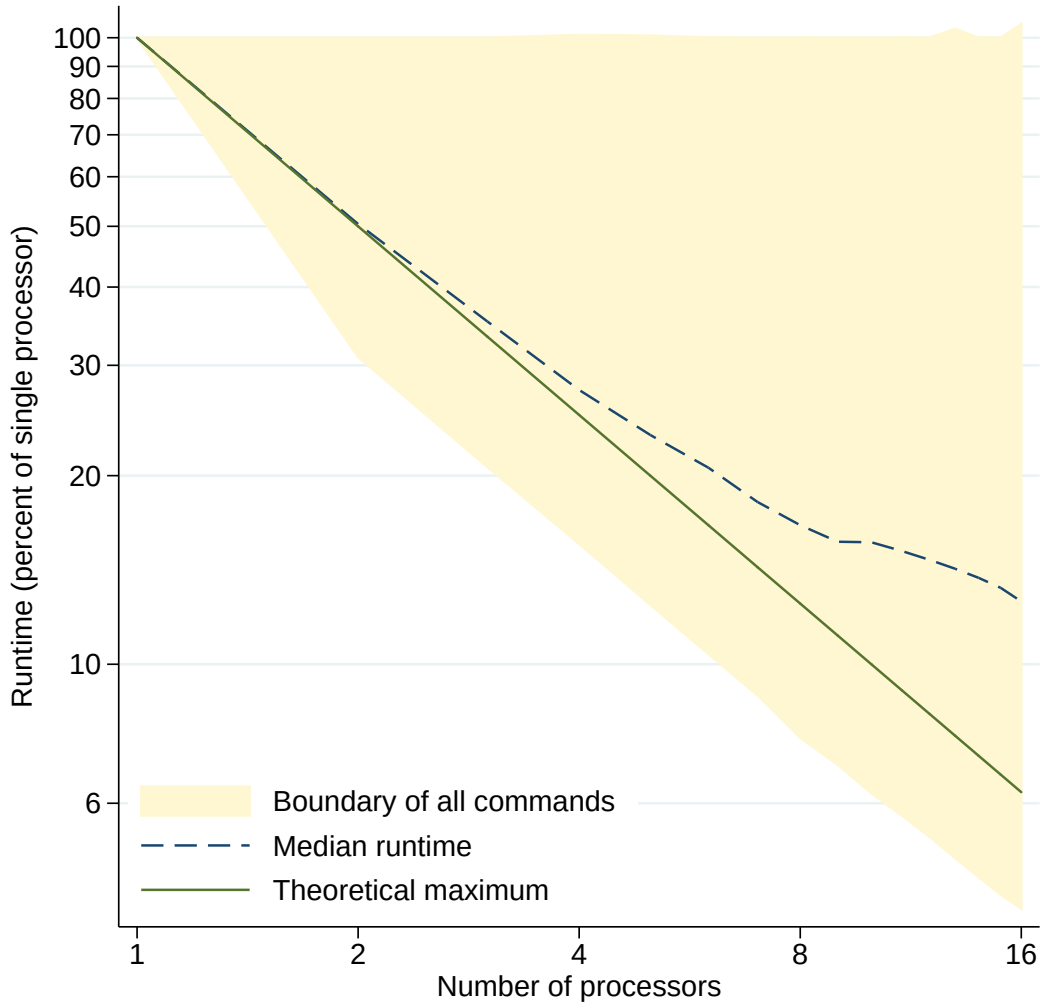
- half of commands run faster
- some even faster than theory due to cache effects
- half of commands run slower
- some not sped up at all
  - inherently sequential/impossible to parallelize (time series)
  - no effort made to parallelize (graph, xtmixed)

Stata/MP

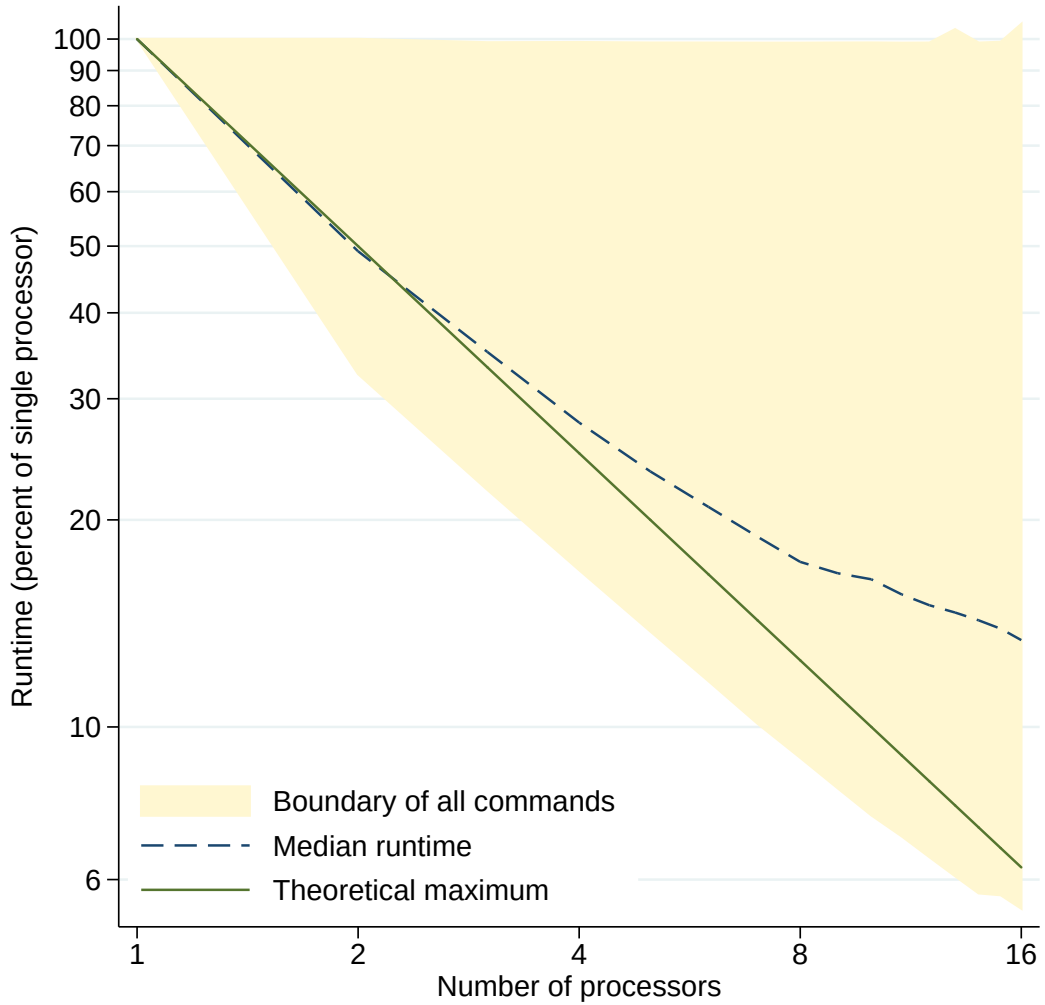
Methods

- Open/MP API
- Core algorithms
  - `generate`, `replace`
  - $X'X$
  - Inverses
  - 'Summers'
  - Solvers
- Modifications to individual important internal routines
- Almost 400 sections of code modified

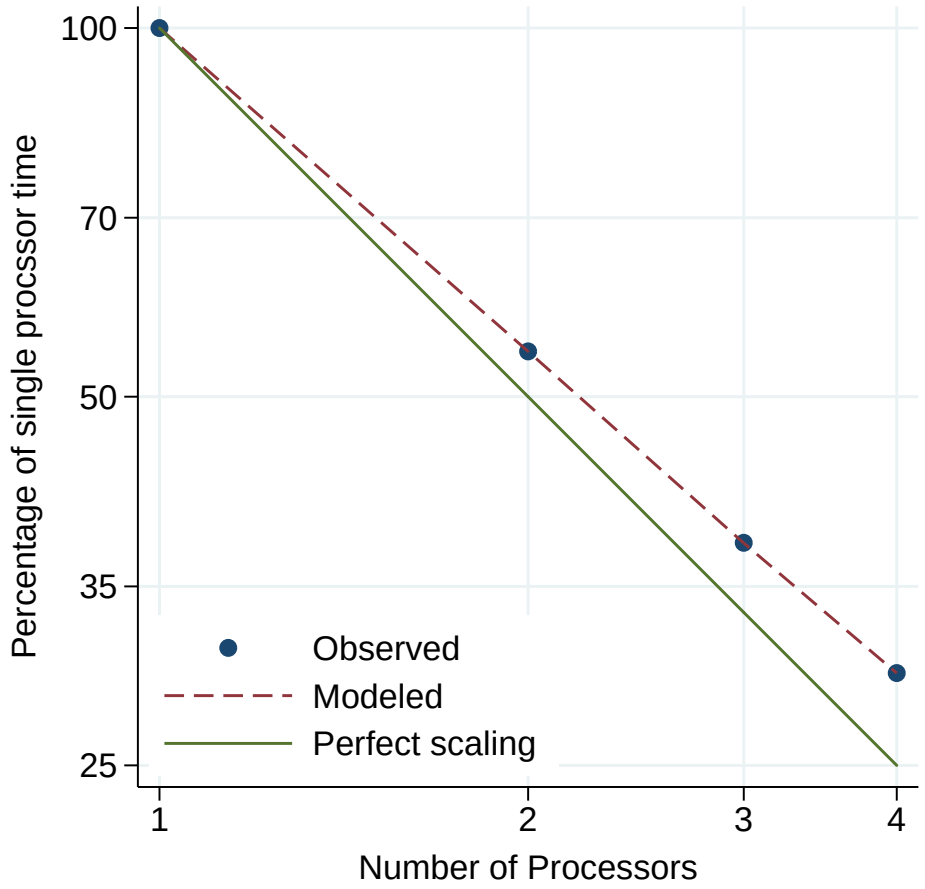
Stata/MP - All commands



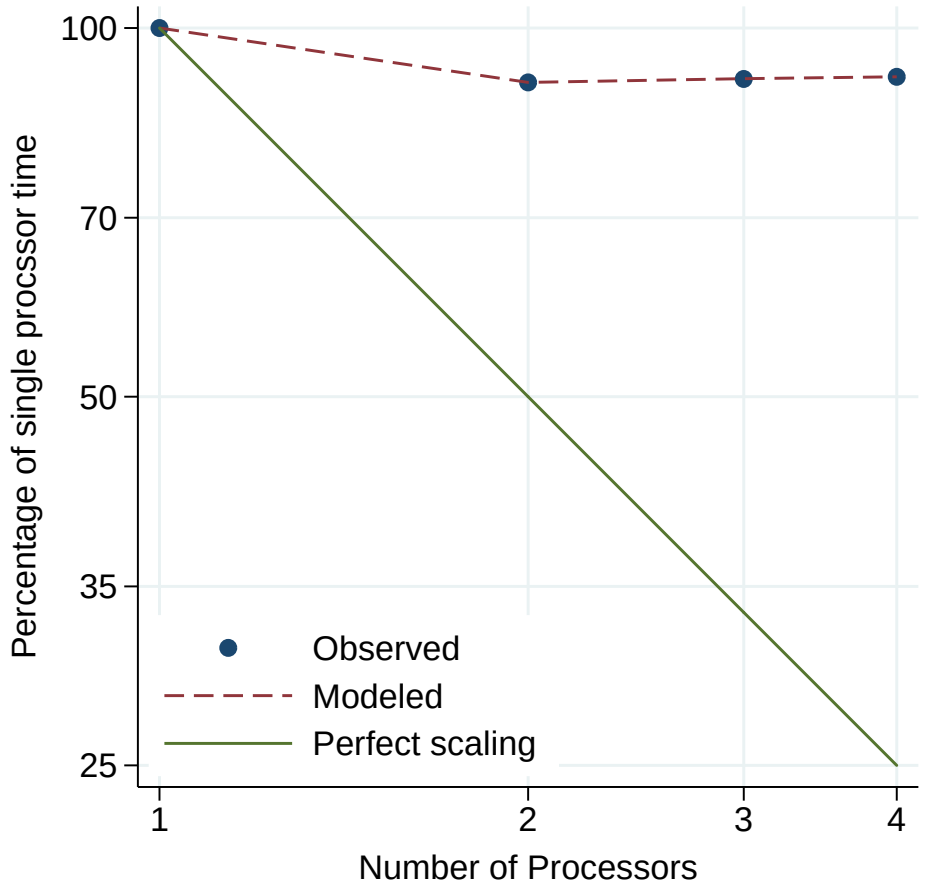
Stata/MP - Estimation commands



Stata/MP - regress

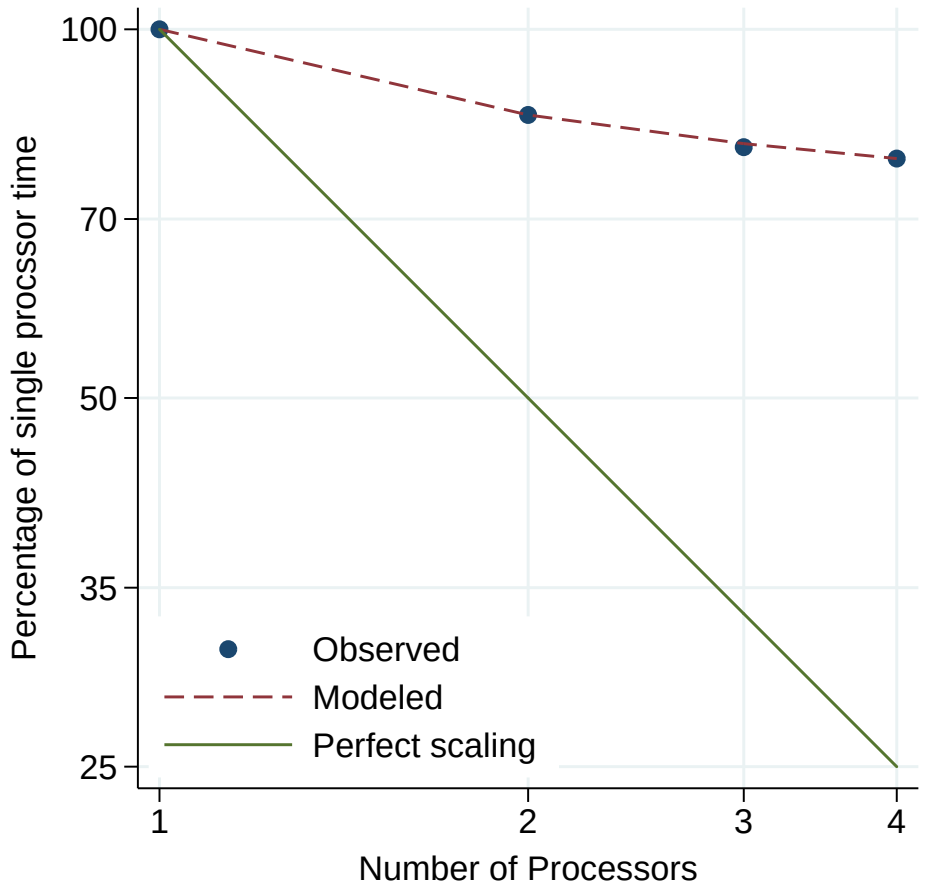


Stata/MP - arima



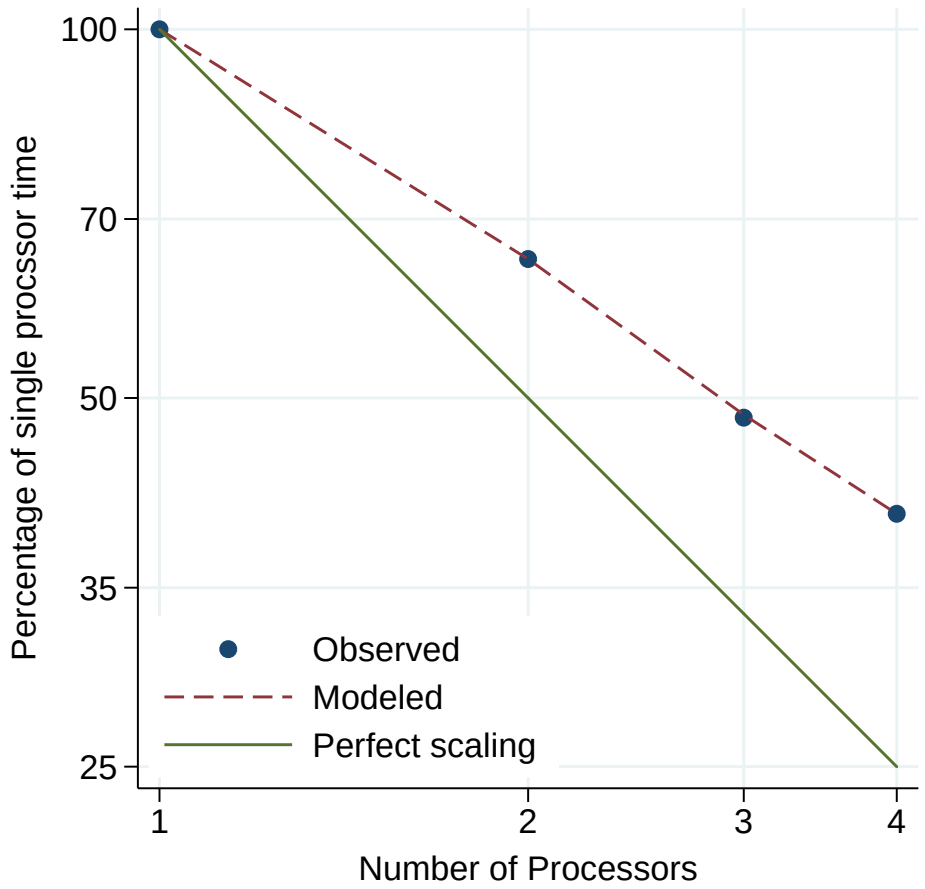
Stata/MP - gllamm

```
gllamm, i() geqs() link(ologit) family(binom)
```



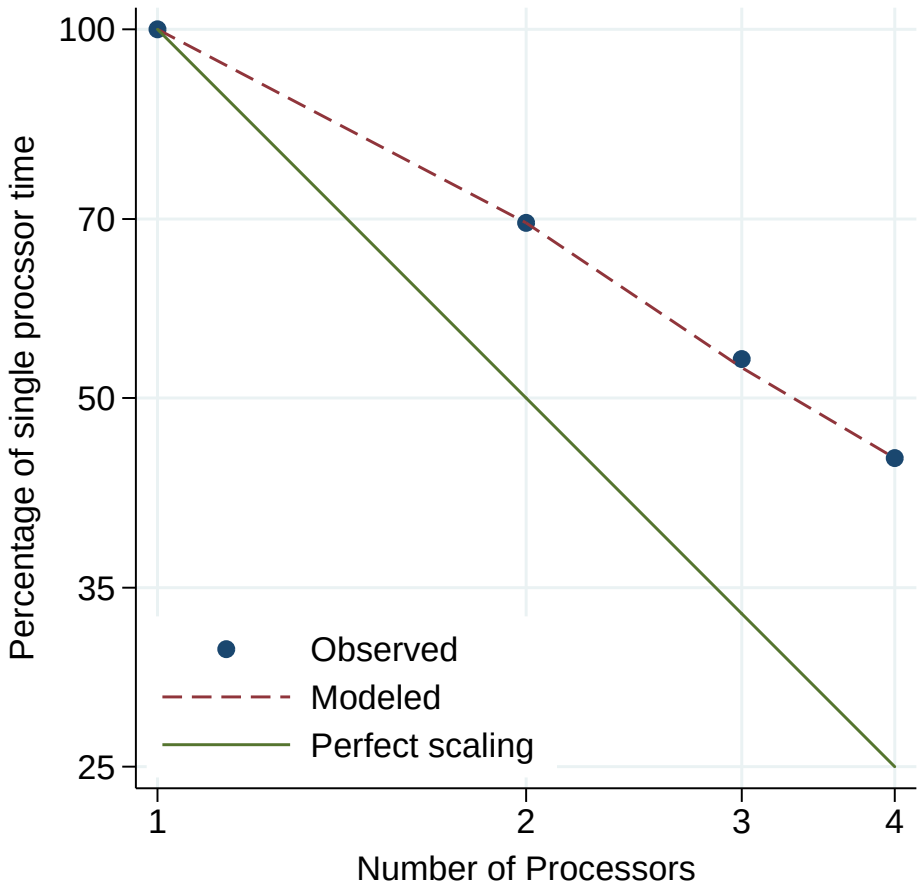
Stata/MP - gllamm

```
gllamm, i() geqs() link(logit) family(binom) nocons
```



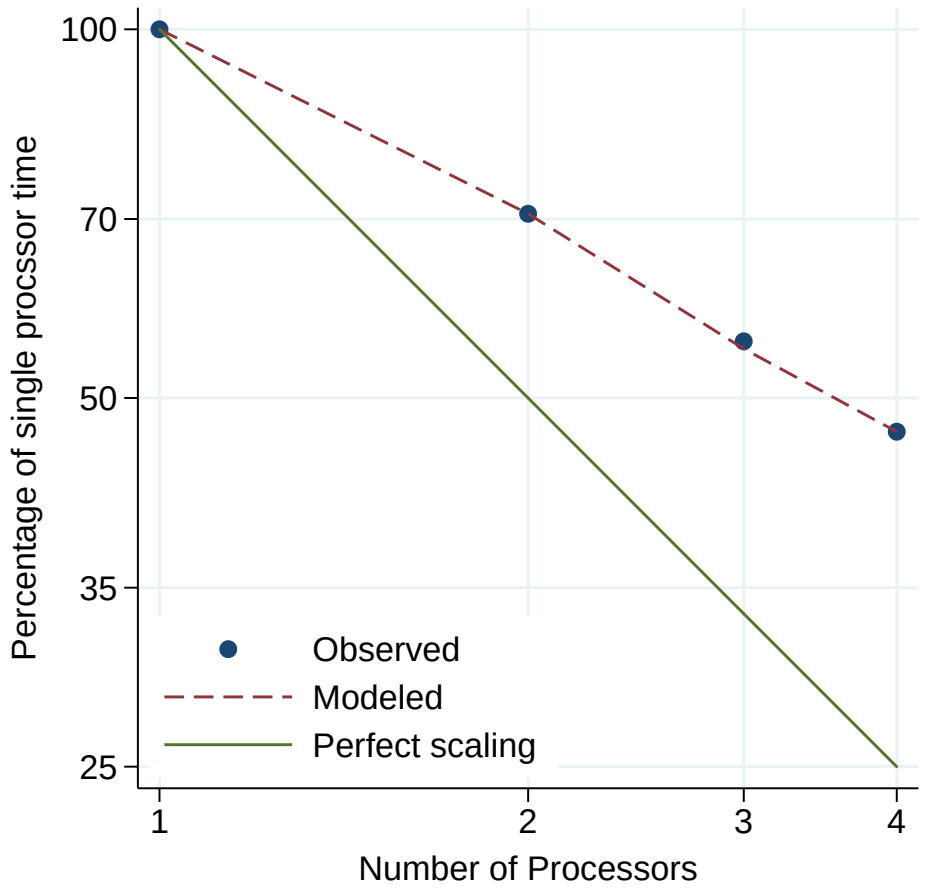
Stata/MP - gllamm

```
gllamm, i()
```

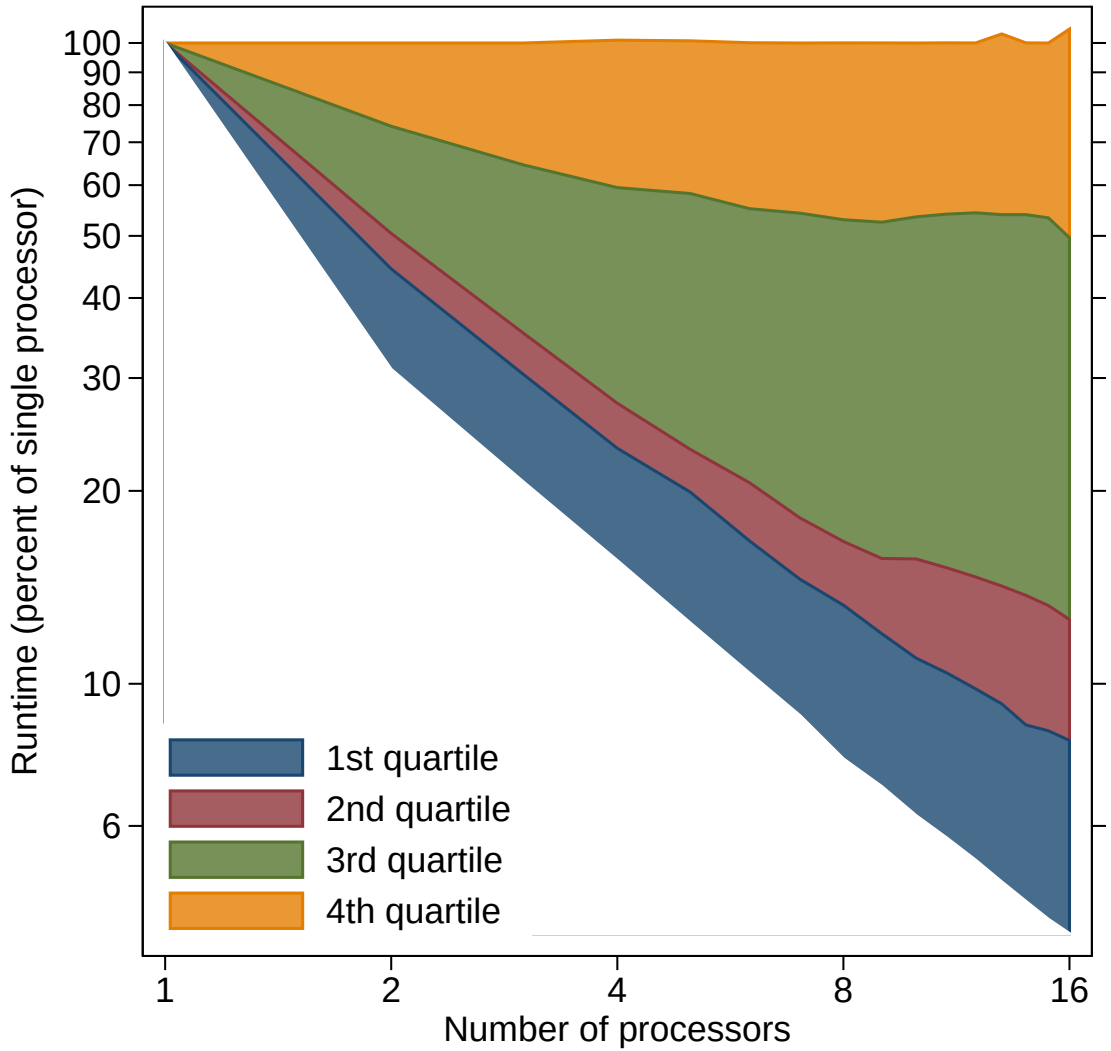


Stata/MP - gllamm

```
gllamm, i() eqs(cons w)
```



Stata/MP


100
90
80
70
60
50
40
30
20
10
6
Runtime (percent of single processor)
1st quartile
2nd quartile
3rd quartile
4th quartile
1
2
4
8
16
Number of processors