



# Helping Non-traditional HPC Users Using XSEDE Resources Efficiently

**PIs: Robert Sean Norman (U South Carolina)  
Atsuko Tanaka and Chao Fu (U Wisconsin)**

**ECSS staff: Shiquan Su**

**National Institute of Computational Science  
University of Tennessee, Knoxville**

**ECSS Symposium, August 18, 2015**

# Outline

- **Project I:**

**Linking Microbial Community Molecular Ecology to Ecosystems Stability**

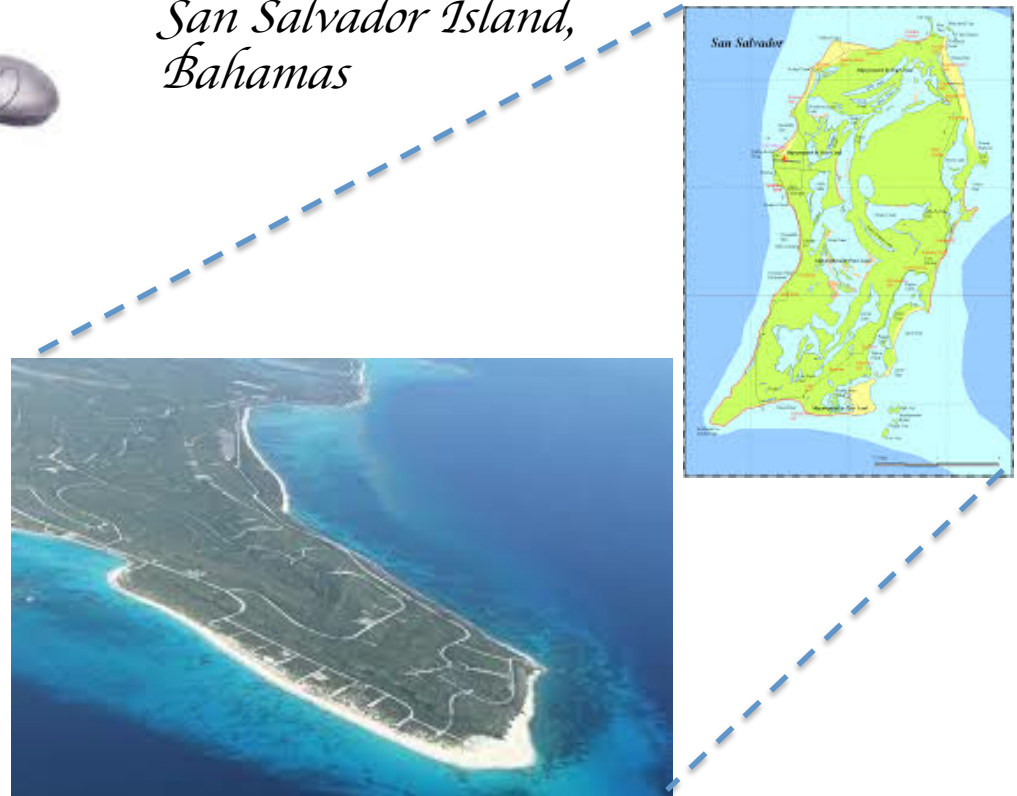
- **Project II:**

**College Loans and Post-schooling Skill Accumulation**

# Linking Microbial Community Molecular Ecology to Ecosystems Stability



*San Salvador Island,  
Bahamas*



- **The project aims to develop a bioinformatic pipeline for analyzing millions of DNA and cDNA sequences on XSEDE supercomputer.**
- **The PI collected experiment data from ocean coast at San Salvador Island, Bahamas, and analyzed the sequences data with BLAST tools on Stampede.**
- **The main effort of the ECSS support is helping the PI to improve the workflow of the BLAST runs.**
- **An in-depth study of running BLAST tools on Stampede can be found in XSEDE 2015 proceeding: “NCBI-BLAST programs optimization on XSEDE resources for sustainable aquaculture”.**

# Original Approach:

for number in range(1, numberOfFile + 1):

```
f.write('QUERY='+my_input + str(number) + '.fasta\n')
```

```
f.write('#BLAST DB\n')
```

```
f.write('DB='+data['db_path']+'\n')
```

```
f.write('#check for Sequences file\n')
```

```
f.write('OUTFMT='+data['OUTFMT']+'\n')
```

```
f.write('MAX='+data['MAX']+'\n')
```

```
f.write('#OUTPUT file\n')
```

```
f.write('OUTPUT=$QUERY.blast.xml\n')
```

```
f.write('\n')
```

```
f.write('# run blast\n')
```

```
f.write('/opt/apps/blast/2.2.28/bin/blastx -query $QUERY -out  
$OUTPUT -db $DB/'+data['DB_PreFex']+' -max_target_seqs $MAX -outfmt  
$OUTFMT -num_threads $COREU\n')
```

```
f.close()
```

# of subdatabase



- **The database has been de divided into sub-databases.**
- **Due to the constraint of maximum number of jobs can be submitted. Only 50 Blast run jobs can be submitted each time.**
- **It took more than 20 days to process the PI's data.**
- **By implementing the SLURM ibrun options for advanced host selection, multiple queries of sub-database can be bundled in 1 job.**
- **After bundling about 4 queries in each job, it took less than a week to process all the data.**

# Improved Job Script

```
f.write('set -o verbose\n')
```

```
f.write('concurrency='+str(concurrency)+'\n')
```

concurrency=4



```
f.write('for (( offSet=0; offSet<$concurrency; offSet++))\n')
```

```
f.write('do\n')
```

```
f.write(' #INPUT Sequences file\n')
```

```
f.write(' qm=$(( ' +str(number)+ '+$offSet))\n')
```



```
f.write(' QUERY='+ my_input + '${qm}.fasta\n')
```

```
f.write(' #OUTPUT file\n')
```

```
f.write(' OUTPUT=$QUERY.blast.xml\n')
```

```
f.write(' # run blast\n')
```

multiple  
queries are  
bundled in  
one job  
submission



```
f.write(' ibrun -n 1 -o $((($offSet*$COREU)) /opt/apps/blast/2.2.28/bin/  
blastx -query $QUERY -out $OUTPUT -db $DB/' +data['DB_PreFex']+' -  
max_target_seqs $MAX -outfmt $OUTFMT -num_threads $COREU &\n')
```



```
f.write('done\n')
```

```
f.write('wait\n')
```



# College Loans and Post-schooling Skill Accumulation



**Student Loan Application Form**

giving as many details as possible. Failure to do so may res

**APPROVED**

**Personal Information**

Forename: JON  
123 MAZ STREET

Other Names:





# **Student Loan and Its Impacts on Career Path**

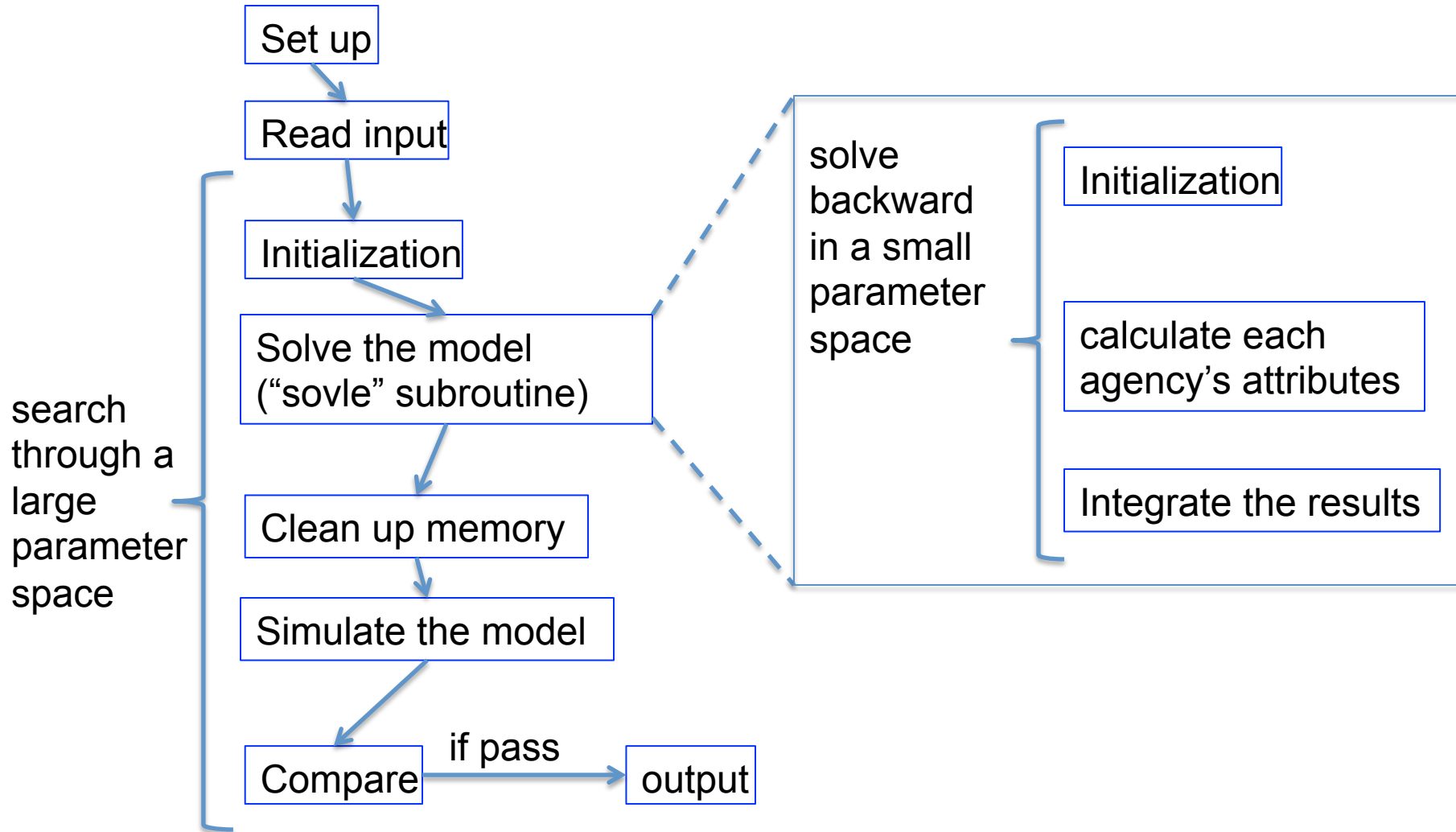
- **Current college loan debt is at a record high of \$1.1 trillion in the U.S.**
- **In 2012, 71% of all students graduating from four-year colleges had student loan debt.**
- **The average debt levels for all graduating seniors with student loans amounted to \$29,400.**

- **The access to college loans can improve welfare ex ante by alleviating the distortion that borrowing constraints may impose on individuals' educational decisions.**
- **The burden of repaying college loans may be associated with long-term costs ex post.**

# Numerical Study Approach

- **The model considers a (male) worker's dynamic decisions of consumption, asset accumulation, labor supply, and skill investment after leaving school.**
- **Solves the model for workers with different characteristics (including college loan take-up).**
- **Estimates the model by matching the above model prediction with the data features(real data) to select the model parameters.**
- **Using the estimated model, perform simulation under different repayment plans .**

# Flowchart of the Code



# Improve the Input Mechanism

```

shiquansu — tg812925@login2.stampede:~/testchaoFu2015/version05282015 — ssh — 101x32

DO count2 = 1, 1, 1 !
  alpha = (/0.0d0 + 0.8d0 *count2, 0.7d0/) !growth up to investment
  DO count3 = 1,1, 10
    deltax = 0.0d0 + count3 * 0.01
    DO count4=20,20,1
      rho = 1.0d0 + count4 * 0.1

    DO count5= 30,30,10!5,5, 5 !if it is too big, the wage takes a negative value
      vareta=(0.001*count5)**2
      DO count6 = 1,1
        lambda20 = - (/ 1.0d-1*count6 , 1.0d-1*count6 - 0.0d-2/) ! disutility fr
om working part-time by type
      DO count7 = 1,1 ! full-time disutility; should be bigger than 1.0d-3*coun
t6: disutility from part-time
        lambda21 = - (/ 5.0d-1*count7, 5.0d-1*count7 - 1.0d-1 /)

      IF (option(2) >= 1 .AND. option(2) <= 2 ) THEN
        CALL error_checker_before_main ! check asset fn etc.
      END IF

=====
! initialization
=====

      CALL initial_data(n_sim) ! calculate initial hk and assign unobserv
ed type based on observed characteristics
! .....
  
```

Original Approach:

1. loop index are hardcoded.
2. comments spread inline.
3. census data are in managed in Excel format

	A	B	C	D	E	F	G	H	I	J	K	L
1	6	1979	3	99	2	13557.89	3	24	1	2	8207.498	
2	7	1979	1	35	1	0	1	22	1	3	4255.74	
3	9	1979	2	51	1	0	1	23	1	4	3039.814	
4	11	1979	3	81	2	16791.99	3	-99	1	5	13172.53	
5	13	1979	1	85	2	0	1	-99	4	6	24115.86	
6	17	1979	1	88	2	0	1	-99	3	8	997.059	
7	18	1979	1	98	2	0	1	27	4	9	29992.83	
8	24	1979	1	61	2	0	1	20	1	11	3039.814	
9	26	1979	3	96	2	15586.79	3	22	4	12	12159.26	
10	34	1979	1	13	1	0	1	-99	1	13	126658.9	
11	39	1979	1	29	1	0	1	18	1	14	26953.02	
12	40	1979	2	38	1	743.4241	2	-99	1	15	22899.93	
13	41	1979	2	54	2	0	1	24	1	16	12199.79	
14	42	1979	1	16	1	2627.181	2	20	1	17	25182.22	
15	45	1979	1	49	1	0	1	18	1	18	21886.66	
16	47	1979	1	77	2	0	1	18	4	19	19252.16	
17	48	1979	1	32	1	0	1	18	1	20	8792.581	
18	52	1979	3	86	2	0	1	22	2	21	56743.2	
19	60	1979	1	75	2	0	1	18	4	22	59783.01	
20	64	1979	3	88	2	0	1	20	1	24	0	
21	66	1979	1	18	1	0	1	19	4	25	3039.814	

# New Input Mechanism:

InputParameters\_temp14\_cal.csv

A1	loop_start	loop_end	loop_step	unfold(1 is y)	estimate	scale	increment	Comment: parameter	Comment	Comment	Comment
1	loop_start	loop_end	loop_step	unfold(1 is y)	estimate	scale	increment	Comment: parameter	Comment	Comment	Comment
2	3	3	2	1	1	0.1	0.0004	gamma_assetlast	4 ok	# of queue	16
3	1	1	1	0	1	0.00001	0.03	gamma_assetlast (quad)	1		
4	-8	-8	3	0	1	0.1	0.07	gamma_assetlast(negative)	1		
5	4	4	2	1	1	1	0.2	gamma_hklast(type1)	4 ok		
6	5	5	10	0	1	1	0	gamma_hklast(type2)	1		
7	1	1	1	0	1	0.1	0	gamma_hk_quad	1		
8	2	2	2	0	1	0.01	0	gamma_a_hk	1		
9	-1	-1	2	0	1	0.1	0	gamma_a_hk (for a<0)	1		
10	3	3	2	0	1	0.1	0	alpha1	1		
11	4	4	2	0	1	0.1	0	alpha2	1		
12	3	3	2	0	1	0.01	0	delta	1		
13	4	4	2	0	1	0.01	0	vareta	1		
14	-1	-1	1	0	1	0.0001	0	lambda20	1		
15	-1	-1	1	0	1	0	0	lambda20(type2)	1		
16	-2	-2	3	0	1	0.05	0	lambda21	2		
17	-3	-3	1	0	1	0.05	0	lambda21(type2)	3		
18	-1	-1	1	0	1	0.0025	0	phi (1)	1		
19	1	1	1	0	1	0.01	0	phi (2)	1		
20	1	1	2	0	1	0.1	0	Ah(1)	2		
21	2	2	1	0	1	0.1	0	Ah(2)	1		
22	1	1	1	0	1	10	0.16	k0(0:)	1		
23	0	0	5	0	1	0.01	0	k0(1,1) age0	1		
24	0	0	1	0	1	0.001	0	k0(1,2) age0^2	1		

```

shiquansu -- tg812925@login2.stampede:~/testchaoFu2015/version08052015 -- ssh -- 139x40
DO count01 = ipm(1,1), ipm(1,2), ipm(1,3)
DO count02 = ipm(2,1), ipm(2,2), ipm(2,3)
DO count03 = ipm(3,1), ipm(3,2), ipm(3,3)
DO count04 = ipm(4,1), ipm(4,2), ipm(4,3)
DO count05 = ipm(5,1), ipm(5,2), ipm(5,3)
DO count06 = ipm(6,1), ipm(6,2), ipm(6,3)
DO count07 = ipm(7,1), ipm(7,2), ipm(7,3)
DO count08 = ipm(8,1), ipm(8,2), ipm(8,3)
DO count09 = ipm(9,1), ipm(9,2), ipm(9,3)
DO count10 = ipm(10,1), ipm(10,2), ipm(10,3)
DO count11 = ipm(11,1), ipm(11,2), ipm(11,3)
DO count12 = ipm(12,1), ipm(12,2), ipm(12,3)
DO count13 = ipm(13,1), ipm(13,2), ipm(13,3)
DO count14 = ipm(14,1), ipm(14,2), ipm(14,3)
DO count15 = ipm(15,1), ipm(15,2), ipm(15,3)
DO count16 = ipm(16,1), ipm(16,2), ipm(16,3)
DO count17 = ipm(17,1), ipm(17,2), ipm(17,3)
DO count18 = ipm(18,1), ipm(18,2), ipm(18,3)
DO count19 = ipm(19,1), ipm(19,2), ipm(19,3)
DO count20 = ipm(20,1), ipm(20,2), ipm(20,3)
DO count21 = ipm(21,1), ipm(21,2), ipm(21,3)
DO count22 = ipm(22,1), ipm(22,2), ipm(22,3)
DO count23 = ipm(23,1), ipm(23,2), ipm(23,3)
DO count24 = ipm(24,1), ipm(24,2), ipm(24,3)
DO count25 = ipm(25,1), ipm(25,2), ipm(25,3)
DO count26 = ipm(26,1), ipm(26,2), ipm(26,3)
DO count27 = ipm(27,1), ipm(27,2), ipm(27,3)
DO count28 = ipm(28,1), ipm(28,2), ipm(28,3)
DO count29 = ipm(29,1), ipm(29,2), ipm(29,3)
DO count30 = ipm(30,1), ipm(30,2), ipm(30,3)
DO count31 = ipm(31,1), ipm(31,2), ipm(31,3)
DO count32 = ipm(32,1), ipm(32,2), ipm(32,3)
DO count33 = ipm(33,1), ipm(33,2), ipm(33,3)

num_loops = num_loops+1
! initialization
!
where_am_i = "main loop"

```

## New Approach:

1. loop index is managed by a input parameter matrix
2. input parameter matrix and comments can be edited via Excel
3. “desktop experience”: Excel vs. script

# **MPI implementation**

## **Goals:**

- 1: unfold the deep nested loops across multiple MPI ranks**
- 2: keep the original serial version working, implement MPI only via enabling a flag during compilation.**

## **Approaches:**

- 1: calculate and reset the loop index based on the MPI rank**
- 2: group the MPI related codes in a module and marked the region with preprocessor directives.**



# MPI implementation

```
=====
PROGRAM Main_CollegeLoanJune3
=====
```

USE Parallelizer

```
USE params
USE globvar
USE setups
```

```
...
...
```

module for MPI related functions

```
#ifdef USE_MPI
CALL ParallelizerInit(ipm,doip,7,mpirank)
#else
OPEN (UNIT = 2, FILE = "InputParameters_temp14_cali.csv")
READ(2,*) headerline
DO jj = 1, doip
  READ(2,*) ipm(jj,1), ipm(jj,2), ipm(jj,3), ipm(jj,4),
  ipm(jj,5), scale_ipm(jj), increment_ipm(jj)
END DO ! jj
CLOSE(2)
n_params = sum(ipm(:,5))
#endif
```

use directive to enable "MPI mode"

```
tnoq=1
DO jj = 1, n
  IF ( 1==ia(jj,4) ) THEN
    tnoq=tnoq*( (ia(jj,2)-ia(jj,1))/ia(jj,3)+1 )
  ENDIF
END DO ! jj
tmpj=tmpi ← mpi rank
DO jj = 1, n
  IF ( 1==ia(jj,4) ) THEN
    tmpk=MOD( tmpj, ((ia(jj,2)-ia(jj,1))/ia(jj,3)+1) )
    ia(jj,1)=ia(jj,1)+tmpk*ia(jj,3)
    ia(jj,2)=ia(jj,1)
    tmpj=tmpj/((ia(jj,2)-ia(jj,1))/ia(jj,3)+1)
  ENDIF
END DO ! jj
```

calculate unfolded loop  
index from MPI rank

```
#ifdef USE_MPI
call MPI_COMM_RANK(MPI_COMM_WORLD, mpirank, mpierr)
out_unit=10000+mpirank
write (outfilename, "('all-',I5,'.txt')") out_unit
OPEN (unit = out_unit, file=outfilename, action="write",
status="UNKNOWN", position="APPEND")
#else
OPEN (unit = out_unit, file="results_all.txt", action="write",
status="UNKNOWN", position="APPEND")
#endif
```

set output files for each MPI rank

# **OpenMP Treatment for a Key Subroutine**

**In the early stage of the project, the researcher explicitly requested the help to shorten the walltime of the key subroutine “solve” in the most inner loop.**

**Conceptually, the “solve” subroutine is more localized, and no I/O is needed.**

**The “solve” subroutine also has a nested loop as its major structure.**

**So the OpenMP treatment is straight forward, but it needs to “collapse” several loops to generate enough parallelism.**

# OpenMP Directive

```
!$OMP PARALLEL DEFAULT(SHARED), PRIVATE(type_count,age_count,aa,a,kk,k), &
!$OMP& PRIVATE(loan_takeup,unpaid,t,st1,st2,etavec1,etavec2,assetmin), &
!$OMP& PRIVATE(v_old,value_wageshock,pp,hh), &
!$OMP& PRIVATE(maxv,maxv_nowork,maxv_part,maxv_full), &
!$OMP& PRIVATE(ii,i,newk,kk1,kk2,dK,etavec,w,wge,income,repay,c_max), &
!$OMP& PRIVATE(aa1,aa2,dA,newaa,newa,c,vt1), &
!$OMP& PRIVATE(probability,emax)
!$OMP DO collapse(2) schedule(dynamic,1)

    DO type_count = 1, num_type ! type; type affects utility fn (work disutility)
    DO age_count = 1, num_age0 ! age0; age(age0+t) affects the credit constraints

        state_ttl = state_ttl + 1
        state_ttl = (type_count-1)*num_age0+age_count

    !   Solve the model backwards
    !   for a particular individual w/ a certain characteristics (state_ttl)
```

- OpenMP clauses: default, private(/shared), schedule, collapse
- collapse clause needs OpenMP3.0+, (OpenMP version after 200805)

# Performance Improvement

```
c557-604.stampede(81)$ ifort -O2 MainJune18.f90
```

```
c557-604.stampede(82)$ /usr/bin/time -p ./a.out
```

**real 8.21**

user 7.08

sys 0.56

```
c557-604.stampede(72)$ ifort -openmp -O2 MainJune18.f90
```

```
c557-604.stampede(73)$ export OMP_NUM_THREADS=1
```

```
c557-604.stampede(74)$ /usr/bin/time -p ./a.out
```

real 8.16

user 7.10

sys 0.58

```
c557-604.stampede(75)$ export OMP_NUM_THREADS=2
```

```
c557-604.stampede(76)$ /usr/bin/time -p ./a.out
```

real 6.21

user 6.77

sys 0.58

```
c557-604.stampede(79)$ export OMP_NUM_THREADS=4
```

```
c557-604.stampede(80)$ /usr/bin/time -p ./a.out
```

real 5.41

user 8.57

sys 0.58

```
c557-604.stampede(106)$ export OMP_NUM_THREADS=6
```

```
c557-604.stampede(107)$ /usr/bin/time -p ./a.out
```

**real 5.01**

user 10.99

sys 0.64

```
c557-604.stampede(108)$ export OMP_NUM_THREADS=8
```

```
c557-604.stampede(109)$ /usr/bin/time -p ./a.out
```

real 5.22

user 13.72

sys 0.70

- Walltime is shorten from 8.2s to 5.0s, when the code runs on 6 threads.
- The researcher has a faster turn around time during the code development period.

# DDT on Stampede

- The researcher's home-brewed code is growing rapidly. Large modifications are often made and a lot of new functionalities are added.
- The code is fragile and requires constantly debugging.
- The DDT on Stampede helps to save a lot of debug time.
- The DDT guide can be found below:

<https://portal.tacc.utexas.edu/software/ddt>

- It is easy to use:

```
ssh -X stampede.tacc.utexas.edu
```

```
mpif90 -fopenmp -O0 -g debug_code.f90
```

```
module load ddt
```

```
ddt ./a.out
```

# Tips of Running DDT Faster:

- Make sure the X server is installed on your local machine (Xquartz, free on Mac).
- Install the local DDT client on your machine. The current version on Stampede is 5.0.1, the users need to download the older version client from <http://www.allinea.com/products/downloads/clients>
- Add -C option to the ssh command may also help. The option “-C” requests compression of all data (including stdin, stdout, stderr, and data for forwarded X11 and TCP connections). The compression algorithm is the same used by gzip.

`ssh -C -X your\_id@stampede.tacc.xsede.org`

**Thank You**