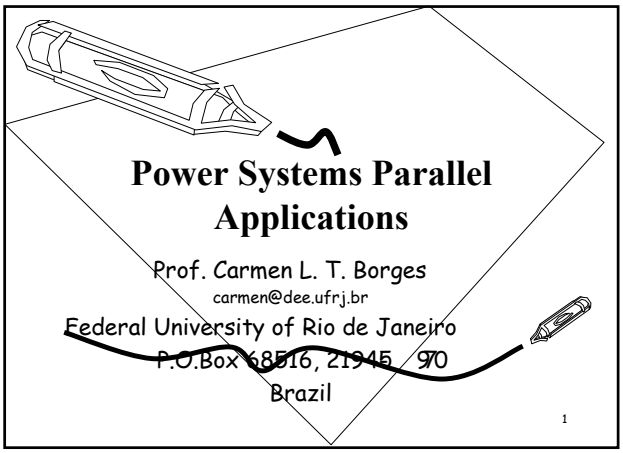
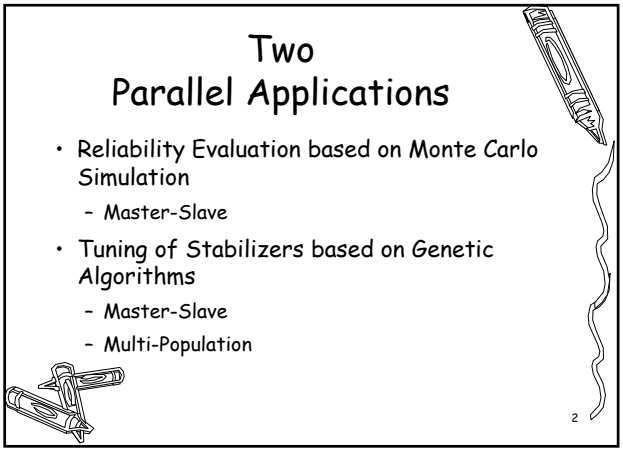
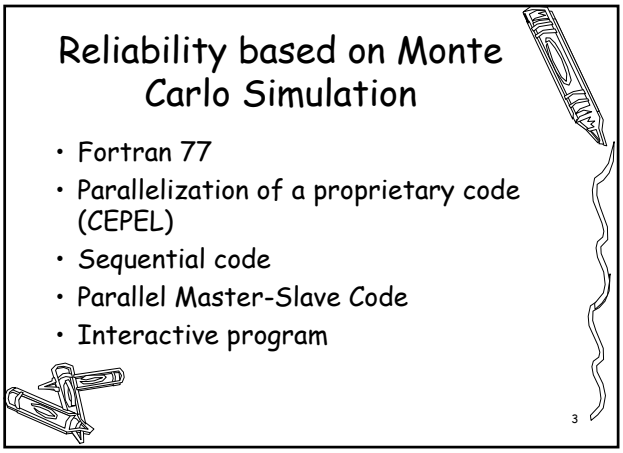

Power Systems Parallel Applications
Prof. Carmen L. T. Borges
carmen@dee.ufrj.br
Federal University of Rio de Janeiro
P.O.Box 68516, 21945 - 970
Brazil
1


Two
Parallel Applications
• Reliability Evaluation based on Monte Carlo Simulation
– Master-Slave
• Tuning of Stabilizers based on Genetic Algorithms
– Master-Slave
– Multi-Population
2


Reliability based on Monte Carlo Simulation
• Fortran 77
• Parallelization of a proprietary code (CEPEL)
• Sequential code
• Parallel Master-Slave Code
• Interactive program
3

Sequential Code (1)

$ ./nh2lib.exe

```
CEPEL - Centro de Pesquisas de Energia Eletrica
Programa de Confiabilidade Composta - NH2
Versao 3.5 - 07/02/97

X-------X---------X-----------------------------------------X
 UNIDADE   NOME            IDENTIFICACAO
 LOGICA  LOGICO            DO ARQUIVO
X-------X---------X-----------------------------------------X

   1   NH2$DADOS
   2   NH2$SAVCA savecase.dat
   3   NH2$CENAR /dev/null
   4   NH2$SAVCE /dev/null
   5   NH2$INPUT /dev/null
   6   NH2$PRINT /dev/null
   7   NH2$PUNCH /dev/null
   8   NH2$PWMOD /dev/null
  11   NH2$ERROS mensagem.out
  12   NH2$REGRE /dev/null

Codigo de Execucao:
```

4

Sequential Code (2)

Codigo de Execucao:
ulog **<ENTER>**
Unidade logica:
1
Arquivo associado:
rts.bat **<ENTER>**

... The program runs ...

Codigo de Execucao:
fim **<ENTER>**
STOP Termino normal statement executed

$

5

Parallel Code

$ mpirun -np 4 nh2lib.exe

- Same user interface
- Files generated:
  - result.<proc_rank>
    - Comunication and processing time
  - zsolv.<proc_rank>
    - Reliability results
  - rts.sai
    - Data output

6

Exercise

1. Evaluate the speedup and efficiency of the parallel code for:
   1. Test system rts.bat
   2. Test system newb.bat
2. Evaluate the influence of the estimation accuracy on the performance for:
   1. Accuracy 5%
   2. Accuracy 3%

7

Stabilization based on Genetic Algorithms

- Standard C
- Sequential code
- Parallel Implementatios:
  - Master Slave code
  - Multi Population code
- Batch program

8

Sequential Code (1)

./ag.exe

Generation	Time (s)	max.Ftnss	min.Ftnss	mean Ftnss	Std.Dsv.
0	1,70000	105,7244	0,1000	17,6542	28,3126
1	3,39000	107,2367	0,1000	27,7753	28,8938
2	5,09000	105,7318	0,1000	44,5163	31,6040
3	6,80000	107,2393	0,1000	59,4529	28,4884
4	8,52000	107,2416	0,1000	68,9128	26,8624
5	10,25000	107,2409	10,0000	74,2908	22,3229
6	11,99000	107,2422	10,0000	80,4421	20,4109
7	13,74000	107,2584	10,0000	84,6305	21,5904
8	15,51000	107,2584	10,0000	86,5707	21,6884
9	17,28000	107,2643	10,0000	90,1810	19,4687
10	19,05000	107,2706	10,0000	89,7103	17,6343

10 FITNESS MAXIMO MAXIMORUM = 107,270645
TOTAL COMPUTATION TIME (CPU function) = 19,049999
TOTAL COMPUTATION TIME (FULL function) = 19,062000

9

Sequential Code (2)

- File generated:
  - control_0s.m
    - Stabilizers details

```
%     MAX. FITNESS  1 = 107.270645
%
%     Parameters of the 2 PSS
%     K 1 =     47.409626
%     w 1 =     29.463194
%     ang. 1 =  8.462158
%     K 2 =      9.387336
%     w 2 =     27.716152
%     ang. 2 =  18.361687

x1 = [ 47.40963 29.46319  8.46216  9.38734 27.71615  18.36169];
```

10

Master-Slave Parallel Code

$ mpirun -np 4 agp2.exe

- Same user interface and output
- File generated:
  - **controlG_Op.m**: Stabilizers details

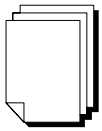
```
%     FITNESS  1 = 107.695602 (numprocs = 1)
%
%     K 1 =     32.409336
%     w 1 =     38.064339
%     ang. 1 =  8.956316
%     K 2 =      6.780302
%     w 2 =     23.655586
%     ang. 2 =  15.954640

          x1 = [ 32.40934 38.06434  8.95632  6.78030 23.65559  15.95464];
```

  - **"aggf",(int)FUN,"n",numprocs,"p.dat"**: Convergence details

11

Master-Slave Program Control

- GA Parameters
  - Population size: POPSIZE
  - No. of Generations: MAXGEN
- Output Control
  - Convergence File creation: SAVE_FILE



12

Master-Slave Exercise

1. Evaluate the speedup and efficiency of the parallel code varying:
   1. The size of the population
   2. The number of generations
   3. File generation *versus* monitor output

13

Multi-Population Parallel Code

$ mpirun -np 4 agpds11.exe

- Same user interface and output
- File generated:
  - **controlD_Ocp.m**: Stabilizers details

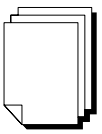
```
%     FITNESS  1 = 106.992889 (numprocs = 4)  MASTER PROCESS
%
%     K 1 =    22.598644
%     w 1 =    26.398087
%     ang. 1 =   7.774734
%     K 2 =     6.712119
%     w 2 =    21.445568
%     ang. 2 =  13.537065

x1 = [ 22.59864 26.39809  7.77473  6.71212 21.44557  13.53706];
```

  - **"agmf",(int)FUN,"n",numprocs,"cp.dat"**: Convergence details

14

Multi-Population Program Control

- GA Parameters
  - Population size: POPSIZE
  - No. of Generations: MAXGEN
  - Migration Rate: TASA_MIGRACION
  - Migration Interval: INTERVALO_MIGRACION
- Output Control
  - Convergence File creation: SAVE_FILE



15

Multi-Population Exercise

1. Evaluate the speedup and efficiency of the parallel code varying (file generation only):
   1. The size of the population
   2. The number of generations
2. Evaluate the influence in parallel performance and in the maximum fitness value of:
   1. Migration Rate
   2. Migration Interval

16

GOOD LUCK !

17