

OpenMP and MPI handson

http://github.org/rhaas/MCMC_bhpire

Roland Haas (NCSA / University of Illinois)
Email: rhaas@ncsa.illinois.edu



NCSA | National Center for
Supercomputing Applications

Get the source code

- We will work with Dimitrios' MCMC code introduced yesterday
- Roland is providing sample solutions in his own repo

```
git clone https://github.com/rhaas80/MCMC_BHPIRE.git  
MCMC_BHPIRE_parallel
```

- three branches
 - master: Dimitrios' version
 - mpi: MPI parallel
 - openmp: OpenMP parallel

```
git checkout master
```

```
git checkout mpi
```

```
git checkout openmp
```

Tasks for this afternoon

- pick one parallelism scheme
 - OpenMP (somewhat easier)
 - MPI (somewhat harder)
- write a parallel version of the MCMC code
- verify correctness
 - how?
 - document what you did?
- measure speedup

WARNING WARNING WARNING

SPOILERS after this slide

Gotchas

- OpenMP
 - remember that variables are shared by default
- MPI
 - seed each rank with a different seed

Gotchas

- OpenMP
 - watch out of static variables
- MPI
 - remember that IO needs to be serialized
 - remember that there is no “critical” section in MPI

Gotchas

- OpenMP constructs used
 - `omp parallel`
 - `omp fo`
 - `omp critical`
 - `omp threadprivate`
- MPI functions used
 - `MPI_Comm_rank`, `MPI_Comm-size`
 - `MPI_Abort`
 - `MPI_Gather`
 - `MPI_Init`, `MPI_Finalize`

Gotchas

- files modified for OpenMP version
 - chain.c
 - twister.c
- files modified for MPI version
 - mcmc.c
 - chain.c



Question?

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

