Tutorial on OpenMP programming
Victor Eijkhout
SSiASC 2016
OpenMP is a flexible tool for incrementally parallelizing a shared memory-based code. This course introduces the main concepts through lecturing and exercises.
Part I

The Fork-Join model
Computer architecture terminology

One cluster node:

A node will have 1 or 2 or (sometimes) 4 ‘sockets’: processor chips. There may be a co-processor attached.
Eight cores per socket, making 16 per node. They all access the same data.
Threads

Process: stream of instructions
Thread: process can duplicate itself, same code, access to same data

The OS will place threads on different cores: parallel performance.
Note: threads are software. More threads than cores or fewer is allowed.
To write an OpenMP program

#include "omp.h"

in C, and

use omp_lib

or

#include "omp_lib.h"

for Fortran.
To compile an OpenMP program

```bash
# gcc
gcc -o foo foo.c -fopenmp
# Intel compiler
icc -o foo foo.c -openmp
```
To run an OpenMP program

export OMP_NUM_THREADS=8
./my_omp_program

Stampede has 16 cores; more than 16 threads does not make sense.

Quick experiment:

for t in 1 2 4 8 12 16; do
    OMP_NUM_THREADS=$t ./my_omp_program
done