OpenMC Module: Description of OpenMC

Computational Reactor Physics Group

Department of Nuclear Science and Engineering
Massachusetts Institute of Technology

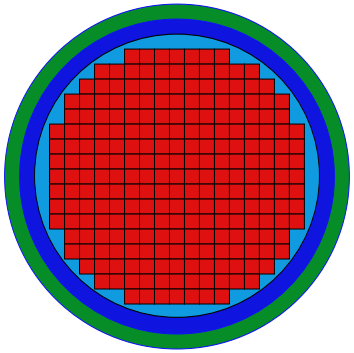
Goals of the OpenMC Project

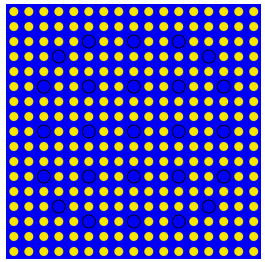
- Develop a Monte Carlo framework for reactor analysis that scales on leadership-class supercomputers (100,000+ cores)
- Provide a simple tool to analyze performance and limitations on proposed architectures
- Objectives:
  - Realistic physics
  - Modern programming style and data structures
  - Extensible for research purposes
  - Open source and freely available

Characteristics of OpenMC

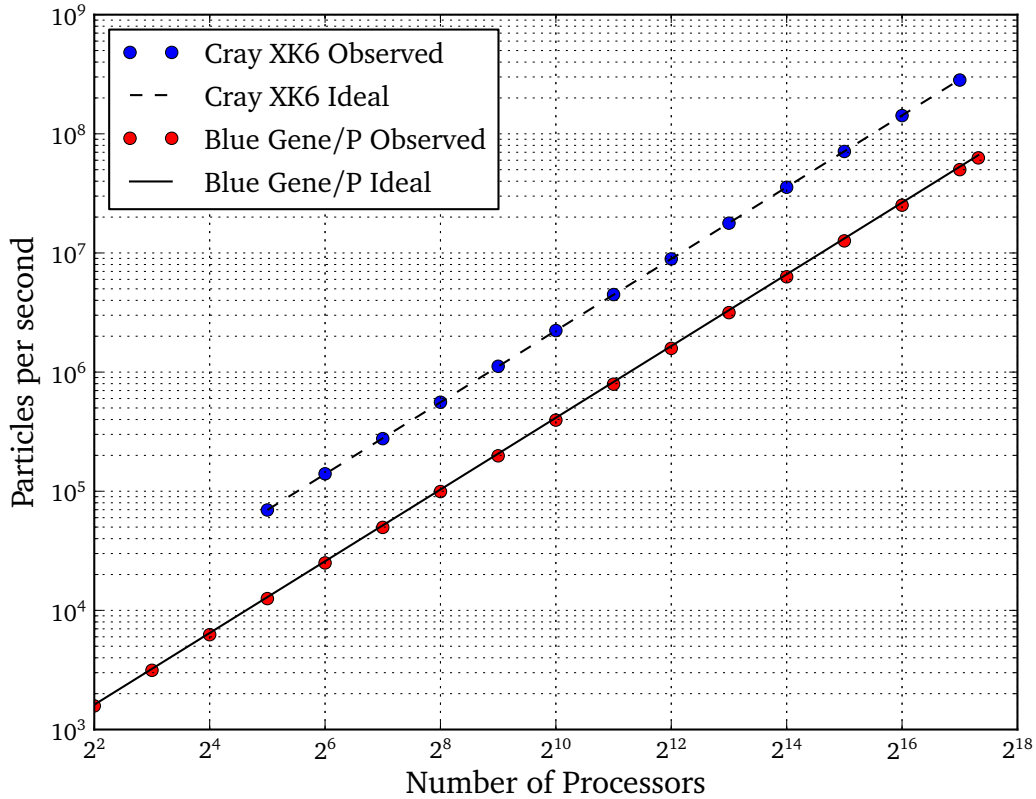
- Fixed source and k-eigenvalue calculations
- Geometry: constructive solid geometry, second-order surfaces
  - Universes and lattices
  - Rotations and translations
- Cross sections: ACE format (MCNP, Serpent)
- Physics: Only neutrons for now
  - $S(\alpha, \beta)$ thermal scattering
  - Unresolved resonance probability tables
  - Free gas thermal scattering
- Plotting: rasterization using find_cell
- Tallies:
  - Collision and path-length estimators
  - Surface and volume mesh tallies
- Distributed memory parallelism via MPI
- XML input format, HDF5 output format

Parallel Performance


Core

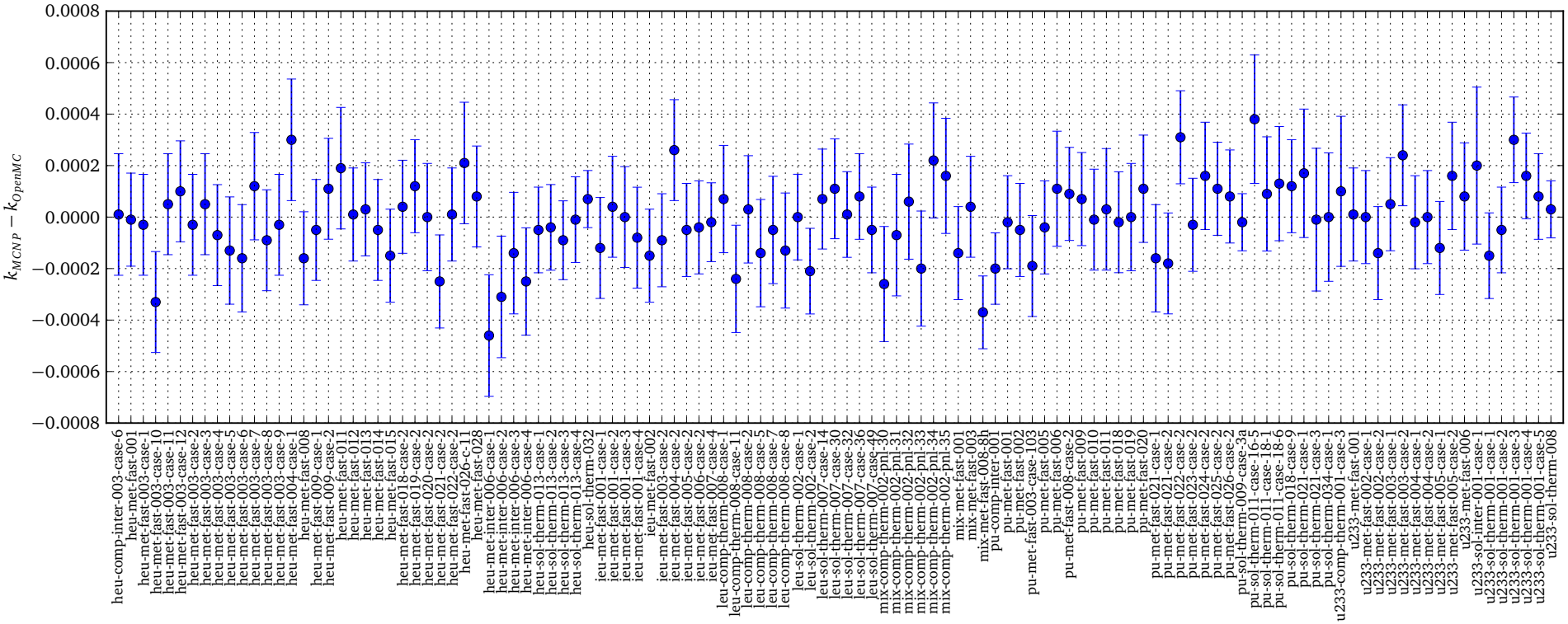

Assembly

- $2^{17} = 131,072$
- Cray XK6 = ORNL Jaguar
- Blue Gene/P = ANL Intrepid



Validation and Verification

- MCNP Criticality Benchmark Suite
  - 119 configurations — different spectra, materials, enrichment
  - Models built for all but two



MIT/X License

- Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
- The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Obtaining OpenMC

- Online at `https://github.com/mit-crpg/openmc`
- git:

```
git clone git://github.com/mit-crpg/openmc.git
```

- Ubuntu PPA:

```
sudo apt-add-repository ppa:paulromano/staging
sudo apt-get update
sudo apt-get install openmc
```

Compiling on Linux/Mac OS X

- ~30,000 lines of Fortran 2008 code
- All handled by a Makefile

```
cd openmc/src
make
```

- Optional dependencies
  - MPI for distributed-memory parallelism
  - HDF5 for portable standard binary output format
  - PETSc for CMFD acceleration
- Works with gfortran, Intel, IBM, Cray, PGI compilers
- On Mac OS X, can obtain compiler via Mac Ports

Compiling on Windows

- Uninstall Windows
- Install a better operating system, like Linux
- Follow previous directions

Compiling on Windows

- No native Windows binaries (yet)
- Options: Cygwin or MinGW (Minimalist GNU for Windows)
- Successful builds using both
- Visual Studio Port:
  `https://bitbucket.org/Lockee/openmc_vs2010`

Cross Sections

- You'll need:
  - A set of ACE format cross sections
  - A cross_sections.xml file
- Cross section data is not currently distributed with the code. Your options are:
  - Obtain JEFF data in ACE format from OECD/NEA
  - Use data from MCNP or Serpent
  - Process raw ENDF/B data into ACE format using NJOY

XML Input Format

- Unlike many nuclear codes, OpenMC uses a modular XML input format
  - Building a geometric model — **geometry.xml**
  - Define materials in problem — **materials.xml**
  - Settings paramters for the simulation — **settings.xml**
  - Determining which quantities to score — **tallies.xml**
  - Creating geometry plots — **plots.xml**

XML Basics

- XML = e**X**tensible **M**arkup **L**anguage
- Two kinds of information in a file:
  - *Markup*, like `<firstName>`, that describes the structure of the document, and
  - *Text*, like "Paul", that is the actual content of the document
- All text must be contained within *opening* and *closing* tags, e.g. `<firstName>Paul</firstName>`

XML Basics

- Nesting of elements usually represents logical relationships:

```
<?xml version="1.0"?>
<EmployeeDatabase>

  <!-- Information about our employees -->

  <employee>
    <firstName>Paul</firstName>
    <lastName>Romano</lastName>
    <age>26</age>
  </employee>

  <employee firstName="Li" lastName="Zhu" age="34" />

</EmployeeDatabase>
```

Configure Cross Section Data

- Download cross section data from website
- In this directory you will find a `cross_sections.xml`
- Open this file and change directory absolute path to your current directory path
  - You can type `echo $PWD` to determine this
- Set up cross section environmental variable
  - `export CROSS_SECTIONS=$PWD/cross_sections.xml`

Running your first simulation

- Executable compiled
  - Download source code from website
  - Navigate to `src` directory
  - Type `make`
- Navigate to directory with XML input files
  - Navigate to `examples` directory: `cd ../examples`
  - Go into `basic`: `cd basic`
- Run OpenMC to make sure code compiled correctly
  - `cd ../../src/openmc`

Online documentation

- When in doubt, check the user's guide!
- `http://mit-crpg.github.com/openmc`