EduHPC Workshop

OnRamp
to Parallel and Distributed Computing

https://github.com/ssfoley/onramp

Samantha Foley
ssfoley@cs.uwlax.edu
http://faculty.cs.uwlax.edu/~ssfoley

Josh Hursey
jjhursey@cs.uwlax.edu
http://faculty.cs.uwlax.edu/~jjhursey/

Project supported by SIGCSE Special Projects Grant (May 2015), Blue Waters Student Internship program (2015), and the UW-L Computer Science Department
There exists a **significant barrier to entry** for learning how to become productive in a Parallel Computing Environment (PCE) due to often unfamiliar and complex system software, programming interfaces, and tools.

<table>
<thead>
<tr>
<th>UNIX Shell (Scripting)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote System Access &amp; File Transfer</td>
</tr>
<tr>
<td>Compilation Environment (Makefile, mpicc, nvcc)</td>
</tr>
<tr>
<td>Runtime Environment (mpirun, srun, aprun)</td>
</tr>
<tr>
<td>Batch Submission Systems (PBS, SLURM, LSF)</td>
</tr>
<tr>
<td>Parallel Programming Paradigms (MPI, OpenMP)</td>
</tr>
<tr>
<td>Parallel Architectures</td>
</tr>
<tr>
<td>Parallel Programming Patterns, Tradeoffs, Scaling</td>
</tr>
</tbody>
</table>

**CS1/CS2 – (Java & Eclipse IDE)**

[https://github.com/ssfoley/onramp](https://github.com/ssfoley/onramp)
The OnRamp Project, provides a web-based portal that coaches users through interactive tutorials that teach them about the software ecosystem and parallel computing while allowing them to launch & explore parallel applications from day one.

https://github.com/ssfoley/onramp
OnRamp to Parallel Computing Project

• High-Level Goals
  • Encourage students to explore parallel and distributed computing concepts without the overhead of PCE system software peculiarities.
  • Help students transition to using the native PCE, eventually.
  • Bring together existing educational hardware & curriculum modules efforts into a flexible, portable architecture.

• Basic Architecture

https://github.com/ssfoley/onramp
OnRamp to Parallel Computing Project

- **Web Portal (client facing side of the OnRamp Server)**
  - Users do not need accounts on each PCE (use a shared UNIX account).
  - Users can be grouped into one or more Workspaces.
    - Workspaces are assigned PCE & Module combinations.
  - Users view Module instructions, launch jobs, view results, transfer files.
    - Automatically generated & validated custom forms for each Module.
  - Administrative panel to manage PCEs, one-click deploy Modules, manage Users, manage Workspaces, monitor usage, …

https://github.com/ssfoley/onramp
OnRamp to Parallel Computing Project

- **OnRamp Server**
  - Broker between the users of the OnRamp Portal and the PCEs.
  - Enforce policy, authenticate users, and manage security.
  - Cache files between Users and PCEs.

- **OnRamp PCE**
  - Custom drivers for each type of software environment (SLURM, PBS, LSF).
  - Manage modules, and user files associated with jobs.
  - Launch and monitor jobs on the system.

Developing ‘turn-key’ scripts to setup virtual clusters.

https://github.com/ssfoley/onramp
OnRamp to Parallel Computing Project

- **Curriculum Modules**
  - Complete freedom to structure the module as you like.
  - Write a few Python scripts to ‘hook’ into the OnRamp architecture.
  - Configuration files allow you to specify custom, module-specific parameters and validation requirements for users running the code.
  - Custom documentation/instructions
    - Can be tailored to the PCE environment

https://github.com/ssfoley/onramp
The OnRamp Project provides a web-based portal that coaches users through interactive tutorials that teach them about the software ecosystem and parallel computing while allowing them to launch & explore parallel applications from day one.

https://github.com/ssfoley/onramp