OnRamp to Parallel and Distributed Computing

Samantha S. Foley, Daniel Koepke, Christa Brehm, and Justin Ragatz
Computer Science
Univ. of Wisconsin-La Crosse
La Crosse, WI 54601
{sfoley, koepke.dan2, brehm.chri, ragatz.just}@uwlax.edu

Jason Regina
Mechanical Engineering
Univ. of Wyoming
Laramie, WY 82071
jregina@uwyo.edu

Abstract

Computer science students must understand parallel and distributed computing (PDC) concepts to be effective computer scientists in the workforce, as reflected in the ACM Curriculum guidelines. Communities of CS educators are creating educational modules, and making parallel compute environments (PCEs) available to educators who are integrating PDC concepts into their existing curriculum. Even with these resources there is a barrier to entry for students to use PCEs, namely the unfamiliar and complex system software ecosystem of modern PCEs. The OnRamp project lowers that barrier to entry for exploring PDC concepts on a variety of PCEs while also providing a path for students to learn how to be productive on the native PCE. OnRamp is designed to be a general purpose web portal for supporting the exploration of PDC concepts that harnesses the existing educational resources created by the CS education community. It coaches students through interactive modules that teach them about PDC concepts and PCEs while allowing them to launch parallel applications from day one. As students become more comfortable with running parallel applications on PCEs, OnRamp transforms into a reference guide as they graduate to using the native PCE. This poster describes the motivation, design and some motivating use cases for the project. We hope that this poster inspires parallel and distributed computing educators to use OnRamp in their courses.

1 Motivation

It has been established that exposure to parallel and distributed computing (PDC) concepts is required for computer science (CS) students. It is also widely known that PDC concepts can be a difficult to teach, and challenging for instructors to find necessary resources. To address this challenge many projects have emerged to provide both hardware and curricular solutions. The OnRamp project leverages the great work done by these various projects and adds a web portal that hides the complexities of using various different parallel compute environments (PCEs) and parallel programming tools. This web portal coaches users through a collection of interactive curriculum modules teaching them about parallel computing and the software ecosystem while allowing them to launch parallel applications on PCEs from day one.

2 Design

OnRamp is designed to be a general purpose web portal for supporting the exploration of PDC concepts and PCEs. The portal is not specific to any one programming language, parallel computing library, application, or PCE. This critical design point allows for maximal flexibility in how and where educators can use it with students. The four main components to OnRamp are:

Web front-end provides the user a view of the OnRamp system, connected PCEs, workspaces, and modules. Student users can explore curricular materials, learn about PDC concepts and PCEs, and launch pre-packaged parallel codes. Administrators can set up user accounts, set up PCEs, deploy modules, and create workspaces from this front-end interface.

Web server provides a centralized interface to the (potentially several) PCE backend(s). It handles authentication of OnRamp users and manages credentials for accessing PCEs. Some validation and logging of user activities is also provided.

PCE service provides PCE specific software to handle job launches, and manage user and module files. PCE agnostic job parameters are translated into PCE specific parameters which are passed to the native batch scheduler, MPI implementation, and module code.

Curriculum modules provide not only lesson plan information, but the necessary parallel code, Makefiles, and OnRamp scripts to allow it to be installed and executed on any PCE which has the OnRamp software installed.

The goals of the OnRamp software infrastructure are to:

- Make the installation and configuration of the system as easy and fast as possible, while being compatible with a wide range of hardware and system software configurations.
- Provide a robust and user-friendly web front-end for both student and instructor (administrator) use.
- Provide a collection of curriculum modules to support PDC education through OnRamp.

3 Use Cases

In order to illustrate how OnRamp can be used, a number of use cases are described:

CS1/2 In an introductory computer science course (i.e., CS1, CS2) students know some programming, but it would take significant time and effort to prepare students to use a PCE directly. However, introducing a few basic PDC concepts at this point in the curriculum can be beneficial and then can be built upon in later courses – following a spiral approach to reinforce and develop these concepts. OnRamp can provide the students with an easy to use mechanism to launch prebuilt applications with a small set of tunable parameters.

300-level Computer Architecture course In a Computer Architecture course where the students have only Java programming experience in an IDE, it would take a considerable amount of time to introduce the students to UNIX, C, parallel programming, remote login, build processes, and the batch scheduler system on a cluster. However, after introducing scaling, Amdahl's law, data and task decomposition, and shared vs. distributed memory architectures it would be great to incorporate a hands-on module where students can explore those concepts on a range of PCE architectures. Using a few modules in OnRamp and a collection of physical and virtual clusters available in the department, the students will gain a much more concrete understanding of the PDC concepts.

Upper-level Parallel Computing course In a parallel computing course where all of the hurdles will eventually be overcome and PCEs will be used directly, it would be great to be able to have students explore some parallel concepts on the first day or in the first week without having to deal with the details. An OnRamp module or set of modules can be used to explore parallel concepts and motivate students to work through the rest of the material of the course.

Computational Science course In a course outside of the computer science curriculum, OnRamp can be used to allow students in a different discipline use parallel applications without a depth of computer science knowledge that most pick up while in graduate school. A module can be developed for students and faculty to run simulations for class or research projects at the undergraduate level and beyond.

4 Future Work

This poster outlines the *Level One* collection of functionality for the OnRamp project. We plan to extend this work to support the learning of parallel programming (*Level Two*), becoming proficient at navigating a UNIX system (including dealing with complex build processes, batch scripts and schedulers, and parallel launch) (*Level Three*), and finally when the student graduates to using the PCE directly, it will serve as a useful resource for composing and running parallel jobs (*Level Four*).