Cloud-powered PDC Computations For a Runestone Interactive Textbook

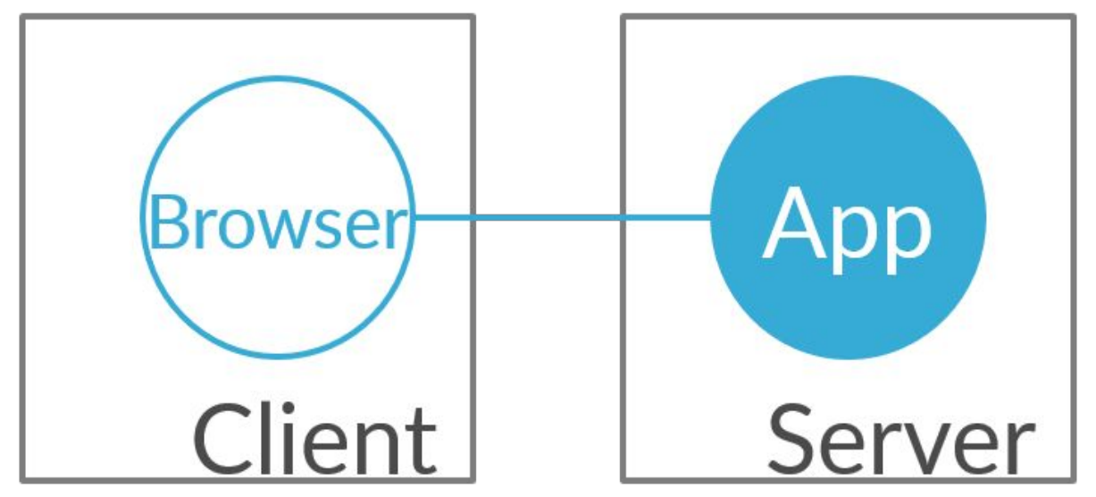
Researchers: Khang Huynh '23 (Leader), Tanaka Khondowe '22, George Kokalas '22 Supervisor: Richard A. Brown Technical Mentors: Anway De '21, Thong Vo '21

Abstract

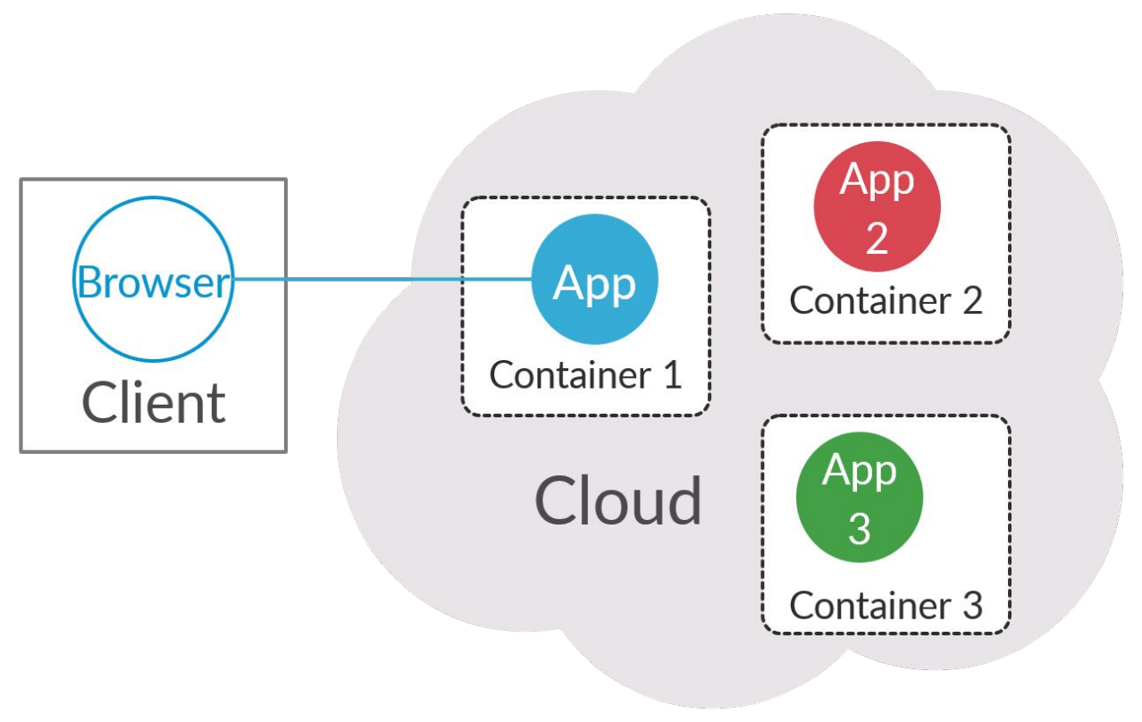
All computer science students need to learn about Parallel and Distributed Computing (PDC), but PDC is often challenging for beginning students. We seek to support a Runestone online interactive textbook for learning PDC, in which a reader can enter, modify and run computer code. This summer, we have created a way for PDC computations to take place within a Runestone book, which makes learning PDC convenient enough for beginners. Our approach is based on cloud computing, using Docker containers and the Kubernetes management system. This makes our solution easy to use elsewhere and more flexible for a large number of users.

Server Computer

Server computers are computers that provide online apps such as Facebook, Maps, Tiktok. When a user wants to use an app online, their computer will connect to the server to connect to the desired app.



Cloud Containers



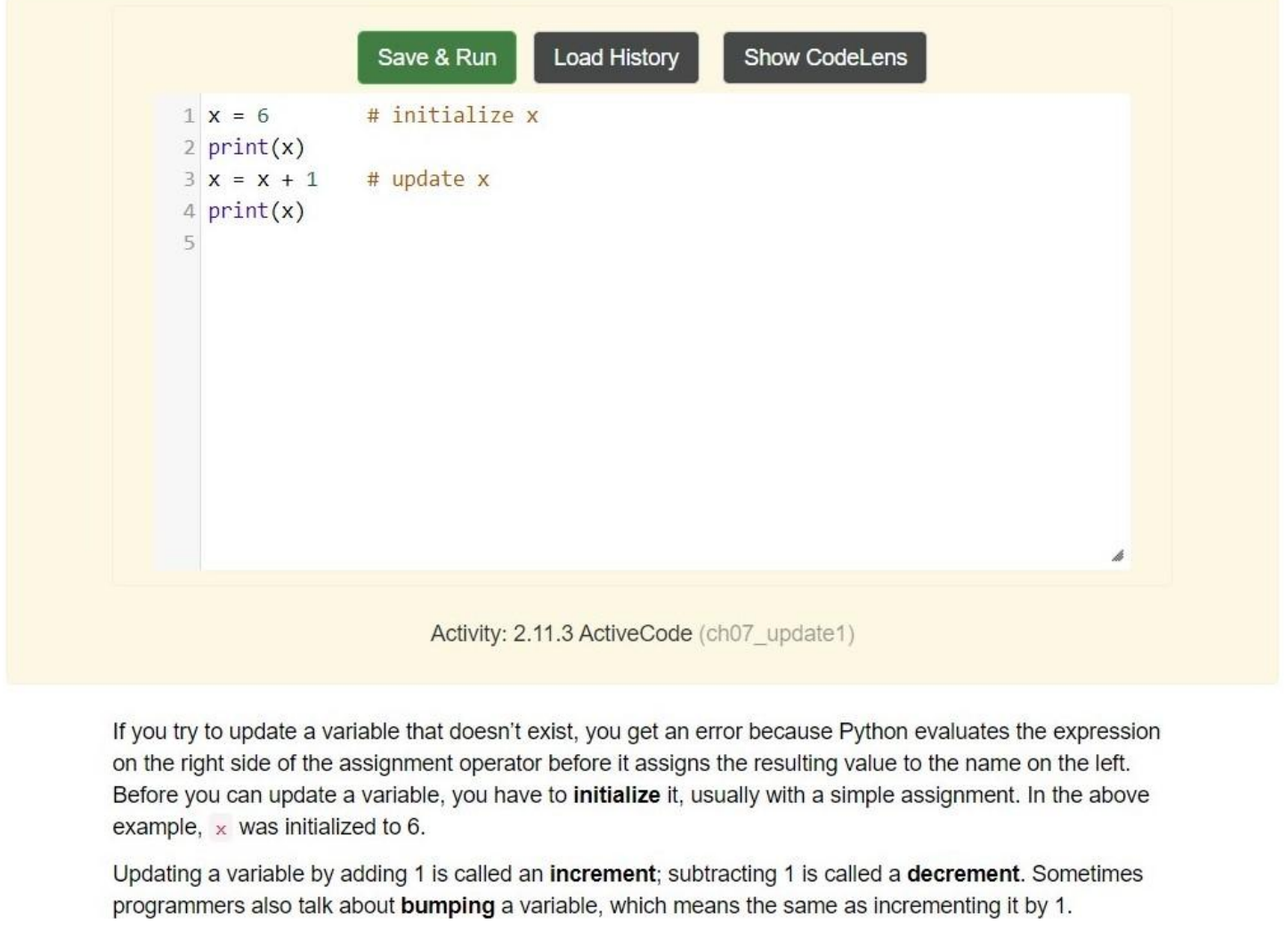
Containers are programs that behave like server computers on the cloud, hosting an app each. Just like before, the user's computer connects to the container as it would with a server computer to access the desired app.

Kubernetes (K8S)

Kubernetes is a system for managing containers in the cloud, including adding and re-running server applications and making them available for more and more users as needed. Many Kubernetes features concern resources that large and complex applications require

Runestone interactive books

Runestone is a server-based system for creating online interactive textbooks. Runestone offers effective learning features such as ActiveCode blocks which enable readers to enter or modify computer code within the online book



```
x = 6        # initialize x
print(x)
x = x + 1    # update x
print(x)
```

Activity: 2.11.3 ActiveCode (ch07_update1)

If you try to update a variable that doesn't exist, you get an error because Python evaluates the expression on the right side of the assignment operator before it assigns the resulting value to the name on the left. Before you can update a variable, you have to **initialize** it, usually with a simple assignment. In the above example, x was initialized to 6.

Updating a variable by adding 1 is called an **increment**; subtracting 1 is called a **decrement**. Sometimes programmers also talk about **bumping** a variable, which means the same as incrementing it by 1.

Kubernetes-powered PDC Runestone backend

Parallel Computing is a technique of using multiple cores and threads of a machine to increase the speed of computations. Distributed computing is a technique using multiple machines (computer, Raspberry Pi, etc.) to perform a single computation.

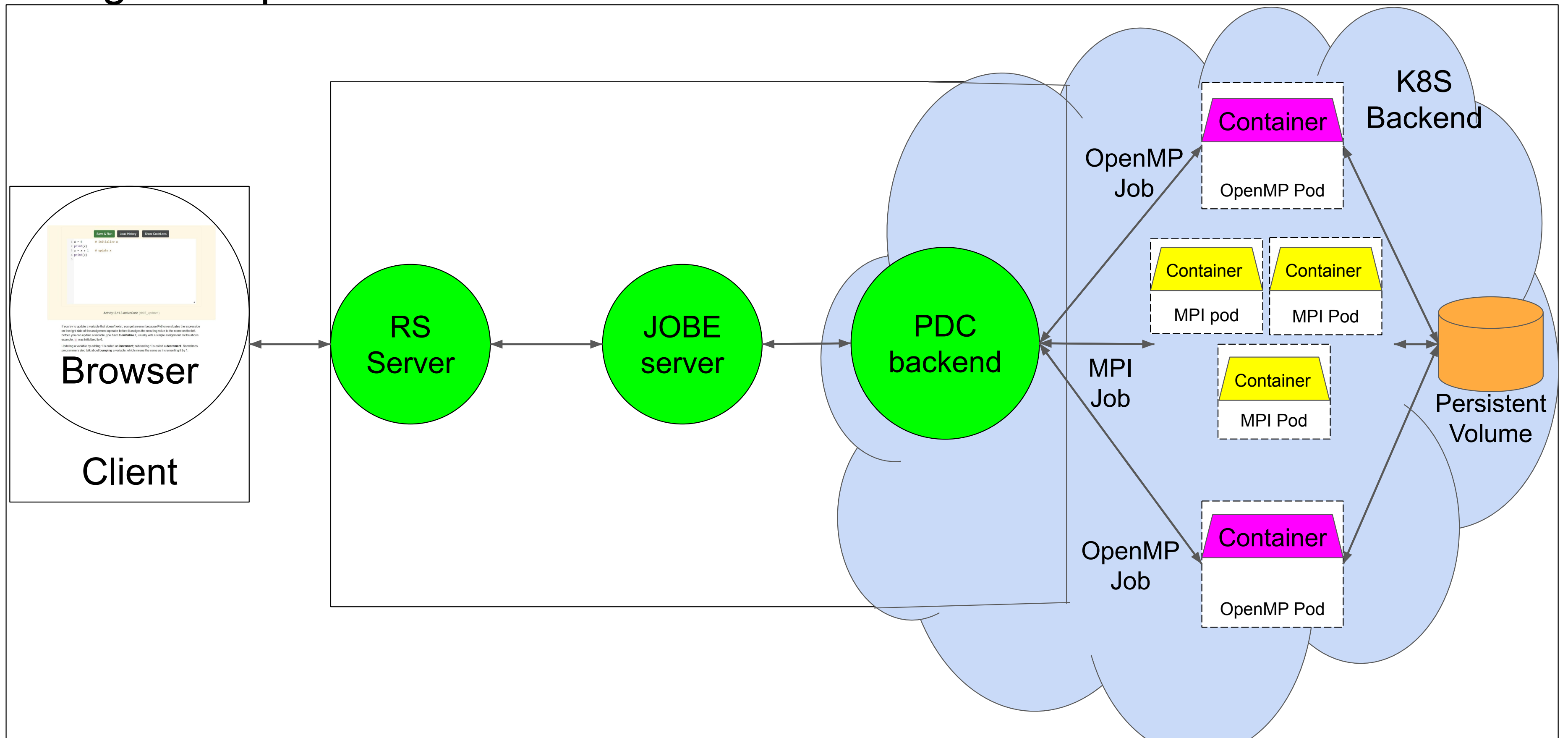


Figure. The Runestone Backend Architecture

In the software architecture diagram above, a Runestone server causes the interactive Runestone book to appear on the user's browser. That Runestone server connects to an additional server called Jobe for providing specialized ActiveCode computations. This summer, we are adding PDC backend code and other software for running those ActiveCode computations on containers through Kubernetes.

C-Kubed - containers for PDC

The backend-computation containers above were products of a prior St. Olaf research project called C-Kubed. In Kubernetes, pods are groups of one or more containers. One type of C-Kubed pod is for parallel computing (with OpenMP), and another C-Kubed pod is for distributed computing (with MPI). These pods require a Kubernetes persistent volume (PV), which provides code and data for the backend computations.

PDC textbook for beginner

Due to the fact that there is not a lot of resources available for beginning programmer to learn about PDC and also PDC is a challenging topics for any programmers, our textbook is intend to help the learners absorb fundamentals of PDC while giving the chance to actually work with some PDC code. With ActiveCode supported, the learners will also be able to challenge themselves by writing their own piece of code and analyze it.

Technical Challenges

Kubernetes is moving fast forward making C-kubed harder to compile and maintain. At the same time, other dependent applications and dependencies such as Docker, Calico, Rook & Ceph, MPI operator, etc. are also developing at the same speed with Kubernetes. Although we had many breakthroughs such as successfully deploying an OpenMP and MPI jobs, there are still some challenges about Persistent Volume in MPI jobs and the change we need to make to adapt to the development of latest Kubernetes version.

Future Plan

The research that we conducted this summer will be very beneficial for the foreseeable future, as it will open the path to "Kubernetize" the backend of Runestone. Here is our future plan:

- Make the PDC backend connected to JOBE server
- Keep the PDC backend up-to-date and easier to maintain
- Add new kinds of C-Kubed containers such as mpi4py, CUDA GPU computation

Acknowledgement

The advancements we made on Kubernetes was made possible thanks to online resources and the official Kubernetes documentation.
We are also grateful for the CSinParallel grant (NSF DUE-1822480)
We are also grateful to Anway De '21 and Thong Vo '21, the creators of the C-Kubed project, who provided us with extensive and indispensable consulting this summer, and to Eliott Cassidy '22 who contributed prior prototype work on the Runestone backend