Chapter 12

An Elastic Platform for Large-scale Assessment of Software Assignments for MOOCs (EPLASAM)

Michael Walker
Vanderbilt University, USA

Douglas C. Schmidt
Vanderbilt University, USA

Jules White
Vanderbilt University, USA

ABSTRACT

To address this efficiency and the resulting inherent scalability problems of Learning-at-Scale, this chapter propose a platform called SPLAShED: Software Platform for Large-Scale Assessment of Software-Development for Education-at-a-Distance, that uses Linux containers to provide OS-level virtualization. This gives each desired service the equivalent of a Virtual Private Server (VPS) that creates a temporary private userspace on the server. Each VPS provides a separate working environment for each desired application, but does not incur the overhead of traditional virtualization techniques. Our SPLAShED platform applies recent advances in Linux container deployment automation, resources isolation, portability, and usability. These advances enable the SPLAShED platform to serve both as. This chapter explores assignments as a case study: an Android based software design assignment. This case study shows how the SPLAShED platform will able to accommodate and facilitate advanced Software Development courses with features and abilities currently not available.

INTRODUCTION

A Massive Open Online Course (MOOC) is a web-based class environment aimed at open large-scale global participation via the Web (EdX, 2015). In contrast to traditional forms of face-to-face (F2F) education, MOOCs enable flexible learning styles, where learners can pick and choose which classes they take, as well as when and where they do their work. Conventional methods and tools used in F2F

DOI: 10.4018/978-1-4666-9743-0.ch012
education, however, are not ideal for use at scale in MOOCs. F2F methods and tools are particularly poorly suited to software-intensive MOOCs where the majority of the assessments focus on design and programming assignments. The following challenges must therefore be addressed by learning management platforms used for these types of MOOCs.

1. **Supporting the Scale of MOOCs**: The number of learners in MOOCs is typically orders of magnitude greater than even the largest F2F courses (i.e., tens of thousands vs. hundreds). The tools and techniques for F2F courses are therefore not sufficient to handle the number of learners and/or assignments to assess. In particular, learning management platforms for software design and programming assignments require new tools/techniques to work effectively at the scale of MOOCs.

2. **Alleviating Limitations of Distance**: The distance between teaching staff and learners also increases significantly in MOOCs as compared to F2F classes, which introduces challenges that traditional F2F courses need not address. For example, it is infeasible for the teaching staff to meet with learners in software-intensive MOOCs to provide them with meaningful feedback on their assignments. Therefore, learning management platforms for these types of MOOCs must alleviate the challenges that distance can cause with respect to personalizing the learning experience.

3. **Providing Security for the Learning Management Platform and its Users**: Security concerns arise when software submitted by learners is automatically compiled and tested. In particular, server-side compilation/execution and peer-review of code by other learners must be handled carefully to avoid security exploits, due either to malicious or accidental harm to the platform due to software bugs in learner solutions. The learning management platform for software-intensive MOOCs therefore requires a secure environment in which assignment submissions can be tested in isolation without harming the evaluation system or other learner submissions.

4. **Minimizing the Heterogeneity of System and Network Environments**: Software-intensive MOOCs that focus on concurrent and distributed applications requires multiple instances of software running in their own-networked environments. The architecture of these environments (e.g., the target architecture, such as mobile devices, embedded, or cloud services, etc.) may be heterogeneous and thus not identical to the assessment platform (e.g., x86 vs. ARM processors). The learning management platform should therefore support customizable target hardware architectures and dynamic virtual network topologies.

5. **Reducing Operational Overhead**: Various operational issues must be addressed by learning management platforms for software-intensive MOOCs, including information technology (IT) infrastructure tasks, such as logging and auditing of system execution and stability, load balancing, and data loss prevention strategies. Other operational issues relate to (1) transferring learner submissions from the front-end MOOC hosting server(s) to the learning management platform and returning result(s) back to the front-end servers and (2) handling robust and secure peer-evaluation of learner submissions.

**BACKGROUND**

To address the challenges listed in previous section, we are developing EPLASAM, which is both a learning management platform and method aimed at supporting the needs of learners and teaching staff in software-intensive MOOCs. EPLASAM is based on our experiences teaching the Mobile Cloud