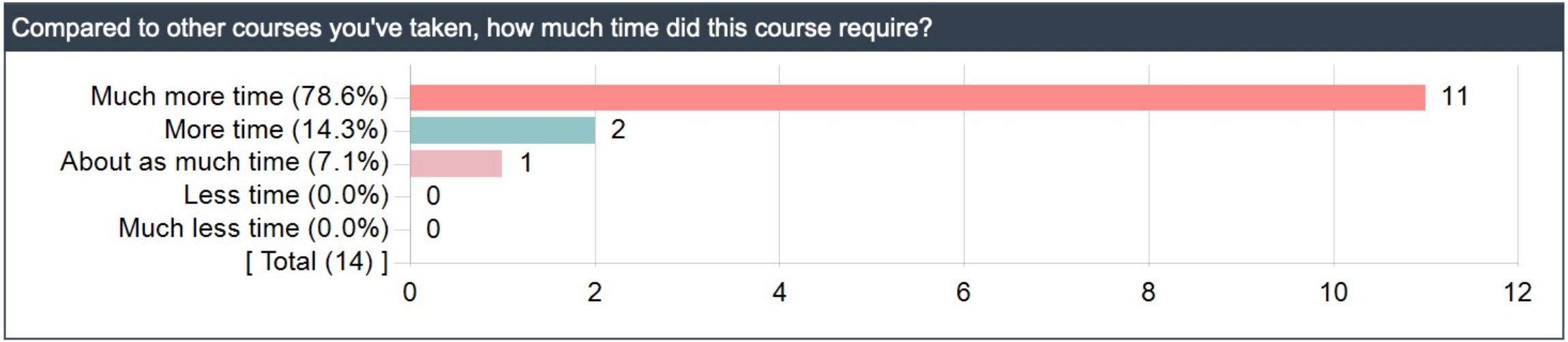
Applied Distributed Systems: Course Introduction

Marlon Pierce

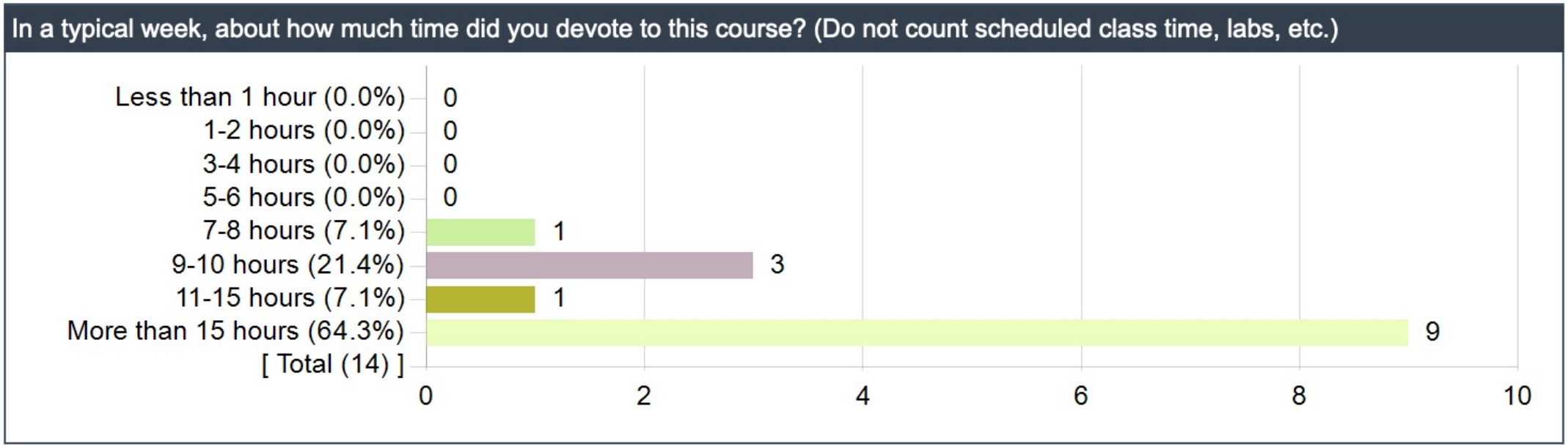
13 January 2022

What to Expect

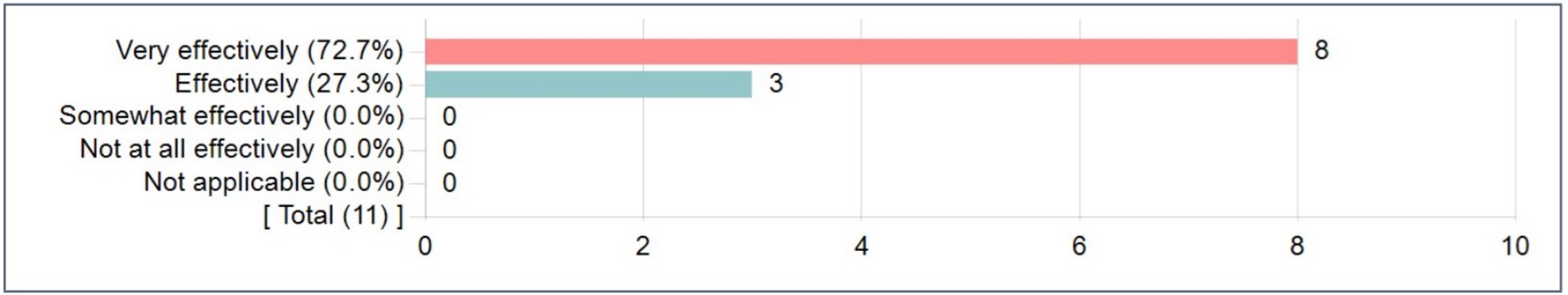
Compared to other courses you've taken, how much time did this course require?



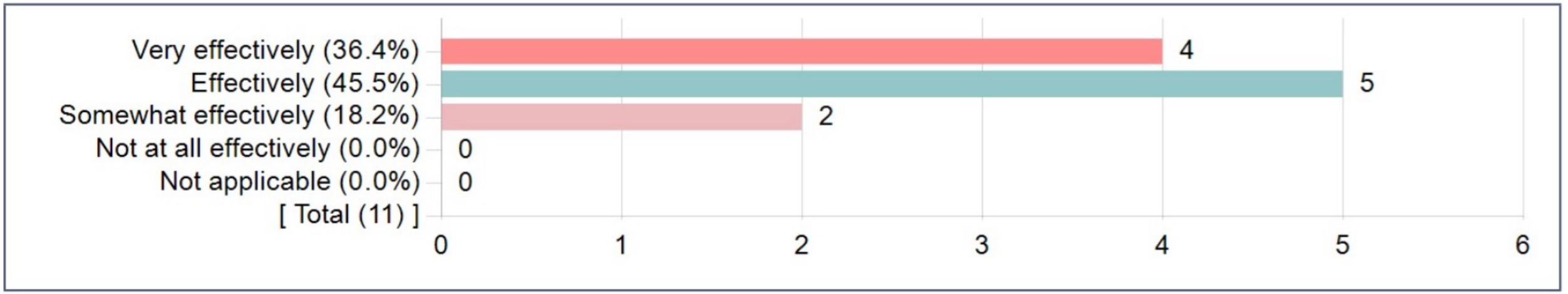
In a typical week, about how much time did you devote to this course? (Do not count scheduled class time, labs, etc.)



How effectively did out-of-class work (assignments, readings, practice, etc.) help you learn?



How effectively was class time used to help you learn?



What Is This Class About?

This Class Is Not About Full Stack Development

This Class Is Not About Becoming a Certified Commercial Cloud Engineer

I hope this class helps prepare you for the CS/IT future that none of us know about yet.

Some Time Invariant Skills:

Know How to Solve Problems

Be Grounded in Fundamentals

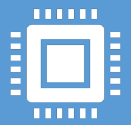
What Does “Cloud Native” Mean?

Cloud Native *X*

Architecture: How do you design a scalable distributed system?

Development: How do build a distributed system when you still do most of your development on a laptop?

Engineering: How do you deploy and operate at scale?

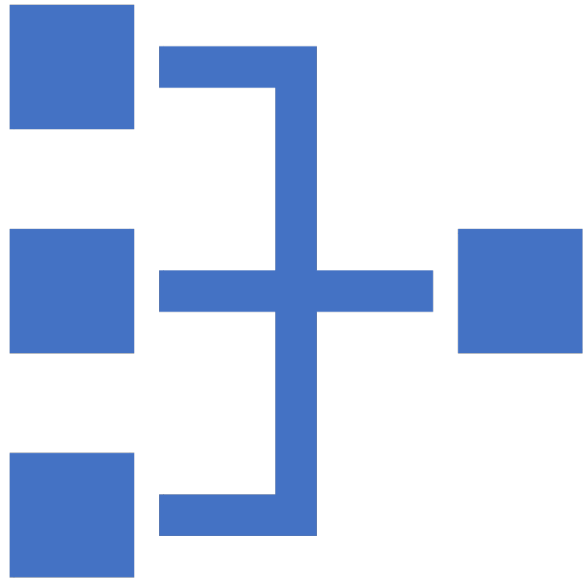


User Environments: How do you design user environments that are as scalable and flexible as the underlying systems?

Distributed Systems --> Cloud Native Systems --> Microservices

- Many “cloud native” services are pushing distributed systems to new scales
- Companies like Netflix have pioneered the use of 100’s of services and 1000’s of service instances
- Internet of Things systems will push this higher by orders of magnitude

What Are Distributed Systems?



Properties of Distributed Systems

Workloads are distributed over multiple independent entities (processes, servers, agents, ...)

Entities communicate with each other using messages sent over network connections

Why Do This? Compared to Monolithic Systems, Distributed Systems Are...

Potentially more scalable

Potentially more fault tolerant

Potentially built by integrating standalone or reusable capabilities

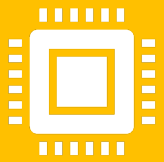
Potentially built using a range of programming languages, frameworks, etc

Potentially evolvable

Distributed Systems Are Hard to Build

Network disruptions will happen, knocking parts of your system offline

Distributed transactions and other coordinated activities are difficult



Security of and between components must be established

What are we to do?

Answer: Build on Foundations

The order is important

Network Messaging, Messaging Patterns, Protocols

Algorithms: Consistency, state management

Design Patterns: Codified best practices

Engineering Practices: DevOps, CI/CD

Tools

Opportunities

- We build lots of systems called science gateways
- We do this by operating a platform called SciGaP
- We base the platform on Apache Airavata software
- We work with outstanding students
  - Google Summer of Code via Apache Software Foundation
  - Advanced class
  - Hourly employees
  - Graduate research assistants