The Cattle of Geryon

Labour 09

lou lou@42.us.org
qst0 qst0@42.us.org

Summary:
Loaded by what was coded

Contents

Chapter I

Foreword

The Cattle of Geryon...

You might be able to tell what is worse:

Fighting countless monsters to steal some cattle.

or

Having to herd countless cattle while fighting monsters.

or

Having countless cattle scattered, leaving you to chase them down.

Heracules was not bothered by these tasks, he was in it for the long haul.
But he certainly wan't bothered by all the killing and fighting.
It was having so many cattle, and having them scatter that made this labour hard.

He spent a year chasing down each of the cattle, only to be stopped by a flood.

Hera, always super angry at Heracules, sent the flood.
She had also sent a gadfly to bite the cattle, making them scatter.
This was annoying sure, but Heracules knew he could fix the problem.
He piled stones into the river to make the water shallow.
Redirecting and balancing the current of the river till the cattle could cross.

Soon he arrived with the cattle in the court of Eurystheus,
The cattle were sacrificed to none other than Hera.
The labour was done...

Chapter II

Introduction

Heracules wasn't thwarted by the flood!
In this story we saw that although his path back to Eurystheus was blocked,
Heracules repaired his route and finished his labour.

For this labour, let's recreate a program called siege,
siege will be playing the part of Hera's flood.

Chapter III

Goals

Learn about how siege works and and can be used to benchmark a server.
For this labour you will write a program like siege.
The point of this program is to simulate placing a server "under siege."
You can use any language.

https://www.joedog.org/siege-home/

Chapter IV

General instructions

Learn about the program siege.
Understand the importance of benchmarking.

Recreate siege by writing your own siege program.

You can use any library or language you want!

Chapter V

Mandatory part

Release a great flood of simulated clients!
As mentioned before, write a program like siege,
Test HTTP load and benchmark how the server runs under different loads / attacks.

Do not just call the original siege program. You must have new
source code.

Chapter VI

Bonus part

Bonus you ask?
Always!

What would make this program better?

What features did you add?

Let's
Get
C r e a t i v e !

How did you go about meeting requirements?

How did you go about adding new features once requirements were met?