

Distributed Computing

Design Document sprint 3

Group C

Members

Evgeni Kurtev
Stefan-Nikola Stanev
Nikolay Gruychev
Vlad Bucur

Renis Hila
Naadir Twahir
Thomas Vasanth

Table of contents:

Members	1
Table of contents:	2
1. Introduction	3
Scope	3
2. Requirements	3
Functional requirements	3
Non-functional requirements	3
3. Architecture	4
Sprint 3	4
4. Design Details	4
5. User Interface Design	5
6. Security	5
7. Testing	5
8. Deployment	6

1.Introduction

The idea of our project is to have a hash calculation script. The script will be deployed to docker containers and each container will perform hash comparison with some wordlists.

Scope

The scope of the project is to create a system that will use multiple pods that will calculate the same hashing script. The information of the results will be distributed across the pods and then when the result is found the system returns it to the user and shuts down.

2.Requirements

Functional requirements

- Program that cracks hashes using wordlists that are distributed among the other worker nodes.
- AKS cluster that will hold the user application and the hash cracking pods.

Non-functional requirements

- The system should be capable of efficiently distributing hashed passwords and wordlists among worker nodes to minimize processing time.
- The system should be highly reliable, minimizing the risk of data loss or corruption during password cracking processes.
- The system should ensure the confidentiality of both hashed passwords and wordlists during transmission and processing.
- The system should optimize resource utilization, effectively leveraging the computing power of each worker node and minimizing idle time.

3. Architecture

Sprint 3

We split in 3 groups this sprint:

- Group 1 - will make the script
- Group 2 - will make the AKS cluster and will also deploy the script of group 1
- Group 3 - will start working on the scaling and security

The goal for this sprint is to have an application, script and some sort of distribution,

4. Design Details

- Application
 - Techstack
 - For the application the techstack is going to consist of 3 different components.
 - Golang with React for both the application and the hash cracking script.
- AKS cluster
 - Usage
 - The Kubernetes (AKS) cluster serves as a centralized hub for the management and execution of the hash cracking script. K8s streamlines the procedure by dividing work into more manageable, smaller parts known as pods. We can divide the submitted code files among several

pods for effective processing because each pod represents a single instance of a running process. By distributing pods among the cluster's available nodes automatically, Kubernetes ensures task balancing and efficient resource allocation. We are going to achieve this by monitoring CPU and memory usage of pods and nodes within the cluster. Then we are going to define thresholds for resource utilization, such as CPU and memory usage percentages. We can then use this to trigger scaling actions. K8s makes it simple to grow our application by adding or removing pods in response to demand, guaranteeing optimal resource usage and performance. Furthermore, Kubernetes has built-in mechanisms to restart failing pods automatically and relocate them.

5. User Interface Design

There will be a basic webpage where the users will be able to choose a hashing algorithm and put the hash value. Then we are going to have pods deployed with the hash script and they will start cracking the hash. After that the user will see the result of the cracked hash as a string.

6. Security

We are going to follow the best AKS/Azure/Docker security principles and make the security as better as we can. We are going to have things like strong networking rules, safe software, encrypted communication between the pods and also some Azure security enabled and configured.

7. Testing

TBD in the next sprints.

- Test plan: Testing objectives, strategies, and methodologies.
- Test cases: Detailed scenarios for testing each feature.
- Tools and frameworks for testing.
- Acceptance criteria.

8. Deployment

- The application with which the users will interact will be hosted inside the cluster with Docker, as well as the hash comparison script.
- The cluster will be created in Azure as AKS. The price for the cluster with 2 basic nodes is 18 USD a month.

We will make the application as automated and scalable as possible so that we can skip the most maintenance tasks. By adding tools like automated code deployment, scaling, monitoring and many more , our application will be easy to use and maintain since we will have most of the tasks automated.