Table of contents:

1. Introduction:

A project is composed of various different domains. Each domain has a set of functions to be carried out. In a project, all the resources available must be made available to all the domains and must be shared in a balanced way. Any imbalance in the resource distribution can cause delay in the completion of work leading to maximized cost. This problem addresses optimization of this kind of scenario.

2. Problem statement:

There is a cluster of computers. Each computer is referred as a node. You are the maintainer of the cluster. Your job is to execute various jobs that you receive for execution. You can transfer a job from a node on the cluster to some other node. It will require some network cost.

The jobs come in the batches. Each job will have its desired initial node on which it was thought to be run by its programmer. It will also require some processing power to run in a node. You have to process the jobs of the batch, i.e., you have to decide on which node the job should be executed. You can transfer the job through various nodes before assigning the final node.

Being the cluster maintainer, your aim is to optimally use the resources of the cluster. You want the load on the nodes of the cluster to be balanced. Also, you don't want to spend a huge networking cost in transferring the jobs in between the nodes.

2.1 Problem description:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

2.2 Procedure:

2.2.1 Identify the problem:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

2.2.2 Understand the problem:

To optimally use the resources of the cluster and want the load on the nodes of the cluster to be balanced and don't want to spend a huge networking cost in transferring the jobs in between the nodes.

3. Requirements:

Basically, requirements are statements that indicate what a system needs to do in order to provide a capability (i.e., utility or benefit.) Requirements are generally prepared during the early stages of a project's system development life cycle (SDLC.) There are many different SDLC methodologies that are used in practice. [1]

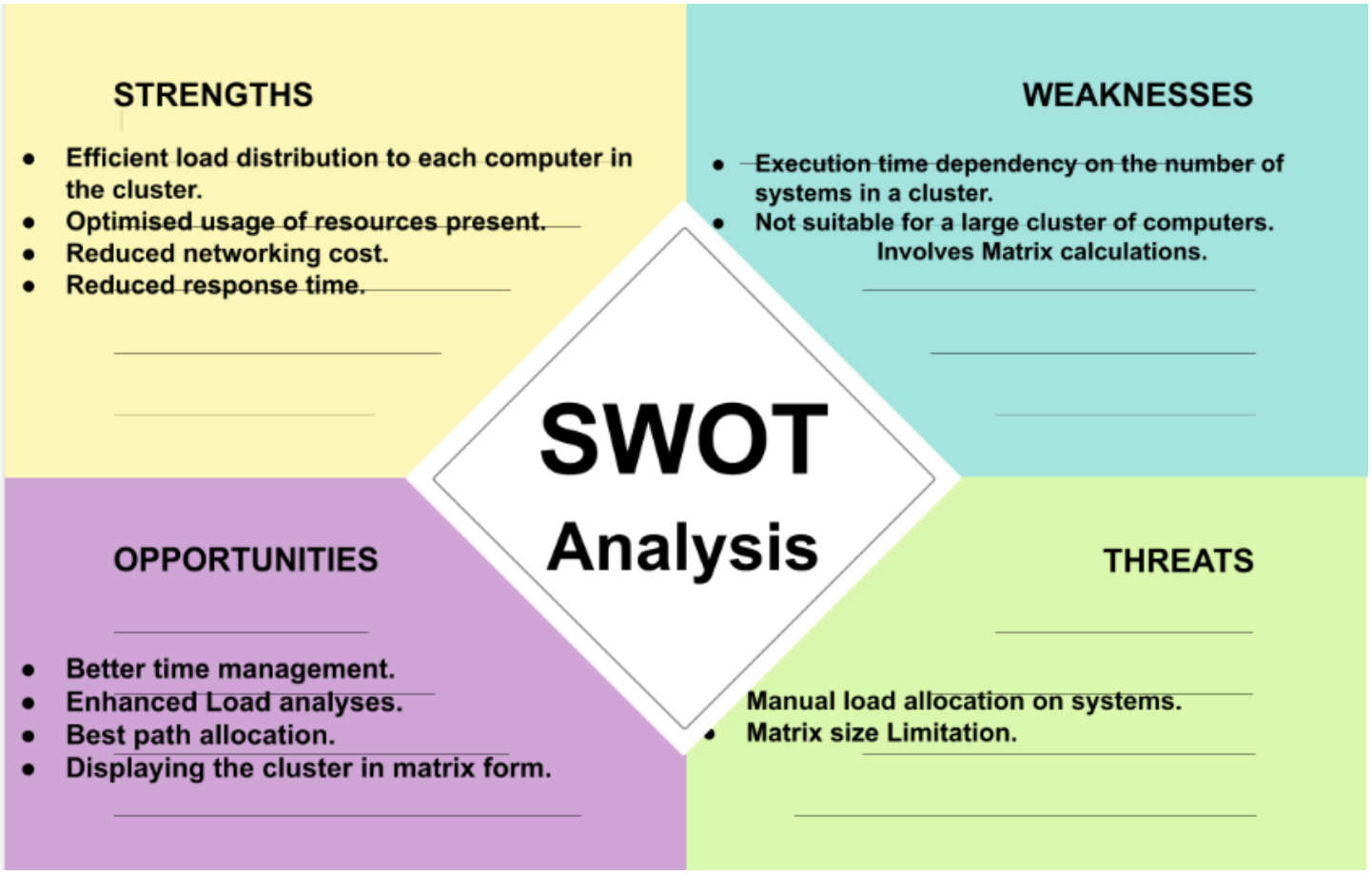
3.1 High level requirements:

Test ID	Description	Category	Status
HL_01	● Read number of computers in a cluster	Technical	Implemented
HL_02	● Read elements in the form of a matrix	Technical	Implemented
HL_03	● Able to identify the next computer (node).	Technical	Implemented
HL_04	● Display the minimum cost of the matrix	Technical	Implemented
HL_05	● Display the path	Technical	Implemented

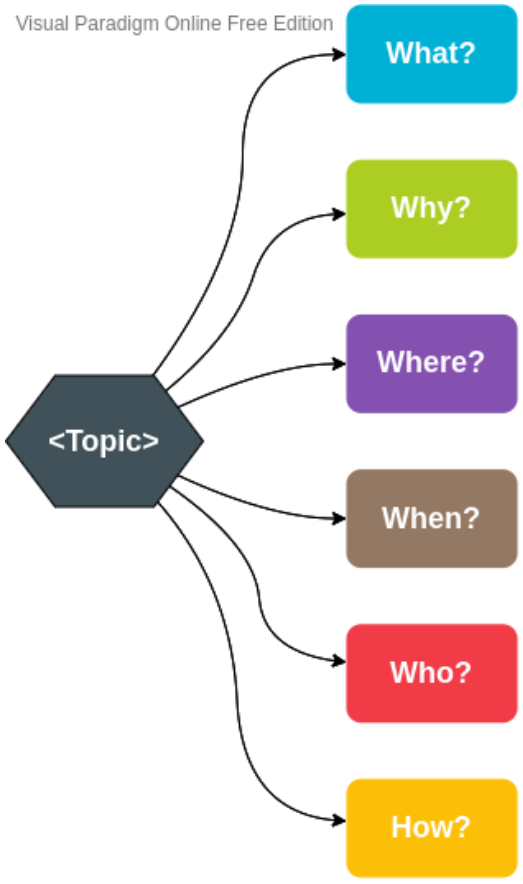
3.2 Low level requirements:

Test ID	Description	HLR_id	Status
LLR_01	● To identify the initial computer to start with among a cluster of computers.	HL_01	Implemented
LLR_02	● Choose the next computer for load enhancement.	HL_02	Implemented
LLR_03	● Evaluate the load from initial computer to present computer.	HL_03	Implemented
LLR_04	● Accordingly select the next computer for load allotment.	HL_04	Implemented
LLR_05	● Reading the number of clusters and reading the elements of the matrix is manual and done by the user.	HL_05	Implemented

4. SWOT analysis:



5. 4 W's and 1 H:



Effective use of resources, reduced run time and ease of functionality.

To find the best way to allocate the load among computers using resources available.

In system work allocation systems using resources.

While finding the best possible solutions among various solutions.

Problems with travelling sales men solution approach.

It works on the principle of finding the shortest path between initial node and final node.

Architecture:

The contents of Architecture include:

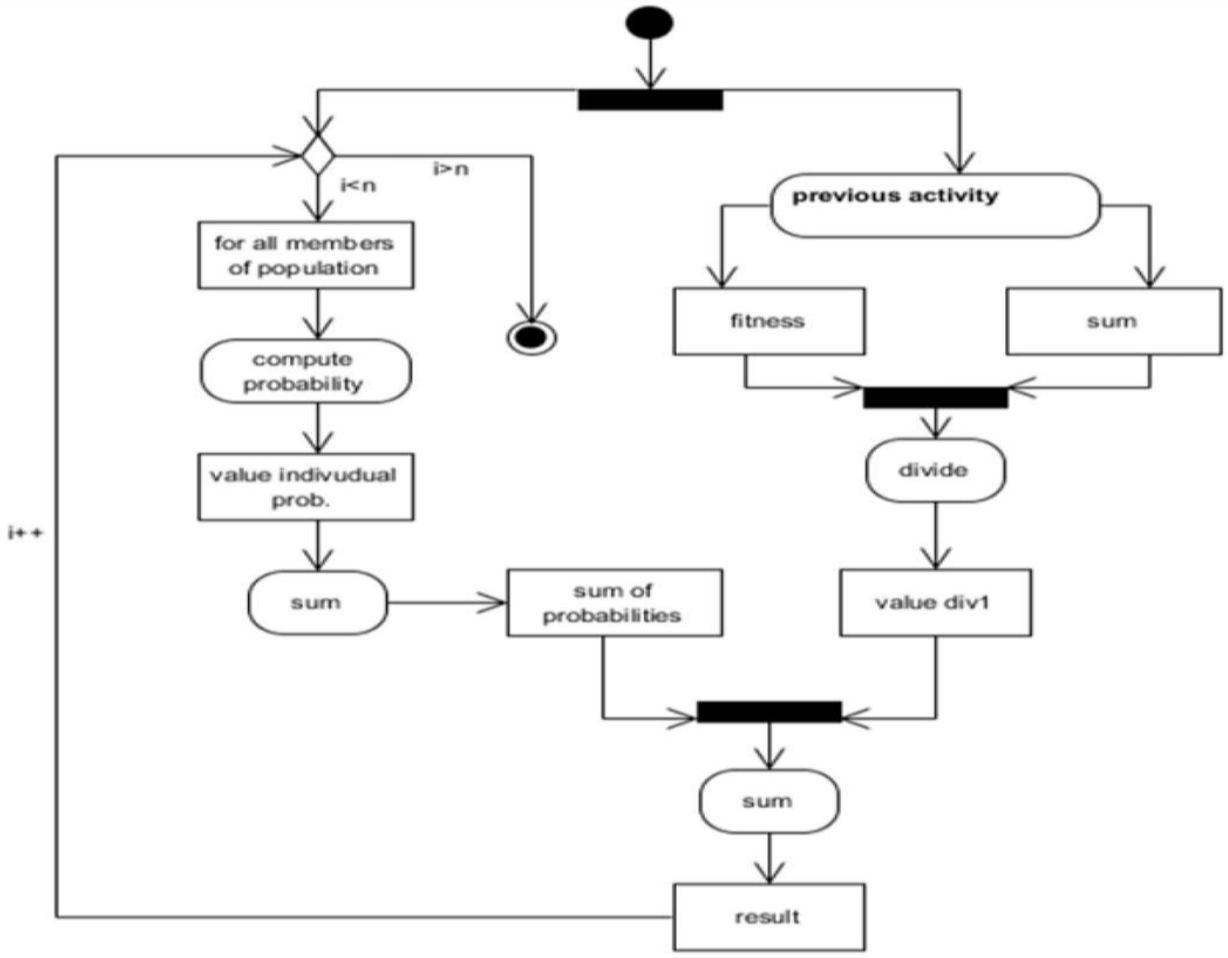
- Design
- Structural design
- Behavioral design

Design:

There are two types of designs; structural and behavioral.

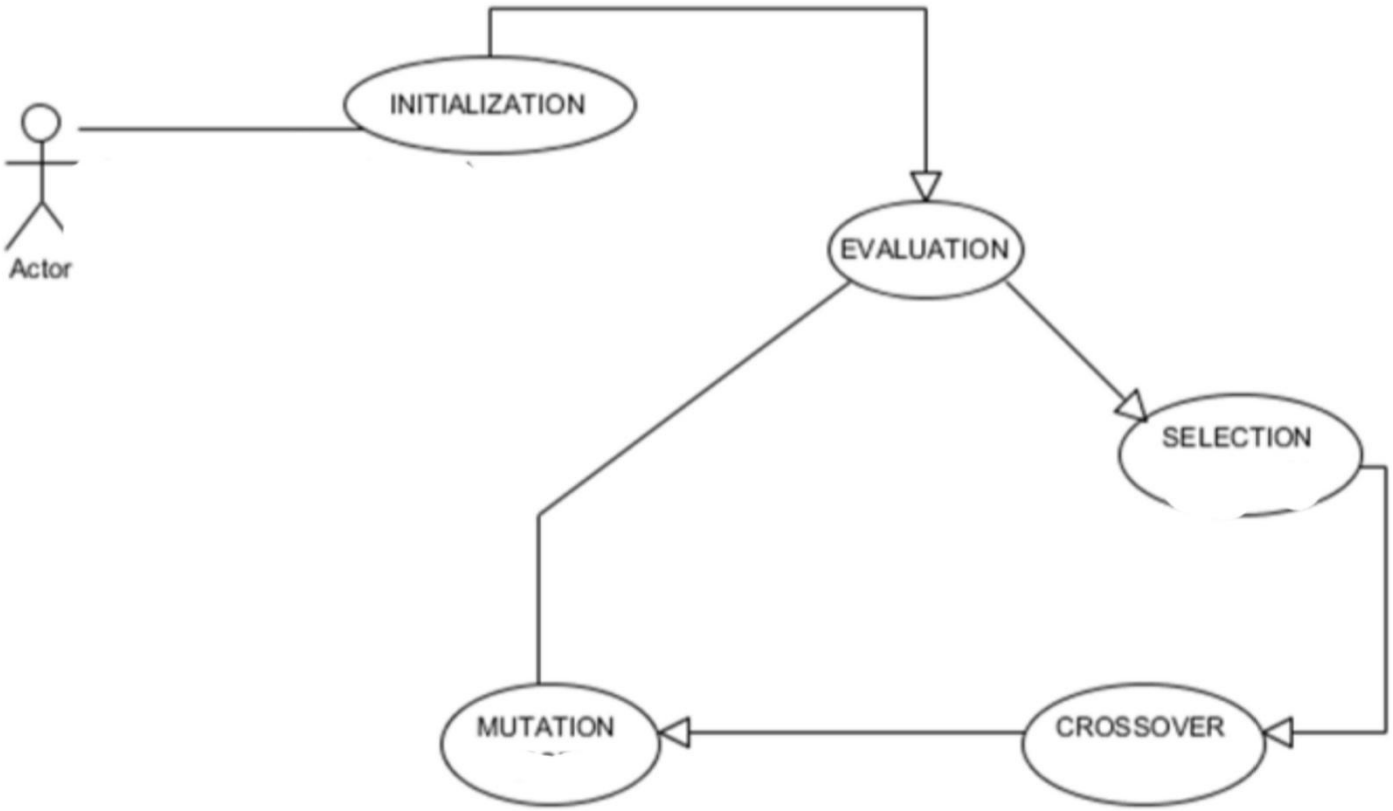
i) Structural Diagrams
ii) Behavioral Diagrams:

1) Activity Diagram

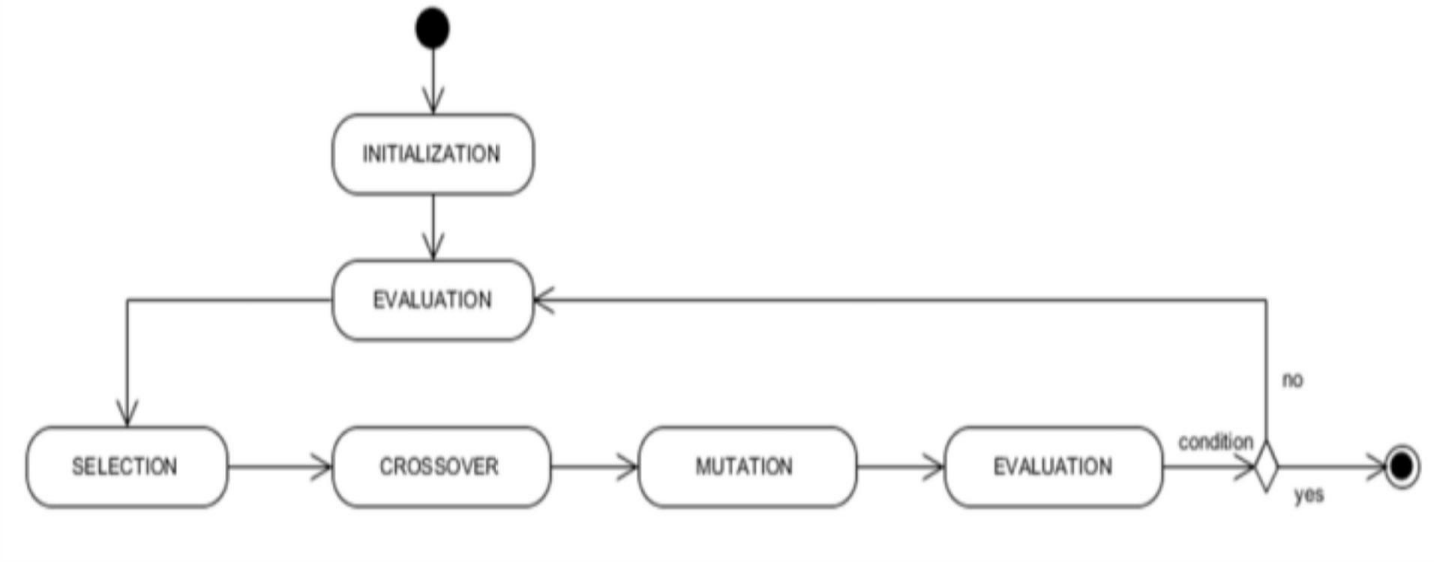


Use case diagram:

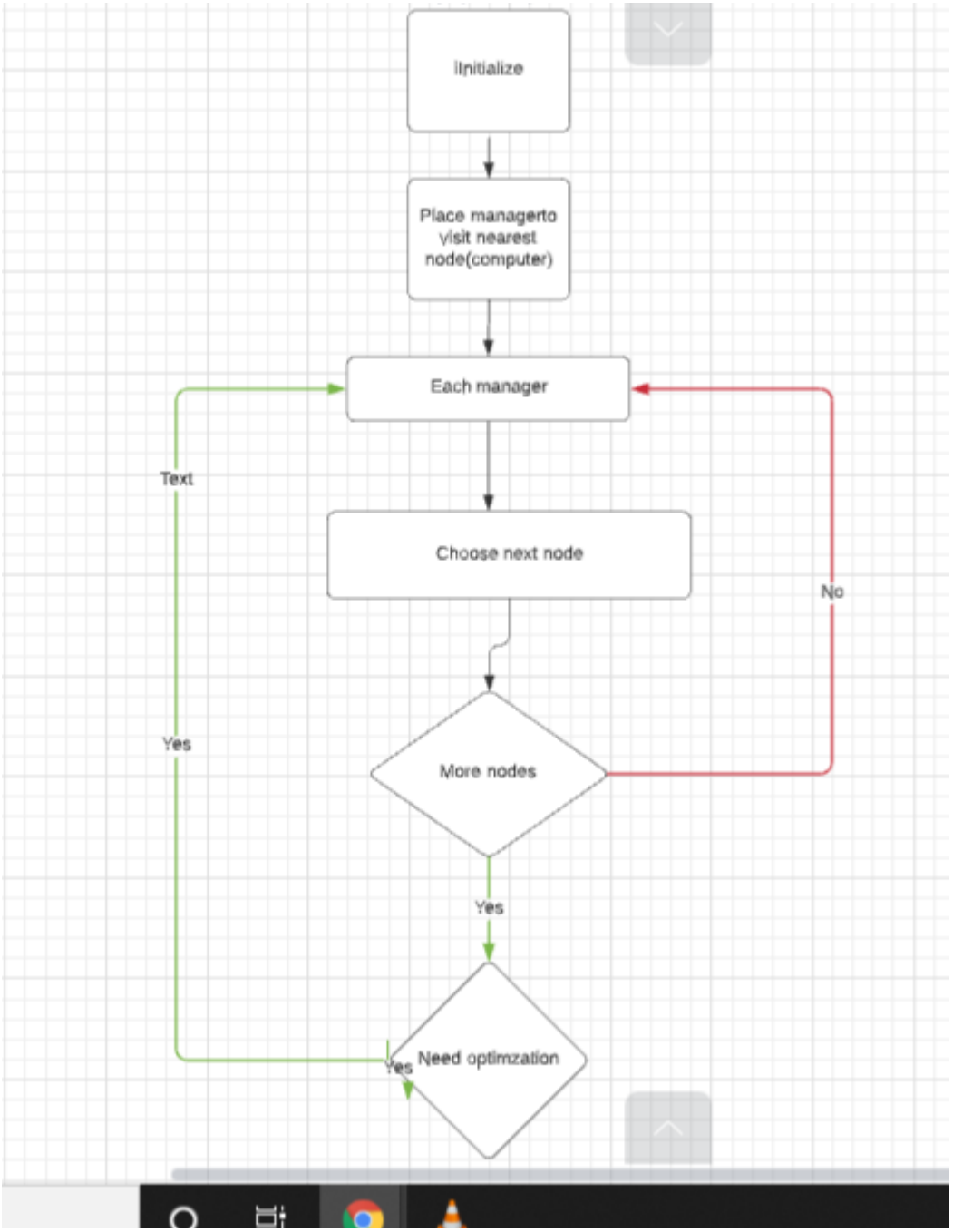
i)



ii)



Flowchart:


Initialize
Place managerto visit nearest node(computer)
Each manager
Text
Choose next node
No
Yes
More nodes
Yes
Need optimzation
Yes

Results and conclusion:

The objective of the problem statements has been achieved. Using the traveling salesman approach, the task of assigning the load to the cluster of computers was made easy. The resources present are used efficiently to find the best path.

References:

1] Solution approach: https://www.thecrazyprogrammer.com/2017/05/TSP.html

2] PDLC: https://stayrelevant.globant.com/en/pdlc-best-practices/

Learning Outcomes:

1)Multi-file programming

2)Improved my Programming skills

3)Familiar with SDLC methodology

4) Makefile knowledge

5) Familiar with the Linux environment.