Load Balancing Algorithms in Cloud Computing: A Study

Project completed during course CSD462
by Rimjhim Singh 1910110317
Under guidance of Dr Dolly Sharma

Contents covered in the paper

Introduction

Challenges and Evaluation Metrics

Classification of load balancing algorithms

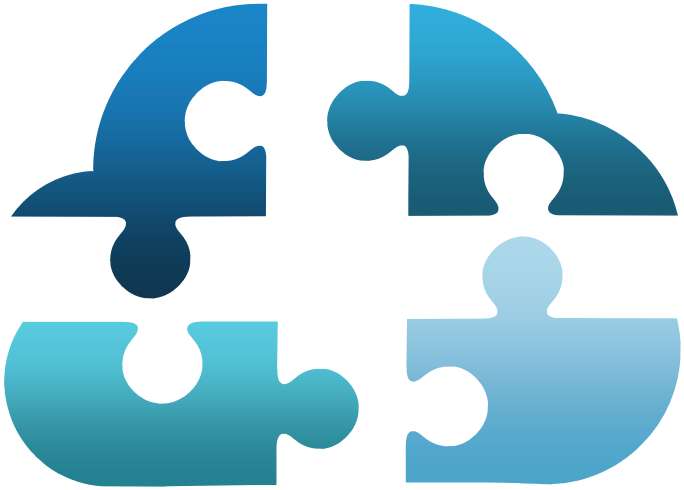
Load Balancing algorithms

Conclusion

Introduction

Pay as per use

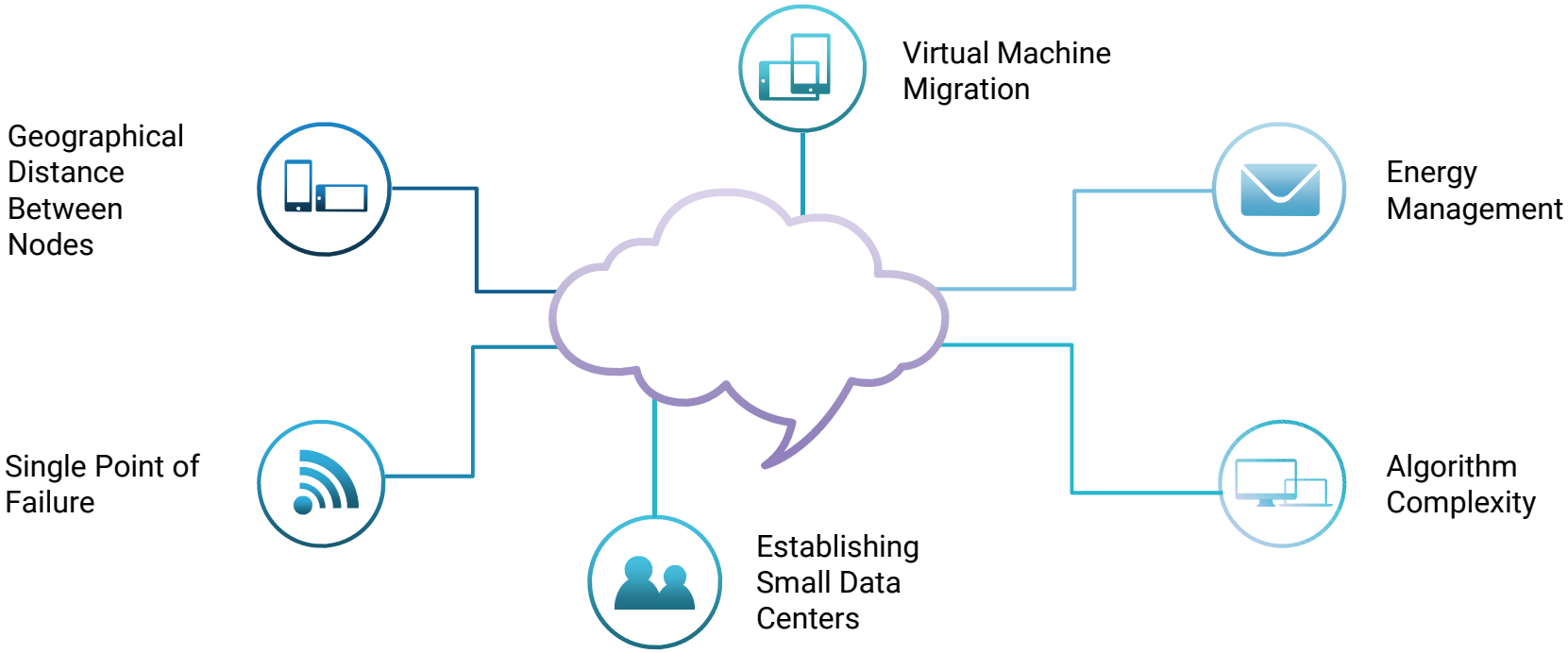
Maximum Services at min cost



Storage and network resources, application development frameworks, application management, and user interface

Remote services

Challenges Faced



Evaluation Metrics for Load Balancing Algorithms

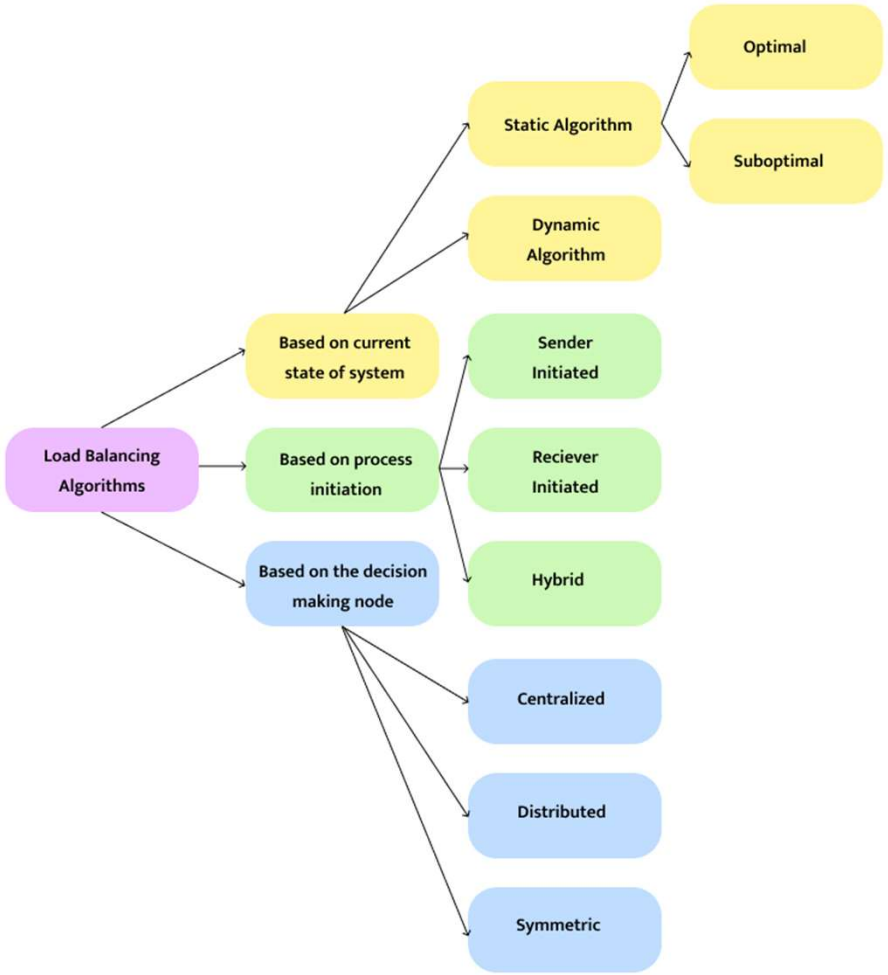
1. Throughput: Processes per unit time
2. Internal Failure Adaption: How adaptive it is
3. Response Time: Should have min response time
4. Scalability: How scalable it is
5. Utilization of resources: Optimal utilization
6. Overhead: How network overhead impacts performance
7. Performance: Overall performance

Classification of load balancing algorithms

Based on current state of system:
1. Static
2. Dynamic

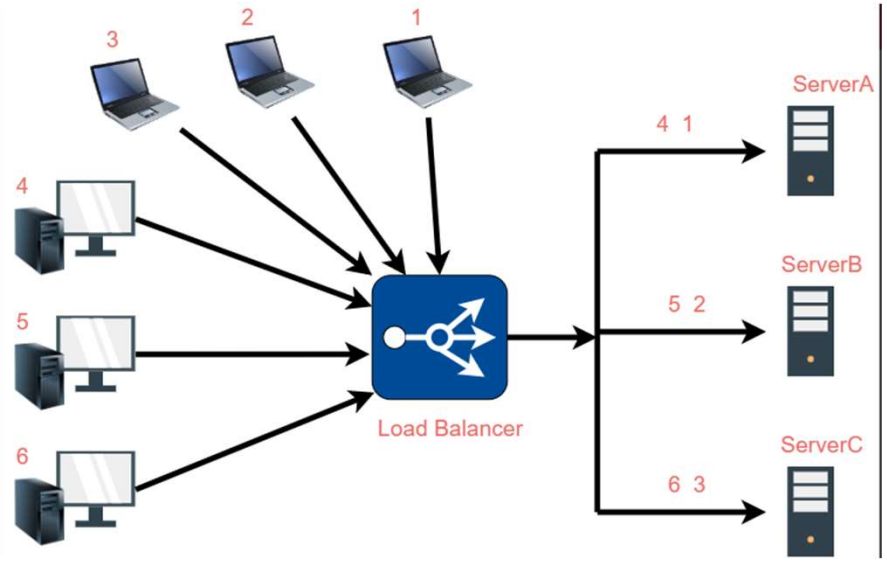
Based on process initiation
1. Sender Initiated
2. Receiver Initiated
3. Hybrid

Based on the decision making node
1. Centralised
2. Distributed
3. Hierarchical



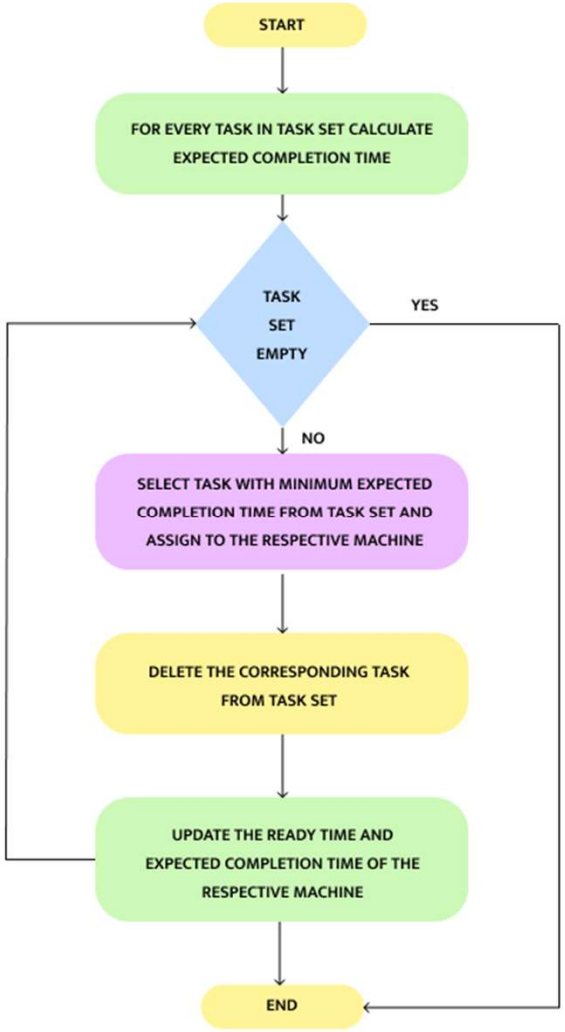
Classification of load balancing algorithms

Round Robin Algorithm



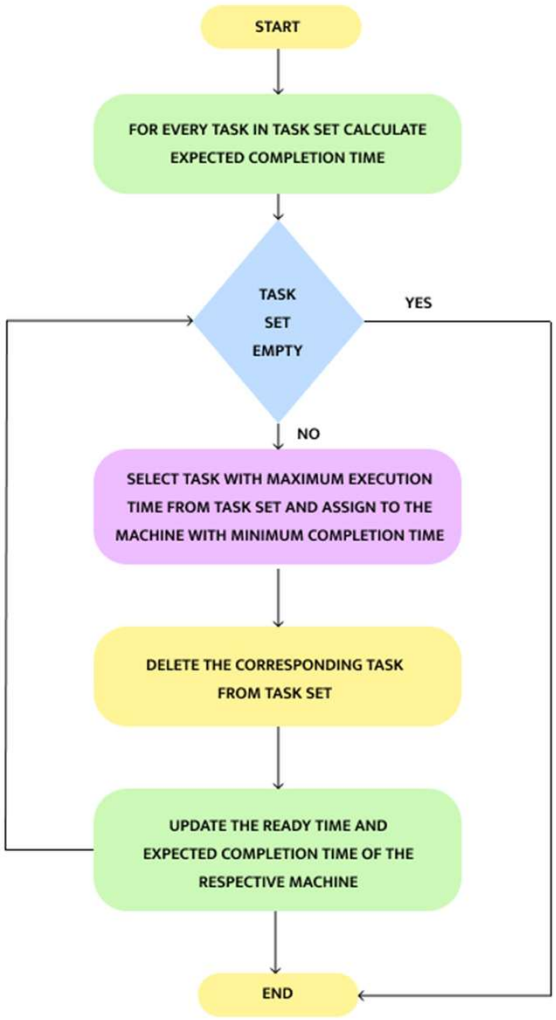
Source: Link

- Static Algorithm
- Uses time quantum concept to distribute tasks
- First load is selected on a random basis and the job is allocated to other nodes in a circular manner.
- Not adaptive
- A server might be overloaded while others are sitting idle
- Resource allocation and task distribution is not efficient.



Min-Min Algorithm

- Static Algorithm
- Minimum completion time and minimum execution time are both considered in this algorithm
- The task is assigned to the node which has the minimum execution time
- The problem of starvation is prevalent in this algorithm
- The tasks with the highest execution time have to wait for an unspecified amount of time



Max-Min Algorithm

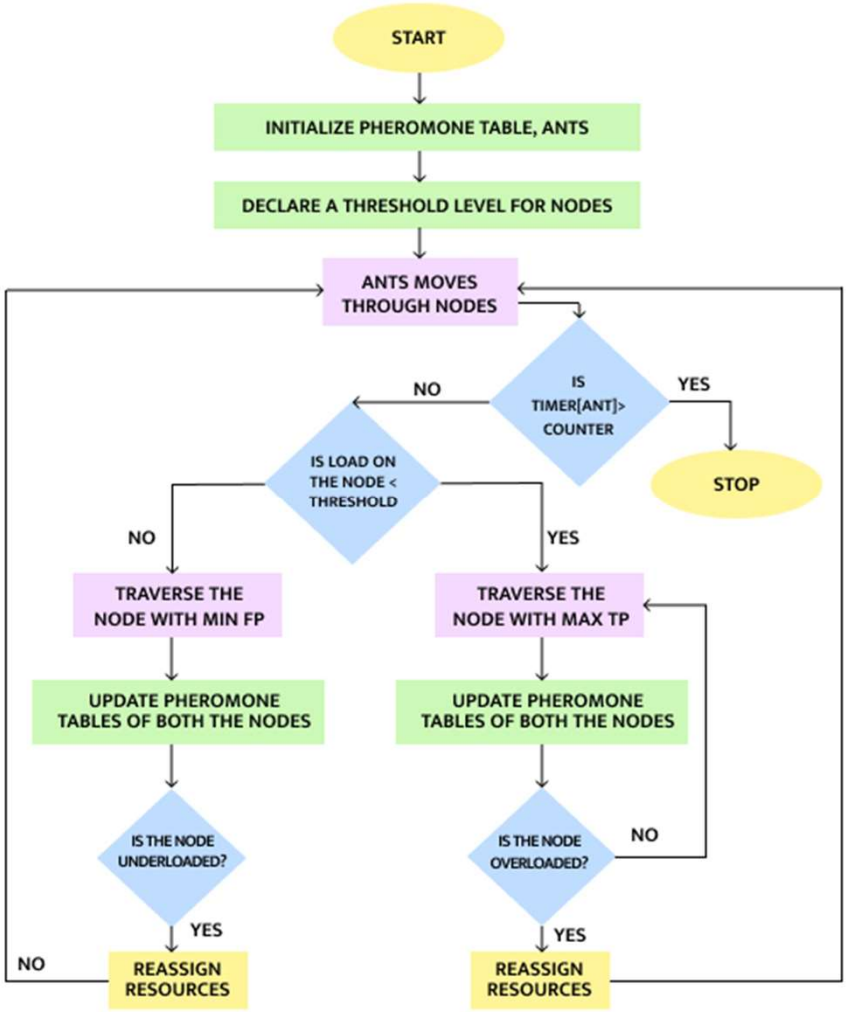
- Static Algorithm
- Same as the Min Min algorithm
- Difference is that task with maximum completion time is hosen and is assigned to a machine with minimum completion time
- Performance is better when there are more number of shorter tasks then longer tasks

Opportunistic Load Balancing

- Static Algorithm
- Was proposed for better resource utilization and response time
- All nodes are kept busy during execution of the task
- Request processing is slow because execution time of the tasks are not calculated, leading to a bottleneck situation even if some nodes are free

LBMM

- Combination of opportunistic load balancing and load balancing min min algorithm
- More efficient utilization of resources and task proficiency is increased.
- Three levels: Request Manager, Service Manager, Service node.
- Incoming task is broken down into subtasks and assigned to nodes
- Depends on availability of nodes, memory availability and CPU capacity.



Ant Colony Algorithm

- Inspired by ant colonies that work together efficiently in the search of food.
- In this technique load is balanced as well as make span is reduced.
- The assigned tasks hold the same weights/priority
- Fault tolerance is not considered.
- The assumption is that all tasks are mutually independent and computationally intensive.

Honey Bee Behaviour

- Honey bees find food, broadcast by dancing
- The tasks are treated as bees and the virtual machines are treated as food resources
- If any virtual machine is over-loaded, it will migrate the task to an underloaded VM.
- After the migration happens, information about load on the machine as well as available tasks with their priorities is broadcasted.
- A task with high priority chooses a VM with minimum number of priority tasks, processed quicker.
- Keeps track of the priority of different tasks.
- Throughput is maximised and response time is minimised.

Throttled Load Balancing

- Dynamic Algorithm
- The task manager makes a list of virtual machines.
- By using the list, client request allotted to the relevant machine.
- A configuration table of virtual machines and their statuses is maintained and is used for balancing the load.
- When a request is received from the datacenter to allocate the VMs, load balancer chooses the VM in the list of available virtual machines.
- If VM not found then this request is added to the queue and waits for a VM to become available.
- Good resource utilization and performance.
- The drawback is that all VMs need to be searched and if it not efficient if the idle VM is at the bottom of the list.

Carton

- Dynamic algorithm
- To increase performance and distribute the workload equally among all the servers, work-load is dynamically assigned

Genetic Algorithm

- Dynamic Algorithm
- Make span is minimized
- A population of possible solutions to the problem at hand is generated and is allowed to evolve for multiple generations.
- This leads to better solutions.

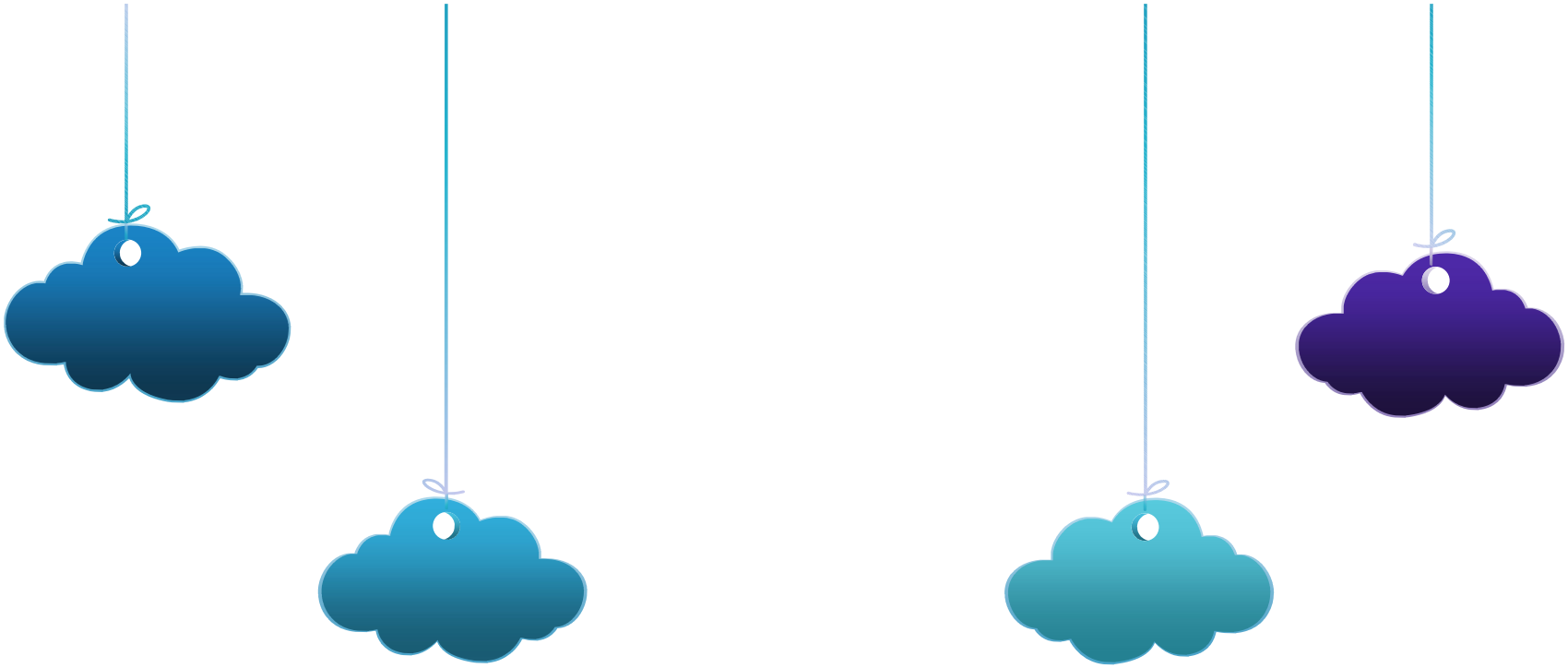
Load Balancing Algorithm	Advantages	Disadvantages
Static Load Balancing	Load Balancing decision is made during compile time Traffic is divided equally among nodes There are few complexes	Used only where load variaitons are few Cannot handle load changes throughput runtime
Dynamic Load Balancing	Load balancing takes place at run time Fault tolerance is there Only current state of system is considered	Nodes need constant check This technique is considered more complex
Round Robin	Uses fixed time quantum Easy to understand Gives better performance for short CPU burst Uses priority in running and arrival time	Longer time for larger tasks Task distribution is not efficient Not flexible to its environment
Min-Min	Completion time is short Handles smaller tasks most efficiently	Difficult to predict machine and task variations Starvation
Max-Min	Works better because requirements are known beforehand	Long time taken for task completion
Honey bee	Throughput increased, response time decreased	High priority tasks can't work without VM.
Ant Colony	Information collection is faster Tasks are independant Computationally intensive	Network overhead makes seraching take longer
Carton	Good performance Low communication required	Lower costs required
Throttled Load Balancing	Good performance Uses lists to manage tasks	Tasks have to wait
Genetic Algorithm	Better performance Better degree of load balancing	Low scalability and availability Overhead not evaluated Bottleneck

Comparison of different Load Balancing algorithms

Algorithm	Fairness	Response time	Throughput	Overhead	Fault Tolerance	Performance	Resource Utilization	Speed	Complexity
Static	Yes	Fast	High	N/A	No	Fast	High	Fast	Low
Dynamic	No	Slow	High	High	Yes	Slow	High	Fast	High
Round Robin	Yes	Fast	High	High	No	Fast	High	N/A	Low
Min-Min	No	Fast	High	High	No	Fast	High	Fast	Low
Max-Min	No	Fast	HIgh	High	No	Fast	High	Slow	Low
Honey Bee	No	Slow	High	Low	No	Slow	High	Fast	Low
Ant Colony	No	Slow	High	High	N/A	Slow	High	Fast	Low
Carton	Yes	Fast	High	N/A	N/A	Fast	High	Fast	High
Throttle	No	Fast	Low	Low	Yes	Fast	High	Fast	Low
OLB+LBMM	No	Slow	High	Low	No	Fast	High	Slow	High

Conclusion

1. Studied the various classifications of different load balancing techniques
2. Compiled resources to better understand load balancing algorithms
3. Studied the advantages and disadvantages of presented load balancing algorithms
4. Compared the algorithms based on evaluation metrics discussed.



Thankyou!