Name – Anshul Choudhary
Roll No – 17CS10005

Cloud Load Balancing Algorithms Comparison

Algorithm	Environ ment	Mode (Online/ Offline)	Parameters	Advantages	Challenges
Opportunistic load balancing (OLB)	Static / Dynamic	Online	● Execution Time	● Increase host machine utilization	● Poor make-span
Minimum Execution Time (MET)	Static / Dynamic	Online	● Execution Time	● Improves make-span of the system	● Does not consider machine ready time. ● It also shows load variations in machines.
Minimum Completion Time (MCT)	Static / Dynamic	Online	● Ready to Execute Time ● Expected Execution Time ● Least Completion Time	● Improves make-span of the system	● Starvation
Min-Min	Static	Offline	● Response Time	● Smallest completion time value. ● Outperforms when small tasks are greater in number	● Starvation ● Machine and task variation can't be predicted.
Load Balanced Improved Min-Min (LBIMM)	Static	Offline	● Response Time	● Optimizes the make-span and resource utilization	● Starvation
Min-Max	Static / Dynamic	Offline	● Waiting Time	● The task with MCT is executed first	● Starvation
Genetic Algorithm (GA)	Static	Offline	● Process utilization	● Finds best-fit solution ● Getting better efficiency of the system ● Decreasing task times ● Better Resource Usage	● Assumes that the jobs are of the same priority ● Tiny Throughput ● No power saving ● Missing the scalability
Simulated Annealing (SA)	Static / Dynamic	Online	● The temperature of the cloud resources	● It avoids local minima and finds the global optimized solution. ●	● Considers cloud resource parameters rather than task parameters
A-star Search	Static	Offline	g - the movement cost to move from the starting point to a given point, following the path generated to get there. h - the estimated movement cost to move from that given point to the final destination	● It is optimally efficient, i.e. there is no other optimal algorithm guaranteed to expand fewer nodes than A*.	● A-star Search Algorithm doesn't produce the shortest path always, as it relies heavily on heuristics/approximations to calculate – h
Tabu Search (TS)	Static	Offline	Previously visited locations	● This method uses adaptive memory that performs a more elastic search behavior. ● Allow non improving solutions to be accepted in order to escape from a local optimum	● Local search procedure ● Aspiration conditions ● Maximum size of tabu list ● Stopping rule.
Switching Algorithm	Dynamic	Online	● Nodes in the cloud system	● Helps in migration of tasks or VMs ● Can achieve the fault tolerant property	● Does not guarantee maximum network lifetime