ABSTRACT

Agents are sophisticated computer programs that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems. Increasingly, however, applications require multiple agents that can work together. A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. Multi-agent systems comprises of agents and their surroundings. In this paper, a review on research based on load balancing and multiple agents has been discussed. From the review, it has been found that most of the techniques have neglected failover issue in dynamic load balancing. Moreover, the usage of multi-agents has also been neglected the most of the existing researchers.

Keywords:- Multi-agents, Load Balancing, Multiple agents

1. INTRODUCTION

A multi-agent system (M.A.S.) is a computerized system composed of multiple interacting intelligent agents within an environment [1]. Multi-agent systems can be utilized to solve issues which are difficult for each agent or a monolithic system to resolve issue. Intelligence may incorporate few methodic, functional, procedural or algorithmic searches, evaluate and processing approach. Although there is substantial overlap, a multi-agent system is always not the same as an agent-based model. The aim of an Agent based model is to find for explanatory insight into the collective behavior of agents obeying simple rules, characteristically in natural systems, more willingly than solving specific practical or engineering problems. The terminology of ABM tends to be used more often in the sciences, and MAS in engineering and technology [1] where multi-agent systems research may deliver an appropriate approach include online trading, [2] disaster response, [3] and modeling social structures [4]. In artificial intelligence research, agent-based systems technology has been hailed as a new paradigm for conceptualizing, designing, and implementing software systems. Agents are sophisticated computer programs that act autonomously on behalf of their users, across open and distributed environments, to solve a growing number of complex problems. Increasingly, however, applications require multiple agents that can work together. A multi-agent system (MAS) is a loosely coupled network of software agents that interact to solve problems that are beyond the individual capacities or knowledge of each problem solver. Multi-agent systems comprises of agents and their surroundings. Characteristically multi-agent systems research refers to agents in software. However, the agents in a multi-agent system could equally well be robots, [5] humans or human teams. A multi-agent system may include combined human-agent teams.

Agents can be divided into different types:
- Passive agents[6] or agent without goals (like obstacle, apple or key in any simple simulation)
- Active agents[6] with simple goals (like birds in flocking, or wolf–sheep in prey-predator model) Or very complex agents (like cognitive agent, which contain complex calculations)
Environment also can be divided into: Agent environments can be arranged as per the properties like: accessibility (depending on if there is probability to gather entire information of the environment), determinism (if an action has been performed in the environment causing a definite effect), dynamics (total entities that influence the surroundings in the moment), discreteness (either the number of possible actions in the environment is finite), periodicity (whether agent actions in certain time periods influence other periods) [7] and dimensionality (either spatial characteristics are significant factors of the environment and the agent considered space in its decision making). Agent actions in the environment are characteristically mediated via an appropriate middleware. This middleware offers a first-class design abstraction for multi-agent systems, on condition that means to govern resource access and agent coordination.

2. BENEFITS OF MULTI-AGENT APPROACH
MAS has the following preferences over a single agent or unified methodology:
- MAS appropriate computational resources and capacities over a network of interconnected agents. Though a centralized system may be plagued by resource constraints, execution bottlenecks, or critical disappointments, MAS is decentralized and decentralized and subsequently does not experience the ill effects of the “single point of failure” issue associated with unified frameworks.
- MAS take into account the interconnection and interoperation of different existing legacy systems. By building an operator wrapper around such frameworks, they can be consolidated into an agent society.
- An MAS models issues regarding autonomous interacting component-agents, which is ended up to be a more common method of representing task allocation, team planning, user preferences, open environments, and so on.
- MAS efficiently recover, filters, and globally facilitate data from sources that are spatially appropriated.
- MAS give solutions in circumstances where expertise is spatially and temporally circulated.
- MAS upgrades general framework execution, particularly along the measurements of computational productivity, reliability, extensibility, robustness, maintainability, responsiveness, flexibility, and reuse.

2.1 Applications of Multi-Agent Research
Their applications cover a variety of domains like
- Maintenance of the aircraft
- Electronic book buying coalitions
- Military demining
- Wireless collaboration and communications
- Military logistics planning
- Supply-chain management
- Joint mission planning
- Financial portfolio management

3. LOAD BALANCING SCHEME
In order to decrease the time of simulation, the PCs required completing the jobs as close as feasible in time. The simulation time can be reduced all the way through the migration of the agents. The simulation consists of several steps having precedence relationship between them [7]. The total simulation time is total of the simulation time of every step. The SCA (Simulation Control Agent) initiates load balancing operation when the standard deviation of the simulation times of the PCs becomes higher than a threshold value. If load balancing is required, the agent in a container taking longer time than $t_{min}$ during one step operation is stimulated to other container. The SCA repeatedly executes the operations until the simulation is over [7]. The TMAs of each container check the time when the agents finish one step of simulation. Then the SCA finds the standard deviation of the simulation times of the agents. If it is higher than the threshold, the SCA initiates load balancing. When load balancing is initiated, the TA receives the data on the resource utilization from the HRMA. Then it finds the CPU utilization of every agent during one step and transfers the result to the SLMA. Subsequently, the SLMA finds the CPU usage of each container. Then the SLMA selects the agents to migrate based on the CPU usage data. Finally, the SLMA migrates the agents of the container of excessive CPU usage to those of low CPU usage [7].
4. LITERATURE REVIEW

Z. Zhang et al. [1] proposed a model based method to predict and compute the resource necessity of every virtual machine, and utilizing this model to outline a load-balancing structure in IaaS Cloud. The contribution of this paper incorporated a model that figures the load and evaluates the resource necessity of virtual machines in IaaS Cloud, a scalable system for load-balancing which utilized their resource necessity forecasting model. W. Haque et al. [2] proposed a new load balancing protocol, Earliest Completion Load Balancing (ECLB) that observed the conveyance of framework load and network latency trying to progressively make and keep up an equally loaded system, even on account of heterogeneous nodes or substantial load conditions. They have evaluated its execution against other load balancing protocols under typical and great circumstances. In situations of high load or high latency, ECLB is competitive with protocols that would overall be better suited to only one or the other. In heterogeneous situations, ECLB beats all different protocols. In addition to different factors, the completion of transactions before their deadlines relies on both the network latency and load on every node. Y.F. Wen et al. [3] Considered the handling of network and system load balancing to obtain greater performance. To appoint the assignments to the same kind of nodes along the connections with least processing and transmission delays were subjected to the limits of nodes and links. Three assignment task plans FCFS, Min-Min, and Min-Max were adopted alongside dynamic clustering, which was a strategy to gather the same sort type of cloud servers. Their study changed the variables manipulated with the number of nodes and the number of tasks and records the maximal end-to-end delay, normal end-to-end delay and fairness index, to analyze the load balancing outputs. The results demonstrated that the Min-Max combination with dynamic clustering had a better impact. C.C. Li et al. [4] Proposed a novel decentralized load balancing architecture, known as two-level decentralized load balancer. This distributed load balancer took benefit of the decentralized architecture for giving scalability and higher availability capabilities to administer more cloud clients. They additionally proposed a neural network-based dynamic load balancing algorithm, known as neural network-based dynamic weighted round-robin, to dispatch an extensive number of requests to various VMs, which were really giving services. In nn-dwrr, they consolidated VM load measurements (CPU, memory, network bandwidth, and disk I/O utilizations) checking and neural network-based load prediction to modify the weight of every VM. Experimental results demonstrated that proposed load balancing algorithm could be applied to a large cloud datacenter. A. Brugues et al. [5] described the security structural architecture of MOSAIC, a protocol for clinical information exchange with multilateral agreement support. The blocks of the architecture were inferred from an arrangement of normal attacks that can be done to the protocol. The reasonable exchange problem of the protocol was examined introducing the administration messages that the agents must exchange in order to approve or not the utilization of information. Due to multilateral agreements, loops can show up in the transaction phase of the protocol. Dezheng Zhu et al. [6] Presented a consensus algorithm for the agents with two fold integrator dynamics. The conventional consensus algorithm for limited space is just applied into rectangular bouncing boundaries, not suitable for non-rectangular space. Hence, they presented the idea of the mirror speed and position matrix that not only could convert the discontinuous real velocity into the continuous mirror speed, but also can expand a bounded space into an infinite space, and introduced the saturation control in order to...
limit the input signals. The speed and position of multi-agents asymptotically join the same qualities. Lastly, the effectiveness of the proposed consensus algorithm was analyzed by numerical simulations. Y.J. Lee et al. [7] proposed a scheme reducing the simulation time of distributed simulation designed on multi-agent platform. To facilitate achieve their goal, the time management agent has been working along with other agents examining the hardware resources and developing the better utilization of them for reducing the simulation time by dynamic load balancing with the PCs. The experiment indicated that the proposed scheme substantially decreased the simulation time in comparison to the already existing scheme. M.e. Renda et al. [8] showed that it is conceivable to adjust system movement stack in a geographic hash table without changing the underlying geo routing convention. As opposed to changing the straight-line geo routing convention used to send a question from the hub issuing the inquiry to the hub dealing with the questioned key, they proposed “figure out” the hash capacity used to store information in the system, executed a kind of “burden mindful” task of key extents to remote sensor hubs. This inventive approach is instantiated into two particular methodologies: a diagnostic one, in which the end of the line thickness capacity yielding quasi perfect burden adjusting is logically described under consistency suspicions for what concerns area of hubs and question sources; and an iterative, heuristic approach that could be utilized at whatever point these consistency suppositions are not satisfied. So as to demonstrate reasonableness of our heap adjusting system, they have performed broad reenactments taking after practical remote sensor system arrangements demonstrating the adequacy of the two proposed methodologies in significantly enhancing burden adjusting and developing system lifetime. Reproduction results demonstrated that their proposed method attained preferred burden adjusting over a current methodology focused around changing geo routing. B. Rajodevic et al. [9] have introduced an examination of located issues for those heap adjusting calculations in this paper, as an arrangement stage for another burden adjusting model (calculation) recommendation. This calculation fused data from virtualized machine situations and end client involvement to have the capacity to proactively impact burden adjusting choices or sensitively change choice in taking care of discriminating circumstances. Burden adjusting models and calculations proposed in the writing or connected in open-source or business load balancers depend either on session-exchanging at the application layer, parcel exchanging mode at the system layer or processor burden adjusting mode. N. Lu et al. [10] researched the capability of giving intra-hour burden adjusting administrations utilizing amassed warming, ventilating, and cooling (HVAC) loads. A direct load control calculation has been exhibited. A temperature-need rundown strategy has been utilized to dispatch the HVAC stacks ideally to keep up client coveted indoor temperatures and burden differences. Practical intra-hour burden adjusting signs were utilized to assess the operational qualities of the HVAC stack under diverse open air temperature profiles and distinctive indoor temperature settings. The quantity of HVAC units required was additionally explored. The results showed that the intra-hour burden adjusting administration gave by HVAC burdens meets the execution necessities and turn into a significant wellspring of income for burden serving substances where the two-way correspondence brilliant network base empowers immediate burden control over the HVAC loads. M. Ajit et al. [11] displayed the dissection of three contemporary calculations in cloud examiner apparatus to resolution the issue of cloud burden adjusting as a readiness stage for new load adjusting system. A Weighted Signature based burden adjusting (WSLB) calculation has been proposed to minimize client’s reaction time. Further, this paper likewise gave the foreseen results the usage of the proposed calculation. W. Wang et al. [12] first found through examination that the essential explanation behind overhead happening in burden adjusting is the heap relocation, and afterward qualitatively gives the granularity recipe of moving load each one time, proposed another element burden adjusting calculation, characterized the four conceivable states of the hub, and talks about the execution principle of the calculation. At last, an examination is made between the working consequences of the calculation and those of other burden adjusting calculations and no heap adjusting through analysis, demonstrating that the heap adjusting calculation are superior to different calculations if there should be an occurrence of the hub with no-heap, diverse burdens, and distinctive information scale. I.shiyo et al. [13] depicted how to facilitate independent portable multi-executors, whose operators stochastically move over a limited asset comprising of cells as per move probabilities. They accepted that every operator can’t move to an objective cell possessed by executors more than the operators of a current cell and their executors had time-slack. At that point it perfect that each cells are constantly possessed by executors, due to the effective utilization of assets, and it is alluring to be the least expected number of cells not involved by operators. The principal result demonstrated that the asset usage of quickened portable multi-operators with suitable normal moving velocity gets to be higher than the asset use of versatile multi-executors without normal moving rate. At that point the asset use could be quickened to greatest by giving suitable moving velocity. The second come about demonstrates that there were the best moving rate of framework particular ideal to use the assets of independent stochastic versatile multi-operators. Z. Taoqing et al. [14] talked about a self-versatile collaboration technique focused around Multi-Agents in chain of command and dissected a dynamic model in which Multi-Agents were isolated into the primary Agents and sub-Agents in diverse layers. Every primary Agent was in charge of administration and coordination of close to various sub-Agents, which could adjust the heap, and synergistic collaboration between fundamental Agents or sub-Agents in same layer could keep administration
well. Multi-Agents stratified system could decrease information showed up for abnormal state, abbreviate the steering and with its self-governance and coordination of each one point, while maintaining a strategic distance from impediments because of single purpose of disappointment. The method could keep up the consistency of system topology between all Agents and guarantee the viability of directing progressively. It could likewise enhance system access directing shooting, furthermore enhance the nature of service

5. GAPS IN LITERATURE

- The failover issue in dynamic load balancing has been neglected in the most of the existing research.
- The use of multi-agents has also been neglected the most of the existing researchers.

6. CONCLUSION AND FUTURE SCOPE

Multi-agent based methods are extensively utilized in mounting scalable and elastic software services. This paper has focused on distributed computing which is based on multi-agent methods. The role of multi-agents in load balancing is very functional due to the intrinsic belongings of scalability and independence of agent system. Load balancing is significant subject in distributed computing which divides the jobs between the numbers of available processing units, i.e. divide the load among available processors in efficient manner. In this paper, review on load balancing and multi-agent system has been accomplished. It has been found that all the techniques lack some important measures. In near future, to accomplish the objectives in effective manner, the time organization agent has to be engaged along with other agents’ inspection the hardware resources. Further enhancement can be made using the failover based multi agents in distributed computing

REFERENCES