International Journal of Engineering & Technology, 7 (3.34) (2018) 206-209

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Multi-Agent -Master Resource Finding Engine for Fast and Efficient Dynamic Resource Finding in Cloud Computing

[1] B.Muthulakshmi, [2] K.Somasundaram

M.E., (Ph,D) ,
[1]Research Scholar, St. Peter's University, Department of Computer Science and Engineering, Chennai, TamilNadu
[2]Professor, Department of CSE, Arupadai Veedu Institute of technology,Kancheepuram.
**Corresponding author E-mail: soms72@yahoo.com*

Abstract

Resource finding is an enormous and tedious task in the cloud environment. There are various protocols suggested in existing approach to deal the resource finding. But from job scheduling to resource identification none of the approaches is available for best. It leads to a tailoring of functionalities in different phases like job scheduling, resource finding and resource identification. This paper analyses the automation from the job scheduling phase to resource allocation phase which includes, semantic model, optimization, Master Resource Finding Engine (MRFE). A decision-making mechanism called adjudicator is a control unit which collects input from all the above-said phase models and provides the best and efficient resource to the requirement. In this paper, a new model is provisioned to attain all the phases explained above and compared with resource simulations. The results were analyzed in terms of quality of resource, performance, resources acquired, patterns considered for semantic finding, interactive jobs, a workload of resources in normal and peak time

***Keywords*:** *Cloud Computing, Resource Finding, Software Resource, Job Scheduling, Semantic Model.*

1. Introduction

The cloud environment is to create a platform to provide an easy operation to the users. Integrating computing devices, through a network with people is called as Cloud computing. The sizes of the devices are from tiny sensors to extremely dynamic and influential. The environment of the Cloud networking is changed according to the people and their environments. Various numbers of clients are connected to a number of servers can provide different types of resources. The resources can be accessed by the clients through the servers. Resource's request, finding and accessing the request is presently used in mobile networks. It is much important to know that how a mobile user / normal user reaches his needed resource. Due to the dynamic nature of the Cloud computing, knowing the above-said problem is more important. Discovering a resource is a process of identifying and recognizing an appropriate resource among the available resources in a network environment and user can request and utilize a resource [1]. The node is the general terms which denote any client or server in the Cloud environment. A register or a folder is maintained to store the entire operations of the properties of any resources [2].

Resource finding is an emerging field in the area of ubiquitous computing. There are two kinds of resource finding is available, they are directory based and directory-less based [14]. The directory based finding is easier than the other one, whereas bottleneck problems arrive and make a single point of failure. Due to this reason and dynamic nature, directory-less based resource finding is used mostly in Cloud environments.

Nowadays, new Resource Finding Protocols (RFP) is designed to reduce the burden of administrative problems and increase the user ability in Cloud computing. During the interaction, the present status of the Cloud devices and resources should be known by the server while requesting the resource in the environment. One of the most serious problem to be challenged is discovering the resource in Cloud computing. While finding, providing privacy for user information and user data is the first one. The second challenge is determining the amount of knowledge a user has for resource finding. There are two different people in this scenario one is user request resource and the other one is resource providers. The next challenge is facing a problem with multiple requests and multiple resources from same location simultaneously. Cloud computing is an environment which interconnects heterogeneous devices in terms of hardware, software, resource providers and protocols used in the network. Since it is essential to design and develop a protocol which can be able to establish a huge set of facilities to discover relevant resources to the requests comes from the users.

In this paper, it is motivated to design and develop a protocol for resource finding statically as well as dynamically in an automatic manner. In this paper, the directory based resource finding model is used, where it has a dedicated component to maintain all the resource information and processes queries announcements. The state of the resources, clients and servers are maintained as soft-state. One of the main objectives of the resource finding in this paper is to reduce the computational complexity of client/resource/server. The resource finding uses the meta-information about the user, semantic-information about the resource and other information about the server and clients. The contributions of the proposed approach in this paper are:

Introducing a Client based Job scheduler to schedule the jobs from client in terms of time and priority.

1. The entire resource-finding, job assignment and acknowledging the end users is carried out by deploying a Master Resource Finding Engine (MRFE).
2. MRFE comprises of individual agents called distance matching agents (DMA), state matching agent (SMA) and Behavior Matching Agent (BMA) and Meta-Data Matching Agent (MDMA) to do resource found in a better manner.
3. Finally, an adjudicator takes a decision by analyzing the matching score produced by the MRFE.

2. Related Works

There are various mechanisms were proposed in the past research works for resource finding. Each individual research work carried out in exclusive ways of different challenges and there is no correlation with the whole. In this section, the earlier mechanisms briefly presented, in order to fetch the problem statement addressed in this paper. One of the significant and well-organized challenges in distributed computing is resource finding [3-8]. Cloud computing environment comprises of wearable, handheld, home-based and embedded devices, including normal computers with clients and servers [1]. The various challenges are discussed in home automation based Cloud computing [9] and the implementation of home automation network layer is discussed in [10]. Some of the resource matching techniques with existing protocols utilize only simple matching schemes [14].

3. Problem Statement

In cloud computing, semantic information is identified by various features like a number of jobs to be resourced in the queue, job scheduler and interaction between the jobs that are executed in different areas. Let us consider the Figure-1Cloud infrastructure and user jobs; it depicts how the users with their jobs are scheduled in the queue. These jobs are identified as high, medium, low priority jobs, with interactions if any. Internal user based jobs were linked to the other jobs due to dependency. These all jobs are not executed at a time, based on the different parameter like user1 may likely to execute the job by resource provided in applications. If user1 need to get the resource for his jobs, the cloud will check for the resource based on metrics and check its availability. Based on the resources available the resource gets assigned.

If a number of jobs in the queue are not resourced yet, there is a possibility for the starvation of job without getting resourced. This kind of situation is not acceptable in case of Cloud distributed systems. At any point in time, the user may request for a resource that should be primarily assessed and to be resourced. User1 is going to interact with several jobs that are dependent on various resources similarly other users will look into the same with various resources there is a possibility of state and its behavior identity match. Based on this, the current requirement is resourced and flushed up from the queue.

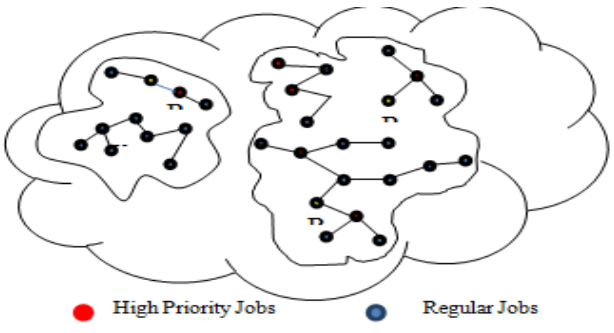

Figure-1: Cloud Computing Infrastructure

4. System Architecture

The proposed model architecture consists of three layers

- Request layer
- Business logic Master Resource Finding Engine Layer
- Resource Layer

Job scheduling based on the user request is generated in the request layer. For distributed system environment amount of job split up which is going to resource at different levels are identified by the job scheduler. There is N number of request is in the queue which given to the user database for validation and to keep track of the record accessed for future reference in the server used in this layer. In future say in any fault tolerance condition, this server acts as an intermediate recovery to move further. A business logic layer is a kind of three-tier architecture, optimizer fetches input job as a raw query and consolidates it with Master Resource Finding Engine. MRFE shares the query to job scheduler to handle for job split up. Job scheduler mechanism is the prime mechanism which is a client/server-based technology. It coordinates the flow between the finding engine, database, and scheduling server/client. Here in this proposed approach, client/server both will act as a job scheduler and resource will obtain in client and server both.

Most possible extend client will act as a job scheduler and distributes job to the finding engine. The job is nothing but a user request based on priority, wish or an entity. A semantic model is a transparent unit found in the business logic layer, which works on pattern matches for semantic finding. The semantic model gives the cumulative data which is used by pattern match data and given to next layer called resource layer. There is a bye pass provision, where job scheduler gives job directly to the resources if the resource is available and decided in job scheduler via adjudicator. Similarly, semantic model derives the pattern and semantic is discovered is given directly to adjudicator unit for allocation of resources. The third layer called resource layer contains storage unit called resource logs and data dictionary, also final resources will be assigned to the jobs. Compiled logs will be stored in the log directory and every update from upper layer logs will be updated automatically.

Resources include the semantic finding is based on location, where the cloud can afford in a Cloud environment. Time, which is considered as a topmost priority within a specific time the model has to support the resource. Quality of Resource in which a number of requests are to be satisfied in a varying time (generally in milliseconds).The metric resources are depicted in figure-2.

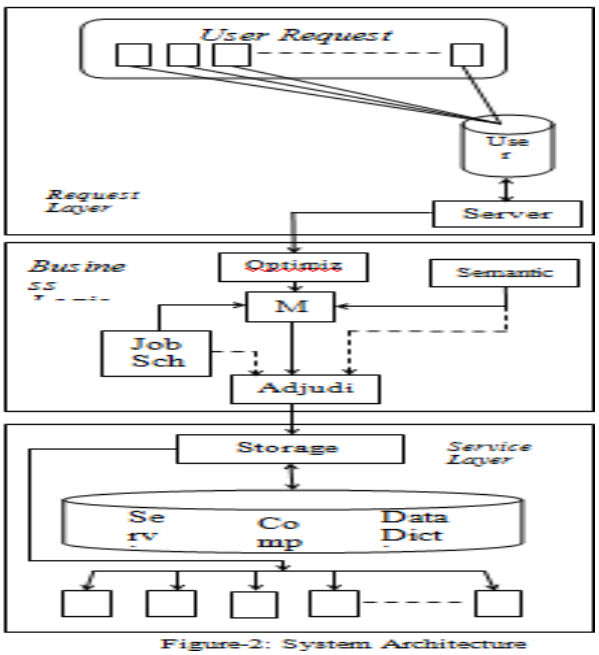

Figure-2: System Architecture

4.1 Master Resource Finding Engine

It is the superset which contains individual agents to simultaneously execute their algorithm and provide the results to the adjudicator. It uses four agents which satisfies the complete request of user queries and to obtain an efficient resource for the same. The metrics like distance is measured by the agent called SDE distance agent. It finds the nearby resource which is more relevant for query raised by the upper layer. SDE allows state matching agent which gives the state and its identity for attribute used for pattern matching. It is shown in the Figure-4. Behavior agent is to identify the master operations which are trying to induce in the task provided. It coordinates with the other tasks used for other agents. The last agent called SDE Meta data agent which describes about the internal details of the data considered for semantic finding. These four agents will execute parallel and gives the results to adjudicator. Adjudicator is a decision-making mechanism which collects data from the previous master resourcefinding engine and provides the resources.

5. Simulation Settings

The entire paper is simulated in a Lab installed with more than 500 systems. The systems are highly configured as Core i5, 2.64 GHz and above processor speed, 4 GB RAM and 500 to 1 TB hard disk drive. Over 500 systems, 25 systems are installed by server configuration software, 50 systems are installed with resources such as software as a resource, platform as a resource and infrastructure as a resource. Currency converter, calendar, astrology software, Language converter, plagiarism software are some more (plugins) are installed in another 15 systems as resources. The remaining systems are assumed as client systems and all the systems are connected to a LAN and Wifi (Airtel Broadband-High Data Rates with unlimited Downloading). The simulation is experimented by client-server-resource communication within a stipulated time interval such as 1 day. The experiment is carried out in a different manner like various numbers of requests generated by less number of clients and more number of clients before and after MRFE installed in resource systems. Also, the finding time is calculated with respect to the number of jobs requested from various numbers of clients and so on. According to the simulation-based experimental results are given in detail and plot in graphs in results and discussion section. The systems are highly configured as Core i5, 2.64 GHz and above processor speed, 4 GB RAM and 500 to 1 TB hard disk drive. Over 500 systems, 25 systems are installed by server configuration software, 50 systems are installed with resources such as software as a resource, platform as a resource and infrastructure as a resource. Currency converter, calendar, astrology software, Language converter, plagiarism software are some more (plugins) are installed in another 15 systems as a resources. The remaining systems are assumed as client systems and all the systems are connected by a LAN and Wifi (Airtel Broadband-High Data Rates with unlimited Downloading). The simulation is experimented by client-server-resource communication within a stipulated time interval such as 1 day. The experiment is carried out in different manner like various numbers of requests generated by less number of clients and more number of clients before and after MRFE installed in resource systems. Also the finding time is calculated with respect to the number of jobs requested from various numbers of clients and so on. According to the simulation based experimental results are given in detail and plot in graphs in results and discussion section.

6. Results and Discussion

The proposed system model is tested under all conditions with respect to metrics, quality of resource, performance, resources acquired, and semantic finding; inter communication jobs, workload of resources etc. The simulation is carried out in a LAN based network, where number of resource system is identified as 1 to 100, for resource based system 25 is provoked to handle all simple protocol resources generate by user system. Logical servers used for business logic called MRFE, Adjudicator, Semantic model, Job scheduler, stored manager resource; log servers are separately identified and interlinked with the setup. Initially all interactions are identified and interface protocol is tested via remote method invocation with all the servers and clients. Selected operations are accessed through interface and component based to the respective nodes. Here nodes are representing the system used in the network. MRFE is a server triggers in daemon process and runs throughout the model. Job scheduling server creates job and placed in the common queue identified. Along with resource manager job scheduler identifies common interaction between the jobs and assigns priority to the job and stores in the server assigned for Job Schedule Server. Jobs are assigned with identity number in first round of algorithm, J1 to J99 jobs are created with different request access resource and maintained in the queue. Optimizer unit in the model considered it and given to scheduler. Job scheduler takes care and filter the valid job and given to MRFE. Semantic model runs in the main server considered the job and given to the algorithm. This finding for the resources happened is mapped to the existing methodologies. In next level of iterations number of resources is raised and to the maximum extent 1000 resources is depicted into a long run. It is shown in Figure-7; finding time is represented in milliseconds. The naïve database search is stored in an organized way; hence for each resourcefinding is in slope form. Though compare to existing approach, within the specific time resource is obtained from MRFE with a KNMP pattern matching algorithm. The number of resource request is considered of 10 iterations with 100 requests of each. Job scheduler uses client as a scheduling agent and distributed these 1000 jobs to 4 servers where each server consists of 250 jobs ordered in random manner by job id. Server1 is handling 1 to 250 jobs, server2 is responsible for 251 to 500 jobs, server3 is for 501 to 750 jobs and server4 is for 751 to 1000 jobs. These servers are assigned with initial probability access index to search for the semanticfinding and to provide the rating scale. Based on the value, adjudicator is assigned to take decision for resource assigned, with each probability index how resource is assigned to discover is depicted in Figure-8. Here for consideration 30%, 60% and 90 % of finding is obtained with the system model.

Here, requirements is resourced by MRFE with the first level of filter in scheduling algorithm hence in valid requirements is called off in initial level itself. Requirements resourced are shown in Figure-9. For the possible extend proposed model is proved that satisfies the entire job in forms of requirement is resourced.

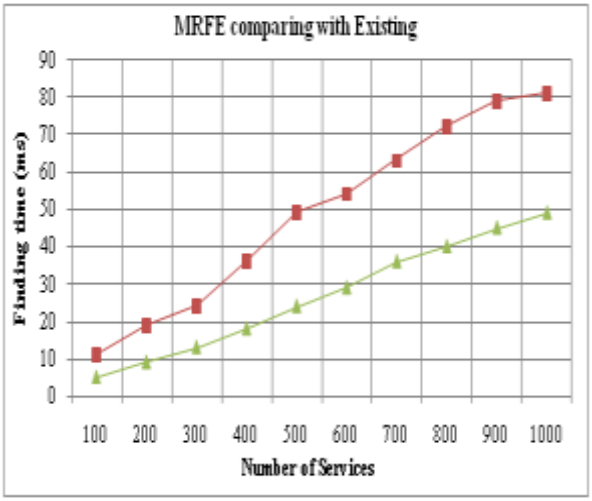

Figure3: Finding Time Comparison between Existing and Proposed

Figure4: Probability of resource Finding

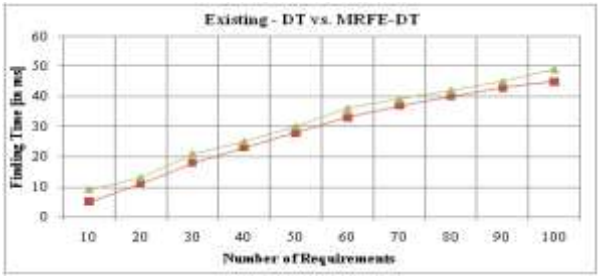

Figure5: Requirements Resource with respect to Existing and proposed approach

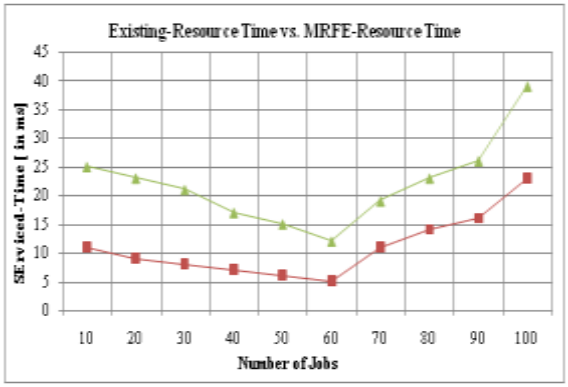

Figure:6 Job Resource Comparison with existing and proposed model

In Figure-10, jobs in Server2 are split into 3 sets in which each set is assigned for batch jobs. 2nd set of batch job is resourced by MRFE model gradually. Each job is resourced and resource time is logged in the resource log and compiled log.

7. Conclusion

This paper work deals about the efficient way of resource allotment from job scheduling to resource departed through Master ResourceFinding Engine. The research includes how effectively job scheduling is possible in case of Cloud environment. Users request for the resource and request shall be of any form, based on the request server filters is applied to segregate the useful information from the requirement and given to the scheduler. Job scheduler used here is a dynamic distributed environment based, in which job is scheduled by client to servers. The actual progress of jobs is resourced in the servers for the most possible extent. MRFE is a special unit which connects with job scheduler, optimizer and semantic model. Optimizer feeds the direct or simple input to the unit; job scheduler gives the filtered semantic to the MRFE based on the interactive jobs or normal jobs considered. Semantic model runs a pattern matching algorithm for the considered data, and gives the mere semantic ranking factor to the unit. Based on the input data given by sub units, MRFE runs a protocol which gives the metrics like distance where job is considered for prime, state matching, behavior matching and meta-data factor. All these values if given to adjudicator, it process the data with the decision making algorithm runs on MRFE and provides the output for each job assigned to MRFE unit. Say phase1 layer request for user with minimal requirements, request server raises the value and via optimizer it gives to MRFE, same time job scheduler gives the differentiated jobs to consider and the same is forwarded to semantic model. This model runs match algorithm for each job assigned by MRFE and returns the rank factor and all this is assigned to adjudicator, based on the internal metrics each job is assigned to get resource to meet the near exact requirement which user is placed. Here provision is given to make a resource log which is handled by storage resource manager. It segregates the meta-data, compilation inputs and resource logs and stores in the respective databases for efficient archive management.

In this paper, the algorithm currently focuses to provide interface for end to end connectivity in a Cloud environment. Performance analysis wise semantic finding takes time to comply the match with each and every request. In future the mechanism shall introduce simple way of semantic derivatives for pattern matching along with all the metrics considered to meet the end user requirement.

References

[1] Jian Su; Wei Guo; "A survey of service discovery protocols for mobile ad hoc networks"; Communications and Systems, pp: 398-404, 2008.

[2] Feng Zhu, Matt W. Mutka and Lionel M. Ni, “Service Discovery inPervasive ComputingEnvironments”, Published by the IEEE CS and IEEE Com. Soc., IEEE, 2005.

[3] The Salutation Consortium Inc., “Salutation Architecture Specification Part 1, Version 2.1 Edition,” http://www.salutation.org, 1999.

[4] F. Casati, S. Ilnicki, L. Jin, V. Krishnamoorthy, and M. Shan, “Adaptive and Dynamic Service Composition in eFlow”, Technical Report, HPL-200039, Software Technology Laboratory, 2000.

[5] H. Chen, A. Joshi, and T. Finin, “Dynamic Service Discovery for Mobile Computing: Intelligent Agents Meet Jini in the Aether,” Baltzer Science J. Cluster Computing, special issue on advances in distributed and mobile systems and comm., 2001.

[6] T. Hodes et al., “An Architecture for a Secure Service Discovery Service,” Proc. Fifth Int’l Conf. Mobile Computing and Networks, 1999.

[7] R.H. Katz, E.A. Brewer, and Z.M. Mao, “Fault-Tolerant, Scalable,Wide-Area Internet Service Composition,” Technical ReportUCB/CSD-1-1129, CS Division, EECS Department, Univ. of California, Berkeley, 2001.

[8] D. Mennie and B. Pagurek, “An Architecture to Support DynamicComposition of Service Components,” Proc. Fifth Int’l WorkshopComponent-Oriented Programming (WCOP), 2000.

[9] V.Sundramoorthy, Hans Scholten, “Challenges In the At Home Anywhere (@HA) Service Discovery Protocol”, Distributed and Embedded Systems group, Faculty of Computer Science-University of Twente.

[10] F. Hanssen, P. Hartel, T. Hattink, P. Jansen and J. Scholten, J. Wijnberg. A Real-Time Ethernet Network atHome.Proceedings Work-in-Progress session fourteenth Euromicro international conference on real-timesystems, Vienna, Austria, pp. 5-8, 2002.

[11] K. Arnold, B. Osullivan, R.W. Scheifler, J. Waldo, and A. Wollrath, The Jini Specification (The Jini Technology). Reading, Mass.: Addison-Wesley, June 1999.

[12] The Salutation Consortium Inc., “Salutation Architecture Specification Part 1, Version 2.1 Edition,” http://www.salutation.org, 1999.

[13] Bluetooth SIG, “Specification,” http://bluetooth.org, 2004.

[14] Dipanjan Chakraborty, AnupamJoshi, Yelena Yesha, and Tim Finin, “Toward Distributed Service Discovery in Pervasive Computing Environments”, IEEE Transactions On Mobile Computing, VOL. 5, NO. 2, 2006.