International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 24 Issue 1 – JANUARY 2017.

DESIGNING CONTENT-BASED PUBLISH/SUBSCRIBE SYSTEMS FOR RELIABLE MATCHING SERVICE

BANDARU SRI JANANI[#1] and VEMAGIRI PREMKUMAR[*2]

[#] PG Scholar, Kakinada Institute Of Engineering & Technology Department of Computer Science , JNTUK,A.P, India.

[*] Assistant Prof, Dept of CSE, Kakinada Institute of Engineering & Technology, JNTUK, A.P, INDIA.

***Abstract*— Security is one of the extensive and complicated requirements that need to be provided in order to achieve few issues like confidentiality, integrity and authentication. In a content-based publish/subscribe system, authentication is difficult to achieve since there exists no strong bonding between the end parties. Similarly, Integrity and confidentiality needs arise in published events and subscription conflicts with content-based routing. The basic tool to support confidentiality, integrity is encryption. In this paper, we propose SREM, a scalable and reliable event matching service for content-based pub/sub systems in cloud computing environment. To achieve low routing latency and reliable links among servers, we propose a distributed overlay SkipCloud to organize servers of SREM. Through a hybrid space partitioning technique HPartition, large-scale skewed subscriptions are mapped into multiple subspaces, which ensures high matching throughput and provides multiple candidate servers for each event. Moreover, a series of dynamics maintenance mechanisms are extensively studied.**

Index Terms*— Publish/subscribe, event matching, overlay construction, content space partitioning, cloud computing.

I. INTRODUCTION

Common requirement for any system is security. The need for security must be extremely high. It is one of the major requirements to protect or control any sort of failures [1]. There are number of mechanisms which are available to provide security. In that one of the most important mechanisms is encryption. In cryptography encryption is the process of converting plain text to cipher text which is unreadable from unauthorized users. The cryptography mechanism is required in publish/subscribe system. In publish/subscribe system publisher is one who publishes his content without specifying a particular destination to reach publisher will not program the documents to be delivered to a particular subscriber. Publisher will classify publishing documents based on different criteria and release it and subscriber will show interest on one or more documents and subscribe to that particular one in order to have access over it. This publish/subscribe system is traditionally carried out in broker-less [2] content based routing which forwards or routes the message based on the content of the message instead of clearly routing to an specified destination.

Publish-subscribe middleware has recently become popular because of its asynchronous, implicit, multi-point, and peer-to-peer style of communication. Components in a publish-subscribe system are strongly decoupled: they can be easily replaced, thus providing a high degree of flexibility both at the application and infrastructure level [3]. A number of publish-subscribe systems have been proposed to date. In this paper we focus on those that seek increased scalability and flexibility by exploiting a distributed architecture for event dispatching, and that empower the programmer with maximum expressiveness by using a content-based scheme for determining the match between an event and a subscription.

Representative examples are [4, 5 and 6]. Although the publish-subscribe model enjoys a growing popularity, we observe that the characteristics of the available systems still fall short of expectations under many respects. For instance, this paper is motivated by the observation that the reliability of the distributed event dispatching infrastructure is rarely guaranteed by dedicated mechanisms: instead, it is typically delegated to the underlying transport protocol, e.g., by assuming the existence of TCP links [7]. Unfortunately, this approach is overly restraining in several scenarios, including simple ones characterized by small scale and a static network topology. For instance, communication can be implemented on top of unreliable transport protocols like UDP for performance reasons; moreover, links and nodes of the dispatching infrastructure may fail altogether. Clearly, the situation is exacerbated in the more dynamic scenarios that are increasingly characterizing modern distributed computing, where publish-subscribe would find its natural use. As an example, mobile computing implies a continuously changing network topology, where reliable links are often difficult to maintain and where the event dispatching infrastructure is itself continuously recon- figured, providing an additional source of event loss [8].

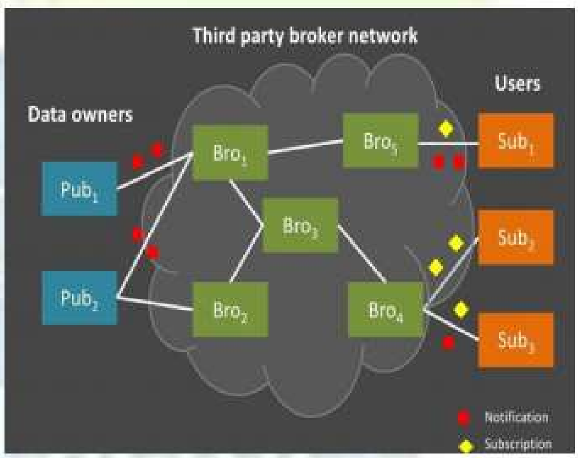

Fig.1 Subscriber/Publisher System

Content based routing applies some set of rules to its content to find the users who are interested in its content. Its different nature is helpful for huge-level scattered applications and also provides a high range of flexibility and adaptability to change. Authorized publisher have permission to publish events in the network and similarly subscribers who likes the content can gets subscribed to a particular published content and have access over it by which high level access control [9] can be achieved. Here published content should not be exposed to routing infrastructure and subscribers should receive content without leaking subscription identity to the system, which is a highly challenging task which needs to be carried out in content-based pub/sub system. Publisher and subscriber are the two entities and they do not trust each other. Even though authorized publisher publish events, nasty publisher pretend to be the real publisher and may spam the network with fake and duplicate contents similarly subscribers are very much eager to find other users and publishers which are challenging tasks [9]. Finally, Transport Layer Security (TLS) or Secure Socket Layer (SSL) is secure channels for distributing keys from key server to the required. Existing security approach deals with traditional network and security is based on restricted manner which tells about key word matching [10]. Key management was the challenging task in the existing approach, so to overcome all these, we use new approach called pairing based cryptography mechanism, which helps in mapping between to end parties so called cryptographic groups. Here, Identity Based Encryption Technique (IBE) [11] is used under this mechanism. New approach IBE provide greater concern towards authentication and confidentiality in the network. Our approach permit users to preserve credentials based on their subscriptions. Secret keys provided to the users are labeled with the credentials. In Identity-based encryption (IBE) mechanisms 1) key can be used to decrypt only if there is match between credentials with the content and the key; and 2) to permit subscribers to check the validity of received contents. Moreover, this approach helps in providing fine-grained key management, effective encryption, decryption operations and routing is carried out in the order of subscribed attributes.

II. LITERATURE SURVEY

There are two entities in the System publishers and subscribers. Both the entities are computationally bounded and do not trust each other. Moreover, all the peers (publishers or subscribers) participating in the pub/sub overlay network are honest and do not deviate from the designed protocol. Likewise, authorized publishers only allow valid events in the system. However, malicious publishers may masquerade the authorized publishers and spam the overlay network with fake and duplicate events. We do not intend to solve the digital copyright problem; therefore, authorized subscribers do not reveal the content of successfully decrypted events to other subscribers.

A. PUBLISHER SUBSCRIBER TECHNIQUE

Publishers and subscribers interact with a key server. They provide credentials to the key server and in turn receive keys which fit the expressed capabilities in the credentials. Subsequently, those keys can be used to encrypt, decrypt, and sign relevant messages in the content based pub/sub system, i.e., the credential becomes authorized by the key server. A credential consists of two parts: 1) a binary string which describes the capability of a peer in publishing and receiving events, and 2) a proof of its identity [12].

B. IDENTITY BASED ENCRYPTION

Identity(ID)-based public key cryptosystem, which enables any pair of users to communicate securely without exchanging public key certificates, without keeping a public key directory, and without using online service of a third party, as long as a trusted key generation center issues a private key to each user when he first joins the network [13].

C. IDENTITY HANDLING

Identification provides an essential building block for a large number of services and functionalities in distributed Information systems. In its simplest form, identification Is used to uniquely denote computers on the Internet By IP addresses in combination with the Domain Name System (DNS) as a mapping service between symbolic Names and IP addresses. Thus, computers can conveniently Be referred to by their symbolic names, whereas, in The routing process, their IP addresses must be used.[3] Higher-level directories, such as X.500/LDAP, consistently Map properties to objects which are uniquely identified by Their distinguished name (DN), i.e., their position in the X.500 tree [14].

D. CONTENT BASED PUBLISH/SUBSCRIBE

Content based networking is a generalization of the content based publish/subscribe model. In content-based networking, messages are no longer addressed to the communication endpoints. Instead, they are published to a distributed information space and routed by the networking sub -state to the “interested” communication end-points. In most cases, the same substrate is responsible for realizing naming, binding and the actual content delivery [15].

E. SECURE KEY EXCHANGE

A key-exchange (KE) protocol is run in a network of interconnected parties where each party can be activated to run an instance of the protocol called a session [16]. Within a session a party can be activated to initiate the session or to respond to an incoming message. As a result of these activations, and according to the specification of the protocol, the party creates and maintains a session state, generates outgoing messages, and eventually completes the session by outputting a session key and erasing the session state [17].

F. BLUE DOVE

It adopts a single-dimensional partitioning technique to divide the entire spare and a performance-aware forwarding scheme to select candidate matcher for each event. Its scalability is limited by the coarse-grained clustering technique [18].

G. SEMAS

It proposes a fine-grained partitioning technique to achieve high matching rate. However, this partitioning technique only provides one candidate for each event and may lead to large memory cost as the number of data dimensions increases. In contrast, HPartition makes a better trade-off between the matching throughput and reliability through a flexible manner of constructing logical space [19].

III. EXISTING SYSTEM

A number of pub/sub services based on the cloud computing environment have been proposed, However, most of them can not completely meet the requirements of both scalability and reliability when matching larger scale live content under highly dynamic environments. This mainly stems from the following facts: Most of them are inappropriate to the matching of live content with high data dimensionality due to the limitation of their subscription space partitioning techniques, which bring either low matching throughput or high memory overhead [20]. These systems adopt the one-hop lookup technique among servers to reduce routing latency. In spite of its high efficiency, it requires each dispatching server to have the same view of matching servers. Otherwise, the subscriptions or events may be assigned to the wrong matching servers, which bring the availability problem in the face of current joining or crash of matching servers. Matching servers. Otherwise, the subscriptions or events may be assigned to the wrong matching servers, which bring the availability problem in the face of current joining or crash of matching servers.

A. DISADVANTAGES

1. Lower rate of scalability and reliability of event matching.
2. High routing Latency

IV. PROPOSED SCHEME

We propose a scalable and reliable matching service for content-based pub/sub service in cloud computing environments, called SREM. Specifically, we mainly focus on two problems: one is how to organize servers in the cloud computing environment to achieve scalable and reliable routing. The other is how to manage subscriptions and events to achieve parallel matching among these servers. We propose a distributed overlay protocol, called Skip Cloud, to organize servers in the cloud computing environment. Skip Cloud enables subscriptions and events to be forwarded among brokers in a scalable and reliable manner. Also it is easy to implement and maintain.

A. ADVANTAGES

1. High scalability and reliability of event matching
2. Reducing the optimal routing latency.

Scope is to design and implement the elastic strategies of adjusting the scale of servers based on the churn workloads. Secondly, it does not guarantee that the brokers disseminate large live content with various data sizes to the corresponding subscribers in a real-time manner. For the dissemination of bulk content, the upload capacity becomes the main bottleneck. Based on our proposed event matching service, we will consider utilizing a cloud-assisted technique to realize a general and scalable data dissemination service over live content with various data sizes.

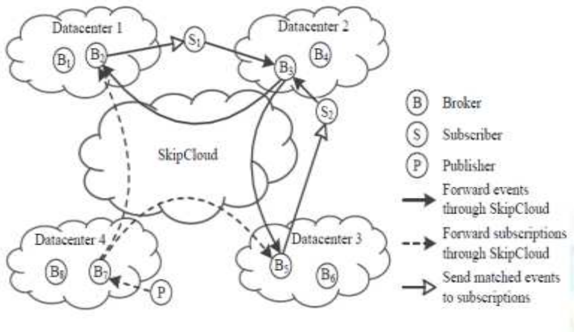

Fig.2 Proposed Architecture

To support large-scale users, we consider a cloud computing environment with a set of geographically distributed data centres through the Internet. Each data center contains a large number of servers (brokers), which are managed by a data center management service such as Amazon EC2 or Open Stack. All brokers in SREM as the front-end are exposed to the Internet, and any subscriber and publisher can associate to them unswervingly. To accomplish reliable connectivity and low routing latency, these brokers are connected through an distributed overlay, called Skip Cloud. The entire content space is partitioned into disjoint subspaces, each of which is managed by a number of brokers. Subscriptions and events are dispatched to the subspaces that are overlapping with them through Skip Cloud. Subscriptions and events falling into the same subspace are matched on the same broker. After the matching process completes, events are broadcasted to the corresponding interested subscribers. The subscriptions generated by subscribers S1 and S2 are dispatched to broker B2 and B5,respectively. Upon receiving events from publishers, B2 and B5 will send matched events to S1 and S2, respectively.

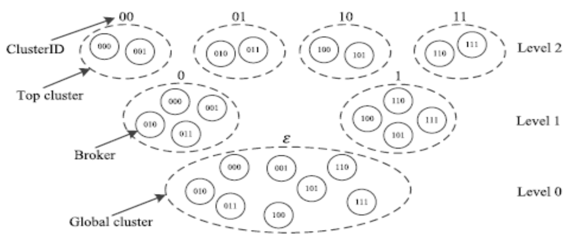

Fig.3 SkipCloud

Every broker is acknowledged by a binary string.

All data canters due to the various skewed distributions of users' interests. The node failure may lead to unreliable and inefficient routing among servers. To this end, it is organized servers into Skip Cloud to reduce the routing latency in a scalable and reliable manner. Such a framework offers a number of advantages for real-time and reliable data dissemination. First, it allows the system to timely group similar subscriptions into the same broker due to the high bandwidth among brokers in the cloud computing environment, such that the local searching Time can be greatly reduced. Second, since each subspace is managed by multiple brokers, this framework is fault tolerant even if a large number of brokers crash straightaway. Third, because the data center management service provides scalable and elastic servers, the system can be easily expanded to Internet-scale.

B. HPARTITION

In order to take benefit of multiple distributed brokers, SREM distributes the entire content space among the top clusters of Skip Cloud, so that each top cluster only switches a subset of the entire space and searches a small number of candidate subscriptions. SREM employs a hybrid multidimensional space partitioning technique, called HP partition, to realize scalable and reliable event matching. Generally speaking, HPartition divides the entire content space into disjoint subspaces. Subscriptions and events with overlapping subspaces are dispatched and matched on the same top cluster of Skip Cloud .To keep workload balance among servers, HPartition divides the hot spots into various cold spots in an adaptive manner.

C. ADAPTIVE SELECTION ALGORITHM

Because of diverse distributions of subscriptions, both HSPartition and SSPartition cannot substitute with each other. HSPartition is striking to divide the hot spots whose subscriptions are uniform dispersed regions. However, it's unsuitable to rift the hot spots whose subscriptions all appear at the same exact point. On the other hand, SSPartition allows to divide any kind of hot spots into multiple subsets even if all subscriptions falls into the same single point. Nevertheless, compared with HSPartition, it has to dispatch an event to multiple subspaces, which brings a higher traffic overhead. To accomplish balanced workloads among brokers, An adaptive selection algorithm to select either HSPartition or SSPartition to assuage hot spots. The selection is based on the similarity of subscriptions in the same hot spot. Specifically, subspace with maximal size of subscriptions in HSPartition. We choose HSPartition as the partitioning algorithm through combining both partitioning techniques, this selection algorithm can alleviate hot spots in an adaptive manner

V. IMPLEMENTATION

The proposed system of this project is divided into five major modules and described as below.

1. DATACENTER / BROKER CREATION
2. CLUSTERING METHOD
3. CONTENT SPACE PARTITIONING
4. EVENT MATCHING
5. ROUTING METHOD

A. MODULES DESCSRIPTION

1) DATACENTER / BROKER CREATION

In the first module, we develop the Data center creation and Broker Creation. To support large-scale users, we consider a cloud computing environment with a set of geographically distributed data centres. Each data center contains a large number of servers (brokers), which are managed by a data center management service. Our approach is suitable for large and reasonably stable environments such as that of an enterprise or a data center, where reliable publication delivery is desired in spite of failures. As future work, we would like to exploit our scheme to allow for multi-path load balancing, and support some of P/S optimization techniques such as subscription covering. It provides an abstract and high level interface for data producers (publishers) to publish messages and consumers (subscribers) to receive messages that match their interest.

2) CLUSTERING METHOD

Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster. Suppose we are given a database of 'n' objects and the partitioning method constructs 'k' partition of data. Each partition will represent a cluster and $k \leq n$. It means that it will classify the data into k groups, which satisfy the following requirements:

- Each group contains at least one object.
- Each object must belong to exactly one group.

3) CONTENT SPACE PARTITIONING

The content space is partitioned into disjoint subspaces, each of which is managed by a number of brokers. Then each top cluster only handles a subset of the entire space and searches a small number of candidate subscriptions. The whole content space into non-overlapping zones based on the number of its brokers. After that, the brokers in different cliques who are responsible for similar zones are connected by a multicast tree.

4) EVENT MATCHING

The data replication schemes are employed to ensure reliable event matching. For instance, it advertises subscriptions to the whole network. When receiving an event, each broker determines to forward the event to the corresponding broker according to its routing table. These approaches are inadequate to achieve scalable event matching.

5) ROUTING METHOD

The routing process usually directs forwarding on the basis of routing tables, which maintain a record of the routes to various network destinations. Thus, constructing routing tables, which are held in the router's memory, is very important for efficient routing. Most routing algorithms use only one network path at a time. Multipath routing techniques enable the use of multiple alternative paths. Prefix routing in Skip Cloud is mainly used to efficiently route subscriptions and events to the top clusters. Note that the cluster identifiers at level are generated by appending one binary to the corresponding clusters at level i. The relation of identifiers between clusters is the foundation of routing to target clusters.

Briefly, when receiving a routing request to a specific cluster, a broker examines its neighbour lists of all levels and chooses the neighbour which shares the longest common prefix with the target Cluster ID as the next hop. The routing operation repeats until a broker cannot find a neighbour whose identifier is more closer than itself.

Fig .4 Scalable and Reliable Matching Service

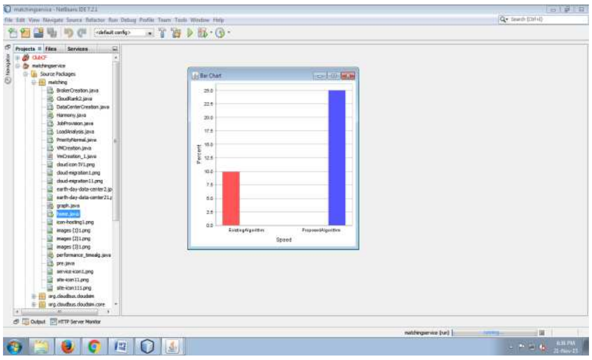

Fig .5 Comparison Graph

VI. CONCLUSION

SREM, a scalable and reliable event matching service for content-based pub/sub systems in cloud computing environment. SREM attaches the brokers over and done with a scattered overlay Skip Cloud, which certifies reliable connectivity among brokers through its multi-level clusters and brings a low routing latency through a prefix routing algorithm. A hybrid multi-dimensional space partitioning technique, helps out SREM in reaching scalable and balanced clustering of high dimensional twisted subscriptions, and each event is permitted to be matched on any of its candidate servers. Extensive experiments with real deployment based on a Cloud Stack test bed are accompanied, producing results which demonstrate that SREM is effective and practical, and also presents good workload balance, scalability and reliability under various parameter settings. Although proposed event matching service can competently filter out extraneous users from big data volume, there are still a number of problems need to be solved. Based on this event matching service, it is considered utilizing a cloud-assisted technique to realize a general and scalable data dissemination service over live content with several data sizes.

REFERENCES

[1] Dataperminite. (2014). [Online]. Available: http://www.domo.com/blog/2012/06/how-much-data-is-created-every-minute/

[2] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, “Characterizing user behavior in online social networks,” in Proc. 9th ACM SIGCOMM Conf. Internet Meas. Conf., 2009, pp. 49–62.

[3] A. Carzaniga, “Architectures for an event notification service scalable to wide-area networks,” Ph.D. dissertation, Ingegneria Informatica e Automatica, Politecnico di Milano, Milan, Italy, 1998.

[4] P. Eugster and J. Stephen, “Universal cross-cloud communication,” IEEE Trans. Cloud Comput., vol. 2, no. 2, pp. 103–116, 2014. [5] R. S. Kazemzadeh and H.-A Jacobsen, “Reliable and highly available distributed publish/subscribe service,” in Proc. 28th IEEE Int. Symp. Reliable Distrib. Syst., 2009, pp. 41–50.

[5] Y. Zhao and J. Wu, “Building a reliable and high-performance content-based publish/subscribe system,” J. Parallel Distrib. Comput., vol. 73, no. 4, pp. 371–382, 2013.

[6] F. Cao and J. P. Singh, “Efficient event routing in content-based publish/subscribe service network,” in Proc. IEEE INFOCOM, 2004, pp. 929–940.

[7] S. Voulgaris, E. Riviere, A. Kermarrec, and M. Van Steen, “Sub-2-sub: Self-organizing content-based publish and subscribe for dynamic and large scale collaborative networks,” Res. Rep. RR5772, INRIA, Rennes, France, 2005.

[8] I. Aekaterinidis and P. Triantafillou, “Pastrystrings: A comprehensive content-based publish/subscribe DHT network,” in Proc. IEEE Int. Conf. Distrib. Comput. Syst., 2006, pp. 23–32.

[9] R. Ranjan, L. Chan, A. Harwood, S. Karunasekera, and R. Buyya, “Decentralised resource discovery service for large scale federated grids,” in Proc. IEEE Int. Conf. e-Sci. Grid Comput., 2007, pp. 379–387.

[10] A. Gupta, O. D. Sahin, D. Agrawal, and A. El Abbadi, “Meghdoot: Content-based publish/subscribe over p2p networks,” in Proc. 5th ACM/IFIP/USENIX Int. Conf. Middleware, 2004, pp. 254–273.

[11] X. Lu, H. Wang, J. Wang, J. Xu, and D. Li, “Internet-based virtual computing environment: Beyond the data center as a computer,” Future Gener. Comput. Syst., vol. 29, pp. 309–322, 2011.

[12] Y. Wang, X. Li, X. Li, and Y. Wang, “A survey of queries over uncertain data,” Knowl. Inf. Syst., vol. 37, no. 3, pp. 485–530, 2013. [14] W. Rao, L. Chen, P. Hui, and S. Tarkoma, “Move: A large scale keyword-based content filtering and dissemination system,” in Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst., 2012, pp. 445–454.

[13] M. Li, F. Ye, M. Kim, H. Chen, and H. Lei, “A scalable and elastic publish/subscribe service,” in Proc. IEEE Int. Parallel Distrib. Process. Symp., 2011, pp. 1254–1265.

[14] X. Ma, Y. Wang, Q. Qiu, W. Sun, and X. Pei, “Scalable and elastic event matching for attribute-based publish/subscribe systems,” Future Gener. Comput. Syst., vol. 36, pp. 102–119, 2013.

[15] A. Lakshman and P. Malik, “Cassandra: A decentralized structured storage system,” Oper. Syst. Rev., vol. 44, no. 2, pp. 35–40, 2010.

[16] M. Sathiamoorthy, M. Asteris, D. Papailiopoulos, A. G. Dimakis, R. Vadali, S. Chen, and D. Borthakur, “Xoring elephants: Novel erasure codes for big data,” in Proc. 39th Int. Conf. Very Large Data Bases, 2013, pp. 325–336.

[17] S. Voulgaris, D. Gavidia, and M. van Steen, “Cyclon: Inexpensive membership management for unstructured p2p overlays,” J. Netw. Syst. Manage., vol. 13, no. 2, pp. 197–217, 2005.

[18] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, “Tapestry: A resilient global-scale overlay for service deployment,” IEEE J. Sel. Areas Commun., vol. 22, no. 1, pp. 41–53, Jan. 2004.

[19] Murmurhash. (2014). [Online]. Available: http://burtleburtle.net/bob/hash/doobs.html

BANDARU SRI JANANI, *is a student of Kakinada Institute Of Engineering & Technology affiliated to JNTUK, Kakinada pursuing M.Tech (Computer Science). Her Area of interest includes Cloud Computing and its objectives in all current trends and techniques in Computer Science.*

VEMAGIRI PREMKUMAR M.TECH *is working as Assistant Professor, Department of Computer Science & Engineering, Kakinada Institute of Engineering & Technology, JNTUK, A.P, INDIA.*